

THE ENGLISH ELECTRIC COMPANY LIMITED

DEPT. DATA PROCESSING AND CONTROL
KIDSGROVE SYSTEMS,
F.S.D. or Order No.

REPORT No.K/GD.u.293
A.C.E.D. Ref. No.
Deptl. Ref.
Date 7.9.62

Copies	TO:-
1	Mr. H.M. Mathews, Director of Engineering, London. (Summary Sheet).
1	Mr. J.M. Ferguson, Chief Engineer, Stafford.
1	Mr. J. Boothroyd, Chief Instructor, DP&CSD, Kidsgrove.
1	Mr. T.H. Elliott, Chief Product Design Engineer, Kidsgrove.
1	Mr. A.C.D. Haley, Chief Development Engineer, Kidsgrove.
1	Mr. G.D. Royle, Group Engineer, F.E.S., Kidsgrove.
1	Mr. C. Robinson, Computing Services Manager, Kidsgrove.
1	Mr. R.H. Allmark, DP&CSD, Kidsgrove.
1	Chief Engineer's File.
1	M.E.L. Library, Whetstone.
60	Instruction Department, DP&CSD, Kidsgrove.

NOTES ON KDF 9 MAIN CONTROL.

Report by
J.A. Edwards.

SUMMARY.

This report is a provisional description of the KDF 9 Main Control. It is not a complete and fully detailed report but may nevertheless be found useful by maintenance and production engineers.

The material in this report has been obtained in discussions with engineers working on the KDF 9 development project - notably Mr. J.P. O'Neil, Mr. W. Jones and Mr. W.R. Abbott.

CONTENTS

- 1.0 General
- 2.0 Instruction Fetching Logic
- 3.0 The Sequence Unit
- 4.0 The Q Mill
- 5.0 The Q Stores
- 6.0 The Jump Nesting Stores.

1.0 GENERAL

Main Control has been described in general terms in Report K/GD.y.77 which provides a list of explanatory drawings. The major divisions of MC are the Instruction Fetching Logic, the Sequence Unit, the Q Store, the Q-Mill, the Jump Nesting Store and the Fetch Buffers.

Note: Most of the drawing numbers quoted in this report are prefixed "K 78031 - " (e.g. K 78031/20261). The "78031" will be omitted in references which are given (e.g. K-/20261).

2.0 INSTRUCTION FETCHING LOGIC

The Instruction Fetching Logic consists of two Instruction Word Buffers (IWB), an Instruction Staticiser (INS), a Next Instruction Word Address Register (NIWA), a Next Instruction (Syllabic) Address Register (NIA) and suitable logic for incrementing the number in the NIA register. The number in the NIWA register is adjusted, as necessary, by means of the Q-Mill.

The Instruction Word Buffers are two 48-bit registers to which instruction words are transferred from the Main Store Buffer. They are filled alternately. The actions which take place during a Main Store Cycle concerned with instruction fetching have been listed in Report K/GD.y.78.

The Instruction Staticiser is a 24-bit register to which can be transferred any three syllables stored in the two IW Buffers. The most significant of the three is that selected by the number in the NIA register. Certain of the outputs of this register are applied to decoding gates which generate a "function level" defining the operation to be performed.

The NIWA register has a capacity of 13 bits; it is used to store the MS address of the next instruction word to be fetched. Its contents can be transferred to, and it is set from, the QBN register in the Q-Mill.

The NIA Register can hold four bits. These bits are decoded and prime one of the twelve transfer gates by which syllables may be transferred from the IW Buffers to the INS. The most significant of the four digits indicates from which buffer the syllables should be taken; the other three specify the most significant of the three syllables to be transferred. For syllabic addressing purposes, the two IW Buffers constitute a cyclic store. If NIA specifies the least significant syllable of IWB 1, for example, then that syllable will be transferred together with the two most significant syllables of IWB 0.

The logic provided for incrementing the number in the NIA register consists of a similar register, CIA (Current Instruction Address) and a block of logic which adds 1, 2, 3 or 4 (the reasons for these counts will be given later) to the number in the CIA register. When a new instruction has been fetched, the number in NIA is transferred to CIA, after which the appropriate outputs of the "count" logic are transferred to the NIA register. Also associated with the NIA register is the Jump Address Buffer (JAB) to which the contents of the NIA are transferred during a "Jump" operation. "Jump" instructions specify the MS and syllabic addresses of the instruction next to be obeyed. During "Jump" operations, therefore, the NIA register is set from the INS. The contents of the NIA register are then transferred to the JAB in order that this number shall be preserved for the use of Arithmetic Control.

2.1 Instruction Fetching Sequence

A flow chart which describes, in general terms, the instruction fetching sequence is provided on drg. CE/SKM 5702; a suitable block diagram is provided on drg. CE/SKM 5726.

When the machine is started initially, the initial input switch which sets up the first read operation from paper tape also sets, or resets, a number of flip-flops in Main Control. This causes the sequence unit in MC to embark, when the peripheral transfer has been completed, on a sequence of actions commencing at point "A" on the flow chart. The significant actions which follow cause the MSA (Main Store Address) and the NIA (Next Instruction Address) registers to be cleared and a MS "fetch instruction" cycle to be initiated. The events which take place during this cycle are described in report K/GD.y.78.

The word in location "0" of the Main Store is transferred to Instruction Word Buffer One (IWB) and the address of the MS location holding the next instruction word is stored in the NIWA (Next Instruction Word Address) register.

The first three syllables of the word just fetched to the IWB are then transferred to the Instruction Staticisers (INS) and the contents of the NIA register are transferred to the CIA (Current Instruction Address) register. The outputs of the appropriate INS flip-flops (the operation code) are decoded and generate a signal indicating the nature of the operation to be performed. The outputs of other flip-flops (which contain digits indicating the number of syllables in the instructions stored in the INS) are applied to the "count" logic, and the appropriate outputs of this logic are transferred to the NIA register. The NIA now contains a number which addresses the first syllable of the next instruction to be obeyed. The sequence of actions specified by the operation code is then carried out.

All sequences, except those involving a "jump", return to the action which transfers from the IWB to the INS the three syllables specified by the number in the NIA register. When this occurs, certain digits in the NIA and CIA registers are inspected by logic which indicates when a new instruction word must be fetched from the Main Store. If a new word is not required, the contents of NIA are transferred to CIA, the number in NIA is suitably incremented and the instruction in the INS is obeyed. If a new word is required, the NIA → CIA transfer does not take place and the number in NIA is left unchanged. The new instruction word is fetched from the Main Store to the IWB which is not in use and the three syllables specified by NIA are transferred to the INS once more. (the correct digit patterns being transferred this time). The number in NIA is then transferred to CIA, and counted back into NIA, and the instruction in the INS obeyed.

2.2 Transfers to the Instruction Staticiser

A KIF 9 instruction may have one, two or three syllables each containing eight binary digits. The two Instruction Word Buffers may, therefore, contain any number (not necessarily an integral number) of instructions between 4 and 12. The first syllable of an instruction may occupy any of the 12 syllabic locations in the two Instruction Word Buffers.

The Instruction Staticiser is a 24-bit register into which any three of the syllables in the Instruction Word Buffers may be transferred. The three syllables are transferred via one of twelve sets of transfer transformers, the primary windings of these being selected (not directly) by the decoded output of the NIA register. These transformers are pulsed by the MC sequence unit when the previous instruction has been completed.

The two most significant digits of every instruction specify the number of syllables in that instruction. The first two bits of single syllable instructions are zeros and the first two bits of two syllable instructions are "01". If the most significant bit is a "1", the instruction is a three syllable instruction. The outputs of the Instruction Staticiser flip-flops containing these digits are, therefore, applied to the "count" logic. These digits may give rise to a count of 1, 2 or 3.

However, Main Control is not required to take any action if a single syllable instruction is encountered. The "count" logic may, therefore, skip over such instructions and cause the next 2 or 3 syllable instruction to be addressed. For this purpose, the count logic is made a little more complex and is supplied with the outputs of INS flip-flops 8, 9, 16 and 17 (the most significant digits of the second and third syllables - known as C, D, E and F in the prototype.)

For this purpose also, three extra flip-flops (W, X and Y) are associated with the staticiser; they contain the two most significant digits of the fourth syllable and the most significant digit of the fifth. Digits are transferred to these flip-flops when syllables are transferred to the staticiser. The six transfers which may take place from either of the buffers are as follows, where $S(n)$ indicates the n th syllable in the buffer and $D(p)$ indicates its p th digit ($S6, S7, S8$ and $S9$) refer to the first four syllables of the "other" IWB):

- (i) $S0, S1, S2 \rightarrow INS; D(0,1) S3 \rightarrow W \text{ \& } X; D(0) S4 \rightarrow Y.$
- (ii) $S1, S2, S3 \rightarrow INS; D(0,1) S4 \rightarrow W \text{ \& } X; D(0) S5 \rightarrow Y.$
- (iii) $S2, S3, S4 \rightarrow INS; D(0,1) S5 \rightarrow W \text{ \& } X; D(0) S6 \rightarrow Y.$
- (iv) $S3, S4, S5 \rightarrow INS; D(0,1) S6 \rightarrow W \text{ \& } X; D(0) S7 \rightarrow Y.$
- (v) $S4, S5, S6 \rightarrow INS; D(0,1) S7 \rightarrow W \text{ \& } X; D(0) S8 \rightarrow Y.$
- (vi) $S5, S6, S7 \rightarrow INS; D(0,1) S8 \rightarrow W \text{ \& } X; D(0) S9 \rightarrow Y.$

The transfer to be performed is selected by decoding the outputs of the NIA register as discussed below. The three least significant digits in this register specify the number of a syllable in binary; the most significant digit selects the IWB.

Packages

A transfer path must be provided between each syllable in the buffers and the three syllables in the INS; thirty-six syllable transfer paths in all. The packages used are Mill transfer boards type K 251/10. Each of these packages bears three transformers of which one has 4 secondaries and the other two have 8. Corresponding secondaries on the three transformers are connected to common output pins. Because of the 'commoned' outputs, two packages are required to transfer a complete syllable to the INS. The large transformers, therefore, are each connected to corresponding halves of different syllables in the buffers. The wiring is arranged so that four of the small transformers can transfer another complete syllable. The 36 paths are thus provided by 30 packages.

In addition to these transfers, paths must be provided for transfers to the W, X and Y flip-flops. Paths are required from the two most significant bits of all syllables to the W and X flip-flops and from the most significant bit of all syllables to the Y flip-flop. Two packages are provided for these transfers.

2.3 The NIA Counter

Since the first syllable of an instruction may be located anywhere in the two buffers, the NIA register may contain any of the combinations of digits shown in fig. 1. In order to obtain the syllabic address of the next instruction, it may be necessary to increment the number in the NIA register by 1, 2, 3 or 4 - depending on the number of syllables in the current instruction and whether any following instructions may be skipped over.

As has been shown, when the Instruction Staticiser has been filled, the contents of the NIA register are transferred to the CIA register. The outputs of this register are applied to logic which gates the secondary windings of transfer transformers so as to set or reset the appropriate NIA flip-flops when a transformer is pulsed. There is one transformer for each of the possible counts; the one to be pulsed is selected by the "count" logic.

Truth tables and Veitch diagrams specifying the actions necessary for each count are provided in figures 2 to 9. In these diagrams the flip-flops of the NIA and CIA registers are called d, c, b, a and $\delta, \gamma, \beta, \alpha$ (most significant on the left) as in the prototype's logic diagrams.

Apart from the four counts already mentioned, the NIA register may be set directly from the CIA register or the Instruction Staticiser or it may be cleared. The direct transfer from the CIA register is used during the interruption sequence and in "Subroutine Jump" operations where the syllabic address to be stored in the Jump Nesting Store may be the address of the current instruction. The three least significant digits of the register are set from the INS, and d is set equal to δ (in order to fetch the new instruction word to the "other" IWB) during "Jump" operations. The three l.s. digits are cleared, and d is set equal to $\bar{\delta}$ during the Interruption sequence and in Short Loop Jump operations where control must be transferred to the first syllable of a selected word.

There is a special count which may be required during the execution of "EXIT" (S8) instructions. During "Jump to Subroutine" operations, the NS and syllabic addresses of the last instruction obeyed are stored in the Jump Nesting Store. When, subsequently, an "EXIT" instruction is obeyed, these addresses are taken from the Jump Nesting Store and adjusted to provide the address of the instruction which must be obeyed. The instruction specifies the number of half-words (three syllables) by which the addresses must be incremented. If this number is odd, three must be added to the syllabic address. However, the instruction word buffer to which the word containing the new instruction is fetched must not be changed since this might allow instructions to be overwritten which are still required by Arithmetic Control.

2.4 The Count Logic

The count logic must specify an increment of 1, 2, 3 or 4 depending on the number of syllables in the current instruction and the number of instructions which may be skipped over. The logic is further complicated by the requirement that "skipping" must not be allowed if control is about to be transferred to the Director program. When this occurs, the count logic is used to indicate the syllable to which the Director must return control when the program is re-entered; if any instructions are skipped by this count, Arithmetic Control will never have the chance of obeying them. Skipping is prevented by the SKIN (Skip Inhibit) flip-flop alias flip-flop I.

The count logic is shown in drg. K 78031/9482. The conditions for generating the four counts are given below, where Dx indicates the xth digit in the Staticiser and C0, C1, C2 and C3 are the four digits of the CIA register:

Count 1

- (a) if the current instruction is a one syllable instruction and it was taken from syllabic location 5:

$$\overline{D0} \quad . \quad \overline{D1} \quad . \quad C1 \quad . \quad C3$$

in the prototype: $\overline{A} \cdot \overline{B} \cdot \alpha \cdot \gamma$

- (b) if the current instruction is one syllable long and either the program is being interrupted or the next instruction is 2 or 3 syllables long:

$$\overline{D0} \quad . \quad \overline{D1} \quad . \quad [D8 \quad /D9 \quad /SKIN \quad]$$

in the prototype: $\overline{A} \cdot \overline{B} \quad (C + D + I)$

Count 2

- (a) if the current instruction has two syllables and it was taken from syllabic location 4 or 5:

$$\overline{D0} \quad . \quad D1 \quad . \quad C1$$

in the prototype: $\overline{A} \cdot B \cdot \gamma$

- (b) if the current instruction has two syllables and either the program is being interrupted or the next instruction is 2 or 3 syllables long:

$$\overline{D0} \quad . \quad D1 \quad . \quad [D16 \quad /D17 \quad /SKIN \quad]$$

in the prototype: $\overline{A} \cdot B \cdot (E + F + I)$

- (c) if the current instruction and the next instruction are both one syllable in length, the program is not being interrupted, the current instruction was taken from syllabic location 0, 1, 2, 3 or 4 and the next but one instruction has two or three syllables:

$$\overline{D0} \quad . \quad \overline{D1} \quad . \quad \overline{D8} \quad . \quad \overline{D9} \quad . \quad \overline{SKIN} \quad .$$

$$(C1 \quad . / \overline{C3} \quad . \quad (D16 \quad / D17)$$

$$\text{in the prototype: } \overline{A}. \overline{B}. \overline{C}. \overline{D}. \overline{I} (\overline{\gamma} + \overline{\alpha})(E + F)$$

Count 3

- (a) if the current instruction is a 3 syllable instruction and it was taken from syllabic location 3, 4 or 5:

$$D0 \quad . \quad [(C2' \quad . \quad C3(\quad) / C1 \quad]$$

$$\text{in the prototype: } A (\alpha\beta + \gamma)$$

- (b) if the current instruction is a 3 syllable instruction and either the program is being interrupted or the next instruction is 2 or 3 syllables long:

$$D0 \quad . \quad [\overline{SKIN} \quad / (W(\quad / X \quad]$$

$$\text{in the prototype: } A (I + W + X)$$

- (c) if the current instruction is a 2 syllable instruction and the next instruction is a one syllable instruction which is followed by a 2 or 3 syllable instruction, provided that the program is not being interrupted and the current instruction was taken from syllabic location 0, 1, 2 or 3. If the current instruction happens to be a three syllable instruction in which digits 16 and 17 are both zero, a count of 3 will be requested by this condition:

$$D1 \quad . \quad \overline{D16} \quad . \quad \overline{D17} \quad . \quad \overline{SKIN} \quad . \quad C1(0)$$

$$[W \quad . \quad / X \quad]$$

$$\text{in the prototype: } \overline{B}. \overline{E}. \overline{F}. \overline{\gamma}. \overline{I} (W + X)$$

- (d) referring to the above condition, if the current instruction was in fact taken from syllabic location 3, a count of three is required whatever the state of the W and X flip-flops:

$$D1 \quad . \quad \overline{D16} \quad . \quad \overline{D17} \quad . \quad \overline{SKIN} \quad . \quad \overline{C1} \quad . \quad C3 \quad .$$

$$C2'$$

$$\text{in the prototype: } B. \overline{E}. \overline{F}. \overline{\gamma}. \overline{I}. (\alpha.\beta)$$

- (e) if the current instruction is one syllable in length and is followed by two one syllable instructions and the instruction after these has two or three syllables, provided that the program is not being interrupted and the current instruction was taken from syllabic location 0, 1, 2 or 3. This condition includes the case of a current instruction being a two or a three syllable instruction in which digits 8, 9, 11 and 17 are all zero:

$$\overline{CT} \cdot \overline{D8} \cdot \overline{D9} \cdot \overline{D16} \cdot \overline{D17} \cdot \overline{SKIN} \cdot [W/X]$$

in the prototype: $\bar{y} \cdot \bar{C} \cdot \bar{D} \cdot \bar{E} \cdot \bar{F} \cdot \bar{I} \cdot (W + X)$

- (f) referring to the above condition, if the current instruction was in fact taken from syllabic location 3, a count of three is required whatever the state of the W and X flip-flops:

$$\overline{D8} \cdot \overline{D9} \cdot \overline{D16} \cdot \overline{D17} \cdot \overline{SKIN} \cdot \overline{CT} \cdot [C3 \cdot C2]$$

in the prototype: $\bar{C} \cdot \bar{D} \cdot \bar{E} \cdot \bar{F} \cdot \bar{I} \cdot \bar{y} \cdot (\alpha \cdot \beta)$

Count 4

- (a) if the current instruction is three syllables long and is followed by a one syllable instruction, provided that the program is not being interrupted and the current instruction was taken from syllabic location 0, 1 or 2:

$$\overline{D0} \cdot \overline{W} \cdot \overline{X} \cdot \overline{SKIN} \cdot \overline{C1} \cdot [C3/C2]$$

in the prototype: $A \cdot \bar{W} \cdot \bar{X} \cdot \bar{I} \cdot \bar{y} \cdot (\bar{\alpha} + \bar{\beta})$

- (b) if the current instruction is a two syllable instruction which is followed by 2 one syllable instructions provided that the program is not being interrupted and the current instruction was taken from syllabic location 0, 1 or 2. This condition includes the case of a three syllable instruction having digits 8 and 9 both zero:

$$\overline{D1} \cdot \overline{D16} \cdot \overline{D17} \cdot \overline{W} \cdot \overline{X} \cdot \overline{SKIN} \cdot \overline{C1} \cdot [C3/C2]$$

in the prototype: $A \cdot \bar{D} \cdot \bar{E} \cdot \bar{F} \cdot \bar{W} \cdot \bar{X} \cdot \bar{I} \cdot \bar{y} \cdot (\bar{\alpha} + \bar{\beta})$

- (c) if the current instruction is a one syllable instruction and it is followed by three more such instructions, provided that the program is not being interrupted and the current instruction was taken from syllabic location 0, 1 or 2. This condition includes the cases of current instructions having two or three syllables in which digits 8, 9, 16 and 17 are all zero:

$$\overline{D8} \cdot \overline{D9} \cdot \overline{D16} \cdot \overline{D17} \cdot \overline{W} \cdot \overline{X} \cdot \overline{SKIN} \cdot \overline{C1} \cdot [C3/C2]$$

in the prototype: $\bar{C} \cdot \bar{D} \cdot \bar{E} \cdot \bar{F} \cdot \bar{W} \cdot \bar{X} \cdot \bar{I} \cdot \bar{y} \cdot (\bar{\alpha} + \bar{\beta})$

2.5 The Spill Condition

Instructions are fetched from INB0 or INB1 as described until the NIA register specifies an instruction, part of which, or all of which, is not contained in the buffer. This condition (the "spill" condition) is detected after the syllables specified by the NIA register have been transferred to the INS; it causes the next instruction word to be fetched from the Main Store (provided that the "other" INB is not being used by Arithmetic Control) and arranges that the number in the NIA register is not incremented. The specified syllables are brought down to the INS again when the new word has been fetched, and the program continues.

The spill condition is generated (drg. K78031/9489) when the NIA register addresses a syllable in the "other" INB or syllabic location 5 in the buffer currently in use or when it addresses syllabic location 4 and the next instruction is a three syllable instruction. The NIL (Next Instruction Length) flip-flop is set if the next instruction contains three syllables.

$$\text{Spill} = (\text{NIA D0} \neq \text{CIA D0}) / \text{NIA D1}(1) \cdot \text{NIA D3}(1) / \text{NIA D1}(1) \cdot \text{NIL}(1)$$

$$= (d \neq c) + c(a + z)$$

Now when the new instruction word has been fetched and the specified syllables are transferred to the INS for a second time, the spill condition is still present. In order to avoid action being taken at this time because of "spill", inspection of the "spill" level is prevented by setting the SPIN (Spill Inhibit) flip-flop during the instruction word fetch sequence. This flip-flop is reset when the program continues (when the number in NIA is next incremented) provided CIA does not contain a count of 4 or 5 - see CE/SK F 5863.

2.6 The NIL Flip-Flop (Flip-Flop Z)

The Next Instruction Length flip-flop is set when the instruction about to be obeyed contains three syllables. It is used as a condition in the generation of the SPILL level; it also chooses between two sets of initial actions which take place at the start of all operations. If NIL is set, these actions include transferring any IS address in the instruction from the INS to the QBM register.

NIL (Z) is shown on drg. K78031/9489. It is set at the start of each operation from digit 8 or 16 of the INS or from the W or the Y flip-flop, depending on whether the count logic specified a count of 1, 2, 3 or 4. It is set from D0 of the INS after a new instruction word has been fetched because of a "Jump" operation. It is set from D0 of INS if the first syllable of the instruction about to be obeyed is the first syllable of an instruction word.

2.7 Generation of "Function" Levels

Details of the KDF 9 Machine Code are provided in Appendix 2 to Report K/GD.y.73. The digits which make up the "Operation" code for each of the 46 basic sequences are listed in fig. 10 of this report. These digits are decoded as shown on drg. K-/9490 and generate one of the 46 "function levels". These levels are used, as will be explained, to define the sequence which the sequence unit must execute.

3.0 THE SEQUENCE UNIT

Actions which occur in Main Control, such as the transfer of syllables from an IWB to the staticiser, are performed at the command of pulses emitted from the sequence unit. These "control pulses" are derived from P1 or P2 clock pulses; one control pulse may cause several basic actions to occur.

The sequence unit consists of two sets of transformers, the primary windings of each set being wired in an 8 x 8 matrix as shown in fig. 11. Associated with each matrix is a 6-bit register, the contents of which are decoded and used to select a transformer. The selected sequence unit transformer (SUT) is supplied with P1 pulses, if it is in one matrix, or P2 pulses if it is in the other. Each SUT carries six secondary windings which generate a 6-bit binary pattern. When the P1 matrix is pulsed, this pattern is set in the register controlling the P2 matrix. The pattern generated by the selected SUT in the P2 matrix is used in a similar manner. Each SUT also carries two windings which generate control pulses which are used to perform a number of basic actions. The group of actions which occur as a result of pulsing one transformer is known as a "micro-operation (MOP)" and the number which selects the transformer is known as a "micro-instruction code (MIC)". The 6-bit registers are called "micro-instruction staticisers (MIS)".

The operation specified by the instruction in the INS (or, rather, that part of the operation for which Main Control is responsible) is performed by a sequence of MOPS, one MOP being executed at every P1 or P2 clock time. The transformer windings which specify the next MI code (the MMIC windings) are arranged to select the transformer which will generate the command pulses required to perform the next MOP in the sequence.

The "output" ends of the MOP windings may be connected to Line Drivers, to Pulse Amplifiers or directly to a logic element. The input ends are connected to a 'ONE' or 'high' level if the command pulse is required to take action. The "output" ends of the MMIC windings are connected to "complementing circuits" which set or reset the flip-flops in the appropriate MIS. If the input of a MMIC winding is connected to a 'one' level (-2v in this case) the complementing circuit to which that winding is connected sets its associated flip-flop. If the MMIC input is not so connected, the flip-flop is reset.

The sequence unit can be considered as a store containing a number of "wired-in" micro-programs (the sequences). The micro-program to be executed is specified by a "function level" which is generated, as already described, by the operation code of the instruction occupying the INS. These function levels are applied to the NMIC windings of the SU transformers. In this way a transformer may generate different MI codes depending on the function level which is currently 'high'. If the same MI Code is required whatever sequence the transformer is used in, a constant -2v level is used instead of a function level.

The range of micro-instructions available must, of course, contain some which have the ability to discriminate; some transformers must be able to generate more than one MI Code during a particular sequence, the code generated depending on the state of some part of the computer. The possible MI Codes which may follow a "discriminating" MIC are chosen so that any of the codes may be generated from a basic code simply by changing certain '0' digits to '1' digits. These extra digits are called "masks". The function level specifies the basic code and "mask levels", which can be changed during the course of a sequence, specify the masks.

It is also convenient to make the generation of certain command pulses conditional upon the state of the machine. The "input" ends of these MOP windings are connected to "IF levels". The command pulse induced in such windings is effective only if the IF level controlling it is high.

Packages

Each SU matrix consists of an 8 x 8 matrix of selection lines to which are connected the primaries of the SU transformers. Eight of the selection lines are controlled by D.C. gates and the other eight by large pulse amplifiers.

The SU transformers each carry nine windings: the primary, two MOP secondaries and six NMIC secondaries. They are mounted on two types of package - the 'large' (K552) and the 'small' (K551). The "large" packages bear two transformers. On these packages each primary winding and each MOP winding is brought out to separate pins; the "input" end of each NMIC winding is brought out to a separate pin and the "output" ends of similar NMIC windings in the two transformers share a common pin. Twelve pins are allocated for "function" and "mask" levels. The "small" packages bear four transformers. On these packages, the "top selection" end of each primary winding is taken to a common pin and the "bottom selection" ends are brought out separately. The "input" end of one MOP winding is connected to a common pin, which is usually held at earth, and the "input" ends of the other MOP windings are brought out separately. The "output" ends, of course, are connected to separate pins. All the "input" ends of the NMIC windings have separate pins and the "output" ends of corresponding windings share a common pin. Eight pins are allocated to "function" and "mask" levels.

The P1 matrix consists of 13 packages type K551, occupying locations 1F7-1F19 in MC3, and 4 packages type K552, which are in locations 1F3-1F6. The P2 matrix consists of 15 packages, type K551. These are in locations 1H7-1H21.

The function and mask inputs may be connected to the NMIC windings on a package in many combinations. All these packages have, in fact, different combinations of connecting diodes.

3.1 SUT Selection

The SUT to be pulsed in each matrix is specified by the contents of a 6-bit "micro-instruction staticiser (MIS)". The outputs of these registers are decoded as two groups; the most significant three digits generate one out of eight selection signals and the least significant three digits generate a similar signal. It is thus convenient to write the contents of each MIS in octal and to number the two groups of eight selection lines from 0 to 7.

A model of the selection arrangements is provided in fig. 12. The more significant octal digit operates the "top" selection. Details of the selection logic are shown on drg. K-/20261.

Packages

The two 6-bit registers require three K255/30 packages which are situated in 1G15, 1G18 and 1G21 of MC 3. It should be noted that this combination uses mesa transistors. These flip-flops are set and reset by pulses from the complementing circuits which are contained on two K557 and two K558 packages in locations 1G16, 1G17, 1G19 and 1G20 of MC 3. The operation of these circuits is similar to the operation of the basic read amplifier described in Report K/GD.y.78. The strobe pulse is obtained by delaying a standard clock pulse (on the K558 packages) by about 50 n/usec. This allows the NMIC windings of the selected transformer to apply pulses to the complementing circuits before they are strobed.

The outputs of each of the registers are decoded using standard K446 packages (locations 1G14 and 1G22), the outputs of which are taken to D.C. Gates. The 'top selection' gates are on K369/20 packages (locations 1G23 and 1F24) and the 'bottom selection' gates are on K254/10 packages (1H22 and 1F20). The top selection gates have a maximum current output of about 30 mA whereas the bottom selection gates, using two transistors, can provide 150 mA pulses.

The top gates on each matrix control the secondary windings of three transfer transformers (a, b and c) on a K293/50 package (1G24 and 1F23), the primary windings of which are supplied with P1 or P2 clock pulses - depending on which matrix is being driven. The secondary windings drive large pulse amplifiers (K382/10) which supply the selected SUT.

3.2 Stop and Single Shot Logic

Certain units which operate more or less independently of Main Control may find it necessary to cause MC to wait, executing a "wait loop" in the micro-program currently set up in its sequence unit; it is also sometimes necessary for Arithmetic Control to stop the generation of command pulses by the MC sequence unit in order to initiate MOPS normally initiated by Main Control. It is also desirable for a maintenance engineer to be able to perform an operation "MOP-by-MOP", or a program 'operation-by-operation".

The stop logic is shown on drg. K-/20261. A switch is provided (area B1) which allows a program to be executed an operation at a time. Switches are also provided (area C7) which permit an operation to be executed one MOP at a time or allows one MOP to be performed repeatedly at high speed.

Instruction Complete Stop

The Stop Switch (area C7) allows a P1 clock pulse to set the STOP flip-flop. The output of this flip-flop then allows the next (0,6)P2 MOP pulse (which also transfers the next instruction from the IWB to IWS) to reset the GO flip-flop. The reset output of this flip-flop turns "off" the gate which controls the supply of P1 pulses to the "top selection" transformers. The supply of P2 pulses to the other matrix is not affected and the sequence unit continues to execute MOP (0,6)P2 at high speed (since the P1 matrix cannot generate a new MIC).

The 'normal' position of the P1 single shot key allows P2 clock pulses to reset the 'PIK' flip-flop, the output of which allows P1 clocks to reset the PIA and PIB flip-flops. If the P1 single shot key is operated, PIK is set and a P1 clock is allowed to set PIB. The next P2, therefore, sets the single shot P1 (SSP1) flip-flop. The next P1 is applied to the SU matrix and an operation is executed. The same P1 sets the GO flip-flop and the P1A flip-flop (which allows the next P2 to reset SSP1).

If the stop key remains set, the GO flip-flop is reset at the next (0,6)P2. If the P1 single shot key has not been released by this time, PIB will still be set but SSP1 cannot be set because P1A will still be set.

MOP by MOP Operation

The supply of clock pulses to the P1 matrix at 1 usec intervals can be cut off by closing the INH P1 switch (after the machine has been stopped). While the INH P1 switch is closed, clocks may be supplied one at a time by operating the single shot key.

The supply of clock pulses to the P2 matrix may be cut off by closing the P2 Inhibit switch. P2 clock pulses may then be supplied using a P2 single shot key which controls logic such as has already been described.

3.3 Wait Logic

If Main Control is required to wait because it wishes to use a unit which is not available for use (for example: the Main Store or the R Sequence Unit) the micro-program executes a "wait loop" in which no significant action is taken. The operation of the sequence unit is in no way affected. When a signal indicates that the required unit is available to Main Control, a mask level causes the micro-program to leave the loop.

However, if Arithmetic Control wishes to initiate actions normally initiated by Main Control (e.g. use of the Q Store) the MC Sequence Unit must be made ineffective. Arithmetic Control signifies its requirement by setting the NOG flip-flop and its sequence unit enters a wait loop, the exit from which is conditional upon the state of MC's GO flip-flop. Built in at suitable points in the MC micro-programs are discriminations on the state of the NOG flip-flop. If NOG is set, a P2 MOP is executed which transfers the gated output of the NOG flip-flop to the GO flip-flop. "GO" is therefore reset, cutting off the supply of P1 clocks and the MC sequence unit executes the P2 MOP repeatedly.

Resetting the GO flip-flop allows Arithmetic Control to continue. As soon as AC has finished the task which necessitated the interference with Main Control, it resets NOG. The P2 MOP which Main Control is performing promptly sets the GO flip-flop and Main Control continues.

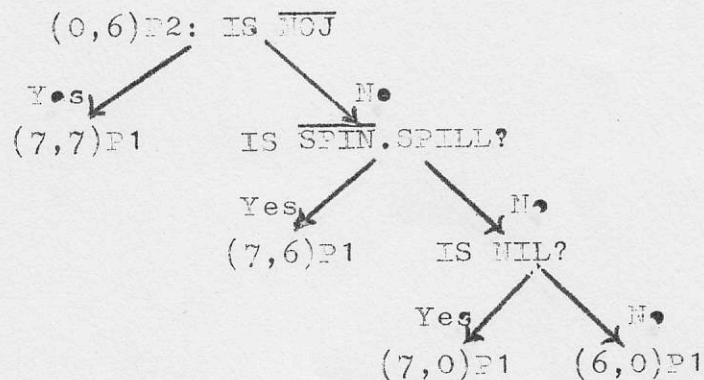
At various times, such as when an interruption is about to occur, Arithmetic Control must be allowed to obey all instructions which Main Control has already obeyed before MC takes any further action. The number of syllables by which AC lags behind MC is given by the difference between the CIA register and a similar register in Arithmetic Control. When the contents of these two registers are identical, Arithmetic Control has obeyed all the instructions completed by Main Control. This equivalence signal is used to reset a flip-flop known as "AGC2" which stops the AC sequence unit.

When MC has to wait for AC to catch up, it simply executes a P2 NOP which transfers the inverted outputs of the AGO2 flip-flop to the SO flip-flop. In this way the supply of clocks to the MC P1 matrix is inhibited until the AC sequence unit stops.

3.4 Examples of Next MIC Generation

(a) Branching

This example is taken from the "Fetch Next Instruction" sequence.



The SUF selected by (0,6)P2 is shown in Fig. 13.

The NOP "(0,6)P2" is common to all sequences. No function levels are therefore applied to the transformer secondaries; the basic next MIC code is generated by connecting the two most significant of the six IMIC windings to a -2v level. This causes (6,0) to be set in the P1 micro-instruction staticiser (MIS1).

If the NIL (Next Instruction Length) flip-flop is set, the M66 (mask) level is generated (see drg. K78031/2489-N.14). This mask lifts the third most significant IMIC winding and thus causes (7,0) to be set in MIS1.

If the "SPIN.SPILL" level is high the M61 mask level lifts the 3rd, 4th and 5th IMIC windings. MIS1 is therefore set to (7,6).

If the NOJ (No Jump) flip-flop is reset the M25 mask level will be high and (7,7) will be set in MIS1.

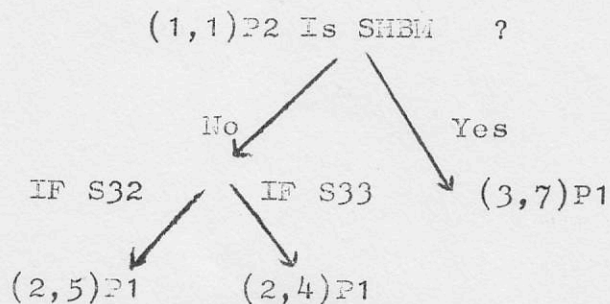
(b) Function Levels

This example is taken from the S34 and S35 sequences. The MOP "(6,2)P2" is common to both sequences and is used only in these sequences. In the S34 sequence the next MIC is (3,3) whereas in the S35 sequence it is (3,0). The SUT windings are shown in fig. 14.

The 2nd and 3rd NMIC windings are connected to -2v and thus a basic code of (3,0) is generated. If the S35 function level is high, the two least significant windings are also lifted.

(c) Combination of (a) and (b)

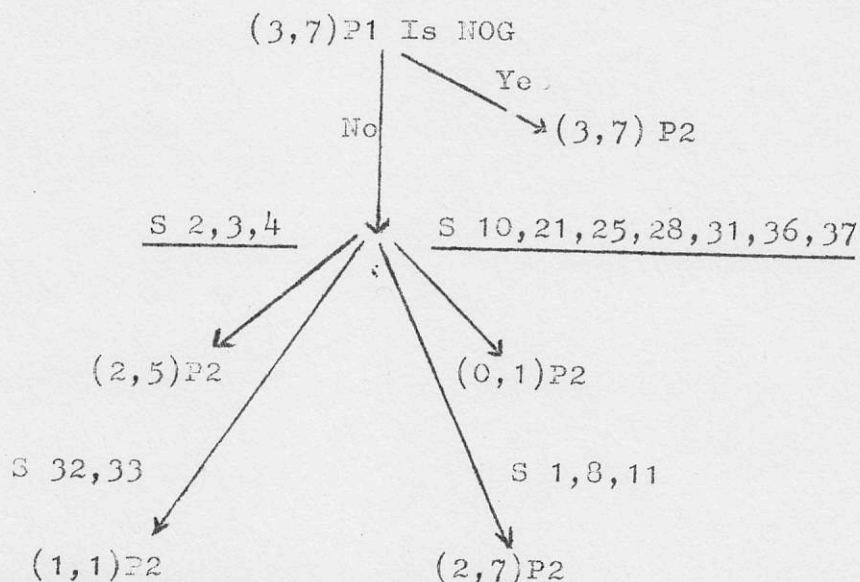
This example is taken from sequences S32 and S33.



The basic code generated is (2,4). However, if the S32 function level is high, the least significant NMIC winding is lifted and (2,5) is generated. If the SHBM flip-flop is set, the M2 level lifts the 3rd, 5th and 6th windings, thus generating (3,7) whether the sequence is S32 or S33.

(d) Function Combinations

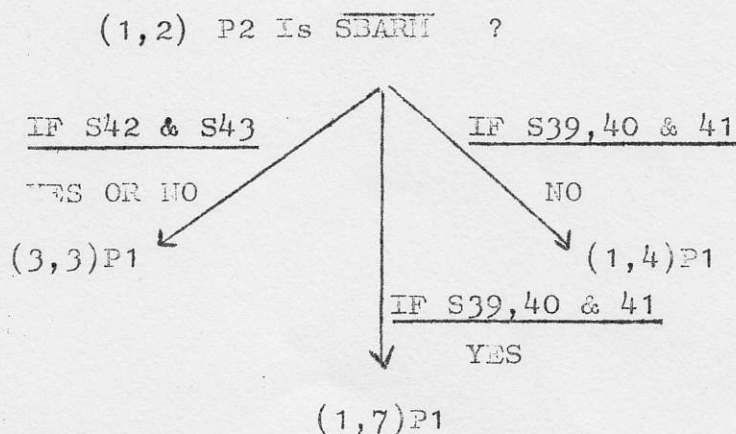
Where Next MI Codes are common to a number of sequences, the Function Levels concerned may be OR'ed to generate a "Combination Level".



The basic MIC generated by the (3,7)P1 SUT is (0,1). If the C2 level is high (drg. K 70831/20265) the 2nd and 4th MMIC windings are lifted and (2,5) is generated. If the C1 level is high the 2nd, 4th and 5th MMIC windings are high, producing (2,7), and if the C19 level is high (1,1) is generated. If the NOG flip-flop is set, the M12 level is high and (3,7) is the number set in MIS2.

(2) Same Mask, Different MIC

One mask condition may generate different MI Codes in different sequences.



There is no basic MIC which is generated irrespective of function level. If the C25 level is high (S39/S40/S41) the 3rd and 4th MMIC windings are lifted and (1,4) is generated. If the C27 level is high (S42/S43) the 2nd, 3rd, 5th and 6th windings are lifted, generating (3,3). If the M16 level is high, the two least significant windings are lifted and the mask (0,3) is generated. This mask alters the C25 MMIC to (1,7) but leaves the C27 MMIC unchanged.

3.6 Examples of Control Pulse Generation

(a) Unconditional Single Action Directly Connected

The (4,4)P1 MOP sets the OUT flip-flop which is part of the Reason-for-Interruption register. It is used in sequence S45 by means of which a program can transfer control to the Director. The pulse is generated by SU transformer (4,4) in the P1 matrix. Since it has to set only one flip-flop no power amplification is necessary; the MOP winding on this transformer is connected directly to the input of the flip-flop. The other end of the winding is held "high" by a constant bias level.

(b) Unconditional Single Action using an SPA

The MOP (4,6)P1 causes the number in the NIA register to be incremented by 3 while leaving the most significant digit unchanged. It is used in sequence S8. The pulse is generated by transformer (4,6) in the P1 matrix; it pulses the transformer which gates the output of the special count logic (see para. 2.3 of this report) to the NIA register.

The transformer's MOP winding is connected to a small pulse amplifier (M240/10 package) which provides the necessary power amplification. The SPA (K-/9491-F12) drives the primary winding of the appropriate transfer transformer (drg. K-/9482-B6).

(c) Several Actions from One Control Pulse

One MOP winding on transformer (0,1)P2 drives a Large Pulse Amplifier (K382 package) and a Small Pulse Amplifier (K240). The LPA generates an 'Add' pulse which transfers the outputs of the Q-Mill adder to the B register (K-/9484-F3). The SPA generates a pulse which clears the selected Fetch Buffer (K-/9266-M2).

This command pulse is generated in the following sequences: S21, S22, S23, S24, S28, S29, S30, S31, S36, S37.

(d) One Action from Several Control Pulses

A basic action may be included in any sequence by arranging to pulse the appropriate SU transformer. However, this is not always convenient because the other actions specified by that transformer may not be required. For example, certain sequences (S9, S34, S35) require the selected Fetch Buffer to be cleared but do not want the Q-Mill adder outputs transferred to the B register. These sequences pulse SUT (1,5)P1 which also provides an input to the SPA mentioned above.

(e) "Don't Care" Actions

Economies may be made by allowing unwanted basic actions to take place provided that they have no effect on the result of an operation. For example, Main Control MOP 30 P2 contains the action "S- L" alias "CIADO → SLJP". This action is required in the Short Loop Jump Sequence (S4) to note the buffer which contains the SLJ instruction. If an interruption occurs while the program is in a short loop the SLJP flip-flop is used to determine the Word address of the instruction to be obeyed on re-entry.

MOP 30P2 is also used in the other Jump sequences. No attempt is made to inhibit this action because if an interruption should occur while the program is not in a short loop, no reference is made to the SLJP flip-flop.

(f) Conditional Actions

An example of the use of conditional MOP windings occurs in the "Instruction Fetch" sequence. In this sequence the contents of the NIA register are transferred to the CIA register in preparation for the count which takes place at the beginning of each instruction. However, if a new instruction word is about to be fetched because of a SPILL condition, the contents of the CIA register must be left unchanged until the word has been fetched and the specified syllables are transferred to the INS for the second time.

The action takes place under the control of the control pulse generated in the conditional MOP winding of SUT (0,6)P2. The voltage level of this winding is earth only when the "IF" level No. 43 $[(SPIN \text{ } \overline{SPILL}/NOJ \text{ })]$ or, in the prototype, $(K + SPILL + J)$ is high. The generation of this level is shown on drg. K-/9479-K20.

4.0 THE Q MILL

The Q Mill consists of a 16-bit parallel adder and various associated registers. It also contains a counter which can subtract one from the contents of one of the registers. The Q Mill logic is shown on drg. K-/9484. It is used:

- (a) for adding the base address of the program in control to the relative MS addresses provided by that program and for checking that these addresses are within the limits for that program (see K/GD.y.77 para. 4.2),
- (b) for adding two relative addresses when address modification is used,
- (c) for loop counting,
- (d) for adjusting the NIWA (Next Instruction Word Address),
- (e) for modifying the contents of the QS Buffer,
- (f) for temporary storage during the manipulation by Main Control of various groups of digits.

The registers directly associated with the adder are the QBI and the A registers (which provide the two numbers to be added) and the B register (which receives the sum). The counter is attached to the QBC register.

The QBC, QBI and QBN registers, which are here considered as separate registers within the Q-Mill, together form the Q Store buffer register. The other registers associated with the Mill are the NIWA (Next Instruction Word Address) register and the BA (Base Address) and NOL (Number of Locations) registers (in time sharing machines). MS addresses required by the Main Store and the Input-Output Block are provided by the B register. Transfer paths also exist between certain of the Q-Mill registers and the Jump Buffer, the INS, the Shift Buffer (SHB) and the NIA register. Data words may be transferred between the Q Buffer and the Arithmetic Unit.

4.1 The Adder

The Q-Mill Adder is required to add two 16-bit binary numbers which are available in flip-flop registers. It consists of sixteen "Carry Propagation" packages (K239/10) and two "Carry Skip" packages (K237/10). Logic representations of these packages are provided in figs. 15 and 16.

Each Carry Propagation package is required to add two binary digits, plus any carry from a previous stage, and to produce a sum digit and a carry digit. The digits to be added are supplied to each package as d.c. levels; the carry input digits are supplied as pulses from the previous stage, or, in the case of the least significant stage, from an "Add" control pulse which is gated by the state of a Carry (CAR) flip-flop. The sum output of each stage is a pulse which sets or resets a B register flip-flop. The carry output of each stage is a pulse which provides the carry input to the next stage.

The operand inputs to each package (the d.c. levels from the QBI and A registers) are applied to logic which generates "digits equivalent" or "not equivalent" levels. These are applied to the windings of two double bead transformers so that, in each transformer, there can be no flux in one bead if the operand digits are not equivalent, and no flux in the other if they are equivalent. One of these transformers is pulsed by the 'carry' signal from the previous stage and the other is pulsed by the 'not carry' signal.

The rules for generating the sum digit are derived in fig. 17. If the operand bits are equivalent, and there is a carry from the previous stage, a '1' sum digit is required. If they are equivalent and there is no carry (a "not carry" pulse) from the previous stage a '0' sum digit is required. If the operand bits are not equivalent the rules for generating the sum digit from the previous carry pulse are reversed. The rules for generating the "carry" and "not carry" pulses are also given in fig. 17. A "carry" pulse is required if the two operand digits are '1' bits or if they are not equivalent and there was a carry from the previous stage. A "not carry" pulse

(no carry output) is required if the two operand digits are both '0' bits or if they are not equivalent and there was no carry from the preceding stage. The outputs dependent on a carry or not carry input are generated by driving a transformer with the appropriate pulse. The outputs which are independent of the previous carry are generated by pulsing a transformer with an "Add" control pulse.

It has been shown that the sum digits are gated out by "carry" and "not carry" pulses generated by the preceding stage. The more significant digits are, therefore, gated out after the less significant digits because of the carry propagation delay. This delay is reduced by using "carry look ahead" logic.

The design of the "Carry Propagation" packages is such that the "carry" or "not carry" pulses from a package are generated by an "Add" pulse if the two operand bits are equivalent. Thus "carry" is propagated through the adder until a stage having equivalent operand bits is reached; the carry outputs from this stage will have been generated, by the Add pulse, at the same time as the input to the least significant stage was generated. The maximum delay therefore occurs when the operands are not equivalent. This is reduced by arranging to by-pass the middle six stages if the middle six operand bits are not equivalent.

The outputs of the fifth and eleventh "Carry Prop." packages are applied to "Carry Skip" boards as shown in drg. K-/9484; a logic representation of the skip package is provided in fig. 16. If the middle six digits are not equivalent, the first skip package passes a "carry" or "not carry" pulse, as appropriate, to both the second skip package and the sixth prop. package. The second carry skip package supplies the twelfth carry prop. stage.

The second carry skip package must also accept inputs from the 11th prop. stage in order to deal with cases where a carry (or not carry) is generated in one of the middle six stages by the Add pulse. When skipping occurs, this package will, therefore, be supplied from the previous carry skip stage, and, after a delay, from the 11th carry prop. stage. The pulse amplifiers to which these pulses are applied are designed not to respond to the delayed inputs.

The adder is operated by an "Add" pulse supplied by the MC sequence unit (K-/20266-J10) using LPA No. 1.

Subtraction is performed by negating the subtrahend, before transferring it to the operand register involved, and setting the CAR flip-flop.

4.2 The QBM Register

The QBM register is a 16-bit section of the Q Buffer. Its use as a buffer will be considered when the Q Store is discussed. The register is contained on 4 packages type K255/10 occupying locations 2B7, 2C9, 2D8 and 2E9 in cubicle MS2. The following transfers may be made:-

(a) Inputs:

- (i) B(0-15) $\rightarrow$ QBM(0-15)
- (ii) NIWA (13 bits) $\rightarrow$ QBM(3-15); CLR QBM(0-2)
- (iii) INS (8-11) $\rightarrow$ QBM(0-3)
- (iv) INS (12-23) $\rightarrow$ QBM(4-15)
- (v) INS (D4) $\rightarrow$ QBM(D3); CLR QBM(0-2)
- (vi) INS (2-4) $\rightarrow$ QBM(1-3); CLR QBM(D0)
- (vii) CLR QBM

(b) Outputs:

- (i) QBM(3-15) $\rightarrow$ NIWA
- (ii) QBM(3-15) $\rightarrow$ JB(3-15)
- (iii) QBM (D0) $\rightarrow$ LIV (RFI No. 11)
- (iv) QBM(0-15) $\rightarrow$ A(0-15)
- (v) QBM(0-14) $\rightarrow$ A(1-15); QBM(D15) $\rightarrow$ TLS

4.3 The A Register

The following transfers may be made to and from the A register. The constants are supplied by connecting the appropriate set or reset input of the flip-flops to '1' levels on bias boards type K346/20. The register is contained on 4 packages type K255/10 which occupy locations 2B17, 2C18, 2D18 and 2E18 in MC2.

(a) Inputs

- +1 $\rightarrow$ A (D15 only set)
- +2 $\rightarrow$ A (D14 only set)
- 1 $\rightarrow$ A (all flip-flops set)
- 2 $\rightarrow$ A (D0-D14 set)

- (i) $QBI(0-15) \rightarrow A(0-15)$
- (ii) $QBI(0-15) \rightarrow A(0-15)$
- (iii) $QBI(0-14) \rightarrow A(1-15)$
- (iv) $\bar{B}(0-15) \rightarrow A$
- (v) $BA(10 \text{ bits}) \rightarrow A(1-10)$ 0's $\rightarrow A(0, 11-15)$
- (vi) $NOL(10 \text{ bits}) \rightarrow A(1-10)$ 1's $\rightarrow A(0, 11-15)$

(b) Outputs

$A \rightarrow QBC$

$A \rightarrow B$

$A \rightarrow QBI$

4.4 The BA and NOL Registers

The BA (Base Address) register is used to store the base MS group address of the program currently in control (K/GD.y.77, para. 4.2). It has a capacity of 10 bits and is contained on three K255/20 packages in locations 2C2, 2D2 and 2E2 of MC2.

The NOL (Number of Locations) register contains the amount of storage space allocated by Director to the program currently in control. It also has a capacity of 10 bits. The NOL register is used when performing the Lock-in Violation (LIV) check. In this check the relative MS address being used is stored in QBI and NOL is transferred to the A register using the transfer path detailed in 4.3(a) (vi). If D0 of QBI is '1' (the relative address is negative) the LIV flip-flop is set. This ensures that information will not be transferred to or from a MS location having a smaller address than the 'bottom' limit of the area of store allocated. The contents of QBI and NOL, with a 'carry' bit, are then added and LIV set if the sum is positive. This ensures that no address greater than the 'top' limit is used.

The BA and NOL registers are set from the W1 register by the Director instruction "=K1" which also sets the Priority Level flip flops in IO control. When this instruction is obeyed, the MC sequence unit generates a "Fetch or Store Director" pulse. If D9 of INS is '1', this pulse causes the following transfers to take place:

W1 (24-33) $\rightarrow$ NOL

W1 (34,35) $\rightarrow$ PL

W1 (38-47) $\rightarrow$ BA

If an interruption occurs, the IC sequence unit generates a pulse which clears BA, sets in NOL the highest group address which the Main Store can accept, and sets the NIFF (No Interruptions Flip-Flop) which indicates that the Director is in control. The constant to be set in the NOL register is "wired in".

The following transfers involving the BA and NOL registers can be made:

$\overline{\text{NOL}} \rightarrow A$

BA $\rightarrow A$

4.5 The B Register

The B Register is a 16 bit temporary storage register. The register is mounted on four K255/10 packages situated in locations 2E4, 2D4, 2C4 and 2B3 of MC2. The outputs of this register drive emitter followers mounted on six K361/10 packages. The following transfers are possible:

Input:

JB (D3) $\rightarrow$ B (D3); CLR B (0-2)

JB (0-3) $\rightarrow$ B (0-3)

QBC $\rightarrow$ B

ADDER $\rightarrow$ B

A $\rightarrow$ B

Output:

B (0-15) $\rightarrow$ QBM (0-15)

B (1-15) $\rightarrow$ MSAR

B (1-15) $\rightarrow$ SBAR

B ($\overline{0-15}$) $\rightarrow$ A

B ($\overline{D0}$) $\rightarrow$ LIV (RFI No. 11)

B (1-15) $\rightarrow$ LB (LOCKOUT BUFFER - ADDRESSES LO STORE
IN IO CONTROL)

4.6 The QBI Register

The QBI register is a 16-bit section of the Q Buffer; its use as a store buffer will be considered when the Q Store is discussed. The register is mounted on four K255/10 packages situated in locations 2B20, 2C21, 2D21 and 2E21 of MC2. The following transfers to and from this register are permitted, the constant being supplied by a diode board in location 2E23.

Inputs:

$[-1] \rightarrow \text{QBI}$
 $\text{NIA}(1-3) \rightarrow \text{QBI}(0-2)$
 $\text{A} \rightarrow \text{QBI}$
 CLR QBI

Output

$\text{QB}^- \rightarrow \text{A}$
 $\text{QBI}(0-2) \rightarrow \text{JB}(0-2)$

4.7 The QBC Register

The QBC register is also part of the Q Buffer and is mounted on K255/10 packages in locations 2A4, 2A9, 2A16 and 2A21. Associated with it is the counter described below. The following transfers are permitted, the constant being supplied by a diode board in location 2A12.

Input:

$[-1] \rightarrow \text{QBC}$
 $[\text{QBC}-1 \rightarrow \text{QBC}] \text{ (count down)}$
 $\text{A} \rightarrow \text{QBC}$
 CLR QBC

Output:

$\text{QBC} \rightarrow \text{B}$
 $\text{QBC}(8-15) \rightarrow \text{SMB}$

4.8 The Counter

The counter is used for subtracting one from the number in the QBC register. It consists of a number of d.c. gates which are "slugged" so that their outputs remain at a set level for a short time after their input levels have changed. The inputs to these gates are provided by the flip-flops of QBC. Their outputs are applied to transfer transformers which are connected to the inputs of the Q^- flip-flops. The counter consists of four K374 packages (locations 2A3, 8, 15, 20) and a

K364/40 diode board (2B9). The transfer transformers are on K251 packages situated in locations 2A2, 7, 14, 19.

Each counter package carries four stages. A truth table for four stages is provided in fig. 18. Logic representations of the packages are provided in fig. 19.

5.0 THE Q STORES

A Q Store is a word organised, fast, random access core store which has a maximum capacity of fifteen 48-bit words. It has a minimum read/write cycle time of $1\frac{1}{2}$ usec of which $\frac{1}{2}$ usec is used for location selection, $\frac{1}{2}$ usec for the read current pulse and $\frac{1}{2}$ usec for the write current pulse. Words are transferred between the cores and a 48-bit flip-flop register known as the Q Buffer (QB), the core location involved in such transfers being specified by the contents of a 4-bit register known as the Q Store Address Register (QSAR). The fifteen store locations correspond to addresses 1 through 15 address 0 causes 48 zeros to be transferred to the QB if a fetch operation is performed. In a time sharing machine there are four stores, one buffer and one address register being common to all four. The particular store required by a program is selected as part of the normal location selection by decoding two extra address digits. These digits are set in a program number register by the Director (using the instruction "= K3") before the Director transfers control to a program; they remain unchanged while the program is in control.

Main Control (or IO Control) can only specify complete read/write cycles when using the Main Store. On the receipt of a "start" pulse, the MS provides its own sequence of timing pulses and proceeds independently. The Q Store, however, is entirely dependent on external sequence units for its timing. Main Control may provide an address at any time and either MC or AC may cause the generation of a read or a write current pulse at any time. A read pulse is not necessarily followed by a write pulse.

Any address stored in the QSAR is decoded immediately and selects the appropriate read/write drive lines in the Q Store being used. Any word stored in the QB selects the digit drivers which must provide current if a "write" command is received. Associated with the QSAR is another 4-bit register known as the QS Address Buffer (AQB) which is used as a temporary store for any QS address required by AC. (See K/GD.y.77, para. 2.4). The AQB has a marker flip-flop (AQBM) which is set when MC supplies AQB with an address and which is reset when AC uses that address.

An introduction to word organised stores is provided in K/GD.y.78.

5.1 Sequences Involving the Q Store

The Q store instructions fall into five categories:

- (i) "QS Fetch" operations (S21-S24)
- (ii) "QS Store" operations (S26-S27)
- (iii) "Q to Q" operations (S20)
- (iv) "Q MOD Q" operations (S12-S19)
- (v) "AU MOD Q" operations (S28-S31)

In all except "QS Store" and "AU MOD Q" sequences, the Q Store is operated entirely by Main Control. In "QS Stores" it is operated by Arithmetic Control and in "AU MOD Q" sequences it is operated first by MC and then by AC.

In all cases except in S20, when MC operates the store it fetches or stores complete 48-bit words; any manipulation of the three 16-bit sections is done in the Q Mill. However, during S20, it may be required to transfer one 16-bit section from one QS location to another. This operation is facilitated by providing three flip-flops, known as the QS Read Strobe Inhibit (QRSI) flip-flops, each of which controls the gating of strobe signals to one section of read amplifiers. Any section can, therefore, be transferred by first reading the source location to the QB (and rewriting it) and then reading the destination location - having inhibited the strobes for the section to be transferred. The complete 48-bit word is then, of course, rewritten in the destination location. Main Control resets the QRSI flip-flops as a standard action at the beginning of every sequence.

Arithmetic Control may be required to store all three or only one of the 16-bit sections. It stores one section by making use of the QRSI flip-flops as Main Control does. However, because MC is expected to proceed with the program, which may include S20 operations, without waiting for AC, temporary storage must be provided for the QRSI bits required by AC (since they cannot be accommodated in the Arithmetic Instruction Staticiser).

The C, I and M flip-flops are provided for this purpose. When MC executes a sequence in the range S26-S31, it transfers an address to the AQB, sets AQBM and transfers the QRSI bits from IMS to the C, I and M flip-flops. When AC uses the store, it transfers the contents of the C, I and M to the QRSI flip-flops; it also uses the C, I and M flip-flops to select the section, or sections, of QB to which data will be transferred from the WI register. The read from the cores to the QB and the transfer from the WI register to the QB take place at the same time (see fig. 20). If all three flip-flops are set, strobes to all three QS sections are inhibited and the complete 48-bit word is transferred from WI. If only one flip-flop is set, the 16 least

significant digits in WI are transferred to the selected section of QB and the strobes for that section are inhibited. The contents of the QB are then written into the Q Store.

The sequence S27 requires the Counter (C) and Modifier (M) sections to be cleared and the Increment section to be reset to "+ 1". (See Programming Manual, section 3.4.1 - "= RCq", "=RIq" and "=RMq"). The location is cleared simply by reading its contents. This is made effectively to reset the QSI section to "+ 1" by wiring the "1" output of the QSI (D15) read amplifier to the "reset" input of QBI (D15) and its "0" output to the flip-flop's "set" input. Where, as is normal, a read follows a write, this has no effect on the operation of the store; the I(D15) bit is simply stored inverted. However, in the S27 sequence, MC performs a read from the selected location before handing over to AC which performs another read followed by a write.

5.2 Q Store Interlocks

Main Control may not

- (a) use the Q Store if AC is using it
- (b) perform its part of a "QS Store" or "AU MOD Q" operation if AC has not yet obeyed the last such instruction
- (c) fetch and use the contents of a location if it has already obeyed an instruction, not yet obeyed by AC, which required fresh data to be stored in that location.

The method by which AC stops MC when it wishes to "borrow" the Q Store has been described in para. 3.3.

The second MOP in the "QS Store" and "AU MOD Q" sequences discriminates on the state of the AQBM flip-flop and transfers the bit in this flip-flop to the NOG flip-flop. If AQBM is set, the sequence is diverted to the (3,7)P2 MOP which transfers the negated NOG bit to the GO flip-flop. When the NOG flip-flop is reset by Arithmetic Control, Main Control again attempts to obey the instruction in the INS.

The first MOP of all sequences in the range S12 to S24 ("Q MOD Q", "Q TO Q", "QS Fetch"), transfers the QS address from the INS to the QSAR. The second MOP discriminates on the state of the QLO (QS Lockout) (NOG) signal. This signal is "high" (i.e. "1") if the contents of the QSAR and AQB are identical and the AQBM flip-flop is set. This condition causes the NOG flip-flop to be set and the sequence to be diverted to the (3,7)P2 MOP as before.

5.3 QS Timing

A QS read is initiated by a $\frac{1}{4}$ usec control pulse provided by SPA No. 82 (fig. 20). This pulse is applied to the Read drivers and to a widening circuit which also drives the Read drivers via a D.C. gate. In this way a $\frac{1}{2}$ usec input signal to the Read drivers is obtained.

The output of each widener is connected to a strobe pulse generator which, if it is not inhibited by one of the QRSI flip-flops, provides a $\frac{1}{4}$ usec pulse to the Read Amplifiers at about $\frac{1}{4}$ usec after the start of the current drive.

A QS write is also initiated by a $\frac{1}{4}$ usec control pulse, this pulse being provided by LPA No. 50. A $\frac{1}{2}$ usec pulse is generated by means of a widener, and this pulse is applied to the write drivers and digit drivers.

Both SPA 82 and LPA 50 may be pulsed by Main Control or Arithmetic Control.

5.4 Storage

Each of the Stores uses three 16 x 16 matrices of Plessey S4M.050" cores for which the nominal drive is 500 mA-turns and the switching time is 1.3 usecs. This switching time is reduced to .5 usec by overdriving the cores, in both the read and the write pulses, with 600-650 mA-turns. The arrangements which allow this overdriving are shown in fig. 21.

Each core is threaded by the read-write winding (4 turns) the digit winding (6 turns), the bias winding (1 turn) and the sense winding (2 turns). When the contents of a core location are to be transferred to the buffer register, 90-100 mA is passed through the selected drive windings in a direction which assists the bias current and which magnetises the cores in the selected location in the '0' direction. A voltage is induced in the sense winding linking any core in which the direction of magnetisation is reversed and this voltage is applied to the appropriate read amplifier when the strobe signal is present. When the contents of the QB are to be written into a core location, a similar current is passed through the selected read/write windings in the opposite direction, opposing the bias current. This produces a resultant magnetising force in each core which is ~~insufficient~~ insufficient to switch the core to a '1' state. The additional magnetising force required to switch a core is applied to selected cores by current in the digit winding.

The direction of current flowing during the read and write current pulses is reversed not by a switching technique as in the Main Store but by the use of separate current drivers for the two pulses (fig. 22). Three sets of read/write drivers are provided - arranged so that a driver provides current through only 16 cores in the selected location. In time sharing machines, the digit and sense windings of all four stores are connected in series. The digit drivers must therefore be capable of driving through 64 cores (of which only one switches).

Location 0 is arranged to provide zeros by arranging that the read/write drive wire for this location does not pass through any cores. However, because of the cross connected wiring on the outputs of the QBI(15) read amplifier, a '1' input must be provided for this amplifier to give a '0' output. The drive wire is therefore arranged to pulse a transformer which pulses the QBI(15) read amplifier.

5.5 Selection

The Store to be used in any program is selected before control is transferred to that program. The store is specified by the two digits in the program number register, the outputs of which are decoded by diode matrices to generate one out of four selection signals. This signal is amplified as shown in fig. 23. The location involved in a QS transfer is specified by the contents of a four digit address register. The outputs of this register are decoded to generate one out of sixteen signals and this signal is amplified as shown in fig. 23.

A model of the location selection arrangements is shown in fig. 22. The store selection signal operates switches such as SA, SB and SD and the location selection signal operates switches such as SC. These switches are transistor switches which will pass current in either direction. Switches such as SB and SC are combined in one transistor which requires two inputs to turn it "on".

Switches like SA and SD are provided so that the current drivers will not have to charge the wire capacitance of stores not selected by the Program Number Register.

The transistor corresponding to switches SB and SC has two inputs-location selection and store selection inputs. The store selection levels are applied in order improve the switching of the location selection gates. The diode output of each location selection gate is applied to four transistors, one in each store. The direct outputs of each store selection gate is applied to all of the switches (such as SB and SC) in one store.

When the location selection transistor is conducting (not selected) it will supply only the switch in the selected store.

There is no digit or sense winding selection. If a digit winding must pass current, its driver is selected by the contents of the appropriate digit in the buffer register. The digit and sense windings of all the Q stores are connected in series.

5.6 The Read and Write Current Drivers

There are three QS Read drivers and three Write drivers. They are similar to the Main Store read/write drivers in that they consist of a constant current generator, the output of which may be made to flow through a load by operating a switch which normally short-circuits the load. These drivers are required to apply $\frac{1}{2}$ usec. pulses of about 90-100 mA. The drivers are mounted on three K261 packages, each package containing one read and one write driver.

Referring to drg. K-S23261, the read current generator consists of a transistor (T3A) connected via a resistor to -20v. The voltage drop across the resistor is compared with the drop across a 3v Zener diode and the difference operates transistor T4A which controls the base current in T3A. The output of the current generator flows through choke L1A which further assists in maintaining a steady current.

Transistor T2A corresponds to switch SF in fig. 22 and output A corresponds to point A in the figure. T2A and T1A are wired as a "super~~alpha~~ pair" and may be considered as one transistor with a very high gain. These transistors pass about 100 mA when both read and write switches are closed. When the write switch (T1B, T2B) is opened, the read switch passes twice as much current and vice versa. The read switch is opened (the transistors are turned off) by a "high" input to the base of T1A. Diode D3A provides a path for the removal of charge from the base of T2A when T1A is non-conducting; it provides for fast switching. While the input is "high" D2A conducts and energy is stored in L2A. When the input drops, this energy transfers charge to the base of T1A and assists the closing of the switch. The choke L3A helps to maintain a steady current through the cores during the pulse period. Some of the current provided by the driver is used to charge the capacitance associated with the drive lines; this charging current gradually decreases, allowing more current to flow through the cores. This increase is corrected by diverting an increasing amount of drive current to L3A throughout the duration of the pulse.

The diode D5A protects against any attempt to turn off T1A and T2A with the load unit not connected. The 3v Zener diode holds D5A at about -18v. The -3v rail permits the circuit to be operated by a flip-flop working between the levels 3 and -5.

The Write current driver is a "mirror image" of the read current driver.

5.7 The Selection Gates

The selection gates consist of single transistor switches which, when turned "on", may conduct in either direction according to the voltage levels existing in the circuit of which the switch is a part. The principle of such a switch is shown in fig. 24. The transistors used in such switches are of symmetrical construction, both junctions being made capable of the power dissipation associated with a collector and the gain of the transistor being made the same in both directions.

The read/write current drive circuits include three of these switches in series with a selected core location. One transistor corresponds to SA in fig. 22, one performs the combined functions of SB and SC and one corresponds to SD. If the read switch (SE) is opened, and the write switch (SF) is closed, point A is at about -15v and point B at -3v.

The circuit diagram of a 2 Store selection switch is shown in drg. K-S23406. The switches SA and SD are on K406/30; the location selection switches are on K406/10 packages. The input to D1 on the K406/10 packages is a location selection level taken from a K369/60 (Mesa) DC gate driven by the decoded 4-bit address. There are 16 such levels, each connected to all four stores. The input to D2 is a store selection level taken from K369/80 DC gate driven by the decoded 2-bit Program Number. When a switch is expected to conduct, its emitter will be at a voltage a little more negative than -3v and its collector at a voltage a little more positive than -15v. If either of the DC gates controlling the switch is turned "on", the base of the bilateral switch will be held just below earth and the transistor will be cut off. If both DC gates are "off" the selection gate is turned "on".

The D3/L2 circuit is a "current squirter" similar to the D2/L2 circuit on the current drivers. The choke L1 maintains constant base current even when the back voltage from the cores lifts the base.

As can be seen from the circuit diagram, the K406/30, which has only one controlling input, is a simplified version of the circuit just described.

5.8 The Digit Drivers

The digit driver packages (K397) are used for two purposes: for timing and generating the digit drive current and for timing the read and write currents. Each circuit has two parts. The first part is a pulse widener which provides a $\frac{1}{2}$ usec pulse and the second is a type of DC gate. When used as a digit driver, the cores provide the load for this DC gate.

The circuit diagram is on drg. K-S23397. The circuit associated with T1 is the pulse widener. Normally T1 is conducting. A high (P pulse) input to D1 cuts T1 off and the current in the primary of L1 is transferred to

the L1/C1 tuned circuit. The condenser is charged and the emitter voltage of the transistor falls and then rises as the L1/C1 circuit oscillates. The high input to D1 is removed after $\frac{1}{4}$ usec. The tuned circuit is arranged to return the emitter of the transistor to a potential more positive than that of the base after $\frac{1}{2}$ usec. When this occurs the T1 again conducts.

The Diodes D4 and D5 clamp the base of T1 so that its voltage level will not go negative when T1 is turned off. If this were allowed the pulse width would change. D4 is biased at about $-\frac{1}{2}$ v by D5 so that the circuit will not rob the transistor of base current when T1 is conducting. D6 clamps the overswing of L1 to avoid damaging the transistor.

When used as a digit driver, the "DC gate" part of the circuit is controlled by a flip-flop in QB. The flip-flop contains a '0' digit, D9 is held high and T2 remains cut off.

5.9 The Read Amplifiers

The read amplifier circuit is the basic circuit described in report K/GD.Y.78. The voltage induced in the sense winding applied to this circuit is -200mV. T24 is maintained at -100mV.

5.10 The Strobe Generator

There are three strobe generators, each priming 16 of the 48 read amplifiers (one section of the Store). The strobe generator produces $\frac{1}{2}$ usec pulse about $\frac{1}{4}$ usec after a positive going signal provided by the "digit driver" package which times the read current pulse. The generation of the strobe may be inhibited by the appropriate QRSI flip-flop (fig. 20).

The circuit diagram is shown on drg. K-Z 23398. The circuit consists of a "Miller" circuit driving a pulse amplifier. The transistor T1 is normally cut off. When the collector of the widener in the "digit driver" goes negative, T1 conducts (provided that its base is not held high by a QRSI flip-flop). A Miller run-down occurs as C1 discharges. When the collector of T1 reaches -2., T2 is cut off. The current in the primary of L1 is transferred to the secondary which, with C2, forms a tuned circuit in the base of T3. Transistor T3 therefore conducts for a period determined by this tuned circuit.

6.0 THE JUMP NESTING STORES

The principle of operation of a "nesting" store is explained in sect. 1.2.1 of the Programming Manual; the use of a Jump Nesting Store (JNS) is explained in sect. 1.8. In a time sharing machine there are four JNS, the particular store to be used during any program being selected by a 2-bit Program Number. This number is set into the Program Number Register (PNR) by the Director before control is transferred to the program.

The selected JNS is used, when a program is interrupted, to store the addresses specifying the first instruction to be obeyed when the program is re-entered; it is also used during "Jump to Subroutine" operations, to store the addresses specifying the "Jump" instruction. Addresses may be set in, and fetched from, the store under program control.

A subroutine is entered by means of an S1 sequence and left by means of an "EXIT (S8)" sequence. Transfer of control to the Director is accomplished by the "Interruption (SJ)" sequence and transfer of control from Director is accomplished by means of the "EXIT D (S9)" sequence. A pair (MS and syllabic) of addresses may be transferred to the store by a "= LINK (S11)" operation, and addresses may be fetched from the store by a "LINK (S10)" operation.

The S1 sequence stores in the JNS being used, the addresses of the "Jump to Subroutine" instruction. The EXIT instruction, by which control is returned to the main program, contains a number which must be added to the addresses of the Jump instruction in order to obtain the addresses of the next instruction of the main program to be obeyed. This number is the number of half-words between the first syllable of the Jump instruction and the first syllable of the instruction to be obeyed. The S8 sequence fetches the addresses in the topmost cell of the JNS and increments them suitably, using the Q Mill. The modified addresses are then presented to the NIWA and NIA registers.

The Interruption sequence is entered if Main Control detects that a Reason for Interruption (RFI) flip-flop has been set. It stores in the topmost cell of the JNS, nesting the others down in the process, the addresses of the instruction which must be obeyed when control is returned to the program. If the RFI is detected during an operation which is successfully completed before the interruption occurs, the addresses of the next instruction in the program are stored. If the current instruction of the program cannot be completed before interruption occurs, then the addresses specifying this instruction are stored. The sequence S9 simply presents to the NIWA and NIA registers the addresses fetched from the topmost cell of the JNS.

During the sequence S11 ("= LINK"), Arithmetic Control transfers digits 32-47 of the WI register to the topmost cell of the JNS, nesting the store down in the process. During the sequence S10 (" LINK"), Main Control transfers the contents of the topmost JNS cell to one of the Fetch Buffers. Arithmetic Control completes the transfer to the WI register as in other "Fetch" operations.

Each JNS consists of a 16 x 24 core plane (of which 16 x 16 cores are used). The sense windings of all four stores are connected in series and taken to 16 read amplifiers which supply a 16-bit flip-flop register known as the Jump Buffer (JB). The "cells" of a store are addressed in a serial fashion by means of counters which add or subtract one from the previous address when the store is nested down or up. There is one set of counters for all four stores, the decoded output of the appropriate counter selecting a location in the selected store only. The current driving and selection arrangements are similar to those used in the Q Stores. Nesting Up takes 1 μ s and Nesting Down takes 1 $\frac{1}{2}$ μ s.

The JB is normally considered as the topmost cell of the store; the "inaccessible" cells are locations in the core stack. However, since this register and the address counters are common to all four stores, their contents must be preserved elsewhere by the Director when control is transferred from one program to another. The contents of the JB are stored in the next vacant location of the appropriate core stack (this reduces by one the number of cells which programmers may use). The contents of the address counters are assembled, together with the Program Number, as one word in the W1 register and stored in part of the Main Store which is reserved for the use of the Director.

The Director stores the contents of JB by means of one of the instructions which will cause the JNS to nest down. It fetches the JNS and NS (Arithmetic Unit Nesting Store) address counters and the Program Number to the W1 register by means of an instruction "K7".

Before transferring control to another program, the Director sets the appropriate counters and the Program Number register from the W1 register by means of the instruction " = K3". This, of course, selects the core stack to be used. The contents of the JB are then set by one of the instructions which will cause nesting up to occur.

Because the contents of the JB are stored in the core stack when an interruption occurs, the number of JNS cells available to the programmer is reduced by one. The Director may also require the use of one cell in addition to this. Programmers should therefore restrict themselves to 15 cells. Logic has been provided which causes an interruption to occur if a program attempts to use more than the permitted number of cells or attempts to extract addresses from a JNS which is empty.

6.1 Operation of a JNS

A JNS is a word-organised store, the principle of operation of which is shown in fig. 25. The "cells" (locations) in the store are used sequentially.

If the JNS, including the buffer, is empty, the first nesting down operation writes '0's' into location 0 and the addresses to be stored are supplied to the JB. The third nesting down operation writes the contents of the JB into location 2 - and so on.

If three nesting down operations have taken place and a nest-up operation is called for, the contents of location 2 will be transferred to the JB. The readout is destructive and the data read are not regenerated. After a read (nest up), the location read from is therefore left empty. This eliminates the necessity, when writing, of performing a read operation simply to clear the selected location as in the Main Store. The stack is cleared initially by the Director which performs the requisite number of read operations before transferring control to a new program.

The selection problem of the 16 cell stack described could be overcome using a 4-bit address register operated as a reversible counter. In such a system, the counter would be cleared initially and thereafter counted up by one before any write operation (nesting down). The count would be decreased by one after any read operation (nesting up).

A disadvantage of this system is that time must be allowed for counting to take place before the write operations and after the read operations. This is overcome by using two 3-bit counters, of which one addresses the 8 odd numbered locations (JODAR) and the other addresses the 8 even numbered locations (JEVAR). The system is arranged so that during any operations one counter is used to address the stack and the other counts up or down as necessary. A flip-flop marker is used to indicate which counter should be used to provide the required address. A flow chart describing the operation of these counters is provided in fig. 26.

It should be noted that, after any operation:

If JTC (1): JEVAR = JODAR

If JTOG (0): JODAR = JEVAR - 1

JEVAR and the marker (as the least significant bit) together provide a 4 bit address indicating the next vacant cell.

6.2 JNS Address Counters

Both address counters are 3-bit flip-flop registers which are decoded and used to close location selection switches in the normal way. Each register can select one of eight of the 16 cells in the stack - thus at any time two locations may be selected. However the drive lines of the two groups of eight locations are supplied by separate sets of current drivers which are selected by the "Toggle" flip-flop already mentioned. Current is therefore provided in only one of the selected locations (see fig. 27).

As demanded by the flow chart in fig. 26, these two registers are connected by four transfer paths:

(i) $A \rightarrow B$

(ii) $B \rightarrow A$

(iii) $(B + 1) \rightarrow A$

(iv) $(A - 1) \rightarrow B$

These paths are arranged as shown in fig. 28.

Truth tables and Veitch diagrams are provided in fig. 29 and a logic representation of the counters is shown in fig. 30.

The flow chart shows that the choice of path depends on the following conditions:

- (i) path (i) is selected if the operation is "Nest Down (Write) and JTOG is not set
- (ii) path (ii) is selected if the operation is "Nest Up" (read) and JTOG is not set
- (iii) path (iii) is selected if the operation is "Nest Down" (Write) and JTOG is set
- (iv) path (iv) is selected if the operation is "Nest Up" (Read) and JTOG is set.

The path selection logic is shown in fig. 28.

As has already been stated, the address registers can be fetched and set by the Director. Only 4 bits are concerned: the contents of the JEVAR and the JTOG flip-flop. If the JTOG flip-flop is set, then the number in the JODAR must be the same as that in the JEVAR. If the JTOG is reset, the number in the JODAR is one less than the number in the JEVAR. The logic which allows the Director to set the counters is shown in fig. 31.

The Director transfers 1 bit to the JTOGB (JNS Toggle Buffer) and 3 bits to the JODAR; it also sets the TJEV (Transfer to the JEVAR) flip-flop. This flip-flop allows the next P1 to effect a transfer between the JODAR and the JEVAR. This P1 pulse also transfers the bit in the TJEV flip-flop to the CJOD (Count to the JODAR) flip-flop. If the CJOD flip-flop is set and the JTOG flip-flop is reset, one is subtracted from the number in the JEVAR and the result transferred to the JODAR.

6.3 Timing Logic

The operation of the JNS is under the control of two $\frac{1}{2}$ usec pulses: WR JNS (Write) and RDJNS (Read). The RD JNS generates a RDJOD or RDJEV pulse depending on whether the information to be transferred to the buffer is stored in an odd or an even numbered location. The WRJNS pulse generates either a WRJOD or a WRJEV pulse. An RDJOD pulse initiates the generation of a 1 usec timing pulse which primes the Read current driver connected to the odd numbered drive lines; it also generates a strobe pulse. A RDJEV pulse causes a timing pulse to be applied to the read current driver which supplies the even numbered drive lines. RDJEV also provides a strobe pulse. This logic is shown in fig. 32.

The WRJOD and WRJEV pulses apply similar timing pulses, delayed $\frac{1}{2}$ usec, to the write current drivers. They also time the bias (digit drive) current. This logic is also shown in fig. 33.

6.4 "Overspill" and "Underspill"

Logic is provided which causes the program in control to be interrupted if it attempts to overfill the JNS (overspill) or to extract addresses from an empty JNS (underspill). Logic is also provided which indicates when the JNS is empty; the signal provided by this logic is inspected when a "JrEJ" or "JrNEJ" (S7) instruction is obeyed (see Programming Manual sect. 3.3.5).

The core stack is addressed cyclicly by the two address registers and the "toggle" flip-flop as shown in figs. 26 & 34. The address registers are initially set to $(000)_2$ and $(111)_2$, as shown, and the toggle is reset. At this stage, when the JNS is empty, writing (nesting down) is permissible but reading (nesting up) is not.

If reading is attempted when the JNS is empty (addresses as at X) or if writing is attempted when the JNS is full (addresses as at Y), the NSP (NS or JNS Spill) reason for interruption flip-flop (RFI No. 10) is set.

Interruption will be inhibited by the NIFF if Director is in control and the JNS overspill signal will be generated if an attempt is made to use more than 15 cells (see para. 6.0.).

6.5 The Core Stack

The JNS is a word organised store which has a capacity of 16 24-bit words; it is used to store up to 16 16-bit words. The store has two address registers, as already described. One register addresses the odd numbered locations and one addresses the even numbered locations. There are two read current drivers and two write current drivers. One read and one write current driver supply the odd numbered drive lines in all four stores and one read and one write current driver supply the even numbered drive lines.

The drive line to be energised is selected by means of selection gates similar to those used in the Q store. As with the Q Stores, the JNS to be used by any program is selected by the output of a 2-bit program number which is set in the Program Number Register before Director hands over control.

There are 16 read amplifiers and 16 digit drivers. The digit drivers are timed by "bias control flip-flops" which are set during the write cycle and reset when writing has been completed (fig. 33). The digit drivers which must pass current are selected by the contents of the JB as described in a previous report.

6.6 Circuits

The circuits used in the JNS are similar to those used in the Q store. Various components have been changed to cope with the different voltages and currents involved.

The read current drivers supply a 1 usec current pulse of about 120 mA and the write current drive supply 60mA. The digit drivers are K253 circuits which are similar to the "DC gate" part of the QS digit drivers.

The read amplifiers are the same as the QS read amplifiers and the strobe is simply an amplified clock pulse.

J. B. Bunting

[Signature]