UNIVERSITY OF ILLINOIS

DIGITAL COMPUTER

LIBRARY ROUTINE A 1 - 63

TITLE Floating Decimal Arithmetic Routine

TYPE Interpretive routine with 18 interpretive orders, entered as a closed routine, left by an 8J interpretive order.

NUMBER OF WORDS 168

PURPOSE This routine manipulates numbers in the floating decimal form, that is numbers which are represented as $A \times 10^p$. It is of the interpretive type. This means that it selects parameters called interpretive orders which are written by the user one at a time and performs a calculation corresponding to each interpretive order. Interpretive orders carry out normal arithmetic operations such as addition and multiplication and some red tape operations such as counting and address changing.

In general, one will use this routine to do computations which do not require the full speed of the computer but which are too time consuming to be done by hand. It is especially effective for problems with scaling difficulties. In a sense one may think of the floating decimal routine as converting the Illiac to a medium speed floating decimal computer having a very convenient order code.

ACCURACY About 9 decimals

TEMPORARY STORAGE 0, 1, 2

PRESET PARAMETERS S3 is used to specify two locations of non-temporary storage, S3 and 1S3, which are used for the floating decimal accumulator.

METHOD OF USE The floating decimal routine is entered as a standard subroutine. Following the entry, i.e. after the transfer of control to the subroutine, one begins writing interpretive orders. These orders each occupy one half word and consist of a pair of function digits followed by a single address. They therefore have the same form as standard machine orders and may be read by the Decimal Order Input with full use of the conventional terminating symbols.

The first of the two function digits of an interpretive order describes the group characteristics of the order and may take values 0, 1,..., 8. Normal arithmetic interpretive orders have this digit equal to 8. The second of the two function digits describes the type of interpretive order.

INTERPRETIVE ORDER LIST WITH FIRST FUNCTION DIGIT b = 8

Let F be the floating decimal number in the floating accumulator and let F(n) be the floating decimal number in location n.

80 N Replace F by F - F(n).

81 n Replace F by -F(n).

82 n Transfer control to the right hand interpretive order in n if F $\geq$ 0.

83 n Transfer control to the left hand interpretive order in n if F $\geq$ 0.

84 n Replace F by F + F(n).

85 n Replace F by F(n).

86 n Replace F by F/F(n).

87 n Replace F by F x F(n).

88 0 Replace F by one number read from the input tape punched as sign, any number of decimal digits, sign, and two decimal digits to represent the exponent. For example, .8971 x 10^{10} would be punched as + 8971 + 10.

89 n Punch or print F as a sign, n decimal digits, sign, two decimal digits to represent the exponent and two spaces. This print out may be re-read by this routine. After F has been punched or printed it may not remain in the floating accumulator unmodified. n can take values 2 to 9.

8K n Replace F by n if 0 $\leq$ n < 200

8S n Replace F(n) by F.

8N n Replace F by |F| - |F(n)|.

8J n Transfer control to the ordinary Illiac order on the left hand side of n. This used to escape from the floating decimal subroutine.

8F n Give a carriage return and line feed and start a new block of printing having n columns. This order is only obeyed <u>once</u> for a particular block of printing. At this time a counter is set up which will cause a carriage return and line feed to occur automatically from then on after every set of n numbers that is printed.

INTERPRETIVE ORDERS WITH $b \neq 8$. If the first function digit of an interpretive order is 0, 1,...,7 it will refer to one of a set of control registers or b-registers in the floating decimal routine which are similarly numbered. These registers are used for counting the number of passages through loops or cycles and for advancing addresses on successive passages. For this purpose a particular b-registers which may be used in a particular cycle contains two counting indice g_b and c_b. These are both integers in the range 0 to 1023. The index c_b is used for counting purposes to determine the number of passages through a loop. The index g_b is used for advancing the addresses of interpretive arithmetic orders. Although the interpretive order with first function digit b is not actually altered in the memory it is obeyed as if g_b were added to its address. The index g_b is increased by one upon each passage through the cycle. The multiplicity of b-registers allows one to program many loops within loops.

ORDER LIST WITH $n \neq 8$

b0 n	Replace F by F - $F(n+g_b)$
b1 n	Replace F by $-F(n+g_b)$
b2 n b3 n	Replace g_b, c_b by $g_b + 1$, $c_b + 1$. Then transfer control to the right hand (if b2 n) or left hand interpretive (if b3 n) order in n if c_b+1 is negative. This transfer is used at the end of a loop.
b4 n	Replace F by F + $F(n + g_b)$
b5 n	Replace F by $F(n + g_b)$
b6 n	Replace F by $F/F(n+g_b)$
b7 n	Replace F by F x $F(n+g_b)$
bK n	Replace g_b, c_b by 0, -n. This interpretive order is used for preparing to cycle around a loop n times.
bS n	Replace $F(n + g_b)$ by F
bN n	Replace F by $\|F\| - \|F(n+g_b)\|$
bL n	Replace g_b, c_b by $g_b + n$, c_b. This interpretive order is used when one wishes to step addresses by some increment other than +1 in a loop. If one places bL 1022 in a loop the effect will be to decrease addresses by one on each passage. bL 1 will increase them by 2 etc.
8L n	Replace g_b c_b by n, c_b, where b is the last b-register referred to by some previous interpretive order.

DURATION OF INDIVIDUAL INTERPRETIVE ORDERS

8N, 80, 84	5 milliseconds + m x (3/2). Where m is the number of shifts required to convert A, p back to standard form.
81, 85	2 milliseconds
82, 83	3 milliseconds
87	5 milliseconds
86	6 millisecons
8K	3 milliseconds
8S	3 milliseconds
8F	3 milliseconds
8L	2 milliseconds
8J	3 milliseconds

When an interpretive order is preceeded by $b \neq 8$, add one millisecond to the above times.

When one wishes to repeat a cycle of interpretive orders n times the interpretive order bK n may be written before entering the loop to set the counter c_b to -n. The interpretive orders in the loop will be obeyed n times if the loop is terminated with a b2 or b3 interpretive order to transfer control to the beginning of the loop. This transfer of control interpretive order will be obeyed n-1 times and disobeyed the nth time.

The following examples illustrate the construction of such loops.

EXAMPLE 1 Calculate x^{10} where x = F(4)

0	8K 1F	Set F = 1
	2K 10F	Set to cycle 10 times
1	87 4F	c_2 = -10
	23 1L	Transfer control to 1 relative 9 times.

EXAMPLE 2 Replace F(100 + i) by F(200 + i) + F(300 + i), $i = 0,1,\ldots,9$

6	0K 10F	$g_0 = 0$, $c_0 = -10$
	05 200F	$F = F(200 + g_0)$
7	04 300F	$F = F(200 + g_0) + F(300 + g_0)$
	0S 100F	$F(100 + g_0) = F(200 + g_0) + F(300 + g_0)$
8	02 6L	increase g_0 by 1, transfer control 9 times to 6 relative.

EXAMPLE 3 Evaluate $\sum_{i=0}^{19} a_i \, x^{19-i}$, where $a_i = F(100 + i)$, $x = F(10)$. The operations form $(((a_0 x + a_1)\, x + a_2)\, x + a_3)\, x + \cdots$

2	8K F	Clear F
	0K 20F	Prepare to cycle 20 times
3	87 10F	multiply by x
	04 100F	add $F(100 + g_0)$
4	03 3L	

EXAMPLE 4 Print 10 numbers each with 5 decimal figures in a block of 3 columns from 100 to 109.

22	8F 3F	Start block with 3 columns
	0K 10F	
23	05 100F	Print 10 numbers
	89 5F	
24	03 23L	

EXAMPLE 5 Place the number F(200 + i) in 300 + 2i for $i = 0,1,\ldots,24$.

5	0K 25F	$g_0 = g_1 = 0$
	1K F	$c_0 = -25$
6	05 200F	
	1S 300 F	
7	1L 2F	advance g_1 by 2
	03 6L	

EXAMPLE 6 Place the number F(200 + i) in 300 - i for i = 0,1,...,24

5	0K 25F	
	1K F	
6	05 200F	
	1S 300F	
7	1L 1023F	reduce g_1 by 1
	03 6L	increase g_0 and c_0 by 1

Use of Auxiliary Routines. It is often convenient to be able to leave the floating decimal routine so as to modify interpretive orders or to perform calculations which may be done more effectively outside of floating point. To leave the floating decimal routine one uses an 8J n order. (All standard floating decimal auxiliaries are entered in this way.) To return to floating decimal one should transfer control to the left hand side of word 29 of the floating decimal routine. The interpretive order following the 8J n order which was last obeyed will then be obeyed and so on. In this way it is not necessary to plant a link in auxiliary subroutines. One may, in fact, think of the 8J n order as a subroutine order. In case any changes are made in the floating decimal accumulator while outside the floating decimal routine, control should be returned to the left hand side of word 19 rather than 29 so that this number may be standardized before reentry.

Handling of Numbers Each number is represented in the form $A \times 10^p$ where $1 > |A| \geq 1/10$, and $64 > p \geq -64$. In a single register of the memory the number A is placed in the 33 most significant binary digits ($a_0, a_1, \ldots, a_{32}$) in the same way as an ordinary fraction is placed in the entire register. An accuracy of between 8 and 9 decimal digits is therefore achieved. The exponent p is stored as the integer p + 64 in the 7 least significant digits of the same register. For convenience the floating decimal accumulator uses two registers S3 and 1S3 for holding the number $A \times 10^p$. The fraction A/2 is in S3 and the integer p + 64 is in 1S3.

The only exeption to the above rules is the number zero which cannot, of course, be represented as $A \times 10^p$ with $|A| \geq 10$. For this reason zero is handled in a special way. It is represented as a number with A = 0 and p = -64.

This representation happens to correspond exactly with the ordinary machine representation of zero.

After each arithmetic interpretive order is obeyed the number in the floating decimal accumulator is standardized, i.e. the number of S3 representing A/2 is adjusted so $1 > |A| \geq 1/10$ and p is changed accordingly. To accomplish this control is transferred to word 19 in the floating decimal routine after each arithmetic order.

If an interpretive store order is attempted when F has an exponent greater than 63 the machine will stop on the order 34 p at location p, where p is word 72 of the routine.

Important Words in the Routine. Word 2 in the floating decimal routine determines the location of the current interpretive order. When obeying the left hand interpretive order in location n this word is 50 nF S5 20F and when obeying the right hand interpretive order in location n it is L5 nF 00 20F. Other words of interest are the b-registers which start at word 158 (for g_0 and c_0) and go to 165 (g_7 and c_7). These register hold g_b and c_b in the form

$$80\ g_bF\ 00\ (2048 + c_b)\ F.$$

Warning When the same number is continually added to a sum, such as when an argument is being increased, the error can be quite large, because it is additive over a decade. For example, if we increase 10 to 100 by units we can get a maximum error of 90×2^{-33} because the errors all have the same sign. If we increase 10^3 to 10^4 we can have a maximum error of $9{,}000 \times 2^{-33}$. This can easily be prevented by writing an auxiliary subroutine to stabilize the fractional part of F, i.e. to replace it by the nearest multiple of say 10^{-7}. Such a subroutine could be as follows

m	50 S3	
	7J m+3F	A contains nearest multiple of 10^{-7}
m + 1	50 1F	Location 1 contains zero
	66 m+3F	converts to fraction again
m + 2	S5 F	
	22 s+18F	return to 18th order of Routine A-1
m + 3	00 F	
	00 20,000,000F	$2 \times 10^7 \times 2^{-39}$

RT: 8/3/60

BY David J Wheeler	DATE November 18, 1953
	APPROVED BY J. P. Nash

LOCATION	ORDER		NOTES
NOTE:			Assume N(S3) = A/2 where F = A x 10^p N(1S3) = p + 64 F(m) = B x 10^q new value of quantity is indicated by a dash.
	00 K(A1)		
0	00 59F		
	L0 166L		Special orders for entry
1	L0 124L		Plant either 50 nF S5 20F or L5 nF 0020F
	40 2L		in 2L
2	50 (n)F		
	S5 20F		Select order to be obeyed
3	40 2F		
	32 150L		Adjust address if necessary
4	46 6L		Insert address in 6L
	10 12F		
5	L4 125L		Form switch order
	46 9L		
6	L5 (m)F		
	10 1F		Select argument
7	40 F		Place fraction B/2 in F
	10 6F		
8	01 7F		
	40 1F		and exponent q in 1F
9	26 F		
	40 F		
10	L5 1S3		
	L0 1F		Test exponent difference p - q
11	50 F		
	32 15L		
12	40 2F		
	S5 F		
13	50 S3		Interchange A and B
	40 S3		
14	L5 1F		and place q in 1S3
	40 1S3		

LOCATION	ORDER	NOTES	PAGE 2
15	L1 2F		
	L4 125L		
16	42 17L	Insert exponent difference in 17L	
	L0 126L		
17	36 29L	Skip addition if exponent difference	
	7J (31 + $\lvert p-q \rvert$)F	$\lvert p - q \rvert > 10$	
18	L4 S3	A + B 10^{-p+q} or B + A 10^{-q+p}	
	40 S3		
19	LL S3	is A' $\geq$ 1/2	
	36 27L		
20	50 S3		
	7J 32L	Then multiply A' by 1/10	
21	40 S3		Standardize A
	L5 43L		and P so that
22	L4 1S3	and increase p' by 1	$1 > \lvert A \rvert \geq 1/10$
	40 1S3		
23	26 27L		
	50 S3		
24	75 127L	A' x 10	
	00 4F		
25	40 S3		
	L1 43L		
26	L4 1S3	p - 1	
	40 1S3		
27	L3 S3	Test if $\lvert A' \rvert < 1/10$	
	36 68L	or equal to zero	
28	L4 42L		
	32 23L		
29	L5 2L	if N(2L) is positive (50 nF S5 20F)	
	36 1L	select right hand order in n.	
30	L4 128L	otherwise select left hand order in n+1	
	22 1L		
31	7L 4095F		
	LL 4095F	10^0 - almost	

LOCATION	ORDER		NOTES
32	00F 00 1000		
	0000 0000 J		10^{-1}
33	00F 00 100		
	0000 0003 J		10^{-2}
34	00F 00 10		
	0000 0000 J		10^{-3}
35	00F 00 1		
	0000 0000 J		10^{-4}
36	00F 00		
	1000 0000 J		10^{-5}
37	00F 00		
	100 0000 J		10^{-6}
38	00F 00		
	10 0000 J		10^{-7}
39	00F 00		
	1 0000 J		10^{-8}
40	00F 00		
	1000 J		10^{-9}
41	00F		
	00 64F		
42	00F 00 499		1/20
	9999 9998 J		
43	00 F		Table of switch orders to transfer control to the appropriate sets of orders.
	00 1F		
44	L1 F		b0
	22 9L		
45	L1 F		b1
	26 67L		
46	L5 131L		b2
	26 133L		
47	L5 131L		b3
	22 133L		
48	J0 1021F		b4 also used as a binary switch'
	26 10L		

LOCATION	ORDER	NOTES	PAGE 4
49	L5 F	b5	
	26 67L		
50	L1 1F	b6	
	26 63L		
51	L0 41L	b7	
	26 60L		
52	49 1F	b8	
	22 77L		
53	L5 6L	b9	
	22 97L		
54	41 1F	bK	
	26 138L		
55	L5 6L	bS	
	26 69L		
56	L7 S3	bN	
	26 76L		
57	L5 2F	bJ	
	22 5L		
58	92 131F	bF	
	26 95L		
59	L5 6L	bL	
	22 149L		
60	L4 1S3	p' = p + q	Multiplication
	40 1S3		
61	50 F		
	7J S3	A' = A B	
62	00 1F		
	22 18L		
63	L4 130L	p' = p - q + 1	Division
	L4 1S3		
64	40 1S3		
	50 42L		
65	75 S3		
	66 F	A' = A x (1/10) + B	

LOCATION	ORDER	NOTES
66	S5 31L	
	22 18L	
67	40 S3	p' = q (Clear add, Subtract)
	L5 1F	
68	40 1S3	A' = -B (Clear add, Subtract)
	26 29L	
69	46 75L	
	L5 1S3	
70	32 71L	Test if p + 64 is negative (Store order)
	41 S3	
71	41 1S3	
	L0 167L	
72	34 72L	Stop machine if p ≥ 64. (Store order)
	L5 S3	
73	10 6F	(Store order)
	50 1F	Form packed A, p (Store order)
74	00 7F	
	L4 1S3	and place in m. (Store order)
75	40 (n)F	
	26 29L	
76	40 S3	A' = |A| (Modulus order)
	L3 F	B' = -|B| Modulus order
77	22 9L	
	81 4F	
78	00 39F	
	36 80L	Input sign place ± 1/2 in 1F
79	L1 1F	
	40 1F	
80	27 84L	
	50 1F	
81	7J 32L	
	40 1F	
82	50 1F	
	75 2F	

LOCATION	ORDER	NOTES
83	00 39F	Input conversion cycle
	L4 S3	
84	40 S3	
	81 4F	
85	40 2F	
	L0 129L	
86	36 87L	
	22 80L	
87	81 4F	
	40 F	
88	81 4F	
	50 F	
89	74 129L	
	L1 2F	Read
90	L4 129L	
	36 92L	Exponent
91	S1 31L	
	22 92L	
92	S5 F	
	L4 41L	
93	40 1S3	
	26 19L	Control to 19L to standardize A and p.
94	L1 ( )F	
	40 4F	Column Count
95	L5 2F	
	10 20F	Arrange new column count and
96	42 94L	give a carriage return and
	47 94L	line feed.
97	26 29L	
	L4 91L	Set address of round-off order
98	46 105L	
	L5 94L	
99	L0 79L	
	36 101L	Give a carriage return

LOCATION	ORDER	NOTES	PAGE 7
100	92 131F	When necessary	Print order
.	22 122L		
101	46 94L		
	L5 S3		
102	40 F		
	36 104L		
103	92 708F		
	22 104L		
104	92 644F	Print sign and round-off	
	50 101L		
105	7J ( )F		
	L6 F		
106	L6 F		
	32 110L	Test if round-off + |A| > 1	
107	L5 43L		
	L4 1S3		
108	40 1S3	If so replace A by 1/20	
	L5 42L		
109	40 S3	and p by p + 1	
	40 F		
110	22 104L		
	10 39F		
111	75 129L		
	00 36F	Print cycle	
112	82 4F		
	10 40F		
113	L5 6L		
	L0 49L		
114	46 6L		
	L0 62L		
115	36 111L		
	92 961F	Space	
116	L1 48L	Binary switch	
	40 48L		

LOCATION	ORDER	NOTES
117	32 118L	Did we print fraction or exponent?
	92 961F	Space
118	26 29L	
	46 6L	
119	L5 1S3	Arrange to print exponent
	L0 41L	
120	40 F	
	50 F	
121	75 33L	
	00 38F	
122	26 102L	
	92 513F	
123	00 24F	
	26 101L	
124	5S F	L5 (n)F 00 20F - 50 (n)F S5 20F
	S5 F	
125	00 172L	Base of switch order = 44 + 128
	00 31L	Base of 10^{-p} table order
126	00 172L	End test for 10^{-p} table order
	00 42L	
127	50 F	10/16
	00 F	
128	5S 1F	
	S5 F	50 (n+1)F S5 20F = L5 (n)F 00 20F
129	00 F	
	00 10F	10×2^{-39}
130	00 F	
	00 65F	65×2^{-39}
131	50 F	Base for 2,3 orders
	S5 20F	
132	00 1F	Double unit for counting
	00 1F	
133	L0 124L	
	40 1F	Prepare for transfer

LOCATION	ORDER	NOTES
134	L5 2F	
	46 1F	
135	32 143L	b2 b3 orders
	L5 S3	Transfer control if positive
136	36 137L	
	26 29L	Select next order if negative
137	L5 1F	
	22 1L	
138	L5 2F	Is b = 8 or ≠ 8?
	36 146L	
139	46 1F	
	50 1F	Place m/200 in S3
140	75 33L	8K
	00 18F	
141	40 S3	
	L5 130L	and 66 x 2^{-39} in 1S3
142	L4 43L	
	40 1S3	
143	26 19L	
	L5 (157 + b)L by 152	b ≠ 8
144	L4 132L	Increase c_b and g_b by 1 — b2 b3 orders
	40 (157+b)L	
145	00 28F	Then test sign
	26 136L	
146	10 20F	
	42 1F	bK b ≠ 8
147	L1 1F	make g_b = 0
	50 158L	c_b = 2048-n
148	L4 157L	
	40 (157+b)L	
149	26 29L	
	46 (157+b)L	From OL switch
150	26 29L	
	50 2F	

LOCATION	ORDER		NOTES Page 10
151	01 3F		
	L4 147L		
152	42 143L		Adjust all addresses
	42 155L		
153	42 148L		dependent on b
	42 149L		
154	42 144L		
	J0 166L		
155	11 3F		Modify <u>order</u> add $g_b + 1$
	L5 157L		
156	S4 F		
	26 4L		
157	80 F		Constants
	00 2048F		
158	80 F		
	00 F		
159	80 F		
	00 F		
160	80 F		
	00 F		
161	80 F		
	00 F		
162	80 F		
	00 F		
163	80 F		
	00 F		
164	80 F		
	00 F		
165	80 F		
	00 F		
166	LL 4094F		
	4K 4076F		
167	00 F		
	00 128F		

A1