THE ENGLISH ELECTRIC COMPANY LIMITED

DEPT. DATA PROCESSING AND CONTROL SYSTEMS
KIDSGROVE

F.S.D. or Order No.

REPORT No. K/GD.u.331
A.C.E.D. Ref. No.
Deptl. Ref.
Date 28.9.62

Copies	TO:-
1	Mr. H.M. Mathews, Director of Engineering, London. (Summary sheet only)
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. C. Robinson, Computing Services Manager, DP&CSD, Kidsgrove.
1	Mr. A.C.D. Haley, Chief Development Engineer, Kidsgrove.
1	Mr. G.D. Royle, Group Engineer, F.E.S., Kidsgrove.
1	Mr. R.H. Allmark, Computer Development, Kidsgrove.
1	Chief Engineer's File.
1	M.E.L. Library, Whetstone.
60	Instruction Department, DP&CSD, Kidsgrove.

NOTES ON KDF 9 ARITHMETIC CONTROL

C.2.

Report by

J.A. Edwards.

SUMMARY

This report presents brief notes on KDF 9 Arithmetic Control. It is not a complete and fully detailed report. Shift Control is not discussed - this is the subject of report K/GD.y.80.

The material in this report has been obtained in discussions with engineers working on the KDF 9 development project - particularly Mr. R.M. Lloyd and Mr. M. Knight.

CONTENTS

N.B. Drawing numbers referred to in this report are of the type K78031 .... (e.g. K78031/9484). In references to such drawings the prefix "78031" has been omitted (e.g. K-/9484).

ARITHMETIC CONTROL

1.0 GENERAL

The division of responsibilities between Main Control and Arithmetic Control has been described in Report K/GD.y.77 and the operation of Main Control has been discussed in Report K/GD.u.293. The operation of Arithmetic Control is shown by a flow chart on drg. CE/SKM5725 and a suitable block diagram is provided on drg. CE/SKM5726. A more detailed block diagram of AC is provided on drg. K-M2987.

Arithmetic Control is required to inspect each instruction in turn, after it has been obeyed by Main Control, and to take all necessary actions relating to the Arithmetic Unit or the Multiplier-Divider. Instruction words are fetched from the Main Store, as already described, by Main Control. Main Control and Arithmetic Control then inspect these instructions quite independently, an interlock being provided to ensure that AC does not attempt to obey an instruction which has not been dealt with by MC. Main Control fetches another instruction word as soon as it has obeyed all the instructions in the word which it fetched last. Another interlock is provided to ensure that instructions are not replaced until AC has finished with them. Arithmetic Control is concerned, therefore, only with transferring instructions from the IWB to its Instruction Staticiser.

Arithmetic Control is required to take action on all one syllable instructions (which are arithmetic operation codes) and on the two and three syllable instructions listed in fig. 1. The other instructions are obeyed as "dummies" and cause no significant action to be taken. The machine codes of the two and three syllable instructions which require AC to take action are arranged so that all the information required by AC is contained in the first syllable (for example, all instructions in which the first two bits are '10' are recognised as "JUMP" instructions). MS and QS addresses required in "Store" operations are preserved in the SBAR and AQB, as described in Report K/GD.u. 293. Constants required by Shift Control are stored in the Shift Buffer (SHB) and the syllabic address which AC requires in order to obey a "JUMP" instruction is left in the "Jump Address Buffer (JAB)" by MC. It is therefore necessary for AC to inspect only the first syllable of each instruction; the Instruction Staticiser has a capacity of only eight bits.

Arithmetic Control is very similar to Main Control: it contains an instruction syllabic address counter, an Instruction Staticiser, decoding logic and a sequence unit. However, certain extra logic has been provided to prevent AC's having to waste time fetching instructions between operations; instruction fetching is performed in parallel with the execution of the previous operation.

The Next Instruction Buffer (ANIB), Initial MIC and Final MIC logic are provided for this purpose. The syllable selected by the instruction address counter is transferred to the NIB where it is stored until the operation currently proceeding has finished. The outputs of this register are used to define the Initial MI code of the sequence as shown in fig. 2. The last control pulse of the current sequence (a P1) transfers the instruction to the Instruction Staticiser and the Initial MIC to Arithmetic Control's MIS2. The next P2 generates the control pulses specified by the contents of the MIS2 and transfers the next instruction to the ANIB. The contents of the Instruction Staticiser are used to generate the Final MIC of the sequence as shown in fig. 3. When the MIC in the MIS1 is equivalent to the Final MIC, a "Final control pulse" is generated and the actions mentioned above take place.

1.1 Operation of AC

The syllable selected by the contents of the ANIA (AC Next Instruction Address) is transferred from the IWB to the NIB on every P2 pulse. When equivalence between MIS1 and FMIC is noted, an AP1F (AC Final P1 - alias P1F) pulse is generated. This pulse takes the following actions:

(i) ANIB⟶AINS (AC Instruction Staticiser)
(ii) IMIC⟶AMIS2
(iii) counts ADIA⟶ANIA (increments the Next Instruction Address)
(iv) sets AEF (AC End of Function flip-flop)
(v) Arithmetic Unit "housekeeping"

The next P2 pulse causes the first MOP of the new sequence to be executed. It also transfers to the NIB a new instruction and generates an AIP2 (AC Initial P2 - alias P2(F+1)) pulse. This pulse takes the following actions:

(i) ANIA⟶ADIA (AC Delayed Instruction Address)
(ii) FMIC⟶FMIC register (removes MIS1 FMIC level)
(iii) Arithmetic Unit "housekeeping".

The next P1, the second pulse of the sequence (this may also be the final pulse), resets the AEF flip-flop unless the contents of the MIS1 and FMIC registers are equivalent.

2.0 INSTRUCTION FETCHING LOGIC

The instruction fetching logic consists of the two Instruction Word Buffers, a Next Instruction Buffer (ANIB), an Instruction Staticiser (AINS) and a syllabic address counter. The IW Buffers are two 48-bit registers which are filled alternately from the Main Store as described in Report K/GD.u.293.

2.1 The ANIB

The ANIB is an 8-bit register to which the syllable specified by the ANIA register is transferred on every P2 clock pulse. It stores the next instruction until this is required by the Instruction Staticiser; it also generates the initial MI Code of the next sequence.

Since the first syllable of an instruction may occupy any of the 12 locations in the IW Buffers, transfer paths from each of these locations to the ANIB must be provided. Arithmetic Control cannot skip over instructions in which it has no interest, so transfers of extra "instruction length" digits are not required.

The transfer paths are shown on drg. K-/9480. The transfer transformers are on K251/10 packages - the same type of package as is used in the IWB⟶INS transfers. Ten packages are provided situated in locations 1F1-1F10 of MC2. As in Main Control, the transformer to be pulsed is selected by the decoded output of the syllabic address register. The ANIA register is shown on drg. K-/9481.

The outputs of the ANIB are applied, via transfer transformers, to the AINS. Two transfer boards are provided - one K293/10 board in 1E2 and one K251/10 board in 1E1 of MC2. It should be noted that the K251/10 package carries the '0' outputs of the seven l.s. digits of the ANIB and the '1' output of the most significant digit. This is because another transformer on the same package, using the same output pins, is used for setting an "artificial jump" instruction in the AINS.

The outputs of the ANIB are also applied to the "count" logic and to the IMIC logic. The count logic simply decodes the two most significant digits and generates a level which specifies a count of 1, 2 or 3 as appropriate (K-/9480-L3). The IMIC logic is described below.

2.2 The ANIA Counter

The ANIA counter consists of a set of D.C. gates, driven by the outputs of the ADIA register, which provide levels specifying the actions required to increment the address in the NIA register by 1, 2 or 3.

The contents of the ANIA register are transferred to the ADIA register on every P2 clock pulse. The AP1F pulse transfers the appropriate outputs of the counter to the ANIA register. The logic for each count is precisely the same as the corresponding logic in the NIA counter in Main Control. Truth tables and Veitch Diagrams are provided in Report K/GD.u.293.

"Jump" instructions require AC to take the next instruction from a location specified by the "Jump" instruction. When MC obeys "Jump" instructions, it fetches the appropriate instruction word and stores in the Jump Address Buffer (JAB) the next instruction's syllabic address. When AC obeys the "Jump", it transfers the contents of the JAB (if this is a valid address) to the ANIA register. The next P2

transfers the appropriate instruction to the ANIB and transfers ANIA to the ADIA register. The AP1F pulse then increments ANIA in the normal way.

The outputs of the ADIA register are applied to a "4-bit equivalence" circuit which compares the contents of the ADIA and the CIA register in Main Control. The CIA register contains either the address of the instruction currently being executed by MC or the address of the instruction just completed if MC is in the "Fetch Next Instruction" sequence. At AP1F time (the last P1 pulse of an AC sequence), the ADIA register contains the address of the instruction which it is about to obey. If at AP1F time the contents of the two registers are the same, AC is stopped since MC must have obeyed an instruction before AC attempts to obey it.

The most significant digit of the ANIA register is applied to "not equivalent" gate which provides a "high" level when the most significant digits of the CIA and ANIA registers are different and the AGO2 flip-flop is set. This level is inspected by Main Control on certain P2 clocks. At P2 time, ANIA contains the address of the next instruction to be obeyed by AC and the AGO2 flip-flop indicates that the AC is not waiting. A "high" level from the gate therefore shows that:

(a) AC has not caught up with MC

(b) AC is not taking its next instruction from the same IWB as MC.

2.3 The IMIC Logic

The AIMS2 register is controlled by two sets of complementing circuits. One set accepts the NMIC outputs of the P1 matrix, as in the MC sequence unit, the other set accepts the outputs of five IMIC sequence unit transformers. Four of these transformers (on a K342 board in 2B24) generate the IMI codes required by the one syllable instruction; the other (on a K302 board in 2B23) generates the IMI codes required by the 2 and 3 syllable instructions.

The arithmetic operation codes (the one syllable instructions) have been arranged in four groups, according to their bit patterns, as follows:

group (i): 00000---
(ii): 00001---
(iii): 0001----
(iv): 001-----

These bits are decoded (K-/9480-H3) and select one of the four IMIC SU transformers mentioned above (K-/9263-D2). The IMI codes required by the various instructions are generated by connecting the outputs of certain ANIB flip-flops to the NMIC windings of the SU transformers. Certain windings are not connected and will therefore always generate a '0' bit; other windings are connected to a bias level and will therefore generate a '1' bit.

Group (i): IMIC = 0 0 0 5 0 0

(ii): IMIC = 0 d4 d4 d5 d6 d7

(iii): IMIC = 0 0 1 d4 d5 d6

(iv): IMIC = 0 0 0 d3 d4 1

The 2 and 3 syllable instructions which require AC to take action are decoded and generate function levels or combination levels. These levels are applied to the NMIC windings of the fifth SUT which provides the appropriate Initial MI Codes by the normal "NMIC" methods.

2.4 The Instruction Staticiser

The AC Instruction Staticiser is an 8-bit flip-flop register, the outputs of which are decoded to provide "Function" levels for the AC sequence unit and the FMIC logic. It is shown on drg. K-/9264-K2.

Instructions are transferred to this register from the ANIB under the control of a AP1F pulse via transformers shown on drg. K-/9480-G12. In addition to instructions supplied in this way, an artificial "JUMP" instruction may be set in the AINS, at the request of Main Control when an interruption occurs. When MC executes an interruption sequence, it waits for AC to catch up (when the A.C. "current" instruction will be a AINS). If AC must not obey this instruction before entering Director it is changed to a "Jump".
The outputs of the AINS are decoded by means of five K443 packages, various odd diodes and 69 DC gates.
Not all instructions are fully decoded - some "combination" levels are generated where this is more convenient. The decoding is shown in figs. 1 & 2. Forty-four one syllable instructions are fully decoded and all eight bits of the first syllable of eight multi-syllable instructions are decoded. Fifteen instructions are not decoded and are obeyed as "dummies". The rest are partially decoded. The partially decoded instructions and certain of the fully decoded multi-syllable instructions generate "combination" levels.

Certain of these "function" and "combination" levels are used to generate the Final MI Code of the sequence. The FMIC register is shown on drg. K-/9264-C18. This register is controlled by complementing circuits as are the MIS1 and MIS2 registers. Inputs to the FMIC complementers are provided by the NMIC windings of five SU transformers under the control of the AC function levels.

The outputs of the FMIC register are applied to the "Not End of Function" gates shown on drg. K-/9263-C10. When equivalence exists between the contents of the MIS1 and FMIC registers, these gates allow the End of Function flip-flop to be set as shown on drg. K-/9267-E9.

3.0 The Sequence Unit

The AC sequence unit is the same, in principle, as the MC sequence unit. It consists of two 8 x 8 matrices of SU transformers, one transformer in each matrix being selected by the decoded outputs of two Micro-instruction Staticisers. The selected transformer in one matrix is supplied with P1 clocks; the selected transformer in the other is supplied with P2 clocks. The selection arrangements are exactly the same as the arrangements for the MC sequence unit - they are shown on drawing K-/9263.

3.1 Manual Stop and Single Shot Logic

There exists, in Main Control, logic which allows the MC sequence unit to be stopped and to be supplied with clock pulses under the manual control of an operator. This logic is described in Report K/GD.u.293. Similar logic is provided to allow manual control of the AC sequence unit; this logic is shown in fig. 7 and on drawing K-/9267.

The AC Single Shot Logic permits the supply of pulses to either, or both, of the SU matrices to be cut off. It also allows clock pulses to be supplied to the matrices one at a time. In this way, an AC sequence may be executed MOP by MOP or one MOP may be performed repeatedly at high speed.

Associated with each matrix are a "GO" flip-flop and an "INHIBIT" switch, either of which may inhibit the clock supply to the matrix in question. The action of inhibit switch, in both cases, may be overridden by operation of the appropriate single shot switch which allows a single clock pulse to be applied to the matrix. If the clock supply is cut off because the controlling flip-flop is reset, the single shot switch has no effect.

Arithmetic Control can be stopped by means of the STOP switch which allows a P2 pulse to set or reset a STOP KEY flip-flop. When this flip-flop is set, the last P1 in the current arithmetic operation resets the STOP flip-flop and so cuts off the supply of clock pulses to the P2 matrix. If the STOP KEY flip-flop is then reset and a P1 Single Shot given, Arithmetic Control is allowed to proceed. The P1 SINGLE SHOT switch allows a P2 to set a SSP1 KEY flip-flop (flip-flop K in fig. 7). This flip-flop allows the next P1 to set flip-flop A and so provides a condition for setting the P1 SINGLE SHOT flip-flop on the next P2. The output of the SSP1 flip-flop gates a clock pulse to the P1 matrix (which is being supplied anyway under these conditions) and allows the STOP flip-flop to be set, thus restoring the P2 clock supply. The P1 SINGLE SHOT flip-flop also allows the B flip-flop to be set. This flip-flop provides a reason for resetting the P1 SINGLE SHOT flip-flop and removes the condition for setting it even though flip-flop A is still set.

If the STOP KEY flip-flop is left set and a P1 Single Shot is given, Arithmetic Control is allowed to obey the next instruction. The STOP flip-flop is set, as before, and the supply of clocks to the P2 matrix is restored. The last P1 in the operation (PIF) however, resets the STOP flip-flop and so stops AC.

If the P2 INHIBIT switch is set, when the machine has been stopped, each operation of the P2 single shot key supplies one P2 clock and one P1 clock. Similarly, if the P1 INHIBIT switch is set, each operation of the P1 single shot key supplies one P1 clock and one P2 clock.

If both INHIBIT switches are set, each operation of the P1 SINGLE SHOT key provides one P1 clock pulse and each operation of the P2 SINGLE SHOT key supplies one pulse to the P2 matrix.

3.2 "Internal" Stop Logic

Arithmetic Control must stop at the end of the operation currently being executed if the next instruction which it is to obey is one which Main Control ma not have finished with. It is made to stop by resetting the "GO" flip-flops which control the supply of clocks to the P1 and P2 matrices.

At PIF time, the ADIA register contains the syllabic address of the instruction which it is about to obey and the CIA register in Main Control contains the address of the instruction currently being obeyed (or just finished if an "Instruction Fetch" sequence is being executed) by MC. If the contents of these two registers are the same, therefore, the PIF pulse is allowed to reset the "GO" flip-flop associated with the P2 matrix, (the AGO 2). The next P2 pulse then resets the AGO 1 flip-flop and so prevents the execution of further P1 MOPS.

As soon as MC moves on to the next instruction (the contents of ADIA are altered) the AGO 2 flip-flop is set by a P1 pulse and the AGO 1 flip-flop is set by the next P2. Arithmetic Control then proceeds normally.

When an interruption occurs, MC has to wait for AC to catch up. The first MOP of the interruption sequence therefore transfers the inversion of the bit in the AGO 2 flip-flop to the GO flip-flop in Main Control (see K/GD.u.293). If AGO 2 is set, the supply of clocks to the MC P1 sequence unit matrix is therefore cut off and the same P2 MOP is executed repeatedly. When the AGO 2 flip-flop is reset, the GO flip-flop is set and Main Control continues. During the interruption MC initiates a "Jump" to syllable 0 of instruction Word 0 and sets an artificial "Jump" instruction into the AINS if AC must not obey the current instruction.

4.0 AC - MC INTERLOCKS

Interlocks exist between AC and MC which ensure that:

(a) AC does not start operations which MC may not have completed. (This is arranged by the ADIA≡CIA condition above).

(b) MC does not replace instruction words which AC has not finished with. (This condition is ANIA(DO) ≠ CIA(Do) & AGO 2).

(c) MC stops when necessary to allow AC to use the Q Store or the JNS (AC sets NOG and executes a wait loop until MC resets GO).

(d) MC waits, before executing an interruption sequence, until AC has caught up (MC transfers AGO2 to GO)

(e) MC does not obey a "QS store" instruction if AC has not dealt with the last such instruction (MC does not start the sequence if AQBM is set - this is reset by AC)

(f) MC does not obey an "MS store" instruction if AC has not dealt with the last such instruction (MC does not proceed with the operation if SBARM is set - this is reset by AC)

(g) MC does not obey any JNS instruction if it has obeyed an "= LINK" instruction which AC has not yet dealt with. (MC inspects the JNLO flip-flop - this is reset by AC.)

(h) MC does not obey any "Fetch" instruction if both Fetch Buffers contain information which AC has not yet used (MC inspects the FBFM flip-flop which is reset when AC transfers data from either of the buffers - see K/GD.y.78 fig. 25)

(i) MC does not attempt to fetch data from a QS location if it has obeyed a "QS Store", specifying that location, which AC has not yet dealt with (AQB ≡ QSAR) & AQBM(1) - AC resets AQBM)

(j) MC does not attempt to fetch and use data from a MS location if it has obeyed a "MS store", specifying that location, which AC has not yet dealt with (SBAR ≡ MSAR) & SBARM(1) - if there is no data in the SB, Main Control fetches a word but does not transfer it from the MSB; if there is data in the SB, this data is stored and it is transferred to the Fetch Buffer.

(k) MC does not attempt to execute a "Jump" if it has obeyed a Jump instruction which AC has not yet dealt with (MC inspects JABM which is reset during the AC "Jump" sequence).

(l) MC does not attempt to obey a "Shift" instruction if it has obeyed such an instruction which AC has not yet dealt with (MC sets SHBM which is reset during the AC shift sequence).

J. Boothroyd
CHIEF INSTRUCTOR

J.A. Edwards

ONE SYLL INSTRUCTIONS DECODED BY ARITHMETIC CONTROL

SHEET NO.:-

OF SHEETS

FIG 1

FUNCTION	BITS DECODED			IMIC (OCTAL)	FMIC (OCTAL)	FUNCTION	BITS DECODED			IMIC (OCTAL)	FMIC (OCTAL)
DUMMY	—	—	—	00	00	DUPD	00	100	011	01	24
VR	00	000	001	00	32	÷I	00	100	100	01	42
=TR	00	000	010	00	02	FIX	00	100	101	01	03
BITS	00	000	011	00	34	STR	00	100	111	01	24
xF	00	000	100	04	53	CONT	00	101	000	03	02
xDF	00	000	101	04	20	REVD	00	101	001	03	24
x+F	00	000	111	04	20	ERASE	00	101	010	03	02
NEGD	00	001	000	30	05	-D	00	101	011	03	
OR	00	001	001	31	50	AND	00	101	100	03	22
PERM	00	001	010	32	03	+D/-D	00	101	-11		05
TOB	00	001	011	33	02	+D				03	
ROUND.H	00	001	100	34	45	÷	00	110	000	05	
NEV	00	001	101	35	50	÷D/÷	00	110	00-		40
ROUND/H-	00	--1	110		47	÷D				05	
ROUND				36		÷F	00	110	010	05	
+				03		÷FD/÷F	00	110	01-		53
-				17		÷FD				05	
ROUND.F	00	010	000	10		÷R/÷I	00	1-0	100		40
ROUND HF				10		÷R				05	
ROUND HF/ ROUND.F	00	010	00-		53	REV	00	110	101	05	03
-DF/+DF	00	010	01-		20	CAB	00	110	110	05	12
-DF				11		FRB	00	110	111	05	02
+DF				11		STAND	00	111	000	07	53
FLOAT	00	010	100	12	53	NEG DF	00	111	001	07	20
FLOAT D	00	010	101	12	20	MAX F	00	111	010	07	31
ABS	00	010	110	13	60	+F/-F	00	111	10-		53
NEG	00	010	111	13	45	+F				07	
ABS F	00	011	000	14	26	-F	00	111	101	07	
NEG F	00	011	001	14	53	SIGN F	00	111	111	07	
MAX	00	011	010	15	31						
NOT	00	011	011	15	03						
XD/X	00	011	10-		44						
XD				16							
X	00	011	101	16	47						
SIGN/SIGN F	00	-11	111		30						
SIGN				17							
ZERO	00	100	001	01	24						
DUP	00	100	010	01	24						

A ORIGINAL ISSUE

TWO & THREE SYLL INSTRUCTIONS DECODED BY AC

Fig 2

FUNCTION	BITS DECODED			IMIC (OCTAL)	FMIC (OCTAL)	FUNCTION	BITS DECODED			IMIC (OCTAL)	FMIC (OCTAL)
IND. FETCH	01	00-	--0	01	12	JUMPS	10	---	---	00	00
IND STORE	01	00-	--1	46	02	DIR FETCH/ SET	11	---	--0	01	12
SHAD/SHL/SHID	01	110	--0	42	00	DIR. STORE	11	---	--1	46	02
SHA	01	110	001	42	42						
X+n	01	110	011	16	05						
SHC	01	110	111	54	64						
WI → Q	01	111	000	75	60						
Q → WI/LINK	01	111	0-1	01	12						
= LINK	01	111	011	75	10						
WI + Q	01	111	010	01	60						
Kn	01	111	110	01	12						
SLJ	01	111	111	00	00						

INSTRUCTIONS NOT DECODED BY AC (DUMMIES)

Q MOD Q (S12-S19)
Q TO Q (S20)
= Kn (S34)
PERIPHERAL (S40-S44)

GENERATION OF INITIAL MI CODES — ARITHMETIC OPERATIONS. K342/240 IN 2824.

Fig 3

ARITHMETIC OPERATION	OP. CODE 01	OP. CODE 2 3 4 5 6 7	INITIAL MIC (OCTAL)	TRANSFORMER SELECTION BITS 01	TRANSFORMER SELECTION BITS 2 3 4 5 6 7	BITS USED TO GEN. IMIC 0 1 2 3 4 5
DUMMY	00	0 0 0 0 0 0	00			
VR	00	0 0 0 0 0 1	00			
=TR	00	0 0 0 0 1 0	00			
BITS	00	0 0 0 0 1 1	00	00	0 0 0 ——	0 0 0 d5 0 0
XF	00	0 0 0 1 0 0	04			
XDF	00	0 0 0 1 0 1	04			
X+F	00	0 0 0 1 1 1	04			
NEGD	00	0 0 1 0 0 0	30			
OR	00	0 0 1 0 0 1	31			
PERM	00	0 0 1 0 1 0	32			
TOB	00	0 0 1 0 1 1	33	00	0 0 1 ——	0 d4 d4 d5 d6 d7
ROUND H	00	0 0 1 1 0 0	34			
NEV	00	0 0 1 1 0 1	35			
ROUND	00	0 0 1 1 1 0	36			
ROUND F	00	0 1 0 0 0 0	10			
ROUND HF	00	0 1 0 0 0 1	10			
-DF	00	0 1 0 0 1 0	11			
+DF	00	0 1 0 0 1 1	11			
FLOAT	00	0 1 0 1 0 0	12			
FLOAT D	00	0 1 0 1 0 1	12			
ABS	00	0 1 0 1 1 0	13			
NEG	00	0 1 0 1 1 1	13			
ABSF	00	0 1 1 0 0 0	14	00	0 1 ——	0 0 1 d4 d5 d6
NEG-F	00	0 1 1 0 0 1	14			
MAX	00	0 1 1 0 1 0	15			
NOT	00	0 1 1 0 1 1	15			
XD	00	0 1 1 1 0 0	16			
X	00	0 1 1 1 0 1	16			
—	00	0 1 1 1 1 0	17			
SIGN	00	0 1 1 1 1 1	17			

CONTINUED

GENERATION OF INITIAL MI CODES — ARITHMETIC OPERATIONS K342/240 IN 2B24.

CONTINUATION

Fig 3

ARITHMETIC OPERATION	OP. CODE 01	OP. CODE 2 3 4 5 6 7	INITIAL MIC (OCTAL)	TRANSFORMER SELECTION BITS 01	TRANSFORMER SELECTION BITS 2 3 4 5 6 7	BITS USED TO GEN. IMIC 0 1 2 3 4 5
ZERO	00	1 0 0 0 0 1	01			
DUP	00	1 0 0 0 1 0	01			
DUPD	00	1 0 0 0 1 1	01			
÷I	00	1 0 0 1 0 0	01			
FIX	00	1 0 0 1 0 1	01			
STR	00	1 0 0 1 1 1	01			
CONT	00	1 0 1 0 0 0	03			
REVD	00	1 0 1 0 0 1	03			
ERASE	00	1 0 1 0 1 0	03			
-D	00	1 0 1 0 1 1	03			
AND	00	1 0 1 1 0 0	03			
+	00	1 0 1 1 1 0	03	00	1 ———	0 0 0 d3 d4 1
+D	00	1 0 1 1 1 1	03			
÷	00	1 1 0 0 0 0	05			
÷D	00	1 1 0 0 0 1	05			
÷F	00	1 1 0 0 1 0	05			
÷FD	00	1 1 0 0 1 1	05			
÷R	00	1 1 0 1 0 0	05			
REV	00	1 1 0 1 0 1	05			
CAB	00	1 1 0 1 1 0	05			
FRB	00	1 1 0 1 1 1	05			
STAND	00	1 1 1 0 0 0	07			
NEGD F	00	1 1 1 0 0 1	07			
MAX F	00	1 1 1 0 1 0	07			
+F	00	1 1 1 1 0 0	07			
-F	00	1 1 1 1 0 1	07			
SIGN F	00	1 1 1 1 1 1	07			

KIDSGROVE STOKE-ON-TRENT

GENERATION OF INITIAL MI CODES — 2 & 3 SYLL INSTRUCTIONS. K302/230. IN 2B23.

SHEET NO.:-

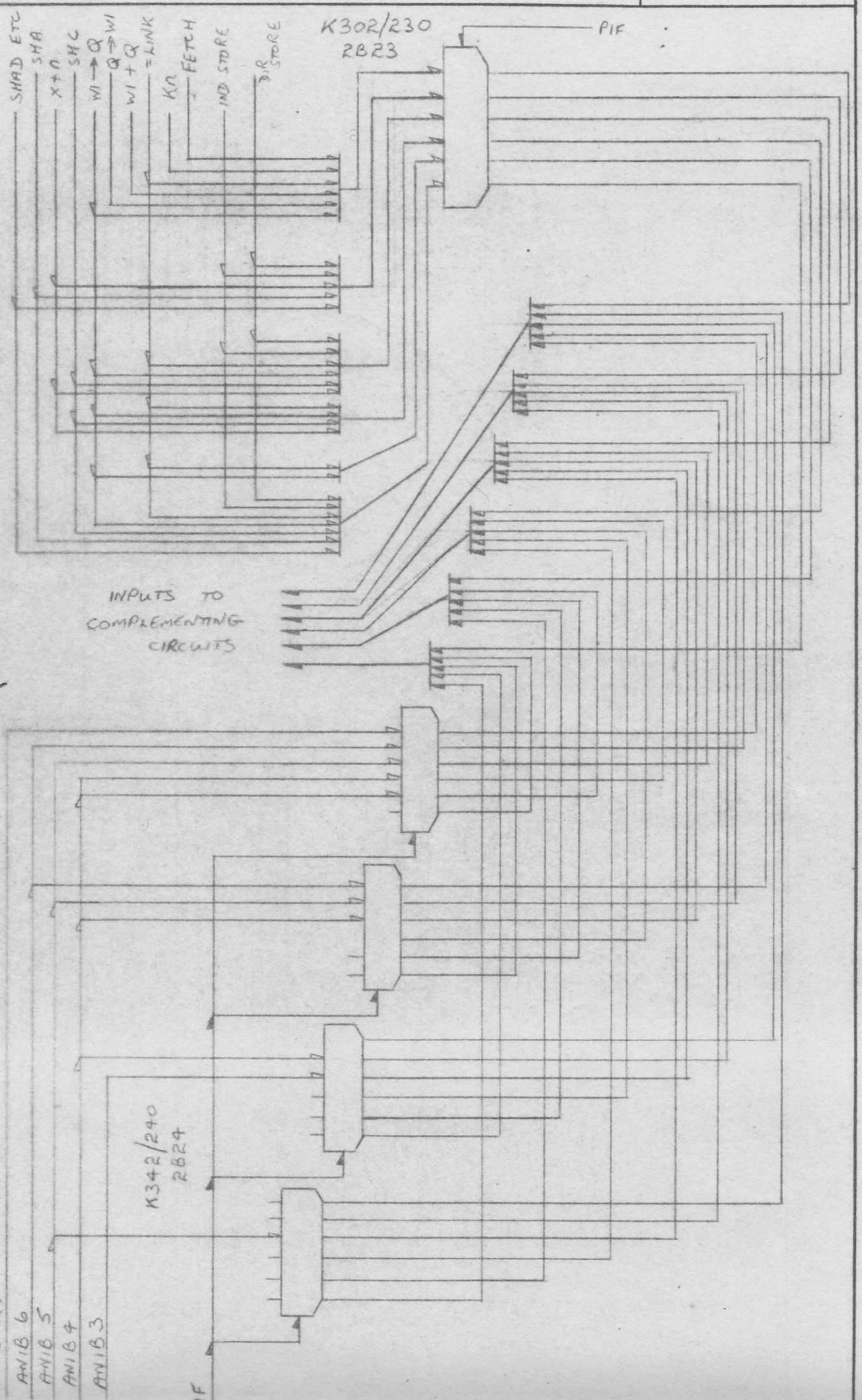
OF SHEETS

FIG 4

	OPERATION CODE		INITIAL MIC (OCTAL)
	0 1	2 3 4 5 6 7	
SHAD/SHL/SHLD	01	1 1 0 X X 0	42
SHA	01	1 1 0 0 0 1	42
X + n	01	1 1 0 0 1 1	16
SHC	01	1 1 0 1 1 1	54
WI → Q	01	1 1 1 0 0 0	75
Q → WI/LINK	01	1 1 1 0 X 1	01
WI + Q	01	1 1 1 0 1 0	01
= LINK	01	1 1 1 1 0 0	75
Kn	01	1 1 1 1 1 0	01
IND. FETCH	01	0 0 X X X 0	01
DIRECT FETCH	11	X X X 0 X 0	
IND. STORE	01	0 0 X X X 1	46
DIRECT STORE	11	X X X 0 X 1	46

GENERATION OF INITIAL MICODES. Fig 5

OF SHEETS



SUT	OPERATION	9 BITS	OP. CODE (OCTAL)	FINAL MIC (OCTAL)	SUT	OPERATION	9 BITS	OP. CODE (OCTAL)	FINAL MIC (OCTAL)
2D19 (K302/240)	VR	0	01	32	2D21 (K302/260)	ZERO	0	41	24
	=TR	0	02	02		DUP	0	42	24
	BITS	0	03	34		DUP D	0	43	24
	XF	0	04	53		÷I	0	44	42
	XDF	0	05	20		FIX	0	45	03
	X+F	0	07	20		STR	0	47	24
	NEGD	0	10	05		CONT	0	50	02
	OR	0	11	50		REVD	0	51	24
	PERM	0	12	03		ERASE	0	52	02
	TOB	0	13	02		+D/-D	0	53/57	05
	ROUND H	0	14	45		AND	0	54	22
	NEV	0	15	50		+/-/ROUND	0	56	47
2D20 (K302/250)	ROUND HF	0	21	53	2D22 (K302/270)	÷D/÷	0	60/61	40
	+DF/-DF	0	22/23	20		÷FD/÷F	0	62/63	53
	FLOAT	0	24	53		÷R	0	64	40
	FLOAT D	0	25	20		REV	0	65	03
	ABS	0	26	61		CAB	0	66	12
	NEG	0	27	45		FROB	0	67	02
	ABSF	0	30	26		STAND	0	70	53
	NEGF	0	31	53		NEG DF	0	71	20
	MAX	0	32	31		MAX F	0	72	31
	NOT	0	33	03		+F/-F	0	74/75	53
	xD	0	34	44		SIGN/SIGN F	0	37/77	30
	X	0	35	47		AGO2 (0)	0	—	77

	OPERATION	CODE	FMIC		OPERATION	CODE.	FMIC.
2D23 (K302/280)	DIR. STORE	D0. D1. D7	02	2D23 (CONT)	W1+Q	$(01111\ 010)_2$	60
	DIR. FETCH	D0. D1. $\overline{D7}$.	12		Q→W1/LINK	$(01\ 111\ 0X1)_2$	12
	INDIR. STORE	$\overline{D0}$. D1. $\overline{D2}$. $\overline{D3}$. D7	02		W1 → Q	$(01\ 111\ 000)_2$	60
	INDIR. FETCH	$\overline{D0}$. $\overline{D1}$. $\overline{D2}$. $\overline{D3}$. $\overline{D7}$	12		SHC	$(01\ 110\ 111)_2$	64
	Kn	$(01\ 111\ 110)_2$	12		X+n	$(01\ 110\ 011)_2$	05
	LINK	$(01\ 111\ 100)_2$	75		SHC	$(01\ 110\ 111)_2$	42

A ORIGINAL ISSUE

ARITHMETIC CONTROL — STOP LOGIC.

SHEET NO.:-

OF SHEETS

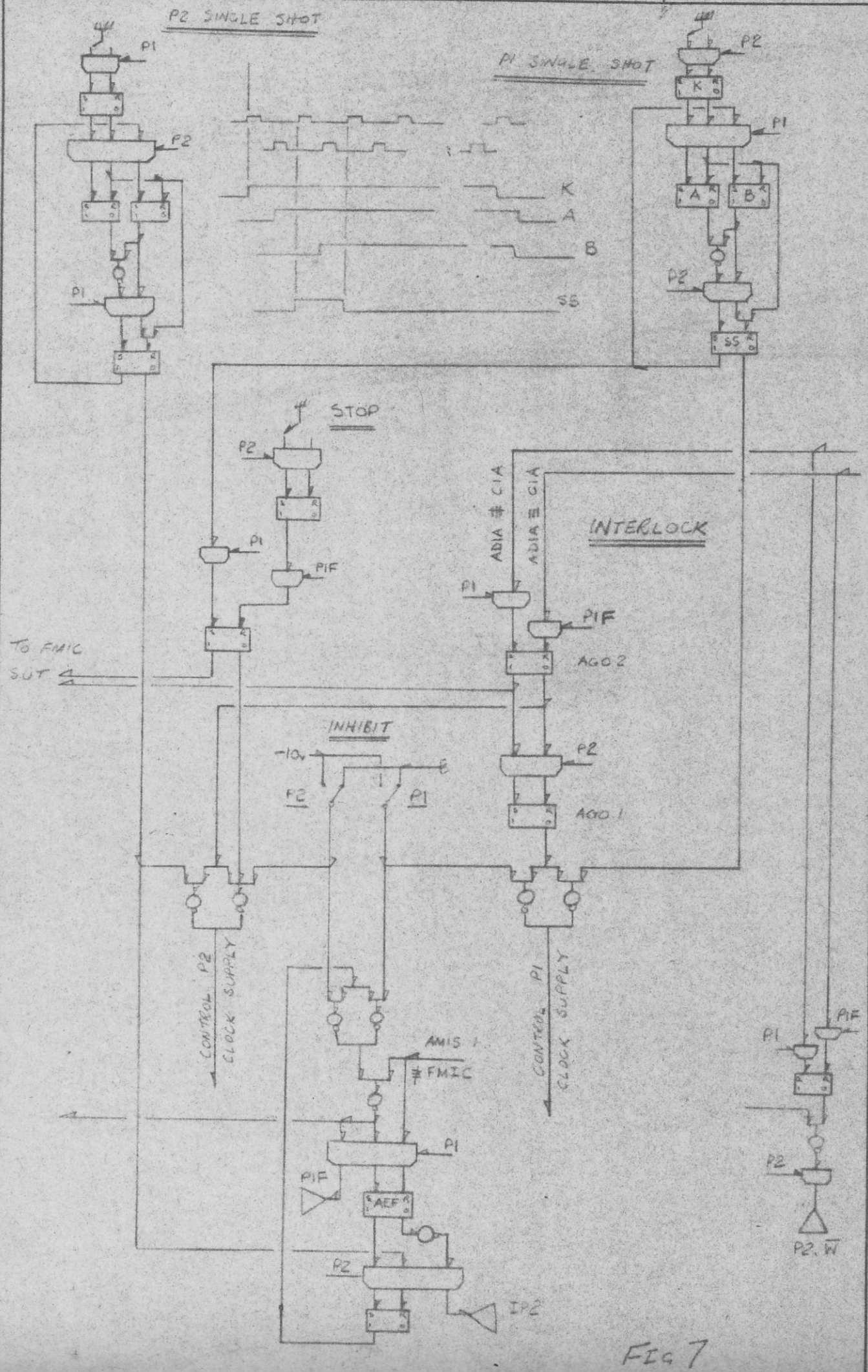


FIG 7

ORIGINAL ISSUE