1 MASKING TO EXTRACT PART OF A WORD OF I.A.S.

Word 19 of block 14 contains unwanted data in positions 1 to 6, the data in position 7 to 12 is required in Register B.

Instruction

	I	D	F	A	R
X		--	37	0019	14
			35	0021	16
X+1		--	---	------	---

data
mask

Before

I.A.S. 21 Block 16

0	0	0	0	0	0	15	15	15	15	15	15

or

0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	1	1	1	1	1	1	2
0	0	0	0	0	0	1	1	1	1	1	1	4
0	0	0	0	0	0	1	1	1	1	1	1	8

I.A.S. 19 Block 14

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

After

I.A.S. 21 Block 16	0	0	0	0	0	0	15	15	15	15	15	15
I.A.S. 19 Block 14	5	6	4	9	3	2	1	6	8	4	7	8
Register A	0	0	0	0	0	0	15	15	15	15	15	15
Register B	0	0	0	0	0	0	1	6	8	4	7	8

Only the matching binary digits for each position arrive in Register B.

2 MASKING TO EXTRACT AN ADDRESS

Mask constants can be used to extract the address from an instruction. An example of this use would be where the instruction tests an indicator, as in this case the address has 4000 added to it. The 4000 can be subsequently removed by using a mask as below

either

0	0	3	15	15	15	0	0	0	0	0	0

or

0	0	0	0	0	0	0	0	3	15	15	15

depending on whether the instruction to be masked is in the most-significant or least-significant half of the word.

3 ODD/EVEN NUMBER DISCRIMINATION

A mask can be used to determine whether a number is odd or even. Here word 17 block 19 is examined, the mask being word 14 block 13.

Instruction

	I	D	F	A	R
X		--	60	0014	13
			35	0017	19
X+1		--	---	------	---

Before

I.A.S. 14 Block 13	0	0	0	0	0	0	0	0	0	0	0	1
I.A.S. 17 Block 19	0	0	0	0	0	0	6	5	2	3	8	3
Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	0	0	0	0	0	0	0	8	9	2	3	4

After

I.A.S. 14 Block 13	0	0	0	0	0	0	0	0	0	0	0	1
I.A.S. 17 Block 19	0	0	0	0	0	0	6	5	2	3	8	3
Register A	0	0	0	0	0	0	6	5	2	3	8	3
Register B	0	0	0	0	0	0	0	0	0	0	0	1

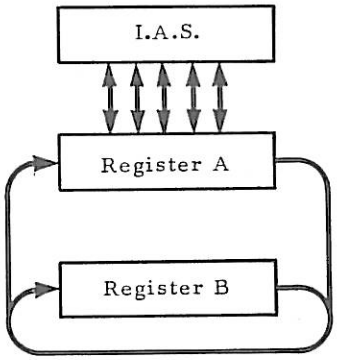
At the conclusion of the 35 instruction, it will be possible to discriminate between the presence or absence of a 1-bit by means of Mill Indicators 01 and 02 (see page 54). In the example above indicator 02 is set, as Register B contains a 1-bit corresponding to the I.A.S. contents being odd. If the contents of Register B are zero corresponding to the I.A.S. contents being even, Mill Indicator 01 is set.

Notes During the execution of function 35 the overflow indicator is inhibited but the other Mill Indicators function normally.

Function 36 2.6.2

Effect Causes the operation of Logical OR 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 OR of the contents of Registers A and B then takes place and the result is placed in Registers A and B and in the original I.A.S. location specified in the instruction.



Examples If 123456123456 is contained in I.A.S. 15 block 12 and 999999666666 is contained in Register B, then at the completion of function 36 the Registers A, B and I.A.S. 15 block 12 will stand thus:

I.A.S.	9	11	11	13	13	15	7	6	7	6	7	6
Register A	9	11	11	13	13	15	7	6	7	6	7	6
Register B	9	11	11	13	13	15	7	6	7	6	7	6

A study of the 1, 2, 4 and 8 streams of the original contents of word 15 block 12 and Register B, together with the result obtained after Logical OR are shown below:-

Word 15 Block 12	1	2	3	4	5	6	1	2	3	4	5	6	
	1	0	1	0	1	0	1	0	1	0	1	0	1
	0	1	1	0	0	1	0	1	1	0	0	1	2
	0	0	0	1	1	1	0	0	0	1	1	1	4
	0	0	0	0	0	0	0	0	0	0	0	0	8

Register B	9	9	9	9	9	9	6	6	6	6	6	6	
	1	1	1	1	1	1	0	0	0	0	0	0	1
	0	0	0	0	0	0	1	1	1	1	1	1	2
	0	0	0	0	0	0	1	1	1	1	1	1	4
	1	1	1	1	1	1	0	0	0	0	0	0	8

	9	11	11	13	13	15	7	6	7	6	7	6	
Result in Word 15	1	1	1	1	1	1	1	0	1	0	1	0	1
Block 12 and	0	1	1	0	0	1	1	1	1	1	1	1	2
Registers A and B	0	0	0	1	1	1	1	1	1	1	1	1	4
	1	1	1	1	1	1	0	0	0	0	0	0	8

The most common use for Logical OR is for modifying an instruction by the insertion of an address without using the Mill and in consequence not setting the overflow indicator. For example to modify the first instruction contained in word 12 of block 23 by inserting an address from word 29 of block 21.

Instructions

I	D	F	A	R
X	--	37	0029	21
		35	0011	23
X+1	--	36	0012	23

Before

I.A.S. 29 Block 21	0	0	1	2	4	8	0	0	0	4	1	9
I.A.S. 11 Block 23	0	0	3	15	15	15	0	0	0	0	0	0
I.A.S. 12 Block 23	6	0	0	0	0	0	6	3	1	1	9	2

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. 29 Block 21	0	0	1	2	4	8	0	0	0	4	1	9
I.A.S. 11 Block 23	0	0	3	15	15	15	0	0	0	0	0	0
I.A.S. 12 Block 23	6	0	1	2	4	8	6	3	1	1	9	2

Register A	6	0	1	2	4	8	6	3	1	1	9	2
Register B	6	0	1	2	4	8	6	3	1	1	9	2

The same result can be achieved without using Logical OR. Whereas there is little difference in the time, this second method affects the overflow indicator.

The instructions would be

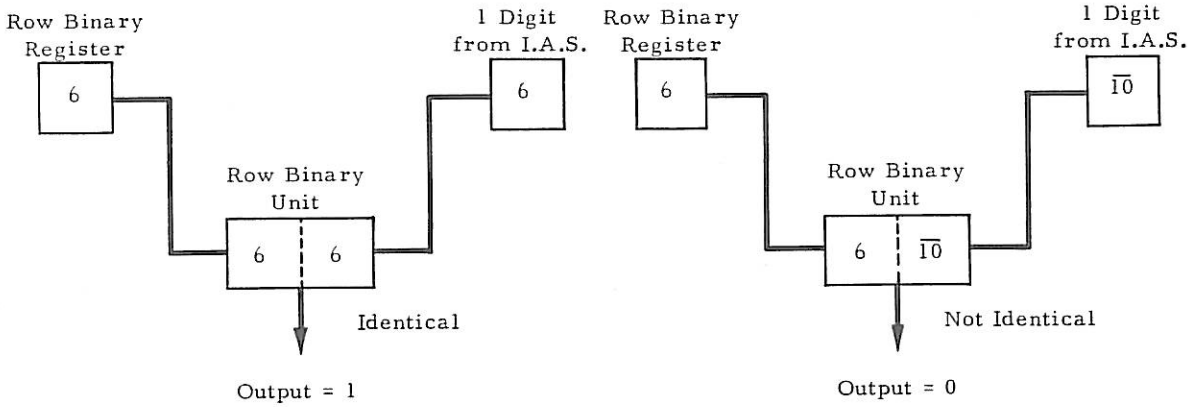
I	D	F	A	R
X	--	37	0029	21
		35	0011	23
X+1	--	64	0012	23

The uses of modifications are discussed in Part 4.

Notes As function 36 does not use the Mill, no Mill Indicators are set by this function.

ROW BINARIZING 2.7

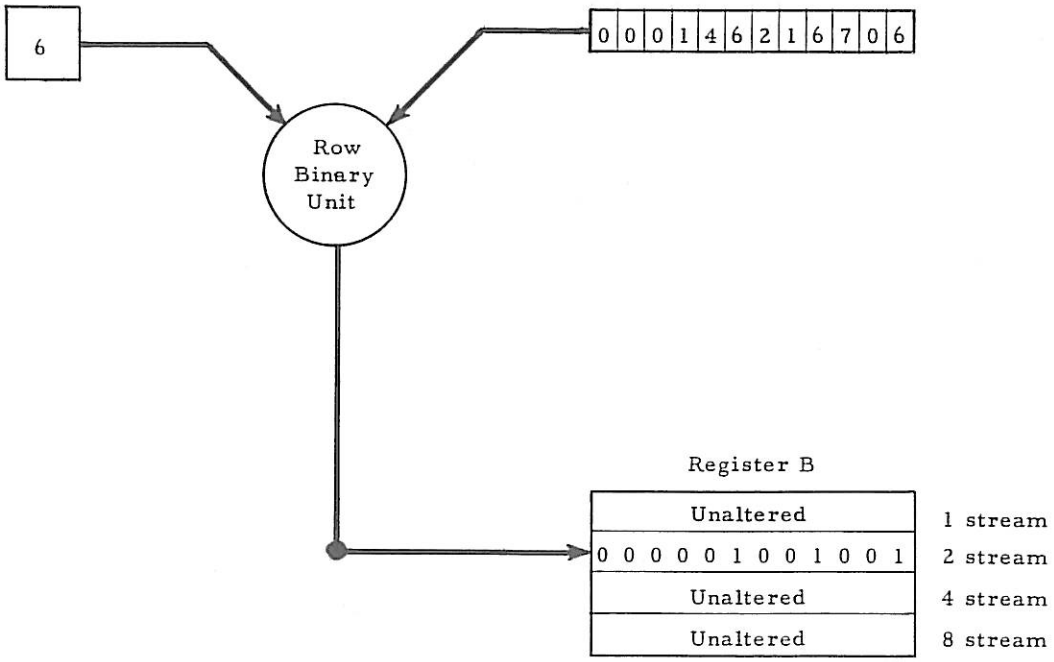
Row-binary is normally created prior to print or punch output as described in Part 3. The components used to create row-binary are Register B, the specified word of I.A.S. and the Row Binary Register together with the Row Binary Unit. As illustrated schematically the Row Binary Register is a single-digit register capable of holding any number from 0 to 15, and the Row Binary Unit uses a technique similar to the Logical AND described in 2.6 but performing the logic on whole digits rather than bits. If a digit from a single position of an I.A.S. location is compared in the Row Binary Unit with the digit in the Row Binary Register, when the digits are identical a 1-bit will be emitted from the output of the device.



According to the function given, the 1s or 0s from the output of the Row Binary Register component can be directed to the appropriate stream of the B Register.

Thus in row binarizing each *digit* of the contents of the specified location of I.A.S. is compared against the digit set in the Row Binary Register. If the digits correspond then a 1-bit is emitted into the appropriate stream of Register B and if the digits fail to correspond a 0 is emitted into the stream of Register B e.g. the Row Binary Register is set to 6 and the instruction is given to create row-binary in stream 2 of Register B for I.A.S. 12 block 11 which contains 000046216706.

6 in the Row Binary Register Word 12 Block 11



The Row Binary Register can be loaded in one of two ways:

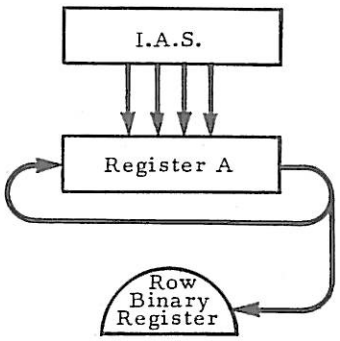
(a) with the least-significant digit of the contents of a specified location of I.A.S. (see function 30 below).

(b) with the contents of the index point (print) counter. This method is described in Part 3.

Function 30 2.7.1

Effect Causes the contents of the least-significant position of a specified location of I.A.S. to be transferred to the Row Binary Register.

Operation A number held in the I.A.S. location specified is put in Register A, the least -significant digit being placed in the Row Binary Register. The eleven most-significant figures do not affect the Row Binary Register. 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 Transfer 6 contained in the least-significant position of word 19 block 12 to the Row Binary Register.

Instruction

I	D	F	A	R
	--	30	0019	12

	Before		After	
I.A.S.	3 7 0 1 4 5 6 2 0 3 9 6		I.A.S.	3 7 0 1 4 5 6 2 0 3 9 6
Register A	0 0 0 0 0 0 0 0 0 0 0 0		Register A	3 7 0 1 4 5 6 2 0 3 9 6
Row Binary Register		8	Row Binary Register	6

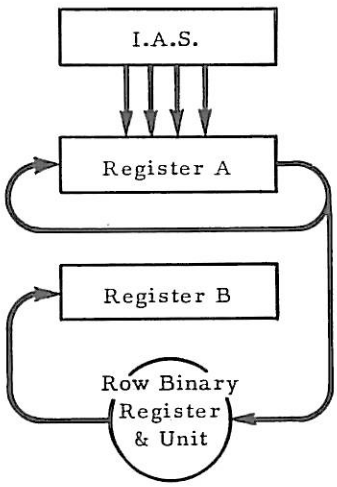
Notes Sometimes a constant can be found in the program with a digit, in the least-significant position of the second address, suitable to load the Row Binary Register. If, however, none is available it is necessary to create a special constant.

Care must be exercised while programming to ensure that the least-significant digit of the chosen constant is that of the *absolute* Address and not the Relative Address and also that the constant is in I.A.S. when function 30 is obeyed. It is safer to use either a constant or an instruction with no relativizer (e.g. those used to set the Decimal Point or Sterling Position Registers) as these will not change if the storage allocation is altered.

Function 31 2.7.2

Effect Creates row-binary into the 1 stream of Register B for the contents of a specified location of I.A.S.

Operation A number held in the specified I.A.S. location is put in Register A and each digit compared with the digit held in the Row Binary Register. When the digits correspond then a 1-bit is emitted into the 1 stream of Register B, when the digits do not correspond then a 0-bit is emitted into the 1 stream of Register B. At the conclusion of the instruction the contents of the 2, 4 and 8 streams of Register B and the original I.A.S. word are unaltered. Also the contents of Register A will be the original I.A.S. word.



Example 1 Create row-binary into the 1 stream of Register B for the contents of I.A.S. 42 block 15. The Row Binary Register contains a 6.

Instruction

D	F	A	R
	31	0042	15

Before

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

Register B													
	0	1	0	0	1	1	0	0	0	1	0	0	1
	0	0	1	0	1	1	1	0	1	0	0	1	2
	0	1	1	0	0	0	0	1	0	1	1	0	4
	1	0	0	1	0	0	0	0	0	0	0	1	8
Row Binary Register												6	

After

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

Register B													
	0	0	1	0	1	0	0	1	0	0	1	1	1
	0	0	1	0	1	1	1	0	1	0	0	1	2
	0	1	1	0	0	0	0	1	0	1	1	0	4
	1	0	0	1	0	0	0	0	0	0	0	1	8
Row Binary Register												6	

Functions 30 to 34 will generally be used for output only, but row-binary techniques can occasionally be used as alternatives to conventional methods of programming in order to save space and/or time.

Example 2 Examine word 17 of block 15 for the presence of digit 3 in position 9.

Instruction

I	D	F	A	R
X		30	0041	19
		31	0017	15
X+1		35	0049	19

Before

I.A.S. 17 Block 15	0	0	0	3	8	3	2	4	3	9	2	6
I.A.S. 41 Block 19	0	0	0	0	8	3	2	6	2	4	0	3
I.A.S. 49 Block 19	0	0	0	0	0	0	0	0	1	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	0	0	0	0
Row Binary Register												0

After

I.A.S. 17 Block 15	0	0	0	3	8	3	2	4	3	9	2	6
I.A.S. 41 Block 19	0	0	0	0	8	3	2	6	2	4	0	3
I.A.S. 49 Block 19	0	0	0	0	0	0	0	0	1	0	0	0

Register A												
	0	0	0	0	0	0	0	0	1	0	0	0

Register B													
	0	0	0	0	0	0	0	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	2
	0	0	0	0	0	0	0	0	0	0	0	0	4
	0	0	0	0	0	0	0	0	0	0	0	0	8
Row Binary Register												3	

By examining Mill Indicators 01 and 02, as described under Section 2.6.1 Example 3 Odd/Even Discrimination, it is possible to jump to the section of the program relating only to digit 3 in position 9 of word 17 of block 15.

Example 3 Examine word 49 of block 22 for the presence of digit 6 in position 4, 7 or 11.

Instruction

I	D	F	A	R
X	--	30	0039	14
		31	0049	22
X+1	--	35	0040	14

Before

I.A.S. 39 Block 14	0	0	0	0	0	0	0	0	0	0	0	6
I.A.S. 40 Block 14	0	0	0	1	0	0	1	0	0	0	1	0
I.A.S. 49 Block 22	0	1	2	6	6	5	7	1	2	8	6	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	1
	0	0	1	0	0	0	0	1	1	0	0	1	2
	0	0	0	0	0	1	0	0	1	0	1	1	4
	0	1	0	1	0	0	0	0	0	1	0	0	8
Row Binary Register												0	

After

I.A.S. 39 Block 14	0	0	0	0	0	0	0	0	0	0	0	6
I.A.S. 40 Block 14	0	0	0	1	0	0	1	0	0	0	1	0
I.A.S. 49 Block 22	0	1	2	6	6	5	7	1	2	8	6	9

Register A	0	0	0	1	0	0	1	0	0	0	1	0

Register B	0	0	0	1	0	0	0	0	0	0	1	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	2
	0	0	0	0	0	0	0	0	0	0	0	0	4
	0	0	0	0	0	0	0	0	0	0	0	0	8
Row Binary Register												6	

By examining Mill Indicators 01 and 02, as described under Section 2.6.1 Example 3 Odd/Even Discrimination, it is possible to jump to the section of the program relating only to instances where a 6 appears in position 4, 7 or 11 of word 49 of block 22.

Note No Mill Indicators are set by function 31 as the Mill is not involved in the operation.

Function 32 2.7.3

Function 32 creates row-binary into the 2 stream of Register B for the contents of a specified word of I.A.S. in the manner described in 2.7.2.

Function 33 2.7.4

Function 33 creates row-binary into the 4 stream of Register B for the contents of a specified word of I.A.S. in the manner described in 2.7.2.

Function 34 2.7.5

Function 34 creates row-binary into the 8 stream of Register B for the contents of a specified word of I.A.S. in the manner described in 2.7.2.

Example of the combined use of Functions 30 to 35 2.7.6

Examine word 16 of block 21 for the presence of digit 3, 6, 7 or 8 in positions 11 or 12.

Instruction

I	D	F	A	R
x	--	30	0039	19
		31	0016	21
x+1	--	30	0040	19
		32	0016	21
x+2	--	30	0041	19
		33	0016	21
x+3	--	30	0042	19
		34	0016	21
x+4	--	35	0043	19

Before

I.A.S. 39 Block 19	8	4	0	4	2	7	9	1	5	8	6	3
I.A.S. 40 Block 19	5	7	9	1	3	6	2	8	0	5	4	6
I.A.S. 41 Block 19	3	7	7	7	5	1	9	6	1	2	4	7
I.A.S. 42 Block 19	3	2	4	8	4	6	4	2	7	1	1	8
I.A.S. 43 Block 19	0	0	0	0	0	0	0	0	0	0	15	15
I.A.S. 16 Block 21	0	0	7	9	6	5	8	9	2	4	3	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	0	0	0	0

Row Binary Register

0

After

I.A.S. 39 Block 19	8	4	0	4	2	7	9	1	5	8	6	3
I.A.S. 40 Block 19	5	7	9	1	3	6	2	8	0	5	4	6
I.A.S. 41 Block 19	3	7	7	7	5	1	9	6	1	2	4	7
I.A.S. 42 Block 19	3	2	4	8	4	6	4	2	7	1	1	8
I.A.S. 43 Block 19	0	0	0	0	0	0	0	0	0	0	15	15
I.A.S. 16 Block 21	0	0	7	9	6	5	8	9	2	4	3	8

Register A

0	0	0	0	0	0	0	0	0	0	15	15

Register B

0	0	0	0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	0	1	8

Row Binary Register

8

By testing Mill Indicators 01 and 02, as described under Section 2.6.1 Example 3 Odd/Even Discrimation, it is possible to jump to the section of the program relating only to instances where a 3, 6, 7 or 8 appears in positions 11 or 12 of word 16 of block 21.

SHIFT INSTRUCTIONS 2.8.

Data in Register B can be moved en bloc to right or left within the register. This technique known as shifting is used to move numbers beyond the capacity of Register B so that they are either eliminated or circulated, that is, they re-enter Register B at the opposite end.

Generally a simple problem of extraction of a part of a word can be solved by the shifting method, using less storage though taking slightly more time than the masking method using Logical AND (see 2.6.1). This is because the latter case requires storage for a special masking constant.

However, a more complicated problem of this type, for example the extraction of positions 1 to 3 and 6 to 9 of a word is more economically solved by the mask constant method both in time and storage required. The method to be preferred will depend on the relative necessity of economizing on time or storage in the rest of the program.

The number of positions shifted is determined by the number in the two least-significant positions of the address part of the instruction which may take any value from 00 to 12. As far as the shift instruction itself is concerned, it is immaterial whether the contents of Register B are in decimal or sterling form. However, when a sterling amount is shifted to a different position in Register B it is necessary to set the Sterling Position Register (see 2.4.1) to correspond to the new tens of shillings position before applying any of the sterling arithmetic functions (i.e. functions 70 to 79) to the contents of Register B.

The shift functions do not send the contents of Register B through the Mill, and therefore cannot set any of the Mill Indicators.

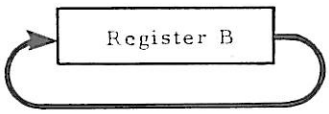
A relativizer should not be used with a shift instruction, otherwise the address part of the instruction which specifies the number of positions to be shifted will be mutilated.

Function 54 2.8.1

Effect Causes the contents of Register B to be circulated to the left.

Operation A number held in Register B is effectively shifted n places to the left of its original position; n is determined by the number (between 00 and 12) in the two least-significant positions of the address part of the instruction. Those digits which move out of the most-significant position re-enter Register B at the least-significant position.

It is actually only possible to perform a right shift in Register B. Function 54 is therefore effected by a right shift of that number of places which achieves the same result as the specified left shift.



Example Circulate the contents of Register B five places to the left.

Instruction

D	F	A	R
	54	0005	-

Before

Register B

0	1	2	3	4	5	6	7	8	9	10	11

After

Register B

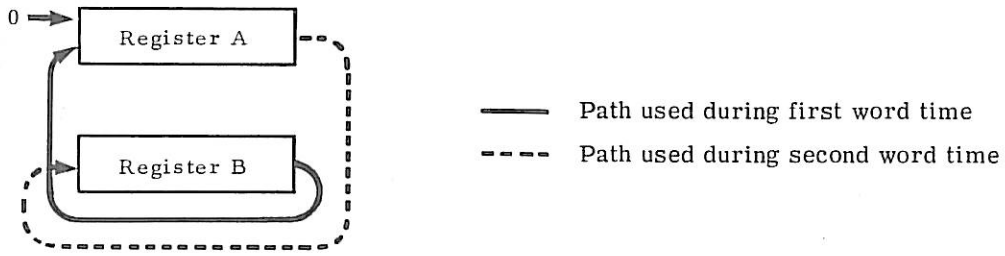
5	6	7	8	9	10	11	0	1	2	3	4

Function 55

2.8.2

Effect Causes the contents of Register B to be shifted to the left, entering zeros in the least -significant position.

Operation This function operates over two word times and unlike other shift functions utilizes Register A. This is necessary to achieve the specified left shift, which must in fact be performed as two right shifts. During the first word time Register A is zeroized and the digits which are to remain in the result are shifted from Register B into the most-significant end of Register A. During the second word time the contents of Register A are circulated into Register B to achieve the required result. Register A contains zeros on completion of the instruction.



Example Shift the contents of Register B six places to the left and fill the vacated positions with zeros.

Instruction

D	F	A	R
	55	0006	-

Before

Register A	0	0	0	0	0	0	0	1	2	3	4	5
Register B	1	2	4	3	6	7	9	10	5	8	11	4

After

Register A	0	0	0	0	0	0	0	0	0	0	0	0
Register B	9	10	5	8	11	4	0	0	0	0	0	0

Note As this function takes longer to execute than function 54, wherever possible function 54 should be used in preference to function 55.

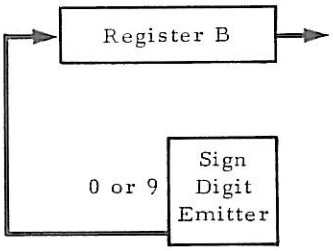
Function 56

2.8.3

Effect Causes the contents of Register B to be shifted to the right, entering either 0s or 9s in the most-significant position according to the value of the original digit in the most-significant position.

Operation A number held in Register B is shifted n places to the right of its original position: n is determined by the number (between 00 and 12) in the two least-significant positions of the address part of the instruction..

Those most-significant positions vacated are filled with either zeros if the original digit in the most-significant position was from 0 to 4, or nines if the original digit in the most-sigificant position was from 5 to 15. Thus the most-significant position is regarded as the sign position, 0 to 4 indicating a positive figure, 5 to 15 a negative figure.



Examples

(a) Shift the contents of Register B five places to the right and fill the vacated positions with eight zeros if 0 to 4 is in position 1 of Register B or nines if 5 to 15 is in position 1 of Register B.

Instruction

D	F	A	R
	56	0005	-

Register B

Before	0	1	2	3	4	5	6	7	8	9	10	11
After	0	0	0	0	0	0	1	2	3	4	5	6

or

Register B

Before	9	9	1	2	3	4	5	0	0	0	0	
After	9	9	9	9	9	9	9	1	2	3	4	5

(b) Shift the contents of Register B seven places to the right, propagating the sign.

Instruction

D	F	A	R
	56	0007	-

Register B

Before	3	4	5	6	7	8	3	4	5	6	7	8
After	0	0	0	0	0	0	0	3	4	5	6	7

or

Register B

Before	6	5	4	3	2	1	6	5	4	3	2	1
After	9	9	9	9	9	9	9	6	5	4	3	2

(c) Shift the contents of Register B twelve places to the right, propagating the sign.

Instruction

D	F	A	R
	56	0012	-

Register B

Before	10	0	0	0	0	0	0	0	4	3	2	1
After	9	9	9	9	9	9	9	9	9	9	9	9

or

Register B

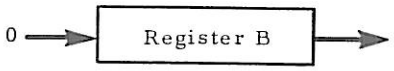
Before	4	6	0	0	0	0	1	2	3	4	5	6
After	0	0	0	0	0	0	0	0	0	0	0	0

Function 57

2.8.4

Effect Causes the contents of Register B to be shifted to the right, entering zeros in the most-significant position.

Operation A number held in Register B is shifted n places to the right of its original position; n is determined by the number (between 00 and 12) in the two least-significant positions of the address part of the instruction. The most-significant positions vacated are filled with zeros.



Examples

(a) Shift the contents of Register B five places to the right, do not propagate the sign.

Instruction

D	F	A	R
	57	0005	–

Register B

	0	1	2	3	4	5	6	7	8	9	10	11
Before												
After	0	0	0	0	0	0	1	2	3	4	5	6

(b) Shift the contents of Register B seven places to the right, do not propagate the sign.

Instruction

D	F	A	R
	57	0007	–

Register B

	6	5	4	3	2	1	6	5	4	3	2	1
Before												
After	0	0	0	0	0	0	0	6	5	4	3	2

Application of the Shift Functions

2.8.5

(a) Zeroizing Register B. Instruction 570012 is the most economical method.

(b) Extraction of part of a word: It is sometimes more economical in time or space or both to use shift functions in preference to Logical AND with a mask constant (described under 2.6.1). In the following examples, (i) the Shifting Method is more economical on storage, (ii) the Mask Constant Method is more economical on time.

Extract from word 25 of block 35 the quantity held in positions 9 to 12 and place it in word 26 of block 35.

(i) Shifting Method.

Instruction

D	F	A	R
	37	0025	35
	54	0008	–
	57	0008	–
	42	0026	35

	Before													After											
Word 25 Block 35	1	2	3	0	1	10	4	6	3	2	1	9		Word 25 Block 35	1	2	3	0	1	10	4	6	3	2	1	9
Word 26 Block 35	0	0	0	0	0	0	0	0	0	0	0	0		Word 26 Block 35	0	0	0	0	0	0	0	0	3	2	1	9
Register A	0	0	0	0	0	0	0	0	0	0	0	0		Register A	0	0	0	0	0	0	0	0	3	2	1	9
Register B	0	0	0	0	0	0	0	0	0	0	0	0		Register B	0	0	0	0	0	0	0	0	3	2	1	9

(ii) Masking Constant Method

Instruction

D	F	A	R
	37	0025	35
	35	0006	18
	42	0026	35

	Before														After													
Word 6 Block 18	0	0	0	0	0	0	0	0	15	15	15	15		Word 6 Block 18	0	0	0	0	0	0	0	0	15	15	15	15
Word 25 Block 35	1	2	3	0	1	10	4	6	3	2	1	9		Word 25 Block 35	1	2	3	0	1	10	4	6	3	2	1	9
Word 26 Block 35	0	0	0	0	0	0	0	0	0	0	0	0		Word 26 Block 35	0	0	0	0	0	0	0	0	3	2	1	9
Register A	0	0	0	0	0	0	0	0	0	0	0	0		Register A	0	0	0	0	0	0	0	0	3	2	1	9
Register B	0	0	0	0	0	0	0	0	0	0	0	0		Register B	0	0	0	0	0	0	0	0	3	2	1	9

(c) Shifting the positions of quantities in a word, e.g. for rounding purposes.

Example 1 Round up quantity held in word 16 block 23.

Instruction

D	F	A	R
	60	0016	23
	62	0010	18
	56	0003	-
	42	0016	23

	Before													After											
I.A.S. 10 Block 18	0	0	0	0	0	0	0	0	0	5	0	0		I.A.S. 10 Block 18	0	0	0	0	0	0	0	0	0	5	0	0
I.A.S. 16 Block 23	0	0	0	0	0	0	6	5	2	8	0	0		I.A.S. 16 Block 23	0	0	0	0	0	0	0	0	0	6	5	3
Register A	0	0	0	0	0	0	0	0	0	0	0	0		Register A	0	0	0	0	0	0	0	0	0	6	5	3
Register B	0	0	0	0	0	0	0	0	0	0	0	0		Register B	0	0	0	0	0	0	0	0	0	6	5	3

Example 2 Multiply word 129 of block 14 by word 28 block 13. Round up the product to the nearest penny and add the result to the quantity in word 39 block 15.

Instruction

I	D	F	A	R	NARRATIVE
X		37	0129	14	Transfer to Register B
		54	0003	—	Circulate left 3 positions
X+1		42	0129	14	Transfer to original locations
		37	0028	13	Transfer multiplier to Register B
X+2		22	0007	—	Set Sterling Position Register
		21	0003	—	Set Decimal Point Register
X+3		79	0129	14	Multiply
		72	0016	32	Round up
X+4		56	0003	—	Right Shift 3 positions
		22	0010	—	Reset Sterling Position Register
X+5		74	0039	15	Add to I.A.S.

Before

Word 28 Block 13	0	0	0,	0	0	0,	0	0	0.	9	9	9
Word 129 Block 14	0	0	0,	0	0	0,	0	0	9	1	9	11
Word 39 Block 15	0	0	0	0	0	0	0	0	0	0	0	0
Word 16 Block 32	0	0	0	0	0	0	0	0	0	5	0	0
Register B	0	0	0	0	0	0	0	0	0	0	0	0

After

Word 28 Block 13	0	0	0,	0	0	0,	0	0	0.	9	9	9
Word 129 Block 14	0	0	0	0	0	9	1	9	11	0	0	0
Word 39 Block 15	0	0	0	0	0	0	0	0	9	1	9	9
Word 16 Block 32	0	0	0	0	0	0	0	0	0	5	0	0
Register B	0	0	0	0	0	0	0	0	9	1	9	9

'DO NOTHING' AND 'STOP' FUNCTIONS 2.9

Function 00 2.9.1

Effect Causes the computer to do nothing and proceed to the next instruction in the normal manner.

Operation The computer obeys the instruction, therefore time is taken in operation but I.A.S. and registers concerning the programmer are in no way affected.

Notes This instruction may be used when it is necessary to modify the other instruction contained in the I.A.S. location. This may be more easily achieved by placing the instruction to be modified in the least-significant part of a location and having a 'do-nothing' instruction in the most -significant part e.g.

I	D	F	A	R
X	--	00	0012	00
		35	0009	12

The address will usually be zero, but as in the illustration, it may be any number between 0 and 3999.

Function 11 2.9.2

Effect Causes the computer to stop all further operations.

Operation The computer will stop all operations when this instruction is obeyed. If the address positions are used to contain a code number in the range 0 to 3999, after the stop the code number will be displayed on the console in CR3 with one added to it. Normal computer operation will be resumed when the Start button is operated.

Note The code numbers used in the address positions for this instruction can be used to indicate particular reasons for the stop e.g. when a sequence error occurs or an abnormal arithmetic condition is reached.

INDICATORS 2.10

Indicators are devices which may be in either one of two states, set or unset, and are used to direct the program into one of two branches according to the state of the indicator. This is achieved by writing an indicator test instruction. If the indicator is unset, the program proceeds to the normal next instruction. If the indicator is set, the program branches to the instruction pair in the I.A.S. location specified in the address part of the test instruction.

A summary of the characteristics of indicators numbers 00 to 29 is given in tabular form in the table overleaf. All other indicators (numbers 30 to 99) are dealt with in Part 3.

A complete summary of indicators is given in Part 6.

Representation of an Indicator Test Instruction 2.10.1

Some examples of indicator test instructions are given in Figure 6 (page 53). The first and second positions of the instruction contain the indicator number. The third position must contain a decimal digit whose binary representation contains a 1 in the 4 stream and a 0 in the 8 stream i.e. the number must be 4, 5, 6 or 7. This indicates to the computer that the instruction is a test instruction. The 4-bit is entered into the third position of the instruction during the Initial Orders program. On both the program sheet and the program card, the 4-bit occupies a separate column known as the designation column.

The third, fourth, fifth and sixth positions contain the I.A.S. address of the instruction to which the program will jump if the tested indicator is set. Thus in the case of a computer fitted with

more than 1,000 words of I.A.S., following completion of the Initial Orders program, position 3 of the instruction may contain both a 1 and a 4 in binary representation. For a machine with 4,000 words position 3 may also contain a 2-bit.

Indicator No.	Class of Indicator	Effect of Designation Indicator			Set by:-	Unset by:-
		4	8	9		
00	Unconditional Jump Indicator	Test	-	-	Permanently set	-
01	Mill Indicator	Test	-	-	Last no. through Mill zero.	Last no. through Mill not zero.
02	Mill Indicator	Test	-	-	Last no. through Mill > 0 (sign digit 0-4)	Last no. through Mill ≤ 0
03	Mill Indicator	Test	-	-	Last no. through Mill < 0 (sign digit 5-9)	Last no. through Mill ≥ 0
04	Error Indicator	Test & Unset	-	-	Overflow (sign digit 1-8 or sign digit 9 & all other digits zero)	Program when tested
06	Error Indicator	Test & Unset	-	-	I.A.S. Parity Check Error	Program when tested
07	Error Indicator	Test & Unset	-	-	Drum Parity Check Error	Program when tested
10 ' ' ' 19	Program Indicators	Test	Set	Unset	Instruction (Des.8)	Instruction (Des.9)
20 ' ' ' 29	Manual Indicators	Test	-	-	Manual Control on Console (Switch on)	Manual Control on Console (Switch off)

TABLE OF INDICATORS 00 TO 29

An indicator test instruction may form either the first or second half of an instruction word i.e. either positions 1 to 6 or 7 to 12.

Test instruction as written on program sheet	Form of instruction in computer following Initial Orders program	Significance of instruction
I: 21 D: 4 F: 03 A: 0037 R: –	034037	Test Indicator 03; if set, jump to I.A.S. 37 (absolute address); if unset proceed to instruction in second half of I.A.S.21.
I: 36 D: 4 F: 15 A: 0106 R: B	154456	Test Indicator 15; if set, jump to I.A.S. 106 of same block; if unset, proceed to instruction in second half of I.A.S. 36 of the same block.
Where block relativizer is set to 350		
I: 53 D: 4 F: 27 A: 0123 R: 72	275983	Test Indicator 27; if set, jump to I.A.S. 123 block 72 if unset, proceed to instruction in first half of I.A.S. 54 of the same block.
Relativizer for block 72 is set to 1860		

Figure 6: EXAMPLES OF INDICATOR TEST INSTRUCTIONS

Incorrect Test Instructions 2.10.2

In the case of incorrect test instructions being given to the computer, the sequence of operations is as follows:

If the specified indicator exists, but the I.A.S. address is outside the specifiable range, then; if the indicator is unset, the computer will proceed in the normal manner to the next instruction; if the indicator is set, the computer will stop with zeros in control registers 1 and 2, and the "I.A.S. Check Error on transfer to control register" visual indicator will be set.

If the specified indicator does not exist the computer will proceed in the normal manner to the next instruction.

Automatic Indicators 2.10.3

Automatic indicators are set, and in some instances unset, by conditions arising in the central processor or in the peripheral units. The conditions arising in the peripheral units are described in Part 3. The other automatic indicators are numbers 00 to 07.

Indicator 00 2.10.4

Purpose Testing indicator 00 causes an unconditional jump to a specified word of I.A.S.

Operation This indicator is permanently set. When tested, the next program instruction will not be taken from the next word of I.A.S., but due to the jump, will be taken from the word of I.A.S. specified in the address part of the indicator test instruction.

Example It is required to transfer the contents of I.A.S. location 12 into I.A.S. location 19 if the contents are positive to zero, but to leave the contents of I.A.S. 19 unaltered if the contents of I.A.S. 12 are negative, I.A.S. 12 and 19 being relative addresses in block 17.

I	D	F	A	R	NARRATIVE
125	-	60	0012	17	Contents of I.A.S. 12, block 17, into Register B via Mill
	4	03	0127	B	Test indicator 03, if set (set on < 0) jump to I.A.S. 127 of same block
126	-	42	0019	17	If indicator 03 is not set, write contents of Register B in I.A.S. 19, block 17,
	4	00	0127	B	then jump to I.A.S. 127 of same block

Notes An unconditional jump may be usefully employed when it is not possible to use the second half of an I.A.S. location for an instruction. The use of an unconditional jump under these circumstances is preferable to the use of function 00 (Do-nothing instruction) because the former takes less time.

Mill Indicators 2.10.5

The Mill Indicators are indicators 01, 02, 03 and 04. These indicators may be set by any function which involves the transfer of words through the Mill i.e. functions 35, 60 to 69, 70 to 79, except that indicator 04 (the overflow indicator) cannot be set by function 35. A word in this case is regarded as a number up to eleven digits together with a sign digit in the first position.

The Mill Indicators are set by the number which leaves the Mill after completion of the appropriate arithmetical operations within the Mill and not by the numbers entering the Mill.

Indicators 01, 02 and 03 remain set until a number with a different sign results in the Mill. Indicator 04 remains set until it is tested; testing automatically unsets it.

The setting of indicators 01, 02, 03 and 04 is shown diagrammatically in Figure 7.

Figure 8 (page 56) depicts a portion of program in which indicators 01, 02, 03, and 04 are tested. The example assumes that according to the nature of the sum of the contents of I.A.S. locations 20 and 31 (relative addresses in block 10) the following action is required:

(a) if an overflow occurs, the program must enter an error routine starting at I.A.S. location 80 of the same block of program;

(b) if the sum of the contents is zero, positive, or negative, a jump is to be made to I.A.S. location 61, 65 or 73 respectively, also in the same block.

The last instruction could equally well be an unconditional jump because if indicators 02 and 03 are both unset, indicator 01 must be set.

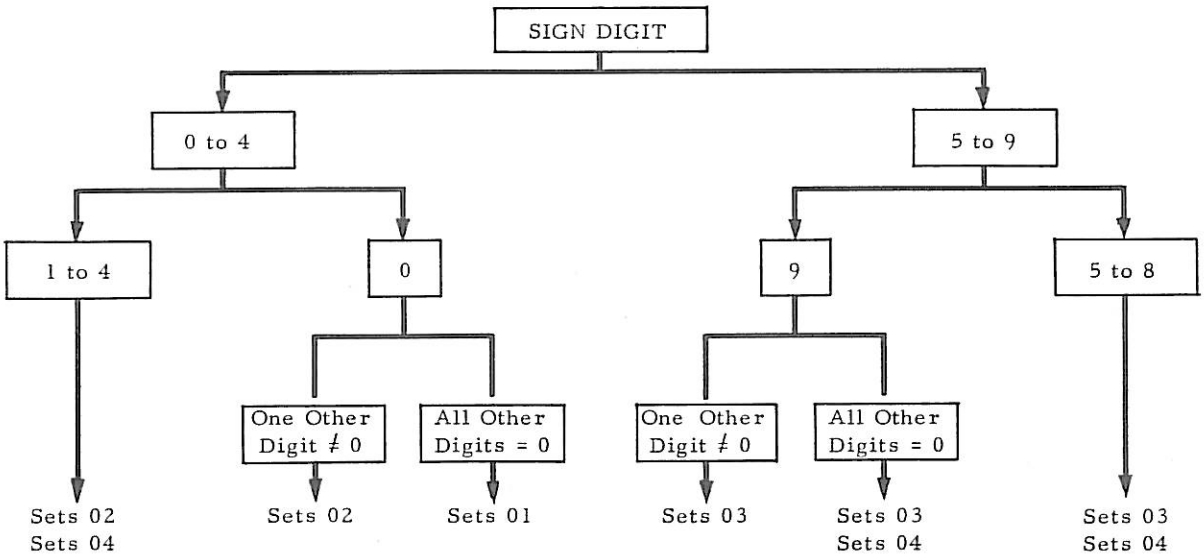


Figure 7: SETTING INDICATORS 01, 02, 03, AND 04

Indicator 01 — 2.10.6

Purpose May be used to test whether the last number resulting in the Mill was zero.

Operation Indicator 01 is set if the entire word is zero, i.e. if the sign digit and all other positions are zero (Figure 7). When indicator 01 is tested and found to be set, a jump is made to the I.A.S. word specified in the address part of the indicator test instruction.

The indicator remains set until a non-zero number results in the Mill.

Example See Figure 8.

Indicator 02 — 2.10.7

Purpose May be used to test whether the last number resulting in the Mill was positive.

Operation Indicator 02 is set if the word is positive, i.e. either if the sign digit is zero but at least one other digit is greater than zero, or, if the sign digit lies between 1 and 4 inclusive regardless of the state of the rest of the word (Figure 7). When indicator 02 is tested and is found to be set, a jump is made to the I.A.S. word specified in the address part of the indicator test instruction.

The indicator remains set until a negative or zero number results in the Mill.

Example See Figure 8.

Indicator 03 — 2.10.8

Purpose May be used to test whether the last number resulting in the Mill was negative.

Operation Indicator 03 is set if the word is negative, i.e. if the sign digit lies between 5 and 9 inclusive regardless of the state of the rest of the word (Figure 7). When indicator 03 is tested

and is found to be set, a jump will be made to the I.A.S. word specified in the address part of the indicator test instruction.

The indicator remains set until a positive or zero number results in the Mill.

Example See Figure 8.

Indicator 04 2.10.9

Purpose May be used to test whether an overflow has resulted in the Mill since the indicator was last tested.

Operation Indicator 04 (overflow indicator) is set if any of the arithmetical functions (functions 60 to 69, 70 to 79) causes either of the following two conditions to result in the Mill:

(a) a sign digit with a value 1 to 8 inclusive, or

(b) a 9 in the sign position with zeros in all other positions.

The overflow indicator remains set until it is tested, testing automatically unsets the indicator. Thus by testing the overflow indicator it is possible to detect whether or not an overflow has occurred into the sign position at any time since the indicator was last tested. If two numbers are added or subtracted and the overflow indicator is set, no information will have been lost. If two numbers are multiplied and the overflow indicator is set, information may have been lost.

When indicator 04 is tested and is found to be set, a jump will be made to the I.A.S. word specified in the address part of the indicator test instruction.

Example

I	D	F	A	R		NARRATIVE
57						
	4	04	0058	B		Test for overflow, to unset indicator
58		60	0020	10		Clear Add I.A.S. 20, block 10 to Register B.
		62	0031	10		Add I.A.S.31, block 10 to Register B
59	4	04	0080	B		If overflow ind set, jump to I.A.S.80, same block
	4	02	0065	B		If ind.02 set (No.>0), jump to I.A.S.65, same block
60	4	03	0073	B		If ind.03 set (No<0), jump to I.A.S.73, same block
	4	01	0061	B		If ind.01 set (No.=0), jump to I.A.S.61, same block

Figure 8: EXAMPLE INSTRUCTIONS TESTING INDICATORS 01, 02, 03, AND 04

Indicator 06 (*I.A.S. Check Error Indicator*) 2.10.10

Purpose Indicator 06 is used to test whether an I.A.S. transfer parity error has been detected since the indicator was last tested.

Operation An I.A.S. parity check is carried out when a word is transferred from I.A.S. to Register A.

Indicator 06 is automatically set if an I.A.S. parity error is detected on a transfer other than a transfer to the control registers.

Indicator 06 is unset when tested by program.

When the indicator is set the I.A.S. Error lamp on the console glows. If the Optional Stop switch is on, then the computer stops as soon as the parity error is detected. In this case the I.A.S. Error light is extinguished when the computer is restarted but indicator 06 remains set until tested by program.

Notes The computer always stops automatically if an I.A.S. parity error is detected on a transfer to the control registers. The I.A.S. parity checking system is described in more detail in 2.13.1.

The technique for restarting after I.A.S. parity failure is discussed in Part 4.

Indicator 07 (*Drum Check Error Indicator*) 2.10.11

Purpose Indicator 07 is used to test whether a drum transfer parity error has been detected since the indicator was last tested.

Operation A drum parity check is carried out when words are transferred from the drum.

Indicator 07 is automatically set if a drum parity error is detected.

Indicator 07 is unset when tested by program.

When the indicator is set the Drum Error lamp on the console glows. If the Optional Stop switch is on, then the computer stops as soon as the parity error is detected. In this case the Drum Error light is extinguished when the computer is restarted but indicator 07 remains set until tested by program.

Notes The drum parity checking system is described in more detail in 2.13.2.

The technique for restarting after drum parity failure is discussed in Part 4.

Programmed Indicators 2.10.12

The ten programmed indicators are indicators 10 to 19. These indicators are identical in operation and are set, unset and tested by means of instructions. The testing will not cause the indicators to be set or unset as a result of the test.

There are ten visual indicators (lamps) on the console, one for each indicator, which show the state of the indicators.

Indicators 10 to 19 2.10.13

Purpose These indicators are program controlled and are provided so that conditions can be designated and then later differentiated.

Operation Indicators 10 to 19 are set by an instruction containing an 8 in the designation position and are unset by a 9 in the designation position. Either of these two designations must be combined with the number of the indicator (10 to 19) in the first and second positions of the instruction. During Initial Orders, the 8 or 9 designation will be placed in the third position of the instruction.

The lamp associated with a particular indicator glows when that indicator is set. The light is extinguished when the indicator is unset.

Example It is required to set indicator 13 if the class of card designation held in position 12 of I.A.S. location 32 (a relative address in block 10) is a 5; (positions 0 to 11 are zeros); the indicator is to be unset later in the program.

Instruction

I	D	F	A	R	NARRATIVE
35	-	60	0032	10	Class of card designation transferred to Reg.B
	-	63	0006	15	Subtract '5' held in I.A.S. location 6, block 15
36	4	01	0037	B	Test if ind. 01 set, i.e. if card designation is 5, if so jump to I.A.S. location 37 of this block
	4	00	0038	B	If ind. 01 unset, jump straight to I.A.S. location 38 of this block
37	8	13	0000	-	If indicator 01 set, set indicator 13
	4	00	0038	B	Jump to I.A.S. location 38 of this block
65	--	---	------	---	----------
	9	13	0000	-	Unset indicator 13

An example of a test instruction for indicator 15 is shown in Figure 6.

Manual Indicators 20 to 29 2.10.14

There are ten manual indicators, i.e. 20 to 29, which are all identical in operation. These indicators are set and unset by means of manual controls (switches) on the console display panel, the operation of one of these switches causing its corresponding indicator to be set or unset. The indicator will remain set until it is unset by operation of the appropriate switch, i.e. its state is unaffected by test instructions.

The manual indicators are intended for use in applications where it is desired to make minor alterations to a program without recourse to reading in a different program.

Example In a payroll application a manual indicator could effect a change-over from weekly to monthly payroll, the indicator being used to modify the program in such a way that the appropriate monthly tax constants would be used instead of weekly constants during the P.A.Y.E. calculations.

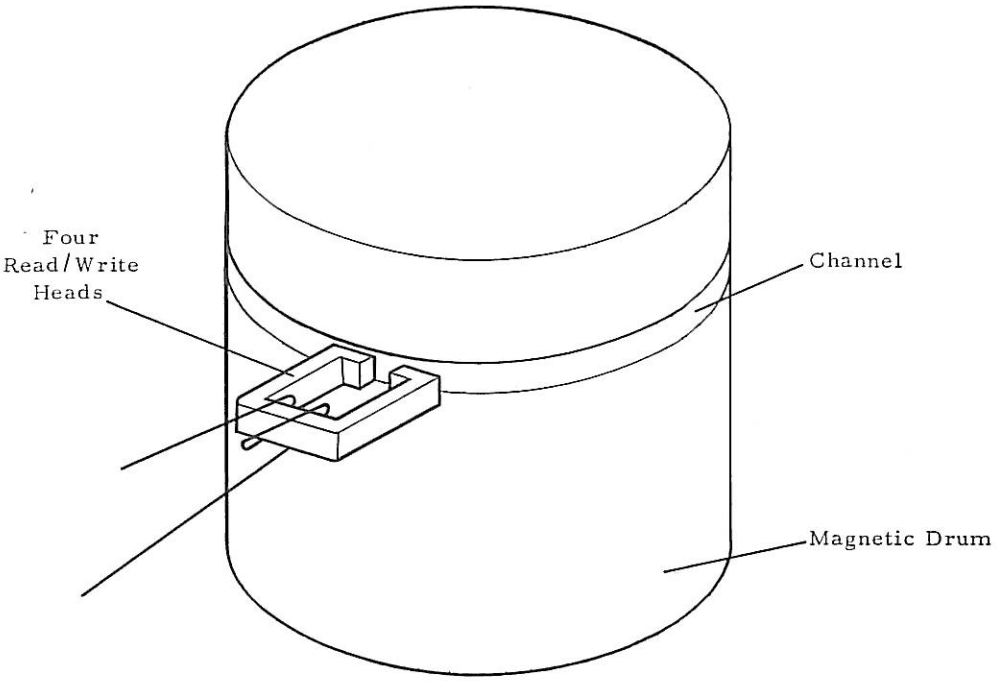


Figure 9: MAGNETIC DRUM

THE MAGNETIC DRUM 2.11

The magnetic drum is a backing store for the computer. Program instructions cannot be obeyed from the drum and the drum cannot be addressed directly other than to effect a transfer to or from I.A.S. Program instructions are provided for executing drum transfers.

The drum store is much larger than I.A.S. and it is therefore usual to hold the bulk of information on the drum and to transfer the information to I.A.S. as it is required. However, access to the drum is time consuming and the program should be arranged so that drum transfers are kept to a minimum.

A standard drum has 12,000 storage locations. A computer may be fitted with a quarter drum of 3,000 words, a half drum of 6,000 words, or one or more standard drums up to a maximum of eight drums.

Each 12,000 word drum is divided into two sections of 6,000 words and for timing purposes is regarded as being two separate drums. There are no such divisions on the 3,000 word drums or the 6,000 word drums.

The drum (Figure 9) is divided along its length into channels. A series of read/write heads are positioned along the length of the drum for transferring information to and from each channel. Each channel contains 200 words divided into groups of 10 words each called decades. The drum rotates at a constant speed, the locations moving in sequence past the read/write heads.

The decade is the smallest unit of transfer possible. In a drum transfer instruction, the drum address specified is a decade address. The largest unit which can be transferred by one instruction is 20 decades. It is possible to effect drum transfers either as a specified number of decades or as a complete channel. If the transfer is more than one decade, then it is possible for the decades to be taken from two separate channels provided that the decade numbering sequence is not broken.

The decade addresses of locations on the drum are shown in the table below.

Words	Drum Number	Decade Addresses
3,000	1	0000 to 0299
6,000	1	0000 to 0599
12,000	1	0000 to 1199
12,000	2	1200 to 2399
12,000	3	2400 to 3599
12,000	4	3600 to 4799
12,000	5	4800 to 5999
12,000	6	6000 to 7199
12,000	7	7200 to 8399
12,000	8	8400 to 9599

DECADE ADDRESSES OF 3,000, 6,000 AND 12,000 WORD DRUMS

If an address outside the range for any machine is specified in an instruction, the machine will stop due to failure to synchronize the clock source and the Slip Pulse lamp will glow.

The gap between decades is one word length and channel switching (i.e. when transferring decades from the end of one channel and the beginning of the next) occurs while this gap is passing the read/write heads.

Example Decades 59 and 60 (i.e. the last decade of one channel and the first decade of the next) are to be transferred. The transfer will be completed 21 word times after the start of the transfer of the first word of decade 59.

It is apparent that when transferring a specified number of decades the transfer time is the same irrespective of whether the transfer extends over one or two channels. This rule does not apply however when switching from the last decade of the first drum section to the first decade of the second drum section or when channel switching between drums.

Switching between drums or switching between drum sections takes up to just over one drum revolution.

Example 1 If decades 1199 and 1200 (i.e. the last decade of drum number 1 and the first decade of drum number 2) are transferred, then, on completion of the transfer of decade 1199, the transfer stops until the central processor is clocked by the second drum. This may take up to just over one drum revolution time. When the computer is clocked, decade 1200 is positioned at the read/write heads and the second transfer is effected.

Example 2 If decades 599 and 600 (i.e. the last decade of the first drum section and the first decade of the second drum section) are transferred, then, on completion of the transfer of decade 599, the transfer stops until the central processor is clocked by the second section. This may take up to just over one drum revolution time. When the computer is clocked, decade 600 is positioned at the read/write heads and the second transfer is effected.

In the two examples given, the maximum time taken for transfer of the two decades is a normal access time, plus a little over one drum revolution clocking time, plus the normal transfer time for two decades. The clock generator of the central processor is synchronized either by the drum or drum section currently in use, or by the drum or drum section to which reference was last made.

The arrangements referred to in the preceding paragraphs are dealt with completely automatically within the computer. It is only necessary to take note of these times under special circumstances which involve time-sharing of the central processor between drum references and input or output equipment.

In addition to the storage quoted for each drum, there are a further two channels (i.e. 400 locations) of reserved storage. On the first drum of each machine the reserved storage holds the Initial Orders program and engineer's Test Routines. The reserved storage is not accessible to the programmer for transfers from I.A.S. though transfers to I.A.S. from the reserved channels are permissible.

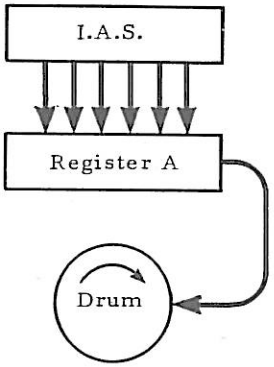
Drum Instructions 2.11.1

All transfer instructions from and to the drum are double-length and occupy all twelve digits of a storage location. There are two types of transfers; decade transfers and channel transfers. The functions are 80 to 87; functions 84 to 87 involve the use of the reserved store.

Function 80 2.11.2

Effect Transfers the contents of a specified number of decades of I.A.S. to the drum.

Operation The first location of I.A.S. from which the transfer is made is that specified in the instruction, successive transfers being from successively higher numbered locations of I.A.S. The information is taken from I.A.S. and is placed in Register A before being written on the drum. The first location of the drum to which the transfer is to be made is the first location of the drum decade specified in the instruction, successive locations being transferred to the second, third ... tenth locations of that decade followed by the first, second, third ....tenth locations of successively higher numbered decades. The contents of I.A.S. remain unaltered.



Example The contents of I.A.S. locations 249 to 388 (i.e. 14 decades starting from location 249) are to be transferred to decades 0805 to 0818 on the drum.

Instructions

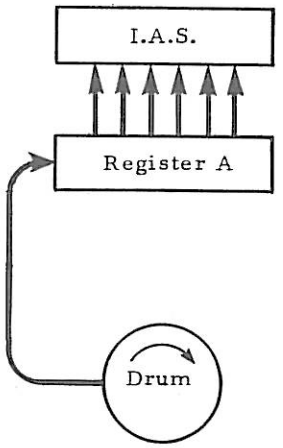
I	D	F	A	R
X	--	80	0249	-
		14	0805	-

Function 81

2.11.3

Effect Transfers the contents of a specified number of drum decades to I.A.S.

Operation The first location of the drum from which the transfer is made is that specified in the instruction, successive transfers being from the second, third.....tenth location of that decade followed by the first, second, third.....tenth locations of successively higher numbered decades. The information is taken from the drum and is placed in Register A before being written into I.A.S. The first location of I.A.S. to which the transfer is to be made is the first location of I.A.S. specified in the instruction, successive transfers being to successively higher numbered location of I.A.S. The contents of the drum remain unaltered.



Example The contents of drum decades 0805 to 0818 (i.e. 140 words starting from decade 0805) are to be transferred to I.A.S. locations 249 to 388.

Instruction

I	D	F	A	R
X		81	0249	-
		14	0805	-

Function 82 2.11.4

Effect Transfers the contents of 200 locations of I.A.S. to a specified channel on the drum.

Operation The first location of I.A.S. from which the transfer is made must be a multiple of 200, i.e. 0000, 0200, 0400..... or 3800. Successive words are transferred from successively higher numbered locations of I.A.S. The information is routed through Register A. The drum address specified in the instruction is the lowest numbered decade of the channel to which the transfer is to be made. The transfer begins at the start of the first decade to reach the read/write heads. This means that if I.A.S. locations 0 to 199 are to be transferred to drum decades 60 to 79 and drum decade 74 is the first decade to reach the read/write heads, then; I.A.S. word 140 is transferred to the first location of drum decade 74 followed in sequence by words 141, 142 .....199, 0, 1, 2..... to 139. The drum decades will be filled in the sequence 74, 75 ..... 79, 60, 61 .... to 73.

Example The contents of I.A.S. locations 200 to 399 are to be transferred to the 32nd channel on the drum (i.e. decades 0620 to 0639).

Instruction

I	D	F	A	R
X		82	0200	-
		20	0620	-

Function 83 2.11.5

Effect Transfers the contents of a specified channel of the drum to 200 locations of I.A.S.

Operation As with function 82, the I.A.S. address must be 0000, 0200, 0400, ...or 3800. Also, the drum address specified in the instruction is the lowest numbered decade of the channel from which the transfer is made. The transfer begins with the first location of the first decade to reach the read/write heads.

Example The contents of the 50th channel of the drum (i.e. decades 0980 to 0999) are to be transferred to locations 0800 to 0999 of I.A.S.

Instruction

I	D	F	A	R
X		83	0800	-
		20	0980	-

Note on Functions 82 and 83 If two channel transfers are succeeding instructions, then the access time for the second channel transfer will be zero provided that both the transfers are to or from the same drum or to or from the same drum section on 12,000 word drums.

Reserved Storage 2.11.6

As stated in 2.11, there are two additional channels of reserved storage on each drum from which the programmer may transfer information or instructions to I.A.S., but to which the programmer can only transfer from I.A.S. if an engineer's adjustment is made to the computer.

On the first drum, part of this storage is used to hold the Initial Orders program of instructions, which could be mutilated accidentally unless there was some restriction placed on its use by the programmer.

This storage is decade addressed as follows:-

(a) 0000 to 0019 plus 1200 (n - 1)
(b) 0100 to 0119 plus 1200 (n - 1)

} where n is the number of the drum and n = 1 for 3,000 word and 6,000 word drums.

It will be appreciated that these addresses are also decade addresses of the 'free' storage on each drum, the distinction being drawn by the computer according to the function codes used. Functions 84, 85, 86 and 87 are used when transferring information or instructions to or from the reserved channels.

If any of these functions is used in such a way that the drum address in the instruction is outside the specifiable address range for any of the reserved channels then the transfer will be made to or from the address specified in free storage on the drums.

Function 84 2.11.7

Effect Transfers the contents of a specified number of decades of I.A.S. to the reserved channels of the drum.

Operation Similar to function 80, but can only be accomplished with the assistance of the engineer and has restricted use.

Example The contents of locations 129 to 248 of I.A.S. are to be transferred to decades 0102 to 0113 of reserved storage on the first drum.

Instruction

I	D	F	A	R
X	--	84	0129	-
		12	0102	-

Function 85 2.11.8

Effect Transfers the contents of a specified number of decades of drum reserved storage to I.A.S.

Operation Similar to function 81.

Example The contents of decades 0004 to 0012 of reserved storage on the first drum are to be transferred to locations 0302 to 0391 of I.A.S.

Instruction

I	D	F	A	R
X	--	85	0302	---
		09	0004	

Function 86 2.11.9

Effect Transfers the contents of 200 locations of I.A.S. to one of the reserved channels of the drum.

Operation Similar to function 82 but can only be accomplished with the assistance of the engineer and has restricted use.

Example The contents of location 1200 to 1399 of I.A.S. are to be transferred to the second reserved channel of the third drum.

Instruction

I	D	F	A	R
X	--	86	1200	-
		20	2500	-

Function 87 2.11.10

Effect Transfers the contents of a reserved channel of the drum to 200 locations of I.A.S.

Operation Similar to function 83.

Example The contents of the first reserved channel of the fifth drum are to be transferred to locations 1800 to 1999 of I.A.S.

Instruction

I	D	F	A	R
X	--	87	1800	-
		20	4800	-

Relative Addressing of Drum Instructions 2.11.11

The drum instructions have been explained using absolute addresses. Relative addresses can be used, in which case the I.A.S. address and the drum decade address can refer to the same, or to different relativizers. The appropriate relativizer reference number must be placed in each half of the double-length instruction.

Example If R.R.N. 15 has an I.A.S. address of 800 and a drum location address of 9800, the instruction

I	D	F	A	R
X	--	83	0	15
		20	0	15

will be obeyed in the same way as

I	D	F	A	R
X	--	83	0800	-
		20	0980	-

and the contents of the 50th channel of the drum (decades 980 to 999) will be transferred to locations 800 to 999 of I.A.S.

Example If R.R.N. 17 has an I.A.S. address of 240 and a drum location address of 5000, and R.R.N. 18 has an I.A.S. address of 360 and a drum location address of 8000 the instruction

I	D	F	A	R
X		81	9	17
		14	5	18

will be obeyed in the same way as

I	D	F	A	R
X		81	0249	–
		14	0805	–

and the contents of drum decades 0805 to 0818 will be transferred to I.A.S. locations 249 to 388.

THE CONTROL REGISTERS 2.12

Instructions are normally obeyed in the sequential order in which they are stored in I.A.S. Instructions are transferred from I.A.S. by way of Register A to the control registers before being obeyed.

There are three control registers CR1, CR2 and CR3, each six digits in length and therefore capable of holding one single-length instruction apiece. The circuitry of the control registers is so arranged that after the second instruction in a word has been obeyed, control is transferred automatically to the first instruction of the next word. This sequence is broken only when a programmed jump occurs.

An instruction is actually obeyed when the instruction is in CR1. Figure 10 shows two transfer paths, one for an instruction other than a jump instruction and the other for a jump instruction. The sequence of operations for each instruction is as follows:

(a) An instruction other than an indicator test entering CR1 is obeyed. The contents of CR3 move into CR2, the previous contents of CR2 move into CR1 and the previous contents of CR1 have one added in the least-significant position before being moved into CR3.

(b) When an indicator test enters CR1 and is unsuccessful, then the transfer sequence is as follows:

(i) The contents of the first half of the I.A.S. location specified in the instruction are transferred to the second half of Register A; the first half of Register A will contain zeros in all six positions.

(ii) The contents of CR3 move into CR2, the previous contents of CR2 move into CR1 and the previous contents of CR1 have one added in the least-significant position before being moved into CR3.

(c) When an indicator test instruction enters CR1 and is successful, a change of control takes place and the transfer sequence is as follows:

(i) The contents of the I.A.S. location in the address part of the instruction are moved from I.A.S. into Register A.

(ii) The contents of Register A are moved into CR1 and CR2. The previous contents of CR2 and CR3 are moved into Register A and the previous contents of CR1 have one added in the least-significant position before being moved into CR3.

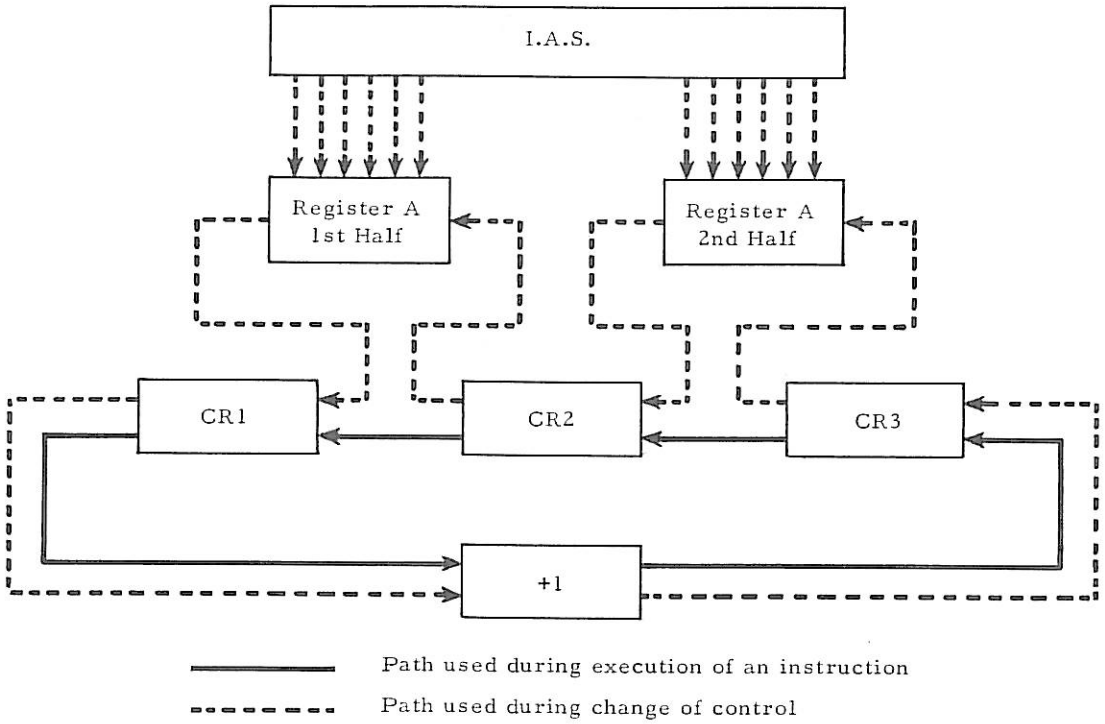


Figure 10: THE CONTROL REGISTERS

An indicator test instruction counts as a jump instruction only if it is successful.

When an indicator test instruction is transferred from CR1 to CR3 (with one added) the function digits are zeroized i.e. the instruction is converted into an unconditional jump instruction.

The following example shows the contents of the different registers for the given portion of program.

I	D	F	A	R	NARRATIVE
20	--	37	120	-	Transfer word 120 of I.A.S. to Register B
		68	121	-	Compare Register B with I.A.S. 121
21	4	02	22	-	Test indicator 02 If difference ≠ 0 jump to word 22
	8	11			otherwise set indicator 11
22					

Assume location 120 contains 000000000615 and that location 121 contains 000000000600 thus making indicator 02 unset.

	CR1	Register A First Half	CR2	Register A Second Half	CR3
Instruction in CR1 obeyed	370120	000000	680121	000615	004021
Contents of control registers shifted - Normal path.	680121	000000	004021	000615	370121
Instruction in CR1 obeyed	680121	999999	004021	999985	370121
Contents of control registers shifted - Normal path.	004021	999999	370121	999985	680122
Change of control. Contents of location 21 enter Register A.	004021	024022	370121	118000	680122
Contents of control registers shifted - Change of control path.	024022	370121	118000	680122	004022
Instruction in CR1 obeyed - unsuccessful indicator test.	024022	000000	118000	Word 22a	004022
Contents of control registers shifted - Normal path, function digits zeroized.	118000	000000	004022	Word 22a	004023
Instruction in CR1 obeyed	118000	000000	004022	Word 22a	004023
Contents of control registers shifted - Normal path.	004022	000000	004023	Word 22a	118001
Change of control. Contents of location 22 enter Register A.	004022	Word 22a	004023	Word 22b	118001
Contents of control registers shifted - change of control path.	Word 22a	004023	Word 22b	118001	004023

It is apparent that adding one to the contents of CR1 before storing in CR3 causes the words to be obeyed in sequential order unless there is a programmed jump to another part of the program.

For explanation purposes the contents of the registers have been shown as though the control register shift does not take place until the instruction in CR1 has been obeyed. In practice the shift takes place as soon as the instruction in CR1 has been initiated and while it is still in the process of being obeyed. For a control change the control register shift is also an integral part of the change of control process. In either case, therefore, the shifts will always be carried out before the computer stops. The quantities in heavy outline (e.g. 680121 000000 004021 000615 370121) are the numbers which will be displayed in the registers if the computer is stopped manually after each instruction.

It is apparent that:

(a) CR3 contains the instruction which has just been obeyed with one added to its address.

(b) CR1 contains the next instruction to be obeyed if the computer is restarted.

(c) When there has been a change of control between consecutive words of program, Register A contains the previous two instructions which have been obeyed, each with one added to its address (and indicator number zeroized if the instruction was an indicator test).

The following example shows the contents of the different registers for the same portion of program used in the preceding example with the assumption that:

Location 120 contains 000000000500 and Location 121 contains 000000000600 thus causing indicator 02 to be set.

	CR1	Register A First Half	CR2	Register A Second Half	CR3
Instruction in CR1 obeyed	370120	000000	680121	000500	004021
Contents of control registers shifted - Normal path.	680121	000000	004021	000500	370121
Instruction in CR1 obeyed	680121	000000	004021	000100	370121
Contents of control registers shifted - Normal path.	004021	000000	370121	000100	680122
Change of control. Contents of location 21 enter Register A.	004021	024022	370121	118000	680122
Contents of control registers shifted - Change of control path.	024022	370121	118000	680122	004022
Programmed change of control. Contents of location 22 enter Register A.	024022	Word 22a	118000	Word 22b	004022
Contents of control registers shifted - Change of control path. Function digits zeroized.	Word 22a	118000	Word 22b	004022	004023

Note that zeroizing the function digits on transfer to CR3 puts an 'unconditional jump to word 23' into CR3. This will ensure that when the instructions in word 22 have been obeyed, control will be transferred to word 23.

The contents of Register A, following a programmed change of control, provide a convenient method for a general purpose routine to restore control to the main program. The contents of Register A following a control change can also be useful for modification purposes. These techniques are discussed in Part 4.

When the first half of a double-length instruction enters CR1 this is detected by the machine and the instruction is obeyed from both CR1 and CR2. During the execution of double-length instructions, the instructions are modified in the control registers. When the instruction has been obeyed it is transferred to CR2 and CR3 but the addresses of both halves may have been modified (other than by having one added). At the change of control which immediately follows a double-length instruction, the contents of CR2 and CR3 are transferred to Register A. These contents can be useful for modification purposes and the relevant details are given in Part 4.

PARITY CHECKING SYSTEMS 2.13

Facilities are provided for checking errors which might occur when information is being transferred either between I.A.S. and the registers, or, between the drum and I.A.S. A word contains 48 data bits representing the 1, 2, 4 and 8 streams for each digit position. In addition to the data bits a word contains extra bits which are used for checking purposes only.

I.A.S. Parity Checking 2.13.1

A word in I.A.S. consists of 48 data bits and two additional bits termed check bits. The two check bits are generated every time a word enters Register A prior to being stored in I.A.S. One check bit is associated with the first half of the word and the other with the second half of the word. The first half of the word consists of the six most-significant digits and the second half of the word consists of the six least-significant digits.

In Register A the word is examined to ascertain whether the sum of the 1-bits in each half of the word is odd or even. If the sum of the 1-bits in the first half of the word is an odd number, a 1-bit is generated to make the total even. This ensures that the first half of every word entering I.A.S. from Register A is of even parity. If the sum of the 1-bits in the second half of the word is an even number, a 1-bit is generated to make the total odd. This ensures that the second half of every word entering I.A.S. from Register A is of odd parity.

Example The word 012345678901 is to be examined to ascertain the parity of each half of the word.

This is recorded in storage as:-

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

Number of 1-bits in first half (012345) = 7.

This is an odd number, therefore a 1-bit is generated to make the total even (8).

Number of 1-bits in second half (678901) = 9.

This is an odd number therefore no 1-bit is generated and the total remains odd (9).

Whenever a word enters Register A from I.A.S. it is checked for even parity in the first half and for odd parity in the second half. Then:

(a) If the parity check fails during a transfer to the control register the computer stops automatically and the I.A.S. to Control Register lamp glows.

(b) If the parity check fails during any other transfer from I.A.S., indicator 06 is set, the I.A.S. Error lamp glows and if the Optional Stop switch is set, the computer stops.

A parity check failure means that a transfer error has occurred either during the current transfer from I.A.S., or during the previous transfer to I.A.S.

When a transfer is made from I.A.S. to Register A the parity check is carried out and then the I.A.S. parity bits are regenerated and the word is re-stored in the location from which it originated. The programming implications of this are discussed in Part 4.

Drum Parity Checking 2.13.2

A word on the drum consists of 12 data digits and one extra digit for checking purposes. The check digit is generated immediately before a word is transferred to the drum. The check digit is generated as follows:

(a) The sum of the 1-bits in the data word is subtracted from 14.

(b) The number of times 14 can be subtracted from the sum of the 1-bits is multiplied by 16.

(c) The product obtained in the second operation is added to the result obtained in the first operation to produce the check digit.

Example

Word	No. of 1-bits	Step 1	Step 2	Step 3	Check Digit
001234432100	10	14 - 10 = +4	10 ÷ 14 = 0 + remainder, 0 × 16 = 0	+4 + 0 =	4
123456789012	17	14 - 17 = -3	17 ÷ 14 = 1 + remainder, 1 × 16 = 16	-3 + 16 =	13
777777777777	36	14 - 36 = -22	36 ÷ 14 = 2 + remainder, 2 × 16 = 32	-22 + 32 =	10
$\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}\,\overline{15}$	48	14 - 48 = -34	48 ÷ 14 = 3 + remainder, 3 × 16 = 48	-34 + 48 =	14

When a word is transferred from the drum, the check digit is calculated from the transferred data bits. This check digit is compared with the check digit stored with the word. If these are not the same then the parity check fails, indicator 07 is set and the Drum Error lamp glows. If the Optional Stop switch is set then the computer stops.

A parity check failure means that a transfer error has occurred either during the current transfer from the drum or when the information was previously transferred to the drum. The programming implications of this are discussed in Part 4.

Timings 2.14

The following table shows the speeds and timings associated with the central processor. Notes on using these to time a complete program are given in Part 4.

	Function	Time taken to execute function
Transfer Instructions	37	21 μs
	40	21 μs
	41	21 μs
	42	21 μs
	43	21 μs
	44	17 μs
	45	26 μs per word transferred
Decimal and Sterling Addition and Subtraction	60, 70	21 μs
	61, 71	21 μs
	62, 72	21 μs
	63, 73	21 μs
	64, 74	25 μs
	65, 75	25 μs
	66, 76	25 μs
	67, 77	25 μs
	68, 78	26 μs
	22	12 μs
	21	12 μs
Multiplication	69, 79	Maximum 44(6n + 2 + m) μs Minimum 44(n + 1 + m) μs where n = number of digits in the multiplier (excluding non-significant zeros) and m = P - n if positive and 0 otherwise, where P is the number entered in the Decimal Point Register.
Logical Functions	35	21 μs
	36	21 μs
Row-binarizing	30	21 μs
	31	21 μs
	32	21 μs
	33	21 μs
	34	21 μs
Shift Instructions	54	17 μs
	55	34 μs
	56	17 μs
	57	17 μs

	Time taken to execute function
Indicators	A program instruction to test, set or unset any indicator takes 12 μs. (See Note.)
Do Nothing	The 'Do Nothing' instruction takes 12 μs.
The Magnetic Drum	The magnetic drum revolves at 5240 r.p.m. i.e. one drum revolution takes 11.45 ms.
	For a decade transfer the access time varies between 0 and 1 drum revolution. Therefore average access time = 5.7 ms. Transfer time = .57 ms per decade. Hence for functions 80, 81, 84, 85, average time = (5.7 + .57n) ms. Maximum time = (11.4 + .57n) ms. Where n is the number of decades transferred. If there is a change of drum or a change to the other half of a 12,000 word drum (this may be before or during a transfer) a maximum of 12 ms must be added.
	For a channel transfer the access time varies between 0 and the time for the decade to pass the read/write heads. Transfer time = 1 drum revolution. Therefore for functions 82, 83, 86, 87, average time = 11.7 ms; maximum time = 12ms. If there is a change of drum or a change to the other half of a 12,000 word drum before the transfer a maximum of 12 ms must be added.
The Control Registers	The change of control between two consecutive words of program takes 12 μs. (See Note.)

Note An indicator test instruction takes 12 microseconds whether or not it is successful. If it is successful then this time represents the change of control, hence, when jumping to a word of program, this takes the place of the normal 12 microseconds control change from the previous word. It follows that a jump instruction effectively takes no time at all and for this reason, if the second half of a word has to be wasted, it is preferable to use an unconditional jump rather than a 'Do Nothing'.

Example

C	I	D	F	A	R		NARRATIVE
	125	-	60	0012	17		Contents of I.A.S. 12, block 17, into Register B via Mill
		4	03	0127	B		Test indicator 03; if set (set on < 0) jump to word 127 of block
	126	-	42	0019	17		If ind. 03 not set, write contents of Reg. B in I.A.S. 19 of block 17 then jump to word 127 of block
		4	00	0127	B		

Here the unconditional jump instruction brings about an immediate control change and the contents of location 127 are transferred to the control registers. A 'Do Nothing' would have taken 12 microseconds and would then have been followed by a control change.