ELLIOTT 900

Volume 4: ENGINEERING MAINTENANCE

Part 1: BASIC EQUIPMENT

Section 2: 903 CENTRAL PROCESSOR (DPA 1)

Contents

Page

Chapter 1: INTRODUCTION

1.1 General.. 1

1.1.1 Mechanical 1

1.1.2 Functional 1

1.2 Logic Boards 2

1.2.1 Standard Logic Boards 2

1.2.2 Processor Matrix Boards 3

1.3 Conventions of Logic Reference 3

Chapter 2: CONTROL UNIT

2.1 Introduction 5

2.2 Computer Modes 5

2.2.1 Auto Mode 5

2.2.2 Operate Mode 5

2.2.3 Test Mode 6

2.3 Control Unit Circuitry 6

Page

2.4 Audio Unit. 6

2.5 Lamp Driver Circuits 7

Chapter 3: MANUAL CONTROL

3.1 Introduction 9

3.2 Logic Operation 9

3.2.1 Auto Mode Operation 9

3.2.2 Operate Mode Operation 10

3.2.3 Test Mode Operation 11

Chapter 4: CENTRAL TIMER

4.1 Introduction 13

4.2 Typical Timer Cycle 13

4.3 Timer Stopping Conditions 14

4.3.1 Order Stop 14

4.3.2 Wait for Store 14

4.3.3 Wait for Peripheral 14

4.3.4 Set Matrix Address 0. 14

4.3.5 Test Mode Stop 15

Chapter 5: CONTROL MATRIX

5.1 Introduction 17

5.2 Matrix Proper 18

5.3 Matrix Address Line Logic. 18

5.3.1 Test Counter for Zero 19

5.4 Function Decode 20

5.5 Matrix Conditions 21

5.6 Microprogram 22

Chapter 6: INSTRUCTION CODE 23

Page

Chapter 7: REGISTERS, FUNCTION AND COLLATE UNITS

7.1 Introduction 25

7.2 SCR and B-Registers 25

7.3 J-Register 26

7.4 G-Register (Buffer) 26

7.5 A- and Q-Registers (Accumulator) 27

7.6 M-Register (Store Access) 27

7.7 I-Register (Functions) 28

7.8 P-Register (Peripheral) 28

7.9 Function Unit (F-Unit) 28

7.9.1 Addition 29

7.9.2 Subtraction.. 29

7.9.3 Summation Rules 29

7.9.4 Carry Rules 31

7.10 Collate (Logical AND NOT) Unit 31

Chapter 8: PROCESS COUNTER

8.1 Introduction 33

8.2 Logic Description 34

Chapter 9: CONTROL INSTRUCTION

9.1 Initial Start 37

9.1.1 Auto Mode Start 37

9.1.2 Operate and Test Mode Start 38

9.2 Manual Entry Loop (Test Mode Only) 38

9.3 Normal Cycle of Control Instruction 39

Chapter 10: OVERFLOW AND SHIFT UNIT

10.1 Introduction 43

10.2 Uses of Overflow and Shift System 43

10.2.1 Overflow Unit Proper 43

Page

10.2.2 Normal Overflow and True Sign.. 44

10.3 Shift and Overflow Logic Description 44

Chapter 11: SHIFT, MULTIPLY AND DIVIDE

11.1 Introduction 45

11.2 Shift (Fn 14) 45

11.3 Examples of Shift 47

11.3.1 Left Shift 47

11.3.2 Right Shift 48

11.4 Function 12 (Multiply) 49

11.4.1 Worked Examples 50

11.5 Function 13 (Divide) 60

11.5.1 Worked Examples 61

Chapter 12: OTHER COMPUTING INSTRUCTIONS

12.1 Introduction 75

12.2 Function 15 75

12.3 Block Transfer (Function 14A) 75

Chapter 13: INITIAL INSTRUCTIONS

13.1 Introduction 79

13.2 Selection 79

Chapter 14: INPUT/OUTPUT

14.1 Introduction 81

14.2 Peripheral Interface Requirements 81

14.3 General Input/Output.. 82

14.3.1 Input.. 82

14.3.2 Output 82

14.4 Paper Tape Input/Output 82

14.4.1 Input 82

Page

14.4.2 Output 83
14.5 Peripheral Addressing 83
14.6 Input/Output Control 84
14.6.1 Logic Description 84

Chapter 15: INTERRUPTS
15.1 Introduction 89
15.2 Initial Conditions 89
15.3 Typical Interrupt Cycle 90
15.4 Interrupt Production 91
15.4.1 On Line and Manual 91
15.4.2 Trace 91
15.5 Set B-Register (Fn 0) 91
15.6 Program Terminate 92

Chapter 16: POWER DISTRIBUTION
16.1 Introduction 93
16.2 A. C. Distribution 93
16.3 D. C. Distribution 94
16.4 D. C. Supplies Availability 94
16.5 Earth Strap 94

IN-TEXT FIGURES
A Functional Representation of Matrix 16
B F-Unit and Collate Unit (Bit 1) 30

Appendix 1: MICROPROGRAM
Appendix 2: PROGRAM X3 (FUNCTION TESTS)
Appendix 3: PROGRAM X5 (INTERRUPT TESTS)
Appendix 4: PROGRAM X8 (TRACE)
Appendix 5: PROGRAM X6 (INSTRUCTION TIMING)
Appendix 6 : 903 INITIAL INSTRUCTIONS TEST XINIT

END-OF-TEXT FIGURES

Figure

1	Overall View
2	Control Unit
3	Central Processor Disposition
4	Cable Connections
5	Manual Controls - Keys and Logic
6	Central Timer
7	Control Matrix Function 15
8	Control Matrix Function 14A
9	Control Matrix Function 14
10	Control Matrix Function 13
11	Control Matrix Function 12
12	Control Matrix Functions 7 & 8
13	Control Matrix Functions 5,6 & 9
14	Control Matrix Functions 2,10 & 11
15	Control Matrix Functions 0,1,3 & 4
16	Control Matrix Function Control
17	Matrix Address Selection
18	Matrix Conditions
19	Register System
20	Sample Bit of Registers & Function Unit
21a	Function & Collate Units & G-Register
21b	Function & Collate Units & G-Register
21c	Function & Collate Units & G-Register
22	I,P & J-Registers
23	Waveform Drivers
24	Overflow & Shift
25	Process Counter
26a	Initial Instructions 1
26b	Initial Instructions 2

Figure

27 Input/Output Control

28 Input/Output Peripheral Interface

29 Input/Output

30 Interrupts

31 A. C. Distribution

32 D. C. Distribution

33 Power Control

34 T-L Board Assembly (Control Unit)

35 Board Assemblies

36 Table of L. S. A.'s.

Printed in England by
Engineering Unit, English Electric Computers Ltd.

Section 4.1.2: 903 CENTRAL PROCESSOR

INFORMATION BULLETIN NO. 1

This section of the Technical Manual should be altered in accordance with the information contained in this bulletin. The bulletin should be inserted in this section immediately in front of Page 1.

1. Figure 20 (Issue 2)

   The note at the foot of the figure should read as follows:-

   "Board position (57 - n) Bits 1-9 inclusive
   (55 - n) Bits 10-18 inclusive". ✓

2. Figure 26A (Issue 2)

   (a) The elements C11, C12 & C13 (common output - IIG) should be F11, F12 & F13 respectively. ✓

   (b) The central area of logic is on Board 36 (A-GK). ✓

Chapter 1: INTRODUCTION

1.1 General

1.1.1 Mechanical

The central processor is mounted in 19" rack modules and housed in a standard desk. Figure 1 shows the basic 903B computer.

Figure 3 shows the position of the central processor and its component parts with the operator's control unit mounted on the top of the desk. A back view of the computer is shown in this Figure together with the connector panel (unit 1) to which all items external to the desk are connected. Each of the connectors on this panel is prefixed with the number 1 (e.g. 1-SKT1) to show to which unit it belongs. Where unit numbers are applicable, connectors on that unit are prefixed with the appropriate number (see Figure 4).

The control unit, which is free-standing on the computer desk is shown in Figures 1 and 2. The front panel view (Figure 2) shows the control switches and lamps.

1.1.2 Functional

The central processor is a synchronous unit which operates in parallel on 18-bit words. When these words are being treated as numbers they are regarded as fractional and lying in the range $-1<N<1-2^{-17}$ where N is the fraction.

Negative numbers are held in "two's complement" form such that bit 18 is the sign bit and is a one when the number represented is negative.

The internal store, which holds 8192 18-bit words, is accessed asynchronously.

Input and output takes place through two interfaces; the first carries parallel 18-bit data lines for the paper tape and teleprinter controller only, the second is a general purpose, parallel, 18-bit data highway which can serve up to 2048 peripheral devices. Both interfaces are such that input/output transfers are asynchronous.

The instruction code of the machine is explained briefly in Chapter 7 of this section, but is explained fully in Volume 1.

A block diagram of the central processor registers is shown in Figure 19.

1.2 Logic Boards (Figure 35)

There are two types of logic boards used on the central processor, Standard logic and Matrix. The logic elements contained on the boards are Logic Sub-Assemblies (LSA) and are described fully in Section 4.6.1.

1.2.1 Standard Logic Boards

Standard logic boards consist of a "mother" board which can take fourteen LSA's, and extra discrete components such as "pull-up" resistors and potential dividers.

There are two types of mother board, those with printed circuit connections to the edge-connector and those with wired connections to the edge-connector.

The differences between the standard logic boards is in the disposition and types of LSA and extra components mounted on the "mother" boards.

Typical logic board layouts are shown in Figure 35 and a table of board types with particular LSA's fitted is given in Figure 36.

1.2.2 Processor Matrix Boards (Figure 35)

The central processor matrix boards are all printed circuit "mother" boards with positions for six LSA elements and the remaining area is taken by the diodes of the diode matrix.

1.3 Conventions of Logic Reference

In order that the reader of this manual is able to associate completely, the descriptions of the processor logic on the respective figures, a convention for referring to any logic element is given below.

In any of the "in-text" or "end-of-text" figures in this manual, a logic board is referred to, firstly by numerical position in the logic racks and then by its own letter code which is marked in brackets, e.g. 1(A-FT) is the logic board in the rack position 1 whose letter code is A-FT.

On each logic board, there can be up to fourteen LSA elements and each LSA position is referred to by a letter of area (see Figure 35). An LSA may contain up to three separate logic elements, so reference to a particular logic element is made by a unique pin connection. Thus, when describing logic in the text, the abbreviation 1-A/12 means board position 1, element area A and output pin 12.

In the case of bistables and other elements made of two or more simple gates, a reference may be made thus:- 1-A/11 and 12 where pins 11 and 12 are output pins of the LSA in area A.

Chapter 2: CONTROL UNIT

2.1 Introduction

The control unit which stands on the computer desk provides all the controls necessary to govern the processor. A detailed view of the front panel of the unit showing all the control switches and buttons is illustrated in Figure 2. Two cableforms, from the rear of the control unit (Figure 3), connect it to the processor at connector panel 1.

2.2 Computer Modes

The control unit provides three working modes of the processor, each of which may be selected by the three-position yale-lock switch. These modes of working are:- AUTO, OPERATE and TEST.

2.2.1 Auto Mode

When the processor is in the Auto mode the effective controls on the control unit are On, Off and Reset. This mode is used when the machine is required to work under control of a stored program with connected peripherals and without the need of operator intervention, and may thus be considered as the "on-line, real-time" mode.

2.2.2 Operate Mode

This is the normal working mode for the programmer/operator where data and control information is input or output under operator control. It is the mode which is used for inputting paper tapes by initial instructions.

The effective controls in this mode are as for the auto mode with these additions:-

Stop, Restart, Jump, the N (address) keys of the word generator, and the Program Interrupt switches.

2.2.3 Test Mode

This mode is not a practical working mode, but is especially provided for the maintenance engineer when fault finding in the processor.

All controls on the control unit become effective in this mode.

2.3 Control Unit Circuitry

The control unit circuit diagram has been divided up between Figures 5 and 30 so as to demonstrate the association of the several parts of the unit with the logic they control. Apart from the interrupt control switches, all other switches cause their associated signals to be either at 0 volts (logic ground) or the open-circuit input potential of the LSA element to which they are connected. However, the full consequences of the control unit in each of its modes is covered in Chapters 3, 4 and 17 of this section.

2.4 Audio Unit

Figure 34 shows the control unit lamp driver and audio unit board (T-L), with the disposition of components and a circuit diagram. This board is mounted on the back panel of the control unit.

In the absence of the input signal AUDIO INNER the inverter VT3 is not conducting and VT4 is just held conducting by the potential divider R12, R13 and R14. The small current through VT4 charges C2 until VT5 emitter rises above the gate potential at the junction of R17 & R18. The unijunction (VT5) then fires and discharges C2 until the emitter/gate potential is too low to maintain conduction. VT5 turns off again and C2 is charged by the constant current delivered by VT4. VT5 oscillates, as a result, at a frequency determined by the charging and discharging rates of C2. VT6 is a power amplifier operating in sympathy with VT5. R18 prevents VT6 from overloading the oscillator when VT6 "bottoms". RV1 adjusts the volume of the note from the loudspeaker.

When the signal AUDIO INNER, which is a 100 ns pulse, goes true, VT3 switches on and rapidly discharges C1. VT4 conducts harder and increases the oscillation frequency. When AUDIO INNER goes false again C1 charges towards the potential determined by R12, R13 and R14. The potential divider and capacitor (C1) values are chosen such that the potential at the base of VT4 rises and falls with the input signal pulse repetition frequency (P.R.F) In this manner the degree of conduction of VT4 and hence the charging rate of C2 is modified by the input P.R.F. The result is that the loudspeaker pitch is modified by the P.R.F. of AUDIO INNER.

2.5 Lamp Driver Circuits

The printed circuit board (T-L) mentioned in para.2.4 of this Chapter contains eight identical lamp driver circuits. These lamp drivers provide illumination for the following indicator/switches of the control unit:-

Reset, Stop
Interrupt/Level 1
Interrupt/Level 2
Interrupt/Level 3

The input to each of the respective lamp drivers is taken to the junction of VT1 base and C5 via the board pins given in Figure 30. For a particular lamp to light, the input should be at logic 0, VT2 should be conducting and VT1 cut off. C5 is a voltage spike suppressor and prevents VT1 from being damaged by the same.

Chapter 3: MANUAL CONTROL

3.1 Introduction

The manual control logic of the central processor (Figure 5) provides a logical analysis of the conditions of the control unit switches. It generates a number of signals which govern the operation of the remainder of the system. One of its major functions is to send control signals to the central timer.

3.2 Logic Operation

Since the manner in which the manual control logic behaves is dependent upon the working mode of the computer the operation details are divided into three parts; one for each mode.

3.2.1 Auto Mode Operation

When the computer is switched on, AUTOk is true and sets the bistable 1-N/11 and 12 such that W17 is true. The gates 1-D/12 and 1-L/13 are opened to permit the setting of the Jump bistable. The RC network R6,C1 on board 2 (A-FL) provides an "initial reset" by ensuring that the computer is reset and not ready when the power is first available. As the power supplies reach their correct respective levels PSC (Power Supplies Correct) goes true. PSC together with STC (Store Temperature Correct) generates $\overline{\text{RDY}}$ which sets the Ready bistable and causes $\overline{\text{J1A}}$ to go false.

The reset condition of the computer causes the matrix address C2 to be selected which in turn, makes WFR (Wait For Restart) true. $\overline{\text{J1A}}$ going false fires a 680 ns pulse generator (1-C/13) which sets the Jump bistable for 680 ns and then resets it. The output of the Jump bistable gated with WFR, causes $\overline{\text{SA0}}$ (Set Matrix Address 0) to go false and resets the Reset bistable (i.e. the reset condition is terminated). $\overline{\text{SA0}}$ puts the computer in matrix address C0 where WTMa becomes true. W17 is gated into the M-register at this point.

$\overline{\text{SA0}}$ is false for 330 ns at the end of which the central timer starts. The computer then proceeds through function control (see Appendix 1 titled "MICROPROGRAM") and in the last address of this function the instruction in the I-register is decoded, so that the bit W17, which became M17, produces Function 8 (JUMP). Note that in the AUTO mode the N (address) keys of the word generator are "floating", and because of the way in which they are wired to the M-register, the address 8181 will always be set (see Figure 28).

Depression of the RESET button on the control unit causes $\overline{\text{RESET''}}$ to go false and provided the store is not busy, $\overline{\text{RESET'}}$ goes false and sets the Reset bistable. $\overline{\text{J1A}}$, which has been held true by RESET being false, will go false as soon as the RESET button is released and JUMP is generated causing the process described above to be repeated (i.e. the computer jumps back to the initial starting address 8181).

If the store goes busy at the time when the Reset key is depressed, then SB (Store Busy) goes true and $\overline{\text{SB}}$ goes false, but since element 6-D/11 is a noise rejecting inverter its output does not go false, until approximately 1 ms after its input goes true. Hence, 6-B/12 goes false for about 1 ms on the receipt of SB and prevents a reset being generated until the store has had time to complete a Read and Write cycle.

NOTE: If STC or PSC go false whilst the store is busy, $\overline{\text{SB}}$ inhibits the setting of the RESET bistable.

3.2.2 Operate Mode Operation

AUTOk is false in this mode so that the Jump bistable cannot be set by $\overline{\text{J1A}}$. WG-17k from the JUMP switch (SW2) is at 0 volts holding W17 false. The centre poles of the N (address) keys are at 0 volts.

At switch-on, the computer becomes reset and ready as described in Para. 3.2.1. Depression of the JUMP switch causes JUMPk and WG-17k to go true and $\overline{\text{JUMPk}}$ and $\overline{\text{WG-17k}}$ to go false (since these signals are on a ganged switch), resulting in the Jump and W17 bistables being set. The computer then commences to cycle through function control and extracts and obeys the instruction in the store address specified.

Depression of the STOP button causes $\overline{\text{STOP}}$ to go false and set the Stop bistable so that OIS becomes true, unless the RESTART button is depressed at the same time. RESTART always overrides STOP since $\overline{\text{RESTARTk}}$, which is input to 1-G/13, when false, holds $\overline{\text{STOP}}$ true and prevents the Stop bistable from being set. With OIS true the processor stops cycling at the quiescent address C2 when WFR becomes true (see Chapter 4 of this Section). This means the computer stops at the finish of the current order.

Depression of the Restart button when the computer is stopped, causes a 330 ns positive-going pulse to be generated at 2-C/13 and since $\overline{\text{OS}}$ and $\overline{\text{CS}}$ are both true, the output of 2-E/12 resets the Stop bistable, making OIS false so that the computer can begin cycling again.

Depression of the Reset button returns the computer to the reset and ready state.

NOTE: In both the AUTO and OPERATE modes the Obey and Enter bistables are held reset by $\overline{\text{TESTk}}$ via the inverters 1-H/12 and 1-H/13.

3.2.3 Test Mode Operation

The processor, when switched on, stops in the reset and ready state. $\overline{\text{TESTk}}$ is false and permits the Enter and Obey bistables to be set.

Provided all keys, apart from the N (address) keys of the word generator are in the normal position, depression of the Jump switch has the same effect as in the operate mode (see Para.3.2.2).

When the Order Stop key is depressed, $\overline{\mathrm{OS}}$ goes false so that OSvCS (element 2-G/13 output) goes true as does OIS.

Depression of the Restart key sends a 330 ns positive-going pulse to the input of 2-G/12 causing $\overline{\mathrm{MST}}$ to go false. The timer then runs for one complete order.

If the Cycle Stop key is operated, then $\overline{\mathrm{CS}}$ and $\overline{\mathrm{CS'}}$ go false. This causes $\overline{\mathrm{CIS}}$ to go false and OSvCS to go true, such that the timer stops at the end of the current cycle.

When the Obey key is set to the "SINGLE" position, $\overline{\mathrm{OWG}}$ goes false and sets the Obey bistable and since 2-E/12 output is normally true, 2-L/11 output goes false and generates $\overline{\mathrm{SA0}}$.

If the Obey key is set to the RUN position, $\overline{\mathrm{CINS}}$k goes false and sets the bistable 1-B/11 and 12 so that CIN is true. Since $\overline{\mathrm{OWG}}$ is false and $\overline{\mathrm{STOP}}$ true, every time function control is entered WFR becomes true and $\overline{\mathrm{CON}}$ (Continue) goes false. $\overline{\mathrm{CON}}$ going false fires the 330 ns monostable 2-C/11, P/12 which causes $\overline{\mathrm{SA0}}$ to go false.

The Enter key has exactly the same effect as the Obey key except that the signal WGA (Word Generator to Accumulator) is generated at 2-F/11 output when the Enter bistable is set.

Chapter 4: CENTRAL TIMER

4.1 Introduction

The central timer is a "master clock" under whose control the sequences of all computing events are carried out. It is assembled on board 4 (A-FC) (see Figure 6) and consists essentially of six 100 ns pulse generators in series.

These pulse generators are arranged such that the trailing edge of one pulse generates immediately the leading edge of the next pulse. Once t1' is produced, all six times will be produced so that the smallest controllable step taken by the computer is a cycle step. Several conditions modify the working of the timer and these are discussed fully in the remainder of this Chapter.

4.2 Typical Timer Cycle

When the computer is switched on, $\overline{\text{RESET1}}$ goes false and resets the Timer Inhibit bistable (4-F/12, E/11) and opens the gate 4-L/12. $\overline{\text{SA0}}$ goes false and terminates the reset state as well as causing $\overline{\text{MST}}$ to go true. 330 ns later when $\overline{\text{SA0}}$ goes true again $\overline{\text{MST}}$ (Manual Stop Timer) goes false and starts the timer. t'2 is fed back via 4-F/13 to set the Timer Inhibit bistable. This prevents a second train of pulses being produced until the current cycle is finished. When t'6 goes true a 470 ns monostable (4-D/13, H/11) is fired at the end of which a 100 ns pulse (tZ1) is produced and peripheral selection is permitted. The collating function VTGa and its inverse are gated with tZ1 at elements 4-B/13 and 4-B/12 respectively. If VTGa is true (i.e. the processor is collating and not using the function unit) then the Timer Inhibit bistable is reset and a second train of timing pulses is produced at the start of tZ1). If, however, VTGa is false (i.e. the processor is using the function unit) then a further delay of 330 ns is introduced by the monostable 4-L/13, L/11 before the next train of pulses is produced. The extra delay allows for carry propagation in the function unit.

4.3 Timer Stopping Conditions

Apart from the Timer Inhibit bistable there are several signals fed into the distributed gate.

4-K/11, 12, 13, G/12, 13 and H/12 which can stop the timer.

4.3.1 Order Stop

When the order stop is given to the computer OIS goes true. At the finish of the current order the computer enters the first matrix address of function control (C2), whereupon WFR becomes true, $\overline{\text{CIS}}$ goes false and the timer is stopped.

4.3.2 Wait for Store

Whenever the store becomes busy SB goes true causing $\overline{\text{CIS}}$ to go false so that the timer stops at the end of the current cycle.

4.3.3 Wait for Peripheral

$\overline{\text{Z}}2$ (peripheral reply) is normally true , thus when WFP (Wait For Peripheral) goes true, $\overline{\text{CIS}}$ goes false and the timer is stopped. Input and output instructions (hardware) generate WFP in the matrix when the peripheral select signal is generated. WFP holds up the timer until the peripheral concerned sends a reply which causes $\overline{\text{Z}}2$ to go false and restart the timer.

4.3.4 Set Matrix Address 0

If the processor is cycling and the Jump key is depressed, $\overline{\text{SA0}}$ goes false when the computer enters function control and the timer is stopped at the end of the cycle C2. This allows the matrix address logic to select address C0 so as to obtain the address of the next instruction to be obeyed from the word generator.

4.3.5 Test Mode Stop

When the computer is in the test mode the timer may be stopped in one of two ways other than those already mentioned.

(1) If the Cycle Stop key is depressed then $\overline{\text{CIS}}$ is made false by element 5-D/11 (see Para.4.2.3).

(2) When matrix address C0 is entered (see Para.4.3.4), the inputs to element 4-H/12 go true (i.e. WTMa, TEST and $\overline{\text{SMNk}}$ are true), $\overline{\text{CIS}}$ goes false and the timer is stopped.

4.4 Manually Stepping the Timer

In the test mode, when the timer is stopped, the manual control logic can restart the timer by making $\overline{\text{MST}}$ go false. If either OS or CS is true and a restart is given $\overline{\text{MST}}$ goes false (see Para.3.2.3) and restarts the timer.

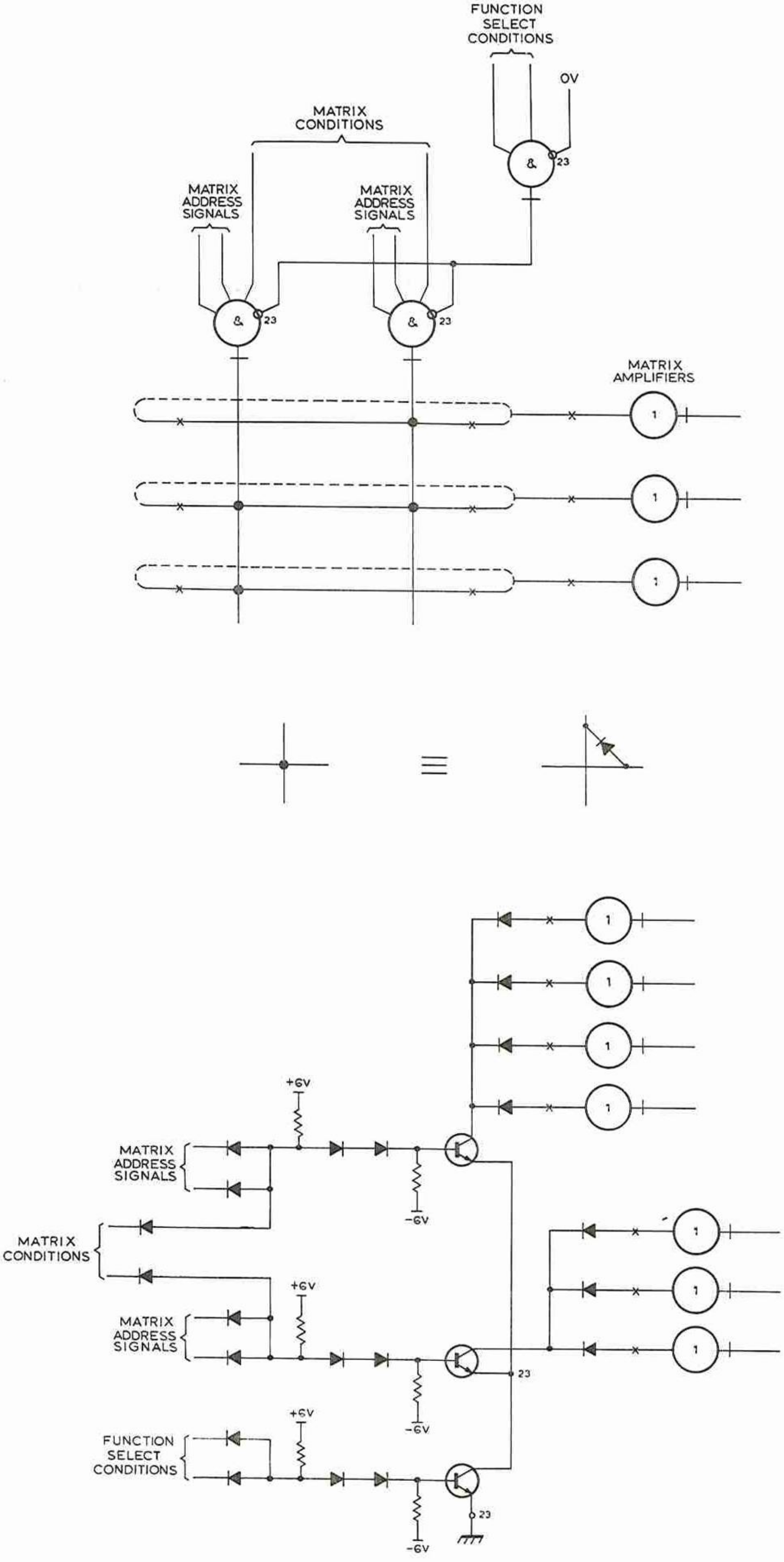

FIG. A.
FUNCTIONAL REPRESENTATION OF MATRIX

Chapter 5: CONTROL MATRIX

5.1 Introduction

The majority of the control waveforms in the 900 are generated within the processor matrix. The matrix proper consists of ten boards, each having 10 Y-ordinates and 28 X-ordinates. Each of the 10 Y-ordinates on any board can select all of the 28 X-ordinates when wired to do so. One Y-ordinate represents one matrix address and the X-ordinates, to which it is connected (via diodes), are the matrix outputs of that address (see Figures 7-16).

Where a matrix output is common to several boards, the X-ordinate pertaining to that output are wired together to form a loop (see Figure A).

On every matrix board, three of the X-ordinates are reserved for "AA" outputs $\overline{\text{AA1m}}$, $\overline{\text{AA2m}}$ and $\overline{\text{AA3m}}$ and are used to select the next matrix address to be entered (i.e. they are microprogram control waveforms).

Selection of any of the Y-ordinates is made by three sets of matrix control waveforms. These are:-

(1) Function Select Conditions (Function Decode Signals)

(2) Matrix Address Line Signals

(3) Matrix Conditions Signals.

The complete processor matrix is located in board positions 6-22 inclusive and the several parts are discussed in the remainder of this Chapter. Figure A demonstrates the construction of the matrix proper.

5.2 Matrix Proper

Each of the ten boards in the matrix proper is arranged to produce the microprogram of one or more complete hardware instructions (see Figures 7-16). The function(s) each board produces is labelled on each of the above mentioned Figures.

To energise a matrix address, each of the inputs to the relative "Nand" gate has to be true except the inhibiting input (pin 23 of every element), which has to be false. When a machine function is decoded those Function Select Conditions go true and remove the inhibits from each matrix address driver in that function.

NOTE: The first 1 or 2 digits at the foot of each Y-ordinate is the hardware function number to which that address belongs.

The matrix address line drivers and the conditions logic then produce the S0, S1, etc. and CA, CB etc. signals in the correct pattern to energise each of the matrix addresses in a function in the correct order. As each address is accessed the matrix outputs (suffixed 'm') of that address go false. Hence each matrix output is marked with a 'bar' (e.g. $\overline{\text{READ}}$m).

Matrix outputs are taken to inverting amplifiers (Matrix Amplifiers) from where some outputs are used directly and others are timed.

5.3 Matrix Address Line Logic (Figure 17)

The matrix address line selection logic is mounted on board 21 (A-FF) and the address line drivers are mounted on board 22 (A-FG). The selection logic consists essentially of two 3-bit registers (bistables), the first of which holds the "next cycle" matrix address and the second holds the "current cycle" matrix address.

At switch on, $\overline{\text{RESET}}$ goes false and sets the "current cycle" register such that $\overline{\text{AA1'}}$ and AA2 are true. This means S2 and $\overline{\text{S4}}$ are true. At the same time, $\overline{\text{RESET}}$ resets the Function Drive bistable such that the function control condition (C) is true. Now S2, $\overline{\text{S4}}$ and C, when true, select the matrix address C2. The matrix output WFR is generated and $\overline{\text{SA0}}$ goes false making $\overline{\text{RESET}}$ go true. The AA2 bistable "turns over" so that $\overline{\text{AA2'}}$ becomes true. The result is that S0, $\overline{\text{S4}}$ and C become true and address C0 is selected. $\overline{\text{SA0}}$ goes true after 330 ns and allows the timer to start. The matrix outputs AA1, AA2 and AA3 are gated into the "next cycle" register in t3-1 (time 3) and from there into the "current cycle" register t6 switching on the appropriate address line drivers.

The signals AA1, AA2 and AA3 generated at each matrix address are output in a binary count, starting with $\overline{\text{AA1}}$, $\overline{\text{AA2}}$ and $\overline{\text{AA3}}$ and arriving at AA1, AA2 and AA3. Skips in the count are made where a function does not require all of the seven possible discrete matrix addresses. However, the last address in each of the functions has AA1, AA2 and AA3 true which in t3 set the AA7 bistable. The signal AA7 then sets the matrix address line logic such that in t6, the signal S2 and $\overline{\text{S4}}$ become true and resets the Function Drive bistable such that the function control select condition (C) becomes true. Hence, the next address selected is C2, the first address in function control (see FLOW DIAGRAM in Appendix 1).

When the Cycle Repeat key is effective and depressed, TCS goes true. This causes 22-G/12 output to go false closing all the gates to the "current cycle" register. TCS is also taken to the matrix conditions logic and function decode logic, so that when TCS is true the current matrix address cannot be changed.

5.3.1 Test Counter for Zero

The matrix address line logic can be modified by the section of logic on board 22 (A-FE) which tests the process counter for zero content (see Figure 18).

In t1 the Test Counter bistable is reset so that $\overline{TC}$ (Test Counter) is true. This removes the inhibits from 21-H/11 and 13 inputs so that AA1 can set the relevant "next cycle" bistable. If C0 (Counter Zero) and TC0 (Test Counter for Zero) signals are true then the Counter Zero and Test Counter bistables will be set in t2. In t3 the respective outputs of the bistables are gated in 22-L/11 and 13 so that C0' goes true and $\overline{C0'}$ goes false. This will cause the address line logic to select an address having either S1 or S2 true, depending on the function being performed.

NOTE: If the Test Counter bistable is not set (i.e. the process counter is not tested for zero content), then the matrix address line logic is not affected by the state of the process counter.

5.4 Function Decode (Figure 17)

The function decode logic is mainly a series of eight 'AND' gates which inspects the contents of the I (Instruction) register and produces the Function Select Conditions (DA to DH) accordingly. Each of the gates is open only when FD (Function Drive) signal is true.

$\overline{RESET}$ going false resets the Function Drive bistable. FD goes false and inhibits a decode of the I-register, and condition C goes true. At the same time the AA7 bistable is reset. Function Control is selected as a result and since SFD (Set Function Drive) does not become true until the last address in this function the decode logic remains unaltered until this time. In t5, SFD and $\overline{TCS}$ cause $\overline{SFD'}$ (7-H/13) to go false and set the Function Drive bistable. FD becomes true in t6 when $\overline{SFD'}$ goes true again and the I-register content is decoded. FD remains true throughout the instruction, but in the last address of the instruction AA1, AA2 and AA3 are arranged to be true, so that in t3 6-J/12 output goes false and sets the AA7 bistable and in t6 the

Function Drive bistable is reset. Condition C becomes true and the computer is returned to Function Control (address C2).

If the Cycle Repeat key is depressed then TCS becomes true and $\overline{\text{TCS}}$ false. This means $\overline{\text{SFD}}$ is held true as is the output of 6-J/12. The result is that the current instruction and hence the current matrix address cannot be changed unless $\overline{\text{RESET}}$ or $\overline{\text{SA0}}$ go false.

5.5 Matrix Conditions (Figure 18)

The matrix conditions logic is mounted on boards 6 and 7 and consists essentially of three sets of distributed gates feeding three bistables.

In some of the hardware instructions of the machine the microprogram can take more than one route (see FLOW DIAGRAM in Appendix 1). The route taken at any time during one of these instructions depends on one or more conditions of certain parts of the processor (e.g. the accumulator is negative or overflow occurs). In these cases two or more matrix addresses will have the same function select conditions and the same matrix address line signals, but entry into such addresses is conditioned. These conditions (CA to CD) are produced by the matrix conditions logic as a result of testing for the afore mentioned processor states.

When the next matrix address to be entered is conditioned, the current address produces the relevant condition test signals. The conditions CD and $\overline{\text{CD}}$ are set in a slightly different manner to CA, CB and CC and are described first. In each of the last addresses of function control (C6b and C7) SFD goes true so that with the next t1 CD is set false and $\overline{\text{CD}}$ true (i.e. the CD bistable is reset). The elements 6-M/11 and E/13 form a "NOT EQUIVALENT" gate of bits 12 and 13 of the M-register.

When TM12M13 is output from the matrix the CD bistable is set in t2 if $M12 \not\equiv M13$; otherwise it will remain reset.

The remaining conditions are set in the following manner:-

RESET goes true initially and resets the two buffer bistables 7-A/12, 13 and 7-C/12, 13. In t5 of the ensuing cycle the result of the conditions test (i.e. TWGA with WGA etc.) is gated through 7-N/12 or 13 to set these buffer bistables accordingly. In t6 the resulting outputs of the buffer bistables set the conditions bistables to CA or $\overline{\text{CA}}$ etc.

NOTE: (1) CC = $\overline{\text{CA v CB}}$

(2) RESET cannot reset the conditions until the next t6.

The buffer bistables are normally reset in t3 of the next cycle in order that the conditions may be reset at the end of that cycle (i.e. in t6). However, if CRS (retain conditions) or TCS is true, the bistables cannot be altered which means that the conditions cannot be altered unless the processor is reset.

5.6 Microprogram

The outputs of the processor matrix are produced in blocks (matrix addresses) and each of these blocks is arranged in a sequence to form the several hardware instructions of the machine. This arrangement of outputs is termed a microprogram and is given in Appendix 1 to this section. The microprogram enables the computer to automatically extract an instruction from store, find the operand to which that instruction applies and obey that instruction. The processor will repeat this process for as long as the stored program demands. Each instruction is obeyed sequentially under control of the sequence control registers (SCR's).

Chapter 6: INSTRUCTION CODE

The computer word consists of 18 bits and can be a data or instruction word, the point in the microprogram at which it is extracted from the store determines which one it is. When the word is an instruction word it is constructed as shown below:-

B	F	N
1 bit	4 bits	13 bits.

The most significant bit (18th bit) is B (address modifier) and whenever it is a 1 the contents of the current level B-register is added to the address bits of the word. Bits 14-17 inclusive are the four function (F) bits and they provide the code for any of the 16 machine functions. Functions 14 and 15 differ from the other functions in that they are sub-divided. Function 14 is sub-divided into function 14 (Shifts) and function 14A (Block Transfer). Function 15 is sub-divided into three instructions; Input, Output and Program Terminate. To select a sub-divided function the following steps are followed:-

The four F bits select function 14 or 15 and the three most significant N bits select the particular sub-divided function. (See the Flow diagram in Appendix 1).

The 13 N bits form the address of the operand to which the instruction applies. This address can be expanded to 16 bits (range 0-65, 535), by the contents of the B-register, when bit 18 is 1. A more detailed description of the instruction code is given in Section 1.2.2 and the machine registers are described in Chapter 7.

Chapter 7: REGISTERS, FUNCTION AND COLLATE UNITS

7.1 Introduction

Figure 19 shows the registers in block diagram form and the relationship between them. Two registers on the machine are not shown because they are not hardware registers but the contents of specific store locations. These two registers are the SCR (Sequence Control Register) and the B (address modifier register).

7.2 SCR and B-Registers

There are four SCR's and four B-registers, one of each for each of the four program levels, and these are located in the store as follows:-

Register	Program Level	Store Location
SCR	1) Highest	0
B	1)	1
SCR	2	2
B	2	3
SCR	3	4
B	3	5
SCR	4) Base	6
B	4)	7

Instructions and data are held in the store in the form of a program and the sequence of extraction is controlled by the SCR. The current level SCR will have a content whose value is that of the next store address required. The SCR is automatically incremented by 1 each time function control is passed through and is set to any required value by the "Jump" functions. The SCR, however,does not itself provide the store address decode logic with the value it requires. The SCR's content is first loaded into the J-register and then an address decode is made.

The B-register is the address modifier register. In certain circumstances an instruction may apply to an address which is greater than 8192 (i.e. in extra store). This requires more than the 13 N-bits allocated to an instruction word. In such cases the 18th bit of the instruction word will be a '1' and this will cause function control to add the content of the current B-register to the address part of the word without affecting the instruction bits (Section 1.2.1).

The address of the required SCR or B-register can be set in the J-register by means of the following signals:-

E, K, ETJ and KTJ (Figure 22).

7.3 J-Register

The J-register is a 16-bit store address register and is used to decode the store address (see Para.7.2). The J-register sets the Process Counter to the required value, using JTC, and replaces the SCR when performing Block Transfer functions.

If the control signals E, K, ETJ and KTJ are not used then the J-register is loaded from the G-register.

7.4 G-Register (Buffer)

The G-register is an 18-bit register and has two uses. The first is to provide a bus so that the contents of any register can be transferred to any other register. The second is to staticise the outputs of the Function Unit (F-Unit). This second requirement is necessary as the F-Unit is usually loaded from the A-register or M-register and the resultant output put back into one of these registers. To ensure that the inputs of the F-Unit are isolated from the output the G-register is provided

The F-Unit outputs are gated into the G-register in three ways:-

(1) Directly with FTG.

(2) Shifted up (left) one place with LTG .

(3) Shifted down (right) one place with RTG.

7.5 A and Q-Registers (Accumulator)

The A-register is an 18-bit register and is used with the M-register and F-Unit for calculations. The result of the calculations is fed back into the A-register. Data from the A-register can also be output direct to a peripheral. The Q-register is an 18-bit auxiliary register and is used with the A-register for double-length arithmetic.

7.6 M-Register (Store Access)

The M-register is a multi-purpose register and is involved in all machine functions. The main purpose of the register is to act as a buffer between the store and the processor. Words held in the register may be Instruction words or Data words, the type of word depends on where the machine is in the microprogram.

If it is an Instruction word bits 14 to 17 of the register are gated into the I-register with MTI and the thirteen address bits are routed either into the J-register, when the store is the recipient of the instruction, or into the P-register when the peripheral is the recipient of the instruction.

The Data word may be written into the store from the M-register if a calculation is completed, or transferred to the F-Unit if calculations are about to begin.

7.7 I-Register (Functions)

The processor has 15 functions and to select any particular function the I-register is set to the function number. The number is set into the I-register by the Instruction word.

7.8 P-Register (Peripheral)

The P-register is a 13-bit register and is used to hold the address of the peripheral to which the instruction applies. All peripherals are identified by their address and the P-register supplies this code to the peripheral interface.

7.9 Function Unit (F-Unit)

The F-Unit is an 18-bit parallel adder having two groups of 18 inputs (X and Y) and a control signal 1TF. The signal 1TF is equivalent to Carry 0 and adds a '1' to the least significant bit of the adder when it is true. For the F-Unit to behave as an adder the conditions shown below must be satisfied:-

Xn	Yn	Cn-1	Fn	Cn
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

LOGICAL ADDER TRUTH TABLE

7.9.1 Addition

Addition is executed in the F-Unit by energising the X and Y-inputs with the contents of the M- and A- or Q-registers respectively.

A number held in the A-register is to be added to a number held in the M-register which was extracted from the store. MTF gates the M-register content to the X-inputs of the F-Unit and ATF gates the number in the A-register to the Y-inputs of the F-Unit resulting in the sum of the two numbers at the F-Unit output.

7.9.2 Subtraction

The F-Unit cannot subtract directly so the number which is to be subtracted is placed in the M-register and $\overline{\text{MTF}}$ with 1TF gates the 2's complement of this number to the X-inputs of the F-Unit. Addition is performed as described in para.7.9.1, with the result that X is subtracted from Y (i.e.-X + Y = Y - X).

There are two control signals DTF1 and DTF2 which together perform the opposite function of 1TF. DTF1 and DTF2, when true, make every Y-input to the F-Unit a logic 1 resulting in the total Y-input being equivalent to -1. Hence, a number held in the M-register can be decremented by 1 if it is passed through the F-Unit with the signals DTF1 and DTF2 true.

7.9.3 Summation Rules

Figure B shows bit 1 of the F- and Collate units.

The sum of the inputs X and Y is in the general case:-

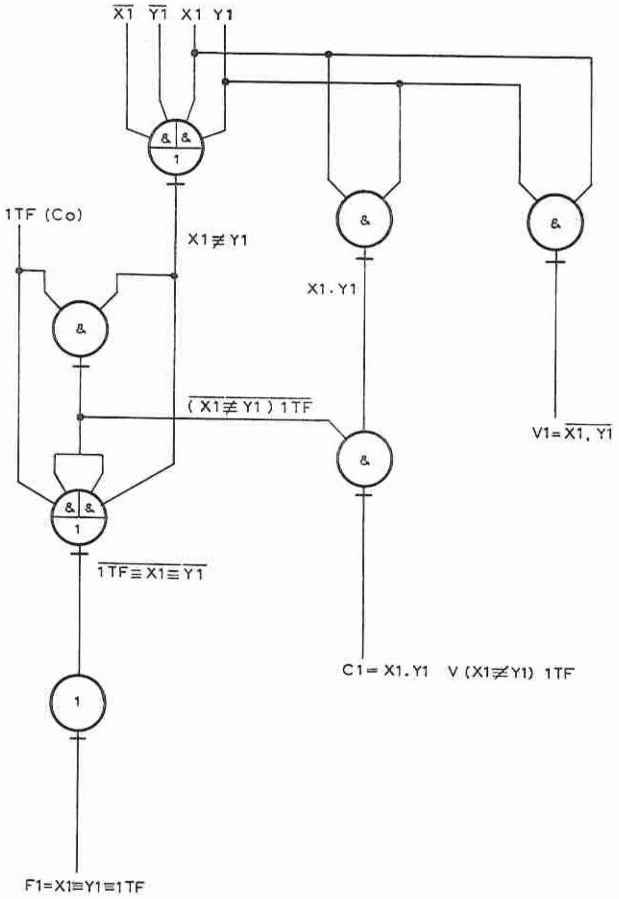
$\overline{X1}$ $\overline{Y1}$ X1 Y1
& & 1
&
&
1TF (Co)
$X1 \not\equiv Y1$
X1 . Y1
&
&
$\overline{(X1 \not\equiv Y1)\ 1TF}$
&
$V1 = \overline{X1}.\ \overline{Y1}$
& & 1
$\overline{1TF \equiv X1 \equiv Y1}$
C1 = X1 . Y1 V $(X1 \not\equiv Y1)$ 1TF
1
$F1 = X1 \equiv Y1 \equiv 1TF$

FIG. B
F–UNIT AND COLLATE UNIT (BIT I)

$$Fn = Xn \not\equiv Yn \text{ when } Cn\text{-}1 \text{ is zero,}$$

and $$Fn = Xn \equiv Yn \text{ when } Cn\text{-}1 \text{ is one.}$$

$$\therefore \quad Fn = (Xn \equiv Yn)Cn\text{-}1 \vee (Xn \not\equiv Yn)\,\overline{Cn\text{-}1}$$

$$= Xn \equiv Yn \equiv Cn\text{-}1.$$

or as in the figure

$$F1 = X1 \equiv Y1 \equiv 1TF\ (C0).$$

7.9.4 Carry Rules

Cn (Carry n) has to be a '1' whenever Xn and Yn are ones or whenever Cn-1 is a one together with Xn or Yn.

$$\therefore \quad Cn = Xn\ Yn \vee (Xn \vee Yn)\ Cn\text{-}1.$$

or as in the figure

$$C1 = X1\ Y1 \vee (X1 \vee Y1)\ 1TF.$$

7.10 Collate (logical AND NOT) Unit

The Collate unit consists of 18 AND NOT gates which collate the inputs X1-18 with Y1-18 of the F-Unit into the G-register. Since the collate unit feeds the G-register directly the appropriate bit of the G-register is set when the output of the collate gate is false.

Therefore, Gn is true when Xn and Yn are true if the collate permit signal VTG is true and the digits Xn and Yn are truly collated in the G-register.

$$\therefore \quad Gn = \overline{Vn} = Xn.Yn.$$

or $$G1 = \overline{V1} = X1.Y1.$$

A common use of the collate unit is as the F-Unit by-pass, such that the outputs of the J- or M-registers can be copied into the G-register.

MTF gates the output of the M-register to the X inputs of the F-Unit and Collate-Unit. DTF1 and DTF2 cause Y1-13 and Y14-18 respectively to become logic '1's. VTG opens the Collate-Unit and the X inputs are copied into the G-register. DTF1 alone is used for collating out the address part of an instruction held in the M-register so that it may be placed in the J-register. Hence, MTF, DTF1, VTG and GTJ take the thirteen less significant bits of the M-register and collate them into the G-register from where they are gated into the J-register with GTJ.

Chapter 8: PROCESS COUNTER

8.1 Introduction

The process counter consists of twelve bistables arranged as a binary counter register. It is mounted on boards 23, 24 and 25 (A-FP) (see Figure 25). Wherever a function has a repeated action (i.e. an instruction loop) such as multiplication, division and block transferring, the process counter records the number of times the loop is to be cycled and provides the condition for leaving such a loop, when that number of cycles is completed.

It may be loaded directly with the content of the store address register by the signal JTC. The counter content is decremented by one with $\overline{\text{1TC}}$ and is reset with WFR (Wait For Restart).

The process counter has three logic outputs and these are:-

C(1-4)1, C(1-4)0 and C(5-12)0.

C(1-4)1 and C(5-12)0 are used to form the signal LW (Last Word) in block transfers. C(1-4)0 and C(5-12)0 are gated together in an AND gate to form the counter zero signal (C0).

Each of the twelve counter bits has a monitor output to socket 1-SKT2 such that the content of the process counter may be observed at any time when using the display unit.

Since the process counter is loaded from the inverse outputs of the J-register (i.e. the binary complement of the number in J), the counter contains 'full-house' when reset so that loading the counter from J actually subtracts the complement of the required value from "full-house". In this manner the process counter content becomes equal to the true content of the J-register.

8.2 Logic Description (Figure 25)

When the general reset is given to the computer from the manual control logic, the processor accesses the quiescent matrix address C2 whereupon WFR becomes true. WFR is input to elements 23, 24 and 25-L/13 and when true, the outputs of these elements are false. This resets the process counter bistables such that $\overline{C1}$-$\overline{C12}$ inclusive are logic '0'.

JTCm is produced by the control matrix which in t5, makes JTC true. This signal opens gates H/13, C/13, M/13 and G/13 on each of the three boards. The inverse outputs of the J-register ($\overline{J1}$, $\overline{J2}$ etc.) are input to the counter through these gates setting each of the counter bits equal to the corresponding J bit (i.e. $\overline{C1} = \overline{J1'}$ etc.).

The counter content is then reduced by 1 with $\overline{1TC}$. Considering the first bit of the counter ($\overline{C1}$) the bistable A/13, H/11 has the monostables J/11, A/12 and J/13, H/12 in 'collector logic', with the respective elements of itself. $\overline{1TC}$ is a common input to both monostables. If $\overline{C1}$ is true, then the output 2 of J/11 is true and input 5 J/13 is false. When $\overline{1TC}$ goes false the monostable J/11 will fire and the other monostable will remain unaffected. A/12 output goes false and the bistable 'turns over' so that $\overline{C1}$ becomes false. Hence if the original content of the counter was:-

$\overline{C12}$ to $\overline{C1}$
111111111110 (content = 1)

it now becomes

$\overline{C12}$ to $\overline{C1}$
111111111111 (content = 0)

and the counter content is reduced by one.

When $\overline{1TC}$ goes true again, it has no effect on the monostables J/11 and J/13 so that the counter is unaffected. Each stage of the counter is identical but the monostable common inputs come from the outputs of the previous 'bits'. Hence, any 'bit' of the process counter is only affected by the previous 'bit' if the previous 'bit' changes from a logic '1' to a logic '0'.

Elements L/11 and 12 on each of the boards form a 4-input AND gate of the four process counter bits on their respective boards, e.g. $\overline{C1}$-$\overline{C4}$ are input to element 23-L/13 and when they are true C(1-4)0 is true (pin 20).

Pin 20 of boards 24 and 25 are connected together such that all eight bits ($\overline{C5}$-$\overline{C12}$ inclusive), have to be true before C(5-12)0 is true.

$\overline{C1}$-$\overline{C4}$ are taken from board 23 to the AND gate 22-N/11 and 12 to produce C(1-4)1. $\overline{C1}$ is inverted by element 22-N/13 before being input to 22-N/12. C(1-4)1 is true, therefore, when $\overline{C1}$ is false and $\overline{C2}$, $\overline{C3}$ and $\overline{C4}$ are true.

Chapter 9: CONTROL INSTRUCTION

9.1 Initial Start

Whichever mode the computer is in when switched on, the matrix address C2 is selected and as the timer is not running signals, $\overline{\text{OTM}}$, $\overline{\text{OTJ}}$, ETJ and READ which are all timed signals, are not produced. Therefore, the significant outputs of this address in 'initial start' are DTE and WFR.

9.1.1 Auto Mode Start

(1) Address C2

DTE establishes the program level which in this case, is level 1. WFR, together with JUMP in the manual control logic, causes $\overline{\text{SA0}}$ to go false. The processor then immediately energises the matrix address C0.

(2) Address C0

DTE has no effect since the program level has been established. TWGA has no effect since the machine is in Auto and will obey the word generator. After $\overline{\text{SA0}}$ has gone true and the timer commences running, $\overline{\text{OTJ}}$ clears the J-register and $\overline{\text{OTM}}$ and WTM place the word generator contents in the M-register. Address, C1B is then energised.

(3) Address C1B

This address has no effect other than to energise the matrix address C5. The remaining microprogram addresses are discussed in Para.9.3 of this Chapter.

9.1.2 Operate and Test Mode Start

The effect of an operate or test mode start is the same as for auto with three exceptions:-

(1) TWGA becomes effective if the computer is in the test mode (see Para.9.2 of this Chapter).

(2) The computer will not transfer control from address C2 to C0 until JUMP is effected manually at the control unit.

(3) Whilst the computer is in address C2 (i.e. in a reset state) the INTERRUPT switches may be pressed causing a demand for a program level change. DTE in address C0 inhibits the effect of these keys in 'initial start' so that the program must remain at level 1.

9.2 Manual Entry Loop (Test Mode Only)

Operation of the Enter key on the control unit when the computer is switched on in the test mode will cause $\overline{\text{SA0}}$ to be generated and the matrix address C0 to be energised. DTE, $\overline{\text{OTM}}$, $\overline{\text{OTJ}}$ and WTM have the same effect as before, but TWGA with WGA from the manual control logic (the latter is generated with ENTER) causes the computer to energise address C1A.

Address C1A

$\overline{\text{OTG}}$, MTF and FTG pass the content of the M-register through the F-Unit into the G-register. $\overline{\text{OTA}}$ and GTA transfer the contents of the G-register to the A-register. Hence, in address C0 and C1A the content of the word generator is entered into the accumulator. Control is then given back to address C2.

9.3 Normal Cycle of Control Instruction

In the description the computer is assumed to have passed through initial start and obeyed an instruction, whereupon it is returned to the control instruction at address C2 (see FLOW DIAGRAM, Appendix 1).

(1) Address C2

$\overline{\text{OTM}}$ and $\overline{\text{OTJ}}$ clear the M- and J-registers respectively such that DTE and ETJ set the J-register with the address of the appropriate SCR (i.e. the SCR of the current program level). When the machine is running and the Jump key is not pressed, WFR has no effect.

READ extracts the relevant SCR from the store.

(2) Address C3

$\overline{\text{OTG}}$, MTF, 1TF, FTG, $\overline{\text{OTM}}$, GTM increment the content of the M-register by one (i.e. the SCR is incremented), and WRITE replaces the incremented SCR to the store.

NOTE: The M-register now holds the address of the next but one instruction. Therefore, the content of M must be decremented by one to restore the required address.

(3) Address C4

$\overline{\text{OTG}}$, MTF, DTF1, DTF2, FTG take the content of the M-register, subtract one from it and place the result in the G-register. $\overline{\text{OTJ}}$ and GTJ place the required store address in the J-register. $\overline{\text{OTM}}$, READ and WRITE cause the content of that store address to be read into the M-register and re-written into the same store location.

(4) Address C5

TM18 tests for the B (address) modifier by inspecting the 18th bit of M. If M18 = 0, then there is no address modification.

(5) Address C6B (No B-modification)

$\overline{\text{OTI}}$ and MTI gate the instruction part of the word in the M-register into the I-register. $\overline{\text{OTG}}$, MTF, DTF1 and VTG copies the thirteen address bits of the word in the M-register into the G-register and $\overline{\text{OTJ}}$ clears the J-register. $\overline{\text{OTM}}$ and GTM then restores the M-register with the thirteen less significant bits of the original word only. SFD sets the Function Drive Bistable and permits entry to the machine instruction given by the content of the I-register.

(6) Address C6A (B-modification)

$\overline{\text{OTI}}$, MTI, $\overline{\text{OTG}}$, MTF, DTF1, VTG do exactly the same as in address C6B.

$\overline{\text{OTM}}$, $\overline{\text{OTJ}}$ and QTQ clear the M, J and Q-registers respectively. GTQ places the content of the G-register into the Q-register such that Q now contains the original address part (less significant 13 bits) of the M-register. K1 and ETJ set the J-register with the address of the relevant B-register. READ and WRITE extract that B-register.

(7) Address C7

$\overline{\text{OTG}}$, MTF, QTF and FTG add the contents of the M- and Q-registers and places the result in the G-register. The G-register now contains the modified address. $\overline{\text{OTJ}}$ clears the J-register. $\overline{\text{OTM}}$ and GTM move the modified address from the G-to the M-register. SFD permits a function decode as described in address C6B.

In addresses C6B and C7 the address of the operand is held in the M-register and not in the J-register, since Functions 14, 14A and 15 require inspection of the M-register initially. Furthermore, Function 15 (INPUT/OUTPUT) sets the peripheral address register (P) from M with MTP (see MICROPROGRAM).

Chapter 10: OVERFLOW AND SHIFT UNIT

10.1 Introduction

Figure 24 shows a logic and schematic presentation of the overflow and shift system. Several conditions which arise during computation, require the use of this section of logic. Such instances are:-

(1) When the sum of two numbers, added together in the F-Unit, is "out of range" (i.e. a 19th digit position is required to hold the sum) and normal overflow occurs.

(2) During the performance of multiplication, division or shifting.

A detailed description of the uses and logical action is provided in this Chapter.

10.2 Uses of Overflow and Shift System

10.2.1 Overflow Unit Proper

The overflow unit consists in essence of two bistables (X and Y) and a special gate. The special gate comprises elements 57-K/11, 13 and H/11 and has two purposes. The first is to provide the regeneration of a sign bit when right-shifting during multiplication (see examples in Chapter 14). The second and more usual use is to gate the content of the X-bistable of the overflow unit into the most significant bit of the G-register during a right-shift. Elements 57-G/13 and H/13 gate the overflow X-bistable content into the least significant bit of the G-register in a left-shift. The Y-bistable is set by F1 when right-shifting or by F18 when left-shifting or by the true sign bit (Z18) when directly transferring from the F-Unit to the G-register. The X-bistable is set to the content of the Y-bistable, when demanded, later in the same cycle.

Because there are two independent bistables in the overflow unit, two digits, derived from separate sources, may be held at the same instant. An example of this is encountered in multiplication.

The X-bistable may hold the partial product digit being moved from the least significant bit of the A-register to the most significant bit of the Q-register whilst the Y-bistable holds the multiplier digit which has just been shifted out of the Q-register.

10.2.2 Normal Overflow and True Sign

Elements 5-J/12 and 13 and 5-B/12 and 13 are used in two ways. The first is to set the Y-bistable of the overflow unit when two numbers added in the F-Unit give a result more negative than -1. The other use is to regenerate the sign bit of a negative number which is right-shifted into G, and is used in conjunction with elements 57-K/11 and 13.

10.3 Shift and Overflow Logic Description

Elements 5-J/12 and 13 form an EQUIVALENCE element as do elements 5-B/12 and 13. The output Z18 is true

whenever $C17 \equiv C18$ and F18 is true
or $C17 \not\equiv C18$ and F18 is false
i.e. $C17 \equiv C18 \equiv F18$.

In t1 $\overline{\text{OTG}}$ goes false and resets the Y-bistable. The output of the distributed gate sets the Y-bistable when false. Therefore, F1 and RTG or Z18 or F18 and LTG when true, set the Y-bistable. $\overline{\text{OTX}}$ goes false in t3 and resets the X-bistable. YTX in t4 sets the X-bistable to the content of the Y-bistable and XTF makes X0 and X1 the same logic state as the X-bistable (i.e. X0 and X1 are true when the X-bistable is set and false when the X-bistable is reset).

If XTF is false however, then $X1 \equiv Z18$ which is the condition for sign regeneration when right-shifting.

Chapter 11: SHIFT, MULTIPLY AND DIVIDE

11.1 Introduction

This chapter details the manner in which the functions shift, multiply and divide operate in the central processor. Examples are given in each case and numbers are treated as fractions.

Since each of the above mentioned functions employs the shift and overflow unit, they have been grouped here for description.

11.2 Shift (Fn14)

In function 14 the computer will shift the contents of both the A- and Q-registers left or right, the number of places specified up to a maximum of 8191 places.

For the computer to enter function 14, M12 ≡ M13 otherwise function14A (block transfer) will be entered (i.e. when M12 ≢ M13) (see Chapter 5 of this section).

If M12 = M13 = 1, the computer performs right-shifting. If M12 = M13 = 0, the computer performs left-shifting (see Flow Diagram, Appendix 1).

A number is doubled each time it is shifted left one place, and halved each time it is shifted right one place, e.g. when right-shifting.

If c(A) = 0.1000 ($+^1/_2$) successive right-shifts
give 0.0100 ($+^1/_4$), 0.0010 ($+^1/_8$) etc.

Right-shifting a negative number requires that the sign bit be regenerated with each shift.

e.g. 1.1000 ($-^1/_2$), 1.1100 ($-^1/_4$),1.1110 ($-^1/_8$) etc.

When left-shifting, negative numbers will automatically have their true sign regenerated if they are to remain in range.

e.g. 1.1110 ($-^1/_8$), 1.1100 ($-^1/_4$), 1.1000 ($-^1/_2$)

1.0000 (-1) No further shifts are valid since the machine cannot represent numbers more negative than -1.

The 13 N-digits of this instruction are used to set the process counter with the number of shifts to be performed. When the word is to be left-shifted, the process counter is set directly to the value of the 13 N-bits. When the word is to be right-shifted however, the N (address) bits are subtracted from 8192 and the process counter is set to the result. The reason for this is evident from the requirements of 12th and 13th address (N) bits given in paragraph 11.2 of this Chapter.

The justification in the subtraction of the address bits is as follows:-

Let the N-bits originally held in the M-register be

1111111111111 (= 8191)

This constitutes a one place right-shift.

$\overline{\text{M}}$TF puts 0000000000000 into the F-unit,

and 1TF puts 0000000000001 into the F-unit.

FTG, GTJ and JTC loads the process counter with

0000000000001 (content = 1) which is the number of places to be shifted.

11.3 Examples of Shift

11.3.1 Left-Shift

Let the number in the A-register be 0.0011 and the number in the Q-register be 0.1011

As a double-length number 0.001101011 (= + $\frac{107}{512}$)

For a left shift two places, the microprogram proceeds as follows:-

Address	Overflow Y	 X	A-Register	Q-Register	Counter Content	Comments
142B GTJ,JTC	0	0	0.0011	0.1011	2	Set up count
144B QTF,LTG, YTX,GTQ, 1TC	0	0	0.0011	1.0110	1	Left Shift Q and collect overflow. Decrement counter.
146B ATF,XTF, LTG,GTA, TC0	0	0	0.0110	1.0110	1	Left shift A putting carry-over digit from Q in A. Test counter for 0
144B QTF,LTG, YTX,GTQ, 1TC	1	1	0.0110	0.1100	0	Set overflow with "carry-over" digit from Q to A. Decrement count.
146B ATF,XTF, LTG,GTA, TC0	0	1	0.1101	0.1100	0	Left shift A collecting carry-over digit.
TC0 145						O Exit loop. EXIT FUNCTION.

The shifted result is 0.1101011 $(+\frac{107}{128})$ which is four times greater than the original number and is correct for a two-place left-shift.

11.3.2 Right Shift

Let the number in the A-register be 1.1000 and the number in the Q-register be 1.0100	As a double-length number 1.1000101 $(= \frac{59}{128})$

For a right shift two places the microprogram proceeds as follows:-

Address	Overflow Y	X	A-Register	Q-Register	Counter Content	Comments
142A MTF, 1TF, FTG, JTC, GTJ	0	0	1.1000	1.0100	2	Set up count
144A ATF, RTG, YTX, GTA, 1TC	0	0	1.1100	1.0100	1	Right shift A propogating Sign bit. Decrement count.
146A QTF, RTG, XTF, GTQ, TC0	0	0	1.1100	0.1010	1	Right shift Q. Collect carry-over digit from A. Test counter for zero.
144A ATF, YTX, RTG, GTA, 1TC	0	0	1.1110	0.1010	0	Right shift A propogating Sign bit. Decrement count.
146A QTF, XTF, RTG, GTQ, TC0	0	0	1.1110	0.0101	0	Right shift Q. Counter Zero Exit loop.
145						EXIT FUNCTION.

Set up first matrix address of control function. The shifted result is 1.111000101 ($-\frac{59}{512}$) which is one quarter of the original number and is correct for a two place right shift.

NOTE: The matrix addresses 143B and 143A have not been included in the examples as they are provided only to prevent a shift of zero places (14-0), which is a valid instruction, from passing through the complete microinstruction. In the event of such an instruction, the above mentioned addresses provide a direct exit from the instruction.

11.4 Function 12 (Multiply)

The multiplication function consists essentially of a closed loop of instructions which perform addition, subtraction and/or right-shifting one place at a time, in accordance with the rules of multiplication which are given later in this paragraph. The two numbers involved in a multiplication process are the multiplicand and the multiplier. When multiplied in parts, they form a partial product, and the sum of the several partial products is the final product.

The multiplier, which is initially held in the A-register is transferred to the Q-register, and the multiplicand is extracted from the store and placed in the M-register. The partial products are added together, as they are formed, and placed in the A-register. With every step of multiplication the contents of the A- and Q-registers are right-shifted until the multiplier is completely removed and the final product lies in the A- and Q-registers (A being the more significant part of the double-length result).

The method of multiplication is to compare the least significant multiplier digit (the digit in position Q1) with the digit of less significance (which may be considered as the digit in position Q0 and is held in the

X-bistable of the overflow unit (see Figure 34). The result, of which there are three possibilities, determines the use to be made of the multiplicand and is expressed in terms of the following rules:-

Rule 1

If the given multiplier digit is a one and the less significant digit a zero, subtract the multiplicand from the partial product which is initially zero.

Rule 2

If the given multiplier digit is a zero and the less significant digit a one, add the multiplicand to the partial product.

Rule 3

If both digits are alike, do nothing.

11.4.1 Worked Examples

In each of the following worked examples, reference should be made to the Chapter 8, in order to realise the timing of the matrix signals and the conditioning of the microprogram flow which produces multiplication (Function 12).

NOTE: Single length multiplication (integer working) leaves the correct answer held in the least significant bit of the A-register and the more significant bits of the Q-register, i.e. for two 18-bit numbers being multiplied together the correct result is A1,Q18.....Q2 where A1 represents the sign of the answer. The result must be shifted left 17 places to get it all into A. This shift is not done by Fn12.

11.4.1.1 Example 1 (Positive, Multiplier and Multiplicand)

Let the multiplier be 0.1101 (+ $\frac{13}{16}$) held in the A-register.

Let the multplicand be 0.1011 (+ $\frac{11}{16}$) held in the M-register.

Matrix Outputs	Register	Underflow	Remarks
Address 120			
MTF, FTG, GTJ,			Transfer address of operand from M to J. Extract operand.
Read/Write	M = 01011		
Address 121			
ATF, FTG, GTQ,	Q = 01101		Transfer contents of A to Q. Set process counter to 4. Compare overflow digit with Q1.
KTJ, JTC, TXQ1			
Address 122B			
($\overline{\text{X}}$Q1)			
ATF	00000		Subtract multiplicand from partial product and right shift propagating sign bit.
$\overline{\text{M}}$TF	10100		
1TF	1		
	F = 10101		collect underflow.
RTG	G = 11010	Y = 1	
YTX, GTA	A = 11010	X = 1	
1TC	Count = 3		Decrement counter.
Address 124			Right shift Q picking up carry-over from A. Collect underflow.
QTF	01101		
XTF	1		
RTG	G = 10110	Y = 1	
YTX, GTQ	Q = 10110	X = 1	
TC0, TXQ1			Test counter for zero and compare signs.

Matrix Outputs	Register	Underflow	Remarks
Address 122A			
$(X\overline{Q1})$			
ATF	11010		Add multiplicand and
MTF	01011		partial product. Right
	F = 00101		shift collecting underflow.
RTG	G = 00010	Y = 1	
YTX, GTA	A = 00010	X = 1	
1TC	Count = 2		Decrement counter.
Address 124			
QTF	10110		
XTF	1		
RTG	G = 11011	Y = 0	
YTX, GTQ	Q = 11011	X = 0	
TC0, TXQ1			
Address 122B			
$(\overline{X}Q1)$			
ATF	00010		
$\overline{M}$TF	10100		
1TF	1		
	F = 10111		
RTG	G = 11011	Y = 1	
YTX, GTA	A = 11011	X = 1	
1TC	Count = 1		
Address 124			
QTF	11011		
XTF	1		
RTG	G = 11101	Y = 1	
YTX, GTQ	Q = 11101	X = 1	
TC0, TXQ1			
Address 122C			
$(X \equiv Q1)$			
ATF	F = 11011		Right shift A propogating
RTG	G = 11101	Y = 1	sign bit. Collect underflow
YTX, GTA	A = 11101	X = 1	
1TC	Count = 0		Decrement counter.

Matrix Outputs	Register	Underflow	Remarks
Address 124			
QTF	11101		
XTF	1		
RTG	G = 11110	Y = 1	
YTX,GTQ	Q = 11110	X = 1	
TC0			Counter zero.
Address 123A			
$(X\overline{Q1})$			
ATF	11101		Add multiplicand and partial product and place in A.
MTF	$\underline{01011}$		
	F = 01000		
FTG	G = 01000		
GTA	A = 01000		

The A-register holds the more significant part and the Q-register the less significant part of the result. Hence, answer obtained is 0.100011110 = $+\frac{143}{256}$ which is correct.

11.4.1.2 Example 2 (Positive Multiplier and Negative Multiplicand)

Let the multiplier be 0.1101 $(+\frac{13}{16})$ held in the A-register.

Let the multiplicand be 1.0101 $(-\frac{11}{16})$ held in the M-register.

Matrix Outputs	Register	Underflow	Remarks
Address 120			
MTF,FTG,GTJ, Read/Write	M = 10101		Transfer address of operand from M to J. Extract operand.
Address 121			
ATF,FTG,GTQ, KTJ,JTC TXQ1	Q = 01101		Transfer contents of A to Q. Set process counter to 4. Compare overflow digit with Q1.

Matrix Outputs	Register	Underflow	Remarks
Address 122B			
($\overline{\text{X}}$Q1)			
ATF	00000		Subtract multiplicand from partial product and right shift.
$\overline{\text{M}}$TF	01010		
1TF	1		
	F = 01011		
RTG	G = 00101	Y = 1	Collect underflow.
YTX, GTA	A = 00101	X = 1	
1TC	Count = 3		Decrement counter.
Address 124			
QTF	01101		Right shift Q picking up carry-over from A.
XTF	1		
RTG	G = 10110	Y = 1	Collect underflow.
YTX, GTQ	Q = 10110	X = 1	
TC0, TXQ1			Test counter for zero and compare signs.
Address 122A			
(X$\overline{\text{Q1}}$)			
ATF	00101		Add multiplicand and partial product. Right shift collecting underflow.
MTF	10101		
	F = 11010		
RTG	G = 11101	Y = 0	
YTX, GTA	A = 11101	X = 0	
1TC	Count = 2		Decrement counter.
Address 124			
QTF	10110		
XTF	0		
RTG	G = 01011	Y = 0	
YTX, GTQ	Q = 01011	X = 0	
TC0, TXQ1			
Address 122B			
($\overline{\text{X}}$Q1)			
ATF	11101		
$\overline{\text{M}}$TF	01010		
1TF	1		
	F = 01000		
RTG	G = 00100	Y = 0	
YTX, GTA	A = 00100	X = 0	
1TC	Count = 1		

Matrix Outputs	Register	Underflow	Remarks
Address 124			
QTF	01011		
ZTF	0		
RTG	G = 00101	Y = 1	
YTX, GTQ	Q = 00101	X = 1	
TC0, TXQ1			
Address 122C			
(X ≡ Q1)			
ATF	F = 00100		Right shift A and collect overflow.
RTG	G = 00010	Y = 0	
YTX, GTA	A = 00010	X = 0	
1TC	Count = 0		Decrement counter.
Address 124			
QTF	00101		
XTF	0		
RTG	G = 00010	Y = 1	
YTX, GTQ	Q = 00010	X = 1	
TC0			Counter zero.
Address 123A			
(X$\overline{Q1}$)			
ATF	00010		Add multiplicand and partial product and place in A.
MTF	$\underline{10101}$		
	F = 10111		
FTG	G = 10111		
GTA	A = 10111		

Result obtained is 1.01110001 = $-\frac{143}{256}$ which is correct.

11.4.1.3 Example 3 (Negative Multiplier and Positive Multiplicand)

Let the multiplier be 1.0011 ($-\frac{13}{16}$) held in the A-register.

Let the multiplicand be 0.1011 ($+\frac{11}{16}$) held in the M-register.

Matrix Outputs	Register	Underflow	Remarks
Address 120			
MTF,FTG,GTJ			Transfer address of operand from M to J. Extract operand.
Read/Write	M = 01011		
Address 121			
ATF,FTG,GTQ	Q = 10011		Transfer contents of A to Q.Set process counter to 4. Compare digit with Q1.
KTJ,JTC			
TXQ1			
Address 122B			
($\overline{X}$Q1)			
ATF	00000		Subtract multiplicand from partial product and right shift propagating sign bit. Collect underflow.
$\overline{\text{MTF}}$	10100		
1TF	1		
	F = 10101		
RTG	G = 11010	Y = 1	
YTX,GTA	A = 11010	X = 1	
1TC	Count = 3		Decrement counter.
Address 124			
QTF	10011		Right shift Q picking up carry-over from A. Collect underflow.
XTF	1		
RTG	G = 11001	Y = 1	
YTX,GTQ	Q = 11001	X = 1	
TC0,TXQ1			Test counter for zero and compare signs.
Address 122C			
(X ≡ Q1)			
ATF	F = 11010		Right shift A propagating sign bit. Collect underflow.
RTG	G = 11101	Y = 0	
YTX,GTA	A = 11101	X = 0	
1TC	Count = 2		Decrement counter.
Address 124			
QTF	11001		
XTF	0		
RTG	G = 01100	Y = 1	
YTX,GTQ	Q = 01100	X = 1	
TC0,TXQ1			

Matrix Outputs	Register	Underflow	Remarks
Address 122A			
($X\overline{Q1}$)			
ATF	11101		Add multiplicand and partial product and right shift collecting underflow.
MTF	01011		
	F = 01000		
RTG	G = 00100	Y = 0	
YTX, GTA	A = 00100	X = 0	
1TC	Count = 1		Decrement counter.
Address 124			
QTF	01100		
XTF	0		
RTG	G = 00110	Y = 0	
YTX, GTQ	Q = 00110	X = 0	
TC0, TXQ1			
Address 122C			
($X \equiv Q1$)			
ATF	F = 00100		
RTG	G = 00010	Y = 0	
YTX, GTA	A = 00010	X = 0	
1TC	Count = 0		Decrement counter.
Address 124			
QTF	00110		
XTF	0		
RTG	G = 00011	Y = 0	
YTX, GTQ	Q = 00011	X = 0	
TC0			Counter zero.
Address 124B			
($\overline{X}Q1$)			
ATF	00010		Subtract multiplicand from partial product and place in A.
$\overline{\text{MTF}}$	10100		
1TF	1		
	F = 10111		
FTG	G = 10111		
GTA	A = 10111		

The result is 1.011100011, but as Q1 is not part of the result in multiplication, the actual product is 1.01110001 = $-\frac{143}{256}$ which is correct.

11.4.1.4 Example 4 (Negative Multiplier and Multiplicand)

Let the multiplier be 1.0011 ($-\frac{13}{16}$) held in the A-register.

Let the multiplicand be 1.0101 ($-\frac{11}{16}$) held in the M-register.

Matrix Outputs	Register	Underflow	Remarks
Address 120			
MTF, FTG, GTJ			Transfer address of
Read/Write	M = 10101		operand from M to J.
			Extract operand.
Address 121			
ATF, FTG, GTQ	Q = 10011		Transfer contents of A to Q.
KTJ, JTC			Set process counter to 4.
TXQ1			Compare overflow digit
			with Q1.
Address 122B			
($\overline{\text{X}}$Q1)			
ATF	00000		Subtract multiplicand from
$\overline{\text{MTF}}$	01010		partial product and right
1TF	1		shift collecting underflow.
	F = 01011		
RTG	G = 00101	Y = 1	
YTX, GTA	A = 00101	X = 1	
1TC	Count = 3		Decrement counter.
Address 124			
QTF	10011		Right shift Q picking up
XTF	1		carry-over from A.
RTG	G = 11001	Y = 1	Collect underflow.
YTX, GTQ	Q = 11001	X = 1	
TC0, TXQ1			Test counter for zero
			and compare signs.

Matrix Outputs	Register	Underflow	Remarks
Address 122C			
(X ≡ Q1)			
ATF	F = 00101		Right shift A.
RTG	G = 00010	Y = 1	Collect underflow.
YTX, GTA	A = 00010	X = 1	
1TC	Count = 2		Decrement counter.
Address 124			
QTF	11001		
XTF	1		
RTG	G = 11100	Y = 1	
YTX, GTQ	Q = 11100	X = 1	
TC0, TXQ1			
Address 122A			
$(X\overline{Q1})$			
ATF	00010		
MTF	10101		
	F = 10111		
RTG	G = 11011	Y = 1	
YTX, GTA	A = 11011	X = 1	
1TC	Count = 1		
Address 124			
QTF	11100		
XTF	1		
RTG	G = 11110	Y = 0	
YTX, GTQ	Q = 11110	X = 0	
TC0, TXQ1			
Address 122C			
(X ≡ Q1)			
ATF	F = 11011		Right shift A propagating sign bit.
RTG	G = 11101	Y = 1	
YTX, GTA	A = 11101	X = 1	Collect underflow.
1TC	Count = 0		Decrement counter.
Address 124			
QTF	11110		
XTF	1		
RTG	G = 11111	Y = 0	
YTX, GTQ	Q = 11111	X = 0	
TC0			Counter zero.

Matrix Outputs	Register	Underflow	Remarks
Address 123B			
($\overline{X}$Q1)			
ARF	11101		Subtract multiplicand from partial product and place in A.
$\overline{\text{MTF}}$	01010		
1TF	1		
	F = 01000		
FTG	G = 01000		
GTA	A = 01000		

The result is 0.100011111, but as Q1 is not part of the result in multiplication, the actual product is 0.10001111 = $\frac{143}{256}$ which is correct.

11.5 Function 13 (Divide)

Division consists essentially of adding and/or subtracting one number from another and shifting the result left one place after each action.

The divide instruction causes the number in the AQ-register to be divided by the number in the specified store location, placing the quotient in the A-register and ignoring the remainder. .

The method of division is to compare the sign digit of the divisor with that of the dividend (remainder) and to apply the following rules:-

Rule 1

If the divisor and dividend digits specified are alike, subtract the divisor from the dividend and add 1 to the partial quotient.

Rule 2

If the divisor and dividend digits specified are not alike, add the divisor to the dividend and add 0 to the partial quotient.

The dividend (remainder) and quotient are left-shifted with each inspection, there being eighteen shifts in all.

Finally, the quotient is "rounded off" by adding one to it.

Generally, the resulting quotient obtained in the A-register may be greater than the true quotient. If the quotient can be expressed exactly in 17 or fewer digits however, then the following statements about the result apply:-

(1) When the divisor is positive, the correct quotient lies in the Q-register and the A-register contains the correct quotient plus 2^{-17}.

(2) When the divisor is negative the correct quotient is c(A) plus 2^{-17} or c(Q) plus 2^{-16}.

NOTE: The least significant digit of the Q-register always becomes zero and the least significant digit of the Q-register always becomes a one.

11.5.1 Worked Examples

The following worked examples of division provide each possible sign of divisor and dividend.

NOTE: As with multiplication, the division process regards the numbers in the operation as fractional. The number held in the AQ-registers is out of range at the start of the operation and is brought into range by the end of the process. The result which is first formed in the Q-register is transferred to the A-register, with a "round off" digit added in the least significant bit position.

11.5.1.1 Example 1 (Positive Divisor and Dividend)

Let the dividend be 0.011011000 (+ $\frac{27}{64}$) held in the A- and Q-registers.

Let the divisor be 0.1001 (+ $\frac{9}{16}$) held in the M-register.

Matrix Outputs	Register	Overflow	Remarks
Address 130			
MTF, FTG, GTJ			Transfer address of operand from M to J. Extract operand.
Read/Write	M = 0.1001		
Address 131			
QTF	F = 11000		
RTG	G = 01100		Right shift contents of Q
GTQ	Q = 01100		
Address 132			
QTF	F = 01100		
LTG	G = 11000		Left shift contents of Q
GTQ	Q = 11000		
K2, KTJ, JTC			Set process counter to 5
Address 133			
ATF	F = 00110		
LTG	G = 01100	Y = 0	Set overflow digit by left shift of A
YTX		X = 0	
TXM18			Compare overflow digit with sign bit.
Address 134B			
(X ≡ M18)			
QTF	11000		Add 1 to partial product and left shift collecting overflow.
1TF	1		
	F = 11001		
LTG	G = 10010	Y = 1	
YTX, GTQ	Q = 10010	X = 1	
1TC	Count = 4		Decrement counter.

Matrix Outputs	Register	Overflow	Remarks
Address 136B			
ATF	00110		Subtract divisor from dividend and left shift picking up overflow.
$\overline{M}$TF	10110		
1TF	1		
	F = 11101		
XTF, LTG	G = 11011	Y = 1	
YTX, GTA	A = 11011	X = 1	
TC0, TXM18			Test counter for zero and compare signs.
Address 134A			
(X ≢ M18)			
QTF	F = 10010		Left shift Q and collect overflow.
LTG	G = 00100	Y = 1	
YTX, GTQ	Q = 00100	X = 1	
1TC	Count = 3		Decrement counter.
Address 136A			
ATF	11011		Add A to M and left shift picking up overflow.
MTF	01001		
	F = 00100		
XTF, LTG	G = 01001	Y = 0	
YTX, GTA	A = 01001	X = 0	
TC0, TXM18			Test counter for zero and compare signs.
Address 134B			
(X ≡ M18)			
QTF	00100		
1TF	1		
	F = 00101		
LTG	G = 01010	Y = 0	
YTX, GTQ	Q = 01010	X = 0	
1TC	Count = 2		
Address 136B			
ATF	01001		
$\overline{M}$TF	10110		
1TF	1		
	F = 00000		
XTF, LTG	G = 00000	Y = 0	
YTX, GTA	A = 00000	X = 0	
TC0, TXM18			

Matrix Outputs	Register	Overflow	Remarks
Address 134B			
(X ≡ M18)			
QTF	01010		
1TF	1		
	F = 01011		
LTG	G = 10110	Y = 0	
YTX, GTQ	Q = 10110	X = 0	
1TC	Count = 1		
Address 136B			
ATF	00000		
$\overline{\text{M}}$TF	10110		
1TF	1		
	F = 10111		
XTF, LTG	G = 01110	Y = 1	
YTX, GTA	A = 01110	X = 1	
TC0, TXM18			
Address 134A			
(X ≢ M18)			
QTF	F = 10110		
LTG	G = 01100	Y = 1	
YTX, GTQ	Q = 01100	X = 1	
1TC	Count = 0		
Address 136A			
ATF	01110		
MTF	01001		
	F = 10111		
XTF, LTG	G = 01111	Y = 1	
YTX, GTA	A = 01111	X = 1	
TC0			Counter zero.
Address 135			
QTF	01100		Add "round off" constant to Q and transfer to A.
1TF	1		
	F = 01101		
FTG	G = 01101		
GTA	A = 01101		

The result obtained is correct according to paragraph 11.5(1) i.e. c(Q) = + $^3/_4$ which is the true quotient and c(A) = + $\frac{13}{16}$ which is greater than the true quotient by the least significant bit.

11.5.1.2 Example 2 (Positive Divisor and Negative Dividend)

Let the dividend be 1.100101000 ($-\frac{27}{64}$) held in the A- and Q-registers.

Let the divisor be 0.1001 ($+\frac{9}{16}$) held in the M-register.

Matrix Outputs	Register	Overflow	Remarks
Address 130			
MTF,FTG,GTJ			Transfer address of operand from M to J.
Read/Write	M = 0.1001		Extract operand.
Address 131			
QTF	F = 01000		
RTG	G = 00100		Right shift contents of Q.
GTQ	Q = 00100		
Address 132			
QTF	F = 00100		
LTG	G = 01000		Left shift contents of Q.
GTQ	Q = 01000		
K2,KTJ,JTC			Set process counter to 5.
Address 133			
ATF	F = 11001		
LTG	G = 10010	Y = 1	Set overflow digit by left shift of A.
YTX		X = 1	
TXM18			Compare overflow digit with sign bit.
Address 134A			
(X ≢ M18)			
QTF	F = 01000		Left shift Q and collect overflow.
LTG	G = 10000	Y = 0	
YTX,GTQ	Q = 10000	X = 0	
1TC	Count = 4		Decrement counter.

Matrix Outputs	Register	Overflow	Remarks
Address 136A			
ATF	11001		Add A to M and left shift picking up overflow.
MTF	01001		
	F = 00010		
XTF, LTG	G = 00100	Y = 0	
YTX, GTA	A = 00100	X = 0	
TC0, TXM18			Test counter for zero and compare signs.
Address 134B			
(X ≡ M18)			
QTF	10000		Add 1 to partial product and left shift collecting overflow.
1TF	1		
	F = 10001		
LTG	G = 00010	Y = 1	
YTX, GTQ	Q = 00010	X = 1	
1TC	Count = 3		Decrement counter.
Address 136B			
ATF	00100		Subtract divisor from dividend and left shift picking up overflow.
$\overline{\text{M}}$TF	10110		
1TF	1		
	F = 11011		
XTF, LTG	G = 10111	Y = 1	
YTX, GTA	A = 10111	X = 1	
TC0, TXM18			Test counter for zero and compare signs.
Address 134A			
(X ≢ M18)			
QTF	F = 00010		
LTG	G = 00100	Y = 0	
YTX, GTQ	Q = 00100	X = 0	
1TC	Count = 2		
Address 136A			
ATF	10111		
MTF	01001		
	F = 00000		
XTF, LTG	G = 00000	Y = 0	
YTX, GTA	A = 00000	X = 0	
TC0, TXM18			

Matrix Outputs	Register	Overflow	Remarks
Address 134B			
(X ≡ M18)			
QTF	00100		
1TF	1		
	F = 00101		
LTG	G = 01010	Y = 0	
YTX,GTQ	Q = 01010	X = 0	
1TC	Count = 1		
Address 136B			
ATF	00000		
$\overline{M}$TF	10110		
1TF	1		
	F = 10111		
XTF, LTG	G = 01110	Y = 1	
YTX, GTA	A = 01110	X = 1	
TC0, TXM18			
Address 134A			
(X ≢ M18)			
QTF	F = 01010		
LTG	G = 10100	Y = 0	
YTX, GTQ	Q = 10100	X = 0	
1TC	Count = 0		
Address 136 A			
ATF	01110		
MTF	01001		
	F = 10111		
XTF, LTG	G = 01110	Y = 1	
YTX, GTA	A = 01110	X = 1	
TC0			Counter zero.
Address 135			
QTF	10100		Add "round off" constant to Q and transfer to A.
1TF	1		
	F = 10101		
FTG	G = 10101		
GTA	A = 10101		

The result obtained is correct according to paragraph 11.5(1).

11.5.1.3 Example 3 (Negative Divisor and Positive Dividend)

Let the dividend be 0.011011000 ($+\frac{27}{64}$) held in the A- and Q-registers.

Let the divisor be 1.0111 ($-\frac{9}{16}$) held in the M-register.

Matrix Outputs	Register	Overflow	Remarks
Address 130			
MTF, FTG, GTJ			Transfer address of operand from M to J.
Read/Write	M = 10111		Extract operand.
Address 131			
QTF	F = 11000		
RTG	G = 01100		Right shift contents of Q.
GTQ	Q = 01100		
Address 132			
QTF	F = 01100		
LTG	G = 11000		Left shift contents of Q.
GTQ	Q = 11000		
K2, KTJ, JTC			Set process counter to 5.
Address 133			
ATF	F = 00110		
LTG	G = 01100	Y = 0	Set overflow digit by
YTX		X = 0	left shift of A.
TXM18			Compare overflow digit with sign bit.
Address 134A			
(X ≢ M18)			
QTF	F = 11000		Left shift Q and collect
LTG	G = 10000	Y = 1	overflow.
YTX, GTQ	Q = 10000	X = 1	
1TC	Count = 4		Decrement counter.

Matrix Outputs	Register	Overflow	Remarks
Address 136A			
ATF	00110		Add A to M and left shift picking up overflow.
MTF	10111		
	F = 11101		
XTF, LTG	G = 11011	Y = 1	
YTX, GTA	A = 11011	X = 1	
TC0, TXM18			Test counter for zero and compare signs.
Address 134B			
(X ≡ M18)			
QTF	10000		Add 1 to partial product and left shift collecting overflow.
1TF	1		
	F = 10001		
LTG	G = 00010	Y = 1	
YTX, GTQ	Q = 00010	X = 1	
1TC	Count = 3		Decrement counter.
Address 136B			
ATF	11011		Subtract divisor from dividend and left shift picking up overflow.
$\overline{\text{M}}$TF	01000		
1TF	1		
	F = 00100		
XTF, LTG	G = 01001	Y = 0	
YTX, GTA	A = 01001	X = 0	
TC0, TXM18			Test counter for zero and compare signs.
Address 134A			
(X ≢ M18)			
QTF	F = 00010		
LTG	G = 00100	Y = 0	
YTX, GTQ	Q = 00100	X = 0	
1TC	Count = 2		
Address 136A			
ATF	01001		
MTF	10111		
	F = 00000		
XTF, LTG	G = 00000	Y = 0	
YTX, GTA	A = 00000	X = 0	
TC0, TXM18			

Matrix Outputs	Register	Overflow	Remarks
Address 134A			
(X ≢ M18)			
QTF	F = 00100		
LTG	G = 01000	Y = 0	
YTX, GTQ	Q = 01000	X = 0	
1TC	Count = 1		
Address 136A			
ATF	00000		
MTF	10111		
	F = 10111		
XTF, LTG	G = 01110	Y = 1	
YTX, GTA	A = 01110	X = 1	
TC0, TXM18			
Address 134B			
(X ≡ M18)			
QTF	01000		
1TF	1		
	F = 01001		
LTG	G = 10010	Y = 0	
YTX, GTQ	Q = 10010	X = 0	
1TC	Count = 0		
Address 136B			
ATF	01110		
$\overline{\text{MTF}}$	01000		
1TF	1		
	F = 10111		
XTF, LTG	G = 01110	Y = 1	
YTX, GTA	A = 01110	X = 1	
TC0			Counter zero.
Address 135			
QTF	10010		Add "round off" constant to Q and transfer to A.
1TF	1		
	F = 10011		
FTG	G = 10011		
GTA	A = 10011		

The result obtained is correct according to paragraph 11.5(2).

11.5.1.4 Example 4 (Negativie Divisor and Dividend)

Let the dividend be 1.100101000 ($-\frac{27}{64}$) held in the A- and Q-registers.

Let the divisor be 1.0111 ($-\frac{9}{64}$) held in the M-register.

Matrix Outputs	Register	Overflow	Remarks
Address 130			
MTF, FTG, GTJ			Transfer address of operand from M to J.
Read/Write	M = 1.0111		Extract operand.
Address 131			
QTF	F = 01000		
RTG	G = 00100		Right shift contents of Q.
GTQ	Q = 00100		
Address 132			
QTF	F = 00100		
LTG	G = 01000		Left shift contents of Q.
GTQ	Q = 01000		
K2, KTJ, JTC			Set process counter to 5.
Address 133			
ATF	F = 11001		
LTG	G = 10010	Y = 1	Set overflow digit by left shift of A.
YTX		X = 1	Compare overflow digit with sign bit.
TXM18			
Address 134B			
(X ≡ M18)			
QTF	01000		Add 1 to partial product and left shift collecting overflow.
1TF	1		
	F = 01001		
LTG	G = 10010	Y = 0	
YTX, GTQ	Q = 10010	X = 0	
1TC	Count = 4		Decrement counter.

Matrix Outputs	Register	Overflow	Remarks
Address 136B			
ATF	11001		Subtract divisor from dividend and left shift picking up overflow.
$\overline{M}$TF	01000		
1TF	1		
	F = 00010		
XTF, LTG	G = 00100	Y = 0	
YTX, GTA	A = 00100	X = 0	
TC0, TXM18			Test counter for zero and compare signs.
Address 134A			
(X ≢ M18)			
QTF	F = 10010		Left shift Q and collect overflow.
LTG	G = 00100	Y = 1	
YTX, GTQ	Q = 00100	X = 1	
1TC	Count = 3		Decrement counter.
Address 136A			
ATF	00100		Add A to M and left shift picking up overflow.
MTF	10111		
	F = 11011		
XTF, LTG	G = 10111	Y = 1	
YTX, GTA	A = 10111	X = 1	
TC0, TXM18			Test counter for zero and compare signs.
Address 134B			
(X ≡ M18)			
QTF	00100		
1TF	1		
	F = 00101		
LTG	G = 01010	Y = 0	
YTX, GTQ	Q = 01010	X = 0	
1TC	Count = 2		
Address 136B			
ATF	10111		
$\overline{M}$TF	01000		
1TF	1		
	F = 00000		
XTF, LTG	G = 00000	Y = 0	
YTX, GTA	A = 00000	X = 0	
TC0, TXM18			

Matrix Outputs	Register	Overflow	Remarks
Address 134A			
(X ≢ M18)			
QTF	F = 01010		
LTG	G = 10100	Y = 0	
YTX, GTQ	Q = 10100	X = 0	
1TC	Count = 1		
Address 136A			
ATF	00000		
MTF	10111		
	F = 10111		
XTF, LTG	G = 01110	Y = 1	
YTX, GTA	A = 01110	X = 1	
TC0, TXM18			
Address 134B			
(X ≡ M18)			
QTF	10100		
1TF	1		
	F = 10101		
LTG	G = 01010	Y = 1	
YTX, GTQ	Q = 01010	X = 1	
1TC	Count = 0		
Address 136B			
ATF	01110		
$\overline{\text{MTF}}$	01000		
1TF	1		
	F = 10111		
XTF, LTG	G = 01111	Y = 1	
YTX, GTA	A = 01111	X = 1	
TC0			Counter zero.
Address 135			
QTF	01010		Add "round off" constant
1TF	1		to Q and transfer to A.
	F = 01011		
FTG	G = 01011		
GTA	A = 01011		

The result obtained is correct according to paragraph 11.5(2).

Chapter 12: OTHER COMPUTING INSTRUCTIONS

12.1 Introduction

Those instructions whose actions may be understood directly from the microprogram and flow diagram are not explained in detail.

In the first matrix address of functions 0, 1, 2, 4, 5, 6, 10, 12 and 13 the store address of the operand is transferred from the M-register to the J-register with the waveforms

$\overline{\text{OTG}}$, MTF, DTF1, DTF2, VTG and GTJ.

Where the operand is to be extracted from the store the waveforms

$\overline{\text{OTM}}$, READ and WRITE

perform this action.

12.2 Function 15

This consists of three short instructions, Input, Output and Program Terminate. Input and Output are considered as one instruction since both access peripherals and the input/output system is described in detail in Chapter 14 of this section.

The Program Terminate instruction provides control signals for the interrupt logic to allow the computer to access the next lower level program which is interrupting.

12.3 Block Transfer (Function 14A)

If the 12th and 13th bits of the M-register are different when function 14 is selected then the logic routes the computer into function 14A. The block transfer microprogram proceeds as follows.

Address 14A2

Set the address of the peripheral in the P-register to/from which the transfer is to take place with MTP.

The number of words in the block to be transferred is set up in the process counter with

QTF, $\overline{\text{OTG}}$, FTG, $\overline{\text{OTJ}}$, GTJ and JTC.

Address 14A1

The starting address, in the main store, is transferred from the A-register to the J-register with ATF, $\overline{\text{OTG}}$, FTG and GTJ.

The block transfer bistable is set with $\overline{\text{SBT}}$.

The 13th bit of the M-register is tested to determine whether the block transfer is of input or output mode with TM13.

If M13 = 0 the transfer is input from peripheral.

If M13 = 1 the transfer is output to peripheral.

Address 14A3D (Input Transfer)

$\overline{\text{OTM}}$ clears the store access register (M) and TC0 tests the process counter for zero content.

Address 14A4D (counter not zero)

SZ1 strobed by tZ1 sets the peripheral select bistable (see Figure 27), and WFP stops the central timer until a peripheral reply ($\overline{\text{Z2}}$) is received. When this happens, the timer starts running again. The word which has been placed on the interface data lines is then input to the G-register with $\overline{\text{OTG}}$ and PTG1. It is transferred to the M-register and then written into the store with GTM, CLEAR and WRITE.

Address 14A6D

The store address is incremented by one with JTF, 1TF, $\overline{\mathrm{OTG}}$, FTG, $\overline{\mathrm{OTJ}}$ and GTJ. The process counter content is decremented by one with 1TC.

The computer then returns to address 14A3D whereupon the cycle is repeated until the process counter is empty (i.e. the block of words has been transferred).

Address 14A5

The BLOCK TRANSFER bistable is reset with $\overline{\mathrm{RBT}}$ and the computer exits from the function.

A similar routine applies to the output transfer (14A3C, 14A4C, 14A6C) except that the words in the block are first read from the store then transferred to the peripheral (c.f. MICROPROGRAM and FLOW DIAGRAM).

Chapter 13: INITIAL INSTRUCTIONS

13.1 Introduction

The initial instructions hardware matrix of 12 x 18 ordinates (i.e. twelve 18-bit words), exists on boards 35 (A-GL) and 36 (A-GK). Figure 26a and 26b show a logical presentation of the hardware in the same manner as the control matrix. The outputs of the matrix are taken via matrix amplifier, directly to the M-register.

The ordinates are energised by the outputs from the store address (J) register, the sequence of addressing being controlled by the SCR (Level 1 only). Therefore, initial instructions appear to the processor to be a specific group of store locations.

13.2 Selection

Before initial instructions can be energised the gating waveforms IIG and IIS have to be true.

IIG is formed by the combination in a large AND gate of the store address register bits and the output of the bistable 14-B/11 & 13, such that IIG is true when the following J-register bits are true:-

$\overline{16}, \overline{15}, \overline{14}, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4$ or 3,

and $\overline{\text{JUMP}}$ is false.

This means that whenever the address keys are set within the range 8180-8191 (inclusive) and the Jump key is pressed, initial instructions are entered.

When IIG (Initial Instructions Gate) goes true $\overline{\text{IIG}}$ goes false and inhibits the read amplifier strobes. The generation of a store clear or read cycle produces IIS (Initial Instructions Strobe), hence initial instructions are entered.

When level 1 is terminated by the program terminate instruction, PTR goes true and resets the bistable 14-B/11 & 13 such that IIG goes false and the initial instructions are disenabled. The store locations 8180-8191 now become available as normal program or data space.

NOTE: Depression of the Jump key will always enable initial instructions when the correct address range is specified.

A detailed description of the use and operation of initial instructions is given in Section 1.2.3 of this manual.

Chapter 14: INPUT/OUTPUT

14.1 Introduction

All inputs and outputs between the processor and peripherals are made under control of Function 14A (block transfer) and Function 15 (normal input/output).

Input and output is entirely asynchronous so that the transfer rate of data depends upon the peripheral being used.

Data is input to the G-register with the signals PTG1 and PTG2 (peripheral to G). Data is output from the accumulator (A) directly.

There are two separate peripheral interface channels through which inputs and outputs take place; the first deals solely with paper tape equipment, which includes the teleprinter (section 4.1.4 gives details) and uses connector 1-SKT10; the other is a general peripheral connection using connector 1-SKT4. Generally, peripherals use the connector 1-SKT13 which carries all the peripheral control signals. In the case of paper tape control the signals are back-wired from 1-SKT13 into 1-SKT10 such that control and data signals for paper tape are provided through 1-SKT10.

14.2 Peripheral Interface Requirements

When the computer requires an input it sends an 'input select' to the specified peripheral and awaits a 'reply' from the peripheral. The computer stops as soon as 'input select' is sent and restarts when 'reply' is received, by which time the computer assumes that the data is on the interface lines ready to be strobed into the processor. After the data has been strobed in,the select signal goes false and the data may then be removed by the peripheral from the interface lines.

When the computer requires to output data it sends an 'output select' and stops until a reply is received, by which time the peripheral has strobed the data from the interface lines. When 'reply' is received 'output select' goes false, but since data is output from the accumulator directly it is only removed from the interface lines when the accumulator is cleared.

14.3 General Input/Output

14.3.1 Input

Incoming data through the interface socket 1-SKT4 is gated into G with PTG1 as a full 8-bit word. The data bits are IG1-IG18 inclusive (see Figure 28).

14.3.2 Output

Data is output as an 18-bit word directly from the accumulator and leaves socket 1-SKT4 as bits OA1-OA18 inclusive.

14.4 Paper Tape Input/Output

14.4.1 Input

Data inputs from paper tape reader or teleprinter via connector 1-SKT10 is accepted in 8-bit characters which have to be assembled into 18-bit words. The character from the paper tape reader or teleprinter (IR1-IR8 inclusive) is gated into the eight less significant positions of the G-register with PTG2. At the same time, PTG2 gates bits 2-11 of the accumulator into positions 9-18 of G, respectively.

The content of G is then put back into the accumulator with GTA. This action constitutes a character shift of seven places left in the accumulator, the new input character being placed in the less significant end.

NOTE: The inverse output from the least significant bit of the accumulator $(\overline{A1})$ is gated into element 29-K/12 in such a manner that if bit 1 of the accumulator (A1) is a logic '1', then the 8th bit of the reader character is overwritten. This means that if A1 is initially logic 1, then A8 will become a logic 1 necessarily on the next character shift.

If bit 1 of the accumulator is a zero however, then the possibility of corrupting this character with the next character input has to be catered for by program.

14.4.2 Output

Data output to the paper tape punch or teleprinter is taken from the inverse output of the eight less significant bits of the accumulator (i.e. $\overline{A1}$-$\overline{A8}$ inclusive) which produce data output bits OP1-OP8 respectively.

There is no facility for shifting the accumulator content right (down) 8 bits at a time. A character is output from the accumulator and the content is then right-shifted 8 places (character shift) by program such that the next character may be output.

14.5 Peripheral Addressing

The peripheral address register P is a 13-bit register which is loaded from M with MTP. The 13 address bits of the instruction word in M are transferred into the corresponding positions in P. The two more significant bits of P (P12 and P13) are not used to form part of the peripheral address but are used in the input/output control logic to generate peripheral select waveforms. The remaining 11 bits form the peripheral address bits OS1-OS11 giving a possible 2048 discrete addresses (address 0 being a permissible peripheral address).

OS1-OS11 are taken to the general peripheral socket 1-SKT4.

The inverse outputs of the four least significant bits of the P-register ($\overline{P1}$, $\overline{P2}$, $\overline{P3}$ and $\overline{P4}$) are used to generate ADDRESS BIT1 - ADDRESS BIT4 respectively. These signals are output from connector 1-SKT10 to the paper tape station, but only ADDRESS BIT3 is used (Section 4.1.3 gives details).

14.6 Input/Output Control

The input/output control logic (Figure 27) which governs the selecting of peripherals and the transference of their replies to the processor is found entirely on board 30 (A-FQ).

The select logic consists of paper tape select signals (STR and STP) and general peripheral select signals (SIP and SOP).

All peripheral selects are instigated by the Peripheral Select bistable which has to be set in order that any of the select signals may become true (i.e. when reset, the bistable inhibits all peripheral selects).

There are three peripheral reply signals, of which two come from the paper tape logic (i.e. RTR and RTP). RTP is not used. The third signal is the general peripheral reply PR.

For any reply to be recognised it has to exist for not less than 470 ns (see Figure 27).

14.6.1 Logic Description (Figure 27)

14.6.1.1 Normal Input/Output (Function 15)

Initially, $\overline{\text{RESET}}$ goes false and resets the Block Transfer and Peripheral Select bistables. SZ1 becomes true which, together with tZ1 from the timing chain, sets the Peripheral Select bistable.

If the paper tape equipment is to be selected (i.e. STR or STP true), then the following address bits have to be true:-

P12, $\overline{P11}$, $\overline{P10}$, $\overline{P9}$, $\overline{P8}$, $\overline{P7}$, $\overline{P6}$ and $\overline{P5}$.

Furthermore, for an input from paper tape to occur STR has to be true which means that $\overline{P13}$ has to be true. If P13 is true, then STP (Select Paper Tape Punch) will be true.

If any one of the inputs to element 30-P/12 or 30-N/12 are false, then selection of the paper tape equipment is inhibited, although general peripheral selection (SIP or SOP true) can occur if P12 is true.

The Peripheral Select bistable is reset in the next t4.

14.6.1.2 Block Transfer (Function 14A)

$\overline{RESET}$ when false resets the Block Transfer and Peripheral Select bistables as before. $\overline{SBT}$ goes false and sets the Block Transfer bistable such that the output of element 30-B/13 is false. This inhibits the selection of paper tape equipment. SZ1 strobed by tZ1 set the Peripheral Select bistable and if P12 and $\overline{P13}$ are true then 30-B/12 output goes false and SIP (Select Input Peripheral) becomes true. The signals SIP and BT will then be used by the peripheral to input a block of data to the computer. When $\overline{P12}$ and P13 are true 30-M/11 output goes false and SOP (Select Output Peripheral) becomes true. In this case, a block transfer output to a peripheral occurs.

The Peripheral Select bistable is reset in the next t4 and the Block Transfer bistable reset by $\overline{RBT}$.

In order to recognise the last word of a block transfer the input/output control logic generates a "last word" signal (LW).

LW is true when BT is true, together with C(1-4)1 and C(5-12)0. The signals C(1-4)1 and C(5-12)0 are output from the process counter which is initially set with the number of words in the block to be transferred.

14.6.1.3 Summary of Select Conditions

The following list shows the signals which must be true to make given select signals true.

STR	-	$\overline{BT}, \overline{P13}, P12, \overline{P11}, \overline{P10}, \overline{P9}, \overline{P8}, \overline{P7}, \overline{P6}$ and $\overline{P5}$.
STP	-	$\overline{BT}, P13, P12, \overline{P11}, \overline{P10}, \overline{P9}, \overline{P8}, \overline{P7}, \overline{P6}$ and $\overline{P5}$.
SIP	-	$\overline{BT}, \overline{P13}$, and $\overline{P12}$ or $BT, \overline{P13}$ and $P12$.
SOP	-	$\overline{BT}, P13$ and $\overline{P12}$ or $BT, P13$ and $\overline{P12}$.

From the above list it is evident that when $\overline{P13}$ is true and input peripheral is selected and when P13 is true an output peripheral is selected.

Hence, the ranges of address for each type of input and output are as follows:-

Block Transfer input	$2048 < N < 4095$
Block Transfer output	$4096 < N < 6143$
Normal input	$0 < N < 2047$
Normal output	$4096 < N < 6143$
Paper Tape Reader input	$N = 2048$
Paper Tape Punch output	$N = 6144$
Teleprinter * input	$N = 2052$
Teleprinter * output	$N = 6148$

where N is the number of the store location.

* These addresses only refer to the teleprinter when it is connected to the paper tape equipment socket (1-SKT10).

NOTE: Peripheral address registers bits P1, P2, P3 and P4 are not used by the input/output control logic and have no effect on the peripheral select signals.

14.6.1.4 Peripheral Replies

The receivers 30-J/11, 12 and 13 are connected as a distributed NOR gate. The receipt of a reply (RTP, RTR and PR) causes the output of this NOR gate to go false. Hence, 30-B/11 output goes true and a 470 ns positive-going pulse is output from 30-F/13. At the end of this pulse, 30-G/13 output goes true for 100 ns. The two inputs to 30-C/12 will now be true and $\overline{Z2}$ (the reply to the processor) goes false for 100 ns.

NOTE: If the peripheral reply (RTR, RTP or PR) does not remain true for at least 470 ns then the output of 30-B/11 will go false again before the 100 ms pulse generator is fired and prevent $\overline{Z2}$ from going false.

Chapter 15: INTERRUPTS (FIGURE 30)

15.1 Introduction

The interrupt system consists basically of two small registers. The first holds any interrupt demands that are made by operators from the keys or by the "on-line" peripherals. The second holds the program level currently operating. For each program level there is a discrete SCR (Sequence Control Register) and B (Address modifier) register which are referred to by the store address register (J) via the interrupt logic.

15.2 Initial Conditions

When the computer is switched on, $\overline{\text{RESET}}$ goes false and resets all the bistables. The Level 1 demand bistable is set by $\overline{\text{RESET}}$ going false such that the Interrupt lamp of level 1 is lit (i.e. INT1 is false).

Provided neither the Fn0 (set B-register), nor the Fn15 (program terminate) instruction is given the logic operates as follows:-

The computer enters address C2 with $\overline{\text{RESET}}$ false and DTE (test for interrupt) is output from the matrix. DTE has no effect at this point since the timer is not yet running. $\overline{\text{SA0}}$ goes false, ends the reset condition and puts the computer in the Matrix address C0. Once again DTE is output from the matrix and as $\overline{\text{SA0}}$ goes true again. $\overline{\text{IC}}$ (Initial Conditions) goes true and the timer starts running. In t1 the content of the Program Terminate bistable is inspected (element 3-C/11) and since it is reset the outputs from the E-bistables are inhibited from setting the interrupt demand bistables. In t2 the E-bistables are reset ($\overline{\text{E1}}$ and $\overline{\text{E2}}$ true) by the output of 3-P/13 going false. This causes the output of 1-E/11 to go false such that the LEVEL 1 lamp lights. In t3 the output of element 3-G/11 goes true and opens the gates 3-E/13, J/12 and K/12 so that interrupt demands are strobed to the E-bistables, but initially level 1 demand only is made so the E-bistables remain unaltered.

The computer will then cycle through function control in program level 1.

As the computer leaves address C0, DTE goes false and output of 3-K/13 goes true. This allows interrupts from the keys or peripherals to be recognised and set the appropriate interrupt demand bistables.

NOTE: Interrupt demands are registered at any time other than in the matrix addresses C0 and C2 of function control. Response to an interrupt is made in the next cycle of the address C0 or C2.

15.3 Typical Interrupt Cycle

Suppose an interrupt demand has been made and the computer has just finished the current instruction and it is returning to function control. As the computer enters the matrix address C2, the test for interrupt signal (DTE) goes true. The output of 3-K/13 goes false and inhibits any further interrupts from being registered in this cycle (C2). In t2 the E-bistables are reset (level 1 active) and in t3 the interrupt demand bistable contents are gated into the E-bistable so that the new program level becomes effective. The contents of the E-bistables are then gated with ETJ (which is output from the matrix in address C2) on board 6(A-FI) to produce $\overline{\text{ETJ2}}$ and $\overline{\text{ETJ3}}$ which set the store address register with the 2-bit address of the pertinent SCR. If address modification is required then the signal K1 becomes true in matrix address C6A. K1, together with ETJ causes $\overline{\text{KTJ1}}$ to go false and this sets a one in the least significant bit of J. The effect is to select the B-register address corresponding to the current program level. The computer now continues to extract and obey instructions selected by the new SCR.

15.4 Interrupt Production

15.4.1 On Line and Manual

Interrupts will normally be produced by the on-line peripherals which produce the signals INT1, INT2 and INT3 accordingly, but when the interrupt mode switches are in the MANUAL position, the signals $\overline{\mathrm{PS1}}$, $\overline{\mathrm{PS2}}$ and $\overline{\mathrm{PS3}}$ (suppress peripheral interrupts) go false and prevent peripheral interrupts from being recognised. Manual interrupts may now be generated by pushing the INTERRUPT/LEVEL switches on the control unit. INT1k, INT2k and INT3k become true as the corresponding INTERRUPT/LEVEL switches are pushed and set the interrupt demand bistables accordingly, when DTE is false.

15.4.2 Trace

The interrupt mode switches have a TRACE position which, when used, provides a permanent interrupt demand at the corresponding level, i.e. when the interrupt mode switch for level 1 is placed in the trace position, then INT1k becomes true until the switch is moved from the trace position.

This trace facility has a special use in fault finding when used with "trace programs".

15.5 Set B-Register (Fn0)

Whenever Fn0 (set the B-register) is entered the conditions DD and DH become true. In t4 of the current cycle, DD and DH set the bistable 3-F/12 and 13 such that 3-F/13 output is false. This inhibits the gates 3-P/11 and 3-P/13. Interrupts demands are strobed into the E (program level) bistables by the end of t3 in the control instruction address C2; but the Inhibit Gate Interrupts bistable (3-F/12 and 13) is not reset until t4. Therefore, the processor has to cycle through control and complete one other instruction after Fn0 before the program level can be changed. The significance of this may be seen in Volume 2.

15.6 Program Terminate

When the "Program Terminate" instruction (Fn15) is given the Program Terminate bistable is set in t3 with PTR. The computer completes the program terminate instruction and enters control with DTE true.

In t1 of the control block C2, the gates 3-E/11, N/11 and N/13 are opened. The outputs of the program level (E) bistables are routed into these gates so as to reset the Interrupt Demand bistable and clear the demand which caused the current level program to interrupt. In t2, the Program Terminate and the E-bistables are reset. In t3 the current interrupt demands are gated into the E-bistables. In this way the computer will select the next lower level program demanded after terminating the current level.

Chapter 16: POWER DISTRIBUTION

16.1 Introduction

The power distribution of the 900 central processor is divided into two parts; the a.c. distribution and the d.c. distribution. Power wiring for the paper tape equipment is not included in this chapter.

16.2 A.C. Distribution

The a.c. supply circuits are shown in Figure 31. The computer does not contain any switching for the a.c. supply and if the fuses are intact the a.c. circuits become "live" when switched on at the local source. The supply is brought to the computer via a free socket which connects to PL1 of the fan and filter unit.

The mains to the fans and the four 13A fleet-pin sockets (SK1, SK2, SK3, SK4) of the fan and filter unit is ring-wired.

Of the four outlets, sockets 1 and 2 are spares, socket 3 supplies the computer power supply unit 7, and socket 4 supplies the paper tape controller power unit.

The mains cable from the fan and filter unit to power unit 7 ends in a 5-way free bayonet type socket, the pins on which are allocated as follows:-

A - Live
B - Neutral
C - Earth
D)
E) - Not connected

16.3 D. C. Distribution

Figure 32 shows the d.c. distribution of the computer power supply. The 26 pin output socket 7-SK2 is connected to the computer connector panel 1-PL6 by a one-to-one 61-way cable.

The distribution of supplies from 1-PL6 is tabulated as well as illustrated. The wiring is shown as far as the pin of the first board position that a particular supply visits. The supply wires may be followed by their respective colour codes and will generally visit the same pin position for every board on which supply(s) is required.

16.4 D. C. Supplies Availability

The d.c. supplies are neither switched on sequentially nor are they monitored for correct level by any logic external to the power unit 7. It is assumed that if the computer switches on with depression of the On button that all the supplies are available and within tolerance. The signal $\overline{\text{PSC}}$ which comes from the power unit at 7/SK2 pin T becomes logic 0 V when the d.c. supplies are available. This is routed to the "PSC Receiver and Delay" which is mounted on the store board A-EC3 (see Figure 33). This logic produces PSC (Power Supplies Correct) which is routed to the manual control logic. If this signal does not become true the computer does not become operational.

16.5 Earth Strap

To improve noise immunity all 900 Systems which include 9 Kc/s Magnetic Tape and, if necessary, other systems, are fitted with an earth strap. Connections are made via earth screws fitted to the frame of the desk or handler. The lengths of the earth straps are the same as the corresponding signal cables.