

ELLIOTT

9000

Volume 2: PROGRAMMING INFORMATION

Part 5: SIR SYSTEMS

Section 2: QDLA (DOUBLE LENGTH FIXED-POINT INTERPRETER)

Contents

	Page
Chapter 1: INTRODUCTION	
1.1 Purpose	1
1.2 Summary	1
1.3 Form of Distribution	1
1.4 Method of Use	1
1.5 Restrictions	1
1.6 Accuracy	1
1.7 Subroutines Used	2
Chapter 2: FUNCTIONS	
2.1 Notation	3
2.2 Format	3
2.3 Entry and Exit	3
2.4 Operations Available	4
2.5 Input and Output	5
2.5.1 Character Set for Input	6
2.5.2 Input	6
2.5.3 Scaling Factor	6
2.5.4 Output Format	7
Chapter 3: ERROR INDICATIONS	9
Chapter 4: METHOD USED	10

	Page
Chapter 5: TIME TAKEN	11
Chapter 6: STORE USED	12

Printed in England by
Engineering Unit, Elliott-Automation Computers Ltd.

Chapter 1: INTRODUCTION

1. 1 Purpose

To perform arithmetic operations upon double-length fixed-point numbers (x) in the range

$$- 1 < x < 1 - 2^{-34}$$

1. 2 Summary

The double-length number routines are interpretive, providing equivalents of most of the machine code instructions of the 903, and allowing input and output of numbers in fraction and integer formats.

When entered, QDLA proceeds to interpret the instructions in the store locations immediately following the entry in the users area. Thus operations can be performed on double-length numbers by placing in store the corresponding single-length instruction.

1. 3 Form of Distribution

The programs are distributed as machine-code tapes for input by 903 SIR.

1. 4 Method of Use

QDLA is assembled as a block of the user's SIR program.

1. 5 Restrictions

Some instructions cannot be interpreted. See notes in 2. 4.

1. 6 Accuracy

With the following exceptions no error is introduced by the routines:

- (a) Multiply: maximum error is $+2^{-34}$ (0.6×10^{-10})
- (b) Divide: maximum error is $+2^{-32}$ (0.2×10^{-9})
- (c) Input and output: maximum error is $\pm 2^{-34}$ (0.6×10^{-10})

1.7 Subroutines Used

QDLA uses all the systems subroutines on the systems tape SSYS1, i. e. QCHOP, QCHIN, QERROR, QPAUSE.

SSYS1 must be assembled as part of the users SIR program.
(See Volume 2. 5. 5.).

Chapter 2: FUNCTIONS

2.1 Notation

$x(\text{m. s.})$ = most significant half of x
 $x(\text{l. s.})$ = least significant half of x
 x = double-length number held in locations X and $X+1$
 p = double-length number held in the pseudo-accumulator, p
 $C(X)$ = single-length number held in location X
 $C(B)$ = single-length number held in pseudo B-register
 $C(S)$ = single-length number held in pseudo S. C. R.
 $:=$ means "becomes equal to".

2.2 Format

A double-length number, x , is held in two consecutive store locations, X and $X+1$.

Location	Bit 18	Bits 17-1
X	sign	most significant bits of x
$X + 1$	0	least significant bits of x

Negative number representation is by the usual 2's complement notation. (N. B. bit 18 of $X+1$ is always zero).

2.3 Entry and Exit

Entry is made by

11 QDLA+20
 8 QDLA+21

The routine proceeds to interpret the double-length instructions that follow the entry, using pseudo registers which are analogous to the registers in the computer hardware. These pseudo registers are represented in locations within QDLA, as follows:

a pseudo-accumulator(double-length) located in $\begin{matrix} \text{QDLA+16} \\ \text{QDLA+17} \end{matrix}$
 a pseudo B-register (single-length) located in QDLA+18
 a pseudo S-register (single-length) located in QDLA+20

Exit is made by placing zero (+0) in the location after the last instruction to be interpreted. Control is transferred to the location following the zero location. The pseudo-accumulator and pseudo B-register are not affected by entry and exit.

2. 4 Operations Available

See 2. 1 for notation used.

Pseudo Instruction	Name	New contents of		Remarks
		P	X	
0 X	Load pseudo B-register	p	x	C(B):=C(X) see notes 1 and 3
1 X	Add	p+x	x	
2 X	Negate and add	x-p	x	
3 X	Store 1. s. only	p	C(X):= the 17 least significant bits of p	C(X+1) unchanged
4 X	Load P	x	x	
5 X	Store P	p	p	
6 X	Store scaling factor	p	x	see 2. 5. 3.
7 X	jump if zero	p	x	if p=0 then C(S):=X
8 X	jump	p	x	C(S):=X
9 X	jump if negative	p	x	if p<0 then C(S):=X
10 X	Count in store	p	C(X):=C(X)+2 ⁻¹⁷	C(X+1) unchanged
11 X	store pseudo	p	C(S)	
12 X	multiply	p*x	x	
13 X	divide	p/x	x	see 3. 2
14 X	(* 2 ^N (* 2 ^{N-8192}	p*2 ^N		n<47
		p*2 ^{N-8192}	x	n>8158
15 2048)	input number	<input>	-	see 2. 5
15 2052)				
15 6144)	output number	unchanged	-	see note 2 and 2. 5
15 6148)				

Notes

1. 0 instruction
The instruction 0 0 is interpreted as a terminator for double-length working (see 2. 3). It does not affect the pseudo B-register. The hardware A-register is set equal to p(m. s.).
2. 15 6144 instruction (and 15 6148).
This must follow by a parameter word to specify the format used. The next instruction interpreted is that following the parameter word. See 2. 5. 2 for format of parameter word.
3. Modified instructions may be used: the contents of the pseudo B-register are added to the address digits to find the address of the operand.
4. Literal instructions may not be used.

2. 5 Input and Output

The 15 instructions corresponding to input and output of paper tape and teleprinter characters in machine code are interpreted as input and output of complete numbers from and to the appropriate devices. e. g. 15 6148 outputs the number in the double-length accumulator to the teleprinter.

The double length arithmetic functions effectively operate on fractional numbers in the range

$$-1.0 \leq x < 1.0$$

However, the programmer may wish to operate on numbers in other ranges. As with single length working, it is frequently convenient to regard a double length value as an integer (in the range $-2^{36} \leq x < +2^{36}$).

Input and output of numbers in this range is allowed, also input and output of mixed numbers which are scaled so that their internal representation is in the correct range. The programmer specifies the scaling constant to be used.

When using input and output of numbers in the additional ranges, the programmer must always remember that the internal representation is in fraction form, particularly when multiplying and dividing.

2. 5. 1 Character Set for Input

All characters which have a representation in SIR internal code (Volume 2. 1. 1. Chapter 1. 5.) are acceptable. Blank, erase and carriage return are ignored wherever they occur.

The following characters are significant in the formation of numbers:

digits 0 to 9
decimal point
+ and -

All other characters are treated as separators.

On commencing input, separators are ignored until one of the significant characters is read. Once a digit has been read the occurrence of a separator terminates the number. Thus separators may be used freely between numbers, including letters used for descriptive text.

2. 5. 2 Input

The format of an input number determines the way it is processed. A number not containing a decimal point is treated as an integer, N , and stored as the fraction $N \times 2^{-34}$

The range of N is:

$$- 17179869184 < N < 17179869184$$

A number containing a decimal point is treated as a scaled fraction, F . If M is the current value of the scaling factor (see 2. 5. 3.) the number will be stored as the fraction:

$$F \times 10^{-M}$$

The range of F is:

$$- 10^M \leq F < + 10^M$$

F may not have more than 10 digits in all.

2. 5. 3 Scaling Factor

The scaling factor, M , used for input and output is set by an interpreted 6 instruction.

A positive scaling factor is set by instruction:

6 M

A negative scaling factor is set by instruction:

6 (8129-M)

Examples:

6 Instruction	Scaling Factor	Number Input	New Content of P
6 3	+3	12. -123.46 1.5	0.012 -0.012346 0.0015
6 8190	-2	-.001234 .0001	-0.1234 0.01
6 0	+0	0.999 -.05	0.999 -0.05

The scaling factor set on the tape of QDLA is +0. However, it is advisable to set the scaling factor explicitly in all programs, even if factor zero is to be used.

2. 5. 4 Output Format

Numbers are output right justified, with non-significant zeros suppressed and the sign floated (i. e. immediately preceding the first significant character).

Output format is controlled by the parameter word which follows the 15 6144 and 15 6148. This is written as a pseudo instruction: /f k. If / is used (i. e. the parameter word is negative) the number output is preceded by newline. The number f controls the output format as described below, and the address part k denotes the type of number to be output.

If k = zero the number p is output as an integer:

$$p \times 2^{34}$$

This output is independent of the scaling factor. The number output occupies 12-f printing positions.

If k = 4096 a fraction will be printed:

$$p \times 10^M$$

f digits are printed after the decimal point, and the total number of printing positions occupied is $f + M + 3$.

Examples:

Parameter	p	Output
/0 0	$-17179869183 \times 2^{-34}$	-17179869183
/0 0	2^{-17}	131072
/0 0	-0.03125(i. e. -2^{-6})	-536870912
/5 0	123456×2^{-34}	123456
/4 4096	0.12345678 (and scaling factor + 3)	123.45678
/0 4096	0.99999 (and scaling factor + 4)	10000.
/8 4096	0.03125 (and scaling factor + 0)	.03125000

Chapter 3: ERROR INDICATIONS

Error messages are output by QERROR, the error number in octal form being preceded by *QDL. The error number is followed by the address of the next instruction to be interpreted by QDLA. After an error, the program may be continued at the next instruction by re-entry at location 9, the state of the pseudo accumulator being as defined below.

Error No.	Meaning	Value of p on continuation
0020	Attempt to divide by zero	Unchanged
0021	Input Format Error e. g. two decimal points in number	0.0
0022	Input overflow i. e. number too large for input with the current scaling factor	$(1.0 - 2^{-34})$
0024	More than 10 digits in input fraction	0.0
0025	15 instruction with illegal address	Unchanged
0026	Overflow during division	$(1.0 - 2^{-34})$

Chapter 4: METHOD USED

4.1 The following steps are carried out for each pseudo-instruction interpreted:-

1. The address (modified if required) is placed in 19; of QDLA.
2. The pseudo S. C. R. is incremented.
3. If the function is f control is transferred to location f; of QDLA.

This location contains a jump to the appropriate routine for the operation. The operation is carried out and then the next pseudo-instruction is interpreted.

4.2 QDLA uses the following locations for the purposes indicated:

16; and 17;	pseudo-accumulator
18;	pseudo B-register
19;	address (modified if necessary) of pseudo-instruction
20;	pseudo S. C. R. and link for exit

4.3 Conditional Jumps

- 7 examines both locations.
- 9 examines first location only.

Chapter 5: TIME TAKEN

The following times are approximate:

Function Number	Operation	Time in μs	
0	Set pseudo B-register	492	
1	Add	658	
2	Negate and add	688	
3	Store least significant half	492	
4	Load	569	
5	Store	525	
6	Store scaling factor	555	
7	Jump if zero	(p<0	440
		(p=0	581
		(p>0	482
8	Jump	509	
9	Jump if negative	(p<0	535.5
		(p>0	457
10	Count in store	467	
11	Store pseudo SCR	539	
12	Multiply	919-1208	
13	Divide	2788-5481	
14	Shift	695+3N	
15	Input or output	Acts at the speed of the appropriate peripheral	

For B modification add 161 μs to the above times.

Chapter 6: STORE USED

QDLA uses approximately 820 locations.

Its use requires that the routines of SSYS1 are in store, these take approximately 200 locations.