The design and construction of the control and arithmetic units of a demonstration electronic digital computer

Raymond H. Brunskill

The Design and Construction of the Control and Arithmetic units of a Demonstration Electronic Digital Computer.

Thesis submitted for the

Degree
of
Master of Science,
in the
University of Durham.

Raymond H Brunskill, C.Eng., M.I.E.E., M.Inst.M.C.

January 1969.

ACKNOWLEDGMENTS.

The author wishes to thank Professor D.A. Wright for his approval of the project and his co-operation in making available the facilities which have enabled its completion. It is pleasing to acknowledge the valuable assistance and guidance of Dr. M.J. Morant who, in addition to supervising this work, has always given encouragement and constructive criticism. Thanks are also due to Dr. B.J. Stanier for his generous counsel and practical aid, especially during the commissioning period.

The author is grateful to the Darlington Education Authority, and the Chairman and Members of the Governing Body of the Darlington College of Technology for their permission to undertake this work; to his Principal, Mr. C.E. Beynon, for his encouragement and active support; to Dr. H. Williams for his initial advice and subsequent co-operation; and to Mr. W.S. Wood for numerous mentorial services.

He is appreciative of the assistance given by: Mr. T. Nancarrow, Mr. L. Erskine and Mr. E.S. Collin in the construction of the machine; and Mrs. E. Brunskill, Mr. R.G. Smith and Mr. A. Lowcock in the production of the typescript.

This list of acknowledgments would be incomplete without mention of my wife, Mary, whose understanding and forbearance proved to be invaluable.

. . .

CONTENTS.

CONTENTS.

CONTENTS.

. . . . . .

CHAPTER I

INTRODUCTION

This thesis is concerned with the design and construction of a small demonstration digital computer. This computer has been built to assist in the teaching of computer electronics. It is not intended to be employed for teaching programming, for which full size computers are more suitable. The Department of Applied Physics has been interested in a project of this nature for some years. The digital computer is seen as an admirable machine on which to demonstrate some of the applications of modern electronic devices with which the department is so actively concerned.

It was obvious that to attempt to construct a machine of adequate specification using second generation components, (i.e. transistors etc.), was totally impractical. The physical effort involved in the assembly of a very large number of discrete circuit elements was considered to be too great for the facilities available and to give no guarantee of success. The difficulties of this form of construction were confirmed by examining the commercially available equipment for teaching computer electronics. Machines costing up to £2,000 (which is considered to be a normal upper limit

for most educational establishments) work only in the serial mode and are mostly limited to demonstrating basic arithmetic processes. The present project had in mind a more sophisticated computer, while still aiming at a possible commercial price limit of around £2,000.

The advent of the cheaper ranges of integrated circuits stimulated further examination of the possibility of constructing a small demonstration digital computer. The project was now much more attractive, since component requirements would be reduced by a factor of ten. The chances of success would be greatly increased owing to improved circuit reliability and the reduction in the number of actual circuit elements. Coupled with this was the confirmation that the cost of these devices was falling and that this trend was likely to continue for some time to come. The prospect of being among the first to apply integrated logic elements, to quite complex switching networks, was most exciting. The obvious enthusiasm was of course tempered by the knowledge that a fundamental error in design or construction would render the equipment virtually redundant.

It was against this background that the project started in September, 1966. The initial discussions,

involving Dr. M.J. Morant, Dr. B.J. Stanier, Mr. A. Smith and the author, were directed at establishing the general requirements of a demonstration digital computer. These are listed in section 2.4. Subsequent discussion led to the formulation of the detailed specification of section 5.1. This specification was drawn up in December, 1966 and approved during the following month. The project was divided into two parts at this stage. The control and arithmetic sections were allocated to the author and the memory section became the responsibility of Mr. A.Smith. This part of the work is the subject of a separate thesis (1).

This thesis records the events, since September, 1966, which have led to the production of the control and arithmetic sections of the demonstration computer. The design, construction and commissioning associated with these sections are, with a small amount of technical assistance, entirely the work of the author. The project has been carried out under the supervision of Dr.M.J. Morant and Dr. B.J. Stanier. The author has, of course, enjoyed the general advice and encouragement of his supervisors.

The project was completed by stages, each of which may be identified with the appropriate sections of the

thesis. The detailed planning of these stages was considered vital to the success of the project. It was felt that the work could not be allowed to proceed without clearly defined terms of reference, involving component costs and time estimates of design and construction effort. A realistic approach to component costs, delivery dates, design effort, construction time, etc. enabled the completion dates for the various stages to be approximated. With few exceptions these estimates have proved to be fairly accurate.

The background work, which involved a study of available digital teaching equipment, current techniques and design procedures, is discussed in Chapters 2, 3 and 4. A detailed analysis of the accepted design is given in Chapter 5. The description of the detailed circuit design, Chapter 6, shows how the individual requirements of each register, timing function, control gate, etc. have been met using the techniques previously established. In Chapter 7 details are given of the methods and processes used in the actual construction. The test procedures and problems associated with persuading the machine to function are recorded in Chapter 8. Chapter 9 lists some typical laboratory excercises, to show how the machine is to be employed in the teaching situation. In

the final Chapter consideration is given to modifications which, it is believed, would improve the performance of the equipment.

CHAPTER II.

THE TEACHING COMPUTER - PRESENT AND FUTURE

2.1 Practical Training in Computer Electronics

It is necessary to examine the type of situation in which a demonstration digital computer would be of assistance. Practical training in the application of logic elements to complex switching networks is primarily the concern of applied science departments in universities and colleges of technology. The digital computer, employing all logic circuit techniques, is the ideal vehicle on which to gain practical experience. The function of such equipment is to demonstrate, clearly and in considerable detail, the various techniques likely to be encountered by prospective engineers and technicians whose later work will be associated with complex logic circuitry. It is considered that practical work, using equipment of the proposed design, could form a valuable part of courses in the following subjects.

(i) Electronics and Electronic Engineering.

It is now certain that students should have practical experience in the application of modern electronic devices to a wide range of digital techniques and in the analysis of complex switching networks.

(ii) Computing Science.

A detailed study of the central processor appears to form an essential part of such courses.

(iii) Measurement and Control.

It is apparent that digital methods are becoming well established in these courses.

(iv) Electronic Technician.

The construction and maintenance of industrial digital equipment forms an important part of the course.

2.2 Review of Present Commercial Equipment.

One of the first exercises undertaken in this project was to examine currently available equipment which has been designed solely for teaching purposes. The literature describing the majority of U.K. equipment was obtained from the manufacturers and, where possible, arrangements were made for actual demonstration. It was necessary to compare the relative merits and limitations in terms of effectiveness, flexibility and cost. The results of this review assisted in the determination of an acceptable specification for the proposed demonstration computer.

It was found that most of the equipments examined were described as being 'Digital Computers' whereas, in

fact, none adequately complied with the B.S. 3527 definition of a digital computer. The equipment reviewed here may be conveniently divided into two groups.

(i) Simple Logic Tutors.

(ii) More elaborate equipment to demonstrate arithmetic processes.

In addition, there are a number of full scale computer systems which are described as being suitable for teaching purposes, but owing to their relatively high cost, these are not considered in the present context.

2.2.1 Simple Logic Tutors.

(i) Lan Electronics Integrated Computer Training System.

Lan Electronics, probably the most active in the development of electronic training aids, marketed one of the first logic trainers (LAN-DEC 15), and has since developed a comprehensive range of teaching aids for the electronics and computing subjects. This first logic tutor is based on the use of NOR elements. A telephone dial, in conjunction with manual push-buttons, is provided to set up the input conditions to some of the fifteen available NOR gates. Monitoring of the state of logic elements is achieved by using six panel-mounted filament lamps.

Although this equipment is described, by the manufacturer, as being suitable for the teaching and demonstration of computing techniques, it is evident from the specifications of this company's more recent equipments that its limitations have been realised. These are as follows:- a) the inability to demonstrate the fundamental AND-OR relationships; b) the degree of complexity involved in assembling astable and bistable circuits, coupled with the risk of their malfunction; c) the comparatively small number of logic elements available, compared with those required to construct any of the basic combinational logic circuits, e.g. counter, shift register. It is worth noting that this equipment was conceived at a time when most of the industrial control equipment was designed around the use of NOR elements, and in this respect it may have been useful as a design aid in industrial situations.

The cost of this equipment is approximately £100 and includes the power unit, relay unit, patch cords and handbooks.

The succeeding equipment, LAN-DEC 100, Training Computer, is much more versatile. The front patching panel connects to eight multiway sockets, each of which accepts a printed circuit card. These cards can

incorporate any type of logic element; typically a card may include 20 gates or perhaps five bistable elements. Input conditions are set up by a telephone dial and toggle switches. Output conditions may be checked using the eight panel-mounted filament lamps.

The effectiveness of this equipment is directly related to the ability of the demonstrator to trouble-shoot incorrect student wiring. With so many front panel terminals available, it is necessary for each student to be thoroughly familiar with the dangers of connecting rail voltages to the output of gates and with the loading capacities of the elements. Another point worth noting is that if the student is able to employ such a large number of elements, as is implied by the number made available, he must surely have encountered counting, coding, timing circuits etc. It would doubtless be of advantage to provide two or three clock generators, and to make these permanently available on the front panel. The cost of the equipment is in excess of £320. Although its flexibility is recognised, the large number of interconnections required to assemble the circuits, the lack of continuous timing pulses and the relatively high cost, make this equipment less than ideal.

The most ambitious logic tutor advertised by this

company is the ROBTOM digital computer, which incorporates four separate panels to accommodate the logic elements. Thirtytwo filament lamps and their associated inverter amplifiers are provided on one panel, a second panel carries a 32 bit store and fast counter, while a third panel provides a number of two and four input NOR gates. Input conditions are set up on the 31 toggle switches which are assembled on the fourth panel. The equipment has the same basic limitations as the LAN DEC 100. It is therefore likely that equipment of this sort would be most useful in a digital control design laboratory where the large numbers of gates and storage facilities would be of advantage in breadboarding control schemes. It is considered that the equipment is too general and without any specific aim. Its cost is £650.

(ii) L.K. 250 Series of Logikits Feedback Ltd.

Each of the logikits obtainable from Feedback Ltd. comprises a base board and a number of lamps and sockets to accept plug-in logic elements. One feature of the range is that both AND and OR functions are available in addition to bistables, monostables, clock generators, pulse shapers and delay units. The degree of flexibility obtained with this equipment is high. Since only those elements required for a particular exercise are included

on the base board, students may cover a considerable range of digital techniques in very simple steps, without redundant terminations or symbols appearing. The handbook, for model L.K. 253, lists some twenty exercises which may be demonstrated. The author has no actual experience in the use of this equipment and cannot therefore comment on reliability and student-proofing. The concept of using only the required logic elements would appear to simplify the 'patching up' procedure, and the symbols appearing on each element provide an acceptable mimic diagram of the system under consideration.

The model L.K. 253, which has nine indicator lamps on the 32 socket base board together with 20 logic elements is listed at £325, whereas the model L.K. 255 with slightly inferior specifications (20 sockets, 6 Lamps, 15 elements) costs £133.

(iii) Logic Tutor - Systems Computers Ltd.

One feature of the Systems Computers Logic Tutor which distinguishes it from other logic tutors is in the method of interconnecting the logic elements. A detachable 336 way patch board may be pre-wired and checked before insertion in the tutor panel. There are 30 NAND gates available, four bistables and two buffer amplifiers. Six toggle switches may be used to set up the

input conditions and eight filament lamps are included to monitor the output conditions of the logic elements. The elaborate 336-way patch panel obviously accounts for much of the cost of £345.

(iv) Digital Logic Instrumentation Kit. Farnell Instruments Ltd.

Priced at £125, the Farnell Logic Kit comprises a base board and 26 plug-in units. Included are 6 NOR gates, 5 AND gates and 5 Binary units. It is aimed at introducing the student to the principles of digital control systems. The chief advantage is its simplicity of assembly, because only those elements required for a particular experiment need be assembled. There are no redundant switches, lamps and free terminations to interfere with the appreciation of the circuit under consideration. Additional units may be obtained to permit the study of larger combinational logic systems. It is possible to demonstrate the basic laws of Boolean algebra as both AND and OR elements may be incorporated. Multiple gating with these basic elements is not possible, unless the coupling between each pair of units is done with a buffer amplifier. It is considered that, apart from this limitation, the equipment is quite effective, easy to demonstrate and flexible, and that the chief aim of

demonstrating simple digital techniques has been achieved.

(v) Other Logic Breadboard Systems.

There are a number of other plug-in logic systems commercially available. These are primarily intended for use in development laboratories, and as they have not been designed solely for educational purposes they are not considered relevant to this discussion.

2.2.2. Digital Equipment to Demonstrate Arithmetic Processes.

(i) Educational Computer, Type B.T.1 - Elliott Bros. Ltd.

The Elliott Bros. computer, type B.T.1, designed in conjunction with the Battersea College of Advanced Technology, performs routing, control and arithmetic operations on binary numbers. Input of data is achieved by push-button and information monitoring is done using filament lamps. A multiway rotary switch is used to select any of the seven machine instructions. Timing pulses are available at the rate of 5kHz, 1Hz or by manual push-button. The machine operates in the serial mode, on pure binary numbers, with a 16 bit word. The main combinational logic circuits incorporated are three 16 bit shift registers, a single bit full adder and a 4 bit binary counter, which is used to generate the shift pulses. The instructions are executed in the serial mode requiring

16 of the available 20 pulses allocated to each word time. Subtraction is performed by the complement method, multiplication by addition and right shifting, but there is no instruction for division.

In February, 1968 it was announced that this equipment was to be part of a complete computer to be manufactured by Feedback Ltd. The specification of this new equipment, ABACUS Educational Computer Ec360/370, indicates that it is now in two sections, an Arithmetic unit, very similar to the type B.T.1. and a Control and Store unit which incorporates a 256 word ferrite core store. The normal operating speed is quoted at 2,500 p.p.s., although an unspecified slow speed is included. The control unit, when operating at slow speed, can show the arithmetic and control processes associated with serial operation. It appears from photographs, that the circuitry is built up using discrete components which obviously accounts for much of the cost of £2,950.

(ii) Teaching Computer. Telefunken Ltd.

Although the Telefunken equipment is manufactured outside the U.K., it is considered here because it demonstrates clearly the link between arithmetic and memory functions. The equipment incorporates a control

unit, an arithmetic unit and a store. The control unit consists of 11 push-button switches to select the required machine instruction and a telephone dial to generate the timing pulses. There are six storage cells, each storing one 8-bit word. Examination of the interrogated storage location is achieved using a relay driven lamp matrix.

The arithmetic operations, add and subtract, are performed using 2 eight-bit shift registers and a single bit full adder, together with suitable gating circuitry. The display panel carries a mimic diagram with lamps to illustrate the operation of all registers and control gates.

Although this equipment appears to be good, from the demonstration point of view, there is no facility for slow automatic operation. If this were available, the student could then become familiar with the machine cycle before assuming the role of the control unit. There is no attempt to demonstrate those instructions dealing with 'jumps' in the programme. The demonstration of binary multiplication and division is difficult, owing to the large number of timing pulses required for serial operation.

This equipment is priced at approximately £800. It is somewhat expensive, considering its component parts; but the layout, which gives a very clear appreciation of

the demonstration, to some extent off-sets this cost.

(iii) Teaching Computer. Mullard Educational Services Ltd.

The Mullard Educational Computer is a suggested design which can be built in five stages. The first stage provides for the addition and subtraction of 2 eight-bit binary numbers in the serial mode, either under manual control or at a clock rate of 20 kHz. A slow automatic speed of 1 Hz is also provided. The succeeding stage carries out multiplication and division by the inclusion of extra switches and logic circuitry. The stage three computer incorporates two additional eight-bit storage registers to replace the manual switches of the stage two equipment. The modifications required to progress to stages four and five are still under development. Preliminary information suggests that a 16-bit word will be used, together with data and programme-storage facilities using one or more 160 bit delay lines. The equipment which is currently available does not include display or control panels, which must be designed and constructed by the purchaser. All logic circuits are built on plug-in printed circuit cards using M.E.L. series two elements.

It is considered that this equipment has evolved as a side-issue from development exercises in the application

of this company's range of 'series two' logic elements. Due consideration has not been given to those primary requirements, a clear display, a comprehensive mimic diagram and manual operating facilities. The cost of the stage one unit is approximately £200.

(iv) Cedus 201. Computer Engineering Ltd.

The Cedus computer training system is based on unit construction and built to the user's specification. Each unit is capable of manual operation without reference to any other unit. The machine operates in the serial mode using a 16-bit word. Arithmetic operations are performed using three 16-bit shift registers in conjunction with the single-bit adder unit. The pulse generator unit incorporates a five stage counter and decoding gates to produce the 16 discrete timing pulses associated with each word. The counter operates at any frequency up to 100kHz and a sinewave of this frequency is generated internally. Clock pulses are generated either internally from the oscillator at this frequency, or by manual push-button. It is also possible to couple a separate pulse generator to the equipment. A plugboard unit is provided to 'patch up' any of the 16 machine instructions, in any order, to form a programme. The memory unit comprises a 1,024 word drum store with associated timing

pulse amplifiers. Interface units permit the use of tape readers and line printers. The average price of the 15 units , not including the £1,500 drum store, is £240.

It is possible to assemble these units to demonstrate most of the salient features of a digital computer system The manual operating facilities allow each step in the programme to be examined before proceding to the next instruction. Filament lamps are used to indicate the flow of data through the machine. It is expected that equipment of this nature would provide the basis of a good demonstration computer, although it is extremely expensive and it operates serially, which is not in accordance with current practice on full scale computers.

2.2.3 Conclusions on Commercially Produced Teaching Computers.

It is evident from this review of equipment that there are two stages involved in the teaching of computer electronics. Units similar to those described in Section 2.2.1 are both necessary and appropriate to give the initial training in logic applications. The range of this equipment and the relatively small costs involved make it unnecessary for university and college departments to undertake special development to build their own equipment of this type. In the case of the more sophisticated

computer trainers it is seen that a considerable outlay, approximately £3,000, is required to obtain even a modest system which is suitable for demonstration work. The most modern of these equipments, ABACUS EC360/370, still employs second generation components and operates in the serial mode.

With reference to the equipment reviewed, it is apparent that the trend has been to adopt a 16 bit word and to provide a minimum of 120 words of storage. Manual control is provided in most cases and facilities are normally made available for interfacing peripheral equipment.

2.3 Review of Literature on Computer Electronics Teaching Equipment

It is the intention, in this section, to establish the stages that appear to be currently adopted in the practical teaching of computing techniques. There have been a number of papers, notably in the Journal of Electrical Engineering Education, describing digital computing teaching aids which have been constructed in Schools, Colleges of Technology and Universities.

In the case of school projects (e.g. 2,3) it is apparent that the limitations of available time, finance and experience have resulted in equipment which is

generally unsuitable for wider use. The tendency has been to produce equipment which will perform arithmetic operations, without prior reference to any teaching aids related to the basic logic functions. It was decided that reports of school projects were of little assistance in the determination of an acceptable specification for a demonstration computer for higher education.

There appear to be only a few references to more advanced machines. One practical aid has been described by Green (4). This incorporates the basic logic functions AND, OR and NOT. The display panel permits access to the terminals of six bistable elements and to the basic gating functions. Filament lamps are used to monitor the state of some of the logic elements, and toggle switches are are used to set up the input conditions. Pre-wired plug-in boards are included, as a special feature, to eliminate the need for extensive re-wiring of the front panel between demonstrations. It is reported that equipment of this sort is a useful aid in the teaching of undergraduate courses, and is seen to prepare students for the design of the complex circuits associated with arithmetic and control operations.

A paper by Hurst (5) describes a logic tutor which utilizes ten identical logic display panels and one master

panel, within the framework of a single unit. Each of the logic display panels provides access to: one of each of the four gating elements AND, OR, NOR and NAND, a bistable element, a buffer amplifier and a filament lamp. The master panel incorporates a telephone dial, a voltmeter and three clock generators. In order to reduce front panel wiring, six of the more complex circuits, e.g. shifting and B.C.D. counting, are pre-wired and selected by means of a rotary switch.

The Low Cost Digital Computer for Teaching described by Abrahams (6) has already been dealt with in Section 2.2.1 and is therefore not given further consideration here.

The main conclusions to be drawn from these articles are:

(i) The initial use of the basic AND and OR functions.

(ii) That teaching aids of this sort are "stage one" in preparing students to deal with the more complex operations of a full computer.

(iii) That front panel wiring should be minimised as it is time-consuming and detracts from the main objective

Most of the textbooks dealing with computer fundamentals provide a good indication of the various stages involved in the understanding of computing techniques. It is not the intention to discuss these books here, although it should be

noted that they all show the same sort of pattern in developing the subject. This appears to be as follows:

(i) Arithmetic / algebra of the binary system.

(ii) Application of electronic devices to switching circuits.

(iii) A detailed treatment of each of the major units, arithmetic, control and memory.

(iv) A description of the design features of either a hypothetical machine (e.g. 7,8) or an actual general purpose computer.

(v) The applications of digital computers, including notes on programming.

It is therefore considered that the logic tutors, referred to in Section 2.2.1, support the teaching and appreciation of topics i and ii. The more complex equipment, similar to that described in Section 2.2.2, is required when dealing with topics iii and iv.

2.4 General Requirements of a Machine for Teaching Computer Electronics

The review of the commercial equipment and of the literature seems to show considerable deficiencies in previous thought on the production of equipment for teaching computer electronics. It has, however, enabled the general requirements to be specified in a rather more precise form and, at this stage, it is desirable to list these since they

are the basis of the design described in the remainder of this thesis.

(i) The requirement appears to be for a computer (9) as distinct from a complex logic trainer. This implies automatic operation of machine instructions and the ability to modify the sequence of instructions.

(ii) It should be able to demonstrate the operations of the machine in much more detail than is possible with a full-size computer. To this end a large display panel is essential with lamps to indicate the flow of information and the state of all major control gates. It should be possible to gain access to the entire logic circuitry for more detailed examination. Push-button switches should enable the machine to be operated manually.

(iii) To enable the operation of the machine to be followed it should be capable of very slow, 1 bit/sec. and manual operation as well as automatic operation at moderate speed.

(iv) It should incorporate up-to-date, preferred techniques and components in order to be of use for as many years as possible. This implies parallel operation using integrated circuits and printed wiring. The machine should be capable of extension in the future.

(v) It should be cheap enough for purchase by university and college departments. This probably means an upper price limit of between £1,500 and £2,000.

(vi) The educational aspect should be held in mind throughout the project and the various student instruction manuals produced at the end.

These basic requirements will be built into a technical specification in Section 5.1 by which time the fundamental decisions on the machine will have been taken.

CHAPTER III

BASIC COMPUTING TECHNIQUES

The basic requirements of the proposed teaching computer were listed in Section 2.4. Before working out a more detailed specification, it is necessary to discuss some of the basic techniques used in digital computation. This is done in the present chapter to prepare the way for the more detailed discussions of chapters 4, 5 and 6.

3.1 Logic Functions

It is necessary to define the basic logic states, connectives and graphical symbols used in this thesis. Many of these are standard notation as may be found in any books on computing electronics, such as 'Digital Computer Design' by Edward L. Braun (10) or 'Fundamentals of Digital Machine Computing' by Guenther Hintze (11).

The following rules are used to define the logic states and notation which exist in digital switching systems.

(i) Two discrete states are considered; any two obviously different from each other may be employed, e.g. 0 - 1, on - off, black - white. In this equipment the logic 1 state is represented by +5 volts and the logic 0 state by 0 volts.

(ii) Every logic quantity must exist in one of these two states.

(iii) Every logic quantity may exist in only one of these states at any one time.

(iv) A quantity representing the logic 1 state is equal to any other quantity which represents the logic 1 state.

(v) Every logic quantity has an opposite. If a quantity is equal to logic 1, then its opposite is equal to logic 0.

(vi) A logic quantity may be constant or variable. If constant it remains in one of the two logic states: if variable it is ~~continuously~~ switched between the two logic states.

(vii) Alpha characters are normally used to identify the various logic functions which exist within a digital switching system. Thus the letter A may represent a logic function, and the letter B used to denote a second logic function. A may therefore be equal to logic 1 or logic 0, but not both at the same time. The notation for the opposite of A, called NOT A, is $\overline{A}$. Thus if $A = 1$, then $\overline{A} = 0$, and if $\overline{A} = 1$, then $A = 0$.

There are three fundamental logic connectives, and, although they are not used in the equipment, it is necessary to define these functions, as the equations specifying the operation of the machine were originally deduced using these basic connectives.

The AND Function

The AND function, or 'gate', is represented by a dot, •, or brackets associated with two or more variables. The graphical symbol used to denote this function is shown in Fig. 1. If A, B and C represent the input variables, then the equation defining the output of the function F is given by,

$$F = A \cdot B \cdot C$$

Thus the function F is 1, when all input variables are 1, and F is 0, if any one of the input variables is 0.

The OR Function

The OR function, or 'gate', is represented by a plus sign, + , associated with two or more variables. The graphical symbol used to denote this function is shown in Fig 2. If A, B and C are input variables, then the equation defining the operation of the function F is given by

$$F = A + B + C$$

Thus the function F is 1, if any one of the input variables is 1, and F is 0, when all input variables are 0.

The NOT Function

The NOT function is represented by a bar above the variable, e.g. $\overline{A}$. The graphical symbol used to denote this function is shown in Fig. 3. If A represents the input variable, then the output of the function F is given by

$$F = \overline{A}$$

'AND' Gate

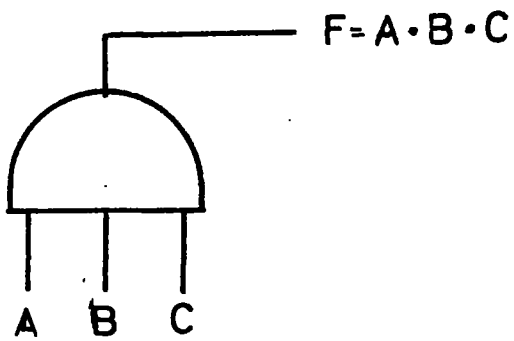


Fig.1

'OR' Gate

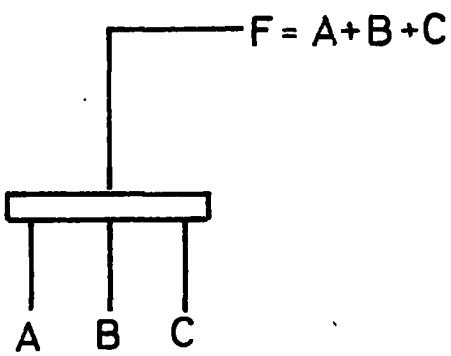


Fig.2

'NOT' Function

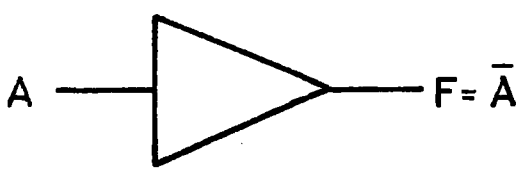


Fig.3

Thus the function F is 1, when the input variable is 0 and F is 0, when the input variable is 1.

The NOR and NAND functions are developments of these three basic gates. It is possible to construct large combinational switching systems using only NOR or NAND elements

The NOR Function

The NOR function is represented by a combination of the symbols used for the OR and NOT functions. The graphical symbol for the NOR function is shown in Fig. 4. If A, B and C represent input variables, the output of the function, F, is defined by the equation

$$F = \overline{A + B + C}$$

Thus the function F is 0, if any one of the input variables is 1, and F is 1, when all input variables are 0.

The NAND Function.

The NAND function is represented by a combination of the symbols used for the basic AND and NOT functions. The graphical symbol for the NAND function is shown in Fig. 5. If A, B and C represent input variables, the output of the function, F, is defined by the equation,

$$F = \overline{A \cdot B \cdot C}$$

Thus the function F is 0, when all input variables are 1, and F is 1, if any one of the input variables is 0.

'NOR' Gate

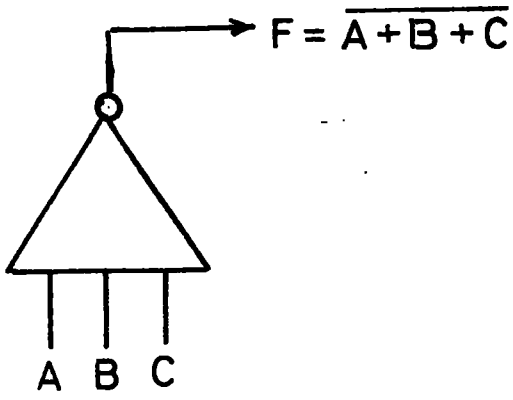


Fig. 4

'NAND' Gate

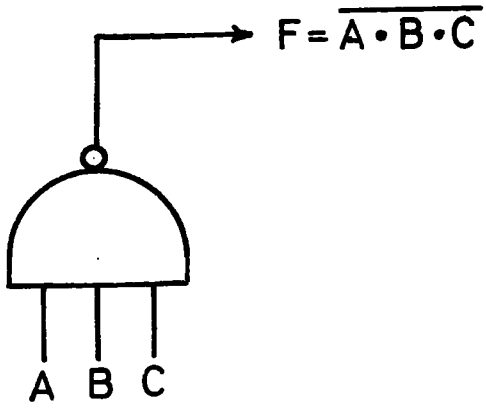


Fig. 5

Exclusive 'OR' Gate

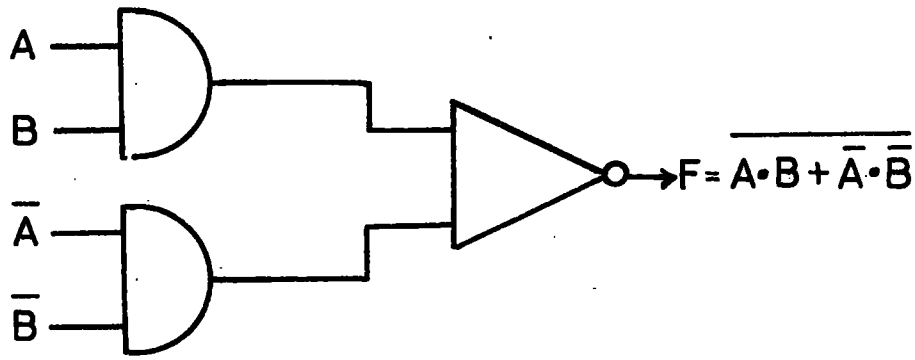


Fig. 6

3.2 Minimal Design Procedure.

It is sometimes convenient, when designing complex switching circuitry, to specify the design in terms of Boolean equations using the basic connectives AND, OR and NOT. It then becomes necessary to rearrange the equations, in order that the specification be presented in terms of the more versatile NOR or NAND elements. Also it is necessary to eliminate any redundant terms which were not initially apparent. The detection of such redundant terms may not be easily seen using the principle laws of Boolean algebra; it may, therefore, be necessary to introduce special techniques (12) which are summarised as follows:

(i) Trial and error.

(ii) Conversion of a sum of products to a product of sums.

(iii) Introduction of dummy factors.

(iv) Chart methods.

(v) Map methods.

It is important to note that, in expressions containing a large number of variables, it may be convenient to break the expression into sections and apply the most appropriate of the above methods to each section. As most of the design specification was required in terms of NAND functions, a simple example is given.

$$F = A \cdot B + \overline{A} \cdot \overline{B}$$

$$\overline{F} = \overline{A \cdot B + \overline{A} \cdot \overline{B}}$$

$$\overline{F} = \overline{(A \cdot B)}\ \overline{(\overline{A} \cdot \overline{B})}$$

$$F = \overline{\overline{(A \cdot B)}\ \overline{(\overline{A} \cdot \overline{B})}}$$

It is seen that the function F may be realised using three NAND elements.

3.3 Combinational Logic Circuits

When it becomes evident that a number of basic logic elements, connected so as to satisfy a complex expression, are being employed on a number of occassions, it is customary to simplify the specification by considering the complex function as a single unit which has straightforward relationships defining its operation. The combinational logic elements used in this equipment are therefore listed and defined.

3.3.1 EXCLUSIVE-OR Gate

The EXCLUSIVE-OR gate is used fairly commonly when a function F is to be in the logic 1 state with only one of two input variables, A + B , in the logic 1 state. Thus

$$F = \overline{A} \cdot B + A \cdot \overline{B} ,$$

which may be rearranged to give,

$$F = \overline{A \cdot B + \overline{A} \cdot \overline{B}}.$$

The use of two AND gates and one NOR gate is required to satisfy this equation. The general arrangement for the

EXCLUSIVE-OR function is shown in Fig. 6.

3.3.2 The Binary Adder

The combinational logic circuit required to add together two binary digits, A and B, with a carry digit C from a previous stage, is obviously used extensively in computing electronics. An examination of the following truth table shows how the Boolean expressions are derived to satisfy the requirements of a full binary adder.

A	B	C	C_{out}	$\overline{C_{out}}$	Sum
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	1

$$C_{out} = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

$$= C(\overline{A}.B + A.\overline{B} + A.B + A^{*}.B) + A.B.\overline{C} + A^{*}.B.C$$

$$= C\left[A(\overline{B} + B) + B(\overline{A} + A)\right] + A.B(\overline{C} + C)$$

$$= A.C + B.C + A.B$$

$$Sum = A.\overline{C_{out}} + B.\overline{C_{out}} + C.\overline{C_{out}} + A.B.C$$

* Dummy factors

The logic circuit required to satisfy these two equations is shown in Fig. 7. and is the circuit of the Texas 7483N I.C. which was used in this computer.

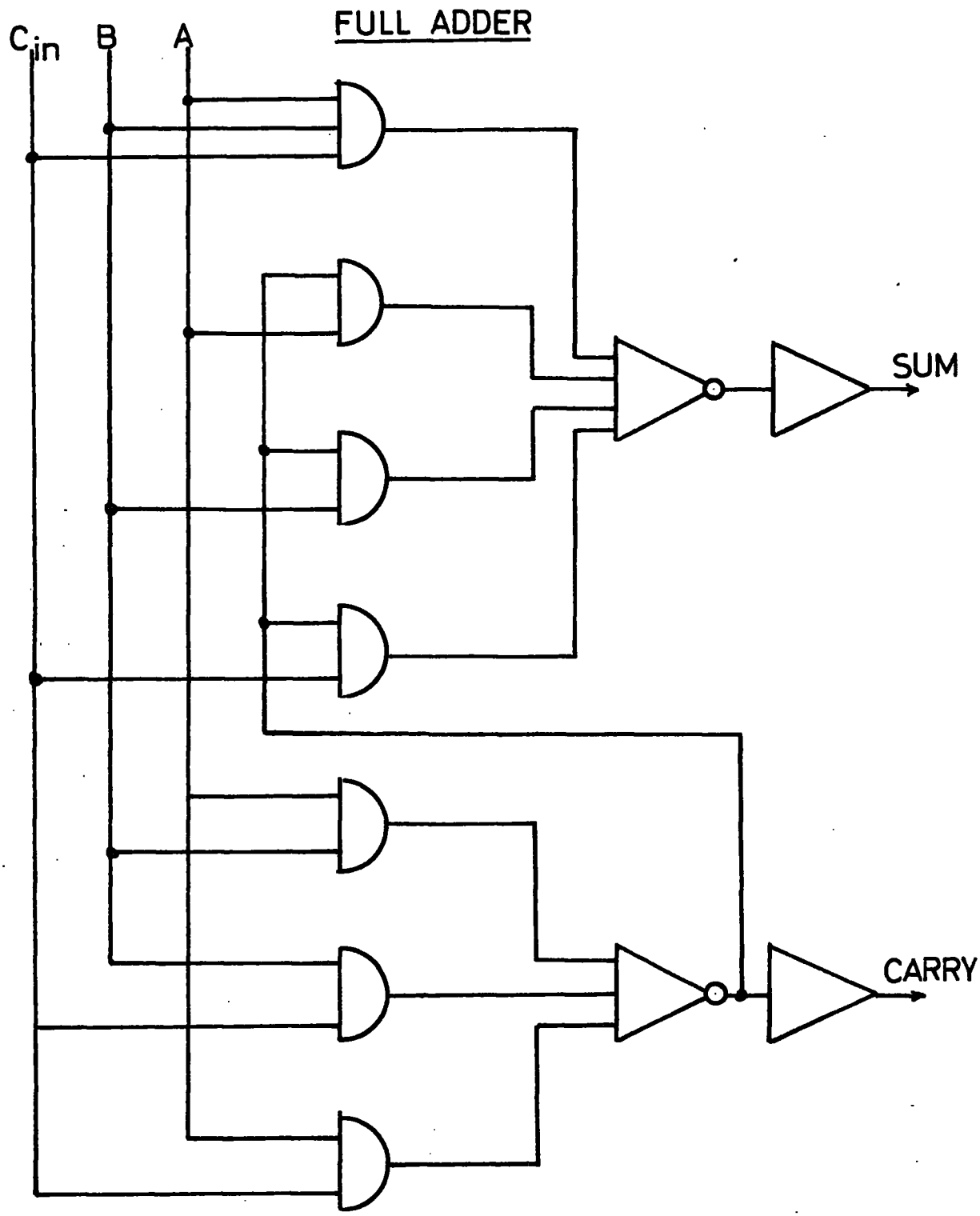


Fig. 7

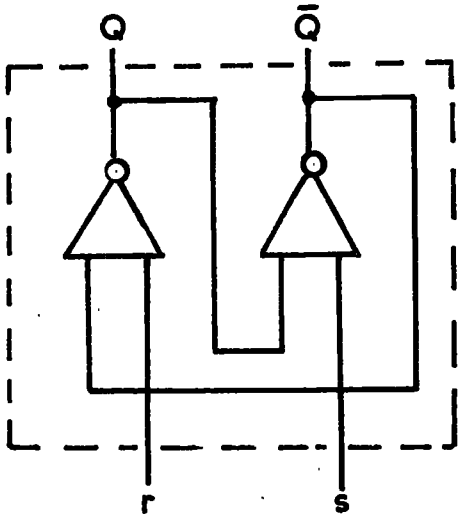
3.3.3 The Bistable Element

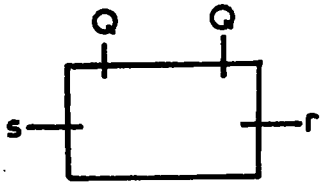
The bistable element is widely used in digital equipment. It is appropriate, therefore, to describe the operation of the simplest form of such elements and to give the characteristic equations and symbols associated with the two types used in this equipment.

This logic circuit is arranged to have two outputs, Q and $\bar{Q}$, only one of which may exist in the logic 1 state at any one time. The state of these two outputs is controlled by the previous application of a logic 1 state to either of the two inputs which are called the set, s, and reset, r, inputs. If a logic 1 is applied to s, then the output Q assumes the logic 1 state and will remain in that state even after the logic 1 has been removed from the set input. Similarly a logic 1 applied to r will cause the output $\bar{Q}$ to assume the logic 1 state and remain in that state after the logic 1 has been removed from the reset input. The logic schematic of Fig. 8 illustrates the mechanism of a very simple R-S bistable element.

The characteristic equation, defining the operation of the bistable element, relates the state of the outputs Q and $\bar{Q}$ at a time (t+1) to the state of the inputs at a previous time (t). Thus the characteristic equations for the R-S bistable element shown in Fig. 8 is simply,

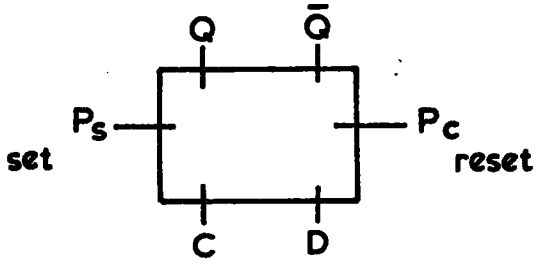
$$Q_{t+1} = s_t \qquad \text{and} \qquad \bar{Q}_{t+1} = r_t .$$

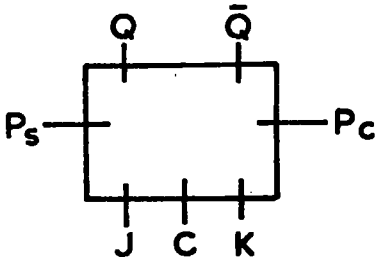




R-S Bistable

Fig. 8.





Clock Data

$$Q_{t+1} = \bar{P}_c \cdot Q_t + P_s + C \cdot D$$

$$\overline{Q_{t+1}} = \bar{P}_s \cdot \bar{Q}_t + P_c + C \cdot \bar{D}$$

Type D Bistable

$$Q_{t+1} = \bar{P}_c \cdot Q_t + P_s + C(J\bar{K} + \bar{J}\bar{K}Q_t + JK\bar{Q}_t)$$

$$\overline{Q_{t+1}} = \bar{P}_s \cdot \bar{Q}_t + P_c + C(\bar{J}K + \bar{J}\bar{K}\bar{Q}_t + JKQ_t)$$

Type J K Bistable

Fig. 9.

The type D and J-K bistables, which are used in this equipment are more complex than the R-S type. They have more input terminals from which the state of the outputs Q and $\bar{Q}$ may be controlled. A detailed analysis of the operation of these elements is given in Ref. 13. The graphical symbol and characteristic equations for each type are shown in Fig. 9.

It is customary to give a title to each bistable, and this is normally done by using the initial letters of the particular function with appropriate numerical subscripts. The schematic of Fig. 10 illustrates the accepted method of designating a series of bistable elements associated with the common function B.

3.3.4 Shifting and Counting

Since the shifting and counting operations are used extensively in digital equipment, it is appropriate to show how they may be realised using both basic logic and bistable elements. An 'n' stage shift register may be constructed from n bistable units, so interconnected by gating elements, that on the application of a shift pulse the contents of each stage, B_x, is transferred to stage $B_{x\pm1}$. Fig. 10 indicates how a shift left register may be constructed using R-S bistable elements. The gates R and S are used to steer the clock pulses from the control line to either the

Shift Left Register

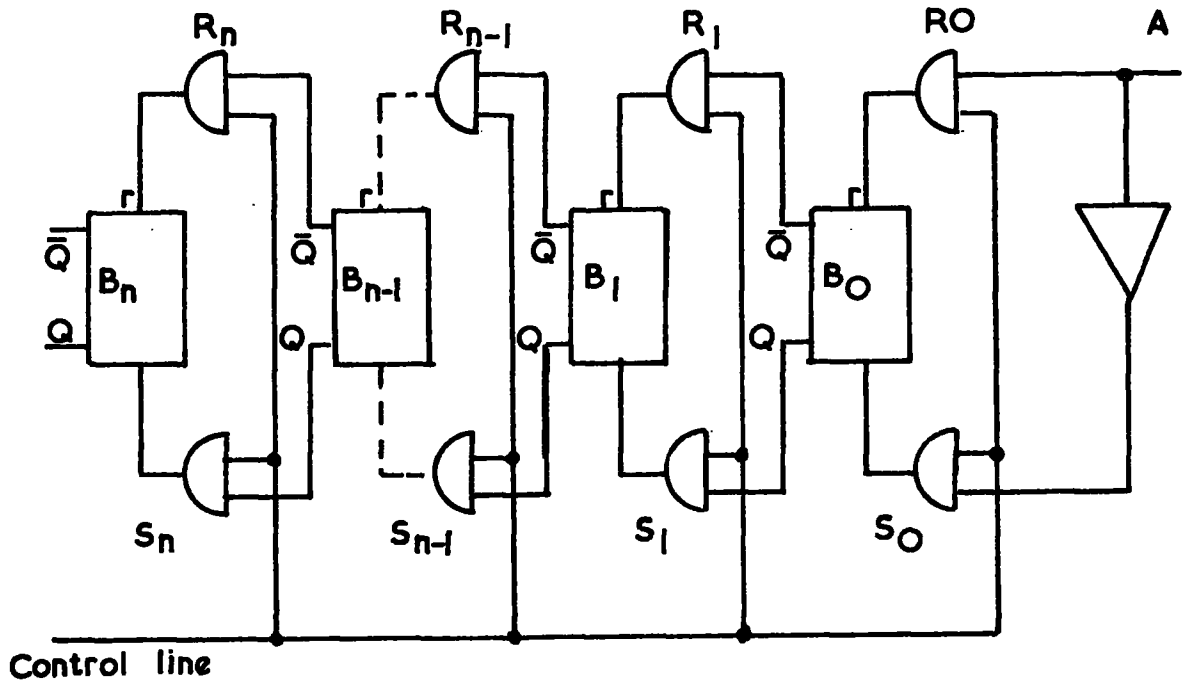


Fig. 10

Divide by Two Circuit

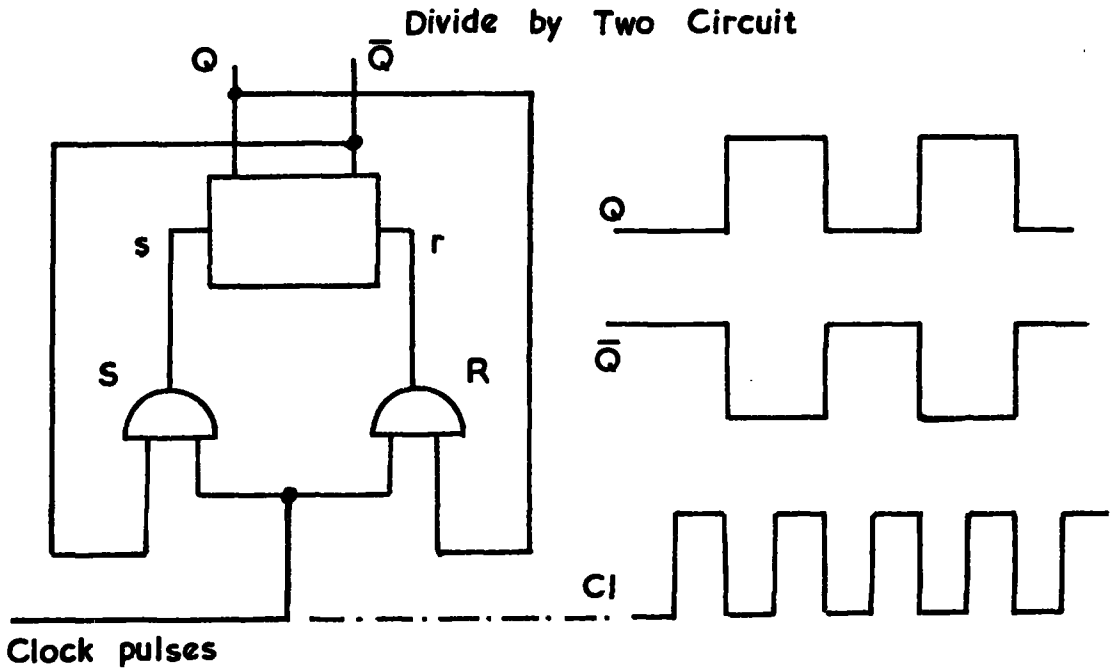


Fig. 11

reset (r) or set (s) input of each bistable element. This steering action is controlled by the state of each preceding element, or in the case of stage B_0, by the state of the input function A. Assuming that element B_0 is 'set' and A is equal to 0, then on the application of the first clock pulse, C_L, gates R_0 and S_1 will operate, causing elements B_0 and B_1 to be 'reset' and 'set' respectively. This steering action applies to all stages of the register, causing the contents of each stage to be shifted one place left each time a clock pulse C_L is applied to the control line. The switching time of the binary element usually provides sufficient delay, so that intermediate storage elements are not required. Shift registers are used for temporary storage of binary numbers, serial to parallel conversion and in the more complex arithmetic processes. The characteristic equations for each bistable element B_n in a shift left register are given by:

$$B_{n_{t+1}} = B_{n-1_t} \cdot C_L \quad \text{and} \quad \overline{B_{n_{t+1}}} = \overline{B_{n-1_t}} \cdot C_L$$

Similarly for a shift right register these equations are:

$$B_{n_{t+1}} = B_{n+1_t} \cdot C_L \quad \text{and} \quad \overline{B_{n_{t+1}}} = \overline{B_{n+1}} \cdot C_L$$

The second of the more complex operations, binary counting, may be realised using similar techniques to those described for the shift register. In this case the steering of the clock pulses is controlled by the state of the

bistable element to which the pulses are applied. The logic schematic, Fig. 11, shows a simple 'divide-by-two' circuit in which the pulses appearing at the output occur at half the rate of those applied to the input. With the bistable initially reset, gate S steers the first and subsequent odd-numbered clock pulses to the set input s, causing the bistable element to change state. The second and subsequent even-numbered pulses are applied to the reset input r, causing the bistable element to switch back to its original reset state. The signal appearing at the $\overline{Q}$ output may be used as a clock signal to a second identical 'divide-by-two' circuit. By cascading n stages in this fashion, a system is produced which will accept and store as a binary number up to 2^n-1 clock pulses. When constructing counters of this sort, it is important to establish whether the bistable elements change state on the leading or trailing edge of the clock pulse. The effect of propagation delay in fast counters may be overcome by gating the clock pulses to each stage of the counter. This ensures that all stages change state at the same time. The characteristic equations for a 'divide-by-two' circuit are:

$$B_{n_{t+1}} = \overline{B_{n_t}} \cdot C_L \quad \text{and} \quad \overline{B_{n_{t+1}}} = B_{n_t} \cdot C_L$$

3.4 Realisation of Arithmetic Processes

The basic processes involved in arithmetic are addition, subtraction, multiplication and division. It is expected that sub-routines will be programmed for the determination of the more complex arithmetic operations e.g. square roots, trignometric functions etc. These sub-routines will also assist in demonstrating the operation of the equipment. Some of the logic methods available for the realisation of the basic processes are described here. Addition has already been dealt with in Section 3.3.2

3.4.1 Binary Subtraction.

It is possible to produce a set of equations to satisfy the laws of binary subtraction, and hence construct a binary subtractor by methods similar to to those described for the full adder in Section 3.3.2. It is common practice to employ the method of complementing the subtrahend and adding, thus eliminating the need for both an adder and a subtractor to be incorporated in the same machine. The logical decisions required to complete a subtraction are given for the types of complement possible in binary algebra. Both methods have certain merits and limitations, the most serious limitation being the complexity of the methods necessary to establish the sign of the remainder. A number of rules may be applied (14) which provide the equations necessary

to determine the sign of the remainder. The two methods are as follows:

(i) Subtraction using Two's Complement.

The two's or radix complement of a binary number is obtained by replacing all 1's by 0's and all 0's by 1's and adding 1 in the least significant position. The difference is formed by adding the two's complement of the subtrahend to the minuend.

e.g. 1001 - 0101 = 0100 i.e. 9-5 = 4

Two's complement of subtrahend	=	1010
		1
	=	1011
adding minuend		1001
difference	= 1)	0100

The 1 in the overflow column indicates a positive difference.

(ii) Subtraction using One's Complement.

The one's complement of a binary number is obtained simply by replacing all 1's by 0's and all 0's by 1's. The one's complement of the subtrahend is added to the minuend to give the difference. With this method the carry produced in the most significant stage must be added together with the least significant digits to complete the subtraction. Using the previous example:

One's complement of subtrahend,		1010
adding minuend		1001
	=	1 0011
The overflow digit is then added in the least significant stage.		: :....1
Difference.	=	0100

3.4.2 Binary multiplication.

As in the case of subtraction it is not economical to construct a multiplier which will generate the product digits simultaneously with the inputs. It is usual to find that the methods adopted are based on 'long multiplication'. The method described here is based on the repeated addition of the multiplicand to correct orders of a sub-product, the final sub-product being the required product. It will be appreciated that if each of the two operands have n digits, then the product will have a maximum of 2n+1 digits. It is customary to allow the product to develop in an accumulator and in the register which is used to store the multiplier. This is possible since the multiplier is successively shifted out of its initial storage register during the operation. It then becomes necessary to round off the product to n digits so that it is again compatible with the data already stored.

This operation may be realised in the following way. It is assumed that the sub-products are to be developed in an accumulator register which has right shifting facilities

and that the multiplier is initially stored in an auxilliary storage register M, which allows right shifting. The first sub-product is formed by examining the least significant digit of the multiplier. If this is a 1, the multiplicand is added to the zero contents of the accumulator. If the least significant digit of the multiplier is 0, the addition is not performed and the contents of the accumulator remain at zero. The contents of both the accumulator and register M are then shifted one place right, with the least significant digit in the accumulator being routed to the most significant position in register M. The process of examining the least significant digit of the multiplier, adding and then shifting is repeated until all the digits of the multiplier have been tested.

The following example is intended to illustrate this method of binary multiplication. Multiply 10011 x 1101 i.e Multiplicand is 10011 and the multiplier 1101.

	Accumulator.	Register M
The L.S.D of multiplier is 1,	00000	
therefore sub-product is	10011	
Shift (ACC) and (Reg.M) right	01001	1
L.S.D of multiplier is 0, sub-product=	01001	1
Shift (ACC) and (Reg.M) right	00100	11
L.S.D of multiplier is 1, add	10011	
Sub-product	10111	11
Shift (ACC) and (Reg.M) right	01011	111
L.S.D of multiplier is 1, add	10011	
Final product is therefore	11110	111

3.4.3 Binary Division.

The method involving counting the number of repeated subtractions of the divisor from the dividend and resulting remainders is considered to be uncharacteristic of a modern computer. The commonly used method requiring trial subtractions, restoring and shifting is considered suitable for the proposed demonstration equipment and is therefore described here.

This method requires that the two operands be initially aligned such that the most significant 1 digits are in identical positions in their respective registers. The difference in the number of digits contained in the dividend, compared with those contained in the divisor, is used to determine the number of digits in the quotient. A trial subtraction of the aligned divisor is performed; if the remainder is positive a 1 is placed in the most significant position of the register in which the quotient is to be developed. If, however, the remainder is negative a 0 is placed in the quotient and the original dividend restored. The divisor is then shifted one place right and a further trial subtraction is performed. The tests on the remainder are repeated to determine the value of the digit to be entered into the quotient register. The process of trial subtraction is repeated until either the required accuracy is obtained or the capacity of the register storing the

the quotient has been exceeded. An example is given to illustrate this method of binary division.

Divide 1100100 by 10100 i.e. 100 by 20.

The difference in the number of digits between the two operands is (7 - 5) = 2; therefore the number of digits in the quotient is (7 - 5 + 1) = 3.

Step 1.	Align both operands	1100100 1010000
Step 2.	Subtract	0010100
Step 3.	Remainder is positive therefore quotient is 1	
Step 4.	Shift divisor one place right	
	Positive remainder	0010100
	Shifted divisor	0101000
Step 5.	Subtract 1	0001100
Step 6.	Remainder is negative therefore quotient is 10	
Step 7.	Shift divisor one place right	
	Restored remainder	0010100
	Shifted divisor	0010100
Step 8.	Subtract	0000000
Step 9.	Remainder is positive therefore quotient is 101.	

The quotient now contains the required three digits and at this stage the operation is terminated.

CHAPTER 1V.

FACTORS AFFECTING DESIGN.

4.1 Primary Considerations.

The general requirements for the demonstration computer have already been established from the literature and in Section 2.4 as follows:

(i) The machine must be a computer in the full sense of the word and not an electronic calculator. This implies that it must work to a programme, which will be fed to the machine, to-gether with the data, and that the programme may be modified during its execution.

(ii) Large display panels are essential for the teaching function.

(iii) Three speed operation is required, manual, slow speed and fast speed.

(iv) The machine must incorporate modern circuit and constructional techniques.

(v) The cost of the equipment should be as low as possible.

(vi) The educational requirements must be borne in mind at every stage. This implies that the circuit must be capable of being broken down into the basic units for individual study, and that each part must be based on straight forward, easily understood electronic circuits.

It is appropriate at this stage to state the non-technical factors which have affected the design. It was

obviously necessary to balance the estimated available man-hours with the requirements of design and construction effort. It was important to ensure that the programme did not become so large that there would not be time to consider it in sufficient detail. Also it was necessary to 'foresee' the state of the art techniques so that components would be available at reasonable cost. In considering these and other aspects, the senior members of the department decided to split the project into three sections. The Memory Unit is described by Smith (1), the Control and Arithmetic Units are the subject of this thesis, while the design and construction of the Input / Output equipment was held in abeyance until after the completion of the other two parts. This arrangement is compatible with the initial proposals, that the three sections should be physically separable so that, in the event of some major malfunction, all would not be lost.

Against this background, there are the following main features of the machine, which should be listed but which obviously warrent no further discussion in this thesis. It was decided that the computer would have:

(i) A ferrite core store.

(ii) A magnetic drum store.

(iii) An adaquate instruction repetoire.

(iv) That it would use integrated circuits and plug-in printed circuit modules.

The following secondary features were also examined:

(i) Aspects of general purpose computers.

(ii) Type and number of machine instructions.

(iii) Serial or parallel operation.

(iv) Number representation.

(v) Address modification.

(vi) Word length.

(vii) Micro-programming.

(viii) Method of demonstration.

These features are discussed in detail in the present Chapter. Decisions on these features enabled a detailed specification of the machine to be drawn up and this will be given in Section 5.1.

4.2 Some Aspects of General Purpose Computers.

In an effort to establish those main characteristics, which are common to the majority of general purpose machines, an examination of the recent survey on British computers(15) was undertaken. The results of this examination assisted in determining a specification which is characteristic of a modern general purpose machine. Those aspects relating to central processor operation are summarised here.

(i) There is no definite preference for pure binary or binary coded decimal number systems. There is a suggestion, that, where the machine is intended for business or accounting

purposes, the latter system is preferred.

(ii) All machines operate with fixed point arithmetic, floating point operation being included in those machines used for scientific applications.

(iii) The word length appears to be governed by accuracy and the expected range of numbers as well as the address format. Variations from 12 to 60 bits are evident.

(iv) With few exceptions, arithmetic operations are carried out in the parallel mode, the execution time for such operat--ions being dependent on the switching speed of the logic elements, the access time of the memory and the complexity of the timing circuits. Times for addition vary from 3μS to 1.3mS, with multiplication and division taking in general 4 to 5 times as long.

(v) Current trends indicate a complete lack of uniformity in respect of the number of instructions offered, the range being from 14 to 170, with the majority of machines having between 50 and 80 instructions.

4.3 Machine Instructions.

A machine instruction comprises a pre-determined sequence of minor commands which are established by the design of arithmetic, control and memory units. The number and type of machine instructions may be classified as those associated with:

(i) transfer of operands between the memory and arithmetic units,

(ii) elementary arithmetic operations,

(iii) transfer of control and the B line modifier,

(iv) logical manipulation of operands,

(v) input and output of data.

The number of instructions in this machine must be kept to a minimum to reduce costs and preserve clarity of demonstration. There must of course be sufficient instructions to enable the machine to operate with all the main features of a full computing system. The instructions should be as simple as possible, and it is hoped that student appreciation of the more complex instructions will be improved by demonstrating one simple step and allowing time for discussion, before proceding to the next. In a general purpose machine a typical instruction may be, "Add the contents of location n to the contents of the accumulator placing the sum in storage location n" In the demonstration computer it is expected that three instructions may be necessary to effect this operation:

(i) Transfer (n) to the arithmetic unit,

(ii) Add,

(iii) Transfer sum to location n.

By placing a stop instruction between steps i and ii, and between ii and iii, it will be possible to demonstrate the operation more effectively.

4.4 Parallel or Serial Operation.

The difference between parallel and serial operation is as follows. In a parallel arithmetic unit all bits in the operands are acted upon simultaneously, requiring a single timing pulse for any one function. In a serial machine, on the other hand, each bit of an n bit word is acted upon sequentially, and this requires n timing pulses to complete each function.

Comparing the two modes of operation, the parallel type requires approximately, n times the circuitry in the arithmetic unit, $1/n$ times the circuitry in the timing unit and $1/n$th of the time to complete a particular function. Other major factors must also be considered: e.g. compatibility with the memory system, the operating speed of the input devices and the expected load on the machine. Nearly all full-size computers operate in the parallel mode and the proposed machine should, if possible, be typical of this modern practice.

The choice of parallel or serial operation in this case is also affected by the demonstration aspects of the machine. One of the major techniques to be demonstrated is timing and sequencing of operations, and it is felt that to introduce many more timing pulses to satisfy the requirements of a serial machine would seriously impair the effectiveness of the demonstration. The parallel mode

is also compatible with the small core memory which is included in the machine. For these reasons it was decided to adopt parallel working in the arithmetic unit.

4.5 Number Representation

The choice of number representation was made on the simplicity of circuitry and student appreciation. The three systems, pure binary, binary coded decimal and octal were each reviewed. The pure binary system, which requires the most simple arithmetic circuitry, was adopted. It was appreciated that the binary coded decimal system has certain advantages in the input / output sections. It was felt that the complexity of the circuitry to deal with the 'excess 6' would severely limit the effectiveness of arithmetic demonstrations.

4.6 Operand Address Modification.

One of the most important aspects of computing is the use of the'B line modifier' or index register. When it is required to repeat an instruction n times with different operands, it is usual to employ the same instruction, modify the address field by the contents of the modifier register and use a conditional transfer instruction to maintain the loop in the programme. The number of address modifications to be applied, (n), is initially stored in the modifier register, the contents of which are reduced

after each loop has been completed. The value of the decrement is left to the programmer and may be 1,2,3,4, etc. It is also usual in large computing systems to have more than one modifier register. It will be appreciated that associated with each register there must be the necessary gating and control circuitry and also special instructions for addressing them. It was decided that, in view of the importance of this technique, at least one modifier register should be included. The procedure normally adopted for adding the contents of the modifier register to the operand address is to use the adder unit contained in the arithmetic unit. It was felt that to do this in the proposed computer would tend to over-complicate the arithmetic unit. By using a single bit full adder in the control unit the arithmetic section would not be affected and it would permit the demonstration of serial addition.

4.7 Word Length.

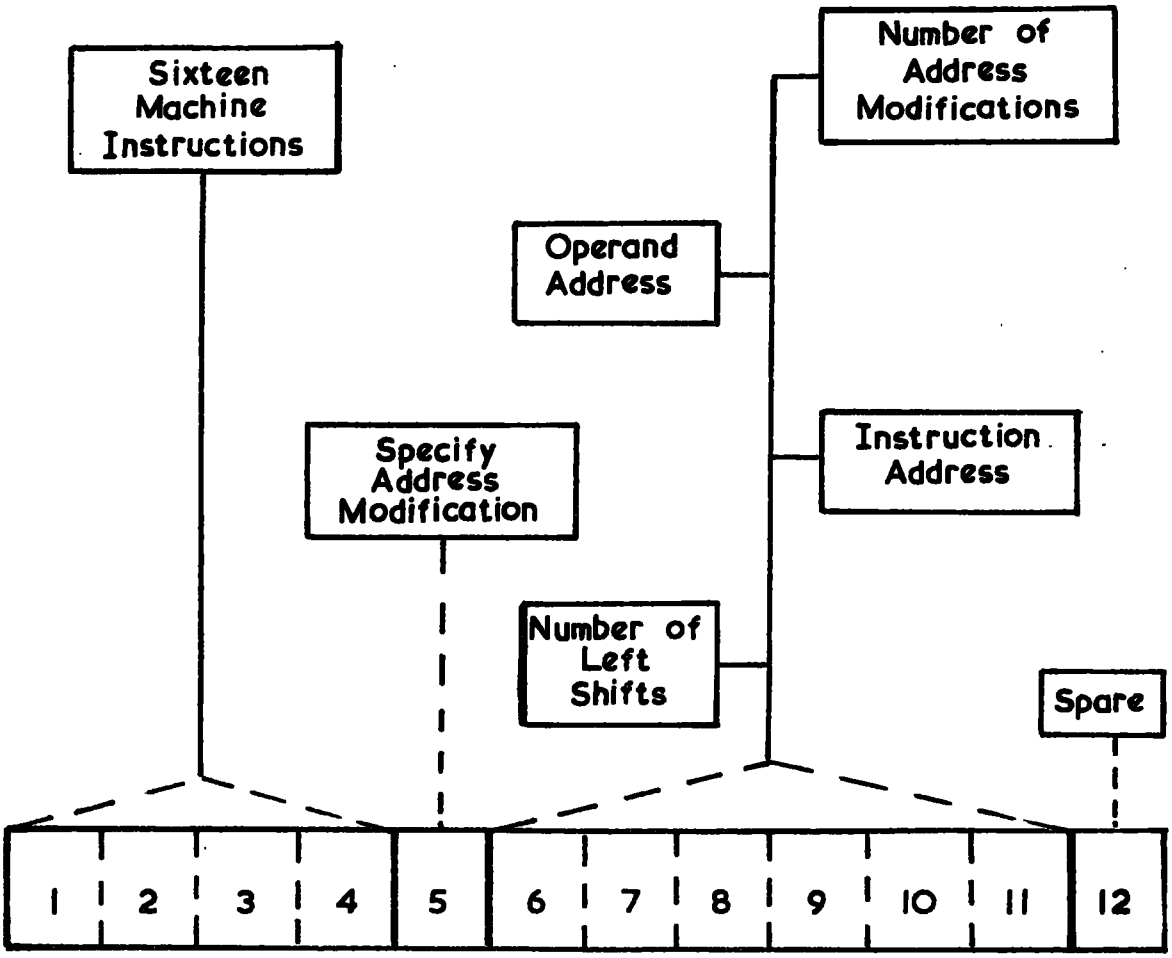
Having decided on parallel operation, it was apparent that the word length would directly affect the amount of circuitry required for storage and arithmetic operations. It was necessary to choose as few bits per word as possible without impairing the effectiveness of the demonstrations.

In order to give an adequate demonstration of multiplication and division, it was felt that the minimum

number of digits in either operand should be four, and the minimum difference in the number of digits between either operand should be three. With one bit reserved as a sign digit, the minimum number of digits per word was therefore established as twelve. As a single address mode was intended this word length was examined in terms of the machine instru--ction. When used as an instruction, the word must contain sufficient bits to specify, in binary code, all machine instructions, all operand or instruction addresses and the code to address the modifier register. Fig. 12 shows the format of the proposed machine word, when used as an instruction, indicating that the twelve bits considered necessary for arithmetic processes satisfy these requirements. For simplicity and minimum cost it was therefore decided that a twelve bit word should be adopted.

4.8 Micro-programming.

It is important that the computer be as flexible as possible in use. If the sequence of minor commands comprising each instruction is not readily alterable, this may not be achieved and may limit the type of calculation which can be performed. A method of altering existing instructions or generating new ones is therefore considered desirable. One method of achieving this is micro--programming. The technique assumes that all minor commands in the control unit are available for assembling



Machine Word

Fig. 12

into any desired sequence to form new instructions. This may be done by programming, or by a pin-board, or by re--wiring of the timing circuitry. By arranging the circuitry associated with the timing pulses in a form suitable for later modification, it is hoped that the effectiveness of the equipment may be maintained in the light of experience. A general purpose integrated circuit carrier board referred to in Section 7.1 was developed especially for this purpose.

4.9 Method of Demonstration.

In Section 2.4 it was established that the method of demonstration was to be pre-eminent throughout the design. Clear display panels are essential, incorporating lamps to show the condition of all the registers and major control gates in the system. Also, for clarity of demonstration, it was decided that the machine should incorporate four modes of operation. As these modes materially affect the actual circuit design, a description of each is given here.

(i) Standstill operation.

This mode is included to provide the bridge between the basic logic tutor and the demonstration computer. In this condition, the machine will operate without the time base. All the minor commands, e.g. load register A, shift (Reg. A) left, transfer sum to ACC etc will be available on a 'Student push-button panel'. A twelve bit

data panel will also be provided. With these facilities the student will be able to transfer information throughout the machine, without reference to any particular sequence or instruction. Demonstration of the basic computing processes, counting, shifting, adding and parallel transfer is possible in this mode.

(ii) Manual operation.

In this condition, each timing pulse will be generated manually. This permits the student to fully appreciate the effect of one timing pulse, before proceding to the next. All instructions may be executed in this mode with the instruction being set up on a twelve bit data switch panel.

(iii) Slow operation.

In this mode, the time base will generate one bit time per second. A student 'Execute' switch, when operated, will allow the machine to complete one instruction and then stop. This mode illustrates the way in which the time base links to-gether the various minor commands to form an instruction.

(iv) Normal operation.

In this mode, the time-base will run at its normal speed. Instructions will be obtained in sequence, (except for jump instructions) from the memory and then executed. Visible indication will be by way of waveform examination in any of the three main sections of the equipment.

CHAPTER V.

GENERAL FEATURES OF THE DESIGN.

5.1 Statement of Specification.

It was decided to draw up a basic specification which would enable an estimate to be made of the circuitry required to construct equipment generally in accordance with the findings of Chapters 2 and 4. The details for each part of the specification followed fairly naturally from the major decisions described in Chapter 4, and the reasons for their choice are not included here.

Based on this specification, block schematic diagrams were produced to show, in detail, the quantity and type of logic elements which would be required to meet it.

Specification

Word length. ---Twelve binary digits, the most significant digit to be used as a sign digit.

Number system.--Pure binary, in the range $+2047 \geq x \geq -2047$

Memory.---------64 word magnetic core store, parallel.
45 word magnetic drum store, serial. *

Modifier.-------One six bit register for operand address modification only.

Arithmetic.-----Parallel. Add, subtract, multiply and divide.

* In practice the drum store is a magnetic tape loop and in later parts of this thesis it is described as the tape store.

Instructions.---Single address, total 16 instructions as follows:

Four to link the memory to arithmetic section.

Four to identify basic arithmetic operations.

Two to specify conditional transfers.

Two to specify logical manipulation of data.

One to address the modifier register.

One to terminate the programme.

One for input of information.

One for output of information.

Input.----------Simulated punched tape or push-button switches at this stage of the project.

Output.---------Filament lamps at this stage of the project.

Controls.-------Three position mode switch:-

Normal - Slow - Manual / Standstill.

Single switches for:-

Start, Stop, Execute, Proceed and Manual.

Construction.---Must incorporate:-

Integrated circuit logic elements.

Plug-in printed circuit modules.

Physical separation of major sections.

Comprehensive mimic diagrams.

Free access to logic elements.

A self-contained power supply.

5.2 Overall Operation of the Computer.

Before proceeding to the more detailed description of the operation of the Control and Arithmetic sections it is necessary to describe briefly how this computer is intended to function. This then forms the basis for the design work required to build a machine to the specification of Section 5.1.

It was assumed initially that the programme would be stored as a sequence of machine instructions in the serial type magnetic drum store, and that operands would be stored in the magnetic core unit. On receipt of the instruction alignment signal, the control unit allows the clock signal, from the drum store, to be gated through to the timebase. The required machine instruction is then fed to the control register where it is decoded and used to direct the timing pulses to the various sections of the machine. These pulses are used to execute the instruction. At the end of the instruction the timebase is returned to the standby condition to await the next instruction alignment signal.

This summary of the overall function of the control of the machine can now be extended in detail. The general features of the design followed logically from the specification and this operating mechanism. The detailed derivation of the basic design will not be given but the next two sections, Section 5.3 and 5.4, describe the

operation of the control and arithmetic units of the machine in outline. As a result of this it is possible to list the 16 instructions (Section 5.5) and the minor commands required for each (Section 5.6). This enables the circuit design to follow logically as described in Chapter 6.

5.3 Final Outline Design of the Control Unit.

The block schematic, Fig. 13, indicates the main interconnections required between the major sub-units and gives the expressions for the timing pulses associated with the control gates. The logic elements in this section will control:

(i) The transfer of instructions and operands from the input devices to the memory.

(ii) The serial transfer of instructions from the drum store to the control register.

(iii) The decoding of the four most significant bits of the control register to give one of the available 16 machine instructions.

(iv) The modification of the operand address.

(v) The next instruction address.

(vi) The generation of all timing pulses.

The gating, shifting and transfer operations in both the arithmetic and control sections are effected by pulses occuring at a pre-determined time and in the correct

sequence. These pulses are derived from the timebase, which generates character (K) and bit (B) times by appropriate division and decoding of the clock signal. The waveform diagram of Fig. 15 illustrates the 12 : 1 time relationship between the character and bit times. Fig. 16 shows the time relationship between bit times (B), master clock (C_L), pre-clock (C_{Lp}) and the square-wave signal (C_{L-10}) which is derived from the drum store.

During K_o time the twelve bit word, representing the in--struction, is shifted from the drum store to the control register by the application of twelve gated master clock pulses to the shift line of the control register. The decoded instruction is then gated with signals from the time--base to produce the correct sequence of timing pulses, or minor commands, necessary to execute the particular instruction. It is expected that all instructions except 'multiply' and 'divide' will require no more than the twelve pulses available during the second character time K_1. The execution of these two exceptional instructions (i.e. 'multiply' and 'divide') will be complete by the end of the fourth character time K_3.

If the specified instruction is one of the four dealing with the transfer of operands between the memory and arithmetic units, the operand address will be defined by the contents of the control register (bits 6 to 11).

CONTROL UNIT BLOCK SCHEMATIC

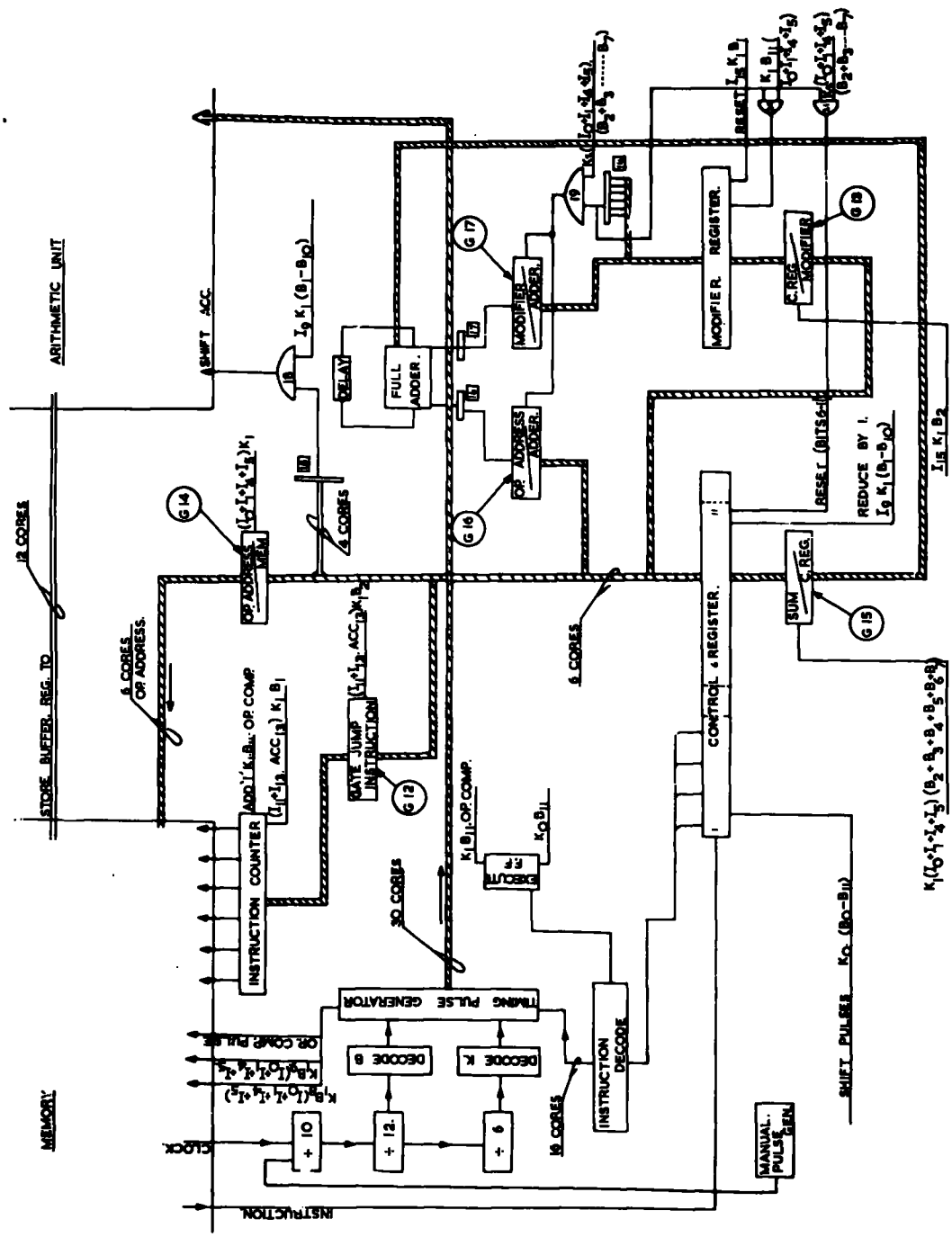


Fig.13

ARITHMETIC UNIT BLOCK SCHEMATIC

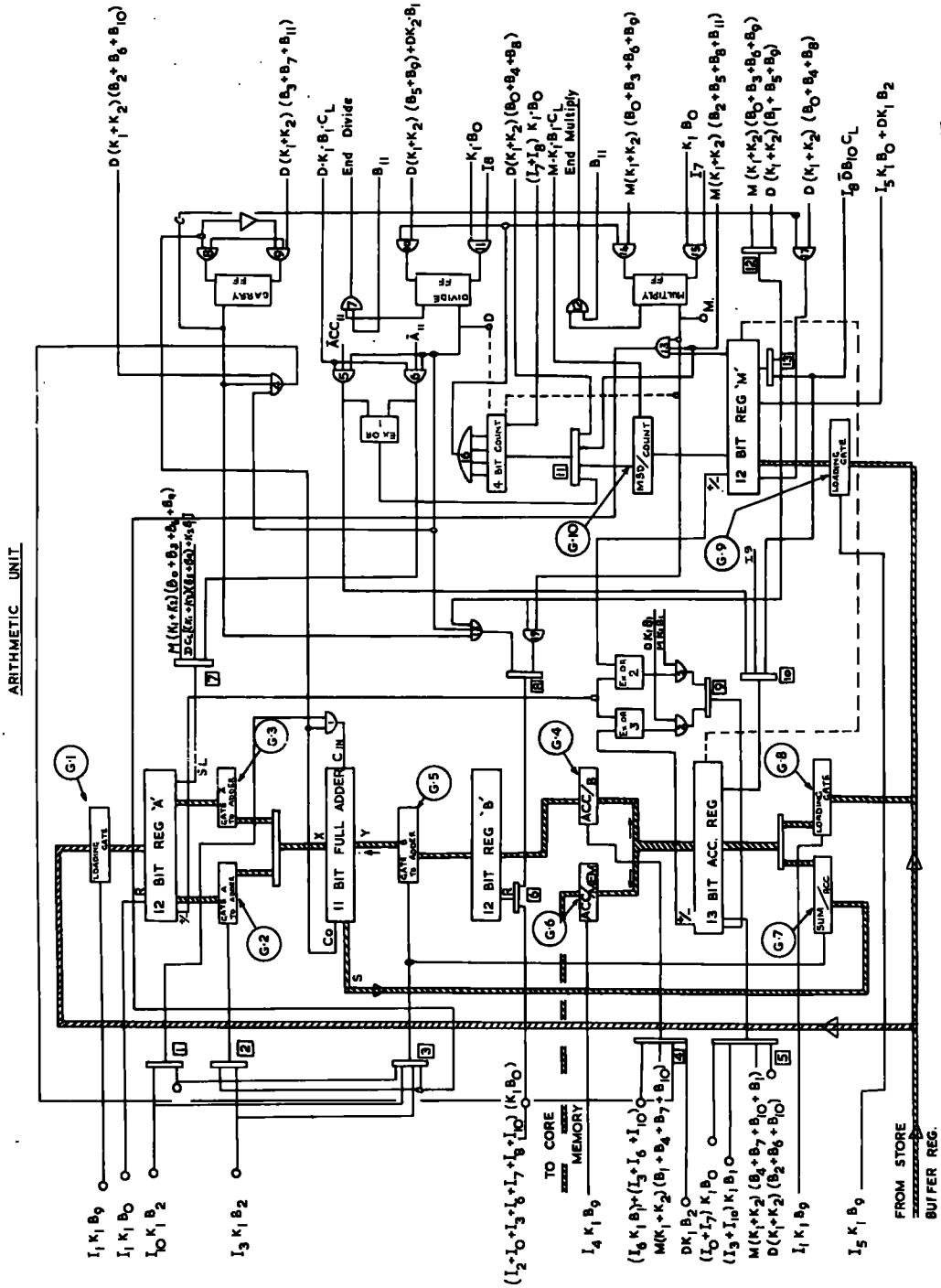


Fig.14

TIME BASE WAVEFORMS

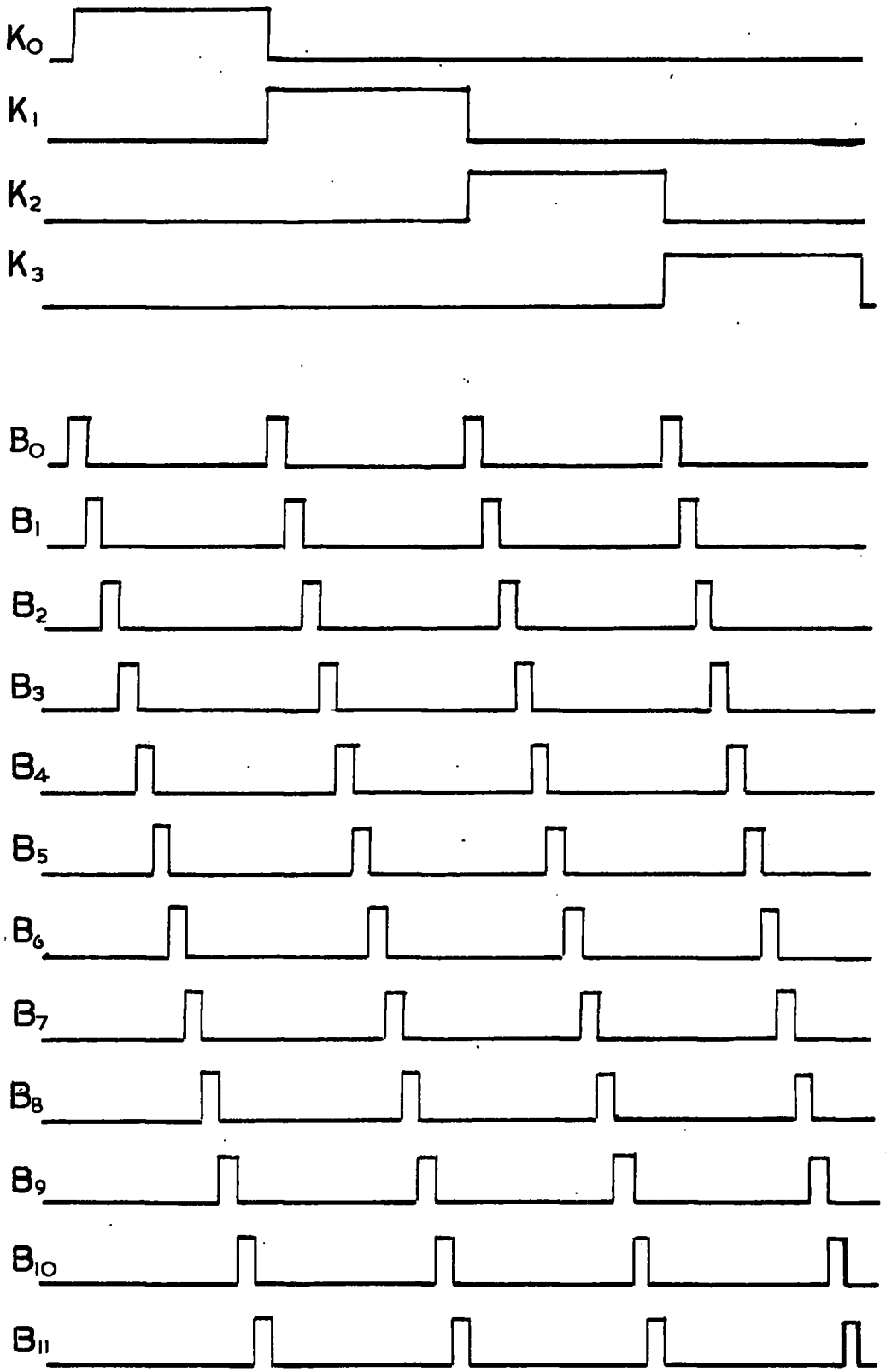


Fig.15

TIME BASE WAVEFORMS

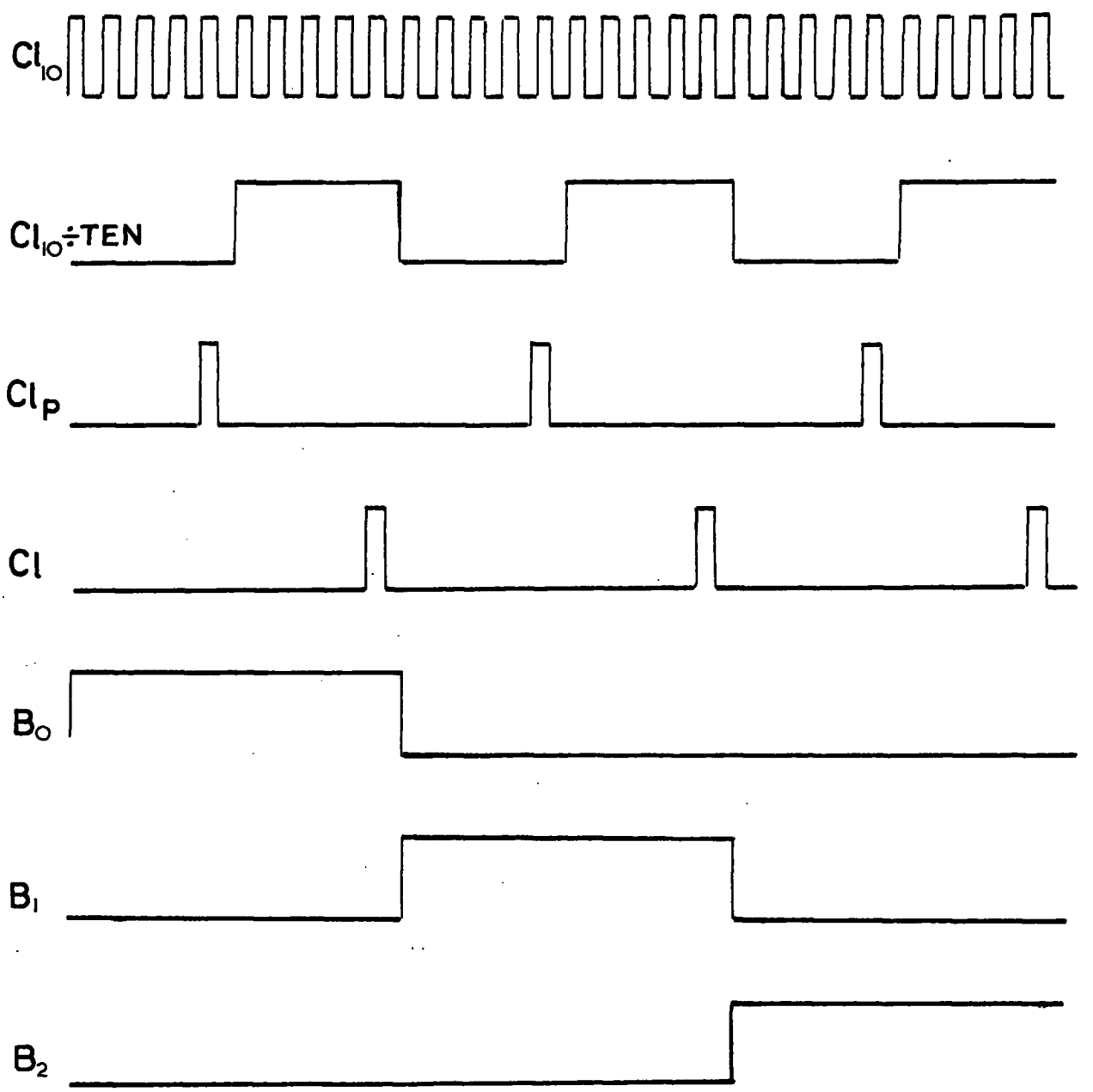


Fig.16

The gate G.14 is used to transfer the operand address to the memory during B_7 time. When operand address modific--ation is required, bit 5 in the control register will be true. The first six bit times are allocated for operand address modification in which the contents of the modifier register are added to the contents of the control register (bits 6 to 11) using the single bit full adder. After modification the operand address is available for transfer to the memory in the normal way.

The six stage instruction counter specifies the drum storage address from where the next instruction is to be obtained. This counter is normally up-dated, by adding one, after each instruction has been executed. In the case of the two instructions which specify 'jumps' in the programme.the address of the next instruction will be contained in the control register (bits 6 to 11) and will be conditionally transferred to the instruction counter during B_2 time.

The individual timing pulses, required to initiate each of the minor commands which are needed for the execution of all instructions, are produced in the timing pulse generator using simple gating techniques.

5.4 Final Outline Design of the Arithmetic Unit.

The major sub-sections of the arithmetic unit must be the four 12 bit registers and the 11 bit full adder. The

block schematic, Fig. 14, illustrates the way in which these components are interlinked by the control gates.

(i) Addition.

The 'addition' operation is initiated by placing the two operands in the register A and in the accumulator register. (The accumulator register is also referred to as " The ACC. register"). The contents of the ACC. register is then transferred to the temporary storage register B. A re-set pulse is subsequently applied to the ACC. register. The true outputs of each bistable in both registers A and B are then simultaneously transferred to appropriate orders of the 11 bit parallel adder. The sum outputs of the adder are subsequently gated into the ACC. register. The operand initially stored in register A remains unaltered after this operation.

(ii) Subtraction.

Subtraction is performed using the one's complement method. The subtrahend is initially placed in the register A and the minuend in the ACC. register. The contents of the ACC. register are transferred to the register B and the ACC. register reset. The false output of each stage in register A and the true output of each stage in register B are simultaneously applied to appropriate orders of the parallel adder. The carry output of the adder is gated to the least significant stage of the adder to maintain the 'end-around-

carry' which is necessary with this method. The sum outputs of the adder are finally gated to the ACC. register to give the difference.

(iii) Multiplication.

The process used for multiplication differs slightly from the method described in Section 3.4.2. The multiplicand is initially stored in the register A and the multiplier in the register M. The least significant digit in the register M is tested for a logic 1. If the test is positive the contents of the register A are added to the zero contents of the ACC. register. The sum outputs of the parallel adder, i.e. the first sub-product, are then gated to the ACC. register If the least significant digit test is negative the add operation is not performed. The contents of the registers A and M are then shifted one place left and right respectively. The process of testing the least significant digit of register M, adding, shifting etc. is repeated until all the digits in the register M have been tested after which time the final product is held in the ACC. register.

This method of multiplication pre-supposes that the product is within the range of the machine, i.e. 11 binary digits, The product is built up in the ACC. register and due to the left shifting in the register A the least signif--icant digit in the product is retained. The limitations imposed by this method were agreed to be of little consequence

compared with the complexity involved in both rounding off the product and in maintaining a clear demonstration.

(iv) Division.

The division operation is certainly the most complex of these four arithmetic processes. The dividend is initially stored in the ACC. register and the divisor in the register A. The quotient is developed in the register M and is transferred to the ACC. register on completion of the division operation. Initially, both the dividend and divisor are shifted left until their respective most signif--icant one digits occupy position 11 in the registers. The difference in the number of shift pulses applied to each register is counted and stored in the ' 4 bit count' and used to determine the number of digits in the quotient. The contents of the register A are then subtracted from the contents of the ACC. register. If the remainder is positive a 1 is stored in position ten of register M, and the positive remainder gated into the ACC. register. If the remainder is negative, a 0 is stored in position ten of the register M and the contents of the ACC. register remain unaltered. The contents of the registers A and M are then shifted one place right. A pulse is applied, to reduce by one, the number stored in the '4 bit count', which is subsequently tested for zero. The process of trial subtraction , right shifting etc. is repeated until the

'4 bit count' shows zero. Finally the quotient is serially transferred to the ACC. register from the register M.

It was agreed that the maximum seven digit quotient resulting from the execution during three character times would be sufficient for demonstration purposes.

5.5 The Machine Instructions.

In Sections 5.3 and 5.4 the required operation of the control and arithmetic units has been described. Each must function in accordance with up to the maximum number of allowed machine instructions (i.e. sixteen). By considering the operation in detail it was possible to list what these instructions should be, and sixteen was found to be adaquate.

The sixteen basic machine instructions suitable for all the operations of the control and arithmetic units plus the transfer of information to and from the memory units were found to be as follows:

I_0-- Transfer the contents of memory location n to the ACC. register.

I_1-- Transfer the contents of memory location n to the register A.

I_2-- Print the contents of the ACC. register.

I_3-- Add the contents of the register A to the contents of the ACC register, placing the sum in the ACC. register.

I_4--Transfer the contents of the ACC. register to memory location n.

I_5--Transfer the contents of memory location n to the register M.

I_6--Transfer the contents of the ACC. register to the register B.

I_7--Multiply the contents of the register A by the contents of the register M, placing the product in the ACC. register.

I_8--Divide the contents of the ACC. register by the contents of the register A, placing the sum in the ACC. register.

I_9--Shift the contents of the ACC. register n places left.

I_{10}-Subtract the contents of the register A from the contents of the ACC. register, placing the difference in the ACC. register.

I_{11}-If the contents of the modifier register are positive take the next instruction from location p, otherwise take the next instruction in sequence.

I_{12}-Read in instructions and data from input devices.

I_{13}-If the contents of the ACC. register are negative take the next instruction from location p, otherwise take the next instruction in sequence.

I_{14}-Stop.

I_{15}-Load modifier register with the number n.

5.6 The Order Code.

To operate the machine, each of the instructions listed in Section 5.5 must give rise to a particular sequence of basic logic functions occurring at the correct times. Each instruction is executed by the application of the appropriate sequence of timing pulses to the various gates and registers throughout the machine.

By considering the operations required for each instruc--tion a list of the basic functions and their timing codes was prepared. This list is specified for each instruction. The time is defined by the character time (K time K_1,K_2,K_3.) and by bit time (B_0------ B_{11}). Each of the timing functions has been allocated a number between F.1 and F.53.

The complete list of instructions, their associated codes and breakdown into minor functions, fully describes the proposed method of execution. The list is also used to establish all the timing functions detailed in Chapter 6 and listed in Appendix 1. Reference is made to Fig. 13 and Fig. 14.

Machine Instruction I_0 Code 0000

Transfer the contents of memory location n to the ACC. register.

Timing Function.	Time.	Operation
F.1	K_1B_1	Reset ACC. register via OR 5.
F.3 F.40 - F.45	$K_1(B_1--B_6)$	Check instruction for operand address modification. If CR.5 is 1, add the contents of the modifier register to the contents of the control register (bits 6 to 11) placing the sum in the control register. (The control gates are G.15, G16 and G.17).
F.6	K_1B_8	Transmit pulse to memory to release data to core buffer register.
F.7	K_1B_9	Gate the contents of the core buffer registervto the ACC. register via loading gate G.8.
F.8	K_1B_{10}	Transmit pulse to memory to re-write data.
F.9	K_1B_{10}	Apply pulse to instruction counter via OR 15.
F.12	K_1B_{10}	If CR.5 is 1, apply pulse to modifier register.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_1 Code 0001.

Transfer the contents of memory location n to the register A.

Timing Function.	Time.	Operation
F.13	K_1B_2	Reset register A.
F.3 F.40 F.41 F.42 F.43 F.44 F.45	K_1B_1--B_6	Check instruction for operand address modification. If CR. 5 is 1, add the contents of the modifier register to the contents of the control register(bits 6 to 11) placing the sum in the control regi--ster. (Control gates are G.15, G.16 and G.17.)
F.6	K_1B_8	Transmit pulse to memory to release data to core buffer register.
F.14	K_1B_9	Gate the contents of the core buffer register to register A via gate G.1.
F.8	K_1B_{10}	TTransmit pulse to memory to re-write data.
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.12	K_1B_{10}	If CR.5 is 1, apply pulse to reduce the contents of the modifier register by one.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_2 Code 0010.

This instruction is reserved for print-out, it is proposed that during execution time the contents of the ACC. register be transferred to twelve bistable elements associated with a combined Binary / Decimal convertor and

paper tape printer. This section of the equipment is not being built at present and therefore the minor operations associated with this instruction cannot be specified.

Machine Instruction I_3 Code 0011.

Add the contents of the register A to the contents of the ACC. register, placing the sum in the ACC. register

Timing Function	Time.	Operation
F.15	K_1B_1	Gate the contents of the ACC. register to register B, via G.4 and OR.4
F.1	K_1B_1	Reset the ACC. register via OR.5.
F.16	K_1B_2	Gate the true side of register A to X input of adder via G.2 and OR.2.
F.17	K_1B_2	Gate contents of register B to Y input of the adder, via G.5 and OR.3.
F.18	K_1B_2	Gate sum output of adder to the ACC. register via G.7 and OR.3.
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.10	K_1B_{11}	Reset time base.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_4 Code 0100.

Transfer the contents of the ACC. register to memory location n.

Timing Function.	Time.	Operation.
F.3 F.40 F.41 F.42 F.43 F.44 F.45	$K_1(B_1--B_6)$	Check instruction for operand address modification. If CR.5 is 1, add the contents of the modifier register to the contents of the control register (bits 6 to 11) placing the sum in the control register. (Control gates are G.15, G.16 and G.17)
F.19	K_1B_9	Gate the contents of the ACC. register to the core buffer register.
F.8	K_1B_{10}	Transmit pulse to memory to write data.
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.12	K_1B_{10}	If CR.5 is 1, apply pulse to reduce the contents of the modifier register by one.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_5 Code 0101.

Transfer the contents of memory location n to the register M

Timing Function	Time.	Operation.
F.20	K_1B_2	Reset register M
F.3 F.40 F.41 F.42 F.43 F.44 F.45	$K_1B_1--B_6$	Check instruction for operand address modification. If CR.5 is 1, add the contents of the control register (bits 6 to 11) to the contents of the modifier register placing the sum in the control register. (Control gates are G.15, G.16 and G.17).
F.6	K_1B_8	Transmit pulse to memory to release data to core buffer register.
F.21	K_1B_9	Gate contents of store buffer register to the register M via loading gate G.9.

F.8	K_1B_{10}	Transmit pulse to memory to re-write data.
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.12	K_1B_{10}	If CR.5 is 1, Apply pulse to reduce the contents of the modifier register by one.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_6 Code 0110.

Transfer the contents of the ACC. register to the register B.

Timing Function.	Time.	Operation.
F.15	K_1B_1	Gate the contents of the ACC. register to register B via gate G.4 and OR.4.
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_7 Code 0111.

Multiply the contents of the register A by the contents of register M placing the product in the ACC. register.

Timing Function.	Time.	Operation.
F.1	K_1B_1	Reset ACC. register via OR.5.
F.22	K_1B_1	Adjust sign via Ex.-OR.2, AND.3 and OR.9
F.16 F.17 F.18	K_1B_2	Examine least significant digit in register M. If positive add the contents of register A to the contents of the ACC. register, placing the sum in the ACC. register (See instruction I_3).
F.2	K_1B_2	Reset register B.
F.23	K_1B_3	Shift the contents of register M one place right via OR.12 and OR.13.

F.24	K_1B_3	Shift the contents of register A one place left via OR.7.
F.15	K_1B_4	Transfer the contents of the ACC. register to register B via G.4 and OR.4.
F.1	K_1B_4	If least significant digit in register M is 1, reset the ACC. register.
F.16 F.17 F.18 F.23	K_1B_5	Repeat operations specified during K_1B_2 time.
F.23 F.24	K_1B_6	Repeat operations specified during K_1B_3 time.
F.15 F.1	K_1B_7	Repeat operations specified during K_1B_4 time.

These last three groups of operations are repeated until K_3B_{10} time.

F.9	K_3B_{10}	Apply pulse to instruction counter.
F.10	K_3B_{11}	Reset timebase.
F.11	K_3B_{11}	Reset EXECUTE bistable.

Machine Instruction I_8 Code 1000.

Divide the contents of the ACC. register by the contents of of register A, placing the quotient in the ACC. register.

Timing Function.	Time.	Operation.
F.22	K_1B_1	Check sign of quotient via EX.-OR.3, AND 2, and OR.9.
F.24	K_1B_1	Shift the contents of register A left until most significant 1 digit is in position 11. (Gates OR.7 and AND 6.)
F.25	K_1B_1	Shift the contents of the ACC. register left until the most significant 1 digit is in position 11.(Gates OR.10 and AND.5)

F.26	K_1B_1	Gate the difference in the number of shift pulses between timing functions F.24 and F.25 to the '4 bit count', via Ex-OR.1 and OR.11.
F.15	K_1B_2	Transfer the contents of the ACC. register to register B via OR.4.
F.1	K_1B_2	Reset the ACC. register via OR.5.
F.51	K_1B_2	Set DIVIDE bistable.
F.20	K_1B_2	Reset register M.
F.27	K_1B_3	Gate the false side of each bistable in register A to the adder via OR.1 and G.3.
F.17	K_1B_3	Gate the contents of register B to the adder via OR.3 and G.5.
F.18	K_1B_3	Gate the sum outputs of the adder to the ACC. register via OR.3 and G.7.
F.38	K_1B_3	Set the CARRY bistable via AND 8 and AND 9.
F.23	K_1B_5	Shift the contents of register M one place right via OR.13 and OR.1.
F.24	K_1B_5	Shift the contents of register A one place right via OR.7.
F.26	K_1B_5	Apply pulse to '4 bit count'.
F.50	K_1B_5	If contents of '4 bit count' is zero, reset DIVIDE bistable.
F.15	K_1B_6	If CARRY bistable is set, transfer the contents of the ACC. register to register B.
F.1	K_1B_6	Reset the ACC. register.
F.17 F.27 F.18 F.38	K_1B_7	Repeat operations specified during K_1B_3 time.
F.23 F.24 F.26	K_1B_9	Repeat operations specified during K_1B_5 time.

F.15 F.1	K_1B_{10}	Repeat operations specified during K_1B_6 time.

These last three groups of operations are repeated sequentially until the DIVIDE bistable is reset.

F.23	K_3B_{10}	Shift the contents of register M ten places right.
F.25	K_3B_{10}	Shift the contents of the ACC. register ten places left.
F.9	K_3B_{L0}	Apply pulse to instruction counter.
F.10	K_3B_{11}	Reset timebase.
F.11	K_3B_{11}	Reset EXECUTE bistable.

Machine Instruction I_9 Code 1001.

Shift the contents of the ACC. register n places left.

Timing Function.	Time.	Operation.
F.33	K_1B_0	Reset SHIFT LEFT counter.
F.34	K_1B_1	Transfer the contents of the control register (bits 8 to 11) to SHIFT LEFT counter.
F.35	K_1B_3	Apply pulses to reduce contents of SHIFT LEFT counter to zero.
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_{10} Code 1010.

Subtract the contents of register A from the contents of the ACC. register, placing the difference in the ACC. register.

Timing Function.	Time.	Operation.
F.15	K_1B_1	Transfer the contents of the ACC. register to register B via OR.1 and G.3.
F.1	K_1B_1	Reset the ACC. register via OR. 5.
F.27	K_1B_2	Gate false side of each bistable in register A to the adder via OR.1 and G.3.
F.17	K_1B_2	Gate the contents of register B to the adder via OR.3 and G.5.
F.18	K_1B_2	Gate the sum outputs of the adder to the ACC. register via OR.3 and G.7
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_{11} Code 1011.

If the contents of the modifier register are positive take the next instruction from location p, otherwise take the next instruction in sequence.

Timing Function.	Time.	Operation.
F.28	K_1B_1	If the contents of the modifier register are positive, reset instruction counter.
F.29	K_1B_2	If the contents of the modifier register are positive, transfer the contents of the control register (bits 6 to 11) to instruction counter via G.12.

F.9	K_1B_{10}	If the contents of the modifier register are not positive,;apply pulse to instruction counter.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_{12} Code 1100.

It is proposed that this instruction be reserved for reading in data and instructions. When this instruction appears in the control register the timebase will be turned off and the information transferred from a tape or card reader by means of pulses generated to synchronise with the input devices. Initially data input will be under manual control and the interfacing of input devices may be dealt with as a future project. The minor operations associated with this instruction are therefore not specified.

Machine Instruction I_{13} Code 1101.

If the contents of the ACC. register are positive, take the next instruction from location p, otherwise take the next instruction in sequence.

Timing Function.	Time.	Operation.
F.28	K_1B_1	If false side of stage 12 of the ACC. register is 1, reset instruction counter.
F.29	K_1B_2	If false side of stage 12 of the ACC. register is 1, transfer the contents of the control register (bits 6 to 11) to the instruction counter via G.12.

F.9	K_1B_{10}	If false side of stage 12 of the ACC. register is 0 apply pulse to instruction counter.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

Machine Instruction I_{14} Code 1110.

Stop.

Timing Function.	Time.	Operation.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.
F.52	K_1B_{11}	Reset START bistable.

Machine Instruction I_{15} Code 1111.

Load modifier register with the number n.

Timing Function.	Time.	Operation.
F.31	K_1B_1	Reset modifier register.
F.32	K_1B_2	Transfer the contents of the control register (bits 6 to 11) to the modifier register via G.13.
F.9	K_1B_{10}	Apply pulse to instruction counter.
F.10	K_1B_{11}	Reset timebase.
F.11	K_1B_{11}	Reset EXECUTE bistable.

The timing chart (A), Fig. 17, conveniently summarizes the information given in the order code.

TIMING CHART (A)

INSTRUCTION TIME (K_0) — EXECUTION TIME (K_1)

Instruction	B_{11} (K_0)	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}
I_0	LOAD. ACC. FROM. MEMORY.	RESET ACC; RESET 'B'	←	CHECK MODIFIER REG. FOR ZERO; IF NOT ZERO ADD TO CONTROL REG. (6-11) SUM IN CONTROL REG.					→	TRANS DATA. FROM MEM. TO STORE B.	GATE. STORE. BUFFER. TO ACC.	RE-WRITE DATA.	OP. COMP.
I_1	LOAD. REG 'A' FROM. MEMORY.	RESET 'A'	←	AS. FOR. I_0					→	AS FOR I_0	GATE. STORE. BUFFER. TO REG 'A'	RE-WRITE DATA	OP. COMP.
I_2	PRINT OUT												OP. COMP.
I_3	ADD REG A TO ACC	RESET 'B'	GATE. ACC. TO 'B' RESET ACC	GATE 'A' AND 'B' TO ADDER SUM TO ACC									OP COMP.
I_4	TRANSFER ACC TO MEM		←	AS	FOR I_0				→		GATE. ACC TO MEMORY	WRITE. DATA.	OP COMP.
I_5	LOAD REG M. FROM MEM	RESET REG 'M'	←	AS	FOR I_0				→	AS FOR I_0			
I_6	TRANSFER ACC. TO B.	RESET REG 'B'	GATE ACC TO REG B										OP. COMP.
I_7	MULTIPLY A BY M	i) SET M[illegible] ii) RESET B iii) RESET ACC	i) SET 'I' DIGIT COUNT ii) ADJUST SIGN	L.S.D. OF M IF '1' ADD ii) ADD I TO DIGIT COUNT XI	i) SHIFT M ii) SHIFT A. iii) RESET B iv) RESET M IF COMPLETE YI	i) GATE ACC TO REG B ii) RESET ACC Z	X_2	Y_2	Z_2	X_3	Y_3	Z_3	X_4
I_8	DIVIDE ACC. BY REG 'A'	i) SET DIVIDE. F.F. ii) RESET B	i) ALIGN A & ACC ii) ADJUST SIGN	i) TRANSFER ACC TO B ii) RESET ACC iii) RESET M	i) SUB 'A' FROM ACC. ii) SET CARRY FF S	i) GATE CARRY TO M ii) ADD '1' TO DIGIT C.T. T	i) SHIFT M & A ii) RESET B iii) RESET D U	TRANSFER ACC TO B ii) RESET ACC V	S	T	U	V	S
I_9	SHIFT ACC LEFT N PLACES		←	APPLY B. IS CHE	PULSES CKING	TO ACC S.L.	SHIFT COUNTER	LINE. FOR	VIA. ZERO.	GATE	WHICH	→	OP. COMP.
I_{10}	SUB. A. FROM ACC	RESET B	GATE ACC TO B RESET ACC	GATE A & B TO ADDER SUM TO ACC									OP. COMP.
I_{11}	JUMP TO INST. P	RESET INST COUNTER	GATE C. REG. TO INST. COUNTER										OP. COMP.
I_{12}	TEST. ACC IF +VE JUMP TO INST P	RESET I. COUNT IF ACC +VE	GATE C. REG. TO INST. COUNTER IF ACC. +VE.										NO OP. COMP. PULSE. IF ACC. +VE.
I_{14}	STOP												RESET. CLOCK. PERM.
I_{15}	PLACE N IN MODIFIER		RESET MODIFIER REG	GATE. C.R. TO MODIFIER									OP. COMP.

K_2

Instruction	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_0
I_0													
I_1													
I_2													
I_3													
I_4													
I_5													
I_6													
I_7	Y_4	Z_4	X_5	Y_5	Z_5	X_6	Y_6					OP COMP	
I_8	T	U	V	S	T	U	V	S	T	U	V SHIFT M TO ACC.	OP COMP	

DIAGRAM INDICATING SEQUENCE OF OPERATIONS DURING EXECUTION TIME.

Fig.17

It will be apparent from the order code, that some timing functions are associated with more than one instruction. The Boolean expressions for these timing functions must therefore combine a number of terms in an OR configuration. It is convenient to include in these expressions an additional term, 'student' (St.25), to meet the requirements for Standstill operation which were given in Section 4.9. The expression for the timing pulse F.25 is given here to illustrate the method.

$$F.25 = I_8 K_1 B_1 \overline{ACC_{11}} C_{L10} + I_9 K_1 B_3 C_{L10} + I_8 K_3 B_{10} C_{L10} + St.25$$

The terms for each of the timing pulses are deduced from the order code and from a consideration of the timing chart (B), Fig. 18.

TIMING CHART (B)

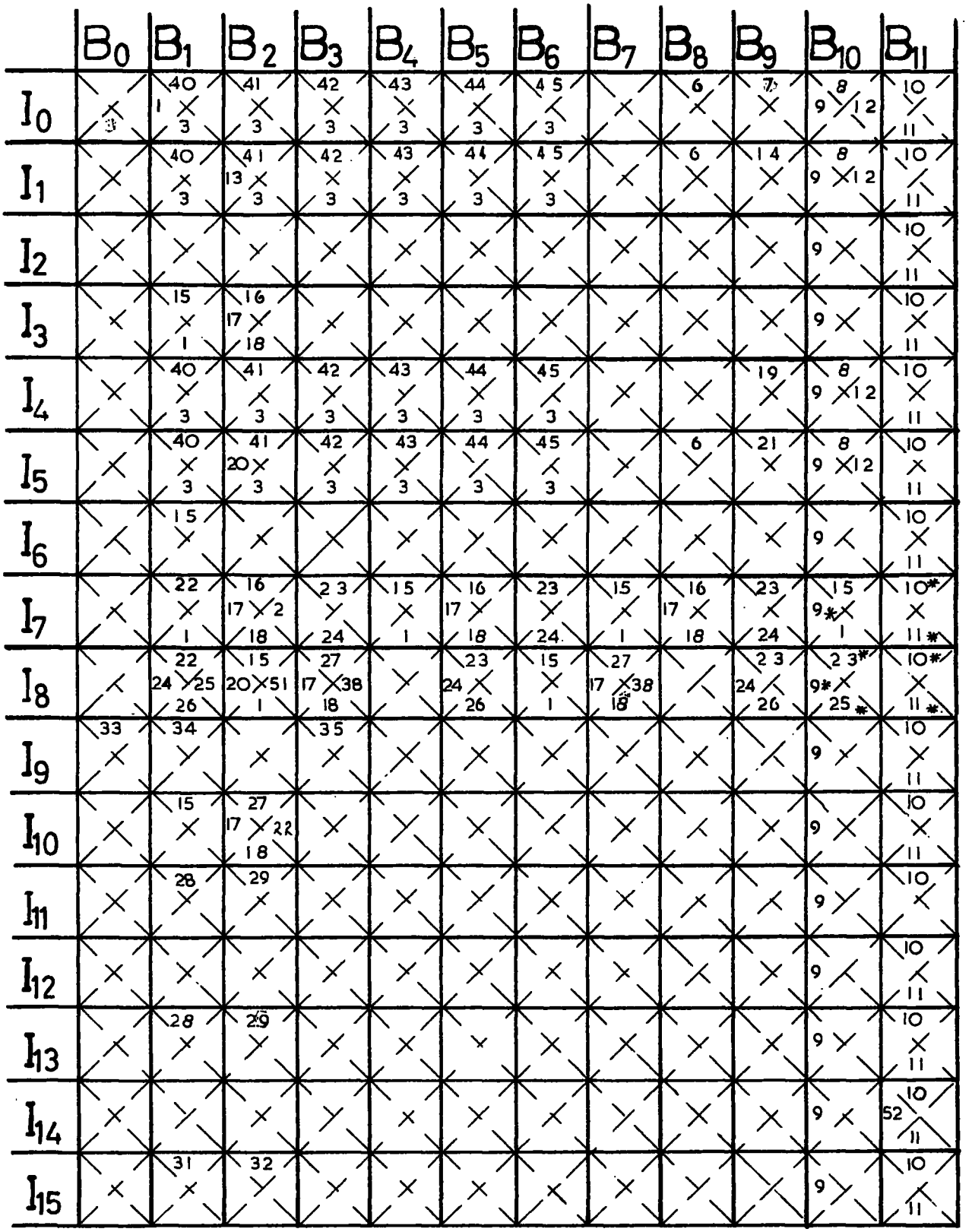


Fig. 18

CHAPTER VI.

DETAILED CIRCUIT DESIGN.

Because of the decision to use one particular family of integrated circuit logic elements to construct the computer, it was necessary to prepare detailed logic diagrams. This was done using the block schematics, Fig.13 and Fig.14, the order code, Section 5.6, and the specification of the chosen family of logic elements. The choice and characteristics of the elements are discussed in Section 6.1 of the present Chapter. The remaining sections deal with the methods employed to meet the specification of Section 5.1 in the light of the overall operation described later in Chapter 5. The conventional practice of making continued reference to the logic schematics is adopted in presenting the discussion of the detailed circuit design.

6.1 Choice of Integrated Circuits.

6.1.1 Practical Considerations.

It was established, in Section 2.4, that integrated circuit elements must be employed, so that the equipment should be characteristic of modern computing techniques and also that it should not be out-dated before completion. In deciding which family of integrated circuit elements to employ it was necessary to examine those which were currently available, in terms of cost, method of assembly and their

long term development.

Before attempting the actual circuit design, a cost estimate was made of the elements required to construct a twelve bit shift register and its associated loading and output gates. This estimate was based on the price per logic element, even though some logic blocks contain more than one element. The cheapest suitable range from each manufacturer was selected and used in the estimate.

The following estimate was completed in April 1967.

(i) S.G.S. Fairchild - RTuL) 900 series.

12 off, type 926, J-K bistable	@ 36/7	£22-00-00
24 off, type 910, two-input gate	@ 19/6	£23-08-00
		£45-08-00

(ii) Ferranti - ZSS - 130 series.

12 off, type ZSS131B bistable	@ 54/6	£32-14-00
24 off, type ZSS133B two-input gate	@ 34/3	£40-00-00
		£72-14-00

(iii) Texas Instruments - T.T.L. - 74N series.

12 off, type SN7474N, bistable	@ 27/-	£16-04-00
24 off, type SN7400N, two-input gate	@ 9/3	£11-02-00
		£27-06-00

(iv) Motorola - R.T.L. - MC700P series.

12 off, type MC790P bistable	@ 10/10½	£6-10-06
24 off, type MC717P two-input gate	@ 5/1½	£6-03-00
		£12-13-06

It was obvious, from the results of this estimate, that the choice lay between the Texas Instrument 74N series and the Motorola MC700P series. The dual-in-line encapsula--tion in both cases was attractive, because it permitted the use of double-sided printed circuit boards. This latter point was most important; it was agreed that double-sided boards were within the manufacturing capabilities of the university workshops; whereas the use of the T-05 encapsula--tion in either the S.G.S. Fairchild or Ferranti range would have necessitated the 'outside' manufacture of multi-layer printed circuits. It was predicted by senior members of the department, that future development effort of integrated circuits would probably be concentrated on T.T.L. devices, so that the range of available logic elements would be likely to increase. It was also predicted that considerable price reductions would follow. In choosing the Texas Instrument 74N series, due consideration was given to loading requirements, delivery dates and the availability, or otherwise, of application notes and specifications. It will be noted that the continued development of the 74N series has already led the manufacturer to introduce two cheaper ranges, the System 11 and Series 10. As these ranges are compatible with the 74N series, they have been used where appropriate with a saving in cost.

6.1.2 Texas Instruments 74N Series.

It is appropriate to summarize the main characteristics of this range, and to list those elements which have been used in the design and construction of the computer. The NAND gate is used as the basic function of this series of transistor - transistor - logic elements. The high fan-out capability (ten) is due largely to the use of the multiple emitter input transistor, which also accounts for the low propagation delay (typically 13nS per gate). The power dissipation associated with each gate is specified as 10mW with a 50% duty cycle, when operated from the specified two wire 5 volt supply. The logic levels used in the 74N series are specified as :-

Input; logic 1 $>$ 2v; logic 0 $<$ 0.8v.

Output; logic 1 $>$ 2.8v; logic 0 $<$ 0.4v.

The d.c. noise margin is specified as 1 volt.

The output impedences of the basic gate are quoted as 12 ohms for the logic 0 state and 70 ohms for the logic 1 state. The more detailed specifications for the more complex elements are given in Ref. 16.

The elements used in this equipment are as follows. In certain cases it was convenient to employ elements from the cheaper series 10 range.

7400N --- Quadruple two input NAND.

7410N --- Triple three input NAND.

7420N -- Dual four input NAND.

7430N -- Single eight input NAND.

7440N -- Dual four input NAND (power).

7402N -- Quadruple two input NOR.

7451N -- Dual EXCLUSIVE OR.

7474N -- Dual type D bistable.

7476N -- Dual type J-K bistable.

7480N -- Single binary full adder.

7483N -- Quadruple binary full adder.

7490N -- Single decade counter.

6.2 Detailed Design of the Arithmetic Unit.

This section deals with the detailed design of the various parts of the arithmetic unit showing how the require--ments are met in terms of logic elements which will be realised in practice with 74N series integrated circuits.

6.2.1 The Accumulator Register.

An accumulator, according to B.S. 3527 (17), will automatically add an incoming number to its present contents and store the sum in the accumulator. It will be appreciated that a number of minor commands are required to achieve this operation. In this computer the accumulator, or the ACC. register as it conveniently referred to, is much less sophisticated in its operation.

LOGIC SCHEMATIC AU 001

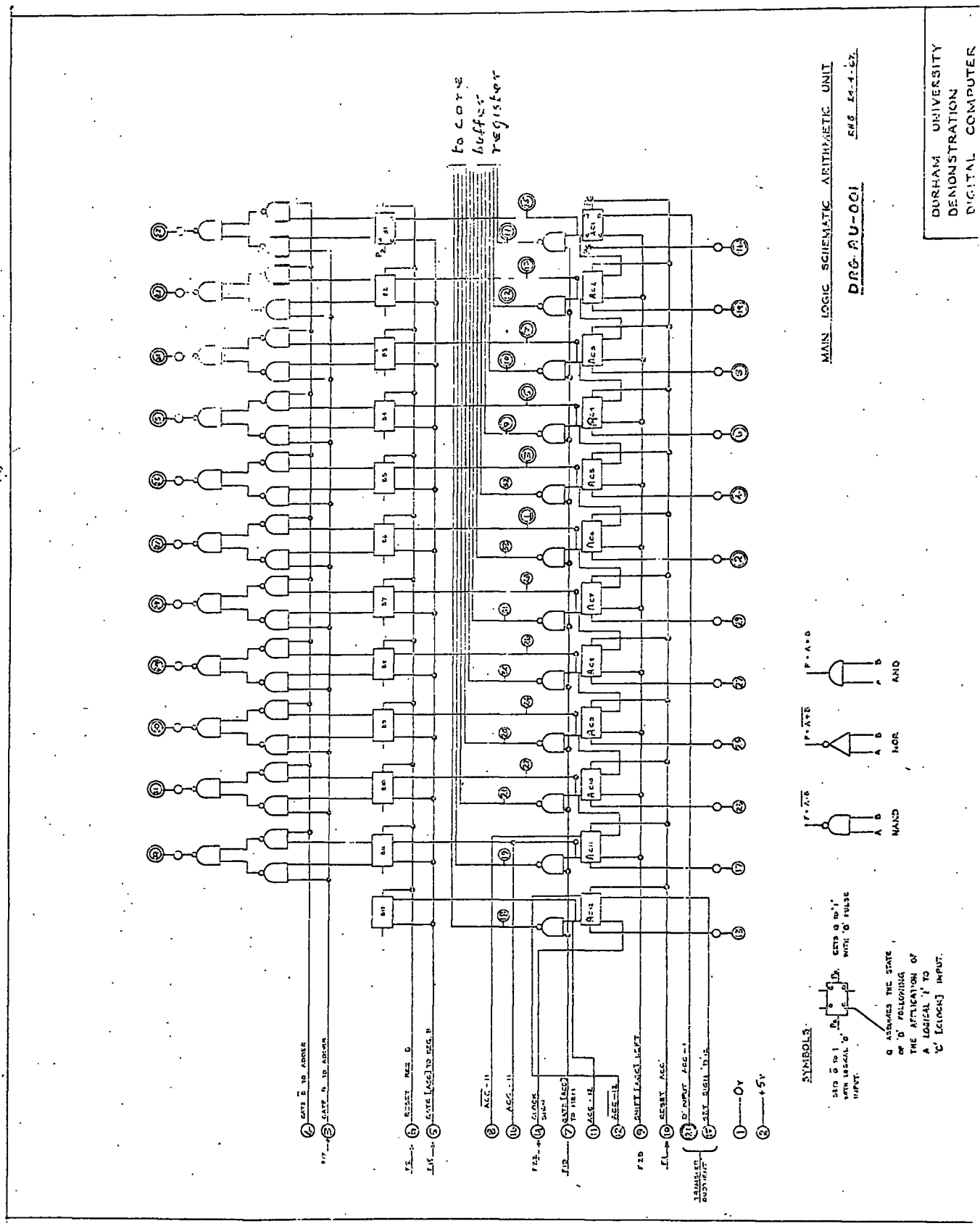


Fig.19

LOGIC SCHEMATIC A.U. 002

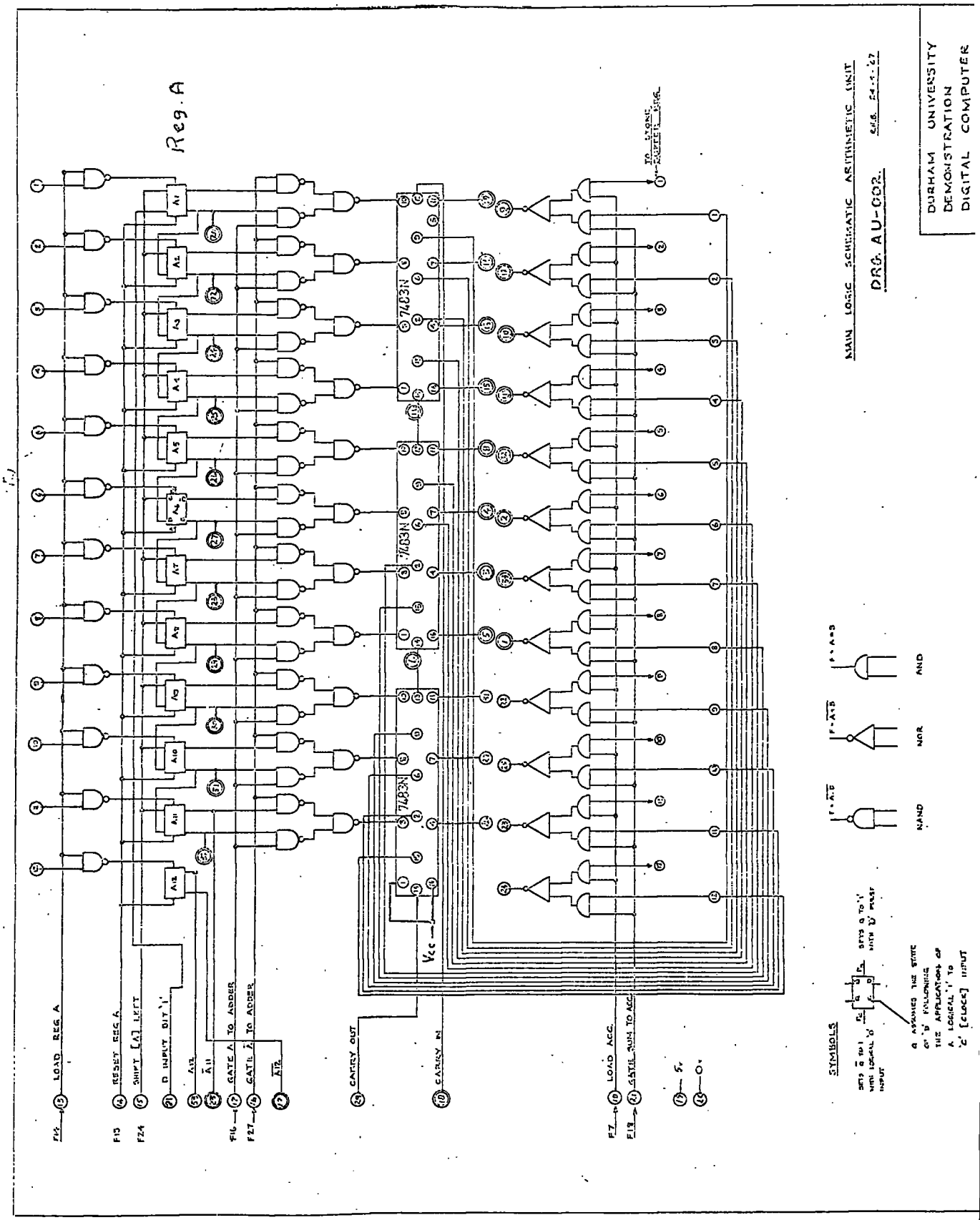


Fig. 20

LOGIC SCHEMATIC A.U. 003.

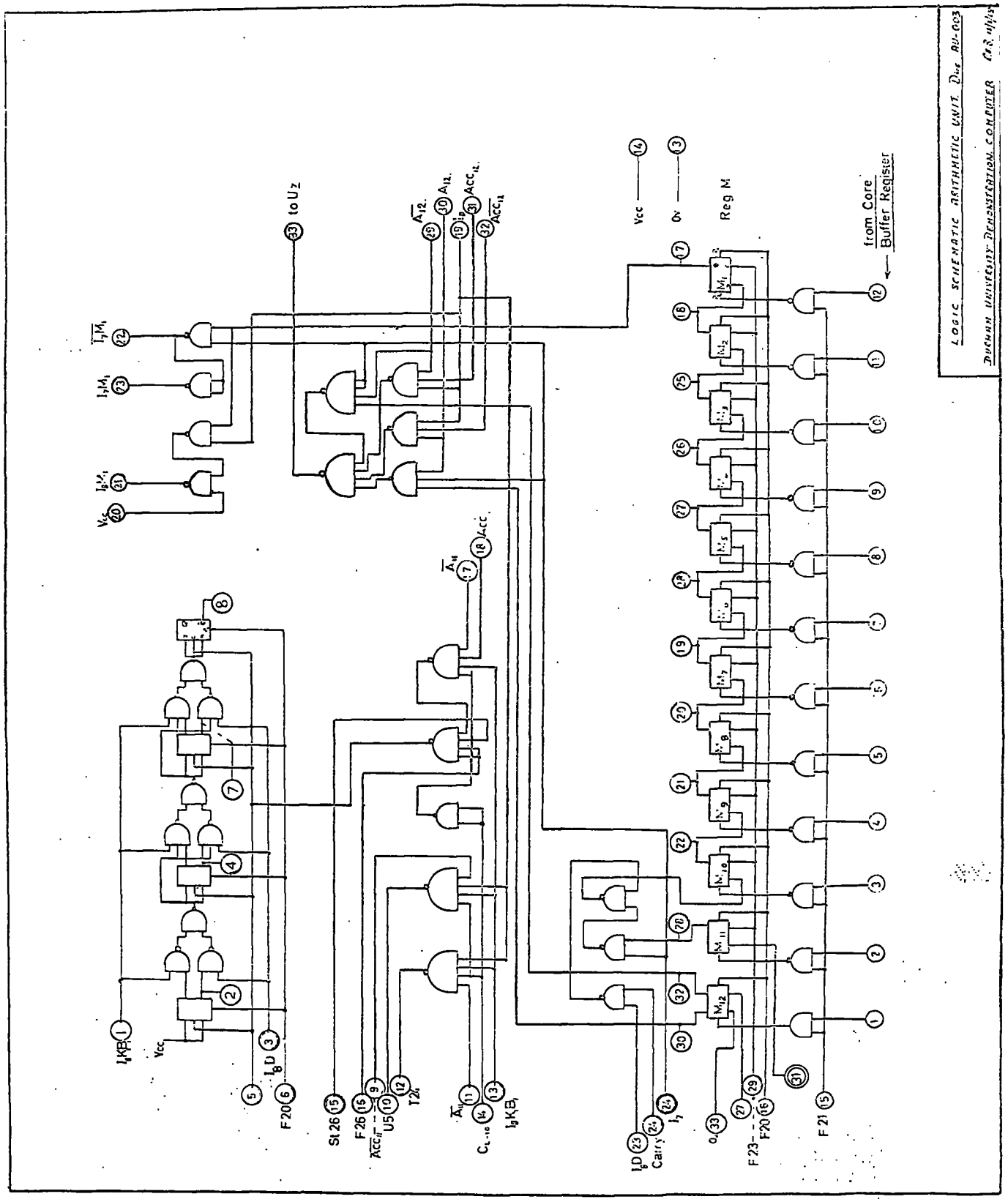


Fig. 21

LOGIC SCHEMATIC C.U. 001

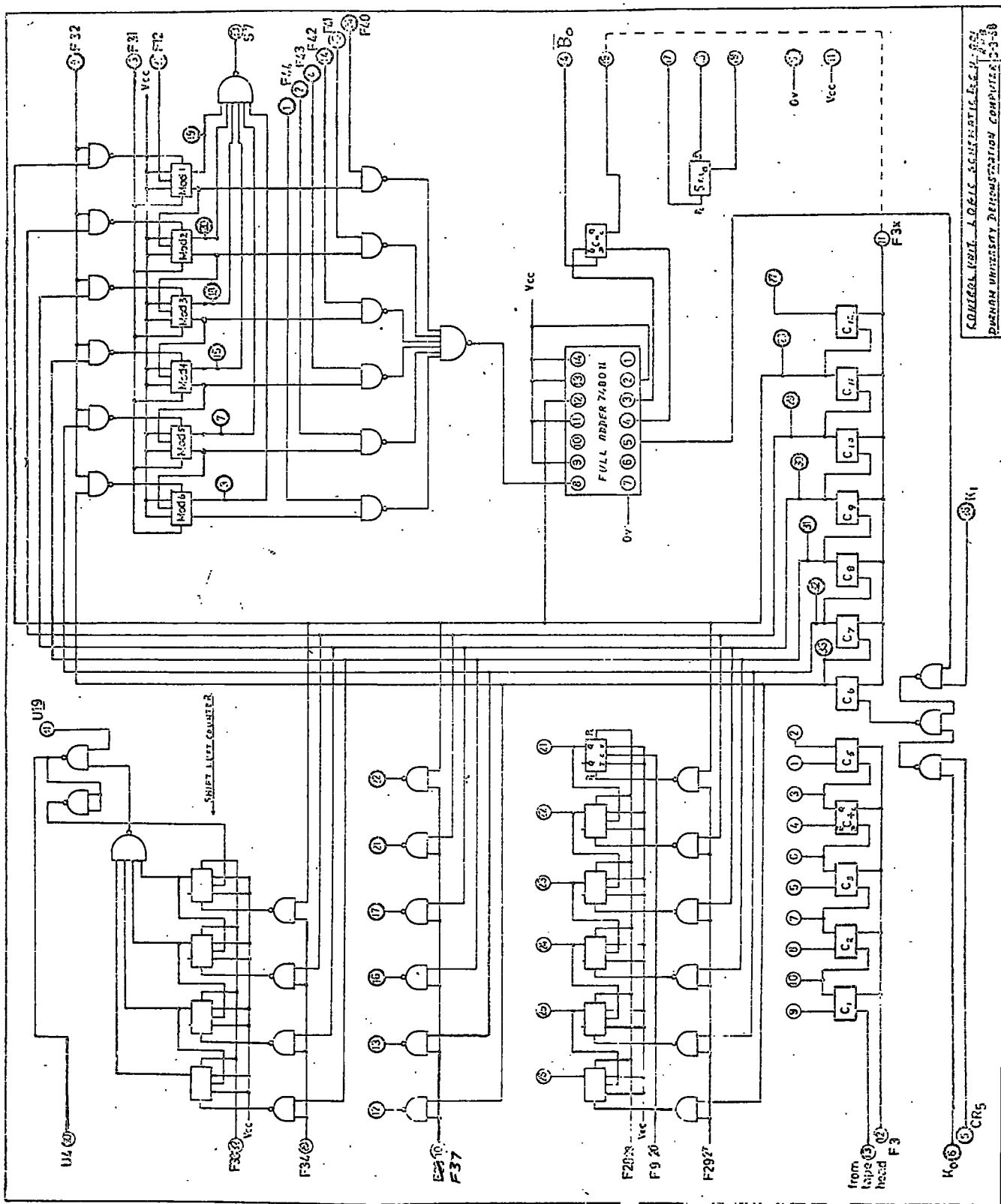


Fig. 22

LOGIC SCHEMATIC T.B. 001

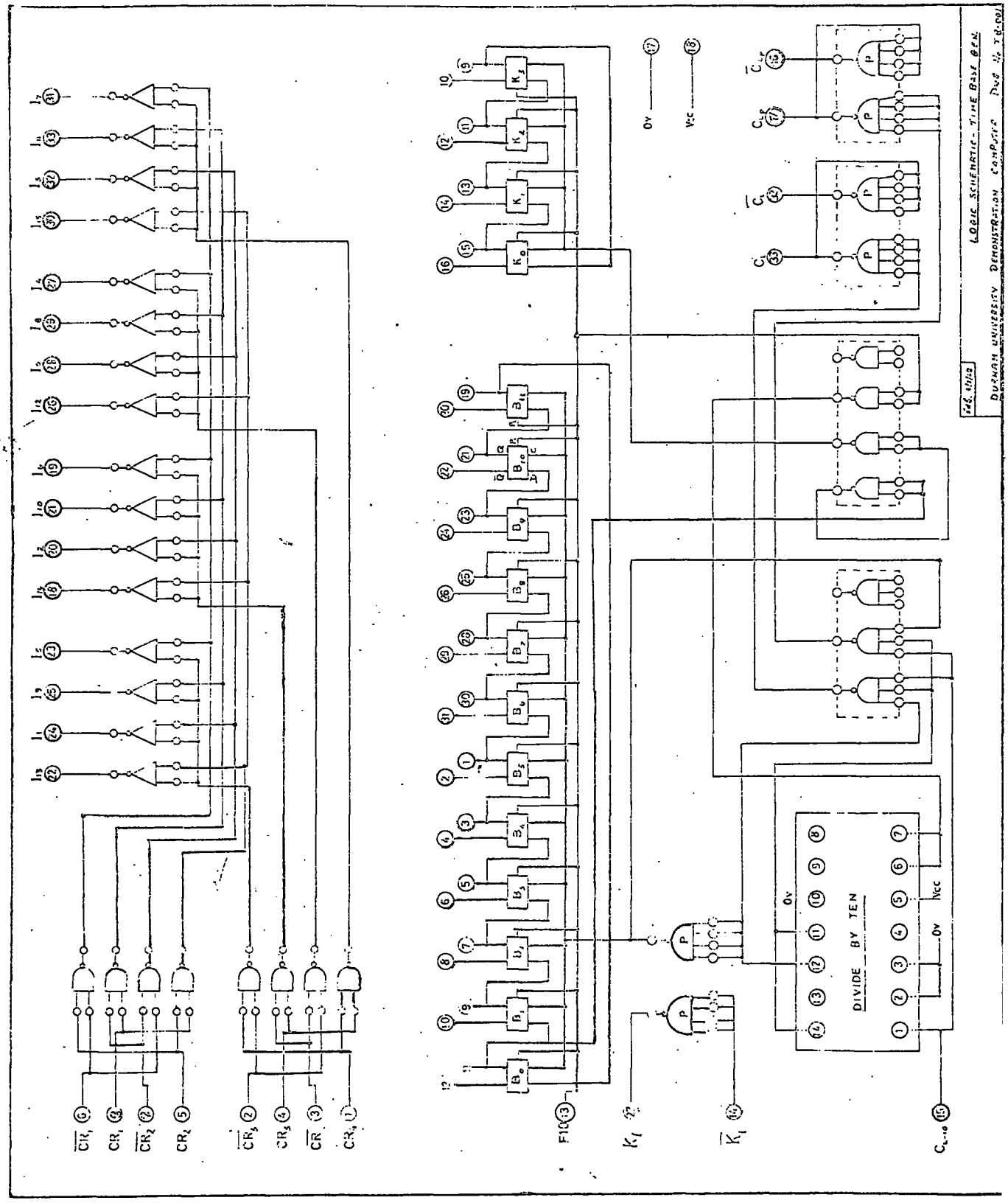


Fig. 23

LOGIC SCHEMATIC TFG 001

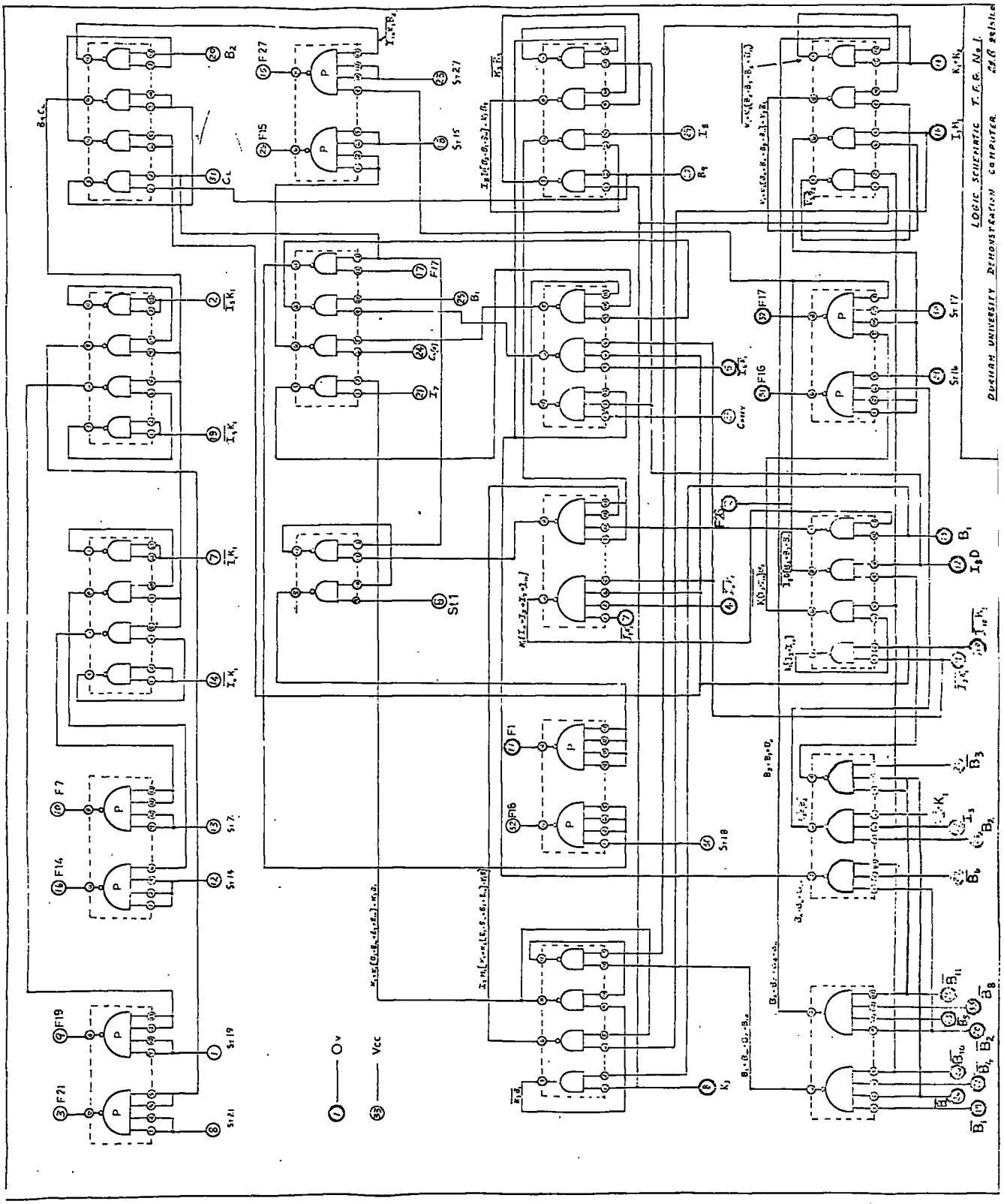


Fig. 24

LOGIC SCHEMATIC T.F.G. 002

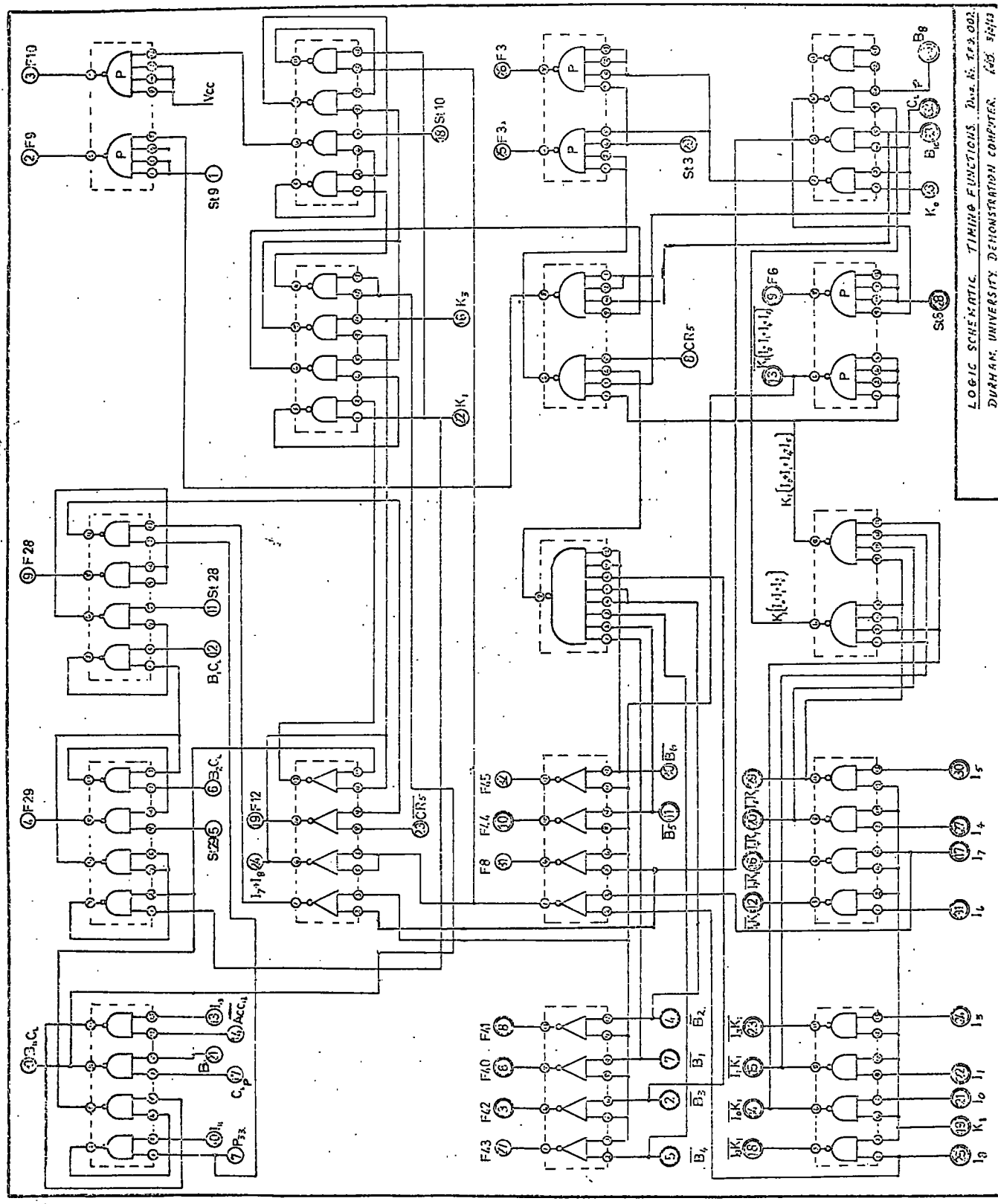


Fig. 25

LOGIC SCHEMATIC T.F.G. 003

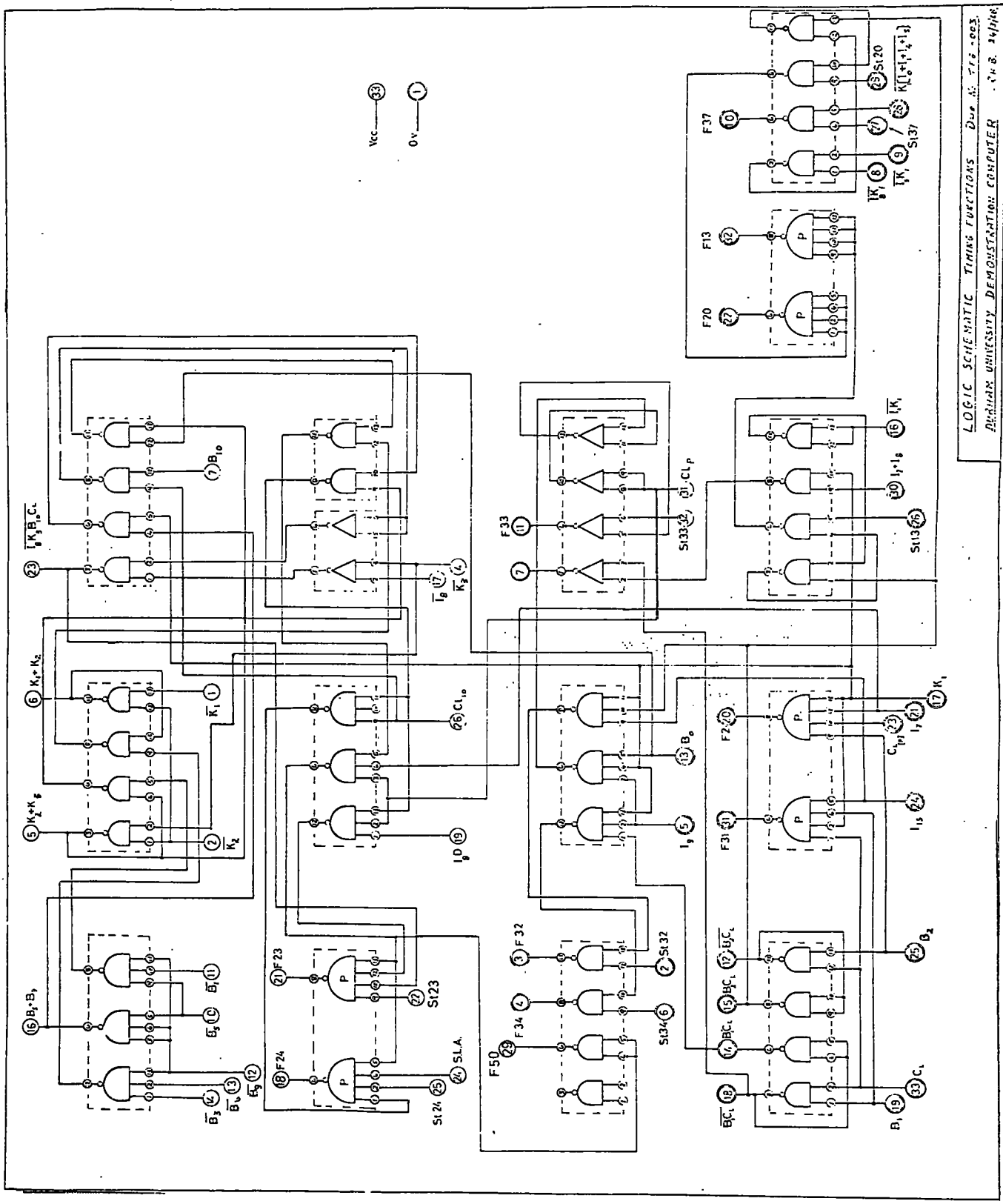
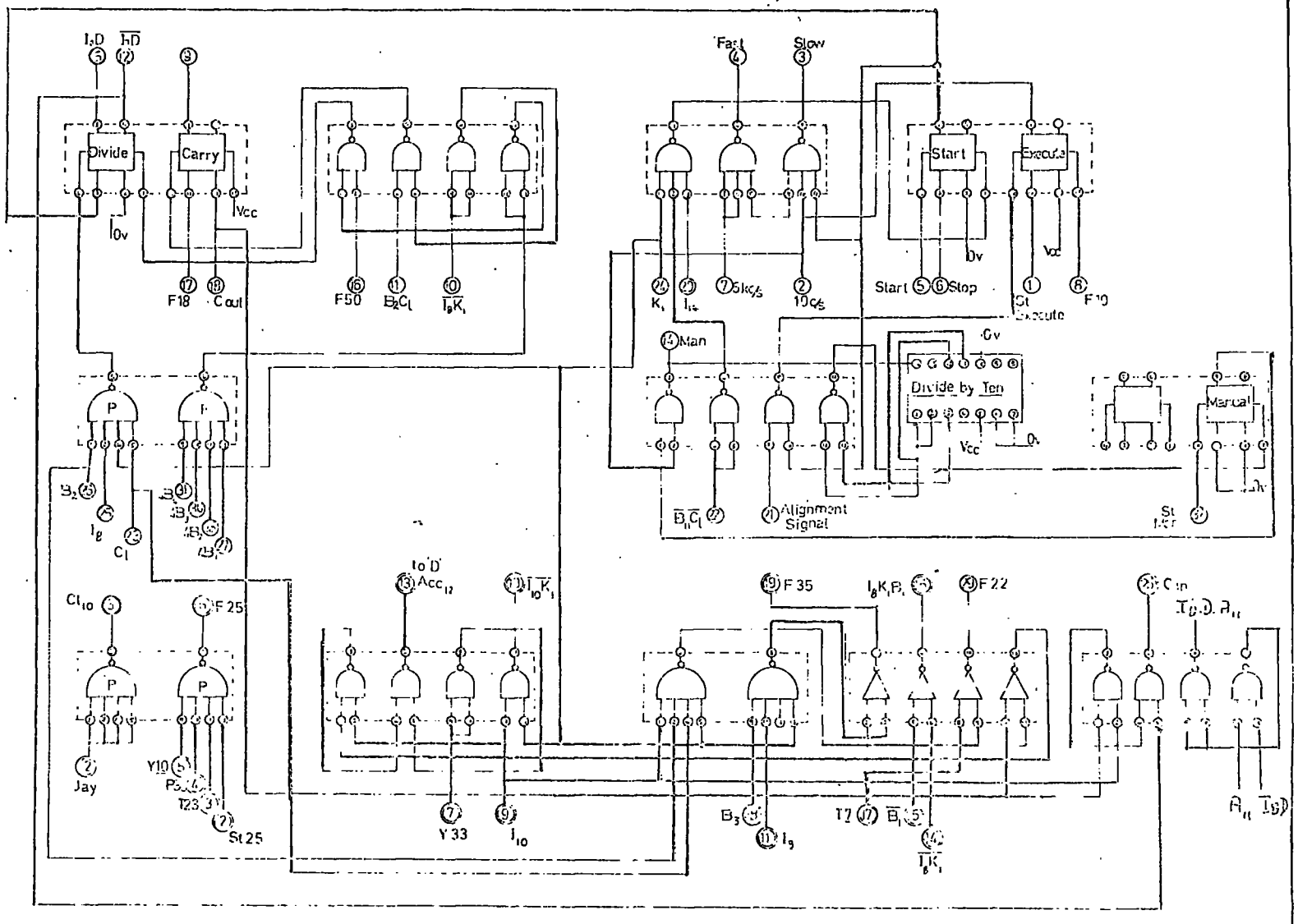


Fig. 26



LOGIC SCHEMATIC Sh. C1. No1

LOGIC SCHEMATIC Sh C1 Ed 1

DURHAM UNIVERSITY DEMODULATION GROUP A/D 6 5/1/65

Fig. 27

The general requirements for the ACC. register are :

(i) Left shifting facilities between stages 1 and 11.

(ii) Parallel entry from either the core buffer register or the sum outputs of the parallel adder.

(iii) Overall reset to zero.

(iv) Parallel transfer of the contents of the register to the core buffer register.

(v) Serial entry from the register M during the division instruction I_8.

The register is formed using twelve Type D bistable elements. The logic schematic, Fig. 19, shows the Q output of each bistable connected to the D input of the stage of immediate higher order. With this arrangement, the positive going timing pulse, F.25, applied simultaneously to the C input of each bistable, causes the contents of the register to be shifted one place left. The expression for the timing pulse F.25, derived from a consideration of the timing chart Fig. 18, is given by:

$$F.25 \equiv I_8.K_1.B_1.\overline{ACC_{11}}.C_{L10} + I_9.K_1.B_3.C_{L10} + I_8.K_3.B_{10}.C_{L10} + St.25$$

(See Figs. 27, 26, 22, 21.)

Parallel entry from two sources is achieved using the twelve EXCLUSIVE-OR gates shown in Fig. 20. The output of e each of these gates is directly coupled to the appropriate pre-set input of the bistables comprising the ACC. register.

The EXCLUSIVE-OR gate is convenient, as it provides the negative going output pulse when either of the two pairs of inputs are at logic 1. The timing pulse F.18, used to transfer the sum outputs of the adder to the ACC. register is derived using the gating circuitry shown in Fig. 24. The timing pulse F.18 is given by:

$$F.18 = I_3.K_1.B_2.C_L + I_{10}.K_1.B_2.C_L + I_8.D(B_3+B_7+B_{11})$$
$$I_7.M_1\left[K_1+K_2(B_2+B_5+B_8+B_{11})+K_3.B_2\right]C_L + St.18.$$

The timing pulse required to transfer the contents of the core buffer register to the ACC. register is given by:

$$F.7 = I_0.K_1.B_9.C_L + St.7.$$

(See Fig.24.)

The transfer of the contents of the ACC. register to the core memory unit is effected by applying the timing pulse F.19 to the commoned inputs of the twelve NAND gates associated with the Q output of each bistable. The output of each of these gates is connected to the corresponding preset inputs of the bistables which form the core buffer register. The positive going timing pulse F.19 is given by:

$$F.19 = I_4.K_1.B_9.C_L + St.19$$

(See Fig.24.)

Whenever parallel entry into the register is required it is necessary that the register be reset to zero before the application of either timing pulse F.18 or F.7. Resetting is done by applying the negative going timing

pulse F.1 to the pre-clear input of each bistable element. The expression for this pulse involves five different machine instructions and is:

$$\overline{F.1} = I_0.K_1.B_1.C_L + I_3.K_1.B_1.C_L + I_{10}.K_1.B_1.C_L$$
$$+ I_7M_1\left[(K_1+K_2)(B_1+B_4+B_7+B_{10})+K_3B_1\right]C_L$$
$$+ I_8\left[I_8D(B_2+B_6+B_{10}) + K_3.B_9\right]C_L + St.1$$

(See Fig.24)

The serial entry of the quotient from the register M is achieved by applying the signal I_8M_1 to the D input of stage 1 in the ACC. register. In order to transfer the quotient at the end of the division operation, the function $I_8K_3B_{10}C_{L10}$, which consists of ten pulses, is applied to the C inputs of the ACC. register and register M. Also during this time, the Q output of stage 1 in register M is applied to the D input of stage 1 in the ACC. register. This is done using two NAND gates, Fig. 21, which produce the function I_8M_1. During all other instructions this D input is held at the logic 0 level. The function $I_8K_3B_{10}C_{L10}$ is included in the expressions for timing pulses F.23 and F.25.

6.2.2 Register B.

During arithmetic operations involving the adder, it is necessary to employ the temporary register B to store the contents of the ACC. register, so that this register may be reset prior to the parallel entry of the sum outputs of the

adder. The register B, Fig.19, comprises twelve Type D bistable elements, with no provision for shifting. Entry of data from the ACC. register is effected by the application of timing pulse F.15 simultaneously to all stages in the register B. The D inputs of register B are directly connected to the Q outputs of the corresponding stages of the ACC. register. The timing pulse F.15, which is applied to the C inputs of register B, is defined by:

$$F.15 = I_6.K_1.B_1.C_{LP} + I_7\left[(K_1+K_2)(B_1+B_4+B_7+B_{11})+K_3.B_1\right]C_{LP}$$
$$+ I_8D.C_y(B_2+B_6+B_{10})C_{LP} + I_3.K_1.B_1.C_{LP} + I_{10}.K_1.B_1.C_{LP}$$
$$+ St.15.$$

(See Fig.24.)

The only occasion on which it is necessary to reset the register B is to form the first sub-product at the beginning of the instruction 'Multiply'. This is done by applying the negative going timing pulse F.2 simultaneously to the pre--clear inputs of each stage. The equation defining the pulse F.2 is given by:

$$\overline{F.2} = I_7.K_1.B_2.C_{Lp}$$

(See Fig.26.)

6.2.3 Register A.

From a consideration of the block schematic,Fig.14, it is evident that register A acts as an auxilliary store buffer register. This is necessary to demonstrate arithmetic proc--esses independently of the Control and Memory sections of

the machine. The register is required to have the following facilities:

(i) Overall reset to zero.

(ii) Parallel input from core buffer register.

(iii) Left shift operation.

(iv) Right shift operation during the 'Division' instruction.

Fig.20 shows how twelve Type D bistables are connected to form the shifting register A.

The register is reset by applying the negative going pulse F.13 simultaneously to the pre-clear inputs of each bistable. The equation defining the operation of pulse F.13 is given by:

$$\overline{F.13} = I_1.K_1.B_2.C_L + St.13$$

(See Fig.26.)

After ensuring that the register is reset to zero, the contents of the core buffer register may be transferred to appropriate stages of the register A by applying the positive going timing pulse F.14 simultaneously to one input of each of the NAND gates linking the core buffer register to the pre-set inputs. Those gates which have a logic 1 input, from the core buffer register, change their output from logic 1 to logic 0 when the pulse F.14 is applied. This change, applied to the pre-set input of a Type D bistable, causes the Q output to have the logic 1 value. The expression defining the timing pulse F.14 is given by:

$$F.14 = I_1.K_1.B_9.C_L + St.14.$$

(See Fig.24.)

The left shift operation is controlled by the timing pulse F.24, which is applied to the C input of each stage in the register. It will be noted that during the division operation the contents of the register must be moved one place right. This is achieved by gating the true output of stage 11 and the Q output of the DIVIDE bistable to the D input of stage 1. By applying ten shift left pulses to the C line the contents of the register are effectively moved one place right. The expression for timing pulse F.24 is:

$$F.24 = I_7\left[B_0(K_2+K_3)+(B_3+B_6+B_9)K_1\right]C_L + I_8\left[K_1(B_5+B_9)+(K_2+K_3)(B_1+B_5+B_9)\right]C_{L10} + St.24.$$

(See Fig.26.)

An additional input to the gate controlling the pulse F.24 is required to deal with the alignment of the operands at the start of the division operation. It is necessary to shift the contents of both registers A and ACC. left, until their most significant one digits are in position 11. These pulses are derived by using the gating circuitry shown in Fig.21. The difference in the number of pulses required to align the operands is gated to the '4 bit count', which during K_1B_1 time is arranged to count up from the reset state of 0001. During the division operation, timing pulse F.26 is applied to the counter to reduce the contents to zero. When the '4 bit count' shows zero, the DIVIDE bistable is reset, signalling the end of the division operation. See Fig.27.

6.2.4 Register M.

The twelve stage register M is required to store the multiplier during the instruction I_7 and the quotient during the instruction I_8. It must be capable of:

(i) Overall reset to zero.

(ii) Parallel entry of data from the core buffer register.

(iii) Right shift operation.

(iv) Serial entry into position 10.

The general arrangement of the twelve Type D bistables which are used in this register, is shown in Fig.21. From a consideration of the order code, it is evident that the register must be reset to zero before the multiplier is entered and before the first digit of the quotient is entered. The negative going timing pulse F.20 is applied to the pre--clear input of each stage to provide the overall reset facility. This pulse is defined by:

$$\overline{F.20} = I_5.K_1.B_2.C_L + I_8.K_1.B_2.C_L + St.20.$$

(See Fig.26.)

Parallel loading from the core buffer register is achieved using the twelve NAND gates which are connected to the corresponding pre-set inputs of the register. The positive going timing pulse F.21 is applied to one input of the gates. The outputs from these gates, which already have a logic 1 applied from the core buffer register, change from logic 1 to logic 0. This negative going edge at the pre-set input causes the Q output of the bistable to be in the logic 1 state

The timing pulse F.21 is given by:

$$F.21 = I_5.K_1.B_9.C_L + St.21.$$

(See Fig.24.)

The right shift operation is achieved by connecting the Q output of each stage, M_x, to the D input of the stage of immediate lower order, M_{x-1}. The shift pulse F.23 is applied to each of the C inputs in the register. The expression for this pulse is:

$$F.23 = I_7\left[(K_1+K_2)(B_3+B_6+B_9)+(K_2+K_3)B_0\right]C_L$$
$$+I_8.D\left[K_1(B_5+B_9)+(K_2+K_3)(B_1+B_5+B_9)\right]C_L$$
$$+I_8.K_3.B_{10}.C_{L10} + St.23.$$

(See Fig.26.)

The three NAND gates connected between the Q output of stage 11 and the D input of stage 10, Fig.21, are required to ensure that, during the division operation, the quotient digits, derived from the CARRY bistable, are entered serially into stage 10 of the register.

When the multiply instruction is being executed, the addition of the multiplicand to the sub-products is condition--al upon the state of the least significant stage M_1. The preposition I_7M_1 is derived using the two NAND gates, shown in Fig.21, and is used to generate some of the timing pulses associated with instruction I_7.

6.2.5 The Parallel Adder.

Three four bit full adders, Type 7483N, form the basic parallel adder. Fig.7 gives the logic circuit for each of the four adders incorporated in the 7483N package, and Fig.20 shows their general arrangement in relation to the other registers in the arithmetic section. Whenever the adder is used, the two operands are temporarily stored in the registers A and B, with the ACC. register and ready to accept the sum outputs. The circuitry linking the registers A and B to the adder is arranged so that either the Q or $\overline{Q}$ outputs of either register may be applied to the adder simultaneously. (See Fig.28.). The conditions associated with the inputs to each stage n are:

$$X_n = A_n.F16 + \overline{A_n}.F27$$

$$Y_n = B_n.F17 + \overline{B_n}.F52$$

These conditions are satisfied using the three two-input NAND gates to link the outputs of the registers A and B to the corresponding inputs of the adder. The equations defining the operation of the timing pulses F.16, F.17 and F.27 are:

$$F.16 = I_3.K_1.B_2 + I_7.M_1\left[(K_1+K_2)(B_2+B_5+B_8+B_{11}) + (K_3.B_2)\right] + St.16$$

$$F.17 = I_3.K_1.B_2. + I_7.M_1\left[(K_1+K_2)(B_2+B_5+B_8+B_{11}) + (K_3.B_2)\right] + I_8.D(B_3+B_7+B_{11}) + I_{10}.K_1.B_2 + St.17.$$

$$F.27 = I_{10}.K_1.B_2 + I_8.D(B_3+B_7+B_{11}) + St.27.$$

(See Fig.24.)

Parallel Adder with Control Gates

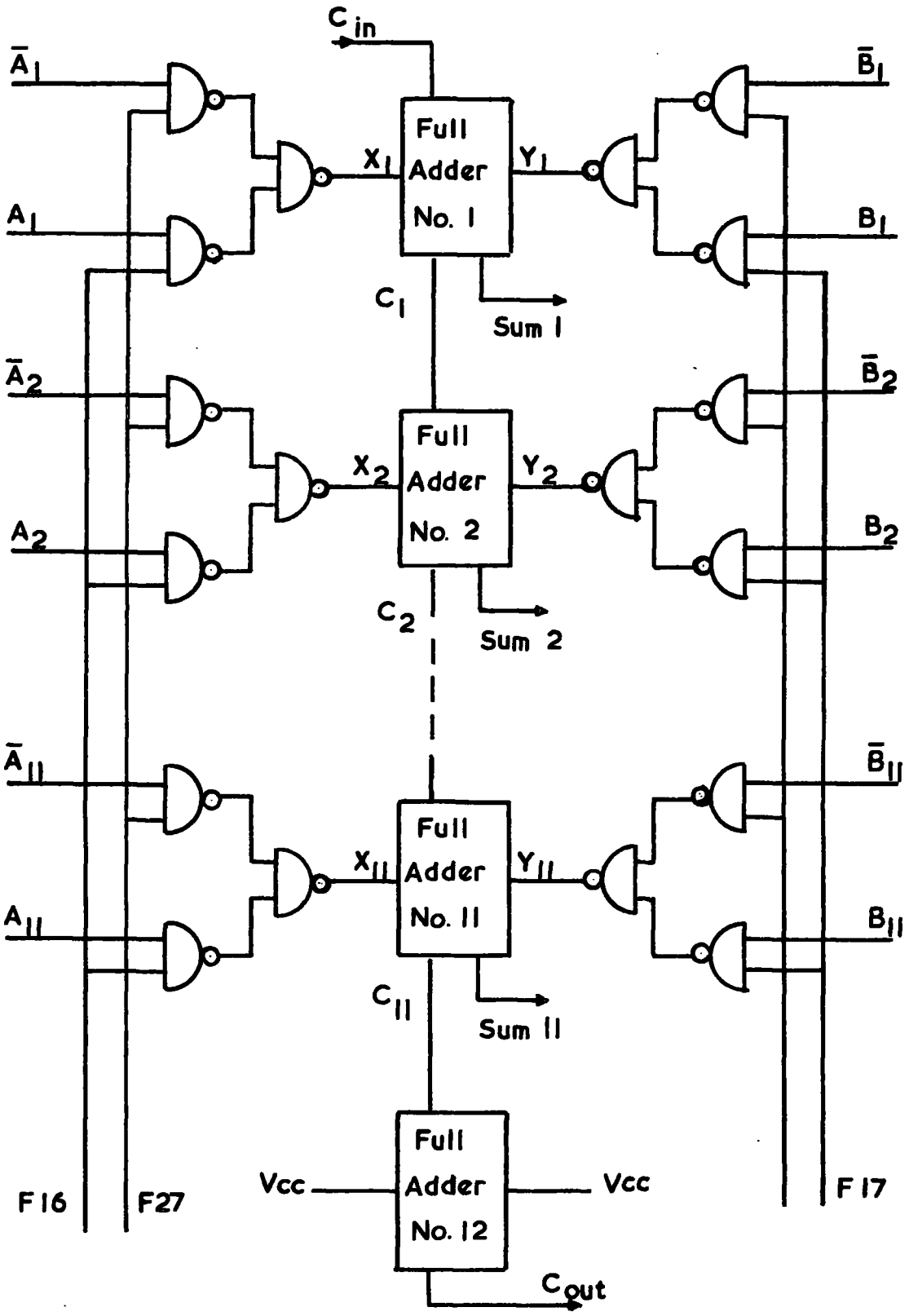


Fig. 28

TIMING PULSES FOR ADDITION

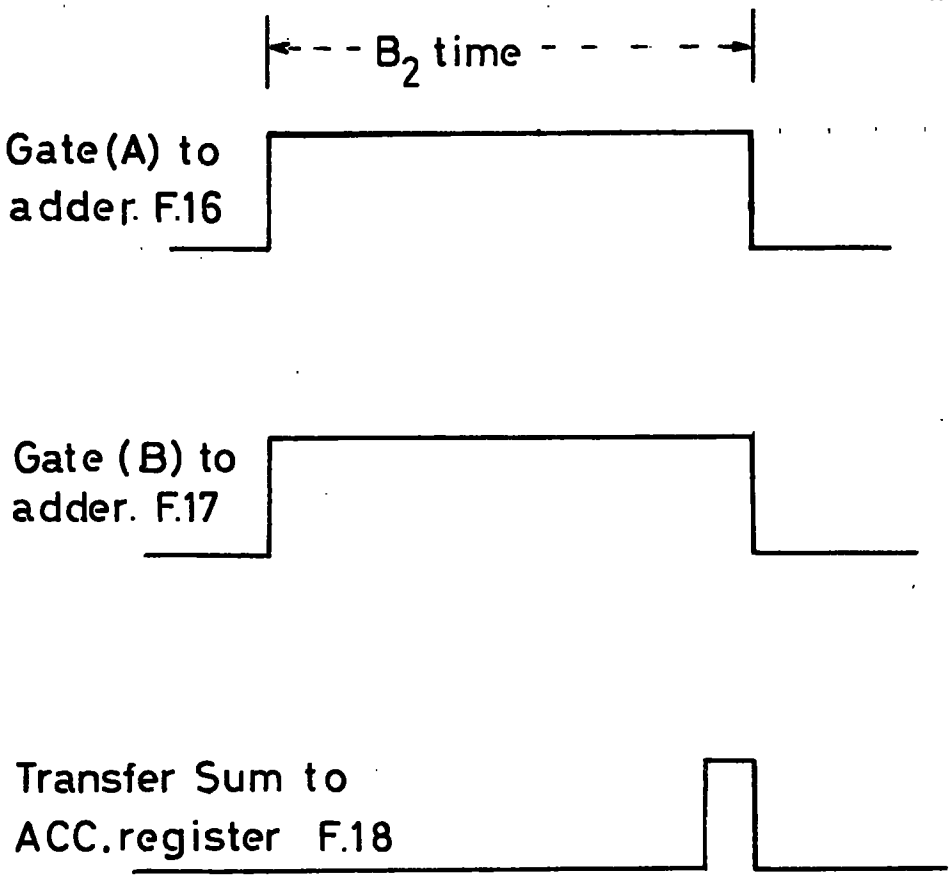


Fig. 29

It will be noted that the operands are applied to the adder for the whole of one bit time. The timing pulse F.18 occurs at the end of the bit time and is used to transfer the sum outputs of the adder to the ACC. register. (Sec. 6.2.1.). This delay allows sufficient time for the carry to propagate the adder twice. It is estimated that the total carry propagation time should not exceed 5μS. Fig.29 shows the relationship between the timing pulses F.16, F.17 and F.18.

It is necessary to define the carry input conditions to the first stage of the parallel adder, when it is used for subtraction and division. For the subtraction instruction I_{10}, an end-around-carry is required while for the division instruction I_8 the carry-in of the first stage is held at the logic one state since only positive remainders are used. This is an important point because if an end-around-carry were used in the trial subtractions of the division operation, the successful subtraction which gives a remainder of zero would have a zero carry-out. Thus a 0 would be entered into the quotient instead of a 1. The expression defining the carry input to the first stage is:

$$C_{in} = I_{10} \cdot C_{out} + I_8 \cdot D$$

(See Fig.27.)

Owing to the physical layout of the Type 7483N unit, it is not possible to obtain the carry signal from stage 11. The correct carry-out is however obtained from stage 12, by applying a logic 1 signal to the X and Y inputs of stage 12.

6.2.6 Miscellaneous Special Circuits.

The '4 bit count' referred to in Section 6.2.3 is a synchronous up-down counter employing four J-K bistable elements. The up-down control is achieved by arranging that either the Q output or $\overline{Q}$ output directs the clock pulse to the succeeding stage. The logic schematic of this circuit is shown in Fig.21. The counter is used only during the divide instruction I_8. During K_1B_0 time, the counter is reset to 0001. During K_1B_1 time, the counter is set to count up, the necessary clock pulses are derived from the operand alignment pulses. The true output of the DIVIDE bistable, $I_8.D$, is used to control the down count and the pulses required to achieve this are given by function F.26:

$$F.26 = I_8D\left[K_1(B_5+B_9) + (K_2+K_3)(B_1+B_5+B_9)\right]C_L$$

(See Fig.26.)

Determination of the sign digit for both the multiply and divide instructions is achieved using the gating circuitry shown in Fig.21. Four three-input NAND gates are required to accept the inputs from the sign digit (bit 12) of registers A, ACC. and M. These signals are gated with the signals for instructions I_7 and I_8 to give the correct determination of the sign digit. The expression for the sign digit of either product or quotient, which is applied to the D input of stage twelve of the ACC. register, is given by:

$$D_{ACC_{12}} = I_7\overline{M_{12}}A_{12} + I_7\overline{A_{12}}M_{12} + I_8A_{12}\overline{ACC_{12}} + I_8ACC_{12}\overline{A_{12}}$$

The timing pulse F.22 is used to enter the sign into stage 12 and is defined by:

$$F.22 = I_7.K_1.B_1.C_L + I_8.K_1.B_1.C_L + I_{10}.K_1.B_2.C_L$$

(See Figs.27, 26)

The CARRY bistable, Fig.27, is necessary in order to have temporary storage of the carry-out during the division instruction. The bistable is reset at the start of the division operation by the application of a negative going pulse, $I_8.K_1.B_2.C_L$. The timing pulse F.18, used to transfer the sum to the ACC. register, is also applied to the C input of the CARRY bistable. Thus the Q output of the bistable will reflect its D input state, which is directly connected to the carry-out of the parallel adder.

6.3 Detailed Design of the Control Unit.

In this section the detailed design of the parts of the control unit will be considered.

6.3.1 The Control Register.

The control register is used to store the machine instruction during execution time. The instruction is entered serially from the tape store during K_0 time and is given out in parallel when required.

The twelve Type D bistables are connected to form a shift right register, the logic diagram being shown in Fig.22. The shift control line is separated, stages 1 to 5 being

controlled from timing pulse F.3, and stages 6 to 12 from timing pulse F.3X. This separation is necessary because the sum output of the single-bit adder may be required to be entered serially into stages 6 to 11 during K_1 time for those instructions associated with operand address modification. The three two-input NAND gates associated with the D input of stage 6 are used to link either the output from stage 5, during K_0 time, or the output from the adder, during K_1 time. The shift pulses F.3 and F.3X are defined by:

$$F.3 = K_0.C_{Lp} + St.3$$

$$F.3X = K_0.C_{Lp} + CR5.K_1(I_0+I_1+I_4+I_5)(B_1+B_2+B_3+B_4+B_5+B_6)C_{Lp} + St.3$$

(See Fig.25.)

6.3.2 The Modifier Register.

The six-stage modifier register must be capable of:

(i) Parallel entry from stages 6 to 11 of the control register.

(ii) Overall reset to zero.

(iii) Binary counting.

In order to meet the parallel entry and counting requirements, Type J-K bistables are employed. The characteristic equation for an n stage reversible counter is

$$Qn_{t-1} = \bar{Q}n.\bar{Q}_{n-1}.....\bar{Q}_1.C_L$$

It was decided to employ the ripple through technique in order to minimise gating requirements. The logic schematic

Fig.22, shows the general arrangement, with the output of each stage connected to the C input of the stage of immediate higher order. The count pulses are applied to the C input of stage one using the timing pulse F.12. The equation for this timing pulse is:

$$F.12 = C_L(Mod_1 + Mod_2 + Mod_3 + Mod_4 + Mod_5 + Mod_6)(I_0 + I_1 + I_4 + I_5) \cdot CR5.K_1.B_{10}$$

The circuit used to obtain the first part of the expression involves the use of the six-input NAND gate, Fig.22, the inputs to which are connected to the $\bar{Q}$ output of each stage in the counter. The gating circuitry associated with the remainder of the equation is shown in Fig.25.

The register is reset by applying the negative going timing pulse F.31 to all pre-clear inputs simultaneously. This pulse is defined by:

$$F.31 = I_{15}.K_1.B_1.C_L.$$

(See Fig.26.)

With the register reset, the data stored in the control register (bits 6 to 11) is transferred to the modifier register by the timing pulse F.32. This pulse is applied to one input of each of the six two-input NAND gates which link the Q outputs of the control register to the corresponding pre-set inputs of the modifier register. Timing pulse F.32 is given by:

$$F.32 = I_{15}.K_1.B_2.C_L + St.32$$

(See Fig.26)

The modification of the operand address is carried out during K_1 time of those instructions dealing with transfer of operands between the Memory and Arithmetic sections. The state of stage 5 of the control register determines whether or not address modification is required. The operation is executed serially by applying the contents of the control register (bits 6 to 11) and the contents of the modifier register to the single stage modifier adder. The sum output of this adder is returned to the control register via stage 6. The six two-input NAND gates, Fig.22, and the timing pulses F.40 ..... F.45 are used to transfer the Q outputs of the modifier register to modifier adder. The timing pulses F.40 ..... F.45 are defined by:

$F.40 = K_1(I_0+I_1+I_4+I_5)B_1$ $\quad$ $F.41 = K_1(I_0+I_1+I_4+I_5)B_2$

$F.42 = K_1(I_0+I_1+I_4+I_5)B_3$ $\quad$ $F.43 = K_1(I_0+I_1+I_4+I_5)B_4$

$F.44 = K_1(I_0+I_1+I_4+I_5)B_5$ $\quad$ $F.45 = K_1(I_0+I_1+I_4+I_5)B_6$

The timing pulse F.3X is used to shift the contents of the control register right. This ensures that the correct order of thevcontrol register is applied to the modifier adder and that the sum output of this adder is serially entered into stage 6 of the control register. As serial addition is employed, the carry storage bistable C_m is required. The bistable C_m is pre-set by the negative going edge of the pulse $\overline{B_0}$ before the addition process commences. The carry--out from the single stage modifier adder is entered into the

carry bistable C_m by the timing pulse F.3X.

6.3.3 The Programme Counter.

The programme, or instruction, counter section of the control unit is similar in construction to the modifier register. Six J-K bistable elements are arranged as a ripple through binary counter. The count up pulses, F.9, are applied to the C input of stage one at the end of every instruction. If, however, a conditional jump instruction is specified and satisfied, the count up pulse is not applied and under these conditions the counter is reset and loaded with the address of the next instruction which is stored in the control register (bits 6 to 11). The circuitry associated with this section of the control unit is shown in Fig.22, and it is similar to that of the modifier register.

The counter is reset by the negative going timing pulse F.28, which is given by:

$$\overline{F.28} = K_1 B_1 \left[I_{11} (Mod_1 + Mod_2 + Mod_3 + Mod_4 + Mod_5 + Mod_6) + I_{13} . \overline{ACC_{12}} \right] C_L + St.28$$

(See Fig.25.)

The positive going timing pulse F.29 is used to transfer the contents of the control register (bits 6 to 11) to the programme counter. The expression for this pulse is:

$$F.29 = K_1 B_2 \left[I_{11} (Mod_1 + Mod_2 + Mod_3 + Mod_4 + Mod_5 + Mod_6) + I_{13} \overline{ACC_{12}} \right] C_L + St.29.$$

(See Fig.25.)

The timing pulse F.9 is given by:

$$F.9 = K_1 B_{10}\left[I_{11}\overline{(Mod_1+Mod_2+Mod_3+Mod_4+Mod_5+Mod_6)} + \overline{I_{13}\overline{ACC_{12}}} + \overline{I_7}+\overline{I_8}\right]C_{Lp} + K_3 B_{10} C_{Lp}(I_7+I_8) + St.9$$

(See Fig.25.)

6.3.4 Miscellaneous Control Circuits.

On each occasion when the programme calls for a transfer of operands between the Memory and Arithmetic units, it is necessary to gate the six-bit operand address to the memory, so that the appropriate sense wires may be energised. The code of the operand address, held in the control register, is gated to the memory during K_1 time by the application of the function F.37 to the commoned line of the six two-input NAND gates, Fig.22. The timing pulse F.37 is defined by:

$$F.37 = K_1(I_0+I_1+I_4+I_5) + St.37$$

(See Fig.26.)

Although the operand address is available in the memory for the whole of K_1 time, it is not used until the read and write pulses are applied during B_8 and B_{10} time respectively. The expression defining the read pulse is:

$$F.6 = (I_0+I_1+I_5).K_1.B_8.C_L + St.8$$

The expression defining the timing pulse F.8, the write pulse, is:

$$F.8 = (I_0+I_1+I_2+I_5)K_1.B_{10}.C_L$$

(See Fig.26.)

The instruction I_9 specifies the left shift operation of the ACC. register. During the execution of this inst--ruction, stages 8 to 11 of the control register specify, in code, the number of left shift pulses to be applied to the ACC. register. The shift left counter, shown in Fig.22, employs four J-K bistable elements arranged as a binary counter. The counter is reset by the negative going timing pulse F.33, which is simultaneously applied to the pre-clear inputs of each stage of the counter. Timing pulse F.33 is given by:

$F.33 = I_9.K_1.B_0.C_L$ (See Fig.26.)

The contents of the control register (bits 8 to 11) are entered into the shift left counter via the four, parallel transfer, two-input NAND gates, each of which is controlled by the timing pulse F.34. The expression for this pulse is:

$F.34 = I_9.K_1.B_1.C_L + St.34.$ (See Fig.26.)

A maximum of ten pulses, F.35, may be applied to shift the contents of the ACC. register left. The number permitted to pass through to this register is controlled by the four--input NAND gate associated with the Q outputs of the shift left counter. The shift pulses, F.35, are applied to the counter until the contents is 1111, when this occurs the output of the four-input NAND gate falls to zero thereby blocking further pulses to both the counter and the ACC.

register. The timing pulse F.35 is given by:

$F.35 = I_9 . K_1 . B_3 . C_{L10}$

(See Fig.27.)

6.4 Detailed Design of the Timebase and Instruction Decode Sections.

This Section deals with the design of the circuits for the basic functions I,K,B and C_L which are used to generate all the timing pulses.

6.4.1 The Instruction Decode.

The sixteen machine instructions listed in Section 5.5 are decoded from the four most significant stages of the control register. In order to minimise gating circuitry, the method shown in Fig.23 was used in preference to either a rectangular or pyramidal decoding network. This method involves decoding 'by pairs'; e.g. stages CR_1 and CR_2 are decoded to produce four intermediate outputs,

$$\overline{CR_1 . CR_2} \ , \quad \overline{\overline{CR_1} . CR_2} \ , \quad \overline{CR_1 . \overline{CR_2}} \quad \text{and} \quad \overline{\overline{CR_1} . \overline{CR_2}} \ .$$

Similarly stages CR_3 and CR_4 are decoded to produce intermediate outputs,

$$\overline{CR_3 . CR_4} \ , \quad \overline{\overline{CR_3} . CR_4} \ , \quad \overline{CR_3 . \overline{CR_4}} \quad \text{and} \quad \overline{\overline{CR_3} . \overline{CR_4}} \ .$$

The intermediate outputs are applied to the inputs of sixteen two-input NOR gates to complete the decoding. There are two advantages with this method, which are:

(i) The total number of gate inputs is reduced from 64 to 48.

(ii) The loading on stages CR_1 --- CR_4 of the control register is reduced from 16 to 2.

The expressions defining the sixteen machine instruct--ions are listed below. For convenience the stages CR_4, CR_3, CR_2 and CR_1 are referred to as A,B,C and D respectively. The equation for I_0 is given in full to show how the intermediate and final decoded outputs are derived using the combination of two-input NAND and two-input NOR elements. The other expressions are derived similarly.

$$I_0 = \overline{\overline{\overline{C}.\overline{D}} \; \overline{\overline{A}.\overline{B}}} = \overline{A}.\overline{B}.\overline{C}.\overline{D}$$

similarly:

$I_1 = \overline{A}.\overline{B}.\overline{C}.D$; $I_2 = \overline{A}.\overline{B}.C.\overline{D}$; $I_3 = \overline{A}.\overline{B}.C.D$;

$I_4 = \overline{A}.B.\overline{C}.\overline{D}$; $I_5 = \overline{A}.B.\overline{C}.D$; $I_6 = \overline{A}.B.C.\overline{D}$;

$I_7 = \overline{A}.B.C.D$ $I_8 = A.\overline{B}.\overline{C}.\overline{D}$ $I_9 = A.\overline{B}.\overline{C}.D$;

$I_{10} = A.\overline{B}.C.\overline{D}$; $I_{11} = A.\overline{B}.C.D$; $I_{12} = A.B.\overline{C}.\overline{D}$;

$I_{13} = A.B.\overline{C}.D$; $I_{14} = A.B.C.\overline{D}$; $I_{15} = A.B.C.D$.

6.4.2 The Timebase.

The requirements of the timebase, specified in Section 5.3, may be met by either of two methods. The squre-wave signal derived from the tape store or from an internal oscillator, may be divided using binary counters, with the outputs of the counters decoded to give the required sequential bit (B) and character (K) times. The second method involves the continued circulation of a 'unique 1 stage' using a

closed-loop shift register or ring counter. In order to minimise circuit components, it was decided to employ a combination of these two methods.

The fundamental square-wave signal, C_{L10}, is derived either from the tape storage unit or from a 10Hz astable multivibrator. The logic schematic, Fig.23, illustrates the method used to generate the four character times K_0, -- K_3, the twelve bit times B_0, --- B_{11}, the master clock signal C_L and the pre-clock signal C_{Lp}. The signal C_{L10} is divided by ten using a B.C.D Decade Counter Type 7490N, which is connected as a 'divide-by-five' followed by a 'divide-by-two' circuit. This is done to maintain an equal mark : space ratio at the output of this unit. The signals 'divide-by-five', 'divide-by-two', and C_{L10} are applied to a three-input NAND gate, the output of which is inverted to produce the master clock signal C_L. A similar gating arrangement is employed to generate the pre-clock pulse C_{Lp}.

The bit time (B) generator consists of twelve Type D bistable elements connected to form a closed-loop shift right register. The single 'one' digit is arranged to be stored in position 11 during standby conditions, this is done by connecting the pre-set input of this stage to the common reset line. On the application of the inverted 'divide-by-ten' signal to the shift control line, the contents of the register are circulated towards the right. Thus the single digit

initially stored in position 11 moves sequentially through the register. The Q and $\overline{Q}$ outputs from each stage are made available for use in the timing function generators.

The character time (K) generator operates on a similar principle, using four Type D elements, arranged as a closed--loop shift right register. A single 'one' digit is stored in position 3 during standby conditions, this is done by connecting the pre-set input of this stage to the common reset line. The shift pulses for this register are obtained from stage B_0 of the bit time generator. Thus the single 'one' digit is circulated right one position on the applicat--ion of the positive going edge of the B_0 pulse. The timing pulse, F.10, is used to reset both K and B generators to the standby condition, K_3B_{11}, at the end of each instruction. This pulse is defined by:

$$\overline{F.10} = K_1 . B_{11} . C_{Ip}(\bar{I}_7+\bar{I}_8) + K_3 . B_{11} . C_{Ip}(I_7+I_8) + St.10$$

(See Fig.25.)

Three bistable elements, entitled START, EXECUTE and MANUAL, Fig.27, are used to control the application of the fundamental square-wave signal to the timebase. With the mode selector switch set to 'FAST' the START bistable is set by the manual operation of the 'start' push-button. On receipt of the alignment signal, which indicates that the next instruction to be executed is in line with the tape head, the EXECUTE bistable is set. This allows the square

wave signal, C_{L10}, to be gated through to the timebase section. On completion of an instruction the EXECUTE bistable is reset by the timing pulse F.11. The gating requirements for the pulse F.11 were found to be the same as those used to establish the timing pulse F.10, the pulse F10 is therefore used to reset the EXECUTE bistable. The timebase then rests in the standby condition to await the next alignment signal.

With the mode selector switch set to 'SLOW', the 10Hz signal is gated through to the timebase when both the START and EXECUTE bistables are set. The START bistable is set at the beginning of a demonstration by manual operation of the 'start' push-button. Manual operation of the 'execute' push-button causes the EXECUTE bistable to be set and allows the 10Hz signal, C_{L10}, through to the timebase. On completion of an instruction the EXECUTE bistable is reset in the normal way. Under these conditions the machine completes one instruction for each operation of the 'execute' push-button.

With the mode selector switch set to MANUAL the 10Hz signal is gated with the Q output of the MANUAL bistable. A 'divide-by-ten' circuit is incorporated to reset the MANUAL bistable after ten pulses. Under these conditions each operation of the 'manual' push-button causes the timebase to advance by one bit (B) time.

The START bistable is reset, either by operation of the

'stop' push-button or by the timing pulse $I_{14}.K_1.B_{11}.C_L$ which occurs at the end of the 'stop' instruction.

6.5 The Timing Function Generators.

The Boolean expressions defining each of the timing functions have been given in the Sections 6.2, 6.3 and 6.4 dealing with detailed circuit design. The procedure adopted for determining the logic circuitry required to generate these functions is discussed in this Section. Appendix 1 gives full details of load, polarity and location for each timing function.

The basic equations, derived from a consideration of the order code and timing charts, were re-arranged and/or simplified using the methods outlined in Section 3.2. In deciding the actual logic circuitry to be used, the follow--ing points were borne in mind:

(i) The master clock pulse, C_L, or the pre-clock pulse, C_{Lp}, should be introduced in the last stage of any chain of gates. This is desirable in order to preserve synchronism and reduce the effects of propagation delay.

(ii) The circuitry should be arranged so as to avoid having partially used logic blocks.

(iii) The equivalent of one logic load should be allowed for driving the display circuitry. The fan out is therefore reduced to 9 in the case of the Series 74N and to 7 for the Series 10 integrated circuits.

(iv) The design should conveniently permit future

modification either to form new instructions or to modify existing instructions in the event of unforseen sequencing problems.

(v) The circuits should be arranged to accept inputs from the 'Student push-button panel' in order to meet the requirements of Section 4.9.

The gating requirements for each function were deduced and tabulated. Common factors were noted and the final circuitry drawn up by a compromise between the limitations imposed by items (i), (ii) and (iv) above and the capacity of the integrated circuit carrier board that was specially designed for the timing function circuits.

6.6 The Display and Manual Control Circuits.

As stated in Section 4.9, an important feature of the machine would be that a large number of filament lamps should be incorporated on the front panel to monitor the various timing functions and the contents of storage registers. It was decided to use 6v-40mA lamps on the mimic diagram and 28v-80mA lamps in the illuminated switch units. The lamp drive circuit, Fig.30, causes the lamp to be fully ON when a logic 1 voltage is applied between points A and B. The majority of timing functions and registers are monitored using this type of circuit. When negative going timing pulses are being monitored the two transistor circuit, Fig.31, is used and, in this case, a logic 0 voltage applied between

LOGIC '1' LAMP DRIVE CIRCUIT

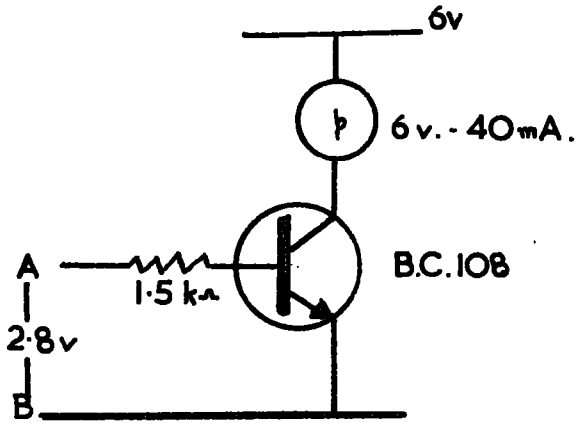


Fig 30

LOGIC '0' LAMP DRIVE CIRCUIT

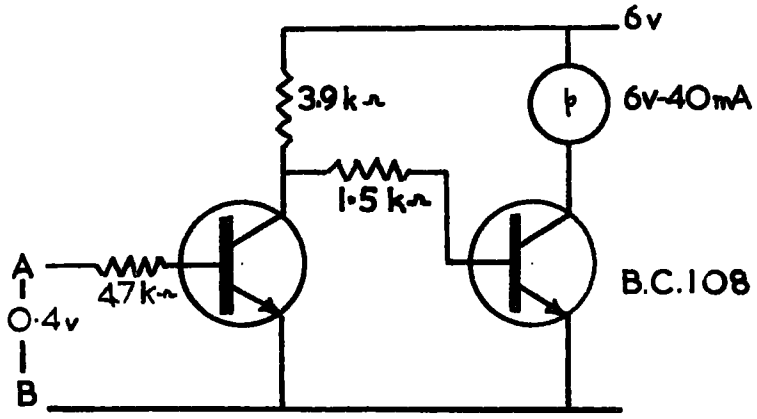


Fig. 31

LOGIC '1' LAMP DRIVE CIRCUIT (28v)

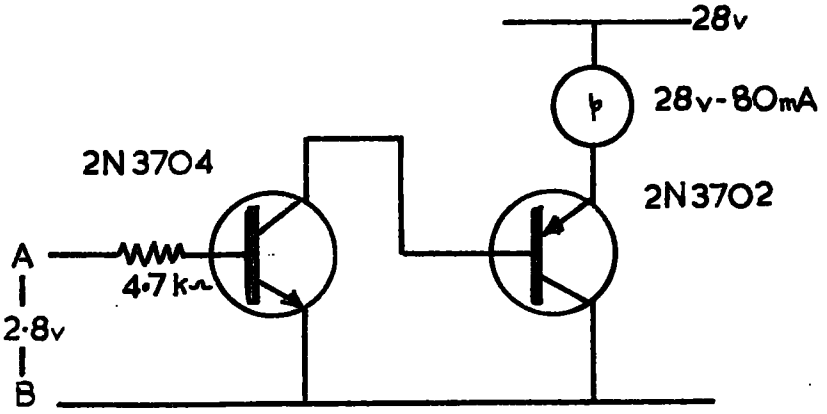


Fig 32

points A and B causes the lamp to be ON. Both circuits are designed to present not more than one logic load to the output of a 74N Series integrated circuit. The worst case conditions, listed in Section 6.1.2, were observed in deducing these circuits. The drive circuit for the 28v-80mA lamps, Fig.32, is used in preference to a Darlington pair arrangement so as to minimise junction potentials existing in series with the input circuit. A logic 1 voltage level applied to the input causes both transistors to be turned on, allowing the lamp circuit to carry the nominal 80mA.

The circuits shown in Figs.33 and 34 are used to derive the logic levels associated with the student operating switches, the main control switches and the data input switches. In those cases, where manual control is associated with shifting or counting, it is necessary to buffer the input by means of a bistable element (Fig.34) in order to eliminate the effects of contact bounce. These circuits have been de--signed in accordance with the input characteristics of the 74N Series integrated circuits.

6.7 The Power Supplies.

A 5v $\pm$ .25v stabilised supply is required to operate the integrated circuits and also to derive the logic states associated with the manual control switches. The total power requirements for this supply were estimated to be:

9.0 watts for the 180 integrated circuits, plus

STUDENT SWITCH CIRCUIT (TYPE A)

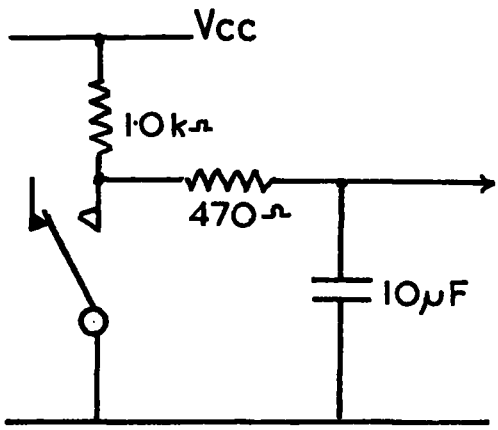


Fig. 33

STUDENT SWITCH CIRCUIT (TYPE B)

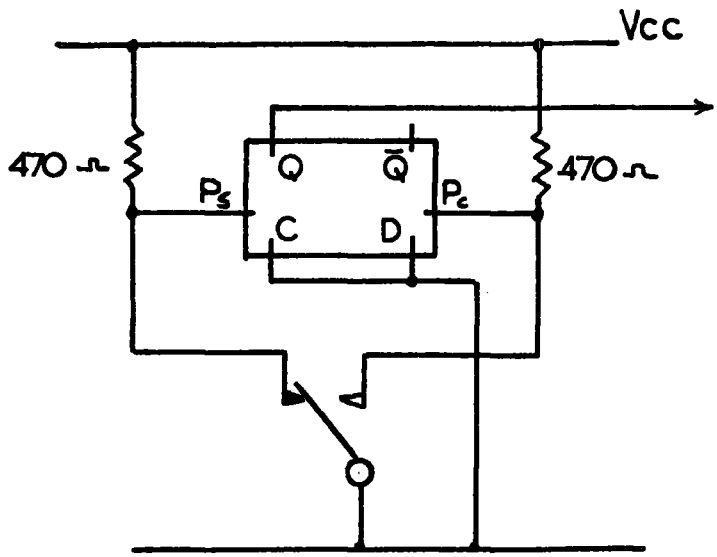


Fig. 34

3.0 watts for the 60 manual control circuits. It was decided to use a 5v, 5A commercially available stabilised power unit. (Farnell Instruments Type V.3.)

A 6v unstabilised supply is needed to operate the 128 display lamps. It was assumed that, under worst case conditions, only 100 lamps would be ON, giving a current drain of approximately 4A. This supply is derived using a conventional bridge rectifier arrangement.

The 28v unstabilised supply associated with the 5 illuminated control switches is required to deliver a total current of 800mA.

The various power requirements for the memory section were also estimated, and the totals combined so that a single power pack could be incorporated in the main equipment cabinet. The ratings of each section were suitably modified to off-set the effect of any reasonable extension to the equipment.

CHAPTER VII.

MAIN ASSEMBLY.

It was decided that the main equipment cabinet should incorporate four 16" rack units. Two of these are required to accommodate the memory section, the third to incorporate the arithmetic and control sections and the fourth to carry the various power supply units. The sketch, Fig.35, shows the general arrangement of the four racks, assembled so that all four may be withdrawn from the cabinet in order to permit detailed examination of the logic circuitry. The cabinet, Fig.36, includes a three section front panel which is intended to be identified with the Memory, Control and Arithmetic sections of the machine. Details of the relevant mimic diagrams are given in Section 7.3.

7.1 Printed Circuit Boards.

The use of double-sided printed circuits was seen to be the most practical method of interconnecting the dual-in-line logic blocks. After making some preliminary test circuit boards, the following design criteria were established in order to reduce the risk of malfunction:

(i) To employ 0.065 in. glass epoxy laminate (See Ref. 18).

(ii) The minimum conductor width to be 0.02 in.

(iii) The termination pads to be 0.062 in. diameter.

(iv) The through-board connections to have 0.125 in. diameter pads on one side.

FOUR 16" I.S.E.P. RACK ASSEMBLY

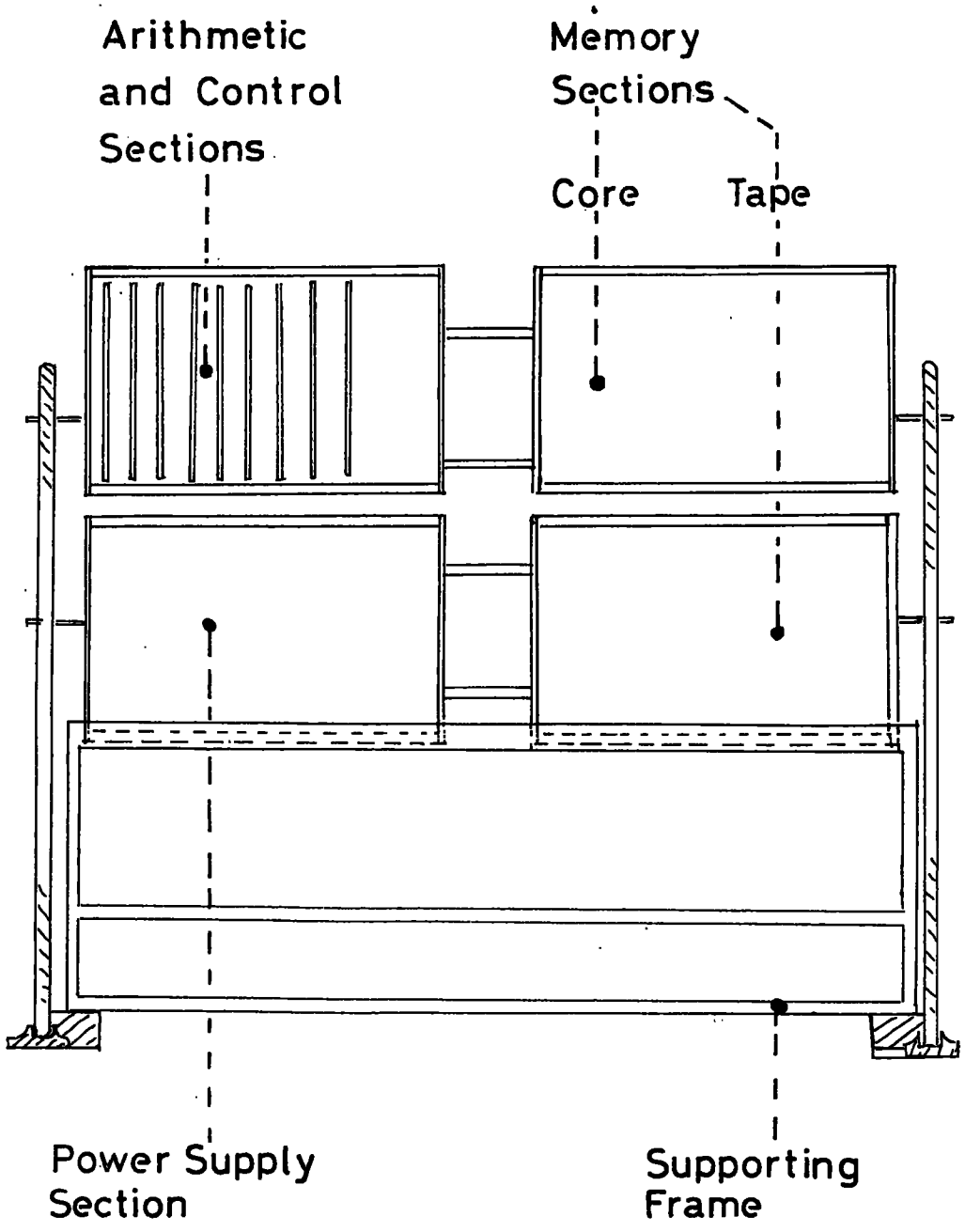


Fig. 35

MAIN EQUIPMENT CABINET

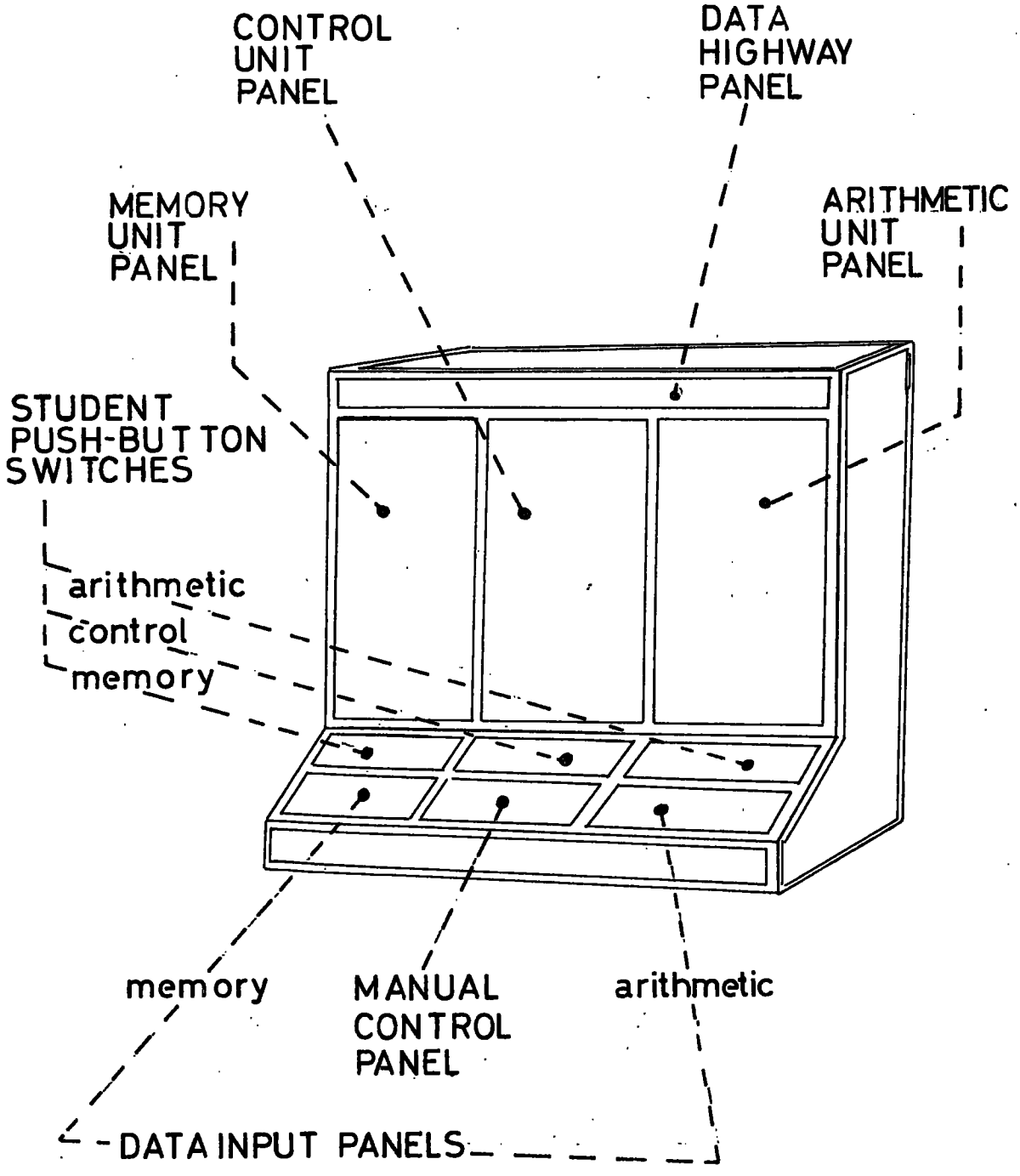


Fig. 36

(v) The overall size of printed circuit board to be approximately 9" x 8".

(vi) The packing density to be approximately 30 dual-in--line integrated circuits per board.

(vii) The edge terminations to be via two single-sided 33 way I.S.E.P. connectors on each board.

In deciding on the number of printed circuit boards to employ, two conflicting factors required careful consideration. The cost:size ratio of the integrated circuit blocks could result in expensive replacements, if each board carried too many units. If only a few units were incorporated on each board, then the number of edge connections would be extremely high. In anticipation of the expected price reductions of integrated circuits and to demonstrate their chief advantage, it was decided to aim at between 20 and 30 circuits per board. An influencing factor was that the circuit requirements seemed to break down into groups of about this size admirably.

The Arithmetic sections are accommodated on three boards AU-001, AU-002 and AU-003. The Timebase and Instruction Decode circuitry is incorporated on a fourth board TB-001. A fifth board, CU-001, accommodates the logic elements associated with the Control section. The circuitry associated with the timing functions is built up using three identical single-sided dual-in-line carrier boards with a small amount of cross-board wiring. Each of these single-sided boards can

accept up to 20 dual-in-line circuit blocks and the inter--connections between gates are made using 'jumper' wires fitted to the non-copper side of the board. The ninth board in the machine carries a small number of special circuits which could not conveniently be included with any other section.

The art-work required for the five double-sided and four identical single-sided boards was produced four times full size using conventional materials. The nine boards were produced in accordance with the procedure developed by senior members of the Department of Applied Physics. After the boards had been drilled and sprayed with a protective flux, they were loaded with the appropriate integrated circuits and each fitted with the two 33 way I.S.E.P. connectors. Examples of the art-work and finished module are shown in Fig.37 and Fig.38 respectively.

7.2 Wiring Schedule.

The layout of the boards in the 16" rack unit is shown in Fig.39. The 264 way tag panel is used as an intermediate termination point between the 18 female I.S.E.P. connectors and the display panel. The detailed wiring schedule given in Appendix 2 specifies the function, location and destination of some 800 interconnections.

7.3 Panel Layout.

It was generally agreed that the effectiveness of the

PRINTED CIRCUIT LAYOUT

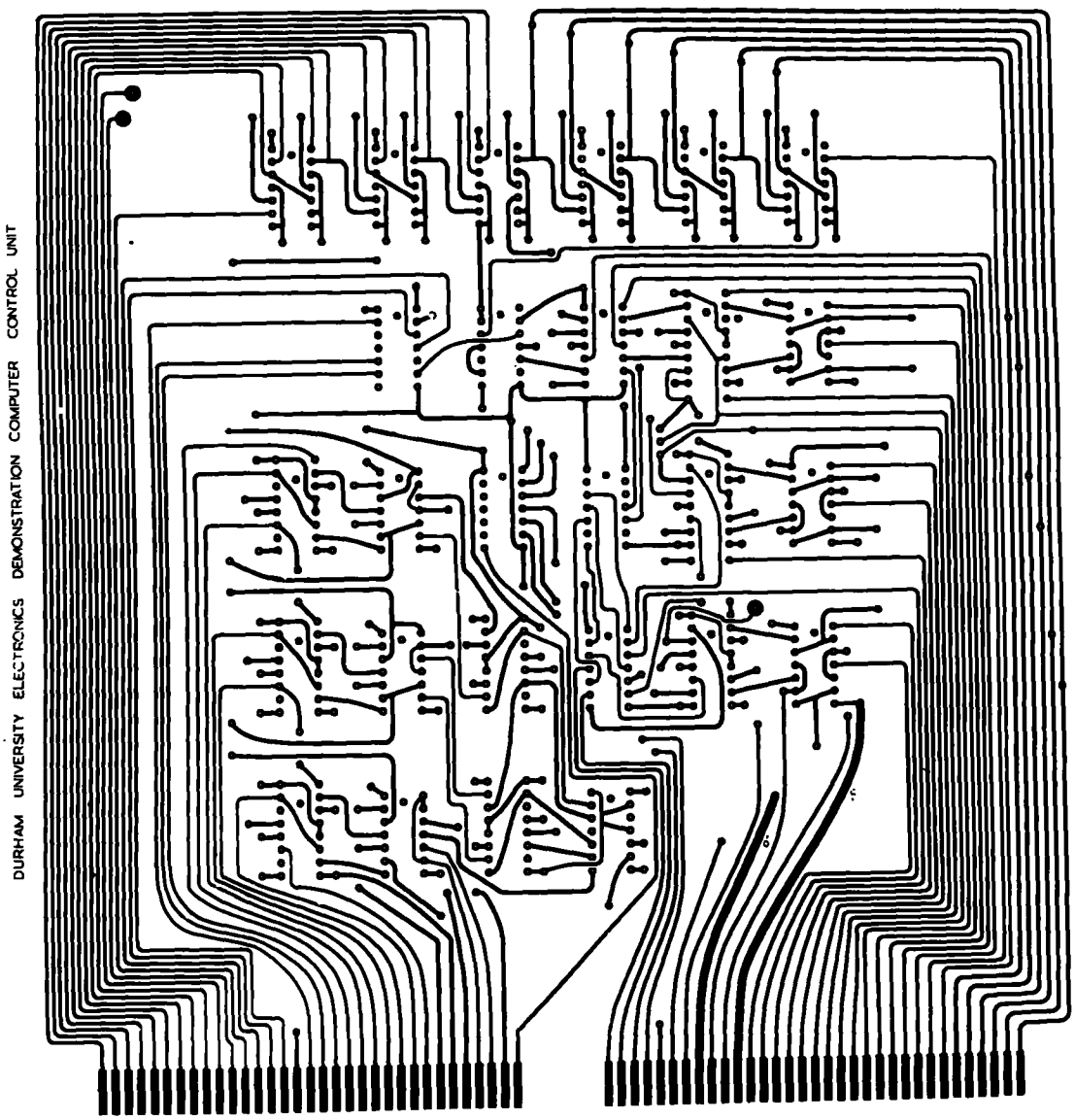
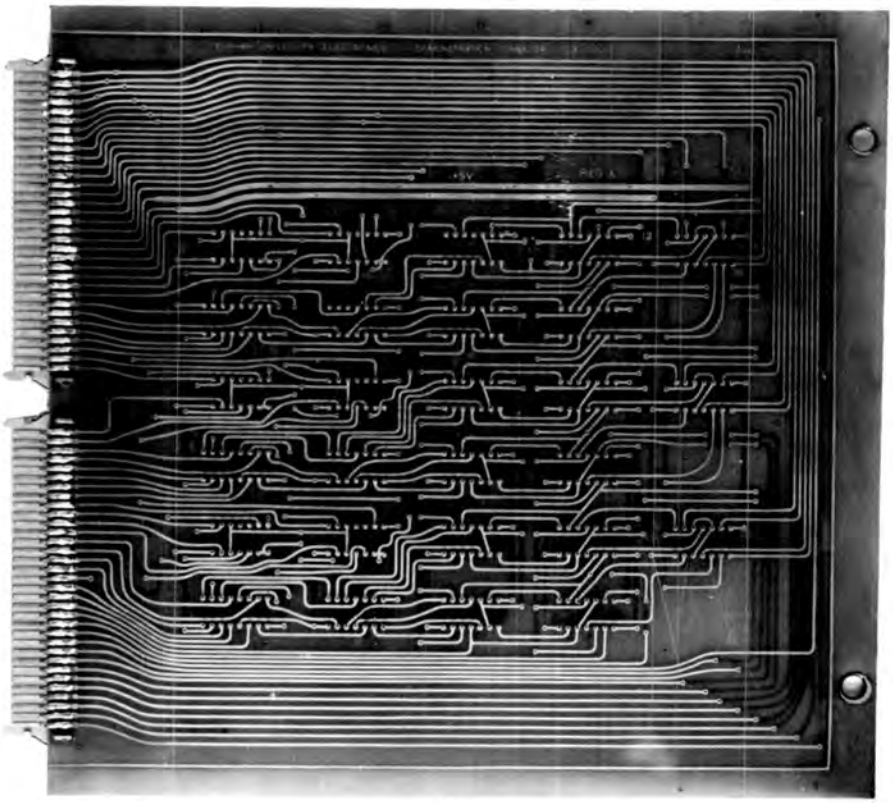


Fig. 37

PRINTED CIRCUIT MODULE



9"

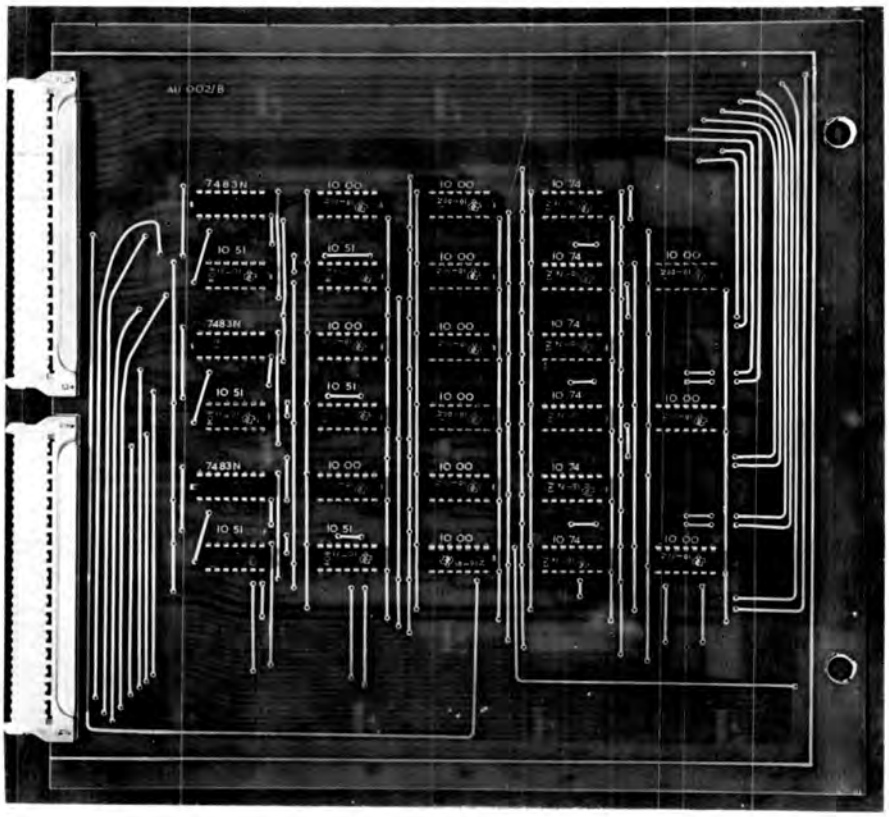


Fig .38

16" I.S.E.P RACK ASSEMBLY

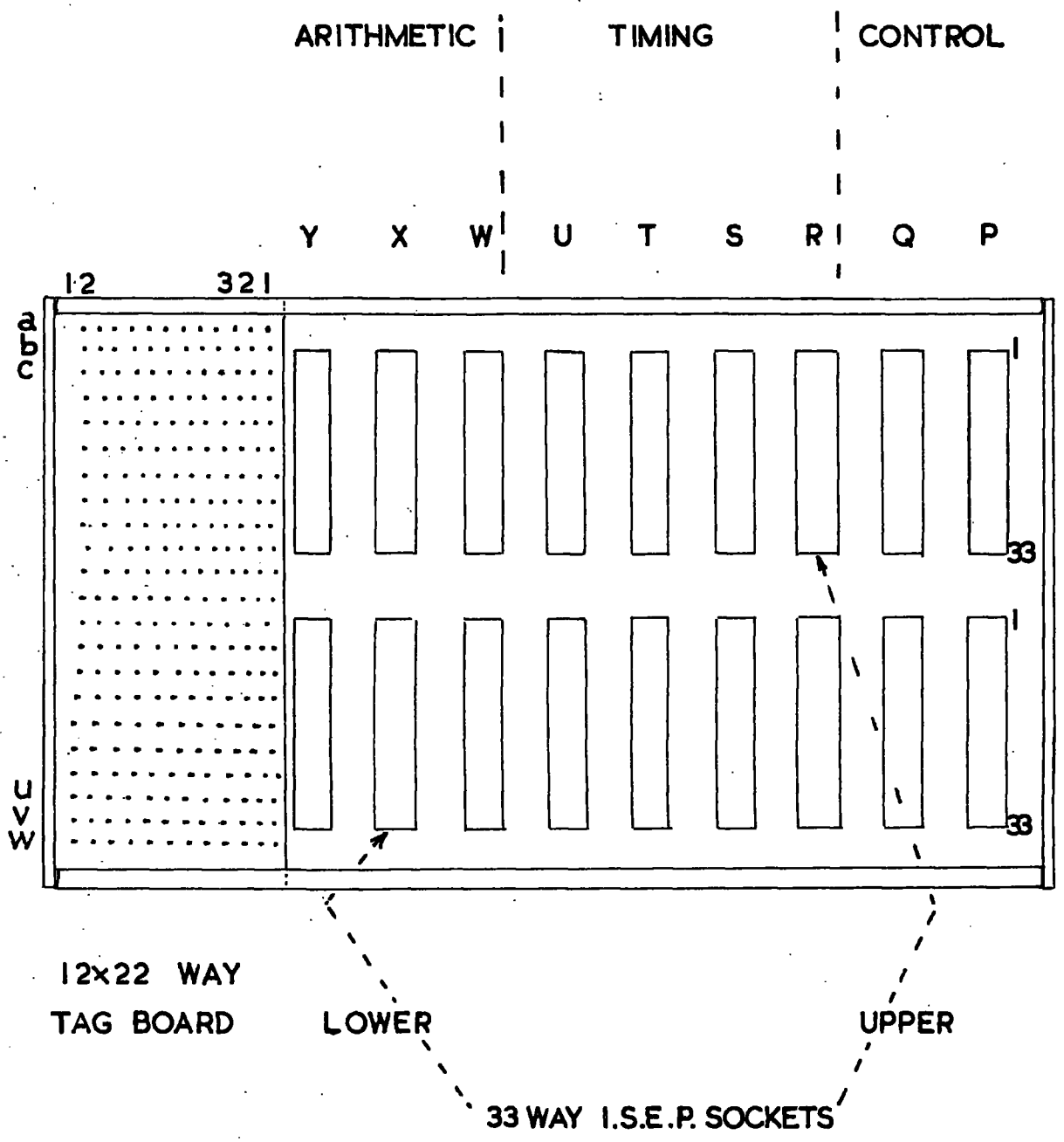


Fig. 39

of the equipment would be directly related to the design and layout of the front panels. It was realised that a compromise was necessary in respect of the amount of detail to be included; too much would lead to confusion and too little would lead to over-simplification. It was decided, in accordance with the specification of Section 5.1, that the front panel should be constructed so that the Memory, Arithmetic and Control sections would be separately identified with a 'Data Highway' incorporated to illustrate the link between the three sections. It was felt that the student operating switches and data input controls should be mounted on separate sub-assemblies, in order to ensure maximum clarity of the front panel. After considerable deliberation and a number of permutations, acceptable layouts were evolved.

The display panel, Fig.40, used to illustrate the operation of the Arithmetic section, incorporates four groups of twelve filament lamps which monitor the contents of the registers A, B, ACC. and M. The timing functions associated with this section are identified and monitored by the lamps 1 to 15. The graphical symbols and 'data paths' are intended to assist in the understanding of the arithmetic processes. The student switch panel allows manual operation of the fifteen timing functions and, when used in conjunction with the data input unit, enables the 'Standstill' operation

DISPLAY PANEL (ARITHMETIC SECTION)

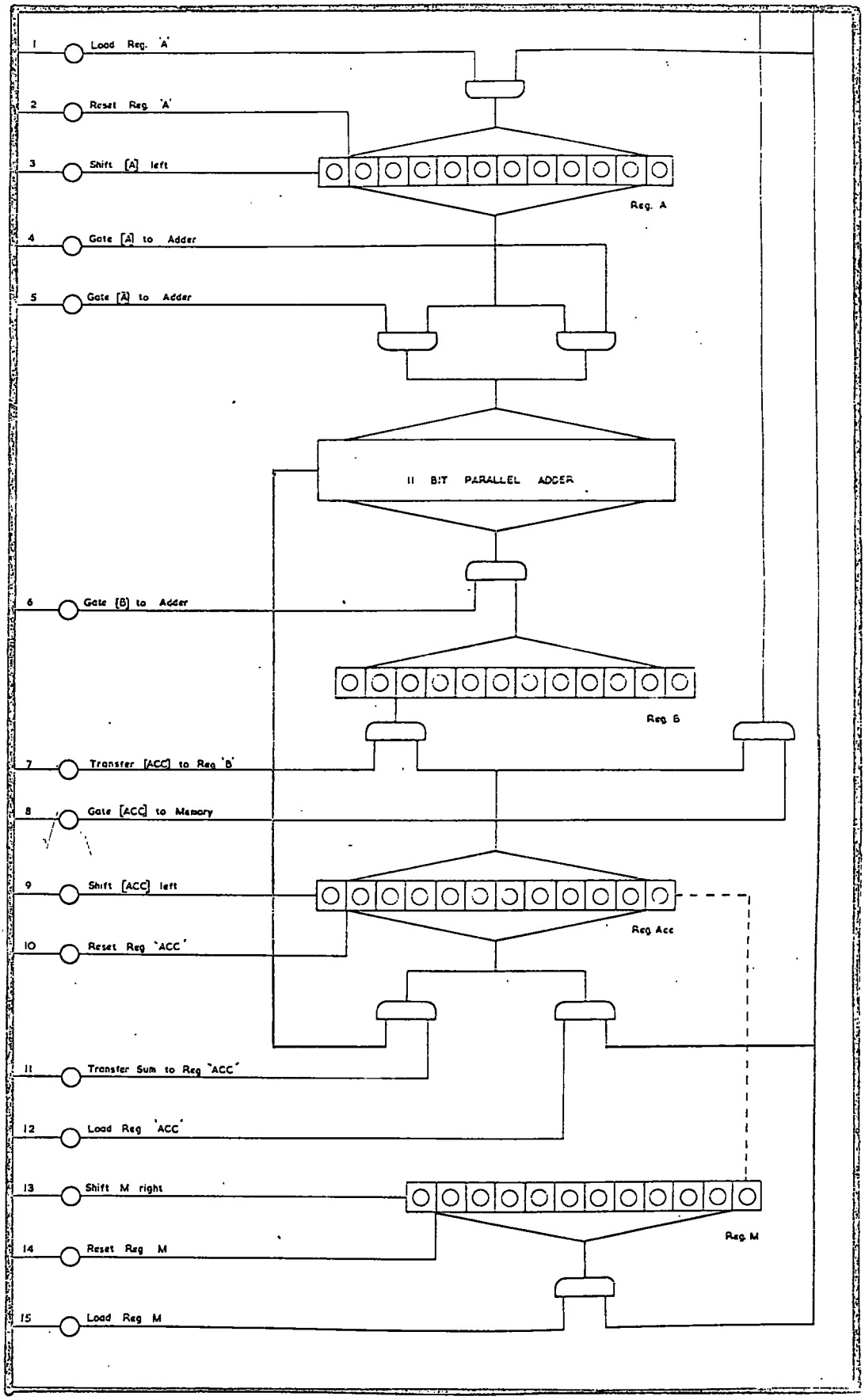


Fig 40

DISPLAY PANEL (CONTROL SECTION)

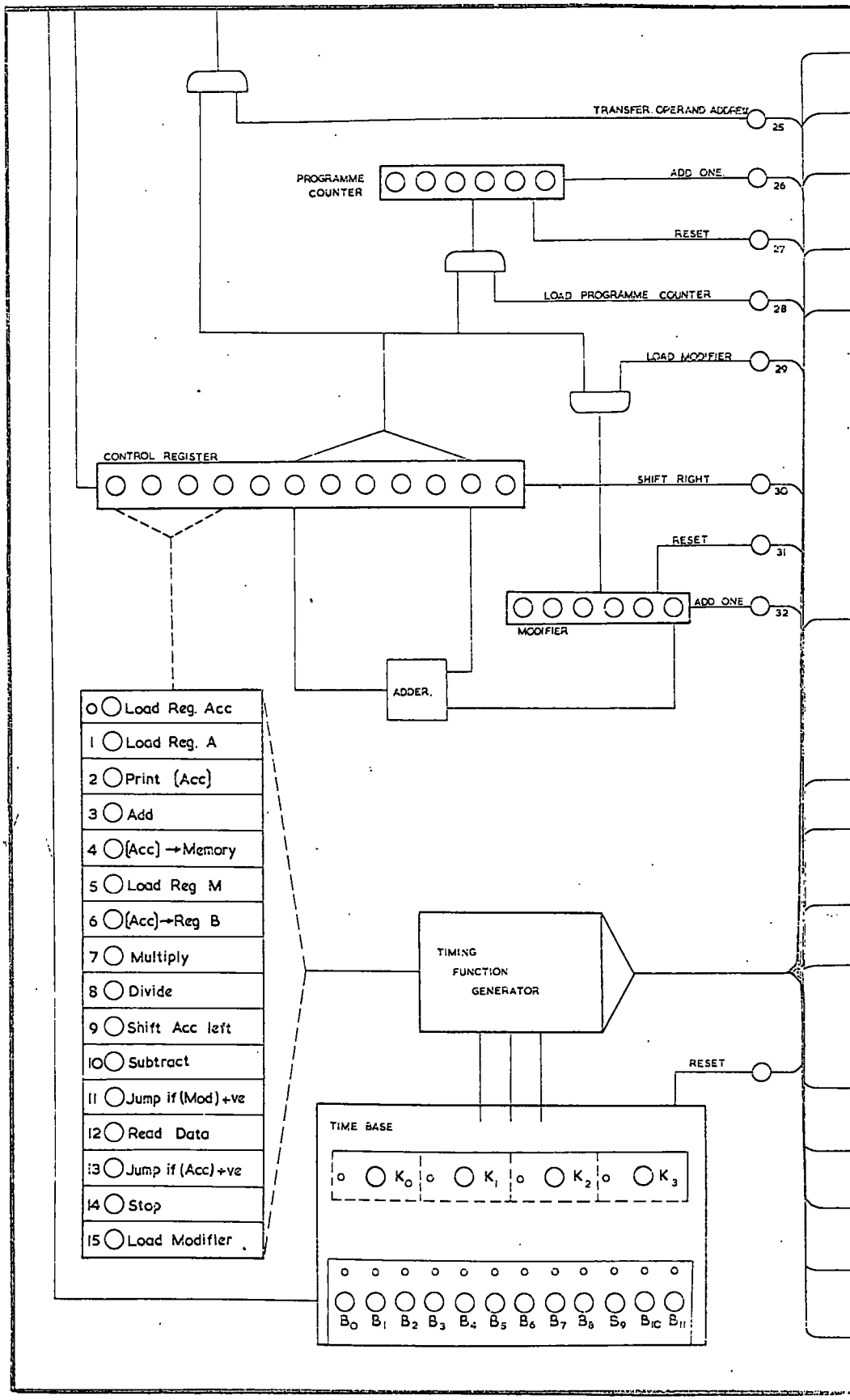


Fig. 41

referred to in Section 4.9 to be performed.

The control register is made the focal point of the Control section display panel, Fig.41. The timebase is represented by 16 lamps, of which twelve identify each of the bit (B) times and the remaining four identify the character (K) times. The sockets adjacent to each lamp may be used for oscilloscope examination of the timebase. The mimic diagram shows the decoding of the four most significant stages of the control register into the sixteen machine instructions, each of which is separately designated. During the execution of an instruction, one of the sixteen lamps will be on, indicating which instruction is being executed.

The Timing Function Generator is included on the front panel in order to show that instructions and timebase signals have to be gated to-gether to produce the timing pulses which are routed to all sections of the machine. The outputs of the control register (bits 6 to 11) are seen to fan out to the Programme Counter, Operand Address Gate and Modifier reg--ister. The single-bit modifier adder indicates the serial method of address modification. The timing functions which are relevant to the operation of the Control section are included and are also monitored by lamps. The student operating switches and the main machine controls are mounted on separate panels beneath the mimic diagram.

7.4 Mechanical Construction.

It will be appreciated that the main equipment cabinet is much larger than that required to accommodate the four I.S.E.P. rack units. This is due to the large display panels which are essential and to the motor driven worm gear arrangement which is incorporated to lift the I.S.E.P. frame clear of the top of the cabinet. This latter facility is desirable in order to gain access to the logic elements while still observing the lamps on the front panel.

The main equipment cabinet is designed to fit onto two 4' x 2' desks fixed back to back for operation from either side.

For classroom use larger display panels can be used. For a large lecture theatre these would be approximately 8' x 2' 6" each, with 24 watt lamps and thyristor driver circuits plugged into the existing machine.

CHAPTER VIII.

COMMISSIONING

8.1 Circuit Testing.

After construction, each of the major sub-sections, such as the timebase, instruction decode and the individual storage registers were separately checked using manually derived control pulses. The polarity and sequence of all timing functions were checked using an oscilloscope and a 5kHz. square wave input to the timebase.

Before attempting a complete check-out of the arithmetic and control sections it was necessary to construct a parallel to serial converter to simulate the operation of the tape storage unit. Each step in the execution of all machine instructions was checked in accordance with the order code of Section 5.6 and the majority of operations were found to function as planned. A small number of modifications were, however, necessary as listed in Section 8.2.

8.2 Circuit Modifications.

As a result of the tests, the following modifications were deemed to be necessary to correct errors and omissions in the design:

(i) At the end of the multiply and divide instructions the pulse, F.10, used to reset the timebase caused the time--base to be 'locked' in the reset state, $(I_7+I_8)K_3.B_{11}.C_L$. It was necessary to replace the master clock pulse C_L by the

pre-clock pulse C_{Lp}, so that under standby conditions the timing pulse F.10 is in the logic 1 state.

(ii) The modifier register was designed and constructed as a backward counter using 7476N J-K bistable elements in the ripple-through technique. It was appreciated that these elements change state during the negative going edge of the clock pulse, i.e. when the $\overline{Q}$ output of a preceding stage changes from logic 1 to logic 0. It was precisely this condition which was overlooked in considering parallel loading of the register. On loading a 1 into the least significant stage, it was found that 1 digits propagated through to the most significant stage. It was realised that a synchronous counter ought to have been employed. A satisfactory solution was obtained by arranging the circuit as a forward binary counter and loading the register with the number (64 - n), n being the required number of address modifications.

(iii) The same problem as in (ii) existed with the counter used to control the number of left shift pulses during the execution of instruction I_9 and the solution followed the same method as outlined for the modifier register. In this case the counter is loaded with the number (16 - n) where n is the number of required left shift pulses.

These changes are included in the logic schematic diagram Fig.22.

(iv) It was noticed that extraneous pulses were being entered into both the programme counter and modifier register. These pulses were coincident with the reset timebase pulse F.10. During the search for the extraneous pulses it was noticed that the timing pulses F.9 and F.12 were being masked by the reset timebase pulse F.10. These two pulses were then changed to operate at B_{10} time, so that oscilloscope examination of their waveform is possible.

The problem of the additional pulses entering the modifier register was solved by separating and re-routing the 0 volt line feeding the control register.

An examination of the wiring showed that considerable electro-magnetic coupling could exist between the conductors carrying pulses F.9 and F.10. This was supported by the fact that the pulse F.10 was supplying 34 logic loads. These conductors were separated in order to effect a solution.

(v) Following a departmental demonstration of the equipment it was decided to introduce a blanking pulse to an additional input of the driver circuit for the instruction decode lamps This pulse, $\overline{K_0}$, will cause all lamps to be turned off during the period when the instruction is being fed into the control register. Arrangements have been made for this modification to be done.

The time spent in commissioning and in tracing the few wiring errors, approximately 100 hours, was considerably less

than anticipated. The major part of this time was spent dealing with the problems outlined in Section 8.2.

CHAPTER IX.

SUGGESTED TEACHING PROGRAMME.

9.1 Instruction Manual.

It is essential that an instruction manual should be made available to those employing this equipment in a teaching laboratory. Although the production of a handbook is outside the scope of this thesis, it is appropriate to make some suggestions as to its form.

It is suggested that the manual should be divided into two sections. The first should describe all aspects of the equipment to assist in future modification and maintenance. It may be possible to assemble this section directly from the two theses concerned with the machine. The material may be conveniently presented in the following sub-sections:

(i) Details of construction and layout, including component identification drawings.

(ii) A description of the function of all circuitry with reference to the complete logic schematic diagrams.

(111) The purpose of the display panels and the associated manual control circuits.

(iv) A description of the four modes of operation, including the methods used in assembling the programme in the tape storage unit.

(v) Wiring and timing function identification tables.

The second section of the handbook should list the detailed procedures to be followed for a number of laboratory

exercises. These should be graded and of such a nature as to lead the student from the basic combinational circuits of the logic tutors to a complete appreciation of the internal functioning of a general purpose central processor.

The object of each of these exercises is likely to be the same for each of the courses referred to in Chapter 11. It is suggested that the exercises may be classified as follows:

(i) Revision of simple logic circuits.

(ii) Central processor organisation.

(iii) Manual operation of basic arithmetic processes.

(iv) Manual operation of machine instructions.

(v) The timebase.

(vi) Sequencing of minor commands.

(vii) Execution of a series of instructions.

(viii) Assembling a programme.

The manner in which the detail is handled, for each exercise, is expected to differ considerably between courses. Students following a technician course might be encouraged to identify 'blocks' of circuitry with particular functions and to become familiar with the methods used to test the performance of such blocks. On the other hand student engineers would be more closely concerned with the design features of the individual logic elements which comprise these blocks, e.g. loading, polarity, waveforms, etc.

Another factor requiring consideration is the student-hour quota which is allocated to particular aspects of the course being followed. It would be inappropriate here to specify the number of hours which should be allowed for work associated with this equipment. The author is, however, under the impression that third year students, following electronics courses in the Department of Applied Physics, at the University of Durham may expect to spend some 60/70 hours on an experiment of this nature. The detailed suggestions in Section 9.2 would be suitable for such a course. It is clear that other exercises must be drawn up to suit the individual requirements of particular courses.

9.2 Notes on Laboratory Exercises.

These notes are intended only as a guide to indicate how some computing techniques may be demonstrated using this equipment. The actual preparation of the exercises must of course be left to the discretion of the individual course tutor.

9.2.1 Revision of Simple Logic Circuits.

This exercise is expected to consolidate the work done on the basic logic tutors. The Type D and J-K bistable elements should be introduced at this stage. Simplified logic schematics may be used to show how the following techniques are realised using NAND/NOR gating with Type D and J-K bistable

elements:

(i) Shifting.

(ii) Binary counting.

(iii) Decoding.

(iv) Parallel transfer of data.

The shifting operation may be performed using the control register. With the timebase reset, the logic level applied to the D input of the register is determined by the state of the switch 2^0 on the data input panel associated with the memory section. By using this switch in conjunction with switch 31, the functions of serial entry and right shifting may be demonstrated.

The programme counter may be used for forward binary counting by simply operating the student switch No.26.

The four most significant digits of the control register are arranged in a pure binary code. The result of decoding these four bits is displayed on the lamps associated with the sixteen machine instructions. The various combinations should be deduced from an examination of a truth table and a logic schematic of the 'two-by-two' decoding network. Encoding of the four bits in the control register may be achieved using the method outlined for shifting.

Two methods of parallel transfer are available for demonstration. The first method utilises NAND gates, which are coupled to the pre-set inputs of either Type D or J-K

bistable elements. This method may be demonstrated by loading the programme counter from the control register. It should be pointed out that, with this method, it is necessary to reset the register to zero (00000) prior to the operation of the loading gate. The two student switches involved are Nos. 26 and 27. The same method may of course be demonstrated using the twelve data input switches and any of the registers in the arithmetic section.

The second method, involving direct coupling to the D inputs of a storage register, may be demonstrated using the two registers ACC. and B. The student switch No.7 applies a signal to the C inputs of the register B. This method allows parallel entry without the need for resetting.

9.2.2 Central Processor Organisation.

The three major sections of the machine are here included in the term 'central processor'. It will be necessary to outline the requirements of each section, and to show how the techniques established in Section 9.2.1 are used to allow communication between the various registers and control gates which form the three sections. The purpose of each register and control gate should be briefly defined. The machine word and data word may be introduced to establish the idea that the machine requires some fundamental direction from outside. The basic characteristics of each of the sub--sections may be demonstrated by using the appropriate

student switches, for example:

(i) Register A. :- used to store an operand during arithmetic operations. This sub-section is capable of overall reset to zero, (student switch No.2), parallel entry from core buffer register (student switch No.1), and left shifting (student switch No.3).

(ii) Programme Counter. :- used to instruct the memory regarding the location of the next instruction. This sub--section is capable of forward binary counting (student switch No.25), overall reset to zero (student switch No.26) and parallel loading of data from the control register (student switch No.27).

The other sub-sections requiring this kind of brief treatment are : Control Register, Modifier Register, Instruction Decode, Operand Address Gate, Register B, Parallel Adder, Register ACC. Register M and the main control switches Stop, Start, etc.

9.2.3 Manual Operation of Arithmetic Processes.

Using the twelve data and fifteen student switches associated with the Arithmetic display panel, the student should be encouraged to deduce the sequence of minor commands required to execute each of the four basic arithmetic processes. Information, in the form of flow diagrams, could be made available to assist in the determination of the correct sequence. Confirmation of these deductions may be

achieved using the single pulse mode of operation.

The sequence of minor commands required for binary multiplication is given as an example.

Multiplicand = 10100 , Multiplier = 101.

Step 1 Set multiplicand on data input switches.

Step 2 Reset register A - St. Sw. No.2

Step 3 Load register A - St. Sw. No.1

Step 4 Set multiplier on data input switches.

Step 5 Reset register M - St. Sw. No.14

Step 6 Load register M - St. Sw. No.15

Step 7 Reset ACC. register - St. Sw. No.10

Step 8 If least significant digit of register M is 1,

a) Transfer (ACC. register) to register B - St. Sw. No.7.

b) Reset Acc. register. St. Sw. No.10

c) Gate (register A) to adder - St. Sw. No.4

Gate (register B) to adder - St. Sw. No.6

Gate sum to ACC. register - St. Sw. No.11

If least significant digit of register M is 0 proceed to step 9.

Step 9 Shift (register A) one place left - St. Sw. No.3

Step 10 Shift (register M) one place right - St. Sw. No.13

Step 11 Repeat steps 8,9 and 10 until the contents of register M is zero when the product 1100100 will be held in the ACC. register.

9.2.4 Manual Operation of Machine Instructions.

In exercise 2, Section 9.2.2, the characteristics of each sub-section were identified and demonstrated using the student push-button switches. It is appropriate at this stage to examine these circuits in detail and identify their characteristics with the corresponding minor commands. To do this it will be necessary to make reference to the logic schematics of the registers, counters etc., and to become familiar with the actual circuit layout.

It should be established that a selection of minor commands is required to be sequenced to form a machine in--struction, and that one or more instructions are associated with each of the various aspects of machine operation, (e.g. input-output, conditional transfers, arithmetic, transfer of data and logical decisions.) By making reference to the order code, Section 5.6, each of the machine instructions may be demonstrated using the student and data input switches. Manual operation of the single pulse facility will allow the sequence of minor commands to be confirmed. This should indicate to the student that pre-determined periods of time are allocated for each operation.

9.2.5 The Timebase.

The timebase circuitry may be examined and identified with the counting and shifting operations of exercise 1, Section 9.2.1, With the timebase running continuously

(i.e. single pulse button held on) it is possible to examine the waveforms of the character, bit and clock times using an oscilloscope. These waveforms show the independent unique periods of time designated K_0, K_1, B_0, B_3, B_{11}, C_L etc. The purpose of the clock pulses ought to be established, (i.e. to maintain synchronism). The timebase observed during the single pulse operation indicates the nature of the machine cycle. During K_0 time the machine instruction may be seen to be entered serially into the control register, and executed during K_1 time. It should be possible to identify certain timing functions which are common to all instructions, e.g. reset timebase, add one to programme counter, and to observe that they occur during the same bit time for each instruction.

9.2.6 Sequencing of Minor Commands.

In this exercise it is expected that each of the timing pulses will be studied. The logic circuits required to realise these functions should be identified, and this will indicate the methods used to derive the timing pulses from the timebase and instruction signals. With the machine set for continuous operation (i.e. single pulse button held on), it is possible to display any of the timing functions using a double beam oscilloscope. It is expected that one or two of the timing functions will be examined in considerable detail. This may be done by encouraging the student to deduce the Boolean expression, timing waveform of input variables and

and actual logic circuitry. This section should be completed by observing the sequence of minor commands associated with each instruction. To do this it is necessary to have the machine set to operate in the slow automatic mode, with instructions being set up on the data input switches beneath the memory display panel. Operation of the Execute push--button will cause the timebase to step through one machine cycle. The cycle may be suspended at any point by operating the Stop switch. This may be required to allow more critical examination of the machine cycle. The Start switch is used to cause the machine to continue the execution of the instruction.

9.2.7 Execution of a Series of Instructions.

9.2.8 Assembling a Programme.

The notes for these exercises cannot be completed until the memory sections of the equipment have been constructed and tested. It is anticipated that these exercises will be concerned with the execution of a series of instructions and with the methods used to assemble a programme.

CHAPTER X.

CONCLUSIONS.

In concluding this thesis it is appropriate to make comments in the following areas.

(i) General observations regarding the project.

(ii) Notes on additions and /or modifications which would extend the capabilities of the present equipment.

(iii) Suggestions, based on the work described in this thesis, which are intended to assist in the construction of a Mark II machine.

10.1 General Conclusions.

It is obviously pleasing to confirm that the equipment does function generally in accordance with its original specification. This is due largely to the predictable performance of the Series 74N integrated circuits. The use of integrated circuits has removed all guess-work from the circuit design. It has been the author's experience, in the development of electronic systems using discrete components, that the majority of teething trouble can be attributed to errors and omissions in the design. The use of integrated circuits not only reduces the number of components to be considered, but also inspires the designer to express the design in considerably more detail than with discrete components. It is clear that the decision to employ integrated circuits has proved to be correct and that the choice of the

Series 74N was most appropriate since this series continues to be extended and is now also available from manufacturers other than Texas Instruments.

The only comment on the construction is concerned with one aspect of the printed circuit boards, since criticism of other constructional features can be made only after a period in service. The decision to fit approximately 25 integrated circuits to each printed circuit board was something of a risk in that a great deal depended on their reliability. It is the author's opinion that in view of their now proven reliability it is worth putting as many integrated circuits as possible on to a printed circuit board. This has the advantage of reducing edge connections which obviously improves reliability and simplifies external wiring.

It is difficult to assess the performance of the machine as a teaching aid, since there has been no oppurtunity to use the equipment in the teaching situation. There have, however, been some impartial comments and observations which indicate that the equipment is expected to be a useful and effective laboratory teaching aid.

10.2 Extensions to the Present Machine.

It is suggested that the following extensions should be incorporated to extend the capabilities of the equipment.

(i) The inclusion of a parallel to serial converter, coupled to the data input switches, which will allow a more convenient

method of manual insertion of instructions into the control register.

(ii) Some form of input-output equipment is desirable. The most likely form of input device is a paper tape reader, while the output could be interfaced to a line printer via a Binary - B.C.D converter and decoder.

(iii) The capacity of the core memory could be doubled, Bit 12 of the instruction word is spare and could be used for selecting this extension.

(iv) A large separate display panel could be incorporated for use in lecture theatres. Brief details are given in Section 7.4.

(v) The design of the display panels was considered to be a very important feature of the project. The discussion of Section 7.3 indicates the nature of the problems which were resolved. After these display panels had been built it was felt that more lamps could have been included without over--complicating the panels. These additions are possible with re-designed front panels.

10.3 Suggested Mark II Equipment.

It is obviously possible to consider the building of considerably enlarged teaching computers. It is felt that this temptation should be resisted in order to maintain an equipment which can still be understood by students at a comparatively early level. Two or three possibilities ~~could~~

could be considered however for changes to the specification for machines of this type. These changes would lead to basic modifications of the circuit design so that they cannot be included in the same way as the additions described in Section 10.2.

One feature of this equipment which differs from the current trend in teaching computers is that a twelve bit word is used compared to sixteen bits in other machines. The discussion of Section 4.7 was completely valid for the machine at that time bearing in mind the uncertainty of integrated circuit devices. In view of their now proven reliability the author would recommend that a future machine could be produced using a sixteen bit word. This would allow more machine instructions, larger numbers and a much higher capacity core memory section.

The other modification which might be considered is the inclusion of both operands and instructions in the core memory unit. This feature would be more typical of modern practice and would allow the tape store to be used as a backing store.

Although these suggestions sound rather basic they in no way minimizes the success of the present machine which, it is hoped, may prove to be of value to many generations of students at the University of Durham, and, possibly, through commercial channels, to other universities and colleges also.

APPENDIX 1.

TIMING FUNCTIONS.

The timing functions, although defined at various points in Chapter 6, are listed here for convenience. The loading, polarity and actual location for each function are given to assist in the future application of the machine. The loading is given in terms of the Series 74N unit load and the polarity refers to either positive going (p) or negative going (n) timing pulses. An interpretation of the location code is given in Appendix 2.

List of Timing Functions.

	Function.	Load.	Polarity.	Location.
F.1	Reset ACC. register.	24	n	R - 1 - 11
F.2	Reset register B.	24	n	T - 1 - 20
F.3	Shift (cont. reg.) right.	12	p	S - u - 26
F.3X	Shift (cont. reg.) right.	12	p	S - u - 25
F.6	Read core memory.	2	p	S - 1 - 9
F.7	Gate (core mem.) to ACC.	12	p	X - u - 18
F.8	Write data in core memory.	2	p	S - u - 31
F.9	Add 1 to prog. counter.	2	p	S - u - 3
F.10	Reset Timebase.	32	n	S - u - 1
F.11	Reset EXECUTE bistable.	2	n	S - u - 1
F.12	Add 1 to Modifier register.	2	p	S - u - 19
F.13	Reset register A.	24	n	T - 1 - 32
F.14	Gate (core mem.) to reg. A.	12	p	R - u - 16
F.15	Transfer (ACC.) to reg.B.	24	p	R - u - 26
F.16	Gate (reg. A) to adder.	11	p	R - 1 - 31
F.17	Gate (reg.B) to adder.	11	p	R - u - 17
F.18	Gate sum to ACC. register.	13	p	R - u - 32
F.19	Transfer (ACC.) to memory.	12	p	R - u - 9

	Function.	Load.	Polarity.	Location.
F.20	Reset register M.	24	n	T - 1 - 22
F.21	Gate (core mem.) to reg. M	12	p	R - u - 3
F.22	Enter sign digit in ACC.$_{12}$	2	p	U - u - 29
F.23	Shift (reg. M.) right.	24	p	T - u - 21
F.24	Shift (reg. A.) left.	24	p	T - u - 18
F.25	Shift (ACC. reg.) left.	24	p	U - 1 - 6
F.26	Clock input '4 bit count'	8	p	R - 1 - 2
F.27	Gate (reg. A not) to adder	11	p	R - u - 15
F.28	Reset programme counter	12	n	S - u - 9
F.29	Transfer (C.R) to prog; counter.	6	p	S - u - 4
F.31	Reset modifier register.	12	n	T - 1 - 31
F.32	Transfer (C.R) to mod. reg.	6	p	T - 1 - 3
F.33	Reset shift left counter.	8	n	T - 1 - 11
F.34	Transfer (C.R) to shift left counter.	4	p	T - 1 - 4
F.35	Add 1 to shift left counter.	4	p	U - 1 - 19
F.37	Gate operand address to mem.	6	p	T - 1 - 10
F.38	Set CARRY bistable.	2	n	U-internal
F.40	Gate mod.$_1$ to adder.	1	p	S - 1 - 6
F.41	Gate mod.$_2$ to adder.	1	p	S - 1 - 8
F.42	Gate mod.$_3$ to adder.	1	p	S - 1 - 3
F.43	Gate mod.$_4$ to adder.	1	p	S - u - 27
F.44	Gate mod.$_5$ to adder.	1	p	S - 1 - 10
F.45	Gate mod.$_6$ to adder.	1	p	S - u - 32
F.50	Reset DIVIDE bistable.	1	p	T - u - 29
F.51	Set DIVIDE bistable.	2	n	U-internal
F.52	Reset START bistable.	2	n	U-internal

APPENDIX 2.

WIRING SCHEDULE.

A.2.1 Notes on Nomenclature.

The wiring schedule lists the interconnections between the various sections of the equipment. There are 18 I.S.E.P. sockets arranged in pairs to accomodate the nine printed circuit boards. Each pair of sockets, upper and lower (see Fig.39), is identified with a particular board as follows:

Sockets	Board
P	C.U.-001
Q	T.B.--001
R	T.F.G.--001
S	T.F.G.--002
T	T.F.G.--003
U	Sp. Ct. No.1
W	A.U.--001
X	A.U.--002
Y	A.U.--003

Each socket has separately identified pins, numbered from 1 to 33. In the destination column of the wiring schedule each pin is specified by the following three part code:

First character denotes the particular socket.
Second character, u or l denotes upper or lower socket.
Third character denotes the pin number of the socket.

e.g. Q - u - 25, i.e. Socket Q upper, pin 25.

Each tag on the 264 way tag board is identified by row and column, e.g. row b, column 5 is designated b5.

A.2.2 Socket P Upper - Control Unit.

Pin	Function	Destination Socket.	Tag.
1	Control register, bit 5 not	S - u - 23	
2	Control register, bit 5	P - 1 - 5	h8
3	Control register, bit 4	Q - u - 1	h9
4	Control register, bit 4 not	Q - u - 3	
5	Control register, bit 3 not	Q - u - 2	
6	Control register, bit 3	Q - u - 4	h10
7	Control register, bit 2	Q - u - 5	h11
8	Control register, bit 2 not	Q - u - 8	
9	Control register, bit 1 not	Q - u - 6	h12
10	Control register, bit 1	Q - u - 7	h12
11	Shift control register right, F.3X	S - u - 25	
12	Shift control register right, F.3	S - u - 26	
13	D input to control register		w1
14	Set modifier carry bistable C_m	Q - 1 - 12	
15	C input to modifier carry bistable C_m	P - u - 11	
16	Reset modifier register, F.31	T - 1 - 31	
17	0 volt line		a9
18	Not used		
19	Not used		
20	Add one to programme counter, F.9	S - u - 3	
21	Programme counter, bit 1		i1
22	Programme counter, bit 2		i2
23	Programme counter, bit 3		i3
24	Programme counter, bit 4		i4
25	Programme counter, bit 5		i5
26	Programme counter, bit 6		i6
27	Load programme counter, F.29	S - u - 4	
28	Reset programme counter, F. 28	S - u - 9	
29	Load shift left counter, F.34	T - 1 - 4	
30	Shift pulses to ACC. register	U - 1 - 4	
31	C input to shift left counter	U - 1 - 19	
32	Reset shift left counter, F.33	T - 1 - 11	
33	Check (modifier register) is zero	S - u - 7	

A.2.3 Socket P Lower - Control Unit.

Pin.	Function.	Destination Socket.	Tag.
1	Gate mod. bit 6 to adder	S - u - 32	
2	Gate mod. bit 5 to adder	S - 1 - 10	
3	Modifier register, bit 6 not		i12
4	Gate mod. bit 4 to adder	S - u - 27	
5	Transfer CR_5 to CR_6	P - u - 2	
6	K_0	Q - u - 15	
7	Modifier register, bit 5 not		i11
8	0 volt line		a9
9	Load modifier register, F.32	T - 1 - 3	
10	Transfer operand address, F_6 F.37	T - 1 - 10	
11	Vcc		b9
12	Operand address, bit 6 not	Direct to memory	
13	Operand address, bit 7 not	Direct to memory	
14	Gate mod. bit 3 to adder	S - 1 - 3	
15	Modifier register, bit 4 not		i10
16	Operand address, bit 8 not	Direct to memory	
17	Operand address, bit 9 not	Direct to memory	
18	Modifier register, bit 3 not		i9
19	Modifier register, bit 1 not		i7
20	Modifier register, bit 2 not		i8
21	Operand address, bit 10 not	Direct to memory	
22	Operand address, bit 11 not	Direct to memory	
23	Clock input to modifier register F.12	S - 1 - 19	
24	Gate mod. bit 1 to adder	S - 1 - 6	
25	Gate mod. bit 2 to adder	S - 1 - 8	
26	K_1	Q - u - 13	
27	Control register, bit 12		h1
28	Control register, bit 11		h2
29	Control register, bit 10		h3
30	Control register, bit 9		h4
31	Control register, bit 8		h5
32	Control register, bit 7		h6
33	Control register, bit 6		h7

A.2.4 Socket Q Upper - Timebase.

Pin.	Function.	Destination Socket.	Tag.
1	Control register, bit 4	P - u - 3	
2	Control register, bit 3 not	P - u - 5	
3	Control register, bit 4 not	P - u - 4	
4	Control register, bit 3	P - u - 6	
5	Control register, bit 2	P - u - 7	
6	Control register, bit 1 not	P - u - 9	
7	Control register, bit 1	P - u - 10	
8	Control register, bit 2 not	P - u - 8	
9	K_3 not	R - 1 - 8	k4
10	K_3 not	T - u - 4	
11	K_2		k3
12	K_2 not	T - u - 2	
13	K_1	P - 1 - 26	k2
14	K_1 not	T - u - 1	
15	K_0	S - u - 28	k1
16	K_0 not		
17	0 volt line		a8
18	Vcc		b8
19	B_{11}	S - u - 21	j12
20	B_{11} not	R - 1 - 25	
21	B_{10}	S - u - 33	j11
22	B_{10} not	R - 1 - 26	
23	B_9	R - u - 27	j10
24	B_9 not	T - u - 12	
25	B_8	S - 1 - 26	j9
26	B_8 not	R - 1 - 30	
27	K_1	R - 1 -18	
28	B_7		j8
29	B_7 not	R - 1 - 24	
30	B_6		j7
31	B_6 not	R - 1 - 27	
32	Master clock not $\overline{C_L}$		
33	Master clock C_L	R - u - 31	

A.2.5 Socket Q Lower - Timebase.

Pin.	Function.	Destination Socket.	Tag.
1	B_5		j6
2	B_5 not	R $\frac{3}{8}$ 1 - 23	
3	B_4		j5
4	B_4 not	R - 1 - 22	
5	B_3	U - 1 - 18	j4
6	B_3 not	R - 1 - 21	
7	B_2	R - 1 - 28	j3
8	B_2 not	R - 1 - 20	
9	B_1	R - 1 - 15	j2
10	B_1 not	R - 1 - 19	
11	B_0	T - 1 - 13	j1
12	B_0 not	P - u - 14	
13	Reset timebase F.10	S - u - 1	
14	K_1 not	Q - u - 14	
15	Clock input to timebase C_{L10}	T - u - 26	
16	Pre-clock pulse not $\overline{C_{Lp}}$		
17	Pre-clock pulse C_{Lp}	R - u - 24	
18	I_{14}	U - u - 20	m3
19	I_6	S - 1 - 31	17
20	I_2		13
21	I_{10}	U - 1 - 9	111
22	I_{13}	S - u - 13	m2
23	I_5	S - 1 - 30	16
24	I_1	S - 1 - 22	12
25	I_9	T - 1 - 5	110
26	I_{12}		m1
27	I_4	S - 1 - 27	15
28	I_0	S - 1 - 21	11
29	I_8	R - u - 29	19
30	I_{15}	T - 1 - 24	m4
31	I_7	R - u - 21	18
32	I_3	R - 1 - 17	14
33	I_{11}	S - u - 10	112

A.2.6 Socket R Upper - Timing Function Generator No.1.

Pin.	Function.	Destination Socket.	Tag.
1	Student switch 19		q7
2	$I_5.K_1$ not	S - 1 - 29	
3	F.21	Y - u - 15	
4	Not used		
5	Not used		
6	Student switch 1		p1
7	$I_1.K_1$ not	S - 1 - 15	
8	Student switch 21		q9
9	F.19	W - u - 7	v8
10	F.7	X - u - 18	v12
11	$I_{10}.K_1$not	R - 1 - 10	
12	Student switch 14		q2
13	Student switch 7		p7
14	$I_0.K_1$not	S - 1 - 14	
15	F.27	X - u - 16	
16	F.14	X - u - 13	v1
17	F.17	W - u - 3	v6
18	Student switch 15		q3
19	$I_4.K_1$not	S - 1 - 20	
20	B_2	R - 1 - 28	
21	I_7	S - 1 - 17	
22	Not used		
23	Student switch 27		r3
24	Pre-clock pulse C_{Lp}	S - 1 - 32	
25	B_1	R - 1 - 15	
26	F.15	W - u - 5	v7
27	B_9	Q - u - 23	
28	Q output of CARRY bistable	U - u - 9	
29	I_8	S - 1 - 25	
30	Student switch 18		q6
31	Master clock C_L	T - u - 31	
32	F.18	X - u - 21	v11
33	Vcc		b7

A.2.7 Socket R Lower - Timing Function Generator No.1

Pin.	Function	Destination Socket.	Tag.
1	0 volt line		a7
2	F.26	Y - 1 - 16	
3	Not used		
4	$I_0.K_1$ not	S - 1 - 14	
5	$I_6.K_1$ not	S - 1 - 12	
6	Not used		
7	$I_7.K_1$ not	S - 1 - 16	
8	K_3	S - u - 16	
9	$I_3.K_1$ not	S - 1 - 23	
10	$I_{10}.K_1$ not	R - u - 11	
11	F.1	W - u - 10	v10
12	$I_8.D$	T - u - 19	
13	Not used		
14	K_1+K_2	T - u - 6	
15	B_1	T - 1 - 19	
16	$I_7.M_1$	Y - u - 23	
17	I_3	S - 1 - 24	
18	K_1	S - 1 - 19	
19	B_1 not	S - 1 - 7	
20	B_2 not	S - 1 - 4	
21	B_3 not	S - 1 - 2	
22	B_4 not	S - 1 - 5	
23	B_5 not	S - 1 - 11	
24	B_7 not	Q - u - 29	
25	B_{11} not	Q - u - 20	
26	B_{10} not	Q - u - 22	
27	B_6 not	S - u - 30	
28	B_2	R - u - 20	
29	Student switch 16		q4
30	B_8 not	Q - u - 26	
31	F.16	X - u - 17	v4
32	F.17	R - u - 17	
33	Student switch 17		q5

A.2.8 Socket S Upper - Timing Function Generator No.2.

Pin.	Function.	Destination Socket.	Tag.
1	F.10	Q - 1 - 13	u12
2	Student switch 9		p9
3	F.9	P - u - 20	m6
4	F.29	P - u - 27	m8
5	Student switch 29		r5
6	$B_2.C_L$	T - 1 - 15	
7	Check (modifier register) is zero	P - u - 33	
8	Control register, bit 5	P - u - 2	
9	F .28	P - u - 28	
10	I_{11}	Q - 1 - 33	
11	Student switch 28		r4
12	$B_1.C_L$	T - 1 - 14	
13	I_{13}	Q - 1 - 22	
14	ACC. register, bit 12 not	W - u - 11	
15	Not used		
16	K_3	R - 1 - 8	
17	Pre-clock pulse C_{Lp}	S - 1 - 32	
18	Student switch 10		p10
19	F.12	P - u - 20	m12
20	$B_{11}.C_L$ not	U - u - 22	
21	B_{11}	Q - u - 19	
22	K_1	S - 1 - 19	
23	Control register, bit 5 not	P - u - 1	
24	I_7 I_8	T - 1 - 30	
25	F.3X	P - u - 11	
26	F.3	P - u - 12	m9
27	F.43	P - 1 - 4	
28	K_0	Q - u - 15	
29	Student switch 3		p3
30	B_6 not	T - u - 13	
31	F.8		s12
32	F.45	P - 1 - 1	
33	Vcc		b6

A.2.9 Socket S Lower - Timing Function Generator No.2.

Pin.	Function.	Destination Socket.	Tag.
1	0 volt line		a6
2	B_3 not	T - u - 14	
3	F.42	P - 1 - 14	
4	B_2 not	R - 1 - 20	
5	B_4 not	R - 1 - 22	
6	F.40	P - 1 - 24	
7	B_1 not	T - 1 - 11	
8	F.41	P - 1 - 25	
9	F.6		s11
10	F.44	P - 1 - 2	
11	B_5 not	T - u - 10	
12	$I_6.K_1$ not	R - 1 - 5	
13	$\overline{K_1(I_0\ I_1\ I_4\ I_5)}$	T - 1 - 28	
14	$I_0.K_1$ not	R - 1 - 4	
15	$I_1.K_1$ not	T - 1 - 16	
16	$I_7.K_1$ not	R - 1 - 7	
17	I_7	S - 1 - 21	
18	$I_8.K_1$ not	T - 1 - 8	
19	K_1	T - 1 - 17	
20	$I_4.K_1$ not	R - u - 19	
21	I_0	Q - 1 - 28	
22	I_1	Q - 1 - 24	
23	$I_3.K_1$ not	R - 1 - 9	
24	I_3	R - 1 - 17	
25	I_8	Y - u - 19	
26	B_8	Q - u - 25	
27	I_4	Q - 1 - 27	
28	Student switch 6		p6
29	$I_5.K_1$ not	T - 1 - 9	
30	I_5	Q - 1 - 23	
31	I_6	Q - 1 - 19	
32	Pre-clock pulse C_{Lp}	S - u - 17	
33	B_{10}	T - u - 7	

A.2.10 Socket T Upper - Timing Function Generator No.3

Pin.	Function.	Destination Socket.	Tag.
1	K_1 not	Q - u - 14	
2	K_2 not	Q - u - 12	
3	Not used		
4	K_3 not	Q - u-- 10	
5	K_2+K_3		
6	K_1+K_2	R - 1 - 14	
7	B_{10}	S - 1 - 33	
8	Not used		
9	Not used		
10	B_5 not	S - 1 - 11	
11	B_1 not	U - 1 - 15	
12	B_9 not	Q - u - 24	
13	B_6 not	S - u - 30	
14	B_3 not	S - 1 - 2	
15	Not used		
16	B_5+B_9		
17	I_8	Y - u - 19	
18	F.24	X - u - 15	
19	$I_8.D$	U - u - 15	
20	Not used		
21	F.23	Y - 1 - 29	
22	Student switch 23		q11
23	$\overline{I_8.K_3.B_{10}.C_{L10}}$	U - 1--3	
24	Operand alignment pulses	Y - 1 - 12	
25	Student switch 24		q12
26	C_{L10}	U - 1 - 8	
27	Not used		
28	Not used		
29	F.50	U - u - 16	
30	Not used		
31	Pre-clock pulse	S - u - 32	
32	Student switch 33		r9
33	Vcc		b5

A.2.11 Socket T Lower - Timing Function Generator No.3.

Pin.	Function.	Destination. Socket.	Tag.
1	0 volt line		a5
2	Student switch 32		r8
3	F.32	P - 1 - 9	
4	F.34	P - u - 29	
5	I_9	U - 1 - 11	
6	Student switch 34		r10
7	F.22	U - 1 - 17	
8	$I_8.K_1$ not	U - 1 - 14	
9	$I_5.K_1$ not	S - 1 - 29	
10	F.37	P - 1-- 10	
11	F.33	P - u - 32	
12	$B_2.C_L$not		
13	B_0	Q - 1 - 11	
14	$B_1.C_L$	S - u - 12	
15	$B_2.C_L$	U - u - 11	
16	$I_1.K_1$ not	S - 1 - 15	
17	K_1	S - 1 - 19	
18	$B_1.C_L$ not		
19	B_1	R --1 - 15	
20	F.2	W - u - 6	
21	I_7	Y - u - 24	
22	F.20	Y - u - 16	
23	Pre-clock pulse C_{Lp}	R - u - 24	
24	I_{15}	Q - 1 - 30	
25	B_2	U --u - 26	
26	Student switch 13		q1
27	Student switch 37		s1
28	$\overline{K_1(I_0+I_1+I_4+I_5)}$	S - 1 - 13	
29	Student switch 20		q8
30	I_7+I_8	S - u - 24	
31	F.31	P - u - 16	
32	F.13	X - u - 14	
33	Master clock C_L	U - u - 23	

A.2.12 Socket U Upper - Special Circuit No.1.

Pin.	Function.	Destination. Socket.	Tag.
1	Student switch 'Execute'		u1
2	Low frequency input (10Hz.)		
3	Slow		u2
4	Fast		u3
5	Student switch 'Start'		u4
6	Student switch 'Stop'		u5
7	5kHz signal from memory unit		u6
8	F.10	S - u - 3	
9	Q output of CARRY bistable	Y - 1 - 24	
10	$I_8.K_1$ not	T - 1 - 8	
11	$B_2.C_L$	T - 1 - 15	
12	I_8.D not		
13	STOP		w2
14	MAN		u7
15	I_8.D	Y - 1 - 23	
16	F.50	T - u - 29	
17	F.18	X - u - 21	
18	Carry-out from parallel adder, C_{out}	X - u - 29	
19	Not used		
20	I_{14}	Q - 1 - 18	
21	Alignment signal from memory		u8
22	$B_{11}.C_L$ not	S - u - 20	
23	Master clock C_L	T - 1 - 31	
24	K_1	S - u - 22	
25	I_8	Y - u - 19	
26	B_2	T - 1 - 25	
27	'4 bit count' bit 1 not	Y - 1 - 2	
28	'4 bit count' bit 2 not	Y - 1 - 4	
29	F.22	W - u - 14	
30	'4 bit count' bit 3 not	Y - 1 - 7	
31	'4 bit count' bit 4 not	Y - 1 - 8	
32	Student switch''Manual'		u9
33	Vcc		b4°

A.2.13 Socket U Lower - Special Circuit No.1.

Pin.	Function.	Destination. Socket.	Tag.
1	0 volt line		a4
2	Student switch 25		r1
3	$I_8.K_3.B_{10}.C_{L10}$	T - 1 - 23	
4	Shift left pulses to ACC. register	P - u - 30	
5	Operand alignment pulses	Y - 1 - 10	
6	F.25	W - u - 9	
7	Sign digit	Y - u - 33	
8	Clock input to timebase C_{L10}	T - u - 26	
9	I_{10}	Q - 1 - 21	
10	$I_{10}.K_1$ not	R - u - 11	
11	I_9	T - 1 - 5	
12	JAY		u10
13	Sign digit ACC. register bit 12	W - u - 15	
14	$I_8.K_1$ not	T - 1 - 8	
15	B_1not	T - u - 11	
16	$I_8.K_1.B_1$	Y - 1 - 13	
17	Not used		
18	B_3	Q - 1 - 5	
19	F.35	P - u - 31	
20	Q output of MANUAL bistable		w6
21	Not used		
22	$I_8.D$	U - u - 15	
23	Not used		
24	Register A bit 11	X - 1 - 32	
25	Carry-in of parallel adder, C_{in}	X - 1 - 18	
26	$I_8.D.A_{11}$	X - 1 - 21	
27	Not used		
28	Not used		
29	Not used		
30	Not used		
31	Not used		
32	Q output of EXECUTE bistable		w5
33	Q output of START bistable		w4

A.2.14 Socket W Upper - Arithmetic Unit AU.001.

Pin.	Function.	Destination Socket.	Tag.
1	0 volt line		a3
2	Vcc		b3
3	F.17	R - u - 17	
4	0 volt line	W - u - 1	
5	Transfer (ACC. register) to register B	R - u - 26	
6	Reset register B F.2	T - 1 - 20	
7	Transfer (ACC.register) to memory	R - u - 9	
8	ACC. register, bit 11 not	Y - 1 - 9	
9	Shift ACC. register left F.25	U - 1 - 6	
10	Reset ACC. register F.1	R - 1 - 11	
11	ACC. register, bit 12	Y - u - 31	e12
12	ACC. register, bit 12 not	Y - u - 32	
13	Set input ACC. register, bit 12	X - u - 24	
14	F.22	U - u - 29	
15	Sign digit ACC. register bit 12	U - 1 - 13	
16	ACC. register, bit 11	Y - 1 - 18	e11
17	Set input, ACC. register, bit 11	X - u - 23	
18	To core buffer register bit 12		d12
19	To core buffer register bit 11		d11
20	To core buffer register bit 9		d9
21	To core buffer register bit 10		d10
22	ACC. register, bit 10		e10
23	Set input ACC. register bit 10	X - u - 25	
24	ACC. register bit 9		e9
25	Set input, ACC. register, bit 9	X - u - 22	
26	ACC. register, bit 8		e8
27	Set input, ACC. register, bit 8	X - 1 - 1	
28	ACC. register, bit 7		e7
29	Set input, ACC. register, bit 7	X - u - 33	
30	To core buffer register bit 8		d8
31	To core buffer register bit 7		d7
32	To core buffer register bit 5		d5
33	To core buffer register bit 6		d6

A.2.15 Socket W Lower - Arithmetic Unit AU.001.

Pin.	Function.	Destination Socket.	Tag.
1	ACC. register, bit 6		e6
2	Set input, ACC. register, bit 6	X - 1 - 2	
3	ACC. register, bit 5		e5
4	Set input, ACC. register, bit 5	X - u - 32	
5	ACC. register, bit 4		e4
6	Set input, ACC. register, bit 4	X - 1 - 11	
7	ACC. register, bit 3		e3
8	Set input, ACC. register, bit 3	X - 1 - 10	
9	To core buffer register bit 4		d4
10	To core buffer register bit 3		d3
11	To core buffer register bit 1		d1
12	To core buffer register bit 2		d2
13	ACC. register, bit 2		e2
14	Set input, ACC. register, bit 2	X - 1 - 12	
15	ACC. register, bit 1		e1
16	Set input, ACC. register, bit 1	X - 1 - 9	
17	Not used		
18	Not used		
19	Not used		
20	Not used		
21	D input of ACC. register, bit 12	Y - u - 21	
22	Register B, bit 1 to adder	X - 1 - 19	
23	Register B, bit 2 to adder	X - 1 - 14	
24	Register B, bit 3 to adder	X - 1 - 13	
25	Register B, bit 4 to adder	X - 1 - 15	
26	Register B, bit 5 to adder	X - 1 - 8	
27	Register B, bit 6 to adder	X - 1 - 4	
28	Register B, bit 7 to adder	X - 1 - 3	
29	Register B, bit 8 to adder	X - 1 - 5	
30	Register B, bit 9 to adder	X - u - 31	
31	Register B, bit 10 to adder	X - u - 27	
32	Register B, bit 11 to adder	X - u - 26	
33	Not used.		

A.2.16 Socket X Upper - Arithmetic Unit AU.002.

Pin.	Function.	Destination Socket.	Tag.
1	Q output, core buffer register bit 1	Y - u - 12	
2	Q output, core buffer register bit 2	Y - u - 11	
3	Q output, core buffer register bit 3	Y - u - 10	
4	Q output, core buffer register bit 4	Y - u - 9	
5	Q output, core buffer register bit 5	Y - u - 8	
6	Q output, core buffer register bit 6	Y - u - 7	
7	Q output, core buffer register bit 7	Y - u - 6	
8	Q output, core buffer register bit 8	Y - u - 5	
9	Q output, core buffer register bit 9	Y - u - 4	
10	Q output, core buffer register bit 10	Y - u - 3	
11	Q output, core buffer register bit 11	Y - u - 2	
12	Q output, core buffer register bit 12	Y - u - 1	
13	Load register A F.14	R - u - 16	
14	Reset register A F.32	T - 1 - 13	
15	Shift register A left F.24	T - u - 18	
16	Transfer A not to adder F.27	R - u - 15	
17	Transfer A to adder F.16	R - 1 - 31	
18	Load ACC. register F.7	R - u - 10	
19	Vcc	X - u - 28	b2
20	0 volt line		a2
21	Transfer sum to ACC. register F.18	U - u - 17	
22	Set input, ACC. register bit 9	W - u - 25	
23	Set input, ACC. register bit 11	W - u - 17	
24	Set input, ACC. register bit 12	W - u - 13	
25	Set input, ACC. register bit 10	W - u - 23	
26	Register B bit 11 to adder	W - 1 - 32	
27	Register B bit 10 to adder	W - 1 - 31	
28	Vcc	X - u - 19	
29	Carry-out from parallel adder C_{out}	U - u - 18	
30	Carry in bit 9	X - 1 - 6	
31	Register B bit 9 to adder	W - 1 - 30	
32	Set input, ACC. register bit 5	W - 1 - 4	
33	Set input, ACC. register bit 7	W - u - 29	

A.2.17 Socket X Lower - Arithmetic Unit AU.002.

Pin.	Function.	Destination Socket.	Tag.
1	Set input, ACC. register bit 8	W - u - 27	
2	Set input, ACC. register bit 7	W - 1 - 2	
3	Register B bit 7 to adder	W - 1 - 28	
4	Register B bit 6 to adder	W - 1 - 27	
5	Register B bit 8 to adder	W - 1 - 29	
6	Carry out bit 8	X - u - 30	
7	Carry in bit 5	X - 1 - 16	
8	Register B bit 5 to adder	W - 1 - 26	
9	Set input, ACC. register bit 1	W - 1 - 16	
10	Set input, ACC. register bit 3	W - 1 - 8	
11	Set input, ACC. register bit 4	W - 1 - 6	
12	Set input, ACC. register bit 2	W - 1 - 14	
13	Register B bit 3 to adder	W - 1 - 24	
14	Register B bit 2 to adder	W - 1 - 23	
15	Register B bit 4 to adder	W - 1 - 25	
16	Carry out bit 4	X - 1 - 7	
17	Register A bit 12 not	Y - u - 29	
18	Carry in bit 1	U - 1 - 25	
19	Register B bit 1 to adder	W - 1 - 22	
20	Register A bit 1		f1
21	D input bit 1 register A	U - 1 - 26	
22	Register A bit 2		f2
23	Register A bit 11 not	Y - 1 - 11	
24	Register A bit 3		f3
25	Register A bit 4		f4
26	Register A bit 5		f5
27	Register A bit 6		f6
28	Register A bit 7		f7
29	Register A bit 8		f8
30	Register A bit 9		f9
31	Register A bit 10		f10
32	Register A bit 11		f11
33	Register A bit 12	Y - u - 30	

A.2.18 Socket Y Upper - Arithmetic Unit AU.003.

Pin.	Function.	Destination Socket.	Tag.
1	Q output, core buffer register bit 12	X - u - 12	c12
2	Q output, core buffer register bit 11	X - u - 11	c11
3	Q output, core buffer register bit 10	X - u - 10	c10
4	Q output, core buffer register bit 9	X - u - 9	c9
5	Q output, core buffer register bit 8	X - u - 8	c8
6	Q output, core buffer register bit 7	X - u - 7	c7
7	Q output, core buffer register bit 6	X - u - 6	c6
8	Q output, core buffer register bit 5	X - u - 5	c5
9	Q output, core buffer register bit 4	X - u - 4	c4
10	Q output, core buffer register bit 3	X - u - 3	c3
11	Q output, core buffer register bit 2	X - u - 2	c2
12	Q output, core buffer register bit 1	X - u - 1	c1
13	0 volt line		a1
14	Vcc		b1
15	Load register M,F.21	R - u - 3	
16	Reset register M,F.20	T - 1 - 22	
17	Register M, bit 1		g1
18	Register M, bit 2		g2
19	I_8	U - u - 25	
20	Not used		
21	$I_8.M_1$	W - 1 - 21	
22	$I_7.M_1$ not		
23	$I_7.M_1$	R - 1 - 16	
24	I_7	T - 1 - 21	
25	Register M, bit 3		g3
26	Register M, bit 4		g4
27	Register M, bit 5		g5
28	Register M, bit 6		g6
29	Register A bit 12 not	X - 1 - 17	
30	Register A bit 12	X - 1 - 33	
31	ACC. register, bit 12	W - u - 11	
32	ACC. register, bit 12 not	W - u - 12	
33	Sign digit	U - 1 - 7	

A.2.19 Socket Y Lower - Arithmetic Unit AU.003.

Pin.	Function.	Destination Socket.	Tag.
1	Set '4 bit count' UP	Y - 1 - 13	
2	'4 bit count', bit 1 not	U - u - 27	
3	'4 bit count', Down	Y - 1 - 23	
4	'4 bit count', bit 2 not	U - u - 28	
5	Not used		
6	Clear '4 bit count' F.20	Y - u - 16	
7	'4 bit count' bit 3 not	U - u - 30	
8	'4 bit count' bit 4 not	U - u - 31	
9	ACC. register, bit 11 not	W - u - 8	
10	Alignment pulses to ACC. register	U - 1 - 5	
11	Register A, bit 11 not	Y - 1 - 17	
12	Alignment pulses to register A	T - u - 24	
13	$I_8.K_1.B_1$	U - 1 - 16	
14	C_{L10}	U - 1 - 8	
15	Not used		
16	Not used		
17	Register A, bit 11 not	Y - 1 - 11	
18	ACC. register, bit 11 not	W - u - 16	
19	Register M, bit 7		g7
20	Register M, bit 8		g8
21	Register M, bit 9		g9
22	Register M, bit 10		g10
23	$I_8.D$	U - u - 15	
24	Q output of CARRY bistable.	U - u - 9	
25	Not used		
26	Not used		
27	C input bit 12 register M		
28	Register M, bit 11		g11
29	Shift (register M) right F.23	T - u - 21	
30	Register M, bit 12 not		
31	D input bit 11 register M		
32	Register M, bit 12		g12
33	D input bit 12 register M		

LIST OF REFERENCES.

1. A.Smith: The Design and Construction of the Memory units of a Demonstration Electronic Digital Computer. Proposed M.Sc. thesis; University of Durham. 1969.
2. S.W. Hockey and S.C.P. Parry: A Computer Project at Marl--borough, and its Implications in Science Teaching. Int. J. Elec. Engng. Educ. Vol. 2 p.p 233 - 240. Pergamon Press, 1964.
3. A. Wilkinson: Computer Models; Edward Arnold, 1968.
4. D.H. Green: A Practical Aid to the Teaching of Digital Computing Techniques; Int. J. Elec. Engng. Educ. Vol. 2 pp 15 - 26. Pergamon Press, 1964.
5. S.L. Hurst: Logic and Counting Demonstration Equipments. Int. J. Elec. Engng. Educ. Vol. 4. p.p 259 - 268. Pergamon Press,,1966.
6. J.R. Abrahams: LAN DEC : a Low Cost Digital Computer for Teaching. Int. J. Elec. Engng. Educ. Vol. 2. p.p 155 - 160. Pergamon Press, 1964.
7. I.H. Gould and F.S. Ellis: Digital Computing Technology, Reinhold, 1962.
8. Litton Industries, Data Systems Div.: Digital Computer Fundamentals, Prentice Hall, 1965.
9. B.S. 3527. 1962. Glossary of Terms used in Automatic Data Processing, p.p 7 - 8, Nos. 10007, 10015.
10. E.L. Braun: Digital Computer Design, Academic Press, 1963.
11. G. Hintze: Fundamentals of Digital Machine Computing; Springer-Verlag, 1966.
12. E.L. Braun: Digital Computer Design; p.p. 52 - 66; Academic Press, 1963.
13. Texas Instruments: Integrated Circuit Application Manual; Henry Bart & Son, 1966.
14. G.A. Maley and E.J. Skiko: Modern Digital Computers p.p. 62 - 64; Prentice Hall, 1964.
15. Daphne.P. Kilner: General Purpose Digital Computers, Control Survey 35; Control, Feb. 1966.
16. Texas Instruments: Data Book 2, Digital Integrated Circuits.
17. B.S. 3527, 1962. Glossary of Terms used in Automatic Data Processing, p.p. 82, No. 34012.
18. Formica Ltd., Specification leaflet:- M.L.80 Glass Epoxy Materials for Multilayer Circuits.

ABSTRACT OF M.Sc. THESIS

The Design and Construction of the Control and Arithmetic units of a Demonstration Electronic Digital Computer.

R.H. Brunskill

This thesis is concerned with the design and construction of the control and arithmetic units of a small digital computer to be used in the teaching of modern computer electronics. A review of existing digital electronic teaching aids of commercial origin shows grave deficiencies in this field. The specification of the present machine is drawn up to include most of the features of modern computer technology such as parallel operation, a B line modifier, core and tape storage facilities, and conditional transfer instructions, while keeping the circuitry simple and the cost as low as possible.

The block schematic diagrams for the machine are derived from the specification and these enable the overall design to be considered in detail. It is shown to be desirable to use integrated logic circuits and in particular the Series 74N which is manufactured by Texas Instruments Ltd. The detailed circuit design is then presented in terms of this series of logic elements. The design of the memory unit of the machine is dealt with elsewhere.

The demonstration aspects of the equipment are borne in mind throughout the design. Large display panels and manual operating facilities are essential, as is the ability to gain access to the entire logic circuitry. The dual-in-line elements are assembled on double-sided plug-in printed circuit boards which are housed in standard 16" rack units. The equipment functioned as intended after a very small number of modifications during the commissioning period.

A guide to laboratory exercises is included to indicate the intended application of the machine. It is concluded that equipment of this type is of great value in the teaching of both under-graduate and technician courses in the fields of electronics, control and computing science.