1301 Programmers Manual

Section = Part Two

Original PDF by Roger Holmes
Jan 2009

Sub Sections and HTML version
by Rod Brown

Supplied by ict1301.co.uk
from the pages of the

ICT 1301
Resurrection Project

Part 2

I·C·T 1300 SERIES

programmers reference manual

THE CENTRAL PROCESSOR

Contents

Contents continued Page

Illustrations

Part 2

The Central Processor

GENERAL 2.1

The central processor consists of the units that are responsible for the overall organization of the routing and processing of data within the system, and comprises the arithmetic unit, the control registers and associated circuitry.

The Arithmetic Unit 2.1.1

The arithmetic unit adds, subtracts and multiplies; division is not a built-in routine but is accomplished by subroutines. The arithmetic unit also performs shift functions, transfer functions, row-binary functions and logical functions (AND, OR). The unit consists of:

three twelve-digit registers - A, B and C,
the Mill and associated indicators,
a one-digit Sterling Position Register,
a one-digit Decimal Point Register and
a one-digit Row Binary Register.

Arithmetic operations are performed on numbers held in Registers A, B and C. Program instructions are available for transferring numbers between the registers and I.A.S. and for carrying out the arithmetic.

Register A is the link between I.A.S. and the arithmetic unit. On completion of a transfer to or from I.A.S., Register A contains the same number as the specified word of I.A.S. Register A is also used when instructions are transferred from I.A.S. to the control registers to be obeyed. It should therefore be noted that when control is transferred from one word of program to the next, the contents of Register A are destroyed.

Arithmetic operations are usually carried out between Register B and a word of I.A.S. For example, during addition , the sequence of operations is as follows:

The contents of the I.A.S. location (word) specified in the add instruction are transferred to the arithmetic unit via Register A and are added in the Mill to the contents of Register B. According to the add instruction given, the result is placed either in Register B or in the I.A.S.

location from which the word originated. In the latter instance, the result is also left in Register A and the contents of Register B are unaffected.

The only arithmetic operation in which Register C is used is multiplication and the process is explained under "Multiplication".

For the instructions described in this part of the manual, the contents of Registers A, B and C may be assumed to be unaltered unless otherwise specified. In particular, the contents of a register are unaltered when they are transferred to another register or to a word of I.A.S. A complete summary of instructions is given in Part 6 together with a chart showing which registers are involved when an instruction is obeyed.

The Mill accomplishes addition digit by digit, commencing with the least-significant digit. Decimal or sterling carries are delayed and are added in with the next (more significant) digit. Although a word may contain up to 15 in one digit position, it is not normally advisable to give instruct - ions for such numbers (10 to 15) to pass through the Mill. Associated with the Mill are indicators that can be tested to ascertain whether the last number to pass through the Mill was positive, negative or zero.

The Sterling Position Register, Decimal Point Register and Row Binary Register are discussed in detail later in this part of the manual.

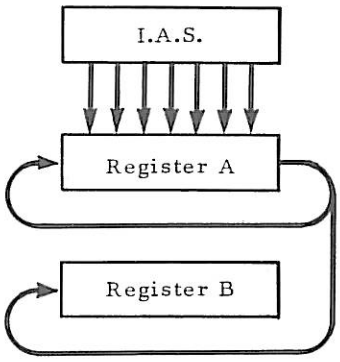
TRANSFER INSTRUCTIONS 2.2

Transfer instructions move data between I.A.S. locations and registers or from one register to another without the data being processed arithmetically. As data transferred by-pass the Mill, the Mill Indicators are not affected.

Function 37 2.2.1

Effect Transfers the contents of the specified location of I.A.S. to Register B.

Operation The content of I.A.S. location overwrites the previous content of Register B. The data reach Register B by way of Register A and thus Register A also contains the contents now in Register B. The contents of I.A.S. remain unaltered.



Example Transfer word 42 of block 11 to Register B.

Instruction

D	F	A	R
--	37	0042	11

Before

I.A.S.	0	0	0	0	0	0	0	5	7	9	4	1
Register A	0	0	0	0	0	0	1	4	7	6	8	2
Register B	0	0	0	0	0	0	0	0	0	5	6	3

After

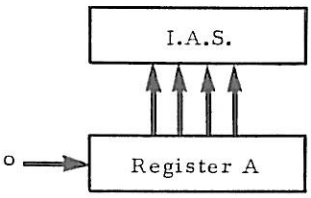
I.A.S.	0	0	0	0	0	0	0	5	7	9	4	1
Register A	0	0	0	0	0	0	0	5	7	9	4	1
Register B	0	0	0	0	0	0	0	5	7	9	4	1

Notes As data being transferred by-pass the Mill, digits from 0 to 15 in any position of one I.A.S. word can be transferred without mutilation.

Function 40 2.2.2

Effect Transfers zeros to all positions of a specified word of I.A.S.

Operation The contents of the specified location in I.A.S. are overwritten with zeros. As zeros are formed in Register A and then transferred to I.A.S., the contents of Register A will also be zero.



Example Word 42 of block 11 is to be zeroized.

Instruction

D	F	A	R
--	40	0042	11

Before

I.A.S.	0	0	0	0	0	0	5	8	9	4	3	0
Register A	0	0	0	0	0	0	6	5	8	7	9	2

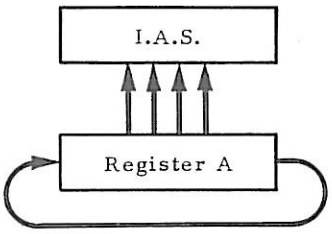
After

I.A.S.	0	0	0	0	0	0	0	0	0	0	0	0
Register A	0	0	0	0	0	0	0	0	0	0	0	0

Notes Zeroizing of words of I.A.S. will be necessary only in certain instances, such as where I.A.S. has been holding group accumulated totals and needs to be zeroized after each group.

Effect Transfers the contents of Register A to a specified word of I.A.S.

Operation The contents of Register A overwrite the previous contents in the specified location of I.A.S. The contents of Register A remain unaltered.



Example If it is necessary to transfer the contents of an I.A.S. location to some other location of I.A.S. while adding it to Register B, then a 41 order may be used as shown.

Program

I	D	F	A	R
X	--	37	0049	17
X+1	--	62 41	0025 0019	17 23
X+2	--	---	------	---

Before

Word 49 Block 17	0	0	0	0	0	0	0	6	9	4	3	2
Register A	0	0	0	0	0	0	0	7	1	5	6	8
Register B	0	0	0	0	0	0	0	8	4	3	8	0

After second instruction in word X

Word 49 Block 17	0	0	0	0	0	0	0	6	9	4	3	2
Register A	0	0	0	0	0	0	0	6	9	4	3	2
Register B	0	0	0	0	0	0	0	6	9	4	3	2

After instructions in word X+1

Word 25 Block 17	0	0	0	0	0	0	0	0	0	5	9	4
Word 19 Block 23	0	0	0	0	0	0	0	0	0	5	9	4
Register A	0	0	0	0	0	0	0	0	0	5	9	4
Register B	0	0	0	0	0	0	0	7	0	0	2	6

In the section of program above, word 49 of block 17 is transferred to Register B. Word 25 of block 17 is added to the contents of Register B and is also stored in word 19 of block 23. The 41 instruction effects the transfer of the contents of Register A to word 19 of block 23 thereby saving a special transfer instruction to Register B.

It must be noted that this will work only if the 41 instruction is the least-significant part of the instruction word. The contents of Register A would have been destroyed if control had been passed to the next word.

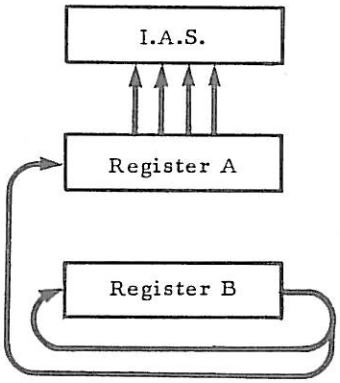
Notes The most common use of function 41 is to store the return jump instruction from a sub-routine. This is discussed fully in Part 3.

On completion of an instruction the contents of Register A are often useful and the use of function 41 should be kept in mind during programming.

Function 42 2.2.4

Effect Transfers the contents of Register B to a specified word of I.A.S.

Operation The contents of Register B are placed in Register A and thence overwrite the previous contents of the specified location in I.A.S. The contents of Register B are preserved.



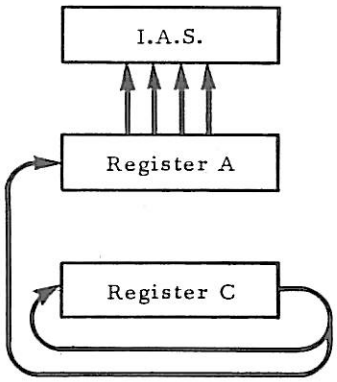
Example It may be necessary to transfer Register B to I.A.S. temporarily while performing calculations on other factors, calling upon the original contents of Register B later. To perform the calculation (a + b) x (c + d) = e, firstly (a + b) will be calculated and the result temporarily stored using the 42 instruction. Secondly (c + d) is calculated, and finally by bringing (a + b) out of I.A.S. the multiplication can be performed.

I	D	F	A	R	NARRATIVE
0		37	0012	13	a
		62	0014	13	(a+b)
1		42	0017	12	Temp Store (a+b)
		37	0013	13	c
2		62	0015	13	(c+d)
		69	0017	12	(a+b) × (c+d)
3		42	0016	13	e

Function 43 2.2.5

Effect Transfers the contents of Register C to a specified word of I.A.S.

Operation The contents of Register C are placed in Register A and thence overwrite the previous contents of the specified location in I.A.S. The contents of Register C are preserved.



Example Transfer the contents of Register C to word 42 of block 11.

Instruction

D	F	A	R
	43	0042	11

Before

I.A.S.	0	0	0	0	0	0	0	7	9	8	1	2
Register A	0	0	0	0	0	0	0	0	8	2	5	1
Register C	0	0	0	0	0	0	0	5	3	7	8	4

After

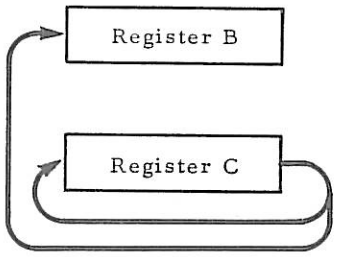
I.A.S.	0	0	0	0	0	0	0	5	3	7	8	4
Register A	0	0	0	0	0	0	0	5	3	7	8	4
Register C	0	0	0	0	0	0	0	5	3	7	8	4

Notes The only occasion, other than on input, that Register C is used is when performing multiplication. The product is left in both Registers B and C at the end of a multiply function; it is possible to use function 43 as an alternative to function 42 when storing the product in I.A.S.

Function 44 2.2.6

Effect Transfers the contents of Register C to Register B.

Operation The contents of Register C are placed in, and overwrite, the contents of Register B. The contents of Register C are preserved.



Example

Instruction

D	F	A	R
	44	0000	00

Before

Register B	0	0	0	0	0	0	0	5	1	4	6	2
Register C	0	0	0	0	0	0	0	0	2	5	9	3

After

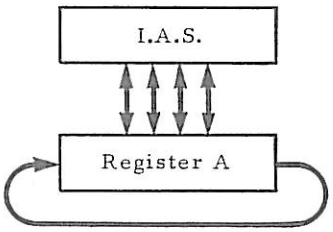
Register B	0	0	0	0	0	0	0	0	2	5	9	3
Register C	0	0	0	0	0	0	0	0	2	5	9	3

Note Register A is not used during this function.

Function 45 2.2.7

Effect Transfers the contents of one block of consecutively numbered locations in I.A.S. to another such block in I.A.S.

Operation A specified block of up to 20 words is transferred via Register A to overwrite another specified block of consecutive I.A.S. locations. The last word in the block transferred will be preserved in Register A but will always be overwritten during the change of control to the next word. Both I.A.S. blocks contain the same data when the function is completed (see Notes).



Instructions using function 45 are double-length and take the form:-

I	D	F	A	R
X	0	45	abcd	R_1
	0	xy	pqrs	R_2

where abcd/R_1 is the lowest numbered source address, pqrs/R_2 is the lowest numbered destination address and xy is the number of words in the block to be transferred, which must lie in the range 1 to 20.

Example

The instruction:

I	D	F	A	R
X	0	45	0010	12
	0	15	0100	16

transfers 15 words from addresses 0010, 0011, 0012..... 0024 of block 12 to addresses 0100, 0101, 0102 ..... 0114 of block 16.

Notes If the two relativizer reference numbers R_1 and R_2 are the same, they should be written in both places.

The previous contents of Register A are destroyed by this instruction.

If the two areas of storage between which the transfer is made do not overlap, then at the conclusion of the instruction both areas contain the information originally held in the source area. If the areas overlap, incorrect transfers can be obtained and to ensure correct transfers, they must be made from higher-numbered locations to lower-numbered ones, bearing in mind that absolute, not relative, addresses must be considered.

Example

INCORRECT TRANSFER

Order in which transfer is made	Source absolute address	Contents of the source address	Destination absolute address	Contents of destination address after transfer
1	0	115	1	115
2	1	125	2	115
3	2	135	3	115
4	3	145	4	115

This is obviously not a correct block transfer, although this property could be useful in zeroizing an area of I.A.S.

CORRECT TRANSFER

Order in which transfer is made	Source absolute address	Contents of the source address	Destination absolute address	Contents of destination address after transfer
1	101	321	100	321
2	102	16	101	16
3	103	143	102	143
4	104	92	103	92

This example demonstrates that the correct block transfer is achieved.

DECIMAL ADDITION AND SUBTRACTION 2.3

The decimal addition and subtraction instructions and the equivalent instructions described later under sterling addition and subtraction (Section 2.4) cause data to be processed in the Mill and therefore affect the Mill Indicators. Figure 4 is a summarized form of the schematic diagrams shown for each instruction under heading, Operation.

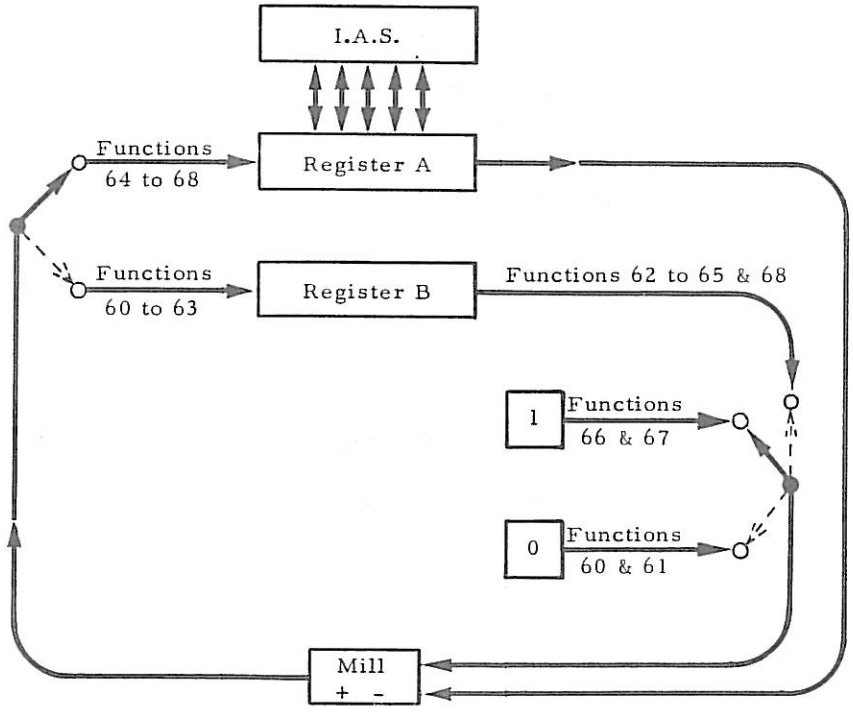
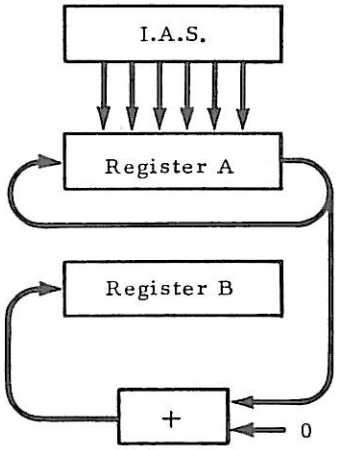


Figure 4: THE MILL INDICATORS AND FUNCTIONS 60 TO 68

Effect This function CLEAR ADDS the contents of a specified location of I.A.S. into Register B.

Operation The specified I.A.S. word is put in Register A, added to zero in the Mill and the result placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The contents of the I.A.S. location are unaltered.



Example To convert pence held in one position of I.A.S. to pence in two positions, a digit can be mutilated to some purpose by the 60 function. Assume that position 12 of word 19 of block 25 holds 11d. After the instruction

D	F	A	R
	37	0019	25

Register B contains

0	0	0	0	0	0	0	0	0	0	11

If however the instruction

D	F	A	R
	60	0019	25

is used, then

	Before											
I.A.S.	0	0	0	0	0	0	0	0	0	0	0	11
Register A	0	0	0	0	0	0	0	0	2	4	5	9
Register B	0	0	0	0	0	0	0	5	6	4	9	6

	After											
I.A.S.	0	0	0	0	0	0	0	0	0	0	0	11
Register A	0	0	0	0	0	0	0	0	0	0	0	11
Register B	0	0	0	0	0	0	0	0	0	0	1	1

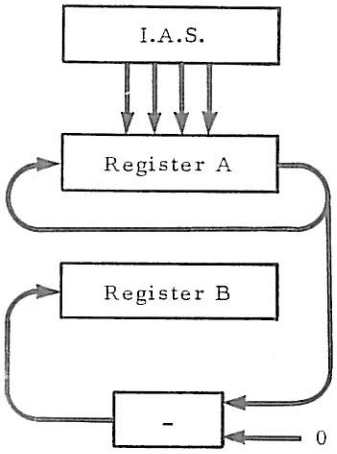
Notes As shown the function is similar to function 37, the essential difference being that function 60 uses the Mill and therefore affects the Mill and Overflow Indicators. Because it uses the Mill the instruction mutilates any digit greater than 9 in any one digit position by subtracting 10 from the digit and carrying 1 to the next most-significant digit.

It is normally good practice to employ a 37 function when a transfer to Register B is required unless the special properties of a 60 function are needed.

Function 61 2.3.2

Effect This function CLEAR SUBTRACTS the contents of a specified location of I.A.S. into Register B.

Operation The specified I.A.S. word is put in Register A, subtracted from zero in the Mill and the result placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The contents of the I.A.S. location are unaltered.



Example Clear subtract word 19 block 25.

Instruction

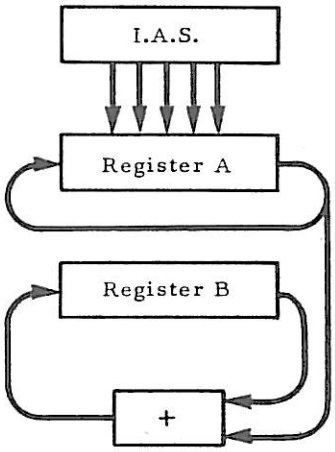
D	F	A	R
--	61	0019	25

	Before												After											
I.A.S.	0	0	0	0	0	0	0	0	0	8	2	4	I.A.S.	0	0	0	0	0	0	0	0	0	8	2	4
Register A	0	0	0	0	0	0	0	6	4	8	9	2	Register A	0	0	0	0	0	0	0	0	0	8	2	4
Register B	0	0	0	0	0	0	0	4	7	8	3	1	Register B	9	9	9	9	9	9	9	9	9	1	7	6

The result in Register B is the tens complement of 824.

Effect ADDS a specified word of I.A.S. to the contents of Register B and places the result in Register B.

Operation The specified I.A.S. word is put in Register A, added in the Mill to the contents of Register B and the result placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The I.A.S. word is unaltered.



Example Add 824 recorded as word 19 block 25 to 7453 contained in Register B.

Instruction

D	F	A	R
	62	0019	25

Before

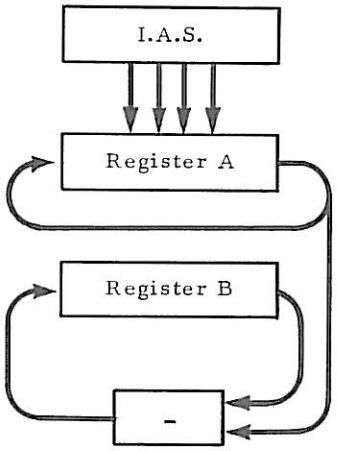
I.A.S.	0	0	0	0	0	0	0	0	0	8	2	4
Register A	0	0	0	0	0	1	2	4	3	6	9	2
Register B	0	0	0	0	0	0	0	0	7	4	5	3

After

I.A.S.	0	0	0	0	0	0	0	0	0	8	2	4
Register A	0	0	0	0	0	0	0	0	0	8	2	4
Register B	0	0	0	0	0	0	0	0	8	2	7	7

Effect SUBTRACTS a specified word of I.A.S. from the contents of Register B and places the result in Register B.

Operation The specified I.A.S. word is put in Register A, subtracted in the Mill from the contents of Register B and the difference placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The I.A.S. word is unaltered.



Example Subtract 421791 recorded as word 17 block 19 from 1924298 contained in Register B.

Instruction

D	F	A	R
	63	0017	19

Before

I.A.S.	0	0	0	0	0	0	4	2	1	7	9	1
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	1	9	2	4	2	9	8

After

I.A.S.	0	0	0	0	0	0	4	2	1	7	9	1
Register A	0	0	0	0	0	0	4	2	1	7	9	1
Register B	0	0	0	0	0	1	5	0	2	5	0	7

Alternatively if 1924298 is recorded as word 17 block 19 and is subtracted from 421791 contained in Register B the result is the complement or minus 1502507.

Before

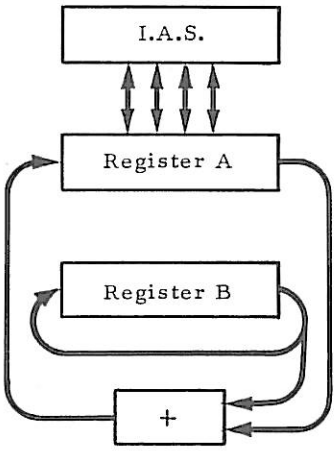
I.A.S.	0	0	0	0	0	1	9	2	4	2	9	8
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	4	2	1	7	9	1

After

I.A.S.	0	0	0	0	0	1	9	2	4	2	9	8
Register A	0	0	0	0	0	1	9	2	4	2	9	8
Register B	9	9	9	9	9	8	4	9	7	4	9	3

Effect This function ADDS the contents of Register B to the contents of a specified location of I.A.S. and places the result in that location.

Operation The specified I.A.S. word is put in Register A, added in the Mill to the contents of Register B and the sum placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the result, which may be useful in certain cases. The original contents of Register B remain unaltered.



Example Add 6543 in Register B to 58216 recorded as word 17 block 19.

Instruction

D	F	A	R
--	64	0017	19
--	---	-------	---

Before

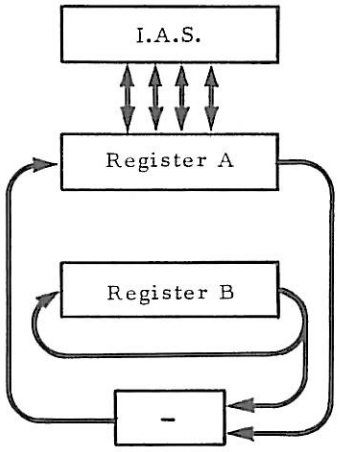
I.A.S.	0	0	0	0	0	0	0	5	8	2	1	6
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	0	0	6	5	4	3

After

I.A.S.	0	0	0	0	0	0	0	6	4	7	5	9
Register A	0	0	0	0	0	0	0	6	4	7	5	9
Register B	0	0	0	0	0	0	0	0	6	5	4	3

Effect This function SUBTRACTS the contents of Register B from the contents of a specified location of I.A.S. and places the result in that location.

Operation The specified I.A.S. word is put in Register A, subtracted in the Mill from the contents of Register B and the difference placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the difference. The original contents of Register B remain unaltered.



Example Subtract 6543 in Register B from 58216 recorded as word 17 block 19.

Instruction

D	F	A	R
--	65	0017	19
--	---	------	---

Before

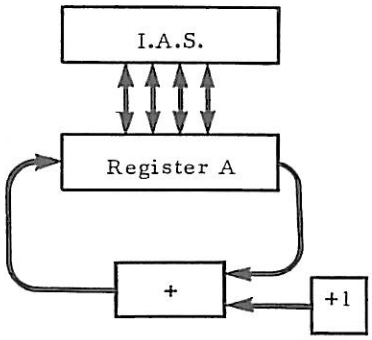
I.A.S.	0	0	0	0	0	0	0	5	8	2	1	6
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	0	0	6	5	4	3

After

I.A.S.	0	0	0	0	0	0	0	5	1	6	7	3
Register A	0	0	0	0	0	0	0	5	1	6	7	3
Register B	0	0	0	0	0	0	0	0	6	5	4	3

Effect This function ADDS 1 (one) to the least-significant position of the I.A.S. word specified in the instruction.

Operation The specified I.A.S. word is put in Register A, 1 is added in the Mill to the contents of Register A and the sum placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the sum. The contents of Register B are not affected by this instruction.



Example Add 1 to word 12 of block 24

Instruction

D	F	A	R
--	66	0012	24
--	---	------	---

Before

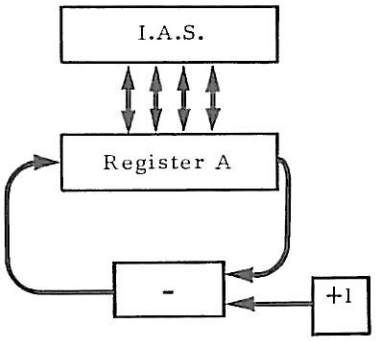
I.A.S.	0	0	0	0	0	0	4	1	3	1	2	9
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	0	0	0	0	0	0

After

I.A.S.	0	0	0	0	0	0	4	1	3	1	3	0
Register A	0	0	0	0	0	0	4	1	3	1	3	0
Register B	0	0	0	0	0	0	0	0	0	0	0	0

Effect This function SUBTRACTS 1 (one) from the least-significant position of the I.A.S. word specified in the instruction.

Operation The specified I.A.S. word is put in Register A, 1 is subtracted in the Mill from the contents of Register A and the difference placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the difference. The contents of Register B are not affected by this instruction.



Example Subtract 1 from word 17 of block 22.

Instruction

D	F	A	R
--	67	0017	22
--	---	------	---

Before

I.A.S.	0	0	0	0	0	0	0	0	0	0	0	0
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	0	0	5	6	4	2

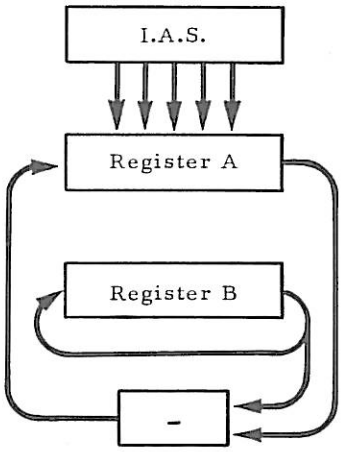
After

I.A.S.	9	9	9	9	9	9	9	9	9	9	9	9
Register A	9	9	9	9	9	9	9	9	9	9	9	9
Register B	0	0	0	0	0	0	0	0	5	6	4	2

The result in word 17 block 22 is the equivalent of -1 as is also the content in Register A.

Effect COMPARES by subtracting the contents of Register B from the contents of the specified I.A.S. location, the result being placed in Register A.

Operation The contents of the specified location are placed in Register A, the contents of Register B are subtracted in the Mill from the contents of Register A, and the difference placed in Register A. The original contents of both Register B and the I.A.S. location are unaffected.



Example Compare the contents of Register B with word 17 of block 22.

Instruction

D	F	A	R
--	68	0017	22
--	---	-------	---

Before

I.A.S.	0	0	0	0	0	0	0	5	9	4	3	1
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	0	5	8	7	2	1

After

I.A.S.	0	0	0	0	0	0	0	5	9	4	3	1
Register A	0	0	0	0	0	0	0	0	0	7	1	0
Register B	0	0	0	0	0	0	0	5	8	7	2	1

STERLING ADDITION AND SUBTRACTION 2.4

Special instructions enable sterling arithmetic to be performed. A Sterling Position Register is also provided which according to its setting ensures that sterling arithmetic is correctly carried out whatever the positions in a word of the pounds, shillings and pence columns. Thus the programmer can make best use of the 12-digit word, be it for several decimal places of pence or alternatively, when working with large values, as many places for pounds as possible.

When sterling arithmetic is performed the computer assumes that the sterling positions conform to the setting of the Sterling Position Register. It is therefore essential that, before an arithmetic instruction is given, the 10/- position in the operands concerned is as indicated by the Sterling Position Register. If the numbers are not correctly positioned they will be mutilated due to sterling carries being performed on the wrong digits.

The Sterling Position Register indicates the position of the tens of shillings (10/-) digit. Two digit positions, tens of shillings (10/-) and units of shillings (1/-) are allowed for shillings and one digit position for pence (excluding decimal positions). 10d and 11d are held in one digit position. The Sterling Position Register must be set before any sterling arithmetic is carried out and remains set until it is subsequently reset to another value.

Function 22 2.4.1

Effect Sets the Sterling Position Register according to the two least-significant digits of the address in the instruction.

Operation The position of the tens of shillings (10/-) digit is determined by using an address in the range 0002 to 0013. This determined position will remain set for all subsequent sterling arithmetic until reset by a further function 22.

Example To specify the pence in position 11 set the Sterling Position Register to 9, indicating the position of the tens of shillings (10/-) digit. Thus the instruction

D	F	A	R
	22	0009	-

causes

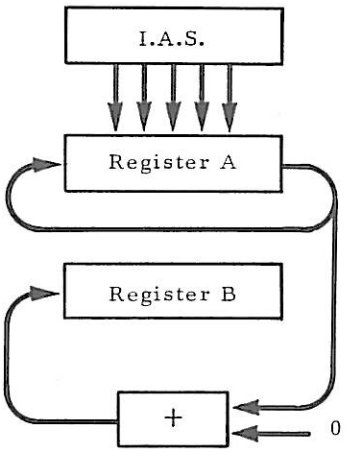
0	0	0	2	9	4	6	2	1	9	10	0

to be operated on as £29462..19..10.0d

Notes It is *not permissible* to allocate position 1 for the tens of shillings (10/-) position. It is permissible to set the Sterling Position Register to 13 in which case sterling functions will be performed as if they were decimal functions. This is useful where a section of program may sometimes be using sterling factors and sometimes decimal factors, the digit entered by function 22 in the Sterling Position Register enabling the program to operate in either sterling or decimal.

Effect CLEAR ADDS the sterling contents of a specified location of I.A.S. into Register B.

Operation The sterling contents of the specified I.A.S. location are put in Register A, added to zero in the Mill and the result placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The I.A.S. word is unaltered.



Example Clear add £25..12..10d recorded as word 19 block 25 in Register B allowing two decimal places of pence. The Sterling Position Register is set to 8.

Instruction

D	F	A	R
--	70	0019	25
--	---	-------	---

Before

I.A.S.	0	0	0	0	0	2	5	1	2	10	0	0
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	5	0	8	11	3	5

After

I.A.S.	0	0	0	0	0	2	5	1	2	10	0	0
Register A	0	0	0	0	0	2	5	1	2	10	0	0
Register B	0	0	0	0	0	2	5	1	2	10	0	0

Notes Any decimal number being processed by this function will be mutilated during its progress through the Mill as illustrated on the next page.

Example Clear Add 45678 recorded as word 19 block 25 into Register B. The Sterling Position Register is set to position 10.

Instruction

D	F	A	R
--	70	0019	25

Before

I.A.S.	0	0	0	0	0	0	0	4	5	6	7	8
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	0	0	5	9	2	3

After

I.A.S.	0	0	0	0	0	0	0	4	5	6	7	8
Register A	0	0	0	0	0	0	0	4	5	6	7	8
Register B	0	0	0	0	0	0	0	4	6	4	7	8

A 37 instruction is usually preferable to a 70 instruction, unless it is required to set the Mill and overflow indicators.

Functions 70 to 78 — 2.4.3

Functions 70 to 78 operate in sterling in the same manner that functions 60 to 68 do in decimal. Apart from function 70 just described only the function and brief operation details of each sterling function are given.

Function 71 — 2.4.4

Effect CLEAR SUBTRACTS the sterling contents of a specified location of I.A.S. into Register B.

Operation The sterling contents of the specified I.A.S. location are put in Register A, subtracted from zero in the Mill and the result placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The I.A.S. word is unaltered.

Function 72 — 2.4.5

Effect ADDS the sterling contents of a specified location of I.A.S. to the sterling contents of Register B and places the result in Register B.

Operation The sterling contents of the specified I.A.S. location are put in Register A, added in the Mill to the sterling contents of Register B and the result placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The I.A.S. word is unaltered.

Function 73 — 2.4.6

Effect SUBTRACTS the sterling contents of a specified location of I.A.S. from the sterling contents of Register B and places the result in Register B.

Operation The sterling contents of the specified I.A.S. location are put in Register A, subtracted in the Mill from the sterling content of Register B and the difference placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The I.A.S. word is unaltered.

Function 74 2.4.7

Effect This function ADDS the sterling contents of Register B to the sterling contents of a specified location of I.A.S. and places the result in that location.

Operation The sterling contents of the specified I.A.S. location are put in Register A, added in the Mill to the sterling contents of Register B and the sum placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the result, which may be useful in certain cases. The original contents of Register B remain unaltered.

Function 75 2.4.8

Effect This function SUBTRACTS the sterling contents of Register B from the sterling contents of a specified location of I.A.S. and places the result in that location.

Operation The sterling contents of the specified I.A.S. location are put in Register A, subtracted in the Mill from the sterling contents of Register B and the difference placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the difference. The original contents of Register B remain unaltered.

Function 76 2.4.9

Effect This function ADDS 1 (one) to the least-significant position of the sterling contents of the I.A.S. word specified in the instruction.

Operation The sterling contents of the specified I.A.S. location are put in Register A, 1 is added to the sterling contents of Register A in the Mill and the sum placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the sum. The contents of Register B are not affected by this instruction.

Function 77 2.4.10

Effect This function SUBTRACTS 1 (one) from the least-significant position of the sterling contents of the I.A.S. word specified in the instruction.

Operation The sterling contents of the specified I.A.S. location are put in Register A, 1 is subtracted from the sterling contents of Register A in the Mill and the difference placed in the specified I.A.S. location by way of Register A. Thus on completion of this function Register A will also contain the difference. The contents of Register B are not affected by this instruction.

Function 78 2.4.11

Effect COMPARES by subtracting the sterling contents of Register B from the sterling contents of a specified I.A.S. location, the result being placed in Register A.

Operation The sterling contents of the specified location are placed in Register A, the sterling contents of Register B are subtracted in the Mill from the sterling contents of Register A,

and the difference placed in Register A. The original contents of both Register B and the I.A.S. location are unaffected.

I	D	F	A	R	NARRATIVE
0		22	0008	-	10/- Position - Position 8
		70	0041	19	Clear Add I.A.S. 41 block 19 to Register B
1		72	0017	19	Add IAS. 17 block 19 to Register B
		74	0019	19	Add Register B to I.A.S. 19 block 19
2		22	0007	-	10/- Position - Position 7
		70	0049	16	Clear Add I.A.S. 49 block 16 to Register B
3		73	0046	16	Subtract I.A.S. 46 block 16 from Register B
		74	0051	16	Add Register B to I.A.S. 51 block 16
4		76	0051	16	Add 1 to I.A.S. 51 block 16

Figure 5: TYPICAL PORTION OF PROGRAM FOR STERLING CALCULATIONS

MULTIPLICATION AND THE DECIMAL POINT REGISTER 2.5

Multiplication is performed by a routine built into the computer and can be initiated by a single instruction. The sequence of operations requires the use of Registers A, B and C and the Mill to perform repeated additions and subtractions for each digit in the multiplier. The multiplier, which must be a decimal number, must be in Register B: the multiplicand, either a decimal or sterling number, must be in I.A.S. and the product will be placed in both Registers B and C. The logic for the multiplication routine is illustrated by the flowchart in Section.2.5.10.

Multiplication can be either

Decimal x Decimal = Decimal (Function 69)

or

Sterling x Decimal = Sterling (Function 79).

Decimal places can be accommodated and multiplication of negative multipliers and multiplicands can be performed. Multipliers and multiplicands should not contain a digit other than 0 or 9 in the sign position (position 1) nor should they consist of a word of all zeros except for a nine in the most -significant position.

The digit positions of both multipliers and multiplicands should not contain values greater than 9 except for $\overline{10}$ or $\overline{11}$ held in the pence position of a sterling multiplicand.

Various examples are given after the functions for multiplication and setting of the Decimal Point Register have been described.

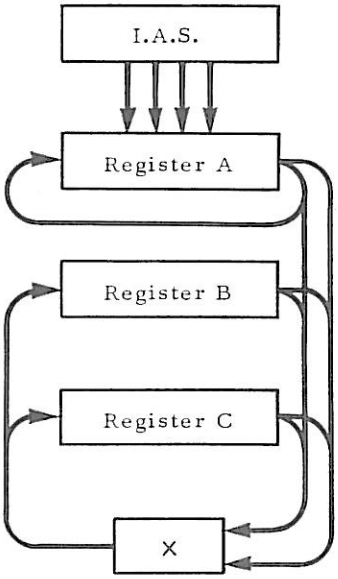
Decimal Multiplication 2.5.1

Although a 24-digit (double-length) product is theoretically produced within the computer, only twelve digits of the product are held at any one time, therefore twelve digits are available as the result, the selection of which twelve being determined by the setting of the Decimal Point Register (see 2.5.5).

Function 69 2.5.2

Effect The contents of a specified location of I.A.S. are multiplied by the contents of Register B. The twelve-digit product is placed in Registers B and C.

Operation The operation which is automatic is detailed in Section 2.5.10. The twelve-digit product formed in Register C is also placed in Register B. During multiplication and at the conclusion of the multiply instruction the specified word of I.A.S. (the multiplicand factor) remains unaltered. The previous contents of Register A are destroyed.



Notes This function causes data to be processed by the Mill and consequently, depending on the resultant product, the Mill Indicators and the Overflow Indicator may be set.

As the time taken to perform multiplication largely depends on the number of digits in the multiplier it is normally advantageous to position the smaller factor in Register B.

Sterling Multiplication 2.5.3

Before attempting to perform sterling multiplication ensure:

(a) that the sterling factor, the multiplicand, is in the specified location of I.A.S.,

(b) that the decimal factor, the multiplier, is in Register B,

(c) the Sterling Position Register is correctly set by using function 22.

Unless (a) and (b) are conformed to, the multiplication routine may enter a closed loop.

Function 79 2.5.4

Effect The sterling contents of a specified location of I.A.S. are multiplied by the decimal contents of Register B. The twelve-digit product is placed in Registers B and C.

Operation The operation which is automatic is detailed in Section 2.5.10. The twelve-digit sterling product formed in Register C is also placed in Register B. During multiplication and at the conclusion of the multiply instruction the sterling contents of the specified location of I.A.S. (the multiplicand) remain unaltered. The previous contents of Register A are destroyed.

Notes As in decimal multiplication, data is processed by the Mill, thus the Mill Indicators and the Overflow Indicator are affected by the product.

The tens of shillings position of the resultant product will be the same as that of the multiplicand factor.

The Decimal Point Register 2.5.5

The Decimal Point Register is a one-digit subtraction counter which may be set by program to any value from 0 to 12. During multiplication, pulses are automatically fed to the counter and cause it to count down and feed a signal to control multiplication. Thus the setting of the Decimal Point Register is used for scaling during multiplication to produce the correct result.

As stated earlier, although the multiplication of two twelve-digit numbers results in a 24-digit product, only twelve digits can be formed in Register B. To determine which twelve of the 24 digits shall be retained divide the 24-digit product by 10^n where n is the number entered in the Decimal Point Register thus losing digits from the least-significant end. The twelve least-significant digits of the result are now taken as the final product.

When performing decimal arithmetic, n corresponds to the number of digits to be lost from the least-significant end of the 24-digit product. For example consider

000000004321 x 000000000005

= 000000000000000000021605

if the Decimal Point Register is set to 3 the product is divided by 10^3 (1, 000) giving 000000000000000000000021. As only the twelve least-significant digits remaining can be formed then the result registered is 000 000 000 021.

When decimal numbers are held in the computer any decimal points exist only in the mind of the programmer and are not physically recorded in the registers. If in the multiplication shown above, the factors comprised .4321 x .5 the product would be .21 (605). The assumed decimal point after position 8 of the multiplicand is moved to after position 10 in the product. This could be a useful technique if the result is required to a specified number of decimal places.

If the Decimal Point Register is set to the number of decimal places in the multiplier word, then the decimal point is in the same position in the resulting product as it was in the multiplicand.

Consider again 000 000 004321 × 000000000005 representing .4321 × .5.

Set the Decimal Point Register to 1 as the number of decimal places in the multiplier is 1.

In the resulting product 000 000 002 160 there are effectively four decimal places with the decimal point after digit position 8 as it was in the multiplicand.

When sterling numbers are held in the computer, the positions of the sterling components are determined by the setting of the Sterling Position Register. In effect, this means that the decimal point is also fixed, being located after the pence position to denote decimals of pence.

When performing sterling multiplication, it is therefore apparent that the Decimal Point Register must be set to the number of decimal places in the multiplier to ensure that the decimal point is in the same position in the resulting product as it was in the multiplicand.

If £472.14.9 × 0.00429 is to result in £2.00.6 (to nearest penny below) with the Sterling Position Register set at 10 then the Decimal Point Register must be set to 5. Failure to set the Decimal Point Register will cause, in effect, £472.14.9 × 429 resulting in £202,804.07.9.

The rules for setting the Decimal Point Register may be summarized as follows:

Decimal × Decimal - Decimal Point Register setting = Number of digits to discard from least-significant end of product.

Sterling × Decimal - Decimal Point Register setting = Number of decimal places in multiplier word.

Function 21

2.5.6

Effect Sets the Decimal Point Register according to the two least-significant digits of the address in the instruction, which are in the range 00 to 12.

Operation The Decimal Point Register must be set by function 21 immediately prior to the multiply instruction, thus ensuring that during the automatic multiplication routine actuated by functions 69 and 79 the correct number of shifts take place. The Decimal Point Register is automatically set to zero when the multiplication has been completed.

If the Decimal Point Register is set and the multiply instruction does not occur until some time later, then the register will be reset to zero by the next instruction (see Example 2). An exception to this rule occurs if an instruction with designation 4, 8 or 9 or a function 00, 11, 22, 38, 39 and 80 to 87 is interposed between the 21 and 69 or 79 instructions.

Example 1 Set Decimal Point Register to 4.

Instruction

I	D	F	A	R
X	--	21	0004	-

Example 2 Set the Decimal Point Register to 4 prior to performing decimal multiplication.

Correct Instructions

I	D	F	A	R
X		37	0123	27
		21	0004	-
X+1		69	0124	27

Incorrect Instructions

I	D	F	A	R
X		21	0004	-
		37	0123	27
X+1		69	0124	27

The incorrect set of instructions causes the Decimal Point Register to be reset to zero by the 37 instruction before multiplication takes place.

Note The programmer must ensure that the setting of the Decimal Point Register will not cause vital non-zero digits to be lost from the most-significant end.

Decimal Multiplication Examples 2.5.7

Example 1

Multiply the decimal contents of I.A.S. 14 block 23 by the decimal contents of
I.A.S. 15 block 23 and store the result in
I.A.S. 16 block 23

Instructions

I	D	F	A	R	NARRATIVE
X		37	0015	23	Transfer I.A.S. to Register B
		21	00??	-	Set Decimal Point Register
X+1		69	0014	23	Multiply
		42	0016	23	Store in I.A.S.

Word 14 block 23 000, 000, 123, 456
Word 15 block 23 × 000, 000, 999, 999
= 000, 000, 000, 000, 123, 455, 876, 544

The digit in the Decimal Point Register determines the position and accuracy of the product in Registers B and C and consequently in I.A.S. 16 block 23.

Decimal Point Register	Product in Register B and Register C
0	123, 455, 876, 544 (sets Overflow Indicator)
1	012, 345, 587, 654 (4)
2	001, 234, 558, 765 (44)
3	000, 123, 455, 876 (544)
4	000, 012, 345, 587 (6544)
5	000, 001, 234, 558 (76544)
6	000, 000, 123, 455 (876544)
7	000, 000, 012, 345 (5876544)
8	000, 000, 001, 234 (55876544)
9	000, 000, 000, 123 (455876544)
10	000, 000, 000, 012 (3455876544)
11	000, 000, 000, 001 (23455876544)
12	000, 000, 000, 000 (123455876544)

The digits in brackets are those which are lost as a result of the content of the Decimal Point Register. There are no decimal places in the multiplier therefore a setting of 0 locates the decimal point in the same position as it is in the multiplicand i.e. after digit 12.

The original contents of I.A.S. 14 block 23 and I.A.S. 15 block 23 are unaltered at the completion of the instructions.

Example 2 Multiply the decimal contents of I.A.S. 92 block 41 by the decimal contents of I.A.S. 76 block 59 and store the result in I.A.S. 48 block 17.

Instructions

I	D	F	A	R	NARRATIVE
X	--	37	0076	59	Transfer I.A.S. to Register B
		21	00??	-	Set Decimal Point Register
X+1	--	69	0092	41	Multiply
		43	0048	17	Transfer (from Register C) to I.A.S.

Word 76 block 59 0.050, 000, 000, 00

Word 92 block 41 × 0.300, 000, 000, 00

= 00.015, 000, 000, 000, 000, 000, 000, 0

Decimal Point Register	Product in Register B and Register C
0 - 7	000, 000, 000, 000
8	500, 000, 000, 000
9	150, 000, 000, 000
10	015, 000, 000, 000
11	001, 500, 000, 000
12	000, 150, 000, 000

In this example any digit between 0 and 8 set in the Decimal Point Register will cause significant product digits to be lost. A setting of 11 locates the decimal point in the same position as it is in the multiplicand i.e. after position 1.

Example 3

000, 000, 000, 123
x 000, 000, 000. 005
= 000, 000, 000, 000, 000, 000, 000.615

Decimal Point Register	Product in Register B and Register C
0	000, 000, 000.615
1	0, 000, 000, 000.61
2	00, 000, 000, 000.6
3 - 12	000, 000, 000, 000

Assuming that it is necessary to have the result with one decimal place, then 2 will be set in the Decimal Point Register.

Sterling Multiplication Example 2.5.8

Example Multiply the sterling contents of I.A.S. 19 block 24 by the decimal contents of I.A.S. 24 block 39 and store the sterling result in I.A.S. 92 block 17. The tens of shillings in the sterling factor is in position 6.

Instructions

I	D	F	A	R	NARRATIVE
X	--	37	0024	39	Transfer multiplier to Register B
		21	00??	-	Set Decimal Point Register
X+1	--	22	0006	-	Set Sterling Position Register
		79	0019	24	Multiply
X+2	--	42	0092	17	Store in I.A.S.

05, 437.18.4.0000
x 000, 000, 000, 008
= 00, 000, 000, 000, 043, 503.06.8.000, 0

Depending on the digit entered in the Decimal Point Register before multiplication there would appear in Registers B and C, and consequently I.A.S. 92 block 17, the following:

Decimal Point Register	Product in Register B and Register C	
0	43503.06.8.0000	($\div 10^0$) (sets Overflow Indicator)
1	04350.06.8.0000	($\div 10^1$)
2	00435.00.8.0000	($\div 10^2$)
3	00043.10.0.8000	($\div 10^3$)
4	00004.07.0.0800	($\div 10^4$)
5	00000.08.8.4080	($\div 10^5$)
6	00000.00.10.4408	($\div 10^6$)
7	00000.00.1.0440 (8)	($\div 10^7$)
8	00000.00.0.1044 (08)	($\div 10^8$)
9	00000.00.0.0104 (408)	($\div 10^9$)
10	00000.00.0.0010 (4408)	($\div 10^{10}$)
11	00000.00.0.0001 (04408)	($\div 10^{11}$)
12	00000.00.0.0000 (104408)	($\div 10^{12}$)

If the multiplier had been 0.08 then a setting of 2 in the Decimal Point Register would have given the correct result.

The product (with 10/- in position 6) will be in both Registers B and C on completion of the multiply instruction and therefore can be stored in I.A.S. 92 block 17 with either a 42 or 43 instruction.

Negative Factors in Multiplication 2.5.9

Example Multiply the decimal contents of I.A.S. 196 block 37 by the negative decimal contents of I.A.S. 43 block 38 and store the negative result in I.A.S. 47 block 38.

Instructions

I	D	F	A	R
X	--	37	0043	38
		69	0196	37
X+1	--	42	0047	38

	000, 000, 000, 129	
x	999, 999, 924, 926	or effectively x (- 75074)
=	999, 990, 315, 454	= - 9684546

The Decimal Point Register is set to zero.

The reader may satisfy himself that similar negative multiplications can be performed when:

(a) the content of the multiplier is negative, or

(b) the content of the multiplicand is negative, either producing a negative result, or

(c) both the contents of the multiplier and of the multiplicand are negative thereby producing a positive result, or

(d) the above when the contents of the multiplicand are sterling.

Narrative

1 To commence the multiplicand is in I.A.S. and the multiplier is in Register B; this stage is achieved by the program.

2 This and following stages are all automatically achieved by the built-in routine.
(a) Unsetting indicator X.
(b) Multiplicand is put in Register A from the I.A.S.
(c) Put zeros in Register C which is used to accumulate the product.

3 (a) The content of Register A is added to that of Register C, the sum being placed in Register C.
(b) Indicator X is set.
(c) The content of Register B is reduced by one.

4 (a) The content of Register A is subtracted from Register C the difference being placed in Register C.
(b) Indicator X is unset.
(c) The content of Register B is increased by one.

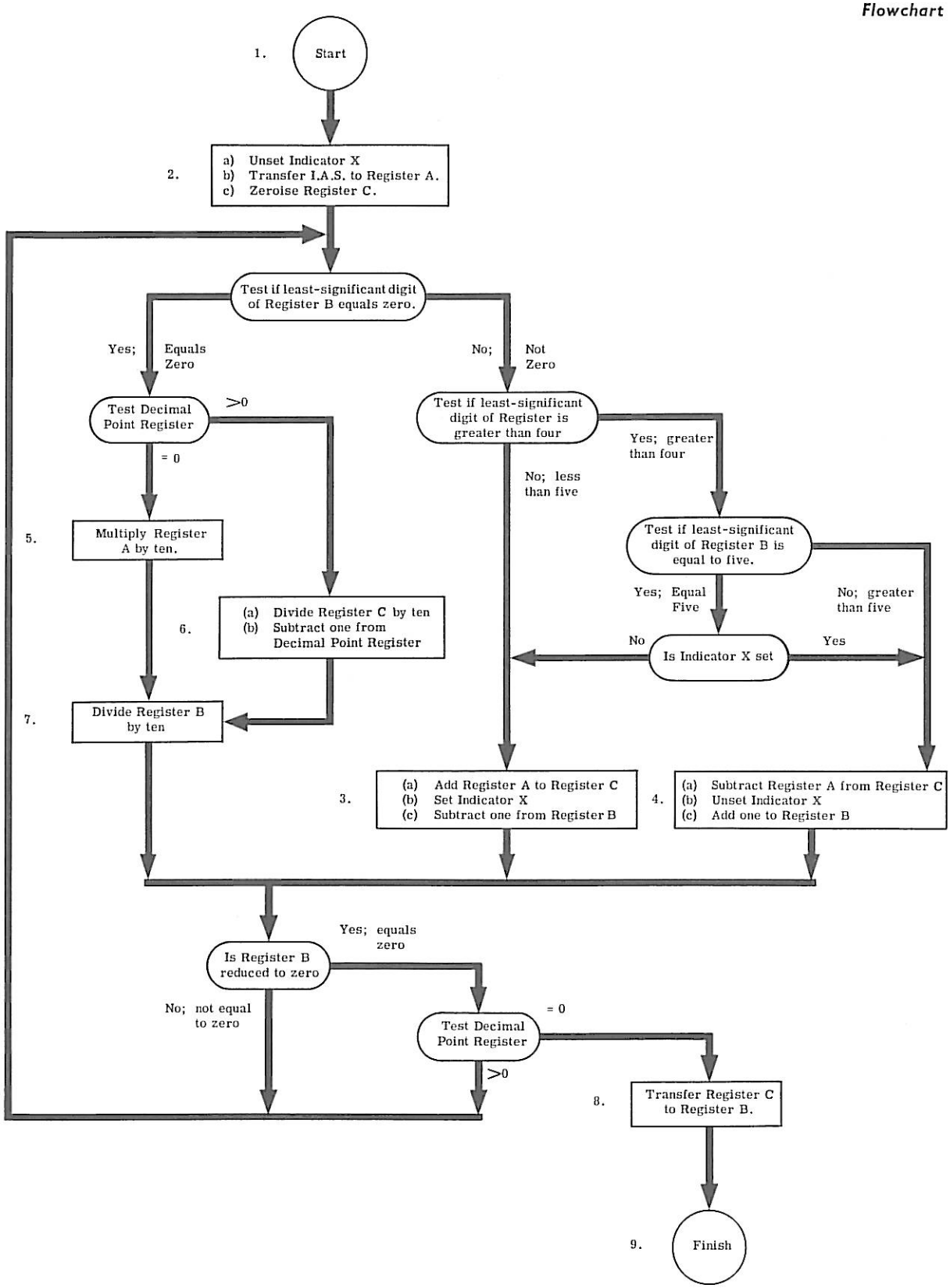
5 The content of Register A is multiplied by ten.

6 (a) The content of Register C is divided by ten. Sign is propagated at the most-significant end.
(b) The Decimal Point Register is counted down by one.

7 The content of Register B is divided by ten. The sign is propagated at the most-significant end.

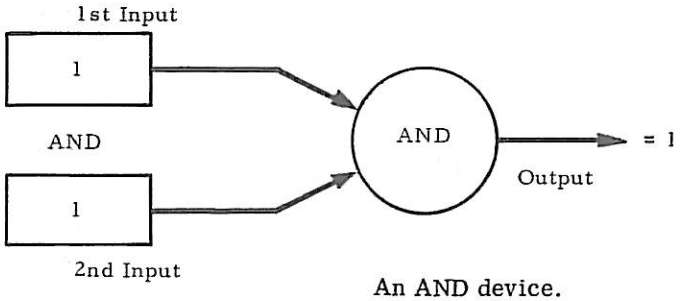
8 The accumulated product in Register C is transferred to Register B.

9 At the conclusion of the multiply function the multiplicand is still in I.A.S. The product is in Registers B and C. The original contents of Register A are mutilated.


1.
Start
2.
a) Unset Indicator X
b) Transfer I.A.S. to Register A.
c) Zeroise Register C.
Test if least-significant digit of Register B equals zero.
Yes; Equals Zero
No; Not Zero
Test Decimal Point Register
>0
= 0
Test if least-significant digit of Register is greater than four
Yes; greater than four
No; less than five
5.
Multiply Register A by ten.
Test if least-significant digit of Register B is equal to five.
Yes; Equal Five
No; greater than five
6.
(a) Divide Register C by ten
(b) Subtract one from Decimal Point Register
No
Is Indicator X set
Yes
7.
Divide Register B by ten
3.
(a) Add Register A to Register C
(b) Set Indicator X
(c) Subtract one from Register B
4.
(a) Subtract Register A from Register C
(b) Unset Indicator X
(c) Add one to Register B
Yes; equals zero
Is Register B reduced to zero
No; not equal to zero
= 0
Test Decimal Point Register
>0
8.
Transfer Register C to Register B.
9.
Finish

The two functions 35 and 36 are termed Logical AND and Logical OR respectively. The logic of AND is:

(a) 0 AND 0 = 0
(b) 1 AND 0 = 0
(c) 0 AND 1 = 0
(d) 1 AND 1 = 1

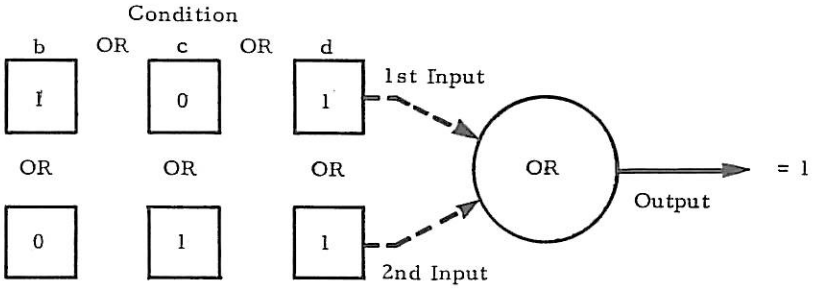


An AND device.

It is apparent that an AND device comprising two inputs must have a 1 at both inputs before a 1 will appear at its single outlet. The only time a 1 would be available at the output of the device illustrated above would be condition (d) 1 AND 1 = 1.

The logic of OR is:

(a) 0 OR 0 = 0
(b) 1 OR 0 = 1
(c) 0 OR 1 = 1
(d) 1 OR 1 = 1



An OR device.

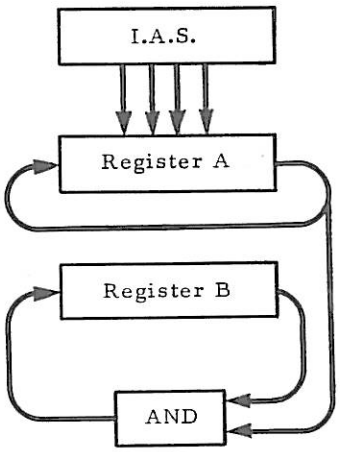
As shown an OR device will have a 1 available at its output when a 1 is entered at either input or both inputs, i.e. conditions (b), (c) and (d) but not (a) cause a 1 to be available at the output.

This concept can be applied to data in the computer by inspecting the binary coded representation or bits of the digits within a word. Thus those bits which match as shown in the logic above will be available in Register B at the conclusion of an instruction. Therefore logical functions are used when it is required to refer to parts of words rather than the complete word. A part may consist of several digits of a word or simply one or more bits which make up a single digit. When using logical functions it is important to remember the binary coded representation of the digits within a word.

Function 35 2.6.1

Effect Causes the operation of Logical AND to be carried out between the contents of Register B and the contents of a specified location of I.A.S. on a bit-for-bit basis.

Operation A number held in the specified I.A.S. location is put in Register A. Logical AND of the contents of Registers A and B then takes place in the Mill and the result is placed in Register B. At the conclusion of the instruction the contents of Register A will be the original I.A.S. word. The I.A.S. word is unaltered.



Examples If 2916359874 is contained in I.A.S. 15 block 12 and 4174322861 is contained in Register B, then at the completion of function 35 the registers and I.A.S. 15 block 12 will stand thus.

I.A.S.	0	0	2	9	1	6	3	5	9	8	7	4
Register A	0	0	2	9	1	6	3	5	9	8	7	4
Register B	0	0	0	1	1	4	3	0	0	8	6	0

A study of the binary representation will clarify this.

Word 15 Block 12

0	0	2	9	1	6	3	5	9	8	7	4	
0	0	0	1	1	0	1	1	1	0	1	0	1
0	0	1	0	0	1	1	0	0	0	1	0	2
0	0	0	0	0	1	0	1	0	0	1	1	4
0	0	0	1	0	0	0	0	1	1	0	0	8

Register B

0	0	4	1	7	4	3	2	2	8	6	1	
0	0	0	1	1	0	1	0	0	0	0	1	1
0	0	0	0	1	0	1	1	1	0	1	0	2
0	0	1	0	1	1	0	0	0	0	1	0	4
0	0	0	0	0	0	0	0	0	1	0	0	8

Result in Register B

0	0	0	1	1	4	3	0	0	8	6	0	
0	0	0	1	1	0	1	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	1	0	2
0	0	0	0	0	1	0	0	0	0	1	0	4
0	0	0	0	0	0	0	0	0	1	0	0	8

Three further examples of the use of this function are given although it has many varied one-off uses to which it may be put.