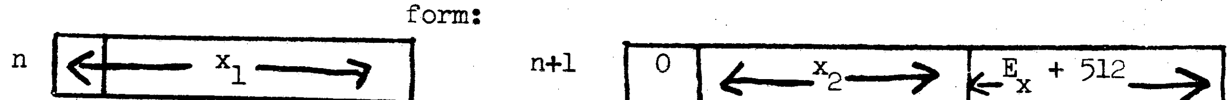
UNIVERSITY OF ILLINOIS

DIGITAL COMPUTER

ILLINOIS CODE 87 - A4

TITLE	1.7 Precision Floating Binary Arithmetic with Floating Decimal Conversion (D.O.I. or SADOI)
TYPE	Interpretive routine, entered like a closed subroutine
NO. OF WORDS	280
TEMPORARY STORAGE	Location specified by preset parameter S3 12 locations, S3 to 11S3
DURATION	See Order Code
ACCURACY	Rounded to 68 binary places, equivalent to about 20 significant decimals. Print out is rounded.
PARAMETERS	S3 ; during input location 3 must contain $t \times 2^{-39}$ where t is the location of the first word of temporary storage.
DESCRIPTION	This routine is designed to do computations where accuracies of twelve to twenty significant decimals are required and where scaling is enough of a problem to justify the use of a floating point routine. This interpretive routine arranges arithmetic operations upon numbers represented in the form $(x)\, 2^{E_x}$ where $-512 \leq E_x < 512$ $\lvert x \rvert < 1$. However, numbers are input and output in the form $z \times 10^p$ where $-1 \leq z < 1$ and $\lvert p \rvert \leq 153$. The modulus of the numbers x has 68 binary digits. When a number $X = x\, 2^{E_x}$ is stored in two successive memory locations n, n+1, (or for simplicity say stored at n), it takes on the following standard form:



$X = (x_1 + 2^{-39} x_2)\, 2^{E_x}$ where $x_2 \geq 0$, $-512 \leq E_x < 512$ and $1 > \lvert x \rvert = \lvert x_1 + 2^{-39} x_2 \rvert > 2^{-1} - 2^{-37}$. The latter inequality is a consequence of the way certain scaling is done. Note that the exponent E is situated in the right hand address of location n+1. This enables us to use a 42 order to extract the exponent.

When numbers $(z_1 + 2^{-39} z_2) 2^{E_z}$ are stored in the floating accumulator, three locations are used and the representation is changed:

$$N(2S3) = 2^{-39} (E_z + 512) = 2^{-39} (E_y + 512)$$
$$N(S3) = z_1/2 = y_1$$
$$N(1S3) = z_2/2 = y_2$$

Hence the floating accumulator holds Y = z/2. When they are stored they are put in standard form.

Henceforth it is assumed that Y is the content of the floating accumulator and X is the content of n(and n+1).

This routine is entered as if it were a closed sub-routine and the first interpretive order is the one following the transfer of control. It should be noted that the order code of this routine is exactly like that of the Floating Decimal Code Al with the exception of the addition of the 8L order.

The order code follows X = X(n)

ORDER	TIME (millisec.)	DESCRIPTION
80 n	8.2	Replace Y by Y - X
81 n	4.0	Replace Y by -X
82 n	7.7	Replace Y by Y - $\|X\|$
83 n	2.7	Select the left hand order at n as the next order to be interpreted if Y $\geq$ 0, otherwise proceed normally.
84 n	7.0	Replace Y by Y + X
85 n	2.8	Replace Y by X
86 n	9.1	Replace Y by Y/X
87 n	6.7	Replace Y by XY
88 n		Replace Y and X by the next number on the input tape. Numbers are punched as sign (K or S) followed by decimal digits up to 23, followed by sign of exponent and three decimal digits giving the magnitude of the exponent. e.g. S578693218157 S102 = $-.578693218157 \times 10^{-102}$

89 n		Print Y as sign and n decimal digits, with a space after every 5th digit, followed by the sign of the exponent base 10 and the three decimal digits of the exponent; destroys Y.
8K n	3.0	Replace the parameter g by n and record the location of the next interpretive order.
8S n		Replace X by Y, having first converted it to standard form.
8N n	3.5	Null; replace Y and X by 0
8J n	2.8	Jump; leave the floating code and transfer control to the left hand order at n.
8F n	3.0	Finish; replace the parameter g by g-n; if it is then positive, select the next order to be interpreted as the one following the last 8K order. Otherwise, proceed normally.
8L n	2.4	Select the left hand order at n as the next one to be interpreted.

NOTES

(1) The first digit of any order may be made 0 instead of 8. In this case, the address of that order, n, will be interpreted as n-g instead of n. This is of use in coding induction loops.

(2) The representation of 0 is 0, thus a pair of clear store orders may be used to clear memory space for a 1.7 precision floating binary number.

(3) Numbers punched on output tape may be read back in by this routine.

(4) Because Y isn't standardized after multiplication and division, no more than 7 multiplications or 3 divisions should be allowed before the results are stored if one would maintain full accuracy.

(5) One may leave the floating code by means of an 8J (jump) order, do a computation, and then return to the floating code and execute the order following the 8J order by transferring control to the left hand side of 38L in the floating code.

(6) The 8K and 8F orders are special tallying orders which may be used to cause a certain group of floating orders (an induction loop) to be repeated a pre-determined number of times. The integer g is the tally which is used for this purpose. The 8K order precedes the induction loop and sets g to an initial value. The 8F order follows the induction loop, decreases g by the amount of its address after each traversal of the loop, and allows the loop to be traversed once more if g is then positive. If g is negative, the loop will not be traversed again and the floating order following the 8F order will be interpreted next.

If some order of the induction loop has a 0 first digit, its address will be treated as n-g when the order is obeyed. Since g decreases at each traversal of the induction loop, the effective address of this order will be stepped by the address of the 8F order each time.

As an example, suppose it is desired to find the sum of the squares of the floating numbers stored at 50, 52, ...,100 and store it at location 10. Further suppose that the 1.7 precision routine starts at the position indicated by the preset parameter S4. Then the following sequence of orders will accomplish the task and transfer control to the Ordinary Order following.

25	41 10F	Clear location 10
	50 25L	
26	26 S4	Enter floating code
	8K 50F	Set tally. (Kount)
27	05 100F	
	07 100F	Form square and add to the sum.
28	84 10F	
	8S 10F	
29	8F 2F	Tally to 26
	8J 30L	

DISCUSSION OF THE DETAILS OF THE CODE

ADDITION Since this is a floating binary code, the only thing necessary is to shift the number of the smaller exponent in the A, Q number register before adding double precision.

MULTIPLICATION Let us consider the product XY.

$XY = x\ y\ 2^{E_x + E_y}$ and $xy = (x_1 + 2^{-39}x_2)\ (y_1 + 2^{-39}y_2)$

Let M(U), L(U) be the most and least significant parts of U. Then, because we can use the hold multiply order (74), a convenient formula with an average round-off error of $-1/4 \times 2^{-78}$ is

$$x_1y_1 + 2^{-39}\ M(x_2y_1) + 2^{-39}\ M[x_1y_2 + 2^{-39}\ L(x_2y_1)].$$

DIVISION Due to the method of standardization used in the store order, all we know about x is that $2^{-1} - 2^{-37} < |x| < 1$. Therefore, it could occur that $|y/x| > 2$ and it is necessary to divide y by 4. Then the following formula holds

$$\frac{Y}{X} = \frac{y\ 2^{E_y}}{x\ 2^{E_x}} = \frac{y}{4x}\ 2^{E_y - E_x + 2}$$

where $|y/4x| < 1$.

Let $y/8 = y^* = y_1^* + 2^{-39}y_2^*$. Then the formula

$$\frac{y/8}{x} = \frac{y_1^* + 2^{-39}y_2^*}{x_1 + 2^{-39}x_2} \simeq (1/x_1)\ (y^* - x_2y_1^*/x_1 2^{-39})$$ is

convenient for computation and has a maximum error of 5×2^{-78} for numbers in the range considered.

ROUND-OFF Since numbers are represented to only 68 binary digits when stored, a round-off is desirable. This is achieved by leaving the quantity $E_x + 512$ in place in the 10 digits after 2^{-68}. On output the quantity 0.5×10^{-n}, where n is the address in the print order, is generated and added to or subtracted from the number (depending upon its sign) before printing it.

EXPONENT CONVERSION

Numbers are input and output in the floating decimal system and operated upon in the floating binary system. This makes exponent conversion necessary. Upon input of Y, we are given $Y = y\ 10^p$ and we are required to find E and x: $Y = x\ 2^E$ so x hasn't lost precision.

I INPUT

Given $y\ 10^p$ to find $x\ 2^E$

Write $y\ 10^p = y\ 2^{p \log_2 10} = y\ 2^{E+q}$ (E an integer $-1 < q < 0$)

Then we have $x = y\ 2^{+q}$ so that always $y/2 < x < y$

Now consider that $y\ 10^p = x\ 2^E$

(a) If $p > 0$, then $[p \log_2 10] + 1 = E$, $E < 0$, $[x = y\ 10^p/2^E = y\ \frac{(10p)}{16}\ 2^{4p-E}]$

The rule is to multiply y by 10/16 and if this is less than 1/2 y to scale back by multiplying by 2. After $|p|$ applications, we must have x.

(b) If $p < 0$, again $[p \log_2 10] + 1 = E$, $E < 0$

Now $x = y\ 2^{|E|}/10^{|p|} = y\ (8/10)^{|p|}\ 2^{|E| - 3|p|}$

Use the same rule with 8/10 instead of 10/16.

II OUTPUT

Given $x\ 2^E$, to find $y\ 10^p$ Here $p = [E \log_{10} 2] + 1$

(a) If $E > 0$, $y = x\ \frac{2^E}{10^p} = x\left(\frac{2}{10}\right)^E\ 10^{E-p}$

Multiply x by 2/10 and if this is less than x/10, scale back by multiplying by 10. After $|E|$ applications, we have y.

(b) If $E < 0$, $y = x\ \frac{10^p}{2^p} = x\ \left(\frac{1}{2}\right)^{|E|}\ 10^{|p|}$

Use the same rule with 1/2 instead of 2/10.

DATE May 8, 1953 RT: 10/14/59
CODED BY B Cobb and S. Best
APPROVED BY J. P. Nash

ns

LOCATION	ORDER		NOTES
	00 K(A4)		
0	00 59F		Form:
	L4 16L		50 (n)F 00F
1	L0 15L		50 (n+1)F S5 20F
	40 2L		L5 (n+1)F 00 20F
2	00 F		
	00 F	By 1'	Selecting orders
3	32 10L		
	46 7L	From 11	Set orders to get X(n)
4	L4 260L		
	46 8L		
5	10 12F		
	L4 18L		Set switch
6	46 10L		
	41 5S3		Clear 5S3 for E_x
7	L5 (n)F	By 31	
	40 3S3		Get x
8	50 (n+1)F	By 41	
	S5 165L		
9	42 5S3		Separate out exponent
	L5 5S3		
10	26 ( )F	By 6	To switch
	L0 17L	From 3	If order is positive, subtract g
11	22 3L		from address.
	40 14L	From 28	
12	L5 7L		Store state of routine and replace g
	46 17L	From 37'	by n.
13	L5 14L		
	22 38L		
14	00 F		
	00 F	By 11'	Link
15	5S F		(50 nF S5 20F) - (L5 nF 00 20F)
	S5 F		
16	00 1F		
	S5 20F		Constant
17	80 (g)F		

LOCATION	ORDER		NOTES
	00 F		Parameter g
18	S1 146L		
	22 98L	80	Hold, subtract
19	40 2S3	81	Clear, subtract
	22 96L		
20	L5 3S3	82	Hold, subtract
	22 66L		Absolute value
21	L5 S3	83	Conditional
	22 40L		Left Hand transfer
22	L5 3S3	84	Hold, add
	26 67L		
23	40 2S3	85	Clear, add
	26 95L		
24	L5 2S3	86	Divide
	26 49L		
25	L4 2S3	87	Multiply
	22 41L		
26	81 4F	88	Input
	26 100L		
27	L5 7L	89	Output
	22 137L		
28	L5 2L	8K	Kount
	22 11L		
29	50 1S3	8S	Store
	22 252L		
30	41 S3		
	22 245L	8N	Null
31	L5 7L		
	26 6L	8J	Jump [Left hand escape]
32	L5 7L		
	22 35L	8F	Finish
33	L5 7L	8L	Left hand unconditional transfer
	46 2L		
34	L5 2L		
	36 2L		

LOCATION	ORDER	NOTES	
35	22 39L		
	50 17L	From 32	Form g-n and test
36	46 17L		
	L1 17L		
37	S4 F		
	32 12L		
38	L5 2L	General	Form next selecting orders.
	36 1L	Entry	
39	L4 260L		
	L4 15L	From 35	
40	22 1L		
	36 33L	From 21	
41	26 38L		Multiply $E_{xy} + 512 = (E_y + 512) +$
	L0 262L	From 25	$(E_x + 512) - 512$
42	40 2S3		
	7J S3		
43	40 6S3		y_1x_2 ; $M(y_1x_2)$ to 6S3
	S5 F		
44	50 3S3		$(x_1y_1 + M(y_2x_1 + L(y_1x_2)\ 2^{-39})2^{-39}$
	74 1S3		
45	50 S3		
	74 3S3		to A and Q
46	40 7S3		
	S5 F		$(x_1y_1 + (M(y_1x_2) + M(y_2x_1 + L[x_2y_1]$
47	50 261L		$2^{-39})\ 2^{-39}$
	74 6S3		
48	L4 7S3		
	26 93L		To store in floating decimal accum.
49	L0 5S3	From 24	Divide
	L4 265L		Form $E_y - E_x + 2$
50	40 2S3		
	7J S3		Form $x_2y_1/4$
51	10 2F		
	66 3S3		
52	S1 F		$x_2(y_1/4)/x_1$

LOCATION	ORDER	NOTES	
	40 4S3		
53	50 1S3		
	L5 S3		$y^* = [y/2] + 4$
54	10 2F		
	40 S3		
55	S5 F		
	50 261L		$\frac{y - y_1 x_2}{x_1} 2^{-39}$
56	74 4S3		
	L4 S3		
57	40 11S3		
	66 3S3		
58	10 1F		
	40 5S3		
59	S5 F		
	40 4S3		
60	50 11S3		
	L5 5S3		
61	10 39F		
	71 65L		
62	66 3S3		
	J0 268L		
63	L5 4S3		
	00 1F		
64	26 93L		
	00 F		
65	80 F		
	00 3F		
66	00 F		
	36 18L		
67	10 1F	From 22, 99	Hold, add
	40 3S3		$x/2$
68	L5 5S3		Form and store $E_x = E_y$
	L0 2S3		
69	40 4S3		
	L3 4S3		If $E_x = E_y$ go directly to add

LOCATION	ORDER		NOTES
70	32 86L		
	L1 4S3		
71	36 78L		If $E_y < E_x$, interchange x and y
	L5 5S3		
72	40 2S3		
	L5 1S3		
73	40 5S3		
	S5 F		
74	40 1S3		Interchange x and y
	50 S3		
75	L5 3S3		
	40 S3		
76	[illegible]		
	40 3S3		
77	50 5S3		
	[illegible] 4S3		If [illegible] E_y [illegible] skip addition
78	[illegible]0 264L	From [illegible]1	
	36 38L		
79	[illegible] 264L		
	[illegible] 1F		
80	42 85L		
	01 1F		Set address of shift orders
81	L4 85L		
	42 82L		
82	L5 3S3		
	10 ( )F	By 81	Shift (integer part of E_x-E_y) +
83	40 3S3		(0 or 1) accordingly as E_x-E_y) is
	L5 269L		(even or odd)
84	L0 85L		Test for 0 shift
	32 86L		
85	L5 3S3		
	10 ( )F	By 80	Shift (integer part of E_x-E_y)/2
86	40 3S3		
	01 39F		
87	50 261L		

[illegible]	ORDER		NOTES
	[illegible] 1S3		Add together the properly [illegible]
88	L4 S3		modulus
	L4 3S3		
89	40 S3		
	LL S3		Scale modulus down if it is greater
90	32 93L		than 1/2.
	L5 2S3		
91	L4 261L		
	40 2S3		Divide modulus by 2.
92	L5 S3		
	10 1F		
93	40 S3		
	S5 108L		Store modulus
94	40 1S3		
	26 (38L)	By 103,	
95	L5 3S3	146,203	Clear, add
	10 1F		
96	26 93L		
	S1 153L		Clear subtract
97	10 39F		
	L0 3S3		
98	22 95L		
	10 39F		Hold subtract
99	L0 3S3		
	26 67L		
100	50 261L	From 26'	Input
	00 39F		
101	10 5F		
	L4 59L		
102	40 111L		
	L5 93L		
103	42 94L		according as sign is + or -.
	L5 273L		
104	40 5S3		
	40 1S3		

LOCATION	ORDER		NOTES
105	L5 271L		
	40 3S3		
106	40 S3		
	41 8S3		Clear 8, 9S3 where modulus will be
107	41 9S3		stored.
	50 261L		
108	81 4F	From 94	
	L0 267L		Input and test for non-decimal
109	36 120L		
	50 261L		
110	L4 267L		
	10 4F		
111	00 F		
	00 F	By 102	$\pm$ D/16
112	50 4S3		
	7J 1S3		
113	50 4S3		-10^{-n} ($\pm$ D/16) 8
	74 S3		
114	00 3F		
	40 2S3		
115	S5 F		
	50 261L		N(8,9S3) + ($\pm$ D/16) 8 10^{-n}
116	74 9S3		
	L4 2S3		
117	L4 8S3		
	40 8S3		
118	S5 F		Store at 8, 9S3
	40 9S3		
119	50 5S3		Go to form $10^{-(n+1)}$ at S3, 1S3
	22 42L		
120	40 2S3	From 109	
	L5 8S3		Store (sign of exp. -10)
121	40 S3		
	10 1F		
122	40 8S3		y to S3, 1S3 and $y_1/2$ to 8S3

LOCATION	ORDER		NOTES
	L5 9S3		
123	40 1S3		
	81 4F		
124	40 3S3		
	81 4F		
125	50 267L		
	74 3S3		
126	S5 F		
	40 3S3		Form decimal exponent p
127	81 4F		
	50 267L		
128	74 3S3		
	L3 2S3		
129	32 130L		
	S1 191L		
130	26 131L		
	S5 147L	From 129	
131	40 5S3	From 130	
	36 136L		
132	L5 273L		If $p \geq 0$ / $p < 0$ set 10/16 / 8/16 to 3, 4S3
	40 3S3		
133	10 1F		
	40 4S3	From 137	
134	L7 105L		Set IN-OUT switch to IN
	40 105L		
135	L5 274L		$(\log_2 10)/4$ to 2S3
	22 178L		
136	L5 270L	From 131'	10/16 to 3, 4S3
	40 3S3		
137	23 133L		
	L0 277L	From 27'	Output
138	40 10S3		$10S3 = (M-1)\ 2^{-19}$
	L0 260L		
139	40 11S3		$11S3 = (n-2)\ 2^{-19}$
	L5 159L		

LOCATION	ORDER	NOTES	PAGE 9 A 4
140	42 258L		Set return after store
	L5 122L		Set address to store in 8, 9S3
141	46 7L		
	26 29L		to store
142	L5 271L	From 258	
	40 3S3		
143	40 S3		Set 1/10 = p
	50 273L		
144	S5 F		
	40 1S3		
145	40 4S3		
	L5 130L		Set return after multiplication
146	42 94L		
	22 42L		
147	50 4S3		
	L5 11S3		Count for generating 10^{-n}
148	L0 260L		
	40 11S3		
149	32 42L	Yes	Is 10^{-n} generated?
	L5 96L		
150	42 258L		Set return after store
	L5 9S3		
151	42 2S3		Set Exp = E_y
	L5 43L		Set address to store to 6, 7S3
152	46 7L		
	26 29L		To store
153	41 2S3	From 258	
	50 9S3		
154	S5 F		
	42 2S3		
155	L5 8S3		y back to f.d. accumulator
	10 1F		
156	40 S3		
	S5 278L		
157	40 1S3		

LOCATION	ORDER		NOTES
	L5 156L		Return after store
158	42 94L		
	50 7S3		
159	41 5S3		Separate out exponent
	S5 142L		
160	42 5S3		
	L5 6S3		
161	10 2F		$- .5 \times 10^{-(n)}$
	40 3S3		
162	L5 S3		if positive go to hold add
	36 22L		if negative go to hold subtract
163	26 18L		
	L5 8L	From 278	Set return after store
164	42 258L		To store at 6, 7S3
	26 29L		
165	L5 13L		
	42 258L		Reset store return
166	41 5S3		
	L5 7S3		
167	42 5S3		
	L5 5S3		
168	50 7S3		
	L0 262L		
169	40 5S3		
	L5 6S3		
170	40 S3		
	S5 F		x to S3, 1S3
171	40 1S3		
	92 513F		
172	50 271L		
	75 S3		1/10 y_1 to 8S3
173	40 8S3		
	L5 5S3		
174	32 175L		2/10 To 3S3
	49 3S3		1/2

LOCATION	ORDER		NOTES
175	23 177L		if exponent is positive / negative
	L5 273L	From 74	
176	10 2F		
	40 3S3		
177	L5 272L		
	40 4S3	From 175	
178	L5 275L		$(\log_{10}2)/4$ to 2S3
	40 2S3	From 135'	
179	L3 5S3		Exponent Conversion
	36 182L		Test exponent for 0
180	50 2S3		If exponent ≠ 0 calculate
	75 5S3		
181	00 2F		[Exp $\log_2 10$] + 1
	L4 261L		or
182	40 2S3	From 179'	[Exp $\log_{10}2$] + 1
	L5 129L		
183	42 94L		
	L3 5S3	From 192	
184	32 186L		
	L4 261L		Go to x to multiply by appropriate
185	40 5S3		constant
	50 4S3		
186	22 42L		
	L5 13L	From 184	
187	42 94L		Reset x return IN-OUT switch
	L1 105L	out	
188	32 198L	in	
	40 105L		Restore to out
189	L5 2S3		
	L4 262L		Form [Exp + 512]
190	40 2S3		
	26 29L		to store
191	L7 S3	From 94'	
	L2 8S3		
192	32 183L		

LOCATION	ORDER	NOTES	
	L5 105		
193	32 196L	In Out	
	50 270L	From 204	
194	7J 1S3		
	50 270L		Scale by 10
195	74 S3		
	00 4F		
196	26 93L		
	L5 S3	From 193'	
197	50 1S3		
	00 1F		
198	26 93L		Print
	L7 S3	From 188	
199	L0 271L		
	L4 261L		
200	32 204L		
	L3 S3		
201	32 204L		
	L5 2S3		
202	L0 261L		
	40 2S3		
203	L5 129L		
	42 94L		
204	22 193L		
	L5 259L	From 200,	(Layout number)
205	40 8S3	201	
	L5 S3		
206	32 209L		
	92 706F		- sign Form |x| and print
207	L1 1S3		
	10 39F		
208	L0 S3		
	40 S3		
209	22 210L		
	92 642F	From 206	+ sign

LOCATION	ORDER		NOTES PAGE 13 A4
210	50 1S3	From 219'	
	7J 270L	From 209	
211	50 S3		Form x 10/16
	J0 268L		
212	74 270L		
	00 1F		Print a digit
213	82 4F		
	10 1F		
214	40 S3		
	S5 F		
215	40 1S3		
	L5 8S3		
216	L4 8S3		Pass test for space
	40 8S3		
217	36' 218L		
	92 963F		Space
218	L5 10S3		
	L0 260L		Pass test to print exponent
219	40 10S3		
	36 210L		
220	92 963F		Space
	49 3S3		Form: $E_x(1/1000 + .0005)$
221	50 2S3		
	00 30F		
222	7J 276L		
	40 S3		
223	32 226L		
	92 706F		- sign
224	L1 1S3		
	10 39F		
225	L0 S3		
	40 S3		
226	22 227L		Print exponent
	92 642F		+ sign
227	50 1S3		

LOCATION	ORDER	NOTES	
	7J 270L	From 226	
228	50 S3		
	J0 268L		
229	74 270L		
	00 1F		
230	82 4F		
	10 1F		
231	40 S3		= -1 after 1st digit
	49 4S3		= -1.5 after 2nd digit
232	L4 3S3		= 0 after 3rd digit
	40 3S3		
233	36 38L	Quit	
	22 227L		Return to print
234	L5 S3	From 240	
	50 1S3		
235	00 1F		
	40 S3		Scale up by 2 and adjust exponent
236	S5 F		
	40 1S3		
237	L5 2S3		
	L0 261L		
238	40 2S3		
	19 1F	From 255	Shift by 1 if $2^{-2} - 2^{-39} \|y\| \geq 0$
239	L0 261L		
	L2 S3		
240	36 234L		
	L5 7L	From 246'	
241	46 244L		
	L4 260L		Set store orders
242	46 258L		
	L5 S3		
243	50 1S3		
	00 1F		
244	40 (n)F	By 241,	Store 1st 1/2 of modulus
	L5 2S3	From 246	

LOCATION	ORDER	NOTES	
245	32 255L		If $E_y < -512$, replace number by 0
	41 2S3	From 247	
246	41 1S3	30	
	26 279L		
247	S3 F		If $y_2 = 0$ also, replace number by 0
	32 245L		
248	L5 S3	From 254'	
	50 1S3		
249	00 8F		Shift by 8 and adjust exponent
	40 S3		
250	S5 F		
	40 1S3		
251	L5 2S3		
	L0 263L		
252	40 2S3		
	L3 S3	From 29'	Is $y_1 = 0$?
253	36 247L	No	
	19 9F		
254	L2 S3		Shift by 8 if $\|y_1\| \leq 2^{-10}$
	36 248L		
255	22 238L		
	10 10F	From 245	
256	L0 261L		
	30 256L		If $E_y \geq 512$, stop unrelentingly
257	01 10F		Clear R_1 and shift left 10
	S4 F		
258	40 (n+1)F	By 242	Store 2nd 1/2 of modulus
	26 (38L)	By 140,	
259	04 528F	150	
	84 528F		Layout number
260	00 1F		
	00 F		2^{-19}
261	00 F		
	00 1F		2^{-39}
262	00 F		

LOCATION	ORDER	NOTES
	00 512F	512×2^{-39}
263	00 F	
	00 8F	8×2^{-39}
264	00 F	
	00 79F	79×2^{-39}
265	00 F	
	00 514F	514×2^{-39}
266	80 F	
	00 F	(-1)
267	00 F	
	00 10F	10×2^{-39}
268	7L 4095F	
	LL 4095F	$(1 - 2^{-39})$
269	L5 3S3	Constant to test for 0 shift
	10 F	
270	50 F	
	00 F	10/16
271	ON 3276F	
	NN 3276F	1st 1/2 of 1/10
272	4N 3276F	
	NN 3276F	2nd 1/2 of 2/10
273	66 1638F	1st 1/2 of 8/10.
	66 1638F	also 2nd 1/2 of 1/10
274	40F 00 3304	
	8202 4000 J	$(\log_2 10)/4$
275	00F 00 752	
	5749 8941 J	$(\log_{10} 2)/4$
276	40 F 00 0140	
	0000 0000 J	.516
277	L5 1F	
	40 3S3	
278	92 131F	
	22 163L	
279	41 S3	
	22 240L	