





FPMP-11 USER'S MANUAL

C

Cy.

For additional copies, order No. DEC- 11-NFPMA-C-D from Software Distribution Center, Digital Equipment Corporation, Maynard, Mass.

digital equipment corporation · maynard. massachusetts

First Printing September, 1972 Revised, April, 1973 Revised, July, 1973

ø

f\$

Your attention is invited to the last two pages of this document. The "How to Obtain Software Information" page explains how to keep up-to-date with DEC's software. The "Reader's Comments" page, when filled in and mailed, is beneficial to both you and DEC; all comments received are acknowledged and considered when documenting subsequent manuals.

Copyright 🕐 1972, 1973 by Digital Equipment Corporation

Changes from the previous version are indicated by a bar () in the margin.

The material in this document is for information purposes and is subject to change without notice.

Teletype is a registered trademark of the Teletype Corporation.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

CDPDigitalComputer LabDNCComtexFlip ChipDECIDACDECtapeIndacDibolKA10	LAB-8/e OMNIBUS OS/8 PDP PHA PS/8 Quickpoint	RAD-8 RSTS RSX RTM SABR Typeset 8 Unibus
--	--	--

PREFACE

ø

0

This manual assumes the reader is familiar with PDP-11 assembly language programming and with floating point operations in general.

For background in the papertape system, refer to the PDP-11 Paper Tape Software Programming Handbook (DEC-11-XPTSA-A-D).

TABLE OF CONTENTS

 ρ

CHAPTER :	1	FPMP-11 OVERVIEW	1-1
1.1		INTRODUCTION HARDWARE REQUIREMENTS	1-1 1-1
1.3		SOFTWARE REQUIREMENTS	1-1
1.4		FLOATING-POINT NOTATION	1-1
1.5		FLOATING-POINT NUMBER STORAGE	1-3
			1-3
1.5.2		Single Precision Double Precision	1-4
CHAPTER 2	2	DESCRIPTION OF PACKAGE	2-1
2.1		SINGLE PRECISION PACKAGE	2-1
2.2		DOUBLE PRECISION PACKAGE	2-2
2.3		SOURCE TAPES	2-2
2.4		CONVERSION ROUTINES	2-2
מינותם געני	2	USING FPMP-11	3-1
CHAPTER .	3	USING FPMP-II	2-1
3.1		USING THE TRAP HANDLER WITH FPMP-11	3-1
3.1.1		Stack Mode	3-2
3.1.2		@R0 Mode	3-2
3.1.3		Immediate Mode	3-2
3.1.4		Relative Mode ACCESSING USER ROUTINES VIA THE TRAP HANDLER	3-2
3.2		ACCESSING USER ROUTINES VIA THE TRAP HANDLER	3-6
3.3		DIRECT CALLS TO OTS ROUTINES	3-8
3.3.1		Polish Mode	3-8
3.3.2		J5RR Mode	3-10
3.3.3		JPC Mode	3-12
J.4		ERRORS	3-15
3.4.1		Using Error Handling Routines CREATING SPECIAL PACKAGES	3-15
3.5		CREATING SPECIAL PACKAGES	3-17
3.5.1		Assembly Switch Tape	3-17
3.6		LOADING INSTRUCTIONS	3-22
CHAPTER 4	,	SAMPLE PROGRAM	4-1
CHAPIER	4	SAMPLE PROGRAM	4-1
APPENDIX	Z	BOOTSTRAP AND ABSOLUTE LOADERS	A-1
		USING THE PAL-11S ASSEMBLER	B-1
		USING LINK-11S	C-1
		SUMMARY OF FPMP-11 ROUTINES	D-1
		FPMP-11 SOURCE LISTING	E-1
	-		

 \mathbf{v}

CHAPTER 1 FPMP-11 OVERVIEW

1.1 INTRODUCTION

The Floating-Point Math Package, FPMP-11, is designed to bring the 2/4 word floating point format of the FORTRAN environment to the paper tape software system of the PDP-11. The numerical routines in FPMP-11 are the same as those of the DOS-11 Fortran Object Time System (OTS). TRAP and error handlers have been included to aid in interfacing with the FORTRAN routines.

FPMP-11 provides an easy means of performing basic arithmetic operations such as add, subtract, multiply, divide and compare. It also provides transcendental functions (SIN, COS, etc.), type conversions (integer to floating point, 2 word to 4 word, etc.) and ASCII conversions (ASCII to 2 word floating point, etc.).

Floating-point notation is particularly useful for computations involving numerous multiply and divide operations where operand magnitudes may vary widely. FPMP-11 stores very large and very small numbers by saving only the significant digits and computing an exponent to account for leading and trailing zeros.

To conserve core space in a small system, FPMP-11 can be tailored to include only those routines needed to run a particular user program.

1.2 HARDWARE REQUIREMENTS

The FPMP-11 package is designed for use on any PDP-11 with at least 8K of core, and can be easily reassembled to take advantage of the 11/20 EAE, 11/45 EIS, or 11/45 FPU (refer to section 3.5 for detailed instructions).

1.3 SOFTWARE REQUIREMENTS

LINK-llS (or the DOS LINK-ll linker) is used to link a user program with an FPMP-ll object module to create a load module. PAL-llS (or MACRO-ll under DOS-ll) is used whenever the FPMP-ll package is reassembled.

1.4 FLOATING-POINT NOTATION

A floating-point number may be written as a mantissa, which consists of the floating-point number with its decimal point shifted a given number of places in either direction, and an exponent which indicates the number of places that the decimal point was shifted and the direction of the shift. A negative exponent corresponds to a shift to the right, while a positive exponent corresponds to a shift to the left. Thus, the mantissa multiplied by the base (radix) of the number system in use, raised to a power as supplied in the exponent, gives the value of the number in fixed-point notation. For example, the decimal number 12 in fixed-point notation can be represented as

12 or 12.0

In floating-point notation with a base of 10, the number might appear as

 $.12 \times 10^{2}$

where the mantissa is .12 and the exponent, 2.

A fraction, such as twelve ten-thousandths, is represented as

.0012

in fixed-point notation and in floating-point notation as

 $.12 \times 10^{-2}$

The minus sign before the exponent indicates that the significant digits of the mantissa are to be shifted right from the decimal point.

In FPMP-11 all numbers are manipulated and stored in binary notation. With a radix of 2, the decimal number 12 is represented as

1100

and in floating-point format as

 $.1100 \times 2^4$

Multiplication and division are accomplished by shift operations: each one - place shift to the left represents multiplication by two; each equivalent shift to the right represents division by two.

A floating-point number may be represented in an infinite variety of ways, since the decimal point may be shifted any number of places in either direction. If the decimal point is shifted until it appears immediately to the left of the most significant digit, the number is said to be normalized. The mantissa of a normalized floating-point number may be stored as an integer, since the decimal point is understood to appear to the left of the most significant digit. In

5

computing a mantissa from decimal input, FPMP-11 uses the convention

$1/2 \leq |MANTISSA| < 1$

to normalize the input value. Note that when |MANTISSA| is stored as a binary fraction in normalized form, the left most (high order) bit is always a 1. The only exception to the normalization rule is the floating-point zero (either single or double precision) which has a mantissa and exponent both equal to zero.

1.5 FLOATING-POINT NUMBER STORAGE

c

FPMP-11 floating-point numbers are stored as two 16-bit PDP-11 words (single precision) or four 16-bit PDP-11 words (double precision). The sign of the number is bit 15 of the first word. (0 indicates positive, 1 indicates negative). The binary exponent is stored in bits 14-7 of the first word. The exponent is stored in excess 128 (200₈) code. The value of the exponent is obtained by subtracting 200₈ from bits 14-7 of the first word.

NOTE

The single and double precision formats shown below are limited to normalized numbers. The high-order bit of the mantissa (which is always 1) is omitted from its implied position (bit 7 of WORD n) in order to allow one more bit in the exponent field.

1.5.1 Single Precision

The mantissa and exponent are stored as follows:

W	ORD	n			
	S	expo	nent	high-order m	antissa
	15	14	7	6	0

WORD n+2

low-order	mantissa		
15	· · ·	0	

The first word (lowest core address) contains the sign of the mantissa, the exponent excess 128_{10} and the high-order mantissa (absolute value). The second word is the low-order mantissa (absolute value continued).

1.5.2 Double Precision

Double precision format is identical to single precision format except that it has two additional words (WORD n+4 and WORD n+6) of low-order mantissa.



The list below provides examples of numbers in decimal form, binary floating point notation and single precision internal form.

Decimal Value	Binary Floating Point	Internal Form Single Precision (octal)
1	0.1 x 2 ¹	040200 000000
2	0.1×2^2	040400 000000
5	0.101×2^3	040640 000000
10	0.101×2^4	041040 000000
$\sqrt{2}$	0.10110101×2^{1}	040265 002363
-1	-0.1×2^{1}	140200 000000
0.5	0.1×2^{0}	040000 000000
0.25	0.1×2^{-1}	037600 000000
0.75	0.11×2^{0}	040100 000000
-0.25	-0.1×2^{-1}	137600 000000

6

CHAPTER 2

DESCRIPTION OF PACKAGE

As distributed the FPMP-ll package contains three sub-packages: the object tape of the single precision functions, the object tape of the double precision functions and the source tapes.

2.1 SINGLE PRECISION PACKAGE

s a

C

The single precision package is an object module tape (DEC-11-NFPMA-A-PR1) which includes the FPMP-11 TRAP and error handlers and the following OTS routines for two-word floating point operation:

\$ADR	Add routine
\$SBR	Subtract routine
\$MLR	Multiply routine
\$DVR	Divide routine
\$CMR	Compare routine
SIN	Sine routine
COS	Cosine routine
AINT	Truncation routine
ATAN	Arctangent routine
ATAN 2	Arctangent routine with two arguments
SQRT	Square root routine
TANH	Hyperbolic tangent routine
EXP	Exponential routine
ALOG	Natural logarithm routine
ALOG10	Base-10 logarithm routine

and the ASCII input/output conversion routines. There are also routines to load and store the FLAC (FLoating-point ACcumulator) which may be called through the TRAP handler. (Refer to section 3.1.)

The functions are identical to their FORTRAN counterparts and are described in more detail in Appendix D.

2.2 DOUBLE PRECISION PACKAGE

The double precision package is an object module tape (DEC-11-NFPMA-A-PR2) which includes the TRAP and error handlers and the following OTS routines for four-word floating point operations:

٩.

\$ADD	Add routine
\$SBD \$MLD	Subtract routine
•	Multiply routine
\$DVD	Divide routine
\$CMD	Compare routine
DSIN	Sine routine
DCOS	Cosine routine
DATAN	Arctangent routine
DATAN2	Arctangent routine with two arguments
DSQRT	Square root routine
DEXP	Exponential routine
DLOG	Natural logarithm routine
DLOG10	Base-10 logarithm routine
DTOGTO	Base-IV IOYaIICHM IOUCINE

and ASCII input/output conversions routines. There are also routines to load and store the FLAC which may be called through the TRAP handler (refer to section 3.1).

Appendix D contains a more detailed description of the functions.

2.3 SOURCE TAPES

The source tapes (DEC-11-NFPMA-A-PA1-PA6) contain the source code for the TRAP handler, the error handler and all the OTS routines described in Appendix D. Conditional assembly instructions are included in the source code to aid in the construction of specially tailored packages. For example, an object tape of only the TRAP and error handlers and the arithmetic functions, add, subtract, multiply and divide can be easily created. Such a package can result in great savings of core when the other functions are not required. (Refer to section 3.5 for information on creating special packages.)

2.4 CONVERSION ROUTINES

The subroutines included in FPMP-11 to perform conversions to and from ASCII strings are those used by FORTRAN to perform Input/Output. The FPMP-11 routines do not perform any actual I/O, but simply convert strings of ASCII characters in memory to the internal form of floating-point numbers or integers used by other FPMP-11 subroutines and convert numbers in internal form to ASCII strings.

In order to effectively use the ASCII conversion routines of FPMP-11, the meaning of the various parameters which must be passed to these routines and the various data formats involved must be understood. Table 2-1 contains the various data formats processed by FPMP-11 conversion routines:

a tillean Die seine		TABLE 2-1
	en de la de	DATA FORMATS
	一种 中部学校	Dilin I Oldmid

æ,

C

CODE	INTERNAL FORM	EXTERNAL INPUT FORM	EXTERNAL OUTPUT FORM
D	Double Precision	Decimal number with or without a decimal point or exponent field.	Decimal number with a D exponent field and a decimal point.
E	Single Precision	Decimal number with or without a decimal point or exponent field	Decimal number with an E exponent field and a decimal point.
F	Single Precision	Decimal number with or without a decimal point or exponent field	Decimal number with a decimal point
 	Single Precision	Decimal number with or without a decimal point or exponent field.	Decimal number with a decimal point and with or without an E exponent field (see table 2-2)
I	Integer	Decimal number with- out a decimal point or exponent	Decimal number without a decimal point or exponent
0	Integer	Octal number	Octal number

The following FPMP-11 routines perform the above conversions:

\$DCI \$DCO	D conversion for input D conversion for output
\$RCI	E,F, and G conversion for input
\$ECO	E conversion for output
\$FCO	F conversion for output
\$GCO	G conversion for output
\$ICI	I conversion for input

ŞICO	I	conversion	for	output
ŞOCI	0	conversion	for	input
\$OCO	0	conversion	for	output

Each of these routines requires one or more of the following parameters:

- w The width of the ASCII field in characters. The field width w of all output conversions should always be large enough to include spaces for the decimal point, sign, and exponent. In all such conversions, if w is not large enough to accomodate the converted number, asterisks are placed in the ASCII field.
- d The decimal position:
 - a) on input, the decimal point is assumed to be d digits from the right hand end of the ASCII field, if no explicit decimal point is found.
 - b) on output, d digits appear to the right of the decimal point.

p the scale factor:

- a) for F type conversion, (ASCII number)=(internal no.) *10^(scale factor)
- b) for D and E type conversions, the scale factor multiplies the fraction by a power of ten, but the exponent is adjusted, leaving the number unchanged except in form.
- c) for G type conversions, the scale factor is not used unless the magnitude of the number is such that E format is used.
- d) In all input operations, the scale factor is not used if there is an exponent in the external field.

NOTE

Input conversion routines handle all blanks as zeros. For example, 3.0E2 in a six character field would be considered to be 3.0E20.

TABLE 2-2

G-TYPE OUTPUT CONVERSIONS

Routine \$GCO is called with parameters p=P, w=W, and d=D (where P, W, D are integer constants):

Magnitude of data	Resulting Conversion	
0.1 <u><</u> M<1 1 <u><</u> M<10	F-type with p=0,w=W-4 and d=D F-type with p=0,w=W-4 and d=D-1.	
$\begin{array}{c} 10^{D-2} < M < 10^{D-1} \\ 10^{D-1} < M < 10^{D} \\ All \text{ others} \end{array}$	F-type with p=0,w=W-4 and d=1. F-type with p=0,w=W-4 and d=0. E-type with p=P,w=W, and d=D.	-

Examples:

The following internal numbers are shown converted according to various format parameters (b=blank):

(A) ONE-WORD INTEGERS:

INTERNAL	I(w=5)	I(w=7)	O(w=10)
<u>NUMBER</u> (Decimal)	Gunna de la constantion de la		
5	bbbb5	bbbbbb5	bbbbbbbbb5
10	bbb10	bbbbb10	bbbbbbbbl2
-23	bb-23	bbbb-23	bbbbbbb-27
0	bbbb0	bbbbbb0	bbbbbbbbbb
123,456	* * * * *	bl23456	bbbb361100

(B) TWO-WORD FLOATING POINT:

INTERNAL NO.	(P=0) E(w=10,d=2)	F(w=10, d=2)	G(w=10,d=2)
0	bb0.00E 00	bbbbbb0.00	bb0.00bbbb
1	bb0.10E 01	bbbbbb1.00	bbl.00bbbb
-1	b-0.10E 01	bbbbb-1.00	b-1.00bbbb
0.1	bb0.10E 00	bbbbbb0.10	bb0.10bbbb
555	bb0.55E 03	bbbb555.00	bb0.55E 03
0.001	bb0.10E-02	bbbbbb0.00	bb0.10E-02

----- (P=1) -----

0	bb0.00E-01	bbbbbb0.00	bb0.00bbbb
1	bb1.00E 00	bbbbb10.00	bbl.00bbbb
0.1	bb1.00E-01	bbbbbbl.00	bb0.10bbbb

(C) FOUR-WORD FLOATING POINT:

D-type conversion is the only one available for 4-word floating-point numbers. It is similar to E format except that the exponent part prints with a D instead of an E.

CHAPTER 3

USING FPMP-11

The user program can access the FPMP-11 routines by TRAP instruction and/or direct call of the routine. (For information on writing a user program, refer to the Papertape Software Programming Handbook.) The TRAP handler saves and restores the contents of the PDP-11 general registers. The OTS routines normally do not. All FPMP-11 entry points used by the program must be declared with a .GLOBL assembler directive in the user program. (The entry points are listed in Appendix D.) To include user floating point error routines, initialize the global location \$ERVEC as described in section 3.4.

3.1 USING THE TRAP HANDLER WITH FPMP-11

In order to simplify use of the various OTS routines, a TRAP handler is included in the FPMP-ll package. If TRAP calls are being used, the user program must initialize the TRAP vector at location 348. The TRAP vector can be initialized by putting the following code in the user program.

.GLOBL TRAPH MOV #TRAPH,@#34 MOV #340,@#36

;address of TRAP handler ;set priority of operation

The TRAP handler, TRAPH, uses software to simulate a floating-point accumulator (FLAC). The FLAC is a pseudo-register which is the implicit destination address of every trapped operation. Operations can be performed on the FLAC by issuing coded TRAP instructions in the user program. In addition to being used with the OTS functions, items can be loaded into and stored from the FLAC.

The FLAC is maintained by the TRAP handler in double precision format; however, it is important to note that single precision operations (e.g. \$ADR or SQRT) destroy the contents of the two lowest order words of the FLAC. In particular, these two words are not set to zero. This means that a single precision function can operate on the FLAC while it contains either a single or double precision number, but the result will be single precision and should not be operated on by the double precision routines. A number can be explicitly converted between the single and double precision formats by the FPMP-11 routines \$RD and \$DR which convert single to double and double to single respectively. These routines, \$RD and \$DR, can not be called via the TRAP handler. Because it contains the floating accumulator, the TRAP handler of FPMP-11 is not re-entrant. For this reason, care must be exercised if the TRAP handler is to be called both in a main program and in an interrupt-driven subroutine. To call the TRAP handler to perform floating point operations within an interrupt-driven subroutine, the contents of the FLAC should be pushed onto the processor stack before any other TRAP calls are executed. The FLAC can be pushed onto the stack by executing the instruction "TRAP 73". After all TRAP calls have been completed by the interrupt-driven subroutine, and before returning from the interrupt, the FLAC must be restored from the stack (it must be at the top of the stack) by executing the instruction "TRAP 71". If the double precision routines are being used, the traps are TRAP 74 and TRAP 72 respectively.

Addressing Modes Available in TRAP Calls:

3.1.1 Stack Mode

The operand is considered to be on the top of the R6 stack. (R6 is General Register 6) The operand is popped off for use. (exception: STR and STD push the FLAC onto the stack.)

3.1.2 @R0 Mode

General Register 0 points to the operand. Register 0 is not changed by FPMP-11.

3.1.3 Immediate Mode

The operand immediately follows the TRAP instruction in the next two or four words depending on whether the operation is single or double precision.

3.1.4 Relative Mode

The address of the operand, relative to the PC, immediately follows the TRAP instruction. For example to address an operand at location A, code the word following the TRAP as .WORD A-.

EXAMPLE:

1010 is internally coded as:

0	10000100	01	00000	000000	0000000000
15	14 7	6	0	15	0

which is 041040 000000 (octal). To add 10_{10} to the FLAC in each of the four modes (single precision): Stack Mode: MOV #000000,-(SP) ; PUSH FLOATING MOV #041040,-(SP) ;10 ONTO THE STACK TRAP ADR+STACKM ; ADD TO FLAC Symbols ADR (for single precision add) and STACKM (for stack mode) are assigned values 12_8 and 0_8 respectively. (Refer to page 17.) @R0 Mode: MOV #TEN, RO ;GET ADDRESS OF OPERAND IN RO ; ADD TO FLAC TRAP ADR+ARM TEN: .WORD 041040,000000 ;FLOATING POINT TEN Symbols ADR and ARM (@R0 mode) are assigned the values 128 and 1008 respectively. Immediate Mode: TRAP ADR+IMMEDM ;ADD TO FLAC .WORD 041040,000000 ;FLOATING POINT TEN Symbols ADR and IMMEDM (immediate mode) equal 128 2008 and respectively.

¢

21

Relative Mode:

• TRAP ADR+RELM •WORD TEN-•

;ADD TO FLAC ;RELATIVE ADDRESS OF OPERAND

TEN: .WORD 041040,000000

FLOATING POINT TEN

Symbols ADR and RELM (relative mode) equal 12 and 300 respectively.

To perform the above operations in double precision, use $ADD=14_8$ instead of ADR, and extend the floating point ten with two more words of zeros (i.e. TEN: .WORD 041040,0,0,0; double precision floating-point ten).

The source form of a TRAP call is:

TRAP num + mode

where num is the number of the OTS routine to be called (refer to Appendix D for the OTS routine numbers), and mode is one of the following addressing modes:

Mode

0	Stack mode
1008	@R0 Mode
2008	Immediate mode
3008	Relative mode

The binary form of the TRAP instruction is:

WORD:	
10001001	mmrrrrrr
15	0

Where

mm = addressing mode bits (00 = Stack mode, 01 = @R0 mode, 10 =
Immediate mode, 11 = Relative mode)
rrrrrr = OTS routine number

It is suggested that commonly used addressing modes and routine numbers be referenced symbolically. For instance, the statements

STACKM=0	;STACK MODE
ARM=100	; @RO MODE
IMMEDM=200	; IMMEDIATE MODE
RELM=300	; RELATIVE MODE

ADR=12	;SINGLE	PRECISION	ADD ROUTINE
SBR=13	;SINGLE	PRECISION	SUBTRACT
MLR=21	;SINGLE	PRECISION	MULTIPLY
DVR=25	;SINGLE	PRECISION	DIVIDE

allow TRAP calls to be coded as follows:

TRAP	ADR+RELM	; ADD IN RI	ELAJ	IVE	MODE	
TRAP	MLR+ARM	; MULTIPLY	IN	@R0	MODE	
TRAP	SBR+IMMEDM	; SUBTRACT	IN	IMMI	EDIATE	MODE

Note that single argument, single result functions, such as square root (SQRT) require no addressing (refer to Appendix D); the argument is taken from the FLAC and the result is stored back into the FLAC. Consequently, the addressing mode of a TRAP call to such a function is ignored by the TRAP handler, and no address is used.

The TRAP handler sets the condition codes to reflect the contents of the FLAC after every operation except a compare. After any operation except floating point compare, the condition bits are set as follows:

Condi	tion	Cod	es

FLAC	N	<u>Z</u>	<u>v</u>	C
<0	1	0	0	0
=0	0	1	0	0
>0	0	0	0	0

After a floating point compare (either single or double precision), the condition codes are set as follows:

FLAC <opr< th=""><th>1</th><th>0</th><th>0</th><th>0</th></opr<>	1	0	0	0
FLAC=OPR	0	1	0	. 0
FLAC>OPR	0	0	0	0

where OPR is the operand addressed by the TRAP compare instruction.

EXAMPLE:

e.

¢

A

To calculate

 $X = \frac{-B + SQRT (B * B - 4 * A * C)}{2 * A}$

the following program might be written:

;LOAD A INTO FLAC TRAP LDR+RELM .WORD A-. ; RELATIVE ADDRESS OF A TRAP MLR+IMM ;MULTIPLY BY 2.0 FTWO: .WORD 040400,0 ;CONSTANT 2.0 TRAP STR+RELM ;STORE FLAC IN TEMP1 ; RELATIVE ADDRESS OF TEMP1 .WORD TEMP1-. ;MPY BY 2.0 TO GET 4*A TRAP MLR+RELM .WORD FTWO-. ;MPY BY C TRAP MLR+RELM .WORD C-. TRAP STR+RELM ;STORE FLAC IN TEMP2 .WORD TEMP2-. ;GET ADDRESS OF B INTO RO MOV #B,RO ;LOAD B INTO FLAC (@RO MODE) TRAP LDR+ARM TRAP MLR+ARM ;CALCULATE B*B TRAP SBR+RELM ;SUBTRACT 4*A*C (IN TEMP2) .WORD TEMP2-. TRAP SQRT ;CALC SQUARE ROOT OF FLAC, ;NO ADDRESSING REQUIRED TRAP SBR+ARM ;ADD MINUS B ; DIVIDE BY 2.0*A IN TEMP1 TRAP DVR+RELM .WORD TEMP1-. TRAP STR+RELM STORE FLAC INTO X .WORD X-. A: .WORD 040400,0 B: .WORD 040640,0 C: .WORD 037600,0 X: .=.+4 TEMP1: .=.+4

;VALUE OF A (2.0) ;VALUE OF B (5.0) ; VALUE OF C (0.25) ;LOCATION FOR RESULT ; TEMPORARY ; TEMPORARY

The above example assumes that the TRAP vector (location 34_{0}) has been initialized as previously described.

3.2 ACCESSING USER ROUTINES VIA THE TRAP HANDLER

Special floating-point functions may be coded as assembly language subroutines and accessed via TRAP calls if one of the following calling conventions is used:

POLISH - receive two arguments, 1. either single or double precision, on the stack, and return one result, of the same precision as the arguments, on the stack. Return must be via a

JMP @(R4)+

TEMP2: .=.+4

2. J5RR - The user routine should be expecting a call of the following form:

	R5,subr	;jump to	subroutine	9
.WOR	D arg	;single	argument's	address

arg is the symbolic address of the subroutine's single or double precision argument. Note that the instruction following the JSR is not necessarily a BR. The returned result in registers RO-R3 is stored in the FLAC by the TRAP handler.

Furthermore, user routines to be called by the TRAP handler must reside within the 8K words physically following the beginning of the TRAP handler in memory, and an entry must be made in the TRAP handler's dispatch table. The dispatch table called TBL\$42 in TRAPH, is organized as follows:

- 1. There is a one word entry corresponding to each routine-number which can be coded in a TRAP call (total of 64 words).
- 2. The position of the word in the table corresponds to the routine-number which calls it (e.g. the word at location TBL\$42 is referenced by "TRAP 0+mode", while the word at location TBL\$42+10. is referenced by "TRAP 5+mode"). In general, the word at location TBL\$42+2n is referenced by the call TRAP n+mode.
- 3. The word at each table location is coded as follows:
 - a. 0-indicates no routine corresponds to this table entry.

0

b.

15

A:

flags relative address

13

In J5RR mode, TRAPH supplies the FLAC as the single argument and stores the result back into the FLAC. No explicit address is accepted in the TRAP instruction. In POLISH mode, TRAPH uses the FLAC as one argument and the location addressed in the TRAP call as the second. The result is stored in the FLAC. Refer to section 3.1 for addressing modes in TRAP calls to FPMP-11.

3.3 DIRECT CALLS TO OTS ROUTINES

Occasionally it is desirable to call OTS routines directly. For instance, some routines cannot be accessed using the TRAP handler (refer to Appendix D). Furthermore, eliminating the TRAP handler overhead decreases the execution time of the user program. Note that when called directly, the OTS routines do not preserve the contents of the general registers, nor do they in general set the condition codes to reflect the result of the operation, these functions are performed by the TRAP handler when it is used.

Any of the OTS routines can be directly called by using its FPMP-11 global entry point and observing the proper calling conventions. Calling conventions fall into a few basic types as follows (the calling conventions for each routine are given in Appendix D):

3.3.1 Polish Mode

Polish mode calls are designed to be most effective in a compiler-generated environment. They are easily produced by a compiler and are particularly efficient in storage space used and interpretation overhead.

The routines that are called with Polish mode are:

Name	No. of Arguments	Location of Result
\$ADD	2	4 word sum on top of stack
\$ADR	2	2 word sum on top of stack
\$CMD	2	sets condition codes
\$CMR	2	sets condition codes
\$DINT	1	integer result on top stack
\$DR	1	2 word result on top of stack
\$DVD	2	4 word quotient on top of stack
\$DVI	2	integer quotient on top of stack
\$DVR	2	2 word quotient on top of stack
\$ID	1	4 word result on top of stack
\$IR	1	2 word result on top of stack
ŞINTR		result on top of stack
\$MLD	2	result on top of stack
ŞMLI	2	result on top of stack
ŞMLR	2	result on top of stack
\$NGD	1	result on top of stack
\$NGI	1	result on top of stack
\$NGR	1	result on top of stack
\$POPR5	4	result in registers R0-R3
\$POPR4	4	result in registers R0-R3
\$POPR3	2	result in registers R0,Rl
\$PSHR1	1	result on top of stack
\$PSHR2	1	result on top of stack
\$PSHR3	2	result on top of stack
\$PSHR4	4	result on top of stack

Name	No. of Arguments	Location of Result
\$PSHR5	4	result on top of stack
\$RD	1	result on top of stack
\$DI	1	result on top of stack
\$RI	1	result on top of stack
\$SBD	2	result on top of stack
\$SBR	2	result on top of stack

Each routine called in Polish mode pops the necessary arguments off the R6 (General Register 6) stack and pushes the final result onto the Multi-word arguments are always pushed onto the stack stack. low-order word first, so that the highest-order word (the one containing the sign and exponent) remains on top of the stack (@SP).

Arguments must be pushed onto the stack before entering Polish mode so that the source operand is on the top of the stack and the destination operand is next down from the top.

Polish mode is entered with a JSR in the form

JSR R4, \$POLSH

where \$POLSH is a global subroutine in FPMP-11.

Routines to be used are then called by supplying a word with the address of the routine.

.WORD \$ADR

Exit from Polish mode is by coding a word containing the address of the next instruction to be executed. For example to execute the next instruction in sequence,

.WORD .+2

Using Polish mode, coding to calculate (A+B)*C with the single precision routines might be written as:

.GLOBL \$POLSH, \$ADR, \$MLR

MOV $C+2, -(SP)$; PUSH C ONTO STACK.
MOV $C_{,-}(SP)$ MOV $B+2_{,-}(SP)$	PUSH B ONTO STACK.
MOV B,-(SP)	• • • • • • • • • • • • • • • • • • • •
MOV $A+2,-(SP)$; PUSH A ONTO STACK.
MOV A,-(SP)	
JSR R4, SPOLSH	;ENTER POLISH MODE
.WORD \$ADR	; ADD A TO B AND LEAVE
	; THE RESULT ON THE STACK
.WORD \$MLR	;MULTIPLY PREVIOUS SUM BY
	;C AND LEAVE RESULT ON STACK.
.WORD .+2	;LEAVE POLISH MODE.
•	

After execution of the above code, the result of the calculation (A+B)*C is on the top of the R6 stack.

The routine "\$POLSH" that causes entry into Polish mode is located at global entry \$POLSH in FPMP-11. It is coded as follows:

	GLOBL	\$POLSH	
\$POLSH:	TST (SP)+		; DELETE OLD VALUE OF R4 FROM
			; THE TOP OF THE STACK.
and the second sec	JMP @(R4)·	+	;ENTER POLISH MODE.

Each routine called in Polish mode takes its operands from the top of the stack and pushes its result, if any, back onto the stack. Each routine returns with a "JMP @(R4)+" which passes control to the next routine in sequence. User routines can be written and called in Polish mode provided they preserve the contents of R4 and return by executing a "JMP @(R4)+". The following is an example of a user subroutine written for Polish calls.

DUP:	MOV	2(SP),-(SP)	; DUPLICATE STACK ITEM
	MOV	2(SP),-(SP)	; TWO WORD ITEM
	JMP	@(R4)+	; RETURN

When executed, this subroutine duplicates the two-word item on the top of the stack.

3.3.2 J5RR Mode

J5RR is the calling convention used by most of the FORTRAN library functions. J5RR mode calls are of the form

JSR R5, subroutine

All argument addresses are placed in a list following the subprogram call. The generalized standard sequence is:

• GLOBL	SUBR
JSR R5,	SUBR
BR XX	
A	
В	
•	
•	
•	
Z	
-	

XX:

where A, B...Z are argument addresses.

Subprograms are responsible for not altering the contents of register R5 since it is the parameter list pointer.

The results of subroutines called in J5RR mode are generally stored as follows: integer results are returned in R0, two-word floating point results in R0 and R1 and four-word results in R0-R3.

An example of a call in J5RR mode is this call to calculate the square root of X:

	.GLOBL SQRT		
	•		
	•		
	JSR R5, SQRT	;CALL TO SQRT ROUTINE	
	BR A •WORD X	;RETURN POINT ;ADDRESS OF ARGUMENT	
A:	• WORD A	CONTINUE PROGRAM	
	•		
Х:	.WORD 040400,000000	;2 WORD FLOATING POINT ;VALUE OF X=2.	NUMBER,

In this example, the result is returned as a two-word floating point number in R0-R1.

The functions which use J5RR mode calls are:

,

t

Function	# of Arguments	Register(s) for Result
ALOG	1	RO,R1
ALOGIO	1	R0,R1
AINT	1	R0,R1
ATAN	1	RO,RL
ATAN2	2	R0,R1
DBLE	1	R0-R3
DLOG	1	R0,R3
DLOG10	1	R0, R3
DCOS	1 by the first second second	R0, R3
DSIN	1 • • • • • • • • • • • • • • • • • • •	R0-R3
DSQRT	1	R0-R3
DATAN	1	R0-R3
DATAN2	2	R0-R3
DEXP	1	R0-R3
EXP	1	R0,R1
FLOAT	1	R0,R1
IFIX	1	R0
IDINT	1	RÖ
INT	1	R0
SIN	ī	R0-R1
COS	1	R0-R1
SNGL	1	RO,RL
TANH	1	R0,R1
	_	· · · · · · · · · · · · · · · · · · ·

3.3.3 JPC Mode

The JPC mode of subroutine call is used for communicating with the ASCII conversion routines in FPMP-11. With JPC mode, the arguments must be pushed onto the stack before the subroutine is called. The call to each individual subroutine is listed in Table 3-1. In general, a JPC mode call is coded as follows:

MOV R3,-(SP) ;push first argument onto stack . . MOV #ARG,-(SP) ;push last argument onto stack JSR PC,subr ;call subroutine .

For example, to convert a ten character ASCII field at location BUFFER to internal single precision format, the following might be coded:

• • MOV #BUFFER,-(SP) ;PUSH ADDRESS OF FIELD MOV #10.,-(SP) ;PUSH LENGTH OF FIELD CLR -(SP) ;D-SCALE IS ZERO CLR -(SP) ;P-SCALE IS ZERO JSR PC,\$RCI ;CALL CONVERSION ROUTINE •

After the above code is executed, the internal representation of the number at location BUFFER is in the top two words of the stack. The ten characters at location BUFFER can be read from an I/O device, or coded as constants: For example,

BUFFER: .ASCII /113.25bbbb/ BUFFER: .ASCII /-3.627E+09/

or

3-12

TABLE 3-1

ROUTINES WHICH USE THE JPC MODE OF CALL

Name	Description	# of Arg	Call Format	Location Of Result
\$DCI	ASCII to dbl. prec.	4	Push addr. of start of ASCII field Push length of ASCII field in bytes Push format scale D (from W.D) position of assumed decimal point Push P format scale JSR PC,\$DCI	4 word result on top of stack
\$DCO	Dbl. prec. to ASCII	4	<pre>Push addr. of start of ASCII field Push length of ASCII field in bytes Push D part of W.D (position of decimal point) Push P scale Push 4 word value to be converted lowest order word first JSR PC,\$DCO</pre>	ASCII field specified
\$ECO	Single prec. to ASCII E format	4	Same calling sequence as \$DCO except that a 2 word value is to be converted. JSR PC,\$ECO	ASCII field Specified
\$FCO	Single prec. to ASCII F format	5	Same calling sequence as \$ECO. JSR PC,\$FCO	ASCII field specified
\$GCO	Single prec. to ASCII G format	5	Same calling as \$ECO. JSR PC,\$GCO	ASCII field specified.
\$ICI	ASCII to integer	2	Push addr. of start of ASCII field Push length in bytes of ASCII field JSR PC,\$ICI	Integer result on top of stack
\$ICO	Integer to ASCII	3	Push addr. of ASCII field Push length of ASCII field in bytes Push integer value to be converted JSR PC,\$ICO	ASCII field specified

TABLE 3-1 (Cont.)

ROUTINES WHICH USE THE JPC MODE OF CALL

Name	Description	# of Arg.	Call Format	Location Of Result
\$OCI	ASCII to octal	3	Same calling sequence as \$ICI JSR PC,\$OCI	Top of stack
\$0C0	Octal to ASCII	3	Same calling sequence as \$ICO JSR PC,\$OCO	ASCII field specified
\$RCI	ASCII to single prec.	4	Same calling sequence as \$DCI JSR PC,\$RCI	Two word result on top of stack

The ASCII input conversion subroutines \$RCI, \$DCI, \$ICI, and \$OCI preserve the contents of the general registers and restore them to their original values before returning. The ASCII output conversion subroutines \$DCO, \$ECO, \$FCO, \$GCO, \$ICO, and \$OCO destroy the contents of general registers R0, R1, R2, and R3, but preserve the contents of R4 and R5.

Errors detected by the ASCII input conversion subroutines \$RCI, \$DCI, \$ICI, and \$OCI cause the subroutine to return with a zero result and with the C bit set in the condition codes.

3.4 ERRORS

1.0

All errors in floating-point operations, such as overflow of the FLAC or an illegal TRAP instruction, are handled by the routines \$ERR and \$ERRA. These routines save the contents of R0, and load the error code into R0. The routines then perform a JSR PC,@\$ERVEC. \$ERVEC is a global location which is initially set to contain the address of a HALT instruction but can contain the address of a user error handling routine. If the user error handling routine is to be used, code is inserted in the initialization of the program as explained in section 3.4.1.

The error code generated by the FPMP-ll subroutine is put in R0 in the following format:

R0:					
error number			error	class	
15	8	7			0

Error codes and their meanings are shown in Table 3-2.

3.4.1 User Error Handling Routines

To include a user error handling routine in a program, the following code must be included in the initialization of the program.

.GLOBL \$ERVEC MOV #ERROR,\$ERVEC ;move address of error routine

ERROR: user's error handling routine

The error handling routine can be written to terminate with a HALT instruction or, if registers 1 through 5 are saved, to continue the program by executing an RTS PC instruction. The only exception is the halt after an illegal TRAP instruction (error 0,0) from which it is impossible to continue. If such a TRAP occurs, its address is in Rl when the error routine is called.

TABLE 3-2

FPMP-11 ERROR CODES

ERROR CODE		
(CLASS, #)	ISSUED BY	EXPLANATION
0,0	TRAPH	Illegal TRAP instruction. Rl points to the TRAP instr.
3,1	\$ADD	Expon. overflow in double prec. addition
3,2	\$ADR	Exponent overflow in real addition
3,3	\$DVD	Double prec. div. by zero
3,4	\$DVD	Expon. overflow in double precision division
3,5	\$DVI	Integer division by 0
3,6	\$DVR	Expon. overflow in real division
3,8	\$DVR \$MLD	Real division by zero Expon. overflow in double prec. mult.
3,10 3,11	ŞMLD ŞNEG	Exponent overflow in double prec. mult.
3,12	SMLR	Exponent overflow during negation Expon. overflow in real multiplication
3,14	ŞMLI	Product outside of range on integer mult.
3,22	\$RI	Real outside range on real to integer
-,	+	conversion
3,23	\$DR	Exponent overflow on double to real conversion
4,2	DEXP	DEXP argument greater than 87
4,3	DLOG	DLOG argument less than or equal to zero
4,4	DSQRT	DSQRT argument less than zero
4,5	EXP	EXP argument greater than 87
4,10	ALOG	ALOG argument less than or equal to zero
4,11	SQRT	SQRT argument less than zero
4,12	SNGL	SNGL exponent overflow in round
5,1	\$ADD	Expon. underflow in double prec. addition (warning)
5,2	\$ADR	Exponent underflow in real addition (warning)
5,3	\$DVR	Expon. underflow in real div. (warning)
5,4	DEXP	DEXP argument less than -88.7 (warning)
5,5	EXP	EXP argument less than -88.7 (warning)
5,6	\$MLD	Expon. underflow in double prec. mult. (warning)
5,7	\$MLR	Expon. underflow in real multiplication
5,8	\$DVD	Expon. underflow in double prec. division
		(warning)

3.5 CREATING SPECIAL PACKAGES

FPMP-11 source code includes PAL-11S conditional assembly instructions which allow tailoring of the FPMP-11 package to include only the functions required by the user program. (Refer to the PAL-11S manual (DEC-11-YRWB-D) for information on conditional assembly instructions.) The desired routines are then assembled to take advantage of whatever hardware features are available.

3.5.1 Assembly Switch Tape

To take advantage of the conditional assembly instructions in the FPMP-11 source code, a separate tape which sets the switches of the desired routines and hardware must be prepared and included in the assembly of the FPMP-11 package.

The switches are set by statements which assign a value to the switch name. For example, to indicate the availability of the 11/45 FPU hardware, the FPU switch is set with the following statement

FPU=1

n

When the FPU switch is set, many FPMP-ll routines assemble differently to take advantage of the FPU.

When using the PDP-11/45 FPU option, it is the user's responsibility to set up the FPU TRAP vector (location 244_8) and the FPU status register (refer to the PDP-11/45 Processor Handbook). Refer to Table 3-3 for hardware switch option names.

Significant size and speed advantages can be expected if one of the hardware options is present and its corresponding switch is set. If no hardware option switch is set the assembler assumes the program uses the basic PDP-11 instruction set. In no case should more than one hardware option switch be set during an assembly.

TABLE 3-3

HARDWARE OPTION SWITCHES

Switch Name	Hardware Option	
FPU EAE MULDIV	PDP-11/45 floating point unit PDP-11/20 EAE PDP-11-40 extended instruction set (EIS) PDP-11/45 processor	or

NOTE

If the FPU switch is set during an assembly, the assembler being used must be capable of processing the extended op codes which will appear. The present version (V002A) of PAL-11S does not support these op codes. MACRO-11 can be used for assembly when the FPU switch is set.

Each section of code in the FPMP-ll package is assigned a number and the switch to cause a particular section of code to be included is called CND\$n.

Table 3-4 lists the sections of the FPMP-ll package, the routines contained in each section and the switch name to be used.

For example, to include the DSQRT routine in the package set the switch with the following code:

CND\$14=1 .EOT
TABLE 3-4

CONDITIONAL FPMP-11 ASSEMBLY CODES

34

43

Þ

		Subroutine Contained
1	CND\$1	\$ADD,\$SBD
2	CND\$2	\$ADR,\$SBR
3	CND\$3	ALOG, ALOG10
4	CND\$4	AINT, \$INTR
5	CND\$5	\$CMD
6	CND\$6	\$CMR
7	CND\$7	DBLE
8	CND\$8	\$DCI,\$RCI
9	CND\$9	\$DCO,\$ECO,\$FCO,\$GCO
10	CND\$10	DLOG, DLOG10
11	CND\$11	\$DINT
12	CND\$12	\$DR
13	CND\$13	DSIN, DCOS
14	CND\$14	DSQRT
15	CND\$15	DATAN, DATAN2
16	CND\$16	\$DVD
17	CND\$17	\$DVI
18	CND\$18	\$DVR
19	CND\$19	DEXP
20	CND\$20	EXP
21	CND\$21	\$FCALL
22	CND\$22	IFIX
23	CND\$23	FLOAT
24	CND\$24	\$ICI,\$OCI
25	CND\$25	\$ICO,\$OCO
26	CND\$26	INT, IDINT
27	CND\$27	\$ID,\$IR
28	CND\$28	SMLD
29	CND\$29	\$MLI
30	CND\$30	\$MLR
31	CND\$31	\$NGI, \$NGR, \$NGD
32	CND\$32	\$PSHR5, \$PSHR4, \$PSHR3, \$PSHR2, \$PSHR1
33		\$POPR5, \$POPR4, \$POPR3
34	CND\$33	\$RD
	CND\$34	\$RI,\$DI
35	CND\$35	
36	CND\$36	SNGL
37	CND\$37	SIN,COS
38	CND\$38	TANH
39	CND\$39	ATAN, ATAN 2
40	CND\$40	<pre>\$POLSH(switch is always set)</pre>
41	CND\$41	SQRT
42	CND\$42	TRAPH
43	CND\$43	<pre>\$ERR,\$ERRA(switch is always set)</pre>
	and the second	
	1	1

The CLASS5 switch can be set (CLASS5=1) to have class 5 (warning) messages interpreted by the error handler of FPMP-11. Normally class 5 errors are ignored. Many of the FPMP-11 transcendental and trigonometric functions do not operate properly if the class 5 switch is set.

There are two additional switches which work together with the others. When these switches are set the standard single or double precision TRAP handler packages are assembled. The two switches are:

SINGLEAssemble the standard single precision (2 word)
package when setDOUBLEAssemble the standard double precision (4 word)
package when set.

The contents of the standard packages are listed in Chapter 2. The SINGLE and DOUBLE switches may be set together to produce a combined package containing both standard packages. It is also possible to include a few double precision subroutines with the standard single precision package or to include some of the non-standard routines (e.g. integer multiply), with the single and/or double precision package. More information on creating these special combinations is given in section 3.5.1.1.

3.5.1.1 Preparing the Assembly Switch Tape

To assemble the FPMP-ll source tape:

- 1. Decide which FPMP-11 routines are to be included in the resulting package. Refer to Appendix D for a list of available routines.
- 2. Obtain the switch names for the desired routines from Table 3-4.
- 3. Decide which, if any, of the hardware option switches is to be set.
- 4. Create a paper tape or source file (either off-line or using the editor) in the following format; (Refer to the Paper Tape Software Programming Handbook for information on using the editor).

switch-name-1 =1 ;FIRST SWITCH TO BE SET
switch-name-2 = 1
.
.
.
switch-name-n = 1 ;LAST SWITCH TO BE SET
.EOT

(Where "switch-name-1" thru "switch-name-n" are the names of the switches to be set.) If preparing the tape off line, be sure to

put a carriage return/line feed after each line. For example, to assemble the standard single precision package to take advantage of the EAE, create the following tape:

SINGLE=1	;USE STANDARD 2-WORD PKG.	
EAE=1	;SPECIFY EAE	
• EOT		

To assemble a standard double precision package plus integer multiply and divide, create the following tape:

DOUBLE = 1 ;GET STD 4-WORD PKG. CND\$17 = 1 ;INTEGER DIVIDE CND\$29 = 1 ;INTEGER MULTIPLY .EOT

It is not necessary to worry about interdependency among FPMP-11 routines. For example, to create a package containing only the single precision function TANH, the tape

CND\$38 = 1 ;TANH .EOT

is sufficient. The fact that the TANH function calls the arithmetic routines and other internal functions is resolved by the FPMP-11 source code. In particular, the above switch being set causes the following routines to be included; TANH, EXP, \$ADR, \$SBR, \$MLR, \$DVR, \$FCALL, \$POLSH, \$PSHR3, \$ERR, \$IR, and \$RI.

- 5. Assemble the FPMP-11 package with PAL-11S loading the FPMP-11 source tapes (1 thru 6) last. Refer to Appendix B.
- 6. The object module produced by PAL-11S can now be used as described in section 3.6.

NOTE

Because of limitations in the symbol table size in the 8K version (V002A) of PAL-11S, it is not possible to include all FPMP-11 routines in a single assembly. The error message produced by the assembler is "S" and the assembly is aborted. It is possible however, to assemble as much as the standard single and double precision packages together. If the integer and conversion routines not included in the standard packages are needed along with both standard packages, they can be assembled separately by PAL-11S, and the resulting tape then linked with the standard packages using the LINK-11S linker. If this procedure is used, the linker produces error messages because of the multiple occurrence of the labels \$POLSH, \$V20A, \$ERR, and \$ERRA. These are non-fatal errors and can be ignored.

3.6 LOADING INSTRUCTIONS

The FPMP-ll package can be used as distributed by linking the object tapes (single or double precision) with the user object program or by using the source tapes to assemble a user-tailored package and then linking the package to the user program.

The Bootstrap and Absolute Loaders must be resident in core before any of the other programs can be loaded. Refer to Appendix A for loading instructions.

The object tape of the user program produced by PAL-11S (or DOS MACRO-11), and the FPMP-11 object tape are linked with LINK-11S (or DOS LINK-11) (refer to Appendix C for LINK-11S instructions). LINK-11S requires two passes and produces a tape called a load module which contains the user program and the FPMP-11 routines.

Use the Absolute Loader to load this module and execute the program. (Refer to Appendix A for details on using the ABS Loader.)

CHAPTER 4 SAMPLE PROGRAM

The following sample program illustrates most of the FPMP-11 modes of calls. Note that execution of this sample program requires the use of the Input/Output Executive (IOX) program which must be loaded before the sample program. This program inputs three F10.0 numbers, stores them as A,B and C and prints the numbers stored for verification. The roots of $AX^2+BX+C=\emptyset$ are calculated using the formula $X=-B\pm\sqrt{B^2-4AC}$. If $A=\emptyset$ the program halts.

		.TITLE	XAMPLE			
	000000	R0=%0	and a second			
	000001	R1=%1				
		R2=%2				
	000002					
	000003	R3=%3				
	000004	R4 = %4				
	000005	85=%5				
	000006	SP=%6				
	000007	PC=%7				
	000100	ARM=100				
	000200	IMM=200				
	000300	RELM=30				
	000071	LUR=71	***			
	000073	STR=73				
	000012	ADR=12				
	000013	SBR=13				
	000021	MLR=21				
	000025	DVR=25				
	000046	SQRT=46				
	000011	MSGLEN=				
	000002	RESET=2				
	000011	READOP=	11			
1 (A 4)	000004	WAITR=4		· · · · · · · · · · · · · · · · · · ·		
	000012	WRITE=1	2			
		.GLUBL	TRAPH, SRCI, SF	00		
000000	012706 BEGIN:	MOV	#2000, SP1	INITIALIZE	STACK	
~~~~~~	002000					
000004		IŬT				
000006	000000	WORD	0;	INIT THE IO	Y PACKAGE	
000010	002	BYTE	RESET,Ø	21121 1114 10	A LAUNAGE	
000011	000		NEVEIJE			
000012		IOT				
			****			
000014		WORD	TITLE			
000016	012	BYTE	WRITE,17	WRITE THE T	TIFE	
000017	001					
000050		MOV	#TRAPH,@#34;	INITIALIZE	TRAP VECTOR	
	000000					
	000034					
000056	012737	MOV	#340,@#36			
	000340					
	000036		Алар —			
000034	004767 RESTAR:	JSR	PC,READ;	READ ONE IN	PUT LINE INTO I	BUFFR
	000374					
000040	012701*	MOV	#BUFFR+6,R1;	GET ADDR OF	BEGINNING OF	BUFFER
	000646					
000044		MOV	#A, RØ;	GET ADDR OF	VAR CAR	
	000454	ar <b>wi</b> t All All All All All All All All All All	ar en <b>y 13.54 y</b>	Sector Carlord Md	s mint int	
	~~~~					

10

000050	010146 ILOOP:	MOV	R1,=(SP);	SAVE R1
000052		MOV	R0,-(SP);	SAVE RØ
		MOV	R1,=(SP);	PUSH ADDR OF ASCII STRING READ
000054				
000056		MOV	#10.,-(SP);	PUSH LENGTH
	000012			
200000	005046	CLR	-(SP);	D FORMAT SCALE
000064	005046	CLR	-(SP);	P SCALE
	004767	JSR	PC, SRCI;	CONVERT ONE NUMBER (F10,0)
0000000				an an is in the second s
	000000	and a second second		
000072		TRAP	LDR;	LOAD FLAC FROM TOP OF STACK
000074	104573	TRAP	STR+ARM;	STORE INTO VARIABLE A, B, OR C
000076	012600	MOV	(SP)+, RØ;	RESTORE RØ
000100		MOV	(SP)+,R1;	AND R1
		CMP	(RØ)+,(RØ)+;	INCR RØ BY 4
000102				
000104		AUD	#10,,R1;	INCR BUFFER POINTER TO NEXT VAR
	000012			
000110	012705'	MUV	#MSGBLK,R5	
	000567			
0000114	004767	JSR	PC, PRINT:	CALL PRINT SUBROUTINE
000114		000		
	000174			
000120	020027 *	CMP	RØ,#C;	LAST VAR?
	000464			
000124	101751	BLOS	ILCOP;	LOOP
000126		TRAP	LDR+RELM;	LOAD A INTO FLAC
000130		, WORD	A	RELATIVE ADDRESS OF A
000135		BEO	END;	EXIT IF $A = 0$
000134	104712	TRAP	ADR+RELM;	A + A TO GIVE 2*A
000136	000316	, WOKD	A	
000140	104773	TRAP	STR+RELM;	STORE 2*A INTO TEMP1
	000326	.WORD	TEMP1	and a second
-		TRAP		MPY BY 2 TO GET 4*A (IMMED MODE
	104621		MLR+IMM;	
	040400	.WORD	040400,000000;	CONST 2.0
000150	000000			
000152	104721	TRAP	MLR+RELM;	MPY BY C
	000310	, WORD	C	
	104773	TRAP	STR+RELM;	STORE 4*A*C IN TEMP2
				SJURG HAAAC IN ICHTE
	000314	.WORD	TEMP2	
000165	015100	MOV	#B,RØ;	GET ADDRESS OF VARIABLE "B"
	000460			
000166	104571	TRAP	LDR+ARM;	LOAD B INTO FLAC
	104521	TRAP	MLR+ARM;	MPY BY B TO GET B**2
	104713	TRAP	SBR+RELM;	SUBTRACT 4*A*C
000174	000300	.word	TEMP2-,	
000176	001430	BEQ	ROOT17	BRANCH IF ONLY ONE ROOT
000200	002441	BLT	IMAG;	B**2 - 4*A*C < 0 ???
	104446	TRAP	SORT:	TAKE SORT OF FLAC
				SAVE SORT (B**2-4*A*C) IN TEMP2
	104773	TRAP	STR+RELM;	ONTE OMATERAAETTAAAJUJ IN IEMEE
	000266	, WORD	TEMP2	
000210	104513	TRAP	SBR+ARM;	ADD MINUS B
000212	104725	TRAP	DVR+RELM;	DIVIDE BY 2*A (IN TEMP1)
000214	000254	, WOKD	TEMP1=.	
	012705"	MOV	#MSG1,R5;	ADDR OF "ROOT 1 = " MESSAGE
DEDETO		NO V	#11001 / K J /	Rook of Kool t - inconac
	000534			
000555	004767	JSR	PC, PRINT	CALL PRINT SUBROUTINE
	000066			
000226	104671	TRAP	LDR+IMM;	ZERO THE FLAC (IMMEDIATE MODE)
	000000 ZERO:		0,0	FLOATING POINT ZERO
		and the second sec	i i i i i i i i i i i i i i i i i i i	्राहर ब्रह्म स्वार्थ का प्राप्त के दिये हैं। सकल्पा किस्ता के प्राप्त कि कि स्वार्थ के प्राप्त के प्राप्त के प्
	000000	ins on a lin	600.4mM.	_ D
	104513	TRAP	SBR+ARM;	a ■ B a second state of a se
000236	104713	TRAP	SBR+RELMI	
000240	000234	.WORD	TEMP2	
	104725	TRAP	DVR+RELM;	DIVIDE BY 2*A
	000224	.WORD	TEMP1	
				ADDR OF "ROOT 2 = "
000246	012705*	MÜV	#MSG2,R5;	AUUR UP "KUUI E = "
	000545			

000252	004767	JSR	PC, PRINT	
	000036			
000256		BR	RESTARI	BRANCH TO GO AGAIN
	104771 ROOT1:	TRAP	LDR+RELMI	ZERO THE FLAC
000565		.WORD	ZERO-	
	*			GET - B
000264		100.00	SBR+ARM;	
000266		TRAP	DVR+RELM;	DIV BY 2*A
000270		.WOKD	TEMP1	
000515	012705	MOV	#MSG3,R5;	"ROOT = "
	000556			
000276	004767	JSR	PC, PRINT	
	000012		•	
000302		BR	RESTAR	
	000004 IMAG:	IOTI	Neo PAR	WRITE IMAGINARY ROOTS MESSAGE
		WORD	MCCAR	
	000600*		MSG4;	ADDRESS OF MESSAGE BUFFER
000310	012	.BYTE	WRITE,1;	WRITE TO SLOT 1
000311	001			
000312	000650	BR	RESTAR	
	7	PRINT SL	BROUTINE	
000314	010546 PRINT:	MOV	R5,=(SP);	SAVE REGS
000316		MOV	R4, - (SP)	
000320	··· •	MOV	R3,-(SP)	
000322		MOV	R2,=(SP)	
000324		MOV	R1,=(SP)	
000350	• •	MOV	R0,-(SP)	
000330		TRAP	STR+RELM;	STORE THE FLAC
000332	000146	.WOKD	TEMP3-	
000334	012704	MOV	#MSGLEN,R4;	CHAR COUNT FOR MESSAGE
	000011			
000340	012703*	MOV	#08UF+6,R31	ADDR OF OUTPUT FIELD
ныянты	000774			Head of Coller Trees
ON CALLAR A IN IN	112523 ML00P:	MOVE	(DE) - (DZ) + 1	MON THE CHAD NECOACE
			(R5)+,(R3)+;	MOV THE CHAR MESSAGE
000346		DEC	R4	
000350		BNE	MLOOP	
000352		MOV	R3,-(SP);	ADDR OF OUTPUT FIELD FOR CONV
000354	012746	MOV	#20.,-(SP);	LEN OFFIELD
	000024			
000360	012746	MOV	#10.,-(SP);	DECIMAL PLACES
	000012			
000364		CLR	-(SP);	P SCALE
000366		MOV		
000300		m ų v	TEMP3+2,-(SP);	PUSH VALUE IN DE CUNVERIEN
-	000110			
000372	016746	MQV	TEMP3,-(SP)	
	000102			
000376	004767 *	JSR	PC,SFCO;	CALL CONVERSION ROUTINE IN FPMP
	000000			
000402	000004	1011		CALL THE IOX PACKAGE
	000766	WORD	OBUF;	WRITE THE OUTPUT BUFFER
000406	012	BYTE	WRITE,1;	TO SLOT 1 (KB)
000407			······································	in one to the transformed and the second se
	001 0000000 HATTO	70 7 .	and the second sec	CALL TOY
	000004 WAITO:	IUT;	1.1 A 17 30 Mar -	CALL IOX
	000410	.WOKD	WAITO;	CREATE WAIT LOOP
000414	004	BYTE	WAITR,1;	WAIT FOR SLOT 1 (KB)
000415	001			
000416	012600	MOV	(SP)+,80;	RESTORE REGS

3

ъ

000420	012601		MOV	(SP)+,R1			
000422	012602		MOV	(SP)+,R2			
000424	012603		MUV	(SP)+,R3			
000426	012604		MOV	(SP)+,R4			
000430	012605		MOV	(SP)+,R5			- · ·
000432	000207		RTS	PC;	RETURN		
		1	READ SU	BROUTINE	n den son fan de service de la service de		
000434	000004	READ:	1077		CALL IOX FOR	READ	
000436	000640	 	, WORD	BUFFR;	ADDR OF INPU	BUFFER	
000440	011		BYTE	READOP,0;	READ SLOT Ø		
000441	000						
000442	000004	WAITI:	IOTI		CALL IOX		
000444	000442		.WOKD	WAITI;	CREATE WAIT	-00P	
000446	004	÷	BYTE	WAITR,0;	WAIT FOR SLO		
000447	000						
000450	000207		RTS	PC;	RETURN	•	
	000000	END:	HALT;	•	FINISHED		
	000000		.WORD	0,0			
000456				-,-			
	000000	в:	, WOKD	0,0	na sa		
000462			•	•		at in a superior of the	
	000000	C :	WOND	0,0			
000466							
	000000	TEMP1:	WOND	0,0			
000472		, 211, 20					
	000000	TEMP2:	.WORD	0,0			
000476							
	000000	TEMP3:	WORD	0,0	18. av.	in the second	
000502			• NOND	~ ~ ~			
	000055	TITLE	.WOND	18.			
000506		1. d. 1. la. hn. 4	WORD	Ø.			
000510			WORD	18.			
000512	015		. BYTE	15,12		and the state of the second	
000513	012			* 7115			ς t _a
000514	124		ASCII	/TEST OF FPMP1	1 /	1. A A A A A A A A A A A A A A A A A A A	
000515	105		. KOWLI	VILSI OF FFEF			
000516	123				1		
000517	124						
000520	040						
000521	117						
000255	106					the second second	
000523	040						
the fact the fe	106						
000524	120						
000526	115						
000527	120						
000530	061					• . · · · ·	
000531	061						
000532	015		BYIE	15,12		and the second s	
000533	012			•			
000534	155	MSG1:	ASUII	/ROOT 1 = /			
000535	117		*********	/ NUUI 1 • /			
000536	117						
000537	124						
000540	040						
51 51 81 51 51 MINU	42 m 42						

£,

000541	061			
000542	040			
000543	075			
000544	040			
000545	122	MSG2:	"ASCII	/ROOT 2 # /
000546	117			
000547	117			
000550	124			
000551	040			
000552	062			14 J.
000553	040			
000554	075			
000555	040			
000556	· 122	MSG31	ASCII	/ROOT = /
000557	117			/ (00) = /
000560	117			and the second
000561	124			
000562	040			
000563	075			
000564	040			
000565	040			
000566	040			
000567	040	MSGBLK:	.ASCII	/ /
000570	040	HODDER	******	
000571	040			
000572	040			
000573	040			
000574	040			
000575	040			
				A State of the second
000576	040			
	040 040		EVEN	
000576 000577	040 040 000600	MSG/J :	, ÉVÊN Word	1
000576 000577 000600	040 040 000600 000032	MSG4:	WORD	26.
000576 000577 000600 000602	040 040 000600 000032 000	MSG4:	77	26. Ø,Ø
000576 000577 000600 000602 000603	040 040 000600 000032 000 000	MSG4:	WORD BYTE	0,0
000576 000577 000600 000602 000603 000603 000604	040 040 000600 00032 000 000 000 000032	MSG4:	.WORD .BYTE .WORD	2,0 26.
000576 000577 000600 000602 000603 000604 000604	040 040 000600 00032 000 000 000032 015	MSG4:	WORD BYTE	0,0
000576 000577 000600 000602 000603 000603 000604 000604 000606	040 040 00600 000032 000 000 000032 015 012	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000603 000604 000604 000607 000610	040 040 000032 000 000 000 000 000 015 012 122	MSG4:	.WORD .BYTE .WORD	0,0 26. 15,12
000576 000577 000600 000602 000603 000603 000604 000604 000607 000610 000611	040 040 000032 000 000 000 000 00032 015 012 122 117	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000603 000604 000604 000607 000610 000611 000612	040 040 000600 000032 000 000032 015 012 122 117 117	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000603 000604 000604 000607 000610 000612 000613	040 040 000600 00032 000 000032 015 012 122 117 117 124	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000603 000604 000604 000607 000610 000611 000612 000613 000614	040 040 000600 00032 000 00032 015 012 122 117 117 124 123	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000603 000604 000604 000607 000610 000612 000613 000614 000615	040 040 000600 00032 000 000032 015 012 122 117 117 124 123 040	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000604 000607 000610 000612 000613 000614 000615 000616	040 040 000600 00032 000 00032 015 012 122 117 117 124 123 040 101	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000604 000607 000610 000612 000613 000613 000615 000616 000617	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000602 000602 000603 000604 000604 000607 000610 000612 000613 000613 000614 000615 000616 000617 000620	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000604 000607 000610 000612 000613 000613 000614 000615 000615 000616 000620 000621	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000604 000607 000610 000612 000613 000613 000614 000615 000615 000616 000621 000621	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040 111	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000604 000607 000610 000612 000613 000613 000614 000615 000615 000615 000620 000621 000623	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040 111 125	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000603 000604 000607 000610 000612 000613 000613 000614 000615 000615 000616 000621 000623 000624	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040 111 125 101	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000603 000604 000607 000610 000612 000613 000613 000614 000615 000615 000615 000621 000623 000623 000624 000625	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040 111 125 101 107	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000603 000604 000607 000610 000612 000612 000613 000613 000614 000615 000615 000621 000623 000623 000624 000625 000626	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040 111 125 101 107 111	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000602 000603 000604 000604 000607 000610 000612 000612 000613 000614 000615 000615 000616 000621 000623 000624 000625 000627	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040 111 125 040 111 125 101 107 111	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12
000576 000577 000600 000602 000603 000604 000603 000604 000607 000610 000612 000612 000613 000613 000614 000615 000615 000621 000623 000623 000624 000625 000626	040 040 000600 00032 000 00032 015 012 122 117 124 123 040 101 122 105 040 111 125 101 107 111	MSG4:	.WORD .BYTE .WORD .BYTE	0,0 26. 15,12

 $\mathcal{G}_{\mathcal{F}}$

D

4-5

ARY***/

000632	131				
000633	052				
000634	Ø52				
000635	052				and the second second
000636	015	BYTE	15,12		
000637	012	- · ·			
	000640	EVEN			
	the operation of the				
000640	000120 BUFFR:	.WORD	80.		
000642	000	BYTE	0,0		
000643	000	•••••			
000644		.WORD	Ø		
0000044	000766	.=.+80.	1		1 6 C
	000100				
000766	0000/00 00UE .	.WOKD	32.		
000766	000040 OBUF:	-			
000770	000	DYTE	0,0		
000771	000		7 0		
000772		.WORD	32.		
000774	040	ASUII			
000775	040				
000776	040				
000777	040				and the second
001000	040				
001001	040				
001002	040				
001003	040				
001004	040				
	001031	.=.+20.			
001031	015	BYTE	15,12,12		
001032	012				
001033	012				
	000000	.ENU	BEGIN		
A	000454R	ADK	= 000012	ARM	= 000100
8	000460R	BEGIN	000000R	BUFFR	000640R
Č	000464R	DVR	= 000025	END	000452R
ILOOP	000050R	IMAG	000304R	IMM	= 000200
LDR	= 000071	MLOOP	000344R	MLR	= 000021
MSGBLK	000567R	MSGLEN	= 000011	MSG1	000534R
MSG2	000545R	MSG3	000556R	MSG4	000600R
OBUE	000766R	PC	=%000007	PRINT	000314R
READ	000434R	READUP		RELM	= 000300
RESET	= 000002	RESTAR	000034R	ROOT1	000260R
		R1	=%000001	R2	=%000002
RØ	= % 0 0 0 0 0 0	R4	=%000004	RS	=%000005
R3 SBR	= % 0 0 0 0 0 3	SP	=%000000	SORT	= 000046
	= 000013	TEMP1	-2000000 000470R	TEMP2	000474R
STR	= 000073	TITLE	000504R	TRAPH	= ******
TEMP3	000500R	WAITÚ	000410R	WAITR	= 000004
WAITI	000442R	ZERO	000230R	SFCO	= ******
WRITE	= 000012	-	= 001034R		
SRCI	≈ ***** G	•	- UUIU34K		

4-6

G

G

;Teletype input in three ;10-character fields

;program verification
;of input

;result

TEST OF FPMP11 4.00 2•Ø 2.00000000000

4.0000000000

2.0000000000

ROOT =-1.00000000000

> 12.5 3.25 5.43 12.5000000000

> > 3.2500000000

5.4299998283

ROOTS ARE IMAGINARY***

2•Ø

ŝ

ಹ

3• E-Ø1	•06E002 40E-001 0•3000000119
	6• 00000000000
	4•0000000000
ROOT $1 =$	-0.6905062795
R00T 2 =	-19.3094921112
5	15 3 5•00000000000
	15.0000000000
	3.0000000000
ROOT 1 =	-0.2154767066
ROOT 2 =	-2.7845234871
Ø	4•0 3•75 0•0000000000
	4•0000000000

3.7500000000

PROGRAM OUTPUT

APPENDIX A

BOOTSTRAP AND ABSOLUTE LOADERS

A.1 THE BOOTSTRAP LOADER

A.1.1 Loading the Bootstrap Loader

The Bootstrap Loader should be toggled into the highest core memory bank.

xx7744	016701
xx7746	000026
xx7750	012702
xx7752	000352
xx7754	005211
xx7756	105711
xx7760	100376
xx7762	116162
xx7764	000002
xx7766	xx7400
xx7770	005267
xx7772	177756
xx7774	000765
xx7776	УУУУУУ

xx represents the highest available memory bank. For example, the first location of the loader would be one of the following, depending on memory size, and xx in all subsequent locations would be the same as the first.

Location	Memory Bank	Memory Size
017744	0	4K
037744	1	8K
057744	2	12K
077744	3	16K
117744	4	20K
137744	5	24K
157744	6	28K

The contents of location xx7776 (yyyyyy) in the instruction column above should contain the device status register address of the papertape reader to be used when loading the bootstrap formatted tapes specified as follows:

Teletype	Paper	Tape	Reader		177560
High-spee	ed Pape	er Tap	e Reader	-	177550



Figure A-1 Loading and Verifying the Bootstrap Loader



A.1.2 Loading with the Bootstrap Loader

.

1

0

Figure A-2. Loading Bootstrap Tapes into Core

A.2 THE ABSOLUTE LOADER

A.2.1 Loading the Absolute Loader

The Bootstrap Loader is used to load the Absolute Loader into core. (See Figure A-2.) The Absolute Loader occupies locations xx7474 through xx7743, and its starting address is xx7500.

A.2.2 Loading with the Absolute Loader

When using the Absolute Loader, there are three types of loads available: normal, relocated to specific address, and continued relocation.

Optional switch register settings for the three types of loads are listed below.

Type of Load	Switch 1 Bits 1-14	Register Bit 0
Normal	(ignored)	• • • • • • • • • • • • • • • • • • •
Relocated - continue loading where left off	0	1
Relocated - load in specified area of core	nnnnn (specified add	l ress)



A-5

APPENDIX B

USING THE PAL-11S ASSEMBLER

Run the assembler according to the directions in Section B.1. If another program is being assembled along with FPMP-11, it should be read before the FPMP-11 package. This other program must be followed by a .EOT instruction and must not define any FPMP-11 labels or conditional switches. After any user program being assembled with FPMP-11 has been read, the assembler prints EOF? and pauses. Place the switch setting tape previously created (refer to section 3.5.1.1) in the reader and type the RETURN key. At the end of this tape the assembler again prints EOF? Place the first source tape of the FPMP-11 package in the reader and type the RETURN key. After this source tape has been read and the assembler prints EOF?, place the next source tape in sequence in the reader and type RETURN. Repeat this sequence until all source tapes have been read. When the last tape has been read, the assembler proceeds to Pass 2. All of the tapes must be read again using the same procedure as above. The assembler produces the FPMP-11 object module on the binary output device specified in the initial assembler dialogue.

B.1 ASSEMBLER OPERATING PROCEDURES

Loading:	Use Absolute Loader. The start address of the Loader must be in the console switches.
Storage Requirements:	PAL-11S uses 8K memory.
Starting:	Immediately upon loading, PAL-11S is in control and initiates dialogue.

Initial Dialogue:

Printout

Inquiry

*S What is the input device of the source symbolic tape?

*B What is the output device of the binary object tape?

*L What is the output device of the assembly listing?

*T What is the output device of the symbol table?

Each of these questions may be answered by any one of the following characters:

Answer Indicated

Character

T Teleprinter keyboard

- L Low-speed reader or punch
- H High-speed reader or punch
- P Line Printer

Each of these answers may be followed by the other characters indicating options:

Option Typed		Function to be performed
/1	on pass	1 is the first first of the state of t
/2 /3	on pass on pass	
/ E		to be listed on the Teleprinter on the same pass of ul only for *B or *L).

Each answer is terminated by typing the RETURN key. Answering with a RETURN alone deletes the function.

Dialogue During Assembly:

Printout

Response

- EOF ? Place next tape in reader and type RETURN. A .END statement may be forced by typing E followed by RETURN.
- END ? Start next pass by placing first tape in reader and typing RETURN.
- EOM ? If the end-of-medium is on the listing device, the device may be readied and the assembly may be continued by typing RETURN.

If the end-of-medium is on the binary device, the assembler will discontinue the assembly and restart itself.

Restarting:

Type CTRL/P. The initial dialogue will be started again.

For more detailed information on the PAL-11S Assembler, refer to the PDP-11 PAL-11S Assembler and LINK-11S Linker Programmer's Manual (DEC-11-YRWB-D).

B.2 ASSEMBLER ERROR CODES

Error Code	Meaning
A	Addressing error. An address within the instruction is incorrect. Also includes relocation errors.
В	Bounding error. Instructions or word data are being assembled at an odd address in memory.
D	Doubly-defined symbol referenced. Reference was made to a symbol which is defined more than once.
I	Illegal character detected. Illegal characters which are also non-printing are replaced by a ? on the listing.
L	Line buffer overflow. All extra characters beyond 72 are ignored.
Μ	Multiple definition of a label. A label was encountered which was equivalent (in the first six characters) to a previously encountered label.
Ν	Number containing an 8 or 9 was not terminated by a decimal point.
$\mathbf{P}_{\mathbf{r}}$, where $\mathbf{r}_{\mathbf{r}}$	Phase error. A label's definition or value varies from one pass to another.
Q	Questionable syntax. There are missing arguments or the instruction scan was not completed, or a carriage return was not followed by a linefeed or form feed.
R	Register-type error. An invalid use of or reference to a register has been made.
S	Symbol table overflow. When the quantity of user-defined symbols exceeds the allocated space available in the user's symbol table, the assembler outputs the current source line with the S error code, then returns to the command string interpreter to await the next command string to be typed.
T	Truncation error. More than the allotted number of bits were input so the leftmost bits were truncated. T error does not occur for the result of an expression.
U	Undefined symbol. An undefined symbol was entered during the evaluation of an expression. Relative to the expression, the undefined symbol is assigned a value of zero.



APPENDIX C

USING LINK-11S

C.1 LOADING AND COMMAND STRING

The Linker is loaded by the Absolute Loader and is self-starting. It uses a simple command dialogue which allows the object module, load module and load map devices to be specified. During pass 1 and pass 2, the Linker asks for each object module individually.

For illustration purposes, the non-printing characters carriage return, line feed and space are represented as <CR>, <LF> and <SPACE>.

Operation begins by the linker typing its name and version. This is followed by the input option printed as *I<SPACE>. The responses are:

<cr></cr>	Read	object	module	from	HSR.
H <cr></cr>	Read	object	module	from	HSR.
L <cr></cr>	Read	object	module	from	LSR.

The input option is followed by the output option *O<SPACE>. The responses are:

<cr></cr>	Punch	load	module	on	HSP.	
H <cr></cr>	Punch	load	module	on	HSP.	
L <cr></cr>	Punch	load	module	on	LSP.	

LINK-11 asks if a load map is desired by typing *M<SPACE>. The legal responses are <CR> for no map, T<CR> or H<CR> or P<CR> for a map on the teleprinter, high-speed punch, or line printer, respectively.

The next two options concern the placement of the relocated object program in memory. The standard version of the Linker assumes it is linking for an 8K machine. It relocates the program such that it is as high as possible in 8K but leaves room for the Absolute and Boot Loaders. (These assumed values may be changed by altering parameters HGHMEM (highest legal memory address +1) and ALODSZ (number of bytes allocated for Absolute Loader and Boot Loader) and reassembling the linker.) The *T and *B options control the relocation of a program. After the option *T<SPACE> is printed, respond as follows:

> <CR> Relocate so that program is up against the current top of memory. If the top has not been changed, then the top is the assembled-in top (HGHMEM-ALODSZ). The standard assumption is 16384.-112.=16272 (37460₈).

n<CR>

n is an octal number (unsigned) which defines a new top address.

If a new top is specified, the *B option is suppressed.

After the option *B<SPACE> is printed respond as follows:

<CR> Use current top of memory.

n<CR> n is an unsigned octal number which defines the bottom address of the program. That is, a new top of memory is calculated so that the bottom of the program corresponds with n.

Once a top of memory has been calculated (by *T or *B), that value is used until it is changed.

LINK-11 indicates the start of pass one by printing PASS 1. The input is requested by the Linker, one tape at a time by printing *<SPACE>. The legal responses are:

<CR> Read a tape and request more input.

- U<CR> List all undefined globals on the teleprinter and request more input.
- E<CR> End of input. If there are undefined globals, list them on the teleprinter and request more input. Otherwise print the load map if requested, and enter pass 2.
- C<CR> End of input. Assign 0 to any undefined globals, print the load map (if requested), and enter pass 2.

The Linker indicates the start of pass 2 by printing PASS 2 and requests each input tape as in pass 1.

A <CR> is the only useful response to an asterisk (*) on pass 2. The modules must be read on pass 2 in the same order as pass 1. When the last module has been read, the Linker automatically finishes the load module and restarts itself.

Leader and trailer are punched on the load module.

If the low-speed punch (LSP) is being used for the load module output, it should be turned on before pass 2 begins, i.e., turn it on before typing E<CR> or C<CR>. The echo of these characters (and the load map) if printed on the Teletype are punched on the load module but may be easily removed since leader is punched on the load module. The LSP can also be turned on while leader is being punched (after the linker has typed PASS 2) to keep the load map, etc., from being punched onto the tape.

NOTE

On all command string options, except for *T and *B, the linker examines only the last character typed preceding the carriage return. Thus,

ABCDEFGH <CR >

is equivalent to H<CR>.

C.2 ERROR PROCEDURE AND MESSAGES

C.2.1 Restarting

CTRL/P is used for two purposes by LINK11-S. If a CTRL/P is typed while a load map is being printed, the load map is aborted and the Linker continues. CTRL/P typed at any other time causes the Linker to restart itself.

input.

C.2.2 Non-Fatal Errors

?MODULE NAME XXXXXX NOT UNIQUE

Message

Explanation

Non-unique object module name this error is detected during pass l and results in an error message and the module is rejected. The Linker will then ask for more

?MAP DEVICE EOM. TYPE <CR> TO CONTINUE

Load map device EOM - this error allows the user an option to fix the device and continue or abort the map listing. Any response, terminated by <CR> or <LF> causes the Linker to continue. A CTRL/P causes the map be to aborted.

Message

Explanation

?BYTE RELOC ERROR AT ABS ADDRESS xxxxxx.

A byte relocation error - the Linker tries to relocate and link byte quantities. However, relocation usually fails and linking may fail. Failure is defined as the high byte of the relocated value (or the linked value) not being all zero. In such a case, the value is truncated to 8 bits. The Linker automatically continues.

?LOAD XXXXXX NEXT

If the object modules are not read in the same order on pass 2 as pass 1, the Linker indicates which module should be loaded next by typing this message and asking for more input.

?XXXXXX MULTIPLY DEFINED BY MODULE XXXXXX.

Multiply-defined globals were discovered, during pass 1. The second definition is ignored and the Linker continues.

C.2.3 Fatal Errors

The Linker restarts after any of the following:

Message

Explanations

?SYMBOL TABLE OVERFLOW - MODULE xxxxx, SYMBOL xxxxx Symbol Table overflow.

?SYSTEM ERROR xx

System Error. Where xx is an identifying number as follows:

Number Meaning

- 01 Unrecognized symbol table entry found.
 - 02 A relocation directory references a global name which cannot be found in the symbol table.

Number	Meaning
03	A relocation directory contains a location counter modification command which is not last.
04	Object module does not start with a GSD.
05	The first entry in the GSD is not the module name.
06	An RLD references a section name which cannot be found.
07	The TRA specification references a nonexistent module name.
08	The TRA specification references a non-existent section name.
09	An internal jump table index is out of range.
10	A checksum error occurred on the object module.
11	An object module binary block is too big (more than 64 words of data).
12	A device error occurred on the load module output device.
numbers 10 program failu Linker or generated the 05 can occur	the program which object module. Error

ß

た

ø

6

C-5

C.2.4 Error Halts

LINK-11 loads all of its unused TRAP vectors with the code:

.WORD .+2,HALT

so that if the TRAP occurs, the processor halts in the second word of the vector. The address of the halt, displayed in the console lights, therefore indicates the cause of the halt.

Address of	HALT (octal)	Meaning
12		Reserved instruction executed.
16		Trace TRAP occurred.
26		Power fail TRAP.
32		EMT executed.

A halt at address 40 indicates an IOXLPT detected error. R0 (displayed in the console lights) contains an identifying code:

Code in RO	Meaning
0 1 2 3 4 5	Illegal memory reference, SP overflow or illegal instruction. Illegal IOX command. Slot number out of range. Device number illegal. Referenced slot not INITed. Illegal data mode.

IOXLPT also sets Rl as follows:

If the error code is 0, Rl contains the PC at the time of the error.

If the error code is 1-5, Rl points to some element in the IOT argument list or to the instruction following the argument list, depending on whether IOXLPT has finished decoding all the arguments when it detects the error.

APPENDIX D

SUMMARY OF FPMP-11 ROUTINES

This appendix lists all the global entry points of FPMP-11 and provides a brief description of the purpose of each. Sections D.1 and D.2 are for reference when it is desired to call FPMP-11 routines directly (i.e., without the use of the TRAP handler). Entry names preceded by an octal number can be referenced via the TRAP handler. The number is the "routine number" referred to throughout this manual. If the number is enclosed in parentheses, the routine cannot be accessed by the present TRAP handler, but has been assigned a number for future use.

Examples of the calling conventions are:

POLISH MODE:

Ó

_	• the second second	
	JSR R4,\$POLSH \$subrl \$subr2	;enter Polish mode ;call desired subroutines
	•	
	\$subrn .WORD .+2	;call last subroutine desired ;leave Polish mode.
J5RR:		
	JSR R5, subr BR XX	;call desired subroutine
	.WORD argl .WORD arg2	;subroutine argument address
XX:	.WORD argn	;last argument ;return point
	•	

push args onto stack JSR PC,subr

D.1 OTS ROUTINES

These are the routines taken from the FORTRAN operating time system. The codes used in the following table are:

- S = Routine is included in the standard single precision (2-word)
 package.
- D = Routine is included in the standard double precision (4-word) package.
- SD = Routine is included in both standard packages.

Octal codes shown in parentheses are not yet implemented.

NAME	OCTAL CODE	PKG	# OF ARGU	MODE	DESCRIPTION
ŞADD	14	D	2	Polish	The double precision add routine. Adds the top stack item (4 words) to the second item (4 words) and leaves the four word sum in their place.
\$ADR	12	S	2	Polish	The single precision add routine. Same as \$ADD except it uses 2 word numbers.
AINT	26	S	l	J5RR	Returns sign of argument * greatest real integer = absolute value of the argument in R0,R1.
ALOG	53	S	l	J5RR	Calculates natural logarithm of its single argument and returns a two word result in R0,R1.
ALOG10	54	S	1	J5RR	Same as ALOG, except calculates base-10 logarithm.
ATAN	42	S	1	J5RR	Returns the arctangent of its argument in R0,R1.

JPC:

NAME	OCTAL CODE	PKG	# OF ARGU	MODE	DESCRIPTION
ATAN 2	(43)	S	2	J5RR	Returns ARCTAN(ARG1/ARG2) in R0,R1.
\$CMD	16	D	2	Polish	Compares top 4 word items on the stack, flushes the two items, and returns the following condition codes: 4(SP) @SP N=1,Z=0 4(SP) = @SP N=0,Z=1 4(SP) @SP N=0,Z=0
\$CMR	17	S	2	Polish	Same as \$CMD except it is for 2 word arguments.
COS	37	S	1	J5RR	Single precision version of DCOS.
DATAN	44	D	1	J5RR	Double precision version of ATAN.
DATAN 2	(45)	D	2	J5RR	Double precision version of ATAN2.
DBLE	(34)		1	J5RR	Returns in R0-R3 the double precision equivalent of the single precision (two word) argument.
\$DCI	(57)	SD	4	JPC	ASCII to double conversion. Calling sequence: Push address of start of ASCII field. Push length of ASCII field in bytes. Push format scale D (from W.D) position of assumed decimal point (see FORTRAN manual). Push P format scale (see FORTRAN manual). JSR PC,\$DCI. Returns 4 word result on top of stack.
	(61)			JPC	Double precision to ASCII conversion. Calling sequence: Push address of start of ASCII field. Push length in bytes of ASCII field (W part of W.D) Push D part of W.D (position of decimal point). Push P scale. Push 4 word value to be convert- ed, lowest order word first. JSR PC,\$DCO.

r

r.

ø

NAME	OCTAL CODE		# OF ARGU	MODE	DESCRIPTION
DCOS	41	D	1	J5RR	Calculates the cosine of its double precision argument and returns the double precision result in R0-R3.
DEXP	52	D	1	J5RR	Calculates the exponential of its double precision argument, and returns the double precision result in RO-R3.
\$DI	(11)	SD		Polish	Converts double precision number on the top of the stack to integer. Leaves result on stack.
\$DINT	(76)	D	1	Polish	OTS internal function to find the integer part of a double precision number.
DLOG	55	D	1	J5RR	Double precision (4 word) version of ALOG.
DLOG10	56	D	1	J5RR	Double precision (4 word) version of ALOG10.
\$DR	(6)		1	Polish	Replaces the double precision item at the top of the stack with its two word, rounded form.
DSIN	40	D	1	J5RR	Calculates the sine of its double precision arg. and returns the double precision result in R0-R3.
DSQRT	47	D	1	J5RR	Calculates the square root of its double precision arg. and returns the double precision result in R0-R3.
\$DVD	23	D	2	Polish	The double precision division routine. Divides the second 4-word item on the stack by the top item and leaves the quotient in their place.
\$DVI	(24)		2	Polish	The integer division routine. Calculates 2(SP)/@SP and returns the integer quotient on the top of the stack.
\$DVR	25	S	2	Polish	The single precision division routine. Same as \$DVD, but for 2 word floating point numbers.

NAME	OCTAL CODE	PKG	# OF ARGU	MODE	DESCRIPTION
\$ECO	(62)	SD	5	JPC	Single precision to ASCII conversion according to E format. Same calling sequence as \$DCO except that a 2-word value is to be converted.
EXP	51	S	1	J5RR	Single precision version of DEXP. Returns result in R0,R1.
\$FCALL	-	S			Internal OTS routine.
\$FCO	(64)	SD	5	JPC	Same as \$ECO except uses F format conversion.
FLOAT	(32)		1	J5RR	Returns in RO-Rl, the real equivalent of its integer argument.
\$GCO	(63)	SD	5	JPC	Same as \$ECO except uses G format conversion.
\$ICI	(65)		2	JPC	ASCII to integer conversion. Calling sequence: Push address of start of ASCII field. Push length in bytes of ASCII field. JSR PC,\$ICI Returns with integer result on top of stack.
\$ICO	(67)		3	JPC	Integer to ASCII conversion. Calling sequence: Push address of ASCII field. Push length in bytes of ASCII field. Push integer value to be converted. JSR PC,\$ICO Error will return with C bit set on. RO-R3 destroyed.
IDINT	(31)		1	J5RR	Returns sign of arg * greatest integer <= arg in R0. Arg is double precision.
\$ID	(5)	SD	1	Polish	Convert full word argument on the top of the stack to double precision and return result as top 4-words of stack.
IFIX	(35)		1	J5RR	Returns the truncated and fixed real argument in R0.

NAME	OCTAL CODE	PKG	# OF ARGU		DESCRIPTION
INT	(30)			J5RR	Same as IDINT for single precision args.
\$INTR	(27)	S	1	Polish	Same function as AINT, but called in Polish mode with argument and returns result on the stack.
\$IR	(4)	SD	1	Polish	Convert full word argument on the top of the stack to single precision and return result as top 2-words of stack.
\$MLD	22	D	2	Polish	Double precision multiply. Replaces the top two doubles on the stack with their product.
\$MLI	(20)		2	Polish	Integer multiply. Replaces the top 2 integers on the stack with their full word product.
\$MLR	21	S	2	Polish	Single precision multiply. Replaces the top two singles on the stack with their product.
\$NGD	(3)	SD	2	Polish	Negate the double precision number on the top of the stack.
\$NGI	(1)	SD	1	Polish	Negate the integer on the top of the stack.
\$NGR	(2)	SD	1	Polish	Negate the single precision number on the top of the stack.
\$OCI	(66)	52	2	JPC	ASCII to octal conversion. Same call as \$ICI.
\$0C0	(70)		3	JPC	Octal to ASCII conversion. Same call as \$ICO.
\$POLSH		SD			Called whenever it is desired to enter Polish mode from normal in-line code. It must be called via a JSR R4,\$POLSH.
\$POPR3		D		Polish	Internal routine to pop 2-words from the stack and place them into R0,R1.
\$POPR4	- 			Polish	Internal routine to pop 4-words from the stack and place them in R0-R3.

D**-6**

-

ł?

٤.

NAME	OCTAL CODE	PKG	# OF ARGU	MODE	DESCRIPTION
\$POPR5		D		Polish	Internal routine to pop 4-words from the stack and place them in registers R0-R3.
\$PSHR1	-	SD	1 .	Polish	Internal routine to push the contents of R0 onto the stack.
\$PSHR2		SD	-	Polish	Same as \$PSHR1.
\$PSHR3	672)	SD		Polish	Push R0,R1 onto stack.
\$PSHR4	em	SD		Polish	Push R0-R3 onto stack.
\$PSHR5		SD	-	Polish	Same as \$PSHR4.
\$RCI	(60)	SD	4	JPC	ASCII to single precision conversion. Same calling sequence as \$DCI. Returns 2-word result on top of stack.
\$RD	(7)			Polish	Converts the single precision number on the top of the stack to double precision format. Leaves result on stack.
\$RI	(10)	SD		Polish	Converts single precision number on the top of the stack to integer. Leaves result on stack.
\$SBD	15	D	•	Polish	The double precision subtract routine. Subtracts the double precision number on the top of the stack from the second double precision number on the stack and leaves the result on the top of the stack in their place.
\$SBR	13	S		Polish	Same as \$SBD but for single precision.
SIN	36	S	l	J5RR	Single precision version of DSIN.
SNGL	(33)		1	J5RR	Rounds double precision argument to single precision. Returns result in R0, R1.
SQRT	46	S	1	J5RR	Single precision version of DSQRT.
TANH	50	S	1	J5RR	Single precision hyperbolic tangent function. Returns (EXP(2*ARG)-1) / (EXP(2*ARG)+1) in R0,R1.

t.

-

ġ.

D-7

D.2 NON-OTS ROUTINES

These routines are written especially for FPMP-11 and should not be called directly by the user.

NAME	OCTAL CODE	PKG	DESCRIPTION
ŞERR	-	SD	Internal error handler.
\$ERRA	-	SD	Similar to \$ERR.
\$LDR	71	S	Load FLAC, single precision.
\$LDD	72	D	Load FLAC, double precision.
\$STR	73	S.	Store FLAC, single precision.
\$STD	74	D	Store FLAC, double precision.
TRAPH	-	SD	The TRAP handler routines and tables.

D.3 ROUTINES ACCESSED VIA TRAP HANDLER

The following is a table of the FPMP-11 routines which can be accessed via TRAPH, the trap handler. Each routine name (entry point) is preceded by its TRAP code number to be used to access it, and followed by a brief description of its operation when called via the TRAP handler. Those entries which are preceded by an asterisk (*) perform operations only on the FLAC, and address no operands. For example, a TRAP call to the single precision square root routine can be coded as follows:

TRAP 46

The net effect of the above TRAP instruction is to replace the contents of the FLAC with its square root and then set the condition codes to reflect the result. Note that since the FLAC is implicitly addressed in this instruction, the TRAP call supplies no other address. For such a TRAP call, the addressing mode bits (bits 6 and 7 of the TRAP instruction) are ignored.

All entries not marked by an asterisk require an operand when called. The operand is addressed in one of the 4 addressing modes explained in section 3.1.1. The addressing mode is specified in bit 6-7 of the TRAP instruction.
("Operand" is the contents of the location addressed in the TRAP call.)

.,**)**,-

fic

*

Ģ.

	OCTAL CODE	NAME	DESCRIPTION
	14	\$ADD	Double precision addition routine. Adds operand to the FLAC. Assumes 4-word operand.
lon en Lone	12	\$ADR	Single precision addition routine. Adds operand to the FLAC. Assumes 2-word operand.
*	26	AINT	Replaces contents of the FLAC by its integer part. SIGN(FLAC) * greatest integer <= FLAC . Assumes 2-word argument in FLAC.
*	53	ALOG	Replaces contents of the FLAC by its natural logarithm. Assumes 2-word argument in FLAC.
*	54	ALOG10	Same as ALOG, except calculates base-10 log.
* * **********************************	42	ATAN	Replaces contents of the FLAC by its arctangent. Assumes 2-word argument in FLAC.
	16	\$CMD	Compares operand to the contents of the FLAC, and returns the following condition codes. FLAC <operand, n="1,Z=0<br">FLAC=operand, N=0,Z=1 FLAC>operand, N=0,Z=0 Assumes 4-word operands.</operand,>
	17	\$CMR	Same as \$CMD, but for 2-word operands.
*	37	COS	Same as DCOS, but for 2-word argument.
*	44	DATAN	Same as ATAN, but for 4-word argument.
*	52	DEXP	Replaces the contents of the FLAC by its exponential. Assumes 4-word argument in the FLAC.
* * *	55	DLOG	Same as ALOG, but for 4-word argument.
*	56	DLOG10	Same as ALOG10, but for 4-word argument.
*	41	DCOS	Replaces the contents of the FLAC by its cosine. Assumes 4-word argument in the FLAC.

	OCTAL CODE	NAME	DESCRIPTION
*	40	DSIN	Same as DCOS, but calculates sine instead of cosine.
*	47	DSQRT	Replaces the contents of the FLAC by its square root. Assumes 4-word argument in the FLAC.
	23	\$DVD	Double precision division routine. Divides the FLAC by the operand and stores the result in the FLAC. Assumes 4-word operands.
a De ag	25	\$DVR	Same as \$DVD, but for 2-word operands.
*	51	EXP	Same as DEXP, but for 2-word argument.
	72	\$LDD	Same as \$LDR, but assumes 4-word operand.
	71	\$LDR	Replaces the contents of the FLAC by the operand. Assumes 2-word operand.
	22	MLD	Double precision multiplication routine. Multiplies the contents of the FLAC by the operand and stores the result in the FLAC. Assumes 4-word operands.
	21	ŞMLR	Same as \$MLD, but for 2-word operands.
	15	\$SBD	The double precision subtraction routine. Subtracts the operand from the contents of the FLAC. Assumes a 4-word operand.
	13	\$SBR	Same as \$SBD, but for 2-word operand.
*	36	SIN	Same as DSIN, but for 2-word argument.
*	46	SQRT	Same as DSQRT, but for 2-word argument.
	73	\$STR	Stores the contents of the FLAC into the operand location. The contents of the FLAC are unchanged.
	74	\$STD	Same as \$STR, but assumes 4-word operand location.
*	50	TANH	Replaces the contents of the FLAC by its hyperbolic tangent. Assumes 2-word argument.

€.

APPENDIX E FPMP-11 SOURCE LISTING

P

\$2

Ę,

Q

This source listing of FPMP-11 is included for documentation of the logic only. The sources provided to users do not have comments because of size restrictions.

1	000001	SINGLE#1
2	000001	DOUBLER1
3	000001	CND\$7=1
4	000001	CND512#1
5	000001	CND\$17=1
6	000001	CND322=1
7	000001	CND523#1
8	000001	CNDS24=1
9	000001	CND\$25=1
10	000001	CND\$26=1
11	000001	CND\$29=1
12	000001	CND\$34#1
13	000001	CND\$36#1
14		EOT
15		······································

ò

FPMP11 FLOATING POINT & MATH PASMACRO VR04-14 07-SEP-72 1143 PAGE 3

1	IPRODUCT CODE	DEC=11=NFPMA=A=LA	
3	COMPUTER	PDP=11	
4 5 6	ICUNFIGURATION	PAPER TAPE CONFIGURATION 8192 WORDS MEMORY	IS MINIMUM
7 8 9	ISOFTWARE REQUIREMENTS	PAL=115 (OR MACRU=11) LINK=115 (OR LINK=11)	
10	PROGRAM NAME	FPMP=11	
12 13 14	JVERSION J	VERSION LEVEL 1 Patch Level A	
15 16 17 18 19	DESCRIPTION 1 1 1	FLOATING POINT MATH PACKA Plus trap Handler (Floating Point Subroutin Dos-11 fortran IV UTS)	
20 21 22	FAUTHOR	E. PETERS (TRAP HANDLER & INTEGRATION)	PACKAGE
23 24	PDATE	AUGUST, 1972	
25 26 27		COPYRIGHT 1972, DIGITAL E Maynard, Massachusetts Ø1	

2_

FPMP11 FLOATING POINT & MATH PA MACRU VR04=14 07=SEP=72 11143 PAGE 4

1	0000001	.CSECT	
3		CONDITIONALS TO	GENERATE THE STANDARD PACKAGES,
4	•	.IFDF SINGLE!	SINGLE PRECISION PACKAGE?
5	000001	CND\$2=1	ISADR, SSBR
6	000001	CND\$3=1	ALOG, ALOG10
7	000001	CND\$4=1	FAINT
8	000001	CND\$6=1	ISCMR
9	000001	CND\$18=1	1SDVR
10	000001	CND\$20=1	TEXP
11	000001	CND\$30=1	13MLR
12	000001	CND\$37#1	ISIN, COS
13	000001	CND\$38=1	TANH
14	000001 -	CND\$39=1	FATAN, ATAN2
15	000001	CND\$41=1	ISQRT
16	000001	CND\$44#1	
17	000001	CND346=1	
18		ENDC	
19			
20		.IFDF DOUBLE!	DOUBLE PRECISION PACKAGE?
21	000001	CND51=1	13ADD, SSBD
22	000001	CND\$5=1	7 SCMD
23	000001	CND\$10=1	JOLOG, DLOG10
24	000001	CND\$13#1	IDSIN, DCOS
25	000001	CN0514=1	IDSORT
26	000001	CND\$15#1	JDATAN, DATAN2
27	000001	CND\$16#1	1 BOVD
28	000001	CND\$19=1	IDEXP
29	000001	CND\$28=1	1\$MLD
30	000001	CND\$45=1	JSLDD
31	000001	CND\$47=1	JSSTD
32		, ENDC	
33			5 1 Y 1 1 6 Y 1 1 1 1 1 1 1 1 1 1 1 1 1 1
34		.IFDF SINGLEI	
35	000001	CND\$8=1	/SDCI, SRCI
36	000001	CND\$9=1	JSECO, SFCO, SGCO, SDCO
37	000001	CND\$31=1	JSNGI, SNGR, SNGD
38	000001	CND\$42=1	JTRAPH
39		, ENDC	

ø

~3

FPMP11 FLOATING POINT & MATH PA MACRU VR04-14 07-SEP-72 11:43 PAGE 5

				000-30.	TANDO		
12	ta ta m tá m 4		.IFDF	CND538;	TANH? ISADR, SS	80	
2	000001		CND\$2=1	10. ⁶	130VR	OR	
3	000001		CND\$18#1		JEXP		
4	000001		CND\$20#1		ISFCALL		
5	000001		CND\$21=1		ISMLR		
6	000001		CND\$30=1		ISPSHR3		
7	000001		CND\$32#1		Jaranka		
8			, ENDC				
9			10 10 A 4 5 A 10	P7 23			
10			IFNDF	FPU	NUEDALENA	# 17 1 P & P & B & B & B	
11	04000A		. IFOF	CHUSOIC	1SADR, SS	\$371CND\$39	
12	000001		CND\$2#1 CND\$18#1		ISDVR	JUN	
13 14	000001		CND530=		ISMLR :		
15	000001		.IFDF	CNDS371	SIN, COST		
16	000001		CND\$4=1	CHUBUI	JSINTR		
17	000001		.ENDC		· i. m Tialiui		
18			ENOC				
19		· .	.IFOF	CNASIAL	CNDSISICN	D\$151CND\$19	
20	000001		CND\$1=1	CHUGAU.	/\$ADD,\$5		
20	000001		CND316#1	n · · ·	7\$0VD		
22	000001		CND\$28#1		SMLD		
23	000001		CND\$33#1		ISPOPR4		
24	000001		.IFDF	CNDS131	USIN, DCO	57	
25	000001		CND\$11#1	· · · · · · · · · · · · · · · · · · ·	JSDINT		
26	000001		.ENDC	•	1.001.01	- 	
27			ENDC				
28			IFDF	CNDSJIC	NDS101	ALOG OR DLOG?	
29	000001		CND\$27#1	-	/SIR,SID		
30	0000004		.ENDC				
31	•		, IFDF	CND\$1911	CN03201	EXP OR DEXP?	
32	000001		CND\$27=1		ISIR, SID		
33	000001		CND\$35#1		/\$R1,501		
34			ENDC				1. A.
35			IFOF	CN05141	DSGRT?		
36	000001		CND31=1		ISADD		
37	000001		CND\$16=1	-	ISOVD		
38	••• • ••• •••		.ENDC				
39			. IFOF	CNDS41;	SORT?		
40	000001		CND 32=1	· · ·	SADR .		
41	000001		CND \$18=1		1SOVR		
42			. ENDC				
43			ENDC				
44							
45			.IFOF	CND\$231	FLOAT?		
46	000001		CND327=1		ISIR, SID		
47	000001		CND\$33=1		ISPOPR3		
48			ENDC				
49	e te e		, IFOF	CND\$2210	CND\$26;	IFIX, INT, OR	IDINT?
50	000001		CND\$35=1		ISRI, SDI	· · · · · ·	
51			.ENDC	:			
52			.IFOF	CND3397	ATAN OR	ATAN27	
53	000001		CND\$33=1	1	IPOPR3		
54			. ENDC				

TRAP02 MACRO VR04=14 07=SEP=72 11:43 PAGE 6

Þ

Q

ø

1				TITLE	TRAP02	
2 3 4				.IFDF	CNDS42	
3				GLOBL	TRAPH, SERRA	
4			7		P=11 TRAP HANDLER	бана на селото на се Селото на селото на се
5		000000		R0=%0		
6 7		000001		R1=%1		
7		000002		R2=%2		
8		000003		R3#%3		
9		000004		R4=%4		
10		000005		R 5 # % 5		
11		0000006		SP#%6		
12		000007		PC=%7		
13						
14	00000	042766	TRAPHI	BIC	#17,2(SP);	CLEAR ALL USER COND CODES
		000017				
		000002				
15	00006	005046		CLR	= (SP) ;	SPACE FOR ADDR MODE
16	00010	010546		MOV	R5,=(SP)1	SAVE THE REGISTERS
17	00012	010446		MOV	R4, = (SP)	
18	00014	010346		MQV	R3,=(SP)	
19		010245		MOV	R2,=(SP)	
20		010146		MOV	R1,=(SP)	
21		010046		MOV	RØ,-(SP)	
22		016603		MOV	20(SP),R31	GET USER'S STATUS WORD
	ar - an	000020				
23	00030	042703		BIC	#20,R31	CLEAR T-BIT FOR US
	298 an 201 - 194	000020		and the state of t		
24	00034	010337		MOV	R3,0#177761	ESTABLISH AS CURRENT STATUS
940 - - - F	00004	177778				
25	00040	016601		MOV	16(SP),R1;	GET USER'S PC
<u> </u>	000-0	000016				च्या स्थल हे स्वयंतित क्रिया⊺ रूपमा । च्या
26	abaAA	010105		MOV	R1, R5;	COPY USERIS PC
27		014104		MOV	= (R1), R47	PICK UP TRAP INSTRUCTION
28		010403		MOV	R4, R31	COPY
29		042704		BIC	#177700,841	CALC TABLE INDEX
63	000-5			H 4 H	****	WHEN INVEN SIVER
2.4	0005¢	177700		ASL	R41	TIMES TWO
30		006304		MOV	THL\$42(R4),R41	GET TABLE ENTRY
31	00060	016404		CIQ Y	1069-566-116-1	
. 3.0	3005×	000500		BEQ	ERR\$421	ERROR: NO ENTRY IN TABLE
32		001556			R4, R21	COPY TABLE ENTRY
		010402				CLEAR MODE BITS
4 ڊ	000010	042702		BIÇ	#140000,R21	APPHUL DAKE BYLW
7 4		140000		Ann	PC, R2;	RELOCATE ROUTINE ADDRESS
		060702			#40000,R41	ADDRESSING REQUIRED
30	00000		PT5421	DÎ I	A40000, K41	WOOKE33140 KEGOIKED
		040000		63 m 65		BANKA TE NAME SEANTOER
		001514			NAD\$421	BRANCH IF NONE REQUIRED
		106103			R31	TEST OPERAND ADDRESS MODE
		100122			PLM5421	BRANCH IF BIT 6 EQUALS 0
	00110	103004		BCC	STK\$421	BRANCH IF ORD MODE
41			1	RELATIVE		
		010500		MOV	R5, R01	COPY USERIS PC
43		062500			(R5)+, R01	CALC ACTUAL OPERAND ADDRESS
44	00116		UPC\$421	MOV	R5,16(SP);	UPDATE USER'S PC
		000016				
45	00122		STK5421	MOV	#FAC\$42+6,R51	ADDRESS OF FLAC
		000440				

46	00126	005704	TST	R41	SINGLE OR DOUBLE?
47	00130	002403	BLT	ST45421	BRANCH IF DOUBLE
48		005015	CLR	@R51	CLEAR LAST 2 WORDS OF FLAC
				= (R5)	प्रसन्धिको स्थान प्रसन्ताः इतिहासि २००३ स्थलः २३ सार्वन्त्र स्थलः स्थलः । अस्तिः २०२४ र
49		005045	CLR		
		005725	TST	(R5)+1	INCR R5
51	00140	011546 ST45421	MOV	@R5, * (SP) 1	PUST THE FLAC
52	00142	014546	MOV	-(R5),=(SP)	
53		014546	MOV	- (R5), - (SP)	
			MOV	-(R5),-(SP)	
54		014546			ATLALT ALL CALINE TA
55		005704	TST	R41	SINGLE OR DOUBLE?
56	00152	002402	B ្ពT	ST6\$421	BRANCH IF DOUBLE
57	00154	022020	CMP	(RØ)+,(RØ)+;	INCR RØ BY 4
		000404	BR	OT2\$42	
		062700 ST05421		#8.,KØ	
59	00100		AUU	HO . FIND	
		000010		· · · · · · · · · · · · · · · · · · ·	Marco Marco - Sanata - 19 - Sanat
60		014045	MOV	-(RØ),-(SP);	PUSH OPERAND
61	00106	014046	MOV	-(R0), -(SP)	
62	00170	014046 0725421	MOV	-(R0),-(SP)	
63		014046	MOV	-(R0),-(SP)	
	091.6	fit the and ground and	* 2 5 47 *		
64		-	6		10 h C
65				TINE IN POLISH N	
66		7		NOT A STANDARD P	
67		1	IN ORDER	R TO REDUCE OVERH	
	00174	012704	MOV	#ADR\$42,R41	ADDRESS OF RETURN ADDR
		0002021			
69	0000a	000112	JMP	ØR2;	CALL SUBROUTINE
					RETURN ADDRESS
70	00505	0002041ADH\$421		,+21	REIURN AVUNGOO
71		ta katika katika 🚛 🗛 🖓		RESULT TU PLAC	
72	00204	012705	MOV	#FAC\$42,R51	ADDR OF FLAC
		0004321			
7 1	00010	012025	MOV	(SP)+, (R5)+	
				(SP)+, (R5)+	
		012625	MOV		
75		012625	MOV	(SP)+, (R5)+	
76	00216		MOV	(SP)+,(R5)+	
77	00220	011700 RET\$421	MOV	ØPC, KØJ	MAKE RØ POSITIVE
78		012705	MOV	#FAC\$42,851	ADDR OF FLAC
, .	3.0 AF 300 TH 900	0004321			
77 (1)	a (4 o (2 e	005725	TST	(R5) +1	TEST THE FLAC
				NEGS421	BRANCH IF FLAC MINUS
	00230	002410	BLT		
			BGT	PLS\$427	BRANCH IF PLUS
82	00234	005725	TST	(R5) +	
83	00235	001011	BNE	PLSS42	
	00240	005725	TST	(R5) +	
			BNE	PLSS42	
85	00242	001007			
86	00244	005725	TST	(R5)+	
87		001005	BNE	PLSS42	
88	00250	005000	CLR	RØJ	FLAG FLAG AS ZERO
89	00252	005400 NEGS421	NEG	RØJ	FLAC IS NEG
90	00254	053765 CMF\$421	BIS	04177776,20(SP)	ISET USERIS CONDS
#	and are set in the	177776			
•		000020	* • *	271 PA 8	SET COND DÜDED
		005700 PL\$\$42:	TST	R01	SET COND CUDES
92	00264	012600 CM15421	MOV	(SP)+, RØ1	RESTORE USER'S REGS
93	00206	012601	MOV	(SP)+,R1	
94	00270	012602	MOV	(SP)+,R2	
		012603	MOV	(SP)+,R3	
		012604	MOV	(SP)+,R4	
a. U	ଧ୍ୟ ଜୁଣ୍ଡ ଲା	H R . T II T	2 S Mar 7	জিল্পির রাজনা .	

ø

ф

9	7	00276	012505		MOV	(SP)+,R5	
		00300			TST	(3P)+1	TEST IF STACK MODE
			001413		BEQ	RTIS421	NO, SO RETURN
					BPL	RT25421	BRANCH IF SINGLE PREC
			100006				POP USER! ARG.
1	01	0300	012666		MOV	(SP)+,6(SP);	FUR WOEK, MRG.
			000000				
. 1	02	0312	012666		MOV	(SP) + , 6(SP)	
			000006				
1	03	0316	022626		CMP	(SP)+,(SP)+	
	04		000002		RTI		RETURN TO USER
· · ·	05			RT2\$421	MOV	(SP)+,2(SP))	POP THO WORD ARG.
	ψ÷	22 Q to 64	000002	1.1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
4	aE	14 72 O #			MOV	(SP)*,2(SP)	
. ł	06	6940	012666		rių v	(0F) + / 2 (0F)	
			000002		0		
-	07		000002	RTI5421	RTI		
1	08	L					
1	09	le de la		1	ROUTINE	TO MAKE JSRR C	
1	10	0334	004512	NADS421	JSR	R5,0821	CALL SUBROUTINE
	11		012705		MOV	(PC)+,R51	PICK UP ADDR OF FLAC
	12		000432	1	WORD	FAC\$421	ARG ADDRESS (FLAC)
	13		010025		MOV	RØ, (K5)+1	STORE RESULT INTO FLAC
			010125		MOV	R1, (R5)+	and black and a second s
	14				MOV	R2, (K5)+	
	15		010225				
			010325		MOV	R3, (R5)+	All man and an an Alice (3) a fill an an an an an an bit
1	17	0352	000722		BR	RET\$421	GO DO STANDARD RETURN
1	18						
1	19			1	MORE MOD	DE CHECKING	
	20		103010	PLMS421	BCC	STM5421	BRANCH IF STACK MODE
	21			1	IMMEDIAT	TE MODE	
	22		010500	*	MOV	R5, R01	ADDR IS USER'S PC
	23		005704		TST	R41	SINGLE OR DOUBLE
	24		002003		BGE	PL15421	BRANCH IF SINGLE
					ADD	#8.,K51	UPDATE USERIS PC
1	25	6304	082705		AUU	#0 s / 10 /	OPPAIS, COST OF 1G
			000010		19 au		
			000652		BR	UPCS42	the second se
	27			PL15421	CMP	(R5) +, (R5) + i	UPDATE PC
1	28	0374	000650		BR	UPC\$42	
1	29			1	STACK MO	ODE	
	30		010000	STM\$421		SP,R0	
	31		062700		ADD	#22, RØ1	CALC ADDR OF ARG ON STACK
	297 (B)	100 mg 20 20	000022				
	32	0404			INC	14(SF)1	FLAG STACK MODE
*	Q 6	****			* 14 4	14(0))/	I BUA ATWEN HAAM
			000014		ŦĊŦ	() A s	CTARLE OF BOUSIES
	33		005704		TST	R41	SINGLE OR DOUBLE?
			002243		BGE	STKS421	BRANCH IF SINGLE
1	35	0414	005466		NEG	14(SP)1	FLAG DOUBLE
			000014				
1	36	0420	000640		BR	STKS42	
	37						
	38			2	ERRORI F	ROUTINE NOT AVA	ALLABLE IN PACKAGE
		0422	005000	ERKS421	CLR	RØJ	SIGNAL TRAPH ERROR
					JSR	R5, SERRAI	R1 POINTS TO BAD TRAP INSTR
Å	40	5 4 5 4	021366		# WT	र के कि सिर्मात के सिर्मा ह	्रम् का प्राप्तक के रहे हैं पर के स्थान अने गा देखा है है के नाइक कि कि है के प्राप्त के कि कि के के प्राप्त क
	<u> </u>	12 A 2 -			80	EUDEAn.	HARD STOP
	41		000774		BR	ERR\$42;	NWUA SIAL
	42				.		
	43			1		G ACCUMULATOR	
1	44	0432	000000	FACS421	, WORD	0,0,0,0	

		and the second second			
	0434	000000			
	0436	000000			
	0440	000000			
145			, IFOF	CNDSO	
				SON FUDGE	
146	** * * *				AND AP DETUDN LAAP
147	0442	012704 CMR\$421	MOV	#CAR\$42,R41 4	ADDR OF RETURN ADDR
		0004521			
148	0446	000167	JMP	SCMR	
	88 Y 1 1 1 1 1	002656		1997 - 1 L 7	
	a . 8 a				
149		0004541CAH\$421	WORD	**2	
150	0454	053766	BIS	##177776+24(SP) #	ISET USER COND
		177776			
		000024			
4 52 4	0160		P MD	(801+ (891++ B	OF STACK
151		022626	CMP		OF UTACK
152	0404	000677	BR	CM1542	
153			. ENDC		
154					
155			. IFOF	CND\$5	
	12 4 8 2	GINTOA CHURADE			
156	6400	012704 CMUS421	MOV	#CAD\$42,R4	
		0004761			
157	0072	000167	JMP	SCMD	
~		002544			
158	QATE	0002541CAD\$421	WORD	CMF\$42	
	est c	DOURD- GARGARE		WITE WAR	
159		A B B	ENDC	a de la companya de l	
160		040000	PMODE=40		
161		100000	OMODE=10	0000	
162	0500	000000 TBL\$421	, WORD	0,0,0,0,0,0,0,0	0-7
		000000		and the second sec	
				· · ·	
		000000			
	0506	000000			
	0510	000000			
	0512				
		· · · ·			
		000000			
		000000			
163	0520	000000	, WORD	0,0	10-11
	0522	000000			
164			, IFDF	CND\$2	
	(A & /) A	44.740		SADR=PT\$42+PMODE	112
165		041712	, WORD		
166	0526	041706	NORD	SSBR-PTS42+PMODE	113
167			, ENDC		
168			IFNDF	CND52	
169			WORD	0,0	112-13
				τφ μ 2×*	7 (A) 498 11 19 14 1
170			.ENDC		
171		1	IFOF	CND\$1	
172	0530	140606	WORD	SADD-PT542+PMODE+	DMODE 114
173		140502	WORD	SSBD-PTS42+PMODE+	DMODE 115
174	/र भा ग €	प्रसः यथुन ^भ ाष्ट्रिय अससः ।	ENDC	ingen es nationale of the new Comments of the transferred many .	en en anni anna 18 668 (2) .
				PHARI	
175			, IFNDF	CNDS1	· · · · · · · · · · · · · · · · · · ·
176			. WORD	0,0	114-15
177			ENDC		
178			, IFOF	CNDSO	
179	0534	140370	WORD	CMD\$42=PT\$42+PMOD	E+DMODE 116
	्र भूम स	का भार 5 वे लिए हैं जिसके से प्राप्त के प		्यत्र र स्ति प्राप्त विश्व प्राप्त र र ज्यान्य स्वयं र र रिक्षिये स्ति	······································
180			.ENDC	an a chi	
181			. IFNDF	CND\$5	
182			WORD	0	116
183			ENOC		
184			, IFOF	CNDSD	
****			फ्राओल संचित्र"	भक्त गर सर्वे भग गर	

<u>م</u>.

TRAP02 MACRO VR04=14 07=SEP=72 11143 PAGE 6+

ø

*

 $\mathcal{A}_{\mathcal{V}}$

D.

185	0536	040344	NORD	CMR\$42-PT\$42+PMODE	117
186	9977	*****	ENDC	and the second	
				CNDSO	
187			, IFNDF		
188			.WORD		717
189			.ENDC		
190	0540	000000	WURD	Ø	120
191			.IFDF	CNDSJØ	
192	0542	057064	WORD	SMLR-PT542+PMODE	121
193			ENDC		
194			IFNDE	CNDSJØ	
					101
195			WORD		121
196			. ENDC		
197			. IFOF	CND\$28	
198	0544	156050	. WORD	SMLD=PTS42+PMODE+DMODE	122
199			, ENDC		
200			, IFNDF	CND\$28	
201			WORD	0	122
202			ENDC	 A second s	
203			IFDF	CND\$16	
	ARAC	150110		SDVD=PTS42+PMODE+DMODE	103
204	0040	152112	.WORD	2040=F12424FMUDE4DMUDE	123
205			. ENDC		
206			. IFNDF	CND516	
207		- 19	WORD		123
208			ENDC		-
209	Ø55Ø	000000	WORD		124
210		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	IFDF	CNDS18	
	0580	063160		SDVR=PTS42+PMODE	125
211	00V2	053160	WORD	apyk Flatetriuvs	124
212			. ENOC		
213			IFNDF	CND518	
214			WORD	Ø	125
215			, ENDC		
216	e e		. IFOF	CNDS4	
217	0554	003026	WORD	AINT-PT542	126
218	** * * *		ENDC		
219			IFNDF	CNDS4	
					106
220			.WORD	0	126
221			. ENDC		
222		000000	WORD	0,0,0,0,0,0,0	127=35
		000000			
	0562	000000			
	0564	000000	and the second second		
	0566	000000			
		000000			
		000000			
223	90r 6	000000	. IFOF	CND\$37	
	11 m 7 4				136-37
224		017766	, WORD	SIN-PT\$42,COS-PT\$42	130437
	05/5	017732			
225			, ENDC		
226			. IFNDF	CNDS37	
227			WORD	0,0	135=37
228			ENDC		
229			IFDF	CNDS13	
	(1. 6: 13 m	107651	WORD	DSIN=PTS42+DMODE	140
230		107654			141
231	0002	107576	.WORD	DCUS=PT\$42+DMODE	7. ₩.₩
232			.ENDC		
233			. IFNDF	CND\$13	
234			. WORD	0,0	140-41

235			. ENDC	0.40 × 10	
236	Ch. 40 CR. 4		. IFOF	CND\$39	142
237	0004	021062	.WORD	ATAN=PT542	/44
238			.ENOC		
239			. IFNDF	CND\$39	
240			. WORD	0	
241			. ENDC		• • *
242	0605	000000	.WORD		143
243			, IFOF	CND\$15	
244	0610	111040	, WORD	DATAN-PT\$42+DMODE	144
245			. ENOC		
246			. IFNDF	CND\$15	
247			. WORD	0	144
248			, ENDC		
249	0612	000000	WORD	0	145
250	·	-	, 1FOF	CND\$41	
251	0614	021552	. WDRD	SURT-PT\$42	146
252			. ENDC	A. 1. A. A. A.	· · · ·
253	e.		. IFNDF	CND\$41	
254			. WORD	0	146
255			. ENDC		
256			. IFDF	CND514	1 15
257	0616	110356	, WORD	DSGRT=PT\$42+DMODE	147
258			. ENDC		
259			, IFNDF	CNDS14	
260			, WORD		147
261		· · · · ·	, ENDC	(1) When the second se second second sec	
262			. IFDF	CND\$38	a an c'h
263	0620	020306	. WORD	TANH=PT\$42	150
264			. ENDC		
265			. IFNDF	CNDSJB	
266			.WORD	Ø	150
267			, ENDC		
268			.IFDF	CND\$20	
269	0622	014456	.word	EXP-PT\$42	151
270			. ENDC		
271			. IFNDF	CND\$20	
272			.WORD		151
273			, ENDC		
274			.IFOF	CNDS19	14 - 14 ⁻
275	0624	113612	.WORD	DEXP-PT\$42+DMODE	152
276			. ENDC		
277			, IFNDF	CND\$19	
278			.WORD	0	152
279			. ENDC		
280			. IFDF	CND33	
281	0626	002452	. WORD	ALOG-PT\$42	153
282		002445	WORD	ALOG10-PT\$42	154
283			, ENDC		
284			. IFNDF	CND\$3	
285			WORD	0,0	153-54
286			ENDC		
287			IFOF	CNDS10	
288	0632	106556	WORD	DLOG=PT\$42+DMODE	155
289	0634	106552	, WORD	DLOG10-PT\$42+DMODE	156
290	··· ··· ·		ENOC		
291			IFNDF	CND\$10	

۴

÷

TRAP02 MACRO VR04=14 07-SEP=72 11:43 PAGE 6+

φ

292				. WORD	0,0			155-56		
293				. ENDC						
294	0636	000000		WORD	0.0.0.0	,0,0,0,0,	0.0	157-70		
	0640	000000		•						
	0642							5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 -		
	0644									
	0646									
		000000								
		000000								
		000000								
	0654									
		000000								
	0660	000000								
295				, IFOF	CNDS44	· · · · · · · · · · · · · · · · · · ·				
296	0662	061746		. WORD	SLOR-PT	542+PMODE	•	171		
297				. ENOC						
298				. IFNDF	CND344					
299				. WORD	0			171		
300				ENDC						
301				IFOF	CND345					
302	1664	161760		WORD		\$42+PMODE	+DMODE	172		
303	200 W N	101.00		ENDC						
				IFNDF	CND345					
304					0			172		
305				.WORD	U			// 5		
306				.ENDC	0.40×40					
307				. IFOF	CND\$46	AA DHADE			-	
308	0606	062000		. WORD	221K-PT	\$42+PMODE	• · · · · · · · · · · · · · · · · · · ·	173		
309				.ENDC						
310				IFNDF	CND\$46					
311				.WORD	Ø			173		
312				. ENOC						
313				.IFOF	CND547					
314	0670	162054		WORD	SSTD-PT	\$42+PMODE	+DMODE	174		
315				ENDC						
316				IFNDF	CND\$47					
317				WORD	Ø			174		
318				ENDC	4.00°					
319	0670	000000		WORD	0,0,0			175-77		
V. V		000000		10 ··· world 100	~ ~ ~ ~			* * ···· * *		
		000000								
300	0010	0000000		ENAC						
320				. ENDC	e A Para (A e					
321				TITLE						
322				. IFDF	CNDS1					
323			.	.GLOBL	SAUD 55	BD, SERR			N 65	
324)			OUBLE PRE	CISION /	AND KUNII	NE	
325			1	SADD	V005A					
326			1 .	COPYRIG	IT 1971,	1972 DIGI	TAL EQU;	IPMENT CO	RP., MAYN	IARD, MA
327			1			CK ITEM T			M	
328			1			UM IN THE				
329			;			OUBLE PRE				
330			1	SUBTRAC	T THE TO	P STACK I	TEM FROM	4 THE SEC	OND ITEM	
331			1	AND LEAT	E THE D	IFFERENCE	IN PLAC	CE OF THE	M	
332		000000	•	R0=%0						
333		000001		R1=%1						
334		000002		R2=%2						
335		000003		R3#%3						
336		0000004		R4=%4						
330 337		0000005		R5=%5						
93/				5 J = A J						

//

13 13 (3		nnakae		50-4 <i>6</i>						
338		000006		SP=%6						
339		000007		PC=%7						
340		000006		A185						
341		000010		B1=8.						
342		000012		C1=10,						
343		000014		D1=12,						
344		000015		A2#14.						
345										
		000020		B2=16.						
346		000022		C2=18.						
347		000024		D2=20.						
348		000000		SIGNSEØ						
349		177304		MQ#1773	04					
350		177312		NOR=177	312					
351		177314	<i>6</i> .	LSH=177						
352		177316		ASH=177						
353		000000		FØRXØ						
354	0700		SSBDI	ADD	#100000	A C C	INFRATE	TOP ST	ACK ITE	4
90M	0/00		294U.	AUV	*********	1.00	THEORIC	101 01	HUN TIC	7
No		100000			1735 million 1					
355				. IFOF	FPU	1				
356			SAUDI	WORD	170011	JISETD	,			
357				WORD	172426	IILDD	(SP)+,F0	0	IGET C	PERAND
358				WORD	172026	TIADDD	(SP)+,F0		FADD	
359				WORD	174046	IISTD	F0, - (SP)			TO STACK
360				JMP	@(R4)+		1.61	•	/ / / / /	
					* (n 42 *					
361				.ENDC	Re ma k i	en e	n an			14 (17 (17 (17 (17 (17 (17 (17 (
362		10 A 10 A 15		. IFNDF	FPU	•				
363	0704	010446	SAUDI	MOV	R4, - (SP					
364	0706	010546		MOV	R5,=(SP)				
365	0710	005046		CLR	=(SP)	ICLEAR	SIGNS			
366	0712	005004		CLR	R4		EXPONENTS	3		
367	0714	005005		CLR	R5			- · ·		
368	0716	006366		ASL	01 (SP)	ISHTET	OUT SIGN		TTEM	
300	0110			MOL	0110-1	1 outra	UUT STON	wr iwr	4 1 5 11	
3 14 13		000014		(2 A)	61 (A B A					
369	0/22	006166		ROL	C1(3P)					
	. ~	000012		<i>.</i>						
37Ø	0726	006166		ROL	81(SP)					
		000010								
371	0732	006166		ROL	A1(SP)	ISHIFT	A1			
		000000			· · · ·					
372	0736	156604		BISB	A1+1(5P)	.R4	JGET E1			
W 7 94		000007					· · · · · · · · · · · · · · · · · · ·			
373	0742			BEQ	A 1 7 @ 1	JUMP I	5 7F20			
		001441			A1251					
		106116		ROLB	PSP	JGET S1				
375	0746	006366		ASL	D2(SP)	ISHIPT	OUT SIGN	UF SEC	UND ITER	
		000024								
376	0752	006166		ROL	C2(SP)					
		000022								
377	0756	006166		ROL	82(SP)					
	··· · · ·	000020		े आ सम्ब						
378	17K0	006166		ROL	A2(SP)	ISHIFT	12			
0/0	4/46			NOF .	ACTO.1	100701	AG			
2	() 19 K #	000016		Д т с Ю						
379	0/00	156605		BISB	A2+1(SP)	CNIC	JGET E2			
		000017								
380		001030		BNE	A2N51		F NOT Ø			
381	0774	106016		RORB	ØSP	IRECONS	TRUCT A1		• • • • • •	
382	0776	006066		ROR	A1(SP)					
		000006		*						

SADD05 MACRO VR04=14 07=SEP=72 11143 PAGE 6+

383	1002	006066		ROR	81(SP)					
384	1006	000010 006066		ROR	C1(SP)					
004		000012								
385	1012	006066		ROR	D1(SP)					
106	1010	000014		MOV	A1 (SP) ,	A2(8P)	FIRST	ARG TO	TOP OF	STACK
200	1010	016666		n Q Y	M1.(0,)/		81 ALV #1		in nart mit	.
		000016								
387	1024	016666		MOV	81(SP),	B2(SP)				
		000010 000020								
388	1032	016666		MOV	C1(SP),	C2(SP)				
		000012								
		000022		MoN	SI CORA	02(80)				
397	1040	016666		MOV	01(SP),	URIGFI				
		000024					• · · ·			
		005728	A12511	TST	(SP)+	IFLUSH	SIGNS			
391	1050	000167		JMP	OUTSI	IDONE				
302	1054	000476	AONELL	ROLB	SIGNS+1	(SP)	JGET S2			
036	1044	000001	76''Ø4'				,			
393	1000	112766		MOVB	#1, A2+1	(SP)	INSERT	NORMAL	BIT	
		000001								
304	1	000017		MOVB	#1,A1+1	(SP)	ITNSFRT	NORMAL	RTT	
394	1000	112766 000001		rių vo	***	(en J	1 TUARU	rin in Arciana≯	na beta≱1,	
		000007								
		160405		SUB	R4, R5	IRS=E2	E1, R4=E	1		
		003011		BGT Mov	EXASI A2(SP),		IF E2>E1 ; R0=A2			
397	1100	016600 000016		TIUY	MELONJI		104206			
398	1104	016501		MOV	B2(SP),	R1	1R1=82			
		0000020								
399	1110	016602		MOV	C2(SP),	RZ .				
400	1114	000022		MOV	D2(SP),	RJ				
400	* 1 * 4	000024								
		000427		BR	SCKS1		CK SIGNS			
		060504	EXASII	ADD	R5,R4		E1,R4=E2 ;R0=A1	2,02401		
403	1124	016600 000006		MOV	A1(SP),		104404			
404	1130			MOV	81(SP),	R1	IR1=B1			
	·	000010				- 0				
405	1134			MOV	C1(SP),	KZ				
406	1140	000012 016603		MOV	D1(SP),	R3				
~~~		000014								
407	1144			MOV	A2(SP),	A1(SP)				
		000016								
408	1152	000000 016666		MOV	82(SP),	B1(SP)			1997) 1997) 1997)	
	4 1 M K	000020	· · · · ·	· · · · · · · · · · · · · · · · · · ·						
		000010								
409	1160			MOV	C2(SP),	C1(SP)				
		000022								

		000012				
				MAN	OBIERS	1100
410	1100	016666		MOV	D2(SP),	
		000024				
		000014				
		000316		SWAB	0 S P	JEXCHANGE SIGNS
412	1176	005405		NEG	R 5	1E1=E2
		126616	SCKS1:	CMPB	SIGNS+1	(SP), @SP ; CUMPARE SIGNS
76.4		000001				
A 4 A	1004	001412		BEQ	ECK\$1	THEY'RE THE SAME, CHECK EXPONENT
		005403		NEG	R3	INEGATE OPERAND
		005502		ADC	R2	
417				ADC	R1	
418	1214	005500		ADC	RU	
419	1216	005402		NEG	R2	
420	1220	005501		ADC	R1	
	1222			ADC	RØ	
	1224			NEG	R1	
423		005500		ADC	RØ	
424		005400		NEG	RØ	
				TST		TOURAY EVONSENTO
		005705	CURSTI		R5	ICHECK EXPONENTS
	1234			BEQ	SFD\$1	JJUMP IF E1=E2
427	1236	022705	SFISIS	CMP	#=57 ., R	5 JIS THERE ANY POINT IN SHIFTING?
		177707				
428	1242	003411		BLE	SFR\$1	IYES
429	1244	016600		MOV	A1 (SP) ,	RU JND, ANSWER IS OPERAND
		000005				and the first of the second
430	1250	016601		MOV	B1(SP),	RI :WITH THE LARGER EXPONENT
		000010				
431	1054	016602		MOV	C1(SP),	2
401	1244	5		115 <b>9</b> V	0110.11	
	1 m 61	000012		1 A A A		
432	1500	016603		MOV	D1(SP),	<b>KO</b>
		000014				
433	1264	000504		BR	NUD\$1	
434	1266	022705	SFRS11	CMP	#=8,1R5	ICHECK # OF BITS TO SHIFT
		177770				
435	1272	003442		BLE	SR8\$1	JJUMP IF NOT MORE THAN 1/2 WORD
436	"		N.	. IFNDF	MULDIV	성영화 그 가슴 그는 것을 알려요. 그는 것은 것을 얻는 것을 못했는 것이 없다.
437	1274	005046		CLR	= (SP)	ISET UP EXTENSION BITS
438	1276	005700		TST	RØ	FACCORDING TO HIGH ORDER FRACTION
	-			BPL	SF151	JJUMP IF +
439	1300	100001				an a
	1302	005116		COM	ØSP	and the second
441				ENDC	*****	
442				.IFDF	MULDIV	
443				TST	RØ	<ul> <li>A set of the set of</li></ul>
444				.WORD	006746	I; SEX - (SP) / EXTEND SIGN
445				.ENDC		
	1304	022705	SF151:	CMP	#=16.,R!	$\mathbf{y}$ is the second s
		177760		-	900 V C B	
417	1310	002411		BLT	S1651	JJUMP IF NOT MORE THAN A WORD TO SHIFT
		010203		Mav	R2,R3	JSHIFT A WORD AT A TIME
						이 같은 사람들은 것이 같아요. 이 것은 것은 것은 것이 같아요. 이 있는 것이 같아요. 이 것이 같아요. 이 있는 것이 없는 것이 같아요. 이 있는 것이 있는 것이 같아요. 이 있는 것이 같아요. 이 있는 것이 없는 것이 않이 않는 것이 있는 것이 않는 것이 않이
		010102		MOV	R1, R2	
		1 × 1 × 1 × 1 × 1		M9 (*1 M	RØ,R1	
450	1316	010001		Mav		المراجع
451	1316 1320	011600		MOV	esp, Kø	JUSE EXTENSION
451	1316 1320	011600 062705				JUSE EXTENSION JADJUST EXPONENT
451	1316 1320	011600		MOV	esp, Ka	
451 452	1316 1320 1322	011600 062705 000020		MOV	esp, Ka	
451 452 453	1316 1320 1322 1326	011600 062705		MOV ADD	05P,KØ #16,/R5	JADJUST EXPONENT

# \$ADD05 MACRO VR04=14 07=SEP=72 11:43 PAGE 6+

				· . •		
455	1332	000430		BR	SFD\$1	ISHIFT IS ALL DONE
456		000-00		IFOF	EAE	a na cramera 1 mai na cana ana ana ana ana ana ana ana ana
457			S16511	CMP	#=3,R5	
458			010011	BLE	\$8A\$1	JUMP IF NOT MORE THAN 3 TO SHIFT
				MOV	R4,eSP	ISAVE EXP
459				MOV	#MQ,K4	POINT TO MQ
460				MOV	R3.eR4	ILOW ORDER PARTS TO AC, MO
461				MOV	R2,=(R4)	
462				MOV	R5,0#LSH	
463						
464				MOV	(R4)+,R2	LOWEST ORDER IS DONE
465				MOV		ILUMEST OKAEK TO DAVE
466				CLR	PR4	- ACT UN NEVT UTAUER MADE
467				MOV	R1,=(R4)	
468				MOV	R5,0#LSH	
469				TST		FOINT TO MQ
470				BIS		IFINISH R2
471				MOV	R1, ORA	
472				MOV	R0,=(R4)	
473				MQV	R5, @#ASH	
474				MOV	(R4)+,RØ	HIGH ORDER DONE
475				MOV	PR4,R1	
476				MOV	(SP)+,R4	
477				BR	SFD\$1	
478				. ENDC		
479		•	÷	. IFNDF	EAE&MULD	IV CARACTER STATES
480	1334	022705	51051:	CMP	##8 . 1R5	
		177770				
		003416		BLE	S8A51	JUMP IF NOT MORE THAN 1/2 WORD TO GO
482	1342	062705		ADD	#16,/85	ISHIFT LEFT 16=X
		000020				
483	1346	006303	SL8S1:	ASL	R3	ISHIFT LEFT
484	1350	006102		ROL	R2	
485	1352	006101		ROL	R1	
486	1354	006100		ROL	RØ	
487	1356	006116		ROL	@SP	
488	1360	005305	· · · · · ·	DEC	R5	ICOUNT LOOP
489		003371		BGT	SL851	
		010203		MOV	R2,R3	
		010102		MOV	R1, R2	
		010001		MOV	RØ,R1	
		012600		MOV	(SP)+,RØ	
494	1374	000407		BR	SFD51	ISHIFT DONE
495				.ENDC		
496				IFDF	MULDIV	
497			S1551:	CMP	#=3,85	JUMP IF NOT MORE THAN 3 TO SHIFT
498			₩°¢ēā.	BLE	SBAS1	
499				MOV	R4,0SP	ISAVE EXP AND SHIFT COUNT
500				MOV	R5,=(SP)	
501			i en la la	MOV	R1, R4	ISAVE R1
502				WORD	073005	1; ASHC R5, RØ ISHIFT HIGH ORDE
503				MOV	R2,R5	ISAVE R2
504				WORD	073416	II ASHC @SP,R4 ISHIFT IT
505				MOV	R2, R4	ਤ ਦਾ ਸਾਮਲਾਂ ਜਦਾ ਦਾ ਸਾਮਲਾਂ ਦਾ ਸਾਲ ਹੈ। 
506				MOV	R5, R2	IR2 DONE
507				MOV	R3, R5	ISET UP LOW ORDER
508				WORD	073426	II ASHC (SP)+,R4 IDQ LOW
500				MOV	R5,R3	বার নেকারের সম্পর্যয়ানের রাজনেরেরের 
203				in <b>y</b> ₹	untua	

\$AD005	MACRO	VR04-14	07=SEP=72	11143	PAGE 6+
---------	-------	---------	-----------	-------	---------

510			MOV	(SP)+,R4 ;RESTORE EXPONENT TO R4
511			BR	SFD\$1
512		e e e general e general	.ENDC	
513	1376	005726 S8A\$1		(SP) + IPOP EXTENSION
514	1400	006200 SR851		RØ ISHIFT RIGHT
515	1402	006001	ROR	R1
516	1404	006002	ROR	R2
517	1406	006003	ROR	R3
	1410	005205	INC	R5 JCOUNT LOOP
518			BLT	SR851
519	1412	002772		
520	1414	066603 SFUS1	I ADD	D1(SP),R3 JFORM THE SUM
-		000014	4.0.0	
521	1420	005502	ADC	R2
522	1422	-	ADC	R1 ends and a second
523	1424	005500	ADC	RØ
524	1426	066602	AOD	C1(SP), R2
		000012		
525	1432	005501	ADC	t R1 - Constant and the second se
526	1434	005500	ADC	RU
527	1436	066601	ADD	B1(SP),R1
		000010		
528	1442		ADC	RØ
529	1444		ADD	A1 (SP), RØ
~ ~ ~	است استغد	000006		
530	1450	126616	Смрв	SIGNSII(SP), OSP JCHECK FOR UNEQUAL SIGNS
000	****	000001		
531	1454	001065	BNE	SUBS1 JGO CLEAN UP SUBTRACT
532	1456	030027	BIT	RØ, #1000
9 Q &	*****			
<b>1</b> 13 12	1.160	001000	BEQ	NODS1 JJUMP IF NO NORMAL BIT OVERFLOW
533		001405	ASR	RØ
534		006200		
535	1466		ROR	R1 Contraction of the second
536	1470	006002	ROR	R2
537	1472	006003	ROR	R3
538	1474	005204	INC	R4 /INCREASE EXPONENT
539	1476	000304 NODS1		R4 IMOVE EXPONENT LEFT
540	1500	001031	BNE	OVFS1 JJUMP IF OVERFLOW
541	1502	150004 NFLS1		RØ,R4 JINSERT HIGH ORDER FRACTION
542	1504	006026	ROR	(SP) + JINSERT SIGN
543	1506	006004	ROR	R4
544	1510	006001	ROR	R1 A second state of the second state of th
545	1512	006002	ROR	R2
546	1514	006003	ROR	R3
547		005503	ADC	R3
548	1520	005502	ADC	R2
549		005501	ADC	R1
550	1524	005504	ADC	R4
550		102417	BVS	OVRS1 JJUMP IF OVERFLOW UN ROUND
552		103416	BCS	OVRS1
		010466	MOV	R4,A2+0-2(SP) ISTORE EXPONENT AND SIGN
553	1045		14 U A	. Γ
604		000014	MAM	R1,82+0-2(SP) ;INSERT LOW ORDER FRACTION
554	1999	010166	MOV	UPIDEANERIALI ITHOEVI PAN MAREK EKVELTAN
		000016	Xá m Xá	00.0.0.0.0
555	1542		MOV	R2,C2+0=2(SP)
		000020		
556	1546	010366	MOV	R3, D2+0=2(SP)
		000022		

(

#### SADD05 MACRO VR04=14 07=SEP=72 11143 PAGE 6+

ð

				44.0014	440.0	
55	7 1552	012005	OUTSII	MOV	(SP)+,R	
55	8 1554	012004		MOV	(SP)+,R	4
55				ADD	#8, SP	JPOP SECOND ARGUMENT
99	* 1040			~~~	444146	ALOL AMPANA WURKING
		000010				
56	0 1562	000134		JMP	@(R4)+	IDONE, RETURN
56			1			
			•		4004	A GARAN A MALI
56	2 1564	005726		TST	(SP)+	IPOP SIGN
56	3 1566	004567	OVKS1:	JSR	R5, SERR	IERROR 3.1
		020214				
				0.0	(m.) (199 és 1	
56		000767		BR	00751	
56	5 1574	003		BYTE	3	
56	17 · · · ·	001		BYTE	1	
			lim Gada			LEURA BAR HAMPER DE CL
56		005704	UTSSII	TST	R4	ICHECK FOR UNDERFLOW
56	8 1600	003336		BGT	NODS1	
56	9 1602	004567	UNFS18	JSR	R5.SERR	JERROR 5,1
-		020200				
				0.11		
57	0 1606	000401		BR	UNDS1	
57	1 1610	005		BYTE	5	
57		001		BYTE	1	
57	3 1612	005000	UNDSI	CLR	RØ	
57	4 1614	005001		CLR	R1	JUNDERFLOW. TREAT AS 0
57		005002		CLR	R2	
57		005003		CLR	R3	
57	7 1622	005016	ZEKSII	CLR	esp	ISET SIGN PLUS
57		005004		CLR	R4	
57	9 1626	000725		BR	NFLS1	FINISH OUT NURMALLY
58	Ø		1			
	1 1630	005700	SUBSII	TST	RØ	ICHECK HIGH ORDER RESULT FRACTION
			40-011			
58		003015		BGT	BT951	FIF POSITIVE SIGN IS OK
58	3 1634	001425		BEQ	ZTSS1	ICHECK FOR ZERO RESULT
58	4 1636	005403		NEG	R3	IGET ABSOLUTE VALUE
58	· • •	005502		ADC	R2	
58	6 1642	005501		ADC	R1	
58	7 1644	005500		ADC	RØ	
58		005402		NEG	R2	
58		005501		ADC	R1	
59	0 1652	005500		ADC	RØ	
59		005401		NEG	R1	
59		005500		ADC	RØ	
59	3 1660	000316		SWAB	PSP	JEXCHANGE SIGNS
59	4 1662			NEG	RØ	
						CHECK FOD ZEDA DESIL T
59		001411	1.5 MB (3) es 4 m	BEQ	ZT551;	CHECK FOR ZERO RESULT
59			BT9311			
59)	7			, IFDF	EAE	
59	R			BIT	R0,#740	
						6 FILLED THE LIDE LIDESE THILLS A MAN ASSEMBL
59				BNE	B9A51	JUMP IF NOT MORE THAN 4 TO SHIFT
601	0			MQV	R4, = (SP)	ISAVE EXP
60				MOV	#MQ, RA	POINT TO MO
60				MOV	R1, ORA	ILOW ORDER FRACTION TO MQ
60	3			MOV	R0, =2(R4	I) JHIGH ORDER FRACTION TO AC
60				CLR	ØNOK	INORMALIZE
60				MOV	@#NOR,=(	
60	6			SUB	#6,0SP	ICOMPENSATE FOR NORMAL BIT POSITION
60				MOV		JGET 2 HIGH ORDER PARTS
60				MOV	R0, = (R4)	
6 12				MON	କର୍ପ୍ତ କିଧାର	
60	9			MOV	@SP, @#LS	H ISHIFT THEM

611			MOV	PR4, R1 / ISAVE PARTIAL R1
612			MOV	R2, ORA IGET NEXT
613			CLR	»(R4)
614		a tai at	MOV	#SP, ##LSH ; SHIFT IT
615			BIS	
616			MOV	R3,0R4 /GET NEXT
617			MOV	$R_{2} = (R_{4})$
618			MOV	esp, exist istift it
619			MOV	(R4)+,R2 SFINISH R2
			MOV	PRA, R3 IR3 DONE
620				
621			SUB	(SP)+, OSP ICOMPENSATE EXPONENT
622			MOV	(SP)+,R4 IRESTORE IT TO R4
623	1		BGT	NODS1 JJUMP IF NO UNDERFLOW
624		w.,	BR	UNF\$1
625			.ENDC	
	566 030027	B9A511	BIT	R0,#400 /CHECK NORMAL BIT
050 10		~3~@\$*	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	a china a china a china manazi a sa sa manazi a sa sa sa sa sa
	000400		199 1024	
	572 001341		BNE	UTSS1 JJUMP IF FOUND
628 10	574 005304		DEC	RA IDECREASE EXPONENT
	576 006303		ASL	R3 FOOUBLE FRACTION
	00 006102		ROL	R2
			ROL	R1
	02 006101			
	04 006100		ROL	RØ
	706 000767		BR	B9AS1 JTRY AGAIN
634 17	10 162704	275511	SUB	#8,, R4 IREDUCE EXPONENT
	000010			
635 17	14 005701	-	TST	R1 States and president and the states of
	16 001020		BNE	ZT181 JJUMP IF ONLY ROAD
637 17	20 162704		SUB	#16. / R4
	000020			
638 17	124 010201		MOV	R2, R1
639 17	26 001012		BNE	ZT251 JJUMP IF R2 NOT 0
	30 162704		SUB	#16,,R4
~~~ **	000020			
e			TOT	
	134 005703		TST	R
642 17	36 001731		BEQ	ZERS1 JANSWER IS 0
643 17	740 150301		BISB	R3,R1 IMOVE BYTES TO R0,R1
644 17	42 000301		SWAB	R1
	44 000303		SWAB	R3
	46 150300		BISB	R3, RU
	50 005003		CLR	R3 IMAKE ALL OTHERS 0
	152 000745	18 N. 19	BR	BT951 JGO NORMALIZE
649 17	54 010302	ZT2511	MOV	R3, R2
	56 005003		CLR	R3
	100 000301	ZT151:	SWAB	R1 IMOVE ALL BYTES LEFT
	62 150100		BISB	R1,R0
			CLRB	R1
	64 105001			
	66 000302		SWAB	R2
	70 150201	general de la composición de	8188	R2,R1
656 17	72 105002		CLRB	R2
	74 000303		SWAB	R3
	76 150302		BISB	R3, R2
	*		CLRB	R3
	100 105003	State and St		
	102 000731		BR	BT951 JGO NORMALIZE WHAT'S LEFT
661			. ENDC	
662			.ENDC	

m

÷

SADR04 MACRU VR04=14 07=SEP=72 11:43 PAGE 7

 $\left(\right)$

13

 $\left(\right)$

-

ŵ

				.TITLE .IFDF	SADRØ4 CNDS2				
				GLOBL		R.SERP			
				SADR			ROUTINE		
			:	SADR	V004A		a la presidente de la presidencia.	,	
			1		T 1071.	DIGITAL	FOUTPMENT	CORP. M	YNARD, MASS.
			<u>'</u>	DOPINIO	11 19/11 THE THO	TTEMS (N TOP OF T	HE STACK	aturnet need
			1			ALCHO .	IN IOF OF I	DE GIOCA	
			1	WITH THE				THE	
			1				STRACT ROUT		
)			1	SUBTRACT	T THE TOP	STACK	ITEM FROM	THE SECUNI	
			1		THEM BOT	M WITH	THE DIFFER	ENCE	2 - 2 4 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 -
		000000		R0=%0					
		000001		R1=%1					
		000002		R2=%2					
		000003		R3#%3					
		000004		R4=%4					
r		000005		R5#%5					
		000005		SP=%6					
		000007	ter at en de se	PC=%7					
		000000		SIGNSEO					
		000004		A1=4					
		000006		B1 46	an sa sa	т. 			
		000010		A2#8					
		000012		B2=10.					
		177302		AC#17736	20				1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1
				MQ=17730					
		177304		NOR#1773					
		177312							
		177316		ASH=1773	910				
		000000		FØRXØ	41 7 7 6 6 6		ACHANCE T	UP STRU OF	TOP ITEM
	02004	062716	SSDRI	ADD	#100000,	40F	ICHANGE I	HE STAN U	<u>nur lien</u>
		100000		-	-				
				. IFDF	FPU				
			SADRI	WORD		IISETF	.		
				.WORD		TILDF			ET OPERAND
				.WORD		ITADDE		1 A G	
				WORD		IISTF	FØ,=(SP)	1.51	JM TO STACK
				JMP	@(R4)+				
				. ENDC					
				IFNDF	FPU				
	a2a1a	010446	SADR:	MOV	R4,=(SP)				
		005046		CLR	= (SP)		SIGNS		
	02014			CLR	R2		EXPONENTS		
	02016	005003		CLR	RJ	а. — та			
	02020			ASL	81 (SP)	ISHIFT	81		
	5 6 9 6 9	000006							
	02024			ROL	A1(SP)	ISHIFT	A 1		
	06064				MI (OF)	1 Out 1	M #		
		000004		0 # # M					
				BISB	A1+1(SP)	100	IGET E1		
	02030			2 S			ten ve C'es es		
		000005		BEQ	OUTS2		LF ZERO		
	02034	001574	• · · · · · · · ·	Kon and the second	an 14 13	JGET SI	•		
	02034 02036	001574 106116		ROLB	ØSP				
ł	02034	001574 106116	,	ASL	82(SP)	ISHIFT			
	02034 02036	001574 106116		ASL	82(SP)	ISHIFT	B2		
ì	02034 02036 02040	001574 106116 006366		ASL			B2		
	02034 02036 02040	001574 106116 006366 000012		ASL	82 (SP) A2 (SP)	ISHIFT ISHIFT	B2 A2		
	02034 02036 02040 02040	001574 106116 006366 000012 006165		ASL	82(SP)	ISHIFT ISHIFT	B2		

SADR04 MACRO VR04=14 07=SEP=72 11:43 PAGE 7+

51	02054	001014		BNE	A2N52	JJUMP I	F NOT 0				
52		106016		RORB	PSP		TRUCT A1,	81			
53		006066		ROR	AL (SP)			-			
	0	000004		1. 1 . 1.		n in the second seco					
54	ADAKA			ROR	B1(SP)	i isaa k					
54	() C () C C	006066		nyn	0110-1						
		000006		A. 76 1 E				0			
55	02010	016666		MOV	A1 (SP) , A	AZ(SP)	IFIRST A	AG TU	IOP OF	STACK	
Æ		000004									
	1	000010				n safa Ingel					
56	02076	016566	le et le t	MOV	81(SP),6	32(SP)					
		000006	1. () y	4 (C) (BA	4 9 V						
		000012									
57	02104	000550		BR	00132	IDONE	Real Action				
58		106166	A2N52:	ROLB	SIGNS+1		IGET S2				
	24 79 7 79 79 79	000001	2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				• • • • • • • • •				
59	09110	112766		MOVB	#1, A2+1	SPY	INSERT	NORMAL	ATT		
33	Ne1+2			NUVD	441 W		1 TUDEN I		911		
		000001									
-	<i></i>	000011									
60	02120	112766		MOVB	.#1,A1+1((3 P)	INSERT	NORMAL	BIT		
		000001									
		000005					2 N - 1				
61	02126	160302		SUB	R3, R2	1R2=E2=	E1, R3=E1				
62		003005		BGT	EXAS2	JJUMP I	F E2>8E1				
63		016600		MOV	A2(SP), P		1R0=42				
	97 m e * m	000010		· · •							
64	02136	016601		MOV	82(SP), P	21	1R1=82				
V 4	06100	000012		HUU Y	netailti	1 1					
66	00140			ün	C B V B Q	IPUECV	PTONE				
65		000415	11. A & C A	BR	SCKS2	ICHECK		Fast			
66		060203	EXASEI	ADD	R2,R3		E1,R3=E2,	E27E1			
67	02146	016600		MOV	A1 (SP) F	(U)	IRB=A1				
		000004				14					
68	02152	016601		MOV	B1(SP), F	1	1R1=81				
		000006									
69	02106	016666		MOV	A2(SP),A	1(SP)					
	·····	000010				and the second					
	- 1	000004									
70	0216A	016666		MOV	82(SP),8	I (SP)					
10	NETAN			HQ Y	06(0)))						
		000012									
		000006		6							
		000316		SWAB	PSP		GE SIGNS	e de set	1		
72	02174	005402			R2	1E1=E2				and the second second	
73	02175	126616	SCKS21	CMPB	SIGNS+1 ((SP), PSP	ISEE IF	SIGNS	ARE THE	SAME	
		000001									
74	02202	001403		BEQ	ECK\$2	IYES, C	HECK EXPO	NENTS			
		005401		NEG	R1		FRACTION				
		005500		ADC	RØ						
		005400		NEG	RØ						
		005702	FAKEDI	TST	R2	1					
			54 L 17 Ø FE 8			THIND T	E 61-50				
		001450	0 m T & C =	BEQ	SFDS2			S ALIN P			6.0
80	05510	022702	341921	CMP	#=25.,R2		TTO THER	G ANT I	UTNI T	N SHIFTIN	97
n -	-0-0-	177747		0. E	0 m m + D	1.45 c					
		003405		BLE	SFR52			50 m m m m m			
82	02224	016600		MOV	A1(5P),R	e e e e e e e e e e e e e e e e e e e	IND, ANS	HEN IS	OPERAN	U	
		000004				a series de la s	-	-	_		
83	02230	016501		MOV	B1(SP),R	1	JWITH TH	E LARGE	ER EXPO	NENT	
		000000									
84	02234	000456		BR	NOD\$2						
	· · ·										

¢,

(

 \sim

<pre>JMOVE FRACTION TO AC,MQ JSHIFT RIGHT TO EQUALIZE EXPONEN JRECOVER SHIFTED FRACTION JASHC R2,R0 IV JCHECK # OF BITS TO SHIFT JJUMP IF NOT MORE THAN 1/2 WORD JSET UP EXTENSION BITS JBASED ON HIGH ORDER FRACTION JJUMP IF + J= OTHERWISE JJUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16-X JCOUNT LOOP JPUT RESULT IN R0, R1 JSHIFT A MIN AND B MIN</pre>
<pre>>>SHIFT RIGHT TO EQUALIZE EXPONEN PRECOVER SHIFTED FRACTION >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>
<pre>####################################</pre>
<pre>####################################</pre>
<pre>####################################</pre>
IV JUMP IF NOT MORE THAN 1/2 WORD JUMP IF NOT MORE THAN 1/2 WORD JUMP IF EXTENSION BITS JUMP IF + TOTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JUMP IF LEFT 16=X JCOUNT LOOP JPUT RESULT IN RØ, R1
IV JUMP IF NOT MORE THAN 1/2 WORD JUMP IF NOT MORE THAN 1/2 WORD JUMP IF EXTENSION BITS JUMP IF + TOTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JUMP IF LEFT 16=X JCOUNT LOOP JPUT RESULT IN RØ, R1
IV JUMP IF NOT MORE THAN 1/2 WORD JUMP IF NOT MORE THAN 1/2 WORD JUMP IF EXTENSION BITS JUMP IF + TOTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JUMP IF LEFT 16=X JCOUNT LOOP JPUT RESULT IN RØ, R1
IV JUMP IF NOT MORE THAN 1/2 WORD JUMP IF NOT MORE THAN 1/2 WORD JUMP IF EXTENSION BITS JUMP IF + TOTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JUMP IF LEFT 16=X JCOUNT LOOP JPUT RESULT IN RØ, R1
IV JUMP IF NOT MORE THAN 1/2 WORD JUMP IF NOT MORE THAN 1/2 WORD JUMP IF EXTENSION BITS JUMP IF + TOTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD TO SHIFT JUMP IF LESS THAN ONE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JUMP IF LEFT 16=X JCOUNT LOOP JPUT RESULT IN RØ, R1
<pre>/CHECK # OF BITS TO SHIFT /JUMP IF NOT MORE THAN 1/2 WORD /SET UP EXTENSION BITS /BASED ON HIGH ORDER FRACTION /JUMP IF + /= OTHERWISE /JUMP IF LESS THAN ONE WORD TO SHIFT /SHIFT RIGHT A WHOLE WORD /USE EXTENSION BITS /ACCOUNT FOR SHIFT /JUMP IF NOT MORE THAN 1/2 WORD /SHIFT LEFT 16=X /COUNT LOOP /PUT RESULT IN RØ, R1</pre>
<pre>/CHECK # OF BITS TO SHIFT /JUMP IF NOT MORE THAN 1/2 WORD /SET UP EXTENSION BITS /BASED ON HIGH ORDER FRACTION /JUMP IF + /= OTHERWISE /JUMP IF LESS THAN ONE WORD TO SHIFT /SHIFT RIGHT A WHOLE WORD /USE EXTENSION BITS /ACCOUNT FOR SHIFT /JUMP IF NOT MORE THAN 1/2 WORD /SHIFT LEFT 16=X /COUNT LOOP /PUT RESULT IN RØ, R1</pre>
JUMP IF NOT MORE THAN 1/2 WORD JSET UP EXTENSION BITS JBASED ON HIGH ORDER FRACTION JUMP IF + J= OTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16-X JCOUNT LOOP JPUT RESULT IN RØ, R1
JSET UP EXTENSION BITS JBASED ON HIGH ORDER FRACTION JUMP IF + J= OTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
JSET UP EXTENSION BITS JBASED ON HIGH ORDER FRACTION JUMP IF + J= OTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
<pre>/BASED ON HIGH ORDER FRACTION /JUMP IF + /= OTHERWISE /JUMP IF LESS THAN ONE WORD TO SHIFT /SHIFT RIGHT A WHOLE WORD /USE EXTENSION BITS /ACCOUNT FOR SHIFT /ACCOUNT FOR SHIFT /SHIFT LEFT 16=x /COUNT LOOP /PUT RESULT IN R0, R1</pre>
JUMP IF + J= OTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
<pre>J= OTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JACCOUNT FOR SHIFT JSHIFT LEFT 16=x JCOUNT LOOP JPUT RESULT IN R0, R1</pre>
<pre>J= OTHERWISE JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JACCOUNT FOR SHIFT JSHIFT LEFT 16=x JCOUNT LOOP JPUT RESULT IN R0, R1</pre>
JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JJUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
JUMP IF LESS THAN ONE WORD TO SHIFT JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JJUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
JSHIFT RIGHT A WHOLE WORD JUSE EXTENSION BITS JACCOUNT FOR SHIFT JJUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
JUSE EXTENSION BITS JACCOUNT FOR SHIFT JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
JACCOUNT FOR SHIFT JJUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
JUMP IF NOT MORE THAN 1/2 WORD JSHIFT LEFT 16=X JCOUNT LOOP JPUT RESULT IN R0, R1
ISHIFT LEFT 16-X Icount Loop Iput result in R0, R1
ISHIFT LEFT 16-X Icount Loop Iput result in R0, R1
ISHIFT LEFT 16-X Icount Loop Iput result in R0, R1
ISHIFT LEFT 16-X Icount Loop Iput result in R0, R1
ISHIFT LEFT 16-X Icount Loop Iput result in R0, R1
ISHIFT LEFT 16-X Icount Loop Iput result in R0, R1
ICOUNT LOOP Iput result in R0, R1
PUT RESULT IN RØ, R1
PUT RESULT IN RØ, R1
PUT RESULT IN RØ, R1
PUT RESULT IN RØ, R1
PUT RESULT IN RØ, R1
FSHIFT A MIN AND B MIN
FSHIFT A MIN AND B MIN
ISHIFT A MIN AND B MIN
IREDUCE EXPONENT DIFFERENCE
0 141+42
4
1 /81+82
\$P),@\$P
SP),@SP /Go Clean up subtract
IGO CLEAN UP SUBTRACT
IGO CLEAN UP SUBTRACT

SADR04 MACRO VR04-14 07-SEP-72 11143 PAGE 7+

133	2370	005203		INC	RJ	IINCREASE EXPONENT
4 3 4	2170	000303	NOD\$21	SWAB	R3	IMOVE EXPONENT LEFT
	0374	491909	14 62 68 69 88 8	BNE	OVR\$2	JUMP IF OVERFLOW
135						AUGHE AF OVEREGU
136	2376	150003		BISB	RU,R3	
137	2400	006016		ROR	OSP	JINSERT SIGN
138	2402			ROR	R3	
139		005001		ROR	R1	
140	2406	005501		ADC	R1	IROUND SUM
141	2410	005503		ADC	R3	
		102411		BVS	OVR\$2	JUMP IF OVERFLOW ON ROUND
						n Marina (1997). A Marina (1997) A Marina (1997) A Marina (1997) A Marina (1997)
143	2414	103410		BCS	OVRS2	
144	2416	010366	STRS2:	MOV	R3, A2(SP) ISTORE EXPONENT AND SIGN
		000010				
145	0420	010166		MOV	R1,82(SF	P) JINSERT LOW ORDER FRACTION
1-4-0	6466					
		000012			6 12 FA & -	IDAD STONE
146		005726	001\$21	TST	(SP) +	POP SIGNS
147	2430	012004		MOV	(SP)+,R/	
148		022626		CMP	(SP)+.(8	SP) + JPOP FIRST ARGUMENT
-				JMP		JDONE. RETURN
149	2434	000134	-	A D.L.	6 (I G & 4	S. P. Milania B. Lither Land Lite
150			A second second		·	
151	2436	004567	OVK\$21	JSR	R5, SERR	JERROR 3,2
		017344				
152	2442	-		BR	OUTS2	
-						
153	2444	003		BYTE	3	
154	2445	002		.BYTE	2	
155			7			
	2146	005700	SUB\$2;	TST	RØ	ICHECK HIGH ORDER RESULT FRACTION
			40 U - 40 in 9		BT952	IIF POSITIVE SIGN IS OK
157	2450			BGT		
158	2452	001413		BEQ	ZTSS2	ICHECK FOR ZERO RESULT
159	2454	005400		NEG	RØ	IGET ABSOLUTE VALUE
160		005401		NEG	R1	
161		005600		SBC	RØ	
162	2462	000316		SWAB	ØSP	JEXCHANGE SIGNS
163	2404		BT9\$21			
164				. IFOF	EAE	
				BIT	R0.4700	
165						THE TH HAT MANE THAN A TO BUTET
166				BNE	B9A\$2	JUMP IF NOT MORE THAN 2 TO SHIFT
167				MOV	R1,0#MQ	FRESULT FRACTION TO AC, MQ
168				MOV	RØ, ØHAC	
				CLR	@#NOK	INORMALIZE
169						
170				SUB	CHNOR, R	
171				MOV	#=6,0#A	SH JSHIFT TO CORRECT POSITION
172				ADD	#6,R3	ICOMPENSATE EXPONENT
				BLE	UNF 52	JUMP IF UNDERFLOW
173						A M Martin (# 6. Martin Merter Merter
174				MOV	@#AC,RØ	a maximum martin a martin martin martin filma
175				MQV	0#MQ/R1	IGET FRACTION BACK
176				BR	NUD\$2	
177				.ENDC		
	0.44		00A#0+		00 4400	
1/8	2404	030027	B9A\$21	BIT	R0,#400	
		000400				
179	2470	001014		BNE	UTS52	JUMP IF NORMAL BIT FOUND
		005303		DEC	R3	IDECREASE EXPONENT
						JDOUBLE FRACTION
		006301		ASL	R1	LACAR LUMPITAN
182	2476	006100		ROL	RØ	
10	2500			BR	B9AS2	ITRY AGAIN
		005701	778491	TST	R1	ICHECK LOW ORDER PART
	C 2 M S	002101	e * 3 6 i			संस्थात आप महाते कि संस्थात त्या स्थापन का प्राप्त के प्राप्त के प्राप्त के प्राप्त के प्राप्त के प्राप्त के प
185				, IFOF	EAE	

22

ين

÷

\$ADR04 MACRO VR04=14 07=SEP=72 11:43 PAGE 7+

-

Q

186				BNE	BT9\$2	
187				BR	ZERS2	
1.88				. ENDC		
189				. IFNDF	EAE	
190	2504	001415		BEQ	ZERS2	
191	2506	000301		SNAB	R1	ISAVE NORMALIZE SOME TIME
192	2510	150100		BISB	R1, R0	IMOVE BITS LEFT
193	2512	105001		CLRB	R1	
194	2514	162703		SUB	#8 ., K3	FTELL EXPONENT ABOUT IT
		000010				
195	2520	000761		BR	81952	
196				.ENDC		
197	2522	005703	UT5521	TST	R3	ICHECK FOR UNDERFLOW
198	2524	003322		BGT	NODS2	JUMP IF NONE
199	2526	004567	UNF\$21	JSR	R5,SERR	JERROR 5,2
		017254				
200	2532	000401		BR	UND\$2	
201	2534	005		BYTE	5	
202	2535	002		BYTE	2	
203		005001		CLR	R1	JUNDERFLOW, TREAT AS 0
204		005003	ZERS21	CLR	R3	ICLEAR EXPONENT
205	2542	000725	•	BR	STR\$2	
206				.ENDC		
207				. ENDC		

A I	.GØ3	MACRO VI	00A=14 0	7=SFP=72	11:43 PAGE 8	24
	.900		an a		के की में लाग । गांत कम भर	/
				.TITLE .IFDF	SALGØ3 CNDS3	
			1	ALOG	VØØJA	
			1			
			I COMAR	IGHI 197		PMENT CORPURATION, MAYNARD, MASS
				.GLOBL .IFNDF	ALOG, ALOG10, SER FPU	
2 1				.GLOBL .ENDC	SPOLSH, SADR, SSB	3R, SMLR, SOVR, SIR;
23			1		RAN ALOG AND AL Sequence:	OG10 FUNCTIONS
4			;	JSR	R5, ALOG (OR ALO)G10)
5 6			1	BR .word	A ARGUMENT ADDRES	SS State
7 8			7 A 1	RETURNS	LN (ARG) (OR LOG	10(ARG)) IN R0,R1.
9 0		000000		RØ=%0		
1		000001		R1=%1		and the second
2		000002		R2=%2		
3		000003		R3=%3		
4		000004		R4=%4		
5		0000005		R5=%5 SP=%6		
6 7		0000007		PC=%7		
8		000000		F0=%0		
3		000001		F1=%1		
2		000002		F2#%2		
ĺ		000003		F3=%3		
2		***		. IFNDF	FPU	
5	02544	011746	ALUG101	MOV	PPC,=(SP)	JGET 0004XX AS A FLAG
4	02546	000401		BR	LOGSS	
5	02550	005046	AL UG :	CLR	- (SP) IFLAG A	LOG
5	02552	016504	L06\$3:	MOV	2(R5),R4	IGET ARG ADDRESS
		000002				
7	02556	012748 071030		MOV	#071030,=(SP)	;PUSH =1/2*LN(2)
3	02562	012746		MOV	#137661,=(SP)	
2	02506	024646		CMP	=(SP),=(SP)	IGET WORK SPACE
		016446		MOV	2(R4), = (SP)	IGET ARG
1	a2574	011446		MOV	@R4,=(SP)	
		003534		BLE	ERRSS JJUMP I	F NOT POSITIVE
		006316		ASL	PSP	
		116666		MQVB	1(SP), 12. (SP)	JGET EXPONENT
		000014				
5	02610	112766 000200		MOVB	#200,1(SP)	TRANSFORM ARG TO (1/2,1)
		000001				
5	02616	006016		ROR	esp	
		012746		MOV	#002363,=(SP)	; PUSH 1/2+ROOT2
		002363				
3	02624	012746		MOV	#040065, - (SP)	

40						
	10530	040065		MOV	6(SP),=(SP)	PUSH X
** 3	06000	016646		FIQ V	0(37);=(37)	JENGII A
		000006			6 4 m m)	
50	02634	016646		MOV	6(SP), = (SP)	
		000006				
51	02640	012746		MOV	#002363, = (SP)	PUSH 1/2+ROOT2
-		002363				
52	12644	012746		MOV	#040065,=(SP)	
₩ 6	NEC-18				H040-007 (0)7	
62 B	~ 13 K 16 m	040065		100		JENTER POLISH MODE
53	02650			JSR	R4, SPOLSH	JENIER POLION HOUE
	-	016770				a na ta ta 20 a sa li an a bha an
54	02654	0020041		, WORD	SSBR, UPS3, SADR	SDVR JGET (X-ROOT2)/
	02656	0027661				
	12660	0020101				
		0132561				
55	1/11/2/11/2					1 (X+ROOT2)
		13 13 12 13 4 A 8		WORD	DUPS3, DUPS3	JGET THREE COPIES
56	02664			SHURU	00-30,00,00	IDEI INVER FALSEA
-	02656	0030141				
57	02670	0171621		, WORD	SMLR, REG\$3, STK	\$3,STK\$3,STK\$3 /SET UP POLYNOMI
	02672	0027421				
	02674	0027541				
	02676	0027541				
	02700	0027541				
60						, SADR, SMLR, SADR, SMLR, SADR
00		017162'		, WORD	aurelaurel	Jannulaurulauru
	415 1	0020101				
	02706	017162				
	02710	002010				
	02712	0171621				
	02714	002010				
		0171621			1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -	
-	N6/60	002010			TENDAN	D DOL VNOM + AL
59					· · · · · · · · · · · · · · · · · · ·	D POLYNOMIAL
60	02722	0030001		. WORD	SCL53, SIR, PL25	3, SMLR JGET LN(EXP)
	02724	0160621				
	02726	0030261				
	02730	017162'				
61	02732	0020101		WORD	17 An a series of the series 205 million	
¥ #					SADR/EXIDO	ICOMBINE WITH FRACTION
	0073A	DANDADI		S NOUD	SADR/EXIS3	ICOMBINE WITH FRACTION
60	02734	0030401		8 3 4 A D	SAURIEXISS	
62						IAND CHECK IF DONE
62 63	02736	0171621		.WORD	SMLR/EXISS	IAND CHECK IF DONE
	02736					IAND CHECK IF DONE
	02736	0171621				IAND CHECK IF DONE
63 64	02736	0171621 0030401				IAND CHECK IF DONE
63 64 65	02736 02740 02742	0171621 0030401 012500	1	. WORD	\$MLR,EXI\$3 (SP)+,R0	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN
63 64 65 66	02736 02740 02742 02742	017162 003040 012600 012601	1	. WORD Mov Mov	\$MLR, EXI\$3 (SP)+, RØ (SP)+, R1	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN POP Y
63 64 65	02736 02740 02742 02742	017162 003040 012600 012601 012702	1 Reus31	. WORD	\$MLR,EXI\$3 (SP)+,R0	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN
63 64 65 66 67	02736 02740 02742 02744 02744	017162 003040 012600 012601 012702 003124	1 Reus31	. WORD Mov Mov Mov	\$MLR,EXI\$3 (5P)+,R0 (SP)+,R1 #CON\$3+4,R2	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN POP Y
63 64 65 66 67 68	02736 02740 02742 02744 02744 02745 02752	017162 003040 012600 012601 012702 003124 000402	1 Reus31	.WORD Mov Mov Mov Br	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN JPOP Y JPOINT TO GOEFFICIENTS
63 64 65 66 67	02736 02740 02742 02744 02744 02746 02752 02754	017162 003040 012600 012601 012702 003124	1 Reus31	. WORD Mov Mov Mov Br Mov	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1, = (SP)	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN POP Y
63 64 65 66 67 68	02736 02740 02742 02744 02744 02745 02752	017162 003040 012600 012601 012702 003124 000402	1 Reus31	.WORD Mov Mov Mov Br	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1, = (SP) R0, = (SP)	JAND CHECK IF DONE JMULTIPLY BY LOGIØ(E) AND RETURN JPOP Y JPOINT TO GOEFFICIENTS JPUSH Y
63 64 65 66 67 68 69	02736 02740 02742 02744 02746 02752 02754 02754	017162 003040 012601 012601 012702 003124 000402 010146 010046	1 RE4531 STK\$31	. WORD Mov Mov Mov Br Mov	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1, = (SP)	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN JPOP Y JPOINT TO GOEFFICIENTS
63 64 65 66 67 68 69 70 70	02736 02740 02744 02744 02746 02752 02754 02756 02750	017162 003040 012601 012601 012702 003124 000402 010146 010046 014246	1 RE4531 STK\$31	. WDRD MOV MOV MOV BR MOV MOV MOV	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1, = (SP) R0, = (SP) = (R2), = (SP)	JAND CHECK IF DONE JMULTIPLY BY LOGIØ(E) AND RETURN JPOP Y JPOINT TO GOEFFICIENTS JPUSH Y
63 64 65 66 67 68 69 70 71 72	02736 02740 02744 02744 02746 02752 02754 02756 02750 02750 02750	017162 003040 012601 012601 012702 003124 000402 010146 010046 014246 014246	1 RE4531 STK\$31	. WDRD MOV MOV MOV BR MOV MOV MOV MOV	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1,=(SP) R0,=(SP) =(R2),=(SP) =(R2),=(SP)	JAND CHECK IF DONE JMULTIPLY BY LOGIØ(E) AND RETURN JPOP Y JPOINT TO GOEFFICIENTS JPUSH Y
63 64 65 66 67 68 69 70 71 72 73	02736 02740 02744 02744 02746 02752 02754 02756 02750 02750 02750	017162 003040 012601 012601 012702 003124 000402 010146 010046 014246	1 RE4531 STK\$31	. WDRD MOV MOV MOV BR MOV MOV MOV	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1, = (SP) R0, = (SP) = (R2), = (SP)	JAND CHECK IF DONE JMULTIPLY BY LOGIØ(E) AND RETURN JPOP Y JPOINT TO GOEFFICIENTS JPUSH Y
63 64 65 66 67 68 70 71 72 73 74	02736 02740 02744 02744 02752 02754 02754 02756 02760 02762 02764	017162 003040 012601 012601 012702 003124 000402 010146 010046 014246 014246 014246 010134	1 RE4531 STK531 STC531	. WORD MOV MOV MOV BR MOV MOV MOV MOV JMP	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1,=(SP) R0,=(SP) =(R2),=(SP) =(R2),=(SP) @(R4)+	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN POP Y POINT TO GOEFFICIENTS PUSH Y PUSH COEFFICIENT
63 64 65 66 67 68 70 71 72 73 74	02736 02740 02744 02744 02746 02752 02754 02756 02750 02750 02750	017162 003040 012601 012601 012702 003124 000402 010146 010246 014246 014246 014246 014246 014246	1 RE4531 STK531 STC531	. WDRD MOV MOV MOV BR MOV MOV MOV MOV	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1,=(SP) R0,=(SP) =(R2),=(SP) =(R2),=(SP)	JAND CHECK IF DONE JMULTIPLY BY LOGIØ(E) AND RETURN JPOP Y JPOINT TO GOEFFICIENTS JPUSH Y
63 64 65 66 67 68 70 71 72 73 74	02736 02740 02744 02744 02752 02754 02754 02756 02760 02762 02764	017162 003040 012601 012601 012702 003124 000402 010146 010046 014246 014246 014246 010134	1 RE4531 STK531 STC531	. WORD MOV MOV MOV BR MOV MOV MOV JMP MOV	<pre>\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1,=(SP) R0,=(SP) =(R2),=(SP) =(R2),=(SP) @(R4)+ (SP)+,10.(SP)</pre>	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN POP Y POINT TO GOEFFICIENTS PUSH Y PUSH COEFFICIENT
63 64 65 66 67 68 70 71 72 73 74 75	02736 02740 02744 02744 02752 02754 02755 02756 02760 02762 02764 02766	017162 003040 012601 012601 012702 003124 000402 010146 010246 014246 014246 014246 014246 014246	1 RE4531 STK531 STC531	. WORD MOV MOV MOV BR MOV MOV MOV MOV JMP	\$MLR, EXI\$3 (SP)+, R0 (SP)+, R1 #CON\$3+4, R2 STC\$3 R1,=(SP) R0,=(SP) =(R2),=(SP) =(R2),=(SP) @(R4)+	JAND CHECK IF DONE JMULTIPLY BY LOG10(E) AND RETURN POP Y POINT TO GOEFFICIENTS PUSH Y PUSH COEFFICIENT

SALG03 MACRO VR04=14 07=SEP=72 11:43 PAGE 8+

ж,

¢

SALG03 MACRO VR04=14 07=SEP=72 11:43 PAGE 8+

77	02776	000134	,	JMP	@(R4)+	
79	03000	005046	SCLS3:	CLR	-(SP)	
80		156515		BISB	6(SP), ØSP	JGET EXPUNENT
00	04046	000006				
81	23006	162716		SUB	#200,0SP	FREMOVE EXCESS 128
· •	00050	000200				
82	03012			JMP	@(R4)+	
83			3			
84	03014	016546	DUPS3:	MOV	2(SP),=(SP)	
		000002				
85	03020			MOV	2(SP),=(SP)	JUUPLICATE STACK ITEM
		000002				
86	03024	000134		JMP	@(R4)+	
87			1		•	
88	03025	012746	PL2531	MOV	#071030, - (SP)	PUSH LN(2)
		071030				
89	03032	012746		MOV	#040061, - (SP)	
		040061				
90	03036	000134		JMP	@(R4)+	
91			1. C			
92	03040		EXIS3:	DECB	5(SP) JCHECK	FOR ALOGIO
		000005		· · · · · · · · · · · · · · · · · · ·		
93	03044			BLT	LGTS3 IND, DO	
94	03046	() (m		MOV	#Ø55731, = (SP)	PUSH LOGIØ(E)
•	a the an Earlie	055731				
95	03052			MOV	#037736,=(SP)	
		037736		1.40		
96	03056			JMP	@(R4)+	POP RESULT
97	03000	••• ••• •••• ••• ••• •••	LGTS3:	MOV	(SP)+,R0	JPUP RESULT
98	03052			MOV	(SP)+,R1 (SP)+ JFLUSH	21 A C
99	03064			TST	• • •	FLAY
10		100 C 100 C 100	CADAL	ADD	R5 #14./SP	
10	1 3070		ERR\$31	AUU	#14810F	
4 /2 /	2 3074	000015		JSR	R5, SERR JERROR	4.10
1,03	2 90/4			4 90	NOISERE LENNON	
4.0	1 2100	016706		RTS	R 5	
10		000205 004		BYTE	4	
10		012		BYTE	10.	
10		ં સ્ટ્રાય		ENDC		
10				IFDF	FPU	
10			ALOG10:	MOV	PPC, R41	GET 0004XX AS ALOGIO FLAG
10				BR	LOGS3;	
11			ALUG:	CLR	R41	GET Ø AS ALOG FLAG
11			LOGSSI	SETF	1	SINGLE PRECISION FP
11				SETI	· · · · · · · · · · · · · · · · · · ·	SHORT INTEGERS
11				MOV	#FC053,R0	;PUINTER TO CONSTANTS FOR ROUTIN
11				LDF	#2(R5),F21	GET ARGUMENT
11				CFCC		
11	6			BLE	EKR\$31	JUMP IF NOT POSITIVE
11				STEXP	F2,R1)	GET EXPONENT OF ARG
11	8			LOCIF	R1,F31	CONVERT T O FP FORM
11				MULF	(RU)+,F31	SCALE FACTUREEXPONENT+LN(2)
12				LDEXP	#0,F2;	TRANSFORM ARG TO (1/2,1)
12				LOF	F2,F1;	
12	2			SUBF	(RU) + F21	X=1/2*SQRT(2)

ø

123				ADDF	(RØ)+,F17	X+1/2+SQRT(2)	
124 125				DIVF Lof	F1,F2; F2,F1;	W#(X=R00T2)/(X+R0	1012)
126				MULF	F1,F1;	Y= W**2	
127			1				
128				MOV	43,R11	COUNT OF CONSTS F	
129				LDF	(RØ)+,FØJ	INITIALIZE ACCUMU	LATOR FOR POLY
130			XPUS31	MULF	F1,F0;	P (1) ALT	
131				DEC Addf	R11 (RØ)*,FØJ	COUNT F0:= Y*F0 + C(I)	
132 133				BGT	XPDS3;	LOOP	
134			4		AT WOT I		
135				MULF	F2, FØ;		
136				ADDF	(RØ)+,FØ)	F01= W+F0 = 1/2+L	N(2)
137				ADDF	F3,FU;	ADD SCALE FACTOR	FOR EXPONENT
138				TST	R41	TEST ALOG10 FLAG	
139				BEQ	LGT\$3;		
140				MULF	(RØ)+,FØI	ALUG10:= ALOG+LOG	10(E)
141			1			NOVE DECILIT THE OF	
142			LGT531	STF	F0, - (SP) /	MOVE RESULT TO ST	ALK
143				MOV Mov	(SP)*,RØJ (SP)*,R17	AND THENCE TO RO,	01
145				RTS	R51	AND THENDE TO NOT	n +
146			ERK\$31	JSR	R5, SERR)	ERROR 4,10	
147				RTS	R51	EXIT-NO STACK CLE	ANUP NECESSARY
148				BYTE	4		
149				.BYTE	10.		
150) - 1 - 1		DEPENDENT CONSTAN		and the second
151			1	RØ POI	NTS AT CURRENT CO	NSTANT IN FPU VERS	ION
152 153			FCUS31	.WORD	040061,071030;	LN(2)	
154			1	a strend to	0400-110-10441	See 1.2. If the st	
155			•	WORD	040065,0023631	1/2*SQRT (2)	
156				. ENDC			
157			7	CONSTAL	NTS FOR POLYNOMIA	L EXPANSION	
158			1				
159		037632		, WORD	037532,014525	1.300974506	
	3106	014525					
160	12 4 4 m	a 2 11 7 4 A	1	MOOD	037714,120036	1,399659100	
101		037714		. WORD	03//14/14/030	1.033023100	•
160	0112	120036	, .				
162	3114	040052	•	WORD	040052,125332	1.666669471	
100		125332			0-00-078-000B		
164	-		1				
	3120	040400	CON\$31	. WORD	040400,000000	11.99999999	
		000000					
166	-			. IFDF	FPU		
167			1		RDER-DEPENDENT CO		
168				.WORD	137661,0710301	=1/2*LN(2)	
169			1	1.1 P		10610751	
170				.WORD	037736,0557311	LOG10(E)	
171				.ENDC			
1/2							

SANTØ3 MACRO VRØ4-14 07-SEP-72 11:43 PAGE 9

1				.TITLE	SANTUS
2				.IFDF	CND\$4
3				.GLOBL	AINT/SINTR
4			,	AINT	VØØ3A
5			1	COPYRIG	HT 1971, DIGITAL EQUIPMENT CORP., MAYARD, MASS.
6			1	AINT	FORTHAN AINT FUNCTION, CALLING SEQUENCE
7			1	JSR	R5, AINT
8			2	BR	
9			1	WORD	ADDRESS OF ARGUMENT
10			1 4 1	and the second sec	
			1	RETHRNS	SIGN OF ARG * GREATEST REAL INTEGER < #
11			*) IN RØ AND R1.
12			/ 	MONTANO	
13				******	SAME FUNCTION AS AINT, BUT CALLED IN THE
14	4 .		1	SINTR	MODE WITH THE ARGUMENT AND RETURN ON THE STACK.
15			.)		ADAR ATTO THE MORANEME WAS ORIGHT ON THE ALLENGE.
16		000000		RØ#%0	
17		000001		R1=%1	
18		000002		R2=%2	
19		000003		R3=%3	
20		000004		R4=%4	
21		000005		R5=%5	
22		0000006		SP = % 6	
23		000007		PC=%7	
24		177304		MQ=1773	Ø 4
25		177314		LSH=177	314
26		000000		F0=%0	
27		000001		F1=%1	
28				. IFOF	S FPU [™] Constant of the second state of the
29			AINT:	, WORD	170001 71SETF
30				.WORD	172475,2 11LDF #2(R5),FØ 1GET ARG
31				, WORD	171467,24 JIMODE UNE, FØ JGET INTEGER PAR
32				WORD	174146 11STF F11=(SP)
33				MOV	(SP)+,RØ JPOP TO USER REGS
34				MOV	(SP)*,R1
35				RTS	R5 IRETURN
36			;		
37			SINTRE	WORD	170001 11SETF
38				WORD	172426 11LDF (SP)+, FØ 1GET ARG
39				WORD	171467,4 JIMODE UNE, FØ IGET INTEGER PAR
40				WORD	174146 11STF F1.=(SP)
41				JMP	Ø(R4)+ IRETURN
			ONES41	.WORD	040200,0 /FLOATING 1.
42			WINH 0 T R	ENDC	
43					e au
44				, IFNDF	FPU 2(R5),R4 JGET ARGUMENT ADDRESS
40	00124	016504	AT NI .	MOV	2(R5),R4 JGET ARGUMENT ADDRESS
A 22		000002		Mi Ch V	@R4,RØ JGET HIGH ORDER ARGUMENT
46		011400		MOV	
47	03132	016401		MOV	2(R4),R1 JLOW ORDER
•		000002			
48		010702		MOV	PC,R2 JMAKE R2 NON Ø
49	03140			BR	AI154
50		005002	SINTRI	CLR	R2 JMAKE R2 0
51		012600		MOV	(SP)+,RØ /GET HIGH URDER ARGUMENT
52		012601		MOV	(SP)+,R1 ILOW ORDER
53		010003	A1154;	MOV	R0, R3
54		006103		ROL	R3 JDUMP SIGN
55	03154	105003		CLRB	R3

28

SANT03 MACRO VR04-14 07-SEP-72 11:43 PAGE 9+

(

in the second se

Q

56	03156	000303		SWAB	R3	IGET EXPONENT
		162703		SUB	#230,R3	TREMOVE EXCESS 200 AND CHECK RANGE
	~ ~ • • •	000230				
58	03164			BGE	DNES4	JJUMP IF IT IS ALREADY AN INTEGER
59	03166	022703		CMP	4-30,R3	
	041-4	177750				
60	03172	002403		BLT	SHFS4	JJUMP IF THERE IS WORK TO DO
	03174	005000		CLR	RØ	JARG IS < 1, SO RETURN Ø
62	03176	005001		CLR	- R1	
63	03200	000412		BR	DNES4	
64	03202	010346	SHFS41	MOV	R3.=(SP)	PUSH -SHIFT COUNT
	04545	010040	Ohiowa	TENDE	EAESMULD	
65	- 1 - A - A - A		DaverAy	ROR	RØ	ISHIFT FRACTION
	03204	006000	ROR\$41	ROR	R1	<u>a Autri Prima (* 1800)</u>
67	03206	006001		INC	R3	ICOUNT LOOP
68	03210	005203			· #	
69	03212	002774		BLT	RUR\$4	JGO AGAIN
70	03214	012603		MOV	(SP)+,R3	
71	03216	006301	ASL541	ASL	R1	ISHIFT FRACTION BACK WITH 0'S
	03220	006100		ROL	RØ	Remaining (American)
	03222	005203		INC	R3	FCOUNT LOOP AGAIN
74	03224	002774		BLT	ASLS4	
75				. ENDC		
76			1. F. 1.	EAE CO		
77				.IFDF	EAE	
78				Mav	#MQ, 83	POINT TO MO
79				MOV	R1, eR3	JINSERT ARG
80				MOV	RØ,=(R3)	
81				MOV	@SP, @#LS	
82				NEG	ØSP	JSET FOR LEFT
83				MOV	(SP)+,04	
84			5* * <u>5</u> * *	MOV	(R3)+,R0	TRESULT TO REGS
85				MOV	@R3, K1	
86				ENDC		
87			1	MULDIV	CODE	
88			1. A.	. IFDF	MULDIV	
89				WORD	073016	I;ASHC @SP,RØ ISHIFT OUT FRACTION
90		177 - E		NEG	OSP	ISET FOR LEFT SHIFT
91				WORD	073026	JJASHC (SP)+, RØ JSHIFT INTEGER P
92			v ter	. ENDC		
93	03226	005702	DNES41	TST	R2	ICHECK ENTRY FLAG
94	03230	001401		BEQ	DN154	JJUMP IF SINTR
95	03232	000205	t de la	RTS	R5	FRETURN IF SAINT
96	03234	010146	DN1541	MOV	R1,=(SP)	PUSH RESULT
97	03236	010046		MOV	RØ, - (SP)	
98	03240	000134		JMP	@(R4)+	POLISH RETURN
99	क्राल क ा €2			ENDC		
100	A			ENDC		and the second
	•			के संस्थात की संस		

SCMD02 MACRO VR04=14 07=SEP=72 11:43 PAGE 10

			TITLE	
				CND\$5
				SCMD
		1	SCMD	THE DOUBLE COMPARE ROUTINE.
		1	8 • M 0	V (2/2 m A
			sunu	V002A
		· · · · · · · ·	CORVETO	HT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
				IN THE POLISH MODE WITH THE TWO
		1		NDS ON THE STACKS
		1 1 1	10 TCT	S AT B(SP), SECOND IS ØSP
				HE TWU COMPARANDS AND RETURN
	•	3		LOWING CONDITION CODES:
		ý		SECOND N=1, Z=0
		· j		SECOND NEØ, ZE1
		1		SECOND NEO, ZEO
	000000		R3=%0	
	000001		R1=%1	
	000002		R2=%2	
	000004		R4=%4	
	000000		SP=%6	
	000000		·	
		Salakana S	N	FPU
		SCMDI		170011 JISETD
				172426 JILDD (SP)+,FO JGET SECOND ARG
1 - C.				173426 JJCMPD (SP)+,FØ JCOMPARE
				170000 FICFCC FGET CONDITION CODES
				@ (R4) +
				FPU
03040	611700	SeMint		PPC, RØ IGET ØØXXXXX XXXX01 IN RØ
		\$L1.0.		8. (SP), R1 JGET HIGH URDER FIRST ARG
92 4 6 -				n a thur the second s
a325a			BGE	FP555 JJUMP IF FIRST ARG +
				RØ /FLAG FIRST ARG -
				(SP)+,R2 JGET HIGH SECOND ARG
	ter de min de de			SMESS JJUMP IF BOTH SIGNS -
			BR	NEGS5 JJUMP IF FIRST - AND SECOND +
		FPS\$51	MOV	(\$P)+,R2
03264	002421		BLT	PLSSD JJUMP IF FIRST + AND SECOND -
03266	020102		CMP	R1, R2 ICOMPARE MAGNITUDES
			BNE	OUTSD JJUMP IF DIFFERENT
03272			CMP	8.(SP), esp
	000010		ana. da	
				OUT\$5
03300			CMP	10.(SP),2(SP)
1. 1. 18 FA			QNE	<u>Λιτ≮∞δ</u>
03306			BNE CMP	0UT\$5 12 (SP) 4(SP)
	UCOUDD U		U M F	12.(\$P),4(\$P)
03310				
03310	000014			
	000014 000004		BNF	nutsa
03316	000014 000004 001001		BNE	DUTSO Pa JFLAG B
03316 03320	000014 000004 001001 005000	0u1 s5 :	CLR	RØ JFLAG =
03316 03320	000014 000004 001001 005000 006000	QUT\$5:		
	03244 03250 03252 03254 03256 03260 03262 03264 03266 03270 03272 03276	000001 000002 000006 000006 000007 000000 000000 000000 000000 000000	000001 000002 000006 000007 000000 scmdi 03242 011700 scmdi 03244 016501 000010 03250 002004 03252 006300 03254 012602 03256 002403 03256 002403 03256 002423 03256 002422 03256 002421 03260 000422 smts5: 03270 001014 03272 026516 000010 03276 001011 03300 026666 000012	. IFDF .GLOBL

ъ

52	03326	005400	NEGSSI	NEG	RØ	IREVERSE C	BIT	
53	03330	062706	PLS551	ADD	#14./SP	IPOP ARGS		
· · ·		000016						
54	03334	005700		TST	RØ	ISET Z AND		CORRECTLY
55	03336	000134	•	JMP	0 (R4) +	IRETURN TO	CALLER	
56				ENDC			-	
57				. ENDC				

1				TITLE	SCMR02			
2		•		IFOF	CNDSD	ta di kacala da kacal		
3				GLOBL	SCMR			
Ā				5CMR		COMPARE P	DUTINE.	
-				₩ ₩ ₩ 1 1 K	t i tan i tan mit	e felorisi talizme s	1999 - C. B. 1999 - B	
5			1	* * *	VadoA		s production and the second	4
6				SCMR	V002A	\$15.		
7			7					
8			ст р ана страната					P., MAYARD, MASS,
9			1	CALLED	IN THE PO	JUISH MODE	WITH THE TH	0
10			1	COMPARA	NDS ON TH	E STACKI		
11			1	FIRST I	S AT 4(SF), SECOND	IS OSP	
12			1			MPARANDS A		
13			1			NDITION CO		
14			,		SECOND			
15			· ·			N=0, Z=1		
			·			N=0, Z=0		
16		000000	1	* . • •	SECUND	1 H 2 1 2 H 2		
17		000000		R0=20				
18		000001		R1=%1				
19		000005		R2=%2				
20		000004		R4=%4				
21		0000006		SP=%6				
22		000007		PC=%7				
23		000000		F0=X0				
24				. IFOF	FPU			
25			SCMRI	WORD		11SETF		
26				WORD	172426	IILDE (S	P)+.FØ	IGET SECOND ARG
27				WORD	173426		P)+,FØ	ICOMPARE
28				WORD	170000		ET CONDITIO	• • • • • •
				JMP	@(R4)+			1 00000
29					W [N 47 4			
30				.ENDC	eff. (5) / .			
31				.IFNDF	FPU	Bana and war in the coloring		
		011700	SCMRI	MOV		IGET ØØXXX		
33	03342	016601		MOV	4(SP),R1	1 G	ET HIGH URD	ER FIRST ARG
		000004						
34	03346	002004		BGE	FPS50		IRST ARG +	
35	03350	006300		ASL	RØ	IFLAG FIRS	IT ARG =	
36	03352	012002		MOV	(SP)+,R2	1 G	ET HIGH SEC	OND ARG
37	03354	002403		BLT	SMESO	JJUMP IF B	OTH SIGNS -	
38	03356	000412		BR	NEGSO	JUMP IF F	IRST - AND	SECOND +
		012602	FPSS6:	MOV	(SP)+,R2			
		002411		BLT	PLSSO		IRST + AND	SECOND -
		020102	SHES6:	CMP	R1, R2	ICOMPARE M		en en en en se
		001004	A.I. Fair 18 A. A.	BNE	OUTSO	JUMP IF D		
								- ON FB
43	00010	026016		CMP	4(SP),05		OMPARE LOW	JAUER
		000004		Bar	10 1 1 10 2 4		T per per per per per site	
		001001		BNE	OUTSO	JUMP IF D	IFFERENT	
		005000		CLR	RU	IFLAG =		
		006000	OUT\$61	ROR	RØ		T AND TEST	SECOND ARG -
47	03402	103401		BCS	PLS50	JJUMP IF S	ECOND ARG +	
48	03404	005400	NEGSOI	NEG	RØ	IREVERSE C	BIT	
49	03406	062708	PLSS6:	ADD	#6,SP	IPOP ARGS		
-		000006					· · · · · · · · · · · · · · · · · · ·	
5 A	03412	005700		TST	RØ	ISET Z AND	N BITS COR	RECTLY
		000134		JMP	@(R4)+	IRETURN TO		
52	*****	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		.ENDC	र कुरालाहवा .	∘ राक्त करणार्थाः । भूत	and the last last last last	
53								
93				.ENOC				
				.TITLE .IFOF	SDBLØ2 CNDS7			
----	--------	--------	---------	-----------------	---	--	--	--------------
			1	DBLE	V002A			
				VOLE	VOUZM			
			I COPYR	IGHT 197	I, DIGITAL	EQUIPMENT	CORPORATION,	MAYNARD, MAS
			1					
				GLOBL	DBLE			
0			1		TRAN DBLE	FUNCTION		
L			1		SEQUENCE			
2			2	JSR	R5, DBLE			
5				BR	Α			
\$			2	WORD	ARGUMENT	ADDRESS		
5					and the second se			
5			2	RETURNS	THE DOUBL	E PRESICION	EQUIVALENT	
7			1			ENT IN RØ -		
8			1					
ģ		000000	•	RORXO				
0		000001		R1=%1				
1		000002		R2=%2				
2		000003		R3=%3				
3		000005		R5=%5				
4	03416	016502	OBLES	MOV	2(R5),R2	IGET	ARG ADDRESS	
		000002						
5	03422			MOV	(R2)+,RØ	IGET	HIGH URDER	
5	03424			MOV		GET LOW ORD		
7		005002		CLR		CLEAR LOWES		
		005003		CLR	R3	1998 Berlin 1997 B	a and a second	
		000205		RTS		RETURN TO C	ALLER	
0	******			ENDC			· · · · · · · · · · · · · · · · · · ·	

SUBL02 MACRO VR04=14 07=SEP=72 11:43 PAGE 12

6

ű,

00101	MACRU VI	R04=14 07=SEP=72 11:43 PAGE 13	34
		TITLE SOCIØ1	
		IFOF CNDS8	
		\$ \$DC1 V001A	
		; COPYRIGHT 1971, DIGITAL EQUIPMENT CORPORATION	, MAYNARU, MASS
		.GLOBL SDCI/SRCI	
		.GLOBL SDCI/SRCI ;	
		\$ SRCI ASCII TO REAL CONVERSION.	
		CALLING SEQUENCE!	
		PUSH ADDRESS OF START OF FIELD	
1		PUSH LENGTH OF FIELD	
•		PUSH FORMAT SCALE D FROM W.D	
		PUSH P FORMAT SCALE	
1		; JSR PC, SDCI (DR SRCI)	
	000000		
	0000001	R1=%1	
	000000		
	000003		
	000004		and the second sec
	000005	R5=X5	
	000006		
	000007		
	000000		
	000002	POINTL#2 DIGITS#4	
	000000	그는 전 것 같아요. 그는 그는 것 같아요. 그는	
		ESIGN=8,	
l		SIGN=10,	
	000014		
	000030	P=30,	
	000040		
		ERF=26,	
		LENGTH=34,	
		TEMPALENGTH Desku tad	
		RESULT=P Start=36,	
		ENDESTART	
	005046		
	005216		AG
	000401	BR CNV\$8	
	005045		ET FOR DOUBLE
		CNV\$8: MOV R0,=(SP)	
	010146		
	010246	MOV R2,=(SP) Mov R3,=(SP)	
	010346		
	010545	MOV R5,	
	005045		
	005046		
03464	005046	CLR - (SP) /CLEAR ESIGN	
03466	012746	MOV #65, (=(SP) /INITIALIZE BEX	2
	000101		
03472	012746		DIGITS
	000022		

 $\left(\right)$

SDCI01 MACRO VR04-14 07-SEP-72 11143 PAGE 13+

ú

55	03476	005048	· · · ·	CLR	-(SP) JCLEAR POINTL
57		005046		CLR	- (SP) ICLEAR NUMEND
58	1. 1. 1.	016005		MÖV	STARI(SP), R5 JGET FIELD START ADDRESS
		000044			
59	03506	066560		ADD	LENGTH(SP), END(SP) JPUINT TO END +1
		000042			
		000044			
60	03514			CLR	RØ /CLEAR NUMERIC WURK SPACE
61		005001		CLR	R1
62		005002		CLR	\mathbb{R}^2
		005002		CLR	RS and the second s
		112504	SCNSBI	MOVB	(R5)+,R4 JGET NEXT INPUT CHARACTER
64			364901	BIC	#177600,R4
65	00060	042704		010	++//×8001
		177600		CHRO	OA AL TREET FOD BLANK
66	00002	120427		CMPB	R4,#1 ITEST FOR BLANK
	يت الان من عان	000040		676 a. (. 677	ARAAN TTE NAT DIANK LARK KAR . ARA -
67		001005		BNE	SGSSB JIF NOT BLANK LOOK FOR + OR -
68	03540			CMP	R5, START(SP) JCHECK END OF FIELD
		000044			an an sina 25. It be and the Data binner Phane Phane T ⁰ ματαβ ⁰ binner Marana
69		002767		BLT	SCNS8 FIF NOT DONE GO GET NEXT
70	03546	000167		JMP	ZERS8 JENTIRE FIELD IS BLANK
		000325			
71	03552	120427	SG \$ 5 8 :	CMPB	R4,#!+ ICHECK FOR + SIGN
		000053			
72	03556	001455		BEQ	FLDS8 /IF FOUND IGNORE IT
73	03560	120427		CMPB	R4,#1# ICHECK FOR # SIGN
		000055			
74	03564	001013		BNE	NCK58 / JIF NOT FOUND CHECK NUMERICS
75	03566	005266		INC	SIGN(SP) /SET = SIGN FLAG
		000012			
76	03572	000447		BR	FLDS8
77		112504	NXTS8:	MOVB	(R5)+,R4 JGET NEXT INPUT CHARACTER
78		042704		BIC	#177600, K4
	<u></u>	177600			
79	03602	120427		CMPB	R4,#1 JCHECK FOR BLANKS
· ···		000040			
80	03606	001002		BNE	NCKSB
81				MOV	HID, NA ITREAT BLANK AS 0
01	00010	000000		1101	and the first state of the second contract of the second
82	03614	120427	NPKSHI	CMPB	R4,#'Ø ICHECK FOR LEGAL CHARACTER
02	00014	000060		Autor D	Uala () Laiteán Lao Peáse á Giovara.
0 1	03690			BLT	PCKSB JCHECK FOR DECIMAL POINT
		002514			NNZSO JUMP IF NOT 0
84		001010		BNE	
85	-	005700		TST	
		001006		BNE	NN Z \$ 8
	03630			TST	
88		001004		BNE	NNZSU
89	107	005702		TST	R2
		001002		BNE	NNZ 58
		005703		TST	RJ
92		001423		BEG	FLD\$8
93	03644	120427	NNZ\$81	CMPB	R4,#19
		000071			ار این
	03650		· · ·	BGT	EXCSB ICHECK FOR EXPONENT
95	03652	005355		DEC	DIGITS(SP) JCOUNT AS A SIGNIFICANT DIGIT
		000004			
96	03656	002003		BGE	A2188 JJUMP IF WE CAN USE THIS DIGIT

97	g366g	005266		INC	EEXP(SP) JCOMPENSATE FOR SKIPPED DIGIT
		000014			
98	03664	000412		BR	FLD\$8
99	03606	162704	A21381	SUB	#60,R4 FCONVERT ASCII TO INTEGER
		000060			
100	3672	004767		JSR	PC,ML558 JMULTIPLY BY 5
		001044			
	7070	004767		JSR	PC,LFTS8 JDUUBLE RESULT FOR 10
101	00/0			990	LATPLIED INCOME WEAREL LOW IN
	-	001106		100	DA DI TABB TH DIDDENT DECTS
		060423		ADD	R4,R3 JADD IN CURRENT DIGIT
		005502		ADÇ	e R2 de la companya d
104	3706	005501		ADC	R1 Control (1997) And the second state of t
105	3710	005300		ADC	RØ JEND OF CONVERT FOR THIS DIGIT
106	3712	020566	FLD\$81	CMP	R5, END(SP) JCHECK FOR END OF FIELD
		000044	**		
107	3716	002725		BLT	NXTS8
		010310		MOV	R5,05P POINTER TO LAST NUMERIC TO NUMEND
		005700	SCLS81	TST	
		001000		BNE	SC158 JJUMP IF NUMBER NOT Ø
		005701		TST	en 👫 planetural, site of the planet site of the set of
112	3730	001004		BNE	50198
		005702		TST	R2
		001002		BNE	SC138
		005703		TST	RJ
		001457		BEQ	ZERSO FINPUT NUMBER IS 0
		021005	SC158:	CMP.	OSP, R5 ICHECK NUMEND
		001003		BNE	NOPSE JJUMP IF THERE WAS AN EXPONENT FIELD
119	3746	166666		SUB	P(SP), EEXP(SP) JUSE THE FURMAT P SCALE
		000036			
		000014			
120	3754		NOP\$8:	TST	PUINTL(SP)
40 mm 147	••••••••	000002		·	
121	3760	001002		BNE	PNTSE JJUMP IF THERE WAS A DECIMAL POINT
				MOV	D(SP), eSP JUSE THE D SCALE
124	0/02	016616		FLO V	DISFILES INC THE DISCARE
	1 1 1 ¹⁷	000040	0. T. O.	6 1 1 6 1	ANTI LEBA ACO
123	3706	166516	PNT58:	SUB	PUINTL(SP), @SP
		000005			
124	3772	101066		SUB	<pre>@SP,EEXP(SP) /FORM COMPLETE DECIMAL EXPONENT</pre>
		000014			
125	3776	003003		BGT	MULS8 IMULTIPLY BY 10**EXP
		002543		BLT	DIVSE JUMP IF DECIMAL EXPONENT IS NEG
		000167		JMP	FLTSØ JUMP IF EXP IS Ø
161	-002			A GLE C	
		000446			
128	4006		MULS8:	CMP	R0,#31462
		031462			
129	4012	101011		BHI	MDV\$8 JJUMP IF FRACT TOO BIG TO MULT BY 5
		004767		JSR	PC,ML5S8 JFRACT=5*FRACT
		000722			
131	4020	005286		INC	BEXP(SP) ITIMES 2
401	~~ () ** ()			and the first	wmar wry risting fisting fa
1 7 4	1001	000006	3.0-9.	nee	
195	4024	005366	010201	DEC	EEXP(SP) JOVER 10
		000014			
133	4030	003366		BGT	MULSO JJUMP IF MORE DECIMAL EXPONENT
134	4032	000167		JMP	FLTSB IDECIMAL EXPONENT GONE
		000416			
135	4036	004767	MDVs81	JSR	PC,M04\$8 JMULTIPLY BY 5/4
	~~ <i>U</i> ~ V	000632	······································	₩₩7 5 . 	ে অনুসংগ্ৰহণ বিভাগ বিধায় হয়। বিধায় হয়। প্ৰতিষ্ঠা বিধায় হয়। বিধায় হয়। বিধায় বিধায় বিধায় বিধায় বিধায় বিধায়
			· · · ·		

36

(

¢

V

ũ.

Ð

136	4042	062766		ADD	#3, BEXP(SP) ;TIMES 8
		000003			
		000000			
137	4050	000765		BR	D1038 JGO DIVIDE BY 10
138	4052	120427	PCKS81	CMPB	R4,#1,
		000056			
139	4056	001006		BNE	ERRSB JJUMP IF NOT A DECIMAL POINT
140		005766	PTFS81	TST	PUINTL(SP)
* 7 ¥		000002			
141	4054			BNE	ERRSB JJUMP IF A . ALREADY ENCOUNTERED
	4006			MOV	R5, PUINTL (SP) ; SAVE A POINTER TO THE . +1
142	40v0			r U V	Kolletidietout towar w contraction to the take
	4 - 7 0	000002		80	FLDS8 IGO FOR NEXT CHARACTER
		000707		BR	
144	4074	105166	ERNSOL	COMB	ERF+1(SP) JFLAG ERROR
		000033			
145		005000	ZENSBI	CLR	RØ IRESULT IS Ø
146		005001		CLR	\mathbb{R}_{1}^{1} , where \mathbb{R}_{1}^{1} , where \mathbb{R}_{1}^{1} , where \mathbb{R}_{1}^{1} , \mathbb{R}_{2}^{1} , R
147	4104	005002		CLR	R2
148	4106	005003		CLR	R3
149	4110	000167		JMP	STRS8 JGO PUSH RESULT AND RETURN
-		000450			
150	4114	120427	EXCS81	CMPB	R4,#1E
		000105			
	4120	001403		BEQ	EXTSB JUMP IF E
151				CMPB	R4,#1D
152	4122			UMP D	
		000104			2011-0 1 10 10 P P P P P P P P P P P P P P P
153		001362		BNE	ERRS8 JIF NOT E OR D THEN ERROR
	4130		EXTSS	MOV	R5,05P ISAVE POINTER TO END OF NUM +1
155	4132			DEC	OSP IDECREMENT NUMEND
156	4134	010366		Mav	R3, TEMP(SP)
		000042			
157	4140	005003		CLR	R3
158		020566		CMP	R5, END (SP)
		000044			
159	4146	002352		BGE	ERRS8 JJUMP IF NO ROOM FOR EXP
160	4150			MOVB	(R5)+,R4
161	4152			BIC	#177000,R4
101	4145	177600		~ 1 ~	
140	A . 8 6			СМРВ	R4,#1+ ICHECK FOR +EXP
104	≈1¥0	120427		enro	Nala A Inich Lau Ardi
	4460	000053		400	
	4102			BEQ	EF158
104	4104	120427		CMPB	R4,#1# ICHECK FOR #EXP
		000055			
		001010		BNE	ENMS8 JGO CHECK FOR NUMERIC
166	4172			INC	ESIGN(SP) /FLAG EXPONENT NEGATIVE
		000010			
167	4176	020566	EF1\$81	CMP	R5, END (SP)
		000044			
168	4202			BGE	ERR\$8
	4204		EF2581	MOVB	(R5)+,R4 JGET NEXT CHAR
	4206			BIC	#177600,R4
., U		177600		~ = =	
1 7 1	4010	120427	FNMCRE	CMPB	R4,#1 ICHECK FOR BLANK
714	-615		tes[4 [1 (2) (4) (8	Sector Ma	्रमालगङ्गात्वरः हा स्वत्यं प्रायम्भ सम्प्रम् । ३. स्वत्यं २९ - स्वत्यं सुध्य त्यः ३३.१७ स्वत्यं स्वत्यं प्रायम्भ सम्प्रम् । ३. स्वत्यं २९ - स्वत्यं स्वत्यं स्वत्यं स्वत्यं स्वत्यं स्वत्यं स्वत्यं स्व
	***	000040		A ME	EN 4 B
		001002		BNE	EN158 HIG WA STREAT REANK AS 0
1/3	4220	012704		MOV	#10, H4 ITREAT BLANK AS Ø
		000060			

174	4224	120427	EN158:	CMPB	R4,#10
· · · · ·		000060			
		002721		BLT	ERR\$8
176	4232	120427		CMPB	R4,#19
		000071			
177	4236	003315		BGT	ERRSB INOT A VALID CHAR
178	4240	162704		SUB	#60,R4 /CONVERT ASCII TU INTEGER
		000060			
		006303		ASL	R3 /X=10+X+D
		060304		ADD	R3, R4
181		006303		ASL	R3
182		006303		ASL	RJ
		000403		ADD	R4, R3 JEND OF ABOVE COMMENT
184	4256	020566		CMP	R5, END (SP)
		000044		0. 16	
***		002750		BLT	EF258 JJUMP IF MORE FIELD TO GO Esign(SP) JCHECK EXPONENT SIGN
186	4204	005766		TST	ESIGN(SP) JCHECK EXPONENT SIGN
4 D 70	1 14 M 14	000010		ure.	EN258 JJUMP IF IT IS +
187		001401		BEQ	
188	4272		61.069.	NEG	R3 IMAKE USER EXPONENT - R3,EEXP(SP) IGET COMPLETE DECIMAL EXPONENT
189	4274		EN2581	AUU	KOLCHVELANY JORI COMPETE ACTINE ENGINEED
100	1200	000014 016603		MOV	TEMP(SP) + R3
190	-300	000042		1101	I MALL POLY FILD
101	4304			JMP	SCLS8 IGO SCALE THE NUMERIC PART
191	백승성학	177412		A Lat.	AAMBA 199 AAUPH (UM UAUHUTA (HUI)
192	4310		DIV58:	TST	Rø
	4312		01.00i	BLT	DV158 JJUMP IF FRACT LEFT JUSTIFIED
		005366	DV2581	DEC	BEXP(SP) JLEFT JUSTIFY NUMERIC BITS
194	****	000000	0 V 8 9 0 1		Enviriante de la contraction de la contra Contraction de la contraction de la contra
105	4320			JSR	PC,LFT38
100		000464			
106	4324	100373		BPL	DV258
		012704	DV158:	MOV	#16. RA ISET FOR SIXTEEN ITERATIONS
131	- Q = Q	000020			in an θ μαθείται εκραθία το ματαγραφορια παραγολικατικό του αφοριατών. Π
198	4332			JSR	PC,RITS8
		000464		•••	
199	4336	010346		MOV	R3, = (SP)
200	4340			MOV	R2, = (SP)
201	4342			MOV	R1,-(SP) /INITIALIZE QUOTIENT
	4344	010046		MOV	R0,-(SP)
203	4346		DV358:	JSR	PC,RIT58
		000450			ng na sana ang na sana ang Na sana ang na s
204	4352			CLC	
205	4354	004767		JSR	PC,RITS8
		000442			
206	4360	012705		MOV	#2,R5
		000002			
207	4364			CLC	and the second secon
208	4366	004767	DV4581	JSR	PC,RAT\$B
<u> </u>		000430			
209	4372	066003		ADD	6(SP),R3
	.	000000		4	and the second
210	4376			ADC	R2
211	4400	005501		ADC	R1
212	4402	005500		ADC	RØ
213	4404	066602		ADD	4(SP),R2

38

180

(

*

000004 R1 ADC 214 4410 005501 215 4412 005500 ADC RØ ADD 2(SP),R1 216 4414 066601 000002 217 4420 005500 ADC RØ OSP, NO ADD 218 4422 061600 FCOUNT TWICE 219 4424 005305 DEC R5 UV458 220 4426 003357 BGT DEC R4 221 4430 005304 222 4432 003345 BGT DV358 #8, SP FPOP DIVIDEND 223 4434 062706 ADD 000010 SUB #3.BEXP(SP) 224 4440 162766 000003 000000 EEXP(SP) JBUMP DECIMAL EXPONENT 225 4446 005266 INC 000014 BLT JUMP IF MORE TO DO 226 4452 002710 DIVS8 JPOST NORMALIZE THE RESULT 227 4454 005366 FLTS81 BEXP(SP) DEC 000006 228 4460 004767 JSR. PC,LFTS8 000324 BCC FLT58 229 4464 103373 #200, BEXP(SP) ISET EXCESS 128 230 4466 062766 ADU 000200 000006 UND\$8 INUMBER TOO SMALL TO REPRESENT 231 4474 003475 BLE BEXP(SP) #377 232 4476 026627 CMP 000000 000377 JJUMP IF NUMBER TOO BIG 233 4504 003071 OVRSO BGT CLRB R3 234 4506 105003 235 4510 150203 BISB R2, R3 236 4512 000303 SWAB R3 237 4514 105002 CLRB R2 238 4516 150102 8158 R1, R2 239 4520 000302 SWAB R2 240 4522 105001 CLRB R1 RØ,R1 IMOVE OUT LOWEST ORDER BITS 241 4524 150001 BISB SWAB 242 4526 000301 R1 CLRB 243 4530 105000 RØ BEXP(SP), RO **FINSERT THE BINARY EXPONENT** 6158 244 4532 156000 000006 245 4536 000300 SWAB RØ **JPUT IN THE RIGHT ORDER** 246 4540 006066 ROR SIGN(SP) ITEST THE ARITHMETIC SIGN 000012 **JINSERT IN RESULT** 247 4544 004767 **JSR** PC,RITS8 000252 248 4550 005503 ADC R3 249 4552 005502 ADC R2 250 4554 005501 **IFINAL ROUND** ADC R1 ADC RØ 251 4556 005500 BVS UVRSU JJUMP IF OVERFLOW 252 4560 102443 BCS OVRSU 253 4562 103442 ERF(SP) ITEST REAL/DOUBLE FLAG 254 4564 105766 STR\$8; TSTB 000032

101	5254	004767 000646	ML1591	JSR	PC, M45\$9	JGET 4/5 FRACTION
102	5270	005266		INC	EEXP(SP)	MULTIPLY BY 10
		000000				
103	5274	162766		SUB	#3,BEXP(SP)	JAND DIVIDE BY 8
	- -	000003				
		000004				
1 (5 A	5100	003350		BGT	MULS9 JUMP	IF BINARY EXPONENT STILL POS.
		001424		BEQ	NOMS9 JJUMP	IF EXPONENT GONE NOW
		020127	n #VeQt	CMP	R1,#146314	BINARY EXPONENT IS NEGATIVE
100	2369		A14926	₩1.1 t	1.1.1.4.4004.4	
4	E	146314		рите	DV159 JUMP	IF NO ROOM FOR 5/4 FRACTION
		103014		BHIS	BEXP(SP)/#=3	who read to where he are not to the second
108	5314			CMP	DEXEVORITHMS	
		000004				
		177775				IF NOT ENOUGH BINARY EXP LEFT
	5322			BGT		IF NUL ENUVER DINARY EAR LERT.
110	5324			JSR	PC, M5459	MULTIPLY FRACTION BY 5/4
		000020				and a second
111	5330	005366		DEC	EEXP(SP)	IDIVIDE BY 10
		000006				
112	5334	062766		ADD	#2,BEXP(SP)	IMULTIPLY BY 4
	•	000002				
		000004				
113	5342	000402		BR	DV259	
		004767	DV1591	JSR	PC,RIT\$9	IDIVIDE BY 2
* * *	* • • •	001264	***	· · · ·		
115	5350	005266	DV289:	INC	BEXP(SP)	MULTIPLY BY 2
* * •	4040	000004	~ * ~ ~ ~ ~ ~	مشر د مشر	and design of the second se	
	6254	001354		BNE	DIVS9 JHIT I	T AGAIN IF BIN.EXP. NOT GONE
	0004	001034	s		POINT THE BINA	RY EXPONENT IS 0
117				AL INTA	FOINT TON TO TN	R1, R2, R3 AND R4,
118	57 m 15 m		J		PRALITUN 15 AN	OVERFLOW ACCUMULATOR
		005000	NOMS91	CLR		MULTIPLY FRACTION BY 5/4
120	2300	004767	N01591	JSR	PC, MD459	AUMPITURI LUCHRITON DI ANH
	No	000464				
121	5364			JSR	PC, ML859	JAND NOW BY 8
		000656				
	5370			TST	RØ	
123	5372	001003		BNE		IF AN INTEGER PART RESULTS
124	5374	005355		DEC	EEXP(SP)	DECREMENT EXPONENT
		000006				
125	5400	000767		BR	NO189 760 AG	AIN TO GET AN INTEGER PART
126			1	AT THIS	POINT THE MOST	SIGNIFICANT NON ZERO DIGIT IS IN
	5402	105766	NOUS9:	TSTB	TYPE (SP)	TEST CONVERSON TYPE
di Barr	****	000014	19 52 - 1 9			
108	5106	001424		BEQ	FFTS9 JJUMP	IF F FORMAT
				RORB	TYPE(SP)	anis a i an i an i an i an i an i an i a
129	5410			NUND	1 IF L SUF 2	
	6	000014		800	EFTS9 JJUMP	IF E FORMAT OR D FORMAT
		103114		BCC		
131	0416	005766		TST	EEXP(SP)	JG FORMAT
		000006				TE DERIN T & 1
	5422			BLT		IF RESULT <.1
133	5424			CMP	EEXP(SP),D(SP)	
		0000006				
		000022			· · ·	
134	5432	003105		BGT		IF RESULT >10++D
	5434			CLRB	TYPE (SP)	IMAKE TYPE F INSTEAD OF G
-		000014				

44

()

-

(

(

-10

Ŷ

136	5440	162766		SUB	#4,L(SP)	ILEAVE ROOM FOR BLANKS ON RIGHT
		000004				
		000024				
137	5446	156566		SUB	EEXP(SP) / D(SP)	JDECREASE D BY # OF DIGITS LEFT
		000006				
		000022				
138	5454	005066		CLR	P(SP) ISUSPE	ND P SCALE
• • •		000020				
139	5460	016605	FFTS91	MOV	EEXP(SP),R5	IF FORMAT
	***	000005		··•		
4 4 63	RARA	056505	FFFeGt	ADD	D(SP),R5	
140	0404		rr #334		DIGRITING	
	18 A 77 m	000022		400	DICOL DE	
141	34/0	066605		ADD	P(SP),R5	
		000020				- OUND WM LADING C A D O F
142	5474	004767		JSR	PC, RUD\$9	IRDUND BY ADDING 5+10++=P=D=E
		000640				
143	5500	016605		MOV	L(SP),R5	
		000024				
144	5504	166605		SUB	D(SP),R5	
-		000022				
145	5510	105765		TSTB	TYPE (SP)	
• • •	****	000014				
1 4 6	5514			BNE	FF559 JJUMP	IF NOT F CONVERSION
						COMBINE P AND EXP
14/	5516	066666		ADD	EEXPISEJIF(SF)	AUVIDINE PANU CAP
		000000				
		000020				
148	5524	003407		BLE	FF559 JJUMP	IF THERE IS NO INTEGER PART IN RE
149	5526	166605		SUB	P(SP),R5	
. .		000020				
150	5532	162705		SUB	#2,R5 ISIGN	SLOT IS S+L=D=E=P=2
0 - -		000002				
151	5536	004767		JSR	PC, ISNS9	INSERT SIGN AND CHECK WIDTH
101	0000			4 3 1		Tenerit serie with Alteria
		000744		D ra	FF359 JJUMP	TO INSERT IDGITS
		000416	ers.o.	BR		
153	5544	162705	6602A1	SUB	#3,R5 /SIGN	SLOT IS S+L=D=3
		000003				
154	5550	004767		JSR	PC, ISNS9	JGU INSERT SIGN AND CHECK WIDTH
		000732				
155	5554	112725		MOVB	#10, (R5) +	JINSERT LEADING Ø
		000060				
156	556a	112725		MOVB	#1., (R5) +	IINSERT .
	****	000056				
157	5564		FF4591	CMP	R5, L(SP)	ICHECK FIELD END
101	<u> </u>	000024	1.1	- () I		
4 6 9	RETO	103003		BHIS	FF389 JJUMP	TE FIFIA FULL
						JPUT IN ANOTHER LEADING ZERO
159	99/2	112725	•	MOVB	#10,(R5)+	FUL IN ANVIALS LEADING LEAD
		000000				
160	5576	000772		BR	FF459	
161	5600	016005	FF3591	MOV	L(SP),R5	
		000024				
162	5604	166605		SUB	D(SP),R5	
* ~ *	*****	000022			्राम् के के से के राज्यां के से के राज्यां के से क सिंह के सी की सी की सी की सी की सी की सी की स	
	RELA			DEC	R5 JLOCAT	ION FOR .
4 4 ² ²		005305			R5, PUINT (SP)	
	-max 1 (1)	010566		MOV	ROTHITOL	ADPURATOR FIG PROVITON
	0015	1				
164		000002				
164		000002 005766 000020		TST	P(SP)	

2

166	5622	003001		BGT	FF689
167	5624	005205	1998 - 1999 - 19	INC	R5 /POINT TO SLOT FOR FIRST NON-ZERO DIGIT
168	5626	166605	FF6591	SUB	P(SP), R5
		000020			
169	5632	004767	;	JSR	PC,DGS\$9 ////////////////////////////////////
		000714			
170	5636	105765		TSTB	TYPE(SP)
		000014		· · · · · ·	
171	5642	001467		BEQ	DNESS JALL THROUGH IF F FORMAT
		000433		BR	EFE39 IGO FINISH E FORMAT
		162766	EFTS9:	SUB	#4,L(SP) ;MAKE ROOM FOR E FIELD
H · <i>F</i>		000004			
		000024			
174	5654	005005		CLR	R5
		005766		TST .	P(SP)
		000020	÷ .		
176	5662	003700		BLE	FFES9 JPROCESS AS F FMT & RETURN TO EFMTE
177	5664	016605	*	MOV	D(SP),R5 JGET ROUNDING FACTOR
		000022			
178	5670	066505		ADD	P(SP),R51 ALLOW FOR P SCALE
		000020			
179	5674	004767		JSR	PC,RUDS9 JGO USE IT
		000440			
180	5700	016605		MOV	L(SP),R5 POINT TO SIGN SLOT
		000024	· · · · · · · · ·	en strander	
181	5704	166605		SUB	D(SP),R5
		000022			
182	5710	005305		DEC	R51 POINT SLOT # L=D=1
183	5712	010566		MOV	R5, PUINT(SP); SAVE LOCATION FOR .
		000002			
184	5716	166605	and the second second	SUB	P(SP), R51
		000020			
185	5722	005305		DEC	R5; SIGN SLOT = L=D=P=2
186	5724	004767		JSR	PC, ISN\$9 JGD CHECK WIDTH AND INSERT SIGN
		000556			
187	5730	004767	the second	JSR .	PC,DGSS9 JGD PROCESS ALL DIGITS
	•	000616			
188	5734	166666	EFES91	SUB	P(SP), EEXP(SP) / CORRECT EXPONENT FOR P
		000020			
		0000006		and the second	
189	5742	016503		MOV	L(SP),R3
		000024		5 - 14 ⁻	
190	5746	116623		MOVB	TYPE+1(SP), (R3)+ JMOVE OUT E OR D
		000015			
191	5752	016604		MOV	EEXP(SP)/R4
		0000006			
192	5756	002004		BGE	EXPS9 JJUMP IF EXPONENT POSITIVE
193	5760	005404		NEG	R4 JGET ABSOLUTE VALUE
194	5762	112723		MOVB	#1=, (R3) + ; INSERT =
		000055			
		000402		BR.	Ex159
196	5770	112723	EXP\$91	MOVB	#1 , (R3) + JINSERT BLANK FOR +
		000040			
197	5774	112713	EX189:	MOVB	HIØ, OR3 ICLEAR TENS DIGIT
		0000060	1		and the second
198	6000	162704	EX3591	SUB	#10,,R4 ITEST FOR TENS
		000012			

46

*

¢

199	6004	002402		BLT	EX239	
200	6006	105213		INCB	PR3	ACCUMULATE TENS
201	6010	000773		BR	EX389	
202	6012	062704	EX2391	ADD	#72,K4	JGET POSITIVE UNITS
		000072				
203	6016	110463		MOVB	R4,1(R3)	IMUVE UNITS OUT
		000001				
204	6022	062706	DNES91	ADD	#8.,SP	
		000010			* * * * *	
205	6026	012605		MOV	(SP)+,R5	
		012604		MOV	(SP)+,R4	
		012066		MOV	(SP)+,6(SP) IMOVE FLAG AND RETURN UP
100 Ber 1	- 1 - 12	000006				
208	6036	012566		MOV	(SP)+,6(SPI
6 U V	~U~V	000000		1964		
209	6040	022020		CMP	(SP)+, (SI	P)+ JFLUSH JUNK
		006126		ROL		ISET C BIT IF ERROR
211		000207		RTS		FRETURN TO CALLER
212	~U=0	00050/		PTV.		Lizeria alizza i en la superiori de la superiori.
				MIN TYOI	Y CONTENT	S OF R1 R4 BY 5/4.
213 214			7		RFLOW GOE	
	508a	010145	/ M6440+	MOV	R1,=(SP)	5/4X=X+X/4
216		010246	110-40-0-0	MOV	R2,=(SP)	
217	6054			MOV	R3,=(SP)	
				MOV	R4, = (SP)	
219	6056			JSR	PC,RITS9	• ¥ / 3
219	6060			13K	FUJRA 109	1×/2
000	40 A S A	000550		160	PC,RITS9	1×/4
220	6064			JSR	FUIRAIDO	
004	8070	000544		100	D A	ROUND
221	6070			ADC ADC	R4 R3	RUURU
222	6072					
223	6076	005502		ADC	R2	
224				ADC	R1	
225	6100	·.,		ADŬ	(\$P)+,R4	
	6102			ADC	R3	
227	6104			ADC	R2	
228	6106			ADC	R1	
		005500		ADC	RØ	
		062603		ADD	(SP)*,R3	
	6114			ADC	R2	
	6116			ADC	R1	
	6120	005500		ADC	RØ	
234	6122	062602		ADD	(SP)+,R2	
235	6124			ADC	R1	
236	0126			ADC	RØ	
237	6130			ADD	(SP)+,R1	
238		005500		ADC	RU	
239	6134	000207		RTS	PC I	IRETURN TO CALLER
240			1			
241	-		1			a subject the matter state of the state of t
242	6136		143591	MOV	#16,/R5	MULTIPLY R1R4 BY 4/5
		000020				
243	6142			JSR	PC,RITS9	
		000466				and the second
244	6146	010446		MOV	$R4_{i} = (SP)$	
245	6150	010346	· · · · · ·	MOV	R3,=(SP)	
246	6125	010240		MOV	R2, = (SP)	

247	6154	010146		MOV	R1,=(SP))	
248		004767		JSR	PC, RITS!		
		000452		~ • • •			
249	6160	004767		JSR	PC,RITSS	a	
243	0102	000446			r wyna i wi		
050	6180		•	MCN	HO BÌÀ		
250	0100	012700		MOV	#2, RØ		
	·	000005					
251	6172	004767	M52591	JSR	PC,RITSS	,	
		000436					
252	6176	066504		ADD	6(SP),R4	4	
		000000					
253	6202	005503		ADC	R 3		
254				ADC	R2		
255				ADC	R1		
256	6210			ADD	4(SP),R3	3	
	****	000004		T SHE HE		•	
257	6214			ADC	R2		
258		005501		ADC	R1		
259	6550			ADD	2(SP),R2	2	
		000002					
260	6224	005501		ADC	R1		
261	6226	061601		ADD	ØSP,R1		
262	6230	005300		DEC	RØ		
263	0232	003357		BGT	M5239		
264				DEC	R5		
265		003347		BGT	M5139		
256	6240	062706		ADD	#8,,SP	161.03	SH MULTIPLIER
6, V V	V U	000010		, y y y	HAT 1	rti kat≱r	ALL HAPTER
067	6044			ÜTE	0.0		
267	6244	000207	•	RTS	PC		
268			I .				6 6 6 6
269			1		Y THE CON		
270			1		FLOW IS A		IPATED
271	6246		ML8591	MOV	R5,=(SP)	r	
272	6250	012705		MQV	#3,R5		
		000003					
273	6254	006304	M81591	ASL	R4		
274	5255	006103		ROL	RB		
275	6260	006102		ROL	R2		
276	6262	006101		ROL	R1		
277		006100		ROL	RØ		
		005303		DEC	RS		
		003371		BGT	M8159		
		012005		MOV	(SP)+,R5	1	
1		000207		ŘTS	PC		
		005726	ERR\$9;	TST	(SP)+		
283	6300	016603		MOV	S(SP),R3)	POINT TO FIELD BEGIN
		000026					
284	6304	016604		MOV	L(SP),R4		JGET FIELD END +1
		000024					
285	6310	105760	14. J. 14.	TSTB	TYPE (SP)		ICHECK IF END MODIFIED
200	***	000014	л. А. С.		. La n r ja r j		TALFAR TI PER RUDITTTC
286	6214	001402		BEQ	STCED	110	THIS IS F FORMAT
287	0210	062704		ADD	#4,84	FUI	BACK EXPONENT SPACE
		000004	a - c - c				
288	0322	112723	ST\$\$91	MOVB	#!*,(R3)	•	FILL FIELD WITH +
		000052					
289	6326	020304		CMP	R3, R4		

48 (

(

(

(**16**)

ø

290	6330	103774		BLO	STS59 JJUMP IF MORE TO GU
		005166		COM	TYPE(SP) JFLAG ERROR
	- .	000014			
292	6336	000631		BR	DNESS
293	****		1		gerate mentant international de la constant
294			1	ROUND	THE CONTENTS OF RØ R4 TO THE PRECISION
295			1		IED BY R5.
296			1		ROUTINE IS SHORTER THAN THE TABLE THAT
297			1		ISE WOULD BE NEEDED.
298	6340	020527	RUD591	CMP	R5, #20,
		000024			
299	6344	003054		BGT	RU159 JJUMP IF NOT WORTH ROUNDING
300		010566		MOV	R5, BEXP+0+2(SP) ;SAVE ROUNDING PRECISION IN TEMP
000		000006		1.1.1.1	
301	6352	001452		BEQ	RU359 JJUMP IF ROUND IS TO LEADING DIGIT
302		002450	4	BLT	RU159 JUMP IF NO ROUNDING TO DO
303		010046		MOV	RØ,=(SP)
304		010140		MOV	R1, = (SP)
		010246		MOV	R2,=(SP)
306		010346		MOV	R3,=(SP)
307		010446		MOV	R4, = (SP)
308	-	012701		MOV	#100000,R1 / INSERT .5
000	00/0	100000		I Q Y	HINDHENSUT LTUARDI RA
300	6474	005002		CLR	R2
					R3
		005003		CLR	
311		005004		CLR	R4 BEXP+0+2+10.(SP) JCOUNT PRECISION
312	0402	005360	KUP 391	DEC	BEXP+0+2+10, (SP) JCOUNT PRECISION
	6400	000020		860	
		001411		BEQ	RODS9 JJUMP IF DONE
014	0410	004767		JSR	PC,M4559 IMULTIPLY BY 4/5
7.5	<i></i>	177522		1 = 13	00 D1##0
910	0414	004767		JSR	PC,RITS9
242	6 4 D m	000214		160	DP DITSO
316	0440	004767		JSR	PC,RIT\$9
317	6 A 7 A	000210		160	PC,RITS9 ;DIVIDE BY 8
317	0454	004767		JSR	PC,RITS9 JDIVIDE BY 8
	& . 1	000204		C1 /74	AREAD ICA CULAK IS NANE WITH SAATAD
318			860.0.	BR	RDFS9 JGO CHECK IF DONE WITH FACTOR
319			RDD\$91	CLR	RØ (SP)+,R4 JADD FRACTION TO RND FACTOR
320		062604		ADD	
		005503		ADC	RJ
322	6440			ADC	
323				ADC	R1 (SP)+,R3
	6444	062603		ADD Adc	R2
		005502			
326		005501		ADC	R1 (SP)+ 80
327		062602		ADD	(SP)+,R2
328	6454	005501		ADC	
329		062601		ADD	(SP)+,R1
330		005500		ADC	
		062600	000-00-	ADD	(SP)+,RØ
332	0404	022700	KUC2AI	CMP	#10./R0
	617-	000012		HAT.	
		003002		BGT	RU189 JJUMP IF NO OVERFLOW
334	04/2	005265		INC	EEXP+2(SP) JBUMP DEGIMAL EXPONENT
	£ . ** *	000010	BUL - 0 -	5 .	
335	0476	000207	KN12A1	RTS	PC JRETURN TO CALLER

\$DI	_GØ3	MACRO VI	R04-14 0	7=SEP=72	11:43 PAGE 15 32
1				TITLE	SDLG03
2				, IFDF	CNUS10
3 4			7	DLOG	V003A
5		•	I COPYR	IGHT 197	1, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS
7 8			1	GLOBL	DLUG, DLUG10, SERRI
9	-			IFNDF	FRU
10				.GLOBL	SPOLSH, SADD, SSBD, SMLD, SDVD, SID, SPOPR41
11	1			.ENDC	TRAN DLOG AND DLOG10 FUNCTIONS
12			- F		SEQUENCEI
14			1	JSR	R5, DLOG (OR DLOG10)
15			1	BR	A
16			1	, WORD	ARGUMENT ADDRESS
17			7 4 1		A STATE FROM LAPPER FADALE TO DO DO
18			1	RETURNS	LN(ARG) (OR LOG10(ARG)) IN RØ = R3.
19 20		000000	. F	RØ=%0	
21		0000001		R1=%1	
22		000002		R2=%2	
23		000003		R3=%3	
24		000004		R48%4	
25		000005		R5=%5	
26 27		000000 000007		5P=%6 PC=%7	
28		000000		F0=%0	
29		000001		F1=%1	
30		000002		F2=%2	
31		000003		F3=%3	
32	an at a station of			IFNDF	
33	06650	011740	DLUG10:	MOV Br	@PC,=(SP] JGET 0004XX AS A FLAG Logs10
34 35	06654		DLUG:	CLR	-(SP) /FLAG DLOG
36	06656				R5,-(SP) ISAVE RETURN POINTER
37	06660			MQV	2(R5),R4 JGET ARG ADDRESS
		000002		1	
38	06664	062704		ADD	#8.,R4 JPOINT TO LEAST SIGNIFICANT PART
3,0	46674	000010 012746		MOV	#147572,=(SP)
0.9	00010	147572		11 9 Y	and an end of the second se
40	06674	012746		MOV	#173721,=(SP)
		173721			
41	06700	012746		MOV	#071027, =(SP) ; PUSH =1/2*LN(2)
		071027			11-77601 -1003
42	00704	012746		MOV	#137061, = (SP)
4.4	06710	137661		SUB	#8., SP ; GET WORK SPACE
	a ₹7 € Ŭ	000010		1. 1 . 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	and the second
44		014446		MOV	=(R4),=(SP) JGET ARG
45		014445		MOV	=(R4), =(SP)
46		014446		MOV .	=(R4), =(SP)
47 48		014446		MOV Ble	=(R4Ĵ,=(SP) Errsiø jjump if not positive
40		006316		ASL	esp
50		116666		MOVB	1(SP),26.(SP) JGET EXPONENT

 $(\)$

(

(-----

¢,

æ,

(

		000001			
		000032			
51	06736	112766	MOVB	#200/1(SP) /TRANSFORM ARG TO (1/2,	1)
		000200			
		000001			
52	06744		ROR	esp	
	-1-				
53	00746		MOV	#137145, "(SP)	
		157145			
54	00752	012746	MOV	#031771, = (SP)	
		031771			
55	06756	012746	MOV	#002363,=(SP) /PUSH 1/2+RODT2	
		002363			
56	06762	012746	MOV	#040065,=(SP)	
**	547 ₹ 7 3 184	040065		and and the second second provide the second s	
57	06766		MOV	14.(SP),-(SP) ;PUSH X	
57	00/00		n u v	TAT (AL)) (AL) ILAOU V	
		000016		4.4. 49m3	
58	007/2	016646	MOV	14.(SP), -(SP)	
	_	000016			
59	06776	016646	MOV	14, (SP), = (SP)	
		000015			
60	07002	016545	MOV	14.(SP), -(SP)	
		000016			
61	07005		MOV	#157145,=(SP)	
~ .	0,0*0	157145			
			14 m 11	1000 4 7 m 4	
62	0/015	012746	MOV	#031771,=(SP)	
		031771			
63	07016	012746	MOV	#002363,=(SP) ;PUSH 1/2+ROOT2	
		002363			
64	07022	012740	MOV	#040065, = (SP)	
		040065			
65	07026	004467	JSR	R4, \$POLSH ;ENTER POLISH MODE	
0.0	0/060		yon	References ferrer reften none	
e 20		012612		BODS UDELA SART CONS. LOCK IN DAGADA	
66		0007001	. WORD	\$\$80,UP\$10,SADD,SDVD /GET (X-ROOT2)/	
	07034				
	07036				
	07040	0122101			
67				7 (X+R00T2)	
68	07042	0072461	WORD	DUPS10, DUPS10 JGET THREE COPIES	
	07044				
69		0161461	. WORD	SMLD JSET UP POLYNOMIAL	
70		0175761	WORD	SPOPRA POP Y	
		0071441	WORD	REG\$10	
		016146 XPDS101			
12			a NURU	SMLD, SADU, SMLD, SADD, SMLD, SADD, SMLD, SADD	
	-	0007041			
	07000	0161461			
	07002	0007041			
	07004	0161461			
		0007041			
		0161461			
		0007041			
38 12	10 · · · · · · · · · · · · · · · · · · ·			6 MI (6) 6 A(1)) (6) (6) (7) (6) (7) (7) (7) (7) (7) (7) (7) (7) (7) (7	
13		0161461	, WORD	SMLD, SADU, SMLD, SADD, SMLD, SADD	
	20 C 20	0007041			
		0161401			
	07102	0007041			
	07104	0161461			
	7	0007041			
74		аранан (т. 1997) Сталан (т. 1997) Сталан (т. 1997)		JEXPAND POLYNOMIAL	
				त्र चल्परागणाः गार्थित्व प्रयाः २४४ राज्यात्वकः वी िरिश्वीर्थिः आदिति आदिति विद्यारि विद्यार्थिः २००० राज्यातः राज्या	

75	07110	0072321	l	.WORD	SCLS10,S	10,PL2\$	S10,5MLD JGET LN(EXP)	
		0160461						
	07114	0072701						
	07116	0161461						,
M 12				unan	SADDIEXI	\$10	COMBINE WITH FRACTION	
/0	1977 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 - 1987 -	0007041		, WORD	BAUNIEAT	a 1 Q	TRANSTAC VALUE VALUE TAN	
	0/125	0073121						
77							JAND CHECK IF DONE	
78	07124	016146	1	.WORD	SMLDIEXI	\$10	IMULTIPLY BY LOG10(E) AND RETURN	iler -
	07126	0073121	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,					
79	07130		ERR\$10:	ADD	#24 + SP	IFLUSH	JUNK	
		000030						
80	07134			JSR	R5, SERR	· · · ·		
v 0	974 * **	012040						
04	07140			BR	EROS10			
81	07140	000504						
82	07142	004		BYTE	4			
	07143	003	_	.BYTE	3			
84		-	1			0		
85	07144	012704	RE45101	MOV	#CON\$10+	8.,R4	POINT TO COEFFICIENTS	
		0074501						
86	07150	012705		MOV	47,R5	ISEVEN	CONSTANTS	
		000007			·			
87	07154	000404		BR	STC\$10			
88	07156	010346	STKS101	MOV	R3,-(SP)			
89	07150	010246	w1	MOV	R2,=(SP)			
90	07162			MOV	R1,-(SP)		IPUSH Y	
					R0,=(SP)			
91	07104		C	MOV		SPI	IPUSH COEFFICIENT	
92	07166	014446	STCS101	MOV	=(R4),=(ILAGU MAELTATEMI	
93	07170	014446		Mav	= (R4) ;= (
94	07172	014446		MOV	=(R4),=(
95	07174	014446	Rest in the	MOV		SP)		
96	07176	005305		DEC	R5	ICOUNT	CONSTANTS	
97		003366		BGT	STK\$10			
98	07202			MOV	#XPD510,	K4	SET UP RETURN TO LIST	
	terff Hann ™ Meda	007054				-		
99	07206			JMP	@(R4)+			
100		*****		W (),				
		010868	105101	MOV	(SP)+,22	(SP)	IMOVE ITEM TO WORK SPACE	
101	1 7210		UP3101	ΗŲ Y	(01) 122	a Cort	ADAAR FIELD IN LOUD OLUME	
		000026		14 m \r	1-01+ 00	(00)		h
104	2 7214			MOV	(SP)+,22	.1341		
		000026						
103	7220	012566		MOV	(SP)+,22	• (SP)		
		000026		2				
104	7224	012666		MQV	(SP)+,22	. (SP)		
		000026			· . · ·			•
105	5 7230	000134		JMP	@(R4)+			
106			1					
107		005046	SCLS101	CLR	=(SP)			
108		156616	न प्रहार पर प्राह कर है। जनसंख्या	BISB	12. (SP),	ØSP	JGET EXPONENT	
104	5.4 44	000014		ल क्षा के लि	**********		y month i yezhon na servizioù de la servizioù d Na	¢.
1 10	7044			SUB	#200,0SP		FREMOVE EXCESS 128	
105	7240	162716		900	HEDUITOF		- 计可计算机 建冷却管理者 机动物	
		000200		TMD	A (0 4)			
112		000134		JMP	@(R4)+			
111			I					
112	2 7246	016646	DUPS101	MOV	6(SP),=(97)		
		000006						
113	7252	016546		MOV	6(SP),=(SP)	IDUPLICATE STACK ITEM	
		000006						

(

xý,

114	7256	016646		MOV	$6(SP)_{,=}$
		0000006			
115	7262	016646		MOV	6(SP),=(SP)
•		000000			
116	7266	000134		JMP	@(R4)+
117	1 100 1 10		t		
118	7270	012746	PL25101	MOV	#147572, = (SP)
110	1210			1101	
	10 - 1 -	147572		Maria	4177701 - (68)
119	12/4	012746		MOV	#173721, = (SP)
		173721			
120	7300	012746		MOV	#071027,=(SP) ;PUSH LN(2)
		071027			
121	7304	612746		MOV	#040061, = (SP)
		040061			
122	7310			JMP	@(R4)+
	1010	000104		* 111	
123	7710		EXISION	DECB	11. (SP) JCHECK FOR ALOG10
124	1215	105366	CX + 3 1 ()	NEVO	TTOPALL LAURAU LAU WEARTH
_		000013		17 Mari	
	7316			BLT	LGTS10 INO, DUNE
126	7320	012746		Mov	#024162, • (SP)
		024162			
127	7324	012746		MOV	#124467,=(SP)
		124467			
128	7330	012746		MOV	#055730,=(SP) ;PUSH LOG10(E)
<u>+</u> <u>-</u> + - + + - +		055730		a dan a	
100	7-14			MOV	#037736, = (SP)
129	1394	012740			
		037736		Tum	A FD 41 .
130	7340			JMP	@(R4)+
131	7342	012600	LGT\$101	MOV	(SP)+,RØ ;POP RESULT
132	7344	012001		MOV	(SP)+,R1
133	7346	012602		MOV	(SP)+,R2
134	7350	012503		MOV	(SP)+,R3
135			ER0\$101	MOV	(SP)+,R5 JRESTORE RETURN
136	7354			TST	(SP) + JFLUSH FLAG
	7356			RTS	R5
	/ 390	DODED4		.ENDC	
138			•	. CHUL	
139			i 🖡 en Arnigae'		ali na sente de la constante d
140				. IFOF	FPU
141			DLOGIØI	MOV	OPC, RA; GET 0004XX AS DLOG10 FLAG
142				BR	LOGS101
143			DLUGI	CLR	R4; GET 0 AS DLOG FLAG
144			LOGS10:	SETD	DOUBLE PRECISION FP
145				SETI	SHORT INTEGERS
146				MOV	HFCOB10, RO; POINTER TO CONSTANTS
147				LDD	@2(R5),F2; GET ARG
148				CFCC	
				BLE	ERRS10; JUMP IF NOT POSITIVE
149					
150				STEXP	
151				LOCID	R1,F3; CONVERT TO FP FORM
152				MULD	(RØ)+,F31 SCALE FACTOR=EXPONENT+LN(2)
153				LDEXP	#0,F2; TRANSFORM ARG TO(1/2,1)
154			ja kang pa		
155				LDD	F2,F1;
156				SUBD	(RØ), F2; X=1/2+SQRT(2)
157				ADDD	(RØ)+,F11 X+1/2*SORT(2)
157				DIVD	F1,F2; W=(X-R00T2)/(X+R00T2)
				LOD	F2,F1
159					in Comença França (Comença de Comença de Comença de Comença de Com

1					
160			MULD	F1,F1;	Y = W * * 2
161		1		· · · · · · · · · · · · · · · · · · ·	
162			MOV	#6,R1;	COUNT CONSTANTS FOR POLYNOMIAL
163			LOD	(R0)+,F01	INITIALIZE ACCUMULATOR
164		XPUS10:		F1,F0;	
165			DEC	R11	COUNT
			ADDD	(RØ)+,FØ7	F0:= Y+F0 + C(I)
166					
167			BGT	XPDS101	LOOP
168			MULD	F2, F0;	
169			ADDO	(R0)+,F01	F01 = W + F0 = 1/2 + LN(2)
170			ADDD	F3, F0;	ADD SCALE FACTOR FOR EXPONENT
171			TST	R41	TEST DLOG10 FLAG
172			BEQ	LGTS10;	
173			MULD	(R0) . F01	DLOG10 = DLOG+LOG10(E)
174					का के हो पर भग को भया। पर के में प्रायों के पर पर पर की भया है। संस्थान प्रायों के स्थान पर के में प्रायों के पर पर की भया है।
		1	STD	F0,=(SP)/	MOVE RESULT TO STACK
175		LGTS10:			
176			MOV	(SP)+, RØ1	AND TENCE TO ROR3
177			MOV	(SP)+,R11	
178			MOV	(SP)+,R21	
179			MOV	(SP)+,R31	
180			RTS	R51	EXIT
181		1		• · · · · · · · · · · · · · · · · · · ·	
182		ERR\$101	JSR	R5, SERR;	ERROR 4,3
		PK00101			EXIT - NO STACK CLEANUP REQUIRED
183			RTS	R51	EXTI # NO OTACH CLEANOR REGULARD
184			BYTE	4	
185			BYTE	3	
186		1			
187		1		EPENDENT CONSTAN	
188		1	RØ POIN	TS AT CURRENT CO	NSTANT IN FPU VERSION
189		1			
190		FCUS101	.WORD	040001,071027;	LN(2)
		1.040464		173721,147572;	977 7 7 984 g
191		•	.WORD	1/0/61/14/0/2/	
192		I Alexandre State		A	4 40 - Conv (m)
193			. WORD	040005,0023631	1/2+SQRT(2)
194			.WORD	031771,1571451	
195			. ENDC		
196		3			
197	7360 037455	i	.WORD	037455,106270	1.16948212488
	7362 106270				
108	7304 157166		WORD	157166,174770	
190			# 19 M 17 M	1011-011.4110	
	7366 174770		¥.		
199		1			A 101143808#049
200	7370 037471		, WORD	037471,072731	1,1811136267967
	7372 072731		•		
201	7374 137716	n a stàitean anns an	WORD	137716,117115	
	7376 117115				
202		7			
	7400 037543		.WORD	037543,111153	1,22223823332791
6 U U			******		ह क्यू तन आदि 244 स्थर स्थास पर पर पर पर किस्टी कि
A	7402 111153		anan	ABAINA AREAR	
204	7404 060101		, WORD	060101,135465	
	7406 135465				
205		1			
206	7410 037022		, WORD	037622,044436	1,2857140915904889
	7412 044430				
207	7414 007306		NORD	007306,063062	
200 X.C 7			· · · · · · · · · · · · · · · · · · ·	a service and a service of the servi	
208	7416 063062	•			

56

(

(

(

de.

\$DLG03 MACRO VR04=14 07=SEP=72 11:43 PAGE 15+

 $\mathfrak{A}_{i'}$

0

209		037714		. WORD	037714,146314	1.40000000120604536	5
		146314					
210	7424	153450		. WORD	153450,165773		
~	7426	165773					
211							
		13 A 13 12 18 19		1.4 m m m	040000 195050		
212		040052		, WORD	040052,125252	1.666666666663366089	14
	7432	125252					
213	7434	125247		.WORD	125247,004643		
		004643					
014		~~~~~					
214							
210		040400	CONS10:	, WORD	040400,000000	12.000000000000000000000000000000000000	ŕ.
	7442	000000					
216	7444	000000		WORD	000000,000057		
	7446				· · · · ·		
	/ = -0	000001		TENE	PDI		
217				. IFOF	FPU		
218			1	MORE UN	RDER-DEPENDENT CO	NSTANIS	
219			1				
220				.WORD	137661,0710271	-1/2 + LN(2)	
221				WORD	173721,147572;		
				# n win w	1,0, -1, 1,0, -1		
222			₩				e +
553				WORD	037736,055730;	LOG10(E)	
224				. WORD	124467,0241627		
225				ENDC	•		
226				ENDC			
220							

1				TITLE	\$UNT02
2				.1FDF	CNOS11
3 4	• • •		1	SDINT	VØ02A
5 6				.GLOBL	SDINT
7			1	COPYRIGH	HT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
8			1	OTS INTE	ERNAL FUNCTION TO FIND THE INTEGER
9			j		A DOUBLE PRECISION NUMBER.
10			1		IN THE POLISH MODE.
11		000000		RØ=%0	
12		000001		R1#%1	
13		000002		R2=%2	
14		000003		R38%3 R48%4	
15 16		000004		R5=%5	
17		0000006		SP=%6	
18		177304		MG#17736	34 - 1997 - 19
19		177310		ASH=1773	
20		000000		FØ=%0	
21		000001		F1=%1	
22				.IFDF	FPU
23			SDINT	.WORD	170011 //SETD
24				, WORD	172426 JILDD (SP)+,FO JLOAD ARG
25				WORD	171467,4 JIMODD ONE,FØ JGET INTEGER PAR 174146 JISTD F1(SP) JPUSH INTEGER
26				WORD	
27				JMP	
28				.WORD	040200,0,0,0 /FLOATING 1.
29				.ENDC .IFNDF	PPU
30	07.50	010600	SDINT:	MOV	(SP)+,RØ IPOP DOUBLE ARG
31		012000	OUANIA.	MOV	(SP)+,R1
32 33		012002		MOV	(SP)+,R2
34	page 1	012603		MOV	(SP)+,R3
35	07400			MOV	R4,-(SP)
36	07462			MOV	R5, - (SP)
37	07464	010004		MOV	RØ, R4 JGET EXPONENT
38	07466	006104		ROL	R4
39		105004		CLRB	R4 Constant of the second s
40		000304		SWAB	R4
41	07474	162704		SUB	#270,R4 /CONVERT TO =SHIFT COUNT
		000270		n o f	SUMAL TUND TO ADD MUST BE TAREPED ALDEADY
		002041		BGE	DNES11 JUMP IF ARG MUST BE INTEGER ALREADY
43		022704 177710		CMP	#=70/R4
44		002405		BLT	SHES11 JJUMP TO GET INTEGER PART
		005000		CLR	RØ JANSWER IS Ø
		005001		CLR	
			C23\$111	CLR	R2 R3
48		005003		BR	NES11
49	07522	000431	SHF \$11:	U R	MARMAN TO THE REPORT OF A DESCRIPTION OF A A DESCRIPTION OF A DESCRIPTIONO
50 51	01022		AU1.04444	IFNDF	EAE&MULDIV
	07520	010405		MOV	R4, R5 ISAVE A COPY OF SHIFT COUNT
		022704		CMP	#-32, R4 JCHECK LOW OR HIGH TRUNCATION
		177740		π. · · ·	
54	07530	002415		BLT	RURS11

58

(

4

4

ø

IGO CLEAR LOW ORDER HALF C23\$11 55 07532 001770 BEQ #32. R4 IDO HIGH ORDER 56 07534 062704 ADD 0000000 57 07540 010405 MOV R4, R5 ISHIFT OUT FRACTION BITS 58 07542 006000 RR1s11: ROR RØ 59 07544 006001 ROR R1 60 07546 005204 INC R4 61 07550 002774 BLT RR1511 R1 ISHIFT IN 01S ASL 62 07552 006301 AS1\$111 ROL RØ 63 07554 006100 R5 64 07556 005205 INC AS1511 BLT 65 07500 002774 IGO CLEAR LOW ORDER BR 623\$11 66 07562 000754 IMOVE OUT FRACTION BITS 67 07564 006002 ROK\$11: ROR R2 ROR R3 68 07566 006003 69 07570 005204 R4 ICOUNT LOOP INC 70 07572 002774 BLT RURS11 71 07574 006303 ASL511: ASL R3 72 07576 006102 ROL R2 R5 ICOUNT LOOP 07600 005205 INC 73 BLT ASLS11 07602 002774 74 . ENDC 75 . IFOF 76 EAE IPOINT TO MQ MOV HMQ, K5 77 78 .ENDC . IFOF MULDIVIEAE 79 ICHECK FUR HIGH OR LOW ORDER TRU CMP #=32.,R4 80 BLT R23\$11 ILOW 81 C23\$11 ICLEAR LOW ORDER BEQ 82 #32., R4 THIGH URDER PARTS Rø1s11: ADD 83 . IFOF MULDIV 84 . WORD R4, RØ ISHIFT OUT FRACTION 85 073004 **JJASHC** ISET TO SHIFT LEFT 86 NEG R4 . WORD R4.RØ IBRING IN THE 0'S 073004 IJASHC 87 .ENDC 88 . IFOF EAE 89 HIGH ORDER TO AC,MG MOV R1,085 90 RØ,=(R5) MOV 91 ISHIFT RIGHT MOV R4, @#ASH 92 R4 93 NEG R4, @#ASH ISHIFT LEFT MOV 94 **FRESULT TO REGS** (R5) +, RØ 95 MOV ØR5, R1 96 MOV .ENDC 97 JGO CLEAR LOW ORDER C23511 98 BR MULDIV . IFOF 99 073204 11ASHC R4, R2 R235111 , WORD 100 NEG R4 101 . WORD 073204 11ASHC R4, R2 ISHIFT IN 0'S 102 . ENDC 103 . IF OF EAE 104 RJ, PR5 ILOW DROER TO AC, MQ R23511: MOV 105 R2, = (R5)MOV 106 R4, CHASH JOUMP BITS MOV 107 NEG R4 108 R4, @#ASH IBRING IN 015 MOV 109 RESULT TO REGS 110 MOV (R5) + R2

SONT02 MACRO VR04=14 07=SEP=72 11:43 PAGE 16+

111				Mav	085,83			
112				. ENDC	n an Anna an Airteanna An Airtean			
113				, ENDC				
114	7604	012005	DNES11:	MOV	(SP)+,R5			
115	7606	012604		MOV	(SP)+,R4			
116	7610	010346		MOV	RJ,=(SP)		PUSH	RESULT
117	7612	010246		MOV	R2,=(SP)			
118	7614	010146		MOV	R1, = (SP)			
119	7616	010046		MOV	R0,-(SP)			
120	7620	000134		JMP	@(R4)+	TRETURN		
121				. ENDC				
122				ENDC				
** "								

¢.

12				.TITLE .IFDF	SDRØ2 CND312
					SDR, SERK
3				,GLOBL	THE DOUBLE PRECISION TO REAL CONVERTER
4			1	SDR	THE DUDGE FRECTOION IN DEAR CONVERTER
5				SDR	V002A
6					1992 m
7			1	COBVETS	GHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
8 9			1	ROUND 1	THE TOP STACK ITEM TO REAL FORMAT.
10		000004	9 -	R45%4	INE THE ALMON STRUCTS THREE CAMPAGES
11		000000		R5#%5	
12		000000		SP=%6	
13		000000		F0=X0	
14		000000		. IFDF	FPU
15			SDR:	WORD	170001 11SETF
16				WORD	177426 JILDCDF (SP)+,FØ JCONVERT ARG
17				WORD	170000 FICFCC JGET CONDITION CODES
18				BYS	OV1812 JJUMP IF OVERFLOW ON ROUND
19				WURD	174046 11STF FØ,-(SP)
20				JMP	@(R4]+;
21				.ENDC	
22				IFNDF	FPU
23	07622	006166	SDR:	ROL	4(SP) FROUND LOW ORDER PART
		000004			
24	07626	025566		ADC	2(\$P)
	-	000002			
25	07632	205516		ADC	PSP
26	07634	103406		BCS	OVR\$12 JJUMP IF OVERFLOW
		102405		BVS	OVR\$12
28			DR1\$12;	MOV	(SP)+,2(SP) ;MOVE HIGHEST ORDER PART
		000002			
29	07644	012066		MOV	(SP)+,2(SP) IMOVE LOW ORDER REAL
		000002			
30	07650			JMP	Ø(R4)+ IRETURN
31	07652		OVH\$121		(SP)+, (SP)+ JFLUSH ARG
32	07654	022525		CMP	(SP)+, (SP)+
33		1		. ENOC	
34	07656		0v1\$12;	JSR	R5, SERR JERROR 3, 23
		012124			
35	07662			BR	DR2\$12
	07664	003		BYTE	3
37	07665	027		BYTE	23.
38	07666		DR2\$121	CLR	-(SP) IRETURN 0.
39	07670	005046		CLR	= (SP)
40	07672	000134		JMP	@ (R4) +
41				.ENDC	

		***(6	SDSN04	
		.TITLE .IFOF	CNDS13	an an an Araba an Araba. An Araba an Araba.
	1	8 4 F W F		
	1	DSINCS	V004A	
	j j			
	, 00	PYRIGHT 197	1, DIGITAL EQ	UIPMENT CORPORATION, MAYNARD, MASS
	1			
		in a start and		
		, GLOBL	DSIN, DCOS;	
		. IFNDF	FPU -	IN WAND BATHT BOOLSU BOODAN
		.GLOBL	JAUUI JODAI JM	LD, SDVD, SDINT, SPOLSH, SPOPR4;
	•	, ENDC DSIN	DCOS THE	DOUBLE PRECISION SIN AND COS
	· · ·	FUNCTIO		DOODEL FREDIVING ONH MAD GOO
	1		SEQUENCE	
	· · · ·	JSR	R5, DSIN (OR	DCOS)
	,	BR	A	
	and a part of	WORD	ARG ADDRESS	
	141			
	1	RETURNS	SIN UR CUS D	F ARG IN RØ = R3,
	0000	R0=%0		
	0001	R1=%1		
- · ·	0002	R2=X2	S. S. S.	
	0003	R3=%3		
	0004	R4=%4 R5=%5		
	0005	SPE%6		
	0007	PC=%7		
	0000	F0=%0		
	0001	F1=%1		
	0002	F2=%2		
	0003	F3=%3		
-	••••••••••••••••••••••••••••••••••••••	. IFNDF	FPU	
07674 01	0546 DCUS		R5,=(SP)	ISAVE RETURN POINTER
07676 01	6504	MOV	2(R5),R4	JGET ARGUMENT ADDRESS
00	0002			
	5046	CLR	-(SP) IMAK	E ROOM FOR QUADRANT FLAG
07704 01		MOV	6(R4) = (SP)	· "你们就是你们都能是我的事情。""你们
	0006	Be on the	11811 - CAR	
07710 01		MOV	4(R4), m(SP)	
00 07714 01	0004	MOV	2(R4),=(SP)	PUSH ARGUMENT
	0002	ET MARK	w Class Care	···································
07720 01		MQV	@R4,=(SP)	
07722 01		MOV	#064302, = (SP) a sequencia de la construcción de la const de la construcción de
	4302		· · · · · · · · ·	
07725 01		MOV	#121041,=(SP	
12	1041			
07732 01	2746	MOV	#007732, - (SP) /PUSH PI/2
	7732			
07736 01		MOV	4040311, - (SP)
	0311	tan	15 g m 13 m i m i i	• 28 7 5 3 AT TOL MART
07742 00		JSR	R4, SPOLSH	JENTER POLISH MODE
	1075	25 m 24	SADD - ANALS	LOBOLVN - SAMAVIDA JON
07746 00		.WORD	SADD, SNC\$13	;COS(X)=SIN(X+PI/2)
07750 01		8 Mm M	DK _febs	CAVE DETIEN
01135 01	0546 DSIN	I MOV	R5,=(SP)	ISAVE RETURN

SOSN04 MACRU VR04=14 07=SEP=72 11:43 PAGE 18

62

喇

-0

Ŷ

48	07754	016504	MOV	2(R5),R4 JGET ARGUMENT ADDRESS
		000002	-	
49	07760	005046	CLR	- (SP) JMAKE ROOM FOR QUADRANT FLAG
50		016446	MOV	6(R4),=(SP)
50	01146	_	DUT -	
	. Did yes Al	000006	5 m m 1 /	
51	07706	016446	MOV	4(R4), -(SP)
		000004		
52	07772	016446	MOV	2(R4) = (SP)
		000002		
53	07776	011445	MOV	OR4,=(SP) JPUSH ARGUMENT
54		006316 SNC5131	ASL	OSP ICLEAR SIGN AND SAVE IT
		-	ROR	8. (SP) JIN QUADRANT FLAG
55	10005	006066	nun	OFFOLD ATH MONDUMIEL LEVA
_		000010	(1 - 12	
56		006016	ROR	0 SP
57	10010	012746	MOV	#064302, = (SP)
		064302		
58	10014	012748	MOV	#121041, = (SP)
		121041		a contrar a serie de la contrar
59	10020		MOV	#007732,=(SP) /PUSH 2+PI
03	10040	007732	1101	and the second
0			Mov	4040711 - (00)
60	10024	012748	MOV	#040711,=(SP)
		040711		
61	10030	004467	JSR	R4, SPOLSH JENTER POLISH MODE
		011010		
62	10034	0122101	WORD	SDVD /X/2PI
63		0101541	WORD	DUPS13 12 COPIES
64		0074501	WORD	SDINT /INT(X/2PI)
				SSBD /FRACT(X/2PI)
65		0007001	.WORD	
66	10044		WORD	X4813 J4+FRACT(X/2PI)
67		0101541	, WORD	DUPS13 12 COPIES
68	10050	0074501	.WORD	SDINT /INT(4*FRACT(X/2PI))
69	10052	0102151	. WORD	QUDS13 ISAVE INT()
70	10054	0007001	WURD	SSBD IY=FRACT(4*FRACT(X/2PI))
71	10056		WORD	QST\$13 FREDUCE Y TO (-1,1)
72	10000		WORD	DUPS13 12 COPIES
73		010154	WORD	DUPS13 13 COPIES
74	-	0161461	.WORD	SMLD JY+Y
75		0175761	. WORD	SPOPR4 ISAVE Y+Y
76		0103001	, WORD	PLYSI3 IPUSH COEFFICIENTS
- 77	10072		. WORD	SMLD, SADD, SMLD, SADD, SMLD, SADD, SMLD, SADD
	10074	0007041		
	10076	0161461		
	10100	0007041		
	10102			
	10104	0007041		
	10100	0161461		
	10110	0007041		
78	10115	0161461	.WORD	SMLD, SADD, SMLD, SADD, SMLD, SADD, SMLD, SADD
	10114	0007041		
	10116	0161461		
	10120	0007041		
		0161401		
	10124			
	10126	0161461		
	10130	0007641		
79	10102	0161461	. WORD	SMLD 1Y+P(Y+Y)
80	10134		WORD	SPOPRA JPOP HIGH URDER RESULT

SOSNU4 MACRU VR04=14 07=SEP=72 11:43 PAGE 18+

81	10136	0101401	, WORD	RTNS13 Land Marken of States and S
82	10140	005726 RTN\$131	ĩst	(SP) + JPOP QUADRANT FLAG
83	10142	002002	BGE	RT1513 JJUMP IF ARGUMENT WAS +
		062700	ADD	#100000,R0 /SIN(=X)==SIN(X)
* .		100000		
85	10150	012605 RT1513:	MOV	(SP)+,R5
	10152		RTS	R5 JBACK TO CALLER
	Teles	0000000	11.1 4	tin the second of the second
87		016646 DUP5131	MOV	5(SP),=(SP) IDUPLICATE STACK ITEM
00	10104		NO Y	CONTRACT INSCRADULE OTHER TIME
.		000006	M 25 14	C(CA) _(\$8)
89	10160		MOV	6(SP),=(SP)
_		000000	12 m 14	61603 (CD)
90	10104	016646	MQV	6(SP), - (SP)
		000006		
91	10170	016646	MOV	6(SP),=(SP)
		000006	·	
92	10174	000134	JMP	@(R4)+
93		1		
94	10176	005716 X4513:	TST	ØSP JCHECK FOR Ø FRACTION
		001403	BEQ	ZERS13; QUIT NOW
	-	105266	INCO	1(SP) JOUADRUPLE STACK ITEM
		000001		
97	10206	000134	JMP	@(R4)+
	~	012704 ZER\$131		#PR4513, R41 RETURN ZERU RESULT
30	10210		1 G V	and the forth the second measure the forth the second s
00	4 (8 /5 1 4	0101341	JMP	e(R4)+; USE POLISH
	10214	000134	JMF	
100		ARIERE AUNALL	0.00	#SP, 16. (SP) ISAVE QUADRANT NUMBER
101	0510	051666 QUUS131	BIS	COFFED, LARS JOANE WORDNAMI HUMDER
		000020	7.1.175	
102		000134	JMP	@ (R4) +
103		la de la companya de	and the second	
104	6224	105766 QST\$131	TSTB	8, (SP) ITEST QUADRANT
		000010		
105			BEQ	Q13513 JJUMP IF FIRST OR THIRD QUAD
100		001415		and the second
100			ADD	#100000, OSP ; NEGATE STACK ITEM
			ADD	#100000, #SP ;NEGATE STACK ITEM
106	0232	062716 100000	ADD	#100000,0SP ;NEGATE STACK ITEM =(SP)
106	0232	062716 100000 005046	CLR	- (SP)
106 107 103	0232 0236 0240	062716 100000 005046 005046	CLR	= (SP) = (SP)
106 107 103 103	0232 0236 0240 0242	062716 100000 005046 005046 005046	CLR CLR CLR	=(SP) =(SP) =(SP) ;PUSH A FLUATING 1.
106 107 103	0232 0236 0240 0242	052715 100000 005045 005046 005046 005046	CLR	= (SP) = (SP)
106 107 103 109 110	0232 0236 0240 0242 0242 0244	062716 100000 005046 005046 005046 012746 040200	CLR CLR CLR Mov	=(SP) =(SP) =(SP) ; PUSH A FLUATING 1. #40200,=(SP)
106 107 103 109 110	0232 0236 0240 0242 0242 0244	052715 100000 005045 005046 005046 012746 040200 004467	CLR CLR CLR	=(SP) =(SP) =(SP) ;PUSH A FLUATING 1.
106 107 103 109 110 111	0232 0236 0240 0242 0244 0244 0250	052715 100000 005045 005046 015046 012746 040200 004467 011370	CLR CLR CLR Mov JSR	-(SP) -(SP) -(SP) ;PUSH A FLOATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH
106 107 103 109 110 111	0232 0236 0240 0242 0244 0244 0250	052715 100000 005046 005046 012746 040200 004467 011370 000704	CLR CLR CLR Mov	=(SP) =(SP) =(SP) ; PUSH A FLUATING 1. #40200,=(SP)
106 107 108 109 110 111 111	0232 0236 0240 0242 0244 0244 0250 0250 0254	052715 100000 005046 005046 012746 040200 004467 011370 000704 010260	CLR CLR CLR Mov JSR ,word	=(SP) =(SP) =(SP) ; PUSH A FLOATING 1. #40200,=(SP) R4,\$POLSH ;ENTER POLISH \$ADD;QSK\$13 ;X=1.=X
106 107 108 109 110 111 111	0232 0236 0240 0242 0244 0244 0250 0250 0254	052715 100000 005046 005046 012746 040200 004467 011370 000704 010260 012704 QSR\$13;	CLR CLR CLR Mov JSR ,word	-(SP) -(SP) -(SP) ;PUSH A FLOATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH
100 107 103 109 110 111 112 112	0232 0236 0240 0242 0242 0244 0250 0250 0254 0256 0250	052715 100000 005046 005046 012746 040200 004467 011370 000704 010260 012704 010260 012704 010260 012704	CLR CLR CLR Mov JSR .word Mov	=(SP) =(SP) =(SP) ; PUSH A FLUATING 1. #40200,=(SP) R4,\$POLSH ; ENTER POLISH \$ADD;QSK\$13 ;X=1.=X #QSE\$13,R4 ; POINT BACK INTO LIST
106 107 108 109 110 111 111	0232 0236 0240 0242 0242 0244 0250 0250 0254 0256 0250	052715 100000 005046 005046 012746 040200 004467 011370 000704 010260 012704 010260 012704 010260 105255 013\$13;	CLR CLR CLR Mov JSR .word Mov	=(SP) =(SP) =(SP) ; PUSH A FLOATING 1. #40200,=(SP) R4,\$POLSH ;ENTER POLISH \$ADD;QSK\$13 ;X=1.=X
100 107 103 109 110 111 112 113 114	Ø232 Ø236 Ø240 Ø242 Ø242 Ø250 Ø256 Ø264	052715 100000 005046 005046 012746 040200 004467 011370 000704 010260 012704 010260 012704 010260	CLR CLR CLR Mov JSR .word Mov	=(SP) =(SP) =(SP) ; PUSH A FLUATING 1. #40200,=(SP) R4,\$POLSH ; ENTER POLISH \$ADD;QSK\$13 ;X=1.=X #QSE\$13,R4 ; POINT BACK INTO LIST
100 107 103 109 110 111 112 113 114 115	0232 0236 0240 0242 0244 0250 0254 0256 0256 0264	052715 100000 005046 005046 012746 040200 004467 011370 000704 010260 012704 QSR\$13: 010060 105255 Q13\$13: 000011 ;	CLR CLR MOV JSR .WORD MOV ASRB	-(SP) -(SP) ;PUSH A FLOATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X=1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT
100 107 103 109 110 111 112 113 114 115 110	0232 0236 0240 0242 0244 0250 0254 0256 0256 0256 0256 0256	052715 100000 005045 005045 012745 040200 004467 011370 000704 010260 012704 QSR\$13: 010060 105255 Q13\$13: 000011 ; 103002	CLR CLR MOV JSR .WORD MOV ASRB BCC	-(SP) -(SP) ;PUSH A FLOATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X=1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT QUT\$13 ;JUMP IF FIRST OR SECOND
100 107 103 109 110 111 112 113 114 115	0232 0236 0240 0242 0244 0250 0254 0256 0256 0256 0256 0256	052715 100000 005045 005045 012745 040200 004467 011370 000704 010260 012704 QSR\$13: 010060 105255 Q13\$13: 000011 ; 103002	CLR CLR MOV JSR .WORD MOV ASRB	-(SP) -(SP) ;PUSH A FLOATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X=1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT
100 107 103 109 110 111 112 113 114 115 110	0232 0236 0240 0242 0244 0250 0254 0256 0256 0256 0256 0256	052715 100000 005045 005045 012745 040200 004467 011370 000704 010260 012704 QSR\$13: 010060 105255 Q13\$13: 000011 ; 103002	CLR CLR MOV JSR .WORD MOV ASRB BCC	-(SP) -(SP) ;PUSH A FLUATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X=1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT QUT\$13 ;JUMP IF FIRST OR SECOND #100000,*SP ;NEGATE STACK ITEM
100 107 103 109 110 111 112 113 114 115 116 117	Ø232 Ø236 Ø240 Ø240 Ø240 Ø240 Ø250 Ø250 Ø256 Ø264 Ø264 Ø264 Ø270 Ø272	052715 100000 005045 005045 012745 040200 004467 011370 000704 012704 010260 012704 012704 012704 012704 012704 012704 012704 012704 012705 012704 013\$13: 000011 ; 103002 052715 100000	CLR CLR MOV JSR .WORD MOV ASRB BCC ADD	-(SP) -(SP) ;PUSH A FLOATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X=1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT QUT\$13 ;JUMP IF FIRST OR SECOND
100 107 103 109 110 111 112 113 114 115 116 117 118	Ø232 Ø236 Ø240 Ø240 Ø240 Ø250 Ø250 Ø256 Ø264 Ø264 Ø270 Ø272 Ø276	052715 100000 005045 005046 012746 040200 004467 011370 000704 010260 012704 QSR\$13: 010060 106265 Q13\$13: 000011 ; 103002 052716	CLR CLR MOV JSR .WORD MOV ASRB BCC ADD	-(SP) -(SP) ;PUSH A FLUATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X#1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT QUT\$13 ;JUMP IF FIRST OR SECOND #100000,*SP ;NEGATE STACK ITEM @(R4)+
100 107 103 109 110 111 112 113 114 115 116 117 118	Ø232 Ø236 Ø240 Ø240 Ø240 Ø250 Ø250 Ø254 Ø256 Ø264 Ø264 Ø276 Ø276	052715 100000 005045 005045 012745 040200 004467 011370 000704 010260 012704 QSR\$13: 010060 105255 Q13\$13: 000011 ; 103002 052715 100000 000134 QUT\$13: ;	CLR CLR MOV JSR .WORD MOV ASRB BCC ADD JMP	-(SP) -(SP) ;PUSH A FLUATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X=1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT QUT\$13 ;JUMP IF FIRST OR SECOND #100000,*SP ;NEGATE STACK ITEM
100 107 103 109 110 111 112 113 114 115 116 117 118	Ø232 Ø236 Ø240 Ø240 Ø240 Ø250 Ø250 Ø254 Ø256 Ø264 Ø264 Ø276 Ø276	052715 100000 005045 005046 012746 040200 004467 011370 000704 010260 012704 QSR\$13: 010060 105265 Q13\$13: 000011 103002 062716 100000 000134 QUT\$13:	CLR CLR MOV JSR .WORD MOV ASRB BCC ADD JMP	-(SP) -(SP) ;PUSH A FLUATING 1. #40200,-(SP) R4,\$POLSH ;ENTER POLISH \$ADD,QSK\$13 ;X#1.=X #QSE\$13,R4 ;POINT BACK INTO LIST 9.(SP) ;TEST QUADRANT QUT\$13 ;JUMP IF FIRST OR SECOND #100000,@SP ;NEGATE STACK ITEM @(R4)+

(

(

4

		co se rat al	01078E		MIDN	40 Pr 1	NTNE	CONSTANTS
	121	0304	012705		MOV	#9,,K5 1	NTNE	CUNSTANTS
			000011				· · · ·	
	122	0310	000404		BR	PY1513		
			010346	PY25131	MOV	R3,=(SP)		
•	124	0314			MOV	R2,=(SP)		
	125		010146		MOV	R1,=(SP)		JPUSH Y+Y
	144							
	126	0320			MOV	RØ,=(SP)		- DIEL CONCTANT
	127			Py15131	MOV	-(R4),-(S		PUSH CONSTANT
	128	0324	014446		MOV	=(R4),=(S		
	129	0326	014446		MOV	=(R4), =(S	P)	
	130		014446		MOV	- (R4), - (S	P)	
	131		005305		DEC	R5 7	COUNT	COEFFICIENTS
			003366		BGT	PY2513		
					MOV	#XPD\$13,R	A	
	133	0300	012704		HQ V	AVE DATON:		
	. ~ .		010072		7.40	A 2 (3 4 1)		
		0342	000134		JMP	@(R4)+		
	135				.ENDC			
	136			1				
	137				IFOF	FPU		
	138			DCUSI	SETD	1		DOUBLE PRECISION FP
	139				LOD	02(R5),F0	1	GET ARGUMENT
	140				ADDO	P12513,F4		$COS(X) \equiv SIN(X+PI/2)$
					BR	SNC513/		
	141					31403437		BAUDIE DEFATSTAN PD
	142			DSINI	SETD	1		DOUBLE PRECISION FP
	143				LOD	@2(R5),FK	91	GET ARGUMENT
	144			SNCS131	SETI	t de la constante de la constan		SHURT INTEGERS
	145				MOV	#FC0513,8	01	POINTER TO CONSTANTS
	146				CLR	R41		SIGN FLAG: + ARG
	147				CFCC			GET SIGN OF ARG
					BGE	PUSS131		
	148							SIGN FLAG: - ARG
	149				INC	R41		
	150				ABSD	FØJ		REMOVE ARGUMENT SIGN
	151			P05\$13;		(RØ)+,FØI		X/2PI
	152				MODD	#1.0,F0;		FØR FRACT(X/2PI)
	153				CFCC			
	154				BEQ	RTN5131		EXIT ON 0 FRACTION
	155				MODO	#4.0,F01		FØ# FRACT(4+FRACT(X/2PI)
	156				STCDI	F1,R1;		QUADE INT(4+FRACT(X/2PI))
				in the second	ROR	R17		
	157				BCC	Q135137		JUMP IF FIRST OR THIR QUAD
	158							AMOD TL L'UNA MO IDTU AMOUN
	159				NEGD	FØ]		
	160			- 19	ADDD	#1.0,F0;		Y=1.0=X
	161			Q13513:	ROR	R17		
	162				BCC	0125131		JUMP IF FIRST OR 2ND QUAD
	163				NEGD	FUI		Y 🖶 🖛 Y
	164			1				
	165			Q12513:	LDD	F0, F2;		
	166			~~~	MULD	F2,F2;		Z=Y**2
					MOV	#8.,K17		COUNT OF CUNSTANTS FOR POLYNOMIA
	167							INITIALIZE ACCUMULATOR
	168			1 1 .	LDD	(RØ)+,F1/		THALTARTER WARMURAINK
	169			XPD5131	MULD	F2, F1;		
	170				DEC	R11		COUNT
	171				ADDD	(RØ)*,F17	1 1	F11 = Z1F1 + C(I)
	172				BGT	XPD\$131		LOOP
	173			1				
	174				MULD	F1,FØ;		FØ1= Y+F1
	175				TST	R41		TEST SIGN FLAG
	110					····		

				1		
176				BEQ	RTN\$131	
177				NEGD	FØI	SIN(=X) = =SIN(X)
178			RTNS131	STD	F0,=(SP)/	MOVE RESULT TO STACK
179				NON	(SP)+,R01	AND THENCE TO ROR3
180				MOV	(SP)+,R11	
181				MOV	(SP)+,R21	
182			Sec. 1	MOV	(SP)+,R31	
183				RTS	R51	EXIT
184			1			
185			P125131	, WORD	040311,0077321	PI/2
186				WORD	121041,064302;	and the second
187			7			
188			1	ORDER-	DEPENDENT GONSTAN	ITS I I I I I I I I I I I I I I I I I I
189			1			
190			FCUS131	WORD	040711,007732;	2*PI
191				WORD	121041,0043021	
192				ENDC		
193	0344	026716		WORD	026716,106703	1.587061098171E=11
- · · ·		106703				
194		045277		, WORD	045277,146362	
		146362		-		
195			1			
196	0354	130467		.WORD	130407,136273	;=,66843217206396E=9
		136273				
197	0300	103054		WORD	103054,123153	
	0362	123153		2 <u>3 2</u> 2 2 2		
198)			
199	0364	032164		. WORD	032104,074657	1,5692134872719023E=7
		074657				
200		047254		. WORD	047254,154742	
		154742				
201		- · · · ·	1			
202	0374	133561	3 . 8 . 19 .	WORD	133501,101646	;=.3598843007208693E=5
		101646				
203	0400	167216		, WORD	167216,134016	
	0402	134016				
204			1 1			
	0404	035050		, WORD	035050,036032	1,1604411847068221E=3
	0405	036032				
206	0410	041214		WORD	041214,103131	
	0412	103131				
207			1			
	0414	136231		, WORD	136231,064546	;=.4681754135302643E=2
	0416	064546				
209		071423		.WORD	071423,125024	
	0422	125024				
210			1			· 7.60.60.60.61.65.15 4
211				. WORD	037243,032743	1,7969262624616544E=1
	0426					
212	0430			. WORD	035655,051557	
	0432	051557				
213			1	5 ^{- 1}		A JEANAADEREALAA
214	0434			, WORD	140045,056747	1-,6459640975062462
		056747				
215	0440	030455		, WORD	030455,171222	
	0442	171222				
216			1			

66

(

(

4

Ф

217			CONS13:	. WORD	040311,007732	11.570796326794897
218	0450	007732 121041 064302		WORD	121041,004302	
219 220	-		1	.ENOC		

SDS003 MACRO VR04-14 07-SEP-72 11:43 PAGE 19

				.TITLE .IFDF	\$0\$QØ3 CND\$14
			- 7 - 7	DSQRT	V003A
			I 2 CnPYR	IGHT 197	1, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS
			1		·· ···································
				GLOBL	DSQRT, SERR;
,				IFNDF	FPU
				.GLOBL	SAUD, SUVU, SPOLSH;
2				, ENDC	THE DOUBLE PRECISION SQUARE ROOT FUNCTION
			1	SDSQRT Calling	
i i			1	JSR	R5, SDSQRT
;			1	BR	A
			1	#ARG	
			141		
)			7	RETURNS	DSQRT IN RØ - R3.
			1		
		000000		RØSXØ	
		000001		81#%1	
		000002		R2#%2	
		000003		R3=%3	
		000004		R4=%4	
		000005		R5=%5	
		000000		F0=%0	
		000001		F1=%1	
		000002		F2=%2	
		000006		SPE%6 .IFNDF	FPU
	10454	010546	DSQRT:	MOV	R5, - (SP)
		016505	DQAN 1 E	MOV	2(R5),R5 ;GET ARGUMENT ADDRESS
	*****	000002			
	10402			MOV	OR5,R1 JGET HIGH ORDER ARGUMENT
	10464	100467		BMI	ERR\$14 JERROR IF ARGUMENT NEGATIVE
	10405			BEQ	ZERS14 JFAST EXIT IF ZERO
	10470	016502		MOV	2(R5), R2
		000002			
	10474	012746		MOV	#4,=(SP) ;PUSH ITERATION COUNT
		000004		A	
	10500			ASR	R1 FORM INITIAL ESTIMATE
		006002		ROR	R2
	10504	062701		ADD	#20100,R1
	10810	020100		CL D	- (69)
	10510	005048 005046		CLR	=(SP) =(SP) JUSE ONLY MIGH ORDER PARTS FIRST
		010246		MOV	R2,=(SP)
		010146		MOV	R1,=(SP) J'CAUSE ADD AND DIVIDE ARE
		005046		CLR	- (SP) JFASTER THAT WAY
	-	005046		CLR	
		016546		MOV	2(R5), = (SP)
		000002		c	
	10530	011545		MOV	ØR5,=(SP)
	10532	005046		CLR	• (SP)
				CLR	= (SP)
	10535	010245		MOV	R2,=(SP)

•

Ø

6

Ċ

53	10540	010146		MQV	R1,=(SP)
54	10542	004407	LUPS14:	JSR	R4, SPOLSH JENTER POLISH MODE
		011076			
55	10546	0122101	le en la tra	. WORD	SDVD, SADD, UPLS14 / (X/E+E)
	10550	0007041	bi katati i		
		0105541			
56			UPLS141	SUB	#200,0SP 1(X/E+E)/2
••		000200			
57	10560	005366		DEC	8. (SP) /COUNT LOOP
•		000010			
58	10564	001420		BEQ	OUTS14
		016546		MOV	6(R5), - (SP)
0¢	10000	000000		FIG F	
6 13	10570	016546		MOV	4(R5),=(SP)
60	100.5			nu v	
	10570	000004		MOV	2(R5),=(SP) JUSE LOW ORDER PARTS
01	10010	016546		MOV	2(R5),=(SP) JUSE LOW ORDER PARTS
		000002		20	AND - (CM)
		011546		MOV	ers, -(SP) ;TOO FROM NOW ON
63	10604	016646		MOV	14.(SP),=(SP)
		000010			
64	10610	016646		MOV	14, (SP), = (SP)
		000016			
65	10614	016646		MOV	14, (SP), = (SP)
		000016			
66	10520	016646		MOV	14.(SP),=(SP)
		000015			
67	10624	000746		BR	LUPS14 JGD FOR ANOTHER ITERATION
			OUTS14:		(SP)+, RØ JGET RESULT INTO RØ-RJ
		012001	***	MOV	(SP)+,R1
70		012502		MOV	(SP)+,R2
71		012003		MOV	(SP)+,R3
		005726		TST	(SP) + JPOP ITERATION COUNTER
72	10040	000720	RTNS141		(SP)+,R5
				RTS	R5 FRETURN TO CALLER
	100~2	000205	EnBelat		R5, SERR JERROR 4,4
75	10644		ERR5141	VGN	KOTSHERV LEVVAR STW.
		011136		0.0	Postal of a
		000773		BR	RTN\$14
	10652	004		BYTE	4
78	10653	004		BYTE	4
79	44		ZERS141		RØ
80	10656	005001		CLR	R1
81	10600	005002		CLR	R2
82	10662	005003		CLR	R3
83	10654	000765		BR	RTNS14
84			1		
85				. ENDC	
86			1	÷	
87			,	IFOF	FPU
88			DSGRTI	MOV	2(R5),R41 GET ARGUMENT ADDRESS
89			**************************************	MOV	ØR4, R17 GET HIGH ORDER ARGUMEN
				BMI	ERR\$14 JERROR IF ARGUMENT NEGATIVE
90				BEQ	ZERS14 JFAST EXIT IF ZERO
91					
92				MOV	2(R4), R21
				ASR	RI FORM INITIAL ESTIMATE
93	~				
93 94				ROR	R2
93	с. 			ROR Add Clr	R2 #20100,R1 ≠(SP)

69

97		CLR		ONLY HIGH ORDER PARTS FIRST
98		MOV Mov	R2,=(SP) R1,=(SP)	I CAUSE ADD AND DIVIDE ARE
99				ITERATION COUNT
100		MOV	#4,801	
101		SETD	1	DOUBLE PRECISION FP
102		LDD	(SP)+,F01	GET INITIAL ESTIMATE
103		LDD	PR4, F21	GET X
104	1			
105		LOD	FØ, F11	E=EI
106		LDD	F2, F0;	an di 🗙 di selamban di s
107		DIVD	F1, F01	X/E Contraction of the second s
108		ADDO	F1, F0;	X/E+E
109		DEC	RUI	COUNT
110		DIVD	#2.0,F01	E!= (X/E+E)/2
111		BGT	LUPS141	LOOP
112	1			
113		STO	FU,=(SP)1	MOVE RESULT TO STACK
114		MOV	(SP)+, RØ1	
115		MOV	(SP)+,R17	
116		MOV	(SP)+,R21	AND THENCE TO RO
117		MOV	(SP)+,R37	ander allender ander ander som
118		RTS	R51	
119				
120	ERRs141	JSR	R5, SERRI	ERROR 4,4
121		RTS	R51	Hendrich (1997)
122		BYTE	4	
123		BYTE	4	
			RUI	
124		CLR	•	
125		CLR	R17	
126		CLR	R21	
127		CLR	R31	
128		RTS	R51	
129		.ENDC		
130		ENDC		

4

ø

56 10710 005046

57 10712 016504

.TITLE SUTN03 1 . IFOF CND\$15 2 3 1 4 DATAN V003A 2 5 COPYRIGHT 1971, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS 6 1 7 8 DATAN, DATAN21 .GLOBL 9 FPU . IFNDF 10 SADD'SSBD', SMLD, SDVD, SPOLSH, SPOPR41 , GLOBL 11 . ENDC 12 THE FORTRAN DATAN AND DATAN2 FUNCTIONS 13 1 CALLING SEQUENCE FOR DATANE 14 1 R5, DATAN 15 JSR 1 BR 16 1 A ARGUMENT ADDRESS . WORD 17 1 18 1 A I RETURNS ARCTAN (ARG) IN RØ AND R1. 19 1 20 CALLING SEQUENCE FOR DATAN2: 21 22 JSR. R5, DATAN2 23 BR ARGUMENT 1 ADDRESS . WORD 24 .WORD ARGUMENT 2 ADDRESS 25 1AI 26 RETURNS ACRTAN (ARG1/ARG2) IN RØ AND R1. 27 IF ABS(ARG1/ARG2) > 2++24, THE RESULT IS 28 SIGN(ARG1) +PI/2. 29 IF ARG2 60 THE RESULT IS ARCTAN(ARG1/ARG2) + 30 SIGN(ARG1) *PI. 31 32 33 000000 ROSXO 34 R1=%1 000001 35 R2=%2 000002 36 000003 R3#%3 37 R4=%4 000004 R5#%5 38 000005 39 SPE%6 000000 40 000000 10=20 41 000001 F1=%1 42 000002 F2=%2 000003 F3=%3 43 F43%4 44 000004 F5#%5 45 000005 . IFNOF FPU 46 47 10666 010546 DATAN2: MOV R5,-(SP) CLR = (SY) ICLEAR SIGN FLAG 48 10670 005046 ICLEAR DATAN2 BIAS CLR = (SP) 49 10672 005046 50 10674 005046 CLR *(SP) 51 10676 005046 CLR * (SP) CLR -(SP) 52 10700 005046 ICLEAR QUADRANT BIAS CLR . (SP) 53 10702 005046 CLR - (SP) 54 10704 005046 -(SP) CLR 55 10706 005046

CLR

MOV

. (SP)

2(R5),R4

JGET FIRST ARG ADDRESS

\$DTN03 MACRO VR04=14 07=SEP=72 11:43 PAGE 20+

		000002			
58	10716	016446		MOV	6(R4),=(SP)
50	10140	000006		1104	O COMPLETE CALLY
80	10900			MON	A (D A) _ (S B)
59	10/22	016446		MOV	4(R4), -(SP)
		000004			
60	10726	016446		MOV	2(R4),=(SP) ;GET FIRST ARG
		000002			
61	10732	011446		MOV	@R4,=(SP)
		011600		MOV	OSP, RO JARGI TO RO
63		016504		MOV	4(R5),R4 JGET SECUND ARG ADDRESS
00	18140				
		000004		MAN	C(D)) (20)
64	10/42	016446		MOV	6(R4),=(SP)
	· · · · · · · ·	000025			
65	10746	016446	i je ver	MOV	4(R4), = (SP)
		000004		18.4	
66	10752	016446		MOV	2(R4),=(SP) JGET SECOND ARG
		000002			
67	10758	011446		MOV	@R4,=(SP)
68		011601		MOV	
69		001445		BEQ	INF515 JJUMP IF DENOMINATOR IS 0
70	10704	006300		ASL	RØ IGET ABS VAL ARG1
71	10766	105000		CLRB	RØ. / GET: EXPONENT
72	10770	000300		SWAB	RØ
73		006301		ASL	R1
		105001		CLRB	R1 / JGET EXPONENT ARG2
75		000301		SWAB	
					The second se
		160100		SUB	
11	11005	022700	i de la companya de l	CMP	#58, RØ ICHECK MAGNITUDE
		000072			
78	11006	002433		BLT	INF\$15 FTREAT AS INFINITY
79	11010	004467	DIV\$151	JSR	R4, SPOLSH
		010630			
80	11014	012210	1	, WORD	SDVD/UPL\$15 /GET ARG1/ARG2
~0		011020		9 -	
81			UPLS15:	TST	04(R5) /IF ARG2 >0, BIAS #0
01	11040		AL#37A6	191	safual it. weas and pixe an
		000004			
		002022		BGE	ATE\$15 /IF ARG2<0, BIAS#SIGN(ARG1)+PI
83	11025	012766		Mav	#040511,16.(SP) /PI
		040511			
		000020			
84	11034	012766		MOV	#007732,18.(SP)
**	******	007732		र स्थर क	्रिक साह रे आ कृति में आ कृति हैं।
		000022			
85	11042	612766		Mov	#121041,20.(SP)
		121041			
		000024			
86	11050	012766		MOV	#064301,22.(SP)
		064301			n na her men sen men gener men gener i de sen er en son son son son son son son son son so
		000025			
0.77	11082			* *	ADENKA TTECT ANCA
87	11000	005775		TST	e2(R5) ITEST ARG1
		000002		60 - F	
88	11062	002003		BGE	ATES15
89	11064	062766		ADD	#100000,16.(SP) ;=PI
		100000			
		000020			
90	11072	005716	ATES151	TST	ØSP ISET CODES
91	11074	000443	· · · · · · · · · · · · · · · · · · ·	BR	AT1\$15 JJOIN MAIN ROUTINE
~ 1	· · · · · · · · · · · · · · · · · · ·	200 O		¥ ۹	· · · · · · · · · · · · · · · · · · ·
				1 K 1 K	

72

C

(

SDTN03 MACRO VR04=14 07=SEP=72 11:43 PAGE 20+

(

\$

¢

				·	
92	11076	062705		ADD	#36, ISP IFLUSH STACK
93	11102	012700 040311		MOV	#040311,R0 JANS = SIGN(ARG1)+PI/2
94	11106	012701 007732		MQV	#007732,R1
95	11112	012702 121041		MOV	#121041, R2
96	11116	012703 064301		MQV	#064301,R3
97	11122	005775		TST	02(R5) ITEST ARG1
98	11126			BGE	INR\$15 JJUMP IF +PI/2
	11130				
23	1+100	062700 100000		ADD	#100000,RØ ;-PI/2
100 101		000205	1	RTS	R5 JRETURN TO USER
102	1136	010546	DATANE	MOV	R5,=(SP)
103	1140	005046		CLR	- (SP) /CLEAR SIGN FLAG
104				CLR	- (SP) / CLEAR ATAN2 BIAS
105				CLR	R(SP)
100					
107				CLR	= (SP)
				CLR	• (SP)
108		005046		CLR	- (SP) /CLEAR QUADRANT BIAS
109		005046		CLR	-(SP)
110		005046		CLR	• (SP)
111				CLR	- (SP)
112	1102	016504		MOV	2(R5),R4 JGET ARG ADDRESS
		000002			
113	1166	016446		MOV	6(R4),=(SP)
		000006			
114	11/2	016446 000004		MOV	4(R4), = (SP)
115	1176	616446		MOV	2(R4),=(SP) JGET LOW ORDER ARG
		000002			
116	1202	011446		MOV	PR4,=(SP) JGET HIGH URDER
117		002004	AT1515;	BGE	
118	1206			ADD	n na martina a substanti a martina a martina a substanti a substanti a substanti a substanti a substanti a subs
* * ~	****	100000			#100000, SP JGET ABS VALUE
110	1212	005266		INC	OA (SP) JELLC -
***	* 5 * 5	000030		ANV .	24. (SP) JFLAG -
120	1216		PLUS151	CMP	@SP,#40200 ;CHECK IF <1,
121	1222	103455		BLO	LE1815 JJUMP IF <1.
		003011		BGT	GT1515 />1.
		005766		TST	2(SP) JCHECK LOW ORDER
	4699	000002		101	ELOFY FUNER LUN UNDER
124	1232	001006		BNE	GT1\$15
	1234	005765		TST	4 (SP)
44 Ge 🕫	8 MA 17 19	000004			
126	1040			ENE	OTIALS
		001003		BNE	GT1915
12/	1545	005766 000006		TST	6 (SP)
128	1246	001443		BEQ	LE1515 /=1.
			GT15151	Mov	
		140311	******	* i W 4	#140311,8.(SP) ;=PI/2

SDTN03 MACRO VR04=14 07=SEP=72 11:43 PAGE 20+

	•	and the Marian Ca	ha ma s r		T
130	1255	012766	MOV	A007/32,10, (SP)	fATAN(X) = PI/2 = ATAN(1/X)
		007732 000012	9		
1 2 1	1064	012766	MOV	#121041,12.(SP)	
101	1544	121041	NOV .	4161-411-61(4F1	
		000014			
132	1272	012766	MOV	#064301,14. (SP)	
4 U M	* 5. ' 5.	064301			
		000016			
133	1300	605366	DEC	24. (SP) JADJUST	SIGN
-	T - or	000030			$(\mathbf{r}_{i}, \mathbf{r}_{i}) \in \mathbf{r}_{i}$
134	1304	016040	MOV	6(SP),=(SP)	IMOVE ARG DOWN
		000006			
135	1310	016046	MOV	6(SP),=(SP)	
A 104 ET		000006	4 (2) 1	6(CD) - (SO)	
136	1314	016646	MOV	6(SP),=(SP)	
4 72 77	1 20 0	000006	MOV	6(SP),=(SP)	
137	1060	000006		0(0,1)=(0,1)	
138	1324	012766	MOV	#40200,8.(SP)	FINSERT 1.
104	*****	040200			
		600010			
139	1332	005066	CLR	10. (SP)	
		000012			
140	1336	005066	CLR	12, (SP)	
		000014	.		
141	1342	005066	CLR	14, (SP)	
		000016	100		TOMPHER & AV
142	1340	004467 010272	JSR	R4, SPOLSH	JCOMPUTE 1./X
143	1350	0102/2	, WORD	SDVD/LE1\$15	
* 4 6		0113561	b	*****	
144		016046 LE15151	MOV	6(SP),=(SP)	INDVE ARG DOWN
		000000			
145	1362	016040	MOV	6(SP),=(SP)	
•		000006			
146	1306	016646	MOV	6(SP), = (SP)	
		000006			
147	1372	016546	MOV	$6(SP)_{,=}(SP)$	
		000005	e . 0	8. (SP) JINSERT	A 6
140	13/0	005066	CLR	O'TOLI ITNOEKI	A Province and the second s
149	1 4:40	000010 005060	CLR	10. (SP)	
143	1402	000012		***	
150	1406	005066	CLR	12. (SP)	
		000014	**		
151	1412	005066	CLR	14. (SP)	
		000016			
152	1416	021027	CMP	05P, #037611	1 TAN (15)
		037011	.		
153		103507	BLO		F LESS THAN TAN (15)
154	1424	101016	BHI	TNSS15 JJUMP II	
155	1426	026627	CMP	2(SP),#030242	
		000002 030242			
156	1434	101012	BHI	TN5\$15	
157		103501	BLO	L15815	
	· ㅋㅋ //	第 개 1년 77 166 章		সময়, আর করা হয়ন বলা শীর্ষে বিষয়া বিষয় বিষয় বিষয়া বিষয় বিষয়া	

74

(

\$
158	1440	026627		CMP	4(SP), #172366
	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	000004			
		172360			
159	1446	101005		BHI	TNSS15
160		103474		BLO	L15515
161		026627		CMP	6(SP),#065261
101	1445			w Pt P	0(341,400301
		000000			
		065261			· · · · · · · · ·
162		101470		BLOS	L15815
163	1462		TN\$\$151	MOV	#040006,8,(SP) ;INSERT PI/6
		040006			
		000010			
164	1470	012766		MOV	#005221,10.(SP)
		005221			
		000012			
165	1476	012766		MOV	#140553,12,(SP)
		140553			· · · · · · · · · · · · · · · · · · ·
		000014			
166	4504	012766		MOV	#115454,14.(SP)
100	1004	115454		101	HITOHOMPENE COLD
		000010		Né da ki	
		011000		MOV	@SP,RØ JARG TO REGS
168	1514	016601		MOV	2(SP),R1
		000002			
169	1520	016602		MOV	4(SP), R2
		000004			
170	1524	016603		MOV	6(SP),R3
		000000			
171	1530	012740		MOV	#062524, = (SP)
		062524			
172	1534	012740		MOV	#041302, = (SP)
99 ° 117		041302			
173	1540	012746		MOV	#131727,=(SP) ;PUSH =R007 3
4.4	** *	131727		· · • •	areatiest frank that areas areas of
174	1644	012746		MOV	#140335,=(SP)
1/7	* 0 - 4	140335		1101	4140400) ··· (0+)
476	AERA			4 O V	8 4 (op)
175		010346		MOV	R3, = (SP)
		010246		MOV	R2,-(SP)
		010146		MOV	R1,=(SP)
178		010046		MQV	RO,=(SP) /PUSH ARG
179	1500	005046		CLR	= (SP)
180	1502	005046		CLR	- (SP)
181	1504	005040		CLR	-(SP) /PUSH 1.
182	1566	012745		MOV	#40200,=(SP)
-		040200			
183	1572	012746		MOV	#062524, = (SP)
		002524			and the second
184	1576	012/40		MOV	#041302, - (SP)
***	*** V	041302		··· ·· ·	TT Ser TT 曲 TT 短 幅 夏 (1) 新 M FT 更 (2)
	1600	~		Menvi	
185	1602	012745		MOV	#131727,=(SP) ;PUSH ROUT3
	A & 2 -	131727		NA (9) 1/	4040375 - 100)
186	1605	012746		MOV	#040335, = (SP)
		040335		4.5 m 3.1	
187		010346		MOV	R3, = (SP)
188	1614	010240		MOV	R2,=(SP)
189	1616	010146		MOV	R1,-(SP) JPUSH ARG
190	1620	010046		MOV	R0, = (SP)

\$

Ô

75

SDTN03 MACRO VR04=14 07=SEP=72 11:45 PAGE 20+

191	1622	004467	JSR	R4, SPOLSH ; TRANSFORM ARG
		010016		
192		1		(RODT3+X=1)/(ROOT3 +X)
193	1626	0161461	, WORD	\$MLD, \$\$BU, UP\$15, \$\$BD, \$DVD, L15\$15
	1630	0007001		
	1632	0117701		
	1634			
	1636	0122101		
	1640			
104	4	011000 L15515:	MOV	OSP, RØ JGET ARG
		· · · · · · · · · · · · · · · · · · ·		2(SP),R1
195	1044	016601	MOV	elopi, ni
		000002	1. ch 1. f	4.6651.05
196	1650		MOV	4(SP), R2
		000004	31.0.17	
197	1654	016603	MOV	6(SP), R3
		000000		
198	1660	010340	MOV	R3, = (SP)
199	1602	010246	MOV	R2,=(SP)
200	1604	010148	MOV	R1,=(SP) /GET THREE COPIES
201	1606	010046	MOV	R0, = (SP)
		010340	MOV	R3, = (SP)
		010246	MOV	R2,-(SP)
	1674	010146	MOV	R1,=(SP)
		010046	MOV	R0,-(SP)
200		004467	JSR	R4, SPOLSH
200	1100		8 a n	
007	4 m (A +	007740		SMLD JGET ARG*+2
207		016146	. WORD	
208		0175761	. WORD	SPOPRA, PLYS15 JSET UP COEFFICIENTS
		0120201		
209		0161461XPU\$15:	.WORD	SMLD, SADD, SMLD, SADD, SMLD, SADD
	1714	0007041		
	1716	0161461		
	1720	0007041	a the second	a Alger and the second
	1722	0161401		
	1724	0007041		
210	1726	0161461	. WORD	SMLD, SADD, SMLD, SADD, SMLD, SADD
		0007041		
		0161461		
	· ·	0007041		
		0161461		$(1, 2, 2, 3) = \frac{1}{2} \left[\frac{1}{$
		0007041		
044			ធិត្រភា	SMLD, SADD, SMLD, SADD, SMLD, SADD
< 1 1		0161461	, WORD	aurniaurataurniaurniaurn
	1744	and the second sec		
		0161461		
		0007041		
	1752	0161461		
	1754	0007041		
212	1756	0607041	.WORD	SADD /P(X)+0 IF X<=1, P(X)=PI/2 IF X>
213	1760	0120641	WORD	SGN\$15 JADJUST SIGN
214		0007041	WORD	SADD JADD ATAN2 BIAS
215		0175761	WORD	SPOPRA JPOP RESULT TO REGS
216		0117701	WORD	EX1\$15
217	1770	005726 EXIS151	TST	(SP) + JPOP SIGN FLAG
218		012605	MOV	(SP)+,R5
219		000203	RTS	R5 IRETURN TO USER
220	477 M	8 8 - 9 - 9 - 9 - 9 - 9 - 9 - 9 - 9 - 9 -	(1 1 V	and to many the second and the secon
	1776	012666 UP\$15:	MOV	(SP)+,22. (SP) IMOVE STACK ITEM UP
221	1110	MISACO MEGIDI	es Q. ¥	PALL ACTION ANAL DINCH TINH OL

76

(

\$

		000026			(0014 00 (00)		
222	2002	012066		MOV	$(SP) + 22 \cdot (SP)$		
		000026			100.4 00 100.		
223	2006	012666		MOV	(SP) + 22 + (SP)		
		000026		· · · · · · · · · · · · · · · · · · ·			
224	2012	012666		MOV	(SP)+,22*(SP)		
	216	000020	San				
225	2016	000134		JMP	@(R4)+		
226			1	ter an br			
227	2020			MOV	#CON\$15+8,,R4	POINT TO COEFFICIENT TABLE	
		0122101	1	· · · ·		and the state of t	
228	2024	012705		MOV	#9., R5 JGET #	OF CONSTANTS	
		000011					
229	2030	000404		BR	PY1515		
230	2032	- CH	PY25151		R3,=(SP)		
231	2034	010246		MOV	$R2_{i} = (SP)$	and a second	
232	2036	010146		MOV	R1,-(SP)	PUSH ARG	
233	2040	010046		MOV	RØ,=(SP)	a ana tiana tia ana kiana %6° a kiana	
234	2042		Py1515;	MOV	= (R4), = (SP)	; PUSH CONSTANT	
235	2044	014446		MOV	=(R4),=(SP)		
236	2046	014446		MOV	= (R4), = (SP)		
237	2050	014446		MOV	= (R4), = (SP)		
238	2022	005305		DEC	R5 ICOUNT		
239	2054	003366		BGT	PY2\$15		
240	2056	012704		MOV	#XPD515,R4		
		0117121	l Alexandre				
241	2002	000134		JMP	@(R4)+		
242			1				
243	2064	005766	SGNS151	TST	16, (SP) JCHECK	SIGN FLAG	
		000020					
244	2070	001402		BEQ	SG1\$15		
245	2072	062716		ADD	#100000, •SP	INEGATE RESULT FOR (-1,0) &	(1,1
		100000	х. ¹¹ г				
246	2076	000134	SG1\$151	JMP	@(R4)+		
247				.ENDC			
248			1				
249				. IFDF	FPU		
250			DATAN2:	SETO		SET OP MODE FOR FPU	
251				MOV .	2(R5),R31	ADDRESS OF ARG1	
252				MOV	4(R5),R41	ADDRESS OF ARG2	
253		-	*	MOV	PR3, KOJ	HIGH URDER ARG1	1. 1
254				MOV	0R4, R11	HIGH ORDER ARG2	
255				BEQ	INF\$157	JUMP IF DENOMINATOR Ø	
256				ASL	RØJ		
257				CLRB	R01		
258				SNAB	R01	EXPONENT OF ARG1	
259				ASL	R11		
260			tant ti	CLRB	R17		
261			e filler og som er s	SWAB	R17	EXPONENT OF ARG2	
262				SUB	R1, R01	GET EXPONENT DIFFERENCE	
263		•, * ·	·	CMP	#58, / RØ1	CHECK MAGNITUDE	
264				BLT	INF\$151	TREAT AS INFINITE	
265				LOD	P1\$15,F37	INITIALIZE BIAS=PI	1.14
266				LOD	@R3,F01	GET ARG1	
267				CFCC			*e. 5
268				BGE	A1P\$157	JUMP IF ARG1>0	1.8.3
269				NEGO	F31	BIAS=SIGN(ARG1)+PI	

SDTN03 MACRO VR04=14 07=SEP=72 11:43 PAGE 20+

278A1PSIDI LDU $GR4,F11$ GET ARG2271CLR0F31IF ARG2AB, BIAS=6273CLR0F31IF ARG2AB, BIAS=6274A2MSIDI DUF1,F01ARG1/ARG2, SET FLOAT CC275BRATISIDIJOIN MAIN MOUTINE276FFAMAG1/ARG2, SET FLOAT CC277INFSIDI LDUF12SIS,F11RESULT=SIGN(ARG1)+P1/2278BREXISIDIFEST ARG1279BCEEXISIDIFEST ARG1280MEGDF11-P1/2281BREXISIDI282JBR283CLR0F31284CLR0F31285LDU#2(P3),F01286CLR0F31287CFCCJ288CLR0F31289CLR0F31			•			
271CFCC811A2MS151273CLRO $+3J$ 274A2MS151DIF ARG2+0, BIAS=0275BRAT1S151276BRAT1S151277INFS151LOD278PISTPR31279BREXIS151279BGEEXIS151279BGEEXIS151279BGEEXIS151279BGEF31270CLROF31271CLROF31272CLROF31273CLEAR ATANE BIAS274CLRO275CRO276F31277CLAR278CLAR279BR271F31271CLAR272CLAR272BR273CLAR274CLAR275CLAR276CLAR277CLAR277CLAR278CLAR279CLAR279CLAR270CLAR271CLAR272CLAR273CLAR274CLAR275CLAR276CLAR277CLAR278CLAR279F31279CLAR279CLAR270CLAR271CLAR271CLAR272CLAR273CLAR274CL		•				
272BLTA2MS151273CLROF31IFARG2×80, BIAS×80274A2MS151DIVOF1,F01ARG2×80, BIAS×80275BRAT1S151JOIN MAIN ROUTINE276INF\$151LDDPI2S13,F11RESULT=SIGN(ARG1)*PI/2277INF\$151DDPI2S13,F11RESULT=SIGN(ARG1)*PI/2278BGEEXIS151+PI/2279BGEEXIS151+PI/2280NEGDF11RESULT=SIGN(ARG1)*PI/2281JSETDSET DP282JATANISETDJ283LDDF2(RD),F07GET ARGUMENT284CLCOF31CLEAN285LDDF2(RD),F07GET ARGUMENT286AT1S151CLFF31CLEAN287CLCOJGET SIGN DF AAG288CLROF31CLEAN299CLNOF31CLEAN291ASDF01ASS(X)292LUSI51JUPP IF293PLUSI51LDD294CLFOF34295CLFOF34296BEELEIS157297GT1S151DEC298CLFOF34299LDDF1,F01291JONANT BIAS292LDDF1,F01293LDDF1,F01294CLFO395CLFO396DIVO297GT1S151DEC298DI			A1P\$15:		@R4, F1;	GET ARG2
273CLR0F33IF ARG2>0, BIA380274A2MS151DIVOF1, F43ARG2>62, BIA380275BRAT1S153JOIN MAIN ROUTINE276IF1S1EDDPI2S15, F13RESULT=SIGN(ARG1)*PI/2278BGEEX1S157TEST ARG1279BGEEX1S157TEST ARG1280NEDDF1 $P1/2$ 281BREX1S157TEST ARG1282JSET DP MODE FOR FPU283CLROF37CLEAR ATAN2 B1AS284CLROF37CLEAR ATAN2 B1AS285LDD#2(A7),F03GET ARGUMENT286AT1S157CLROF33CLEAR BIGN FLAG287CLROF33CLEAR GUADAANT B1AS288STDF3,F03F5847AN2 B1AS299CLROF33CLEAR GUADAANT B1AS294BGEPLUS157JUMP IF GUADRANT 1 OR 3295BLELEIS157CMCC R44296BLELEIS157STGN FLAG297GT1S151CLROF3,F44F440ADAANT B1AS298DIVOF4,F411,974299LDDF15,F37GUADRANT B1AS293LDDF16,F04ATAN(X)=F1/2=ATAN(157)294CLROF3,F44F349,40295GT1S151STDF3,F44296GEEL155157X=* TAN(157)397LDDF16,F04ATAN(X)=F1/2=ATAN(157)398LDDF46,F14F349,40						
274 275 276A2MS150 DTVO $F1_1F0_1$ ARE1/ARG2, SET FLOAT CC 301N MAIN ROUTINE276 277INFS151 LDDPI2S15,F11 TST 4731RESULT=SIGN(ARG1)+PI/2 TEST ARG1278 279BGE BGE BGE BGE 284FIJ 284FIJ 284284 285 286JATANI SETD CLRD FJSET DP MODE FOR FPU CLEAR ATAN2 DIAS CLEAR SIGN FLAG 286 287 286286 287 286 287 286ATISISI CLR CLCR FJR41 286 CLRD FJ286 287 288 289 289 289 289 289 289 280 289 280 289 280 280 289 280 280 280 289 280<						
P75BRATISISJOIN MAIN MOUTINE276INFSIST LDDPI2SIS,F11RESULTESIGN(ARG1)+PI/2278TSTPR31PI2SIS,F11RESULTESIGN(ARG1)+PI/2279BGEEXISIS;+PI/2280NEGDF1;-PI/2281JSET DPI2SIS;282JCLROF3;CLEAR ATAN2 BIAS283DATANISET D;SET DP MODE FOR FPU284CLROF3;CLEAR ATAN2 BIAS285LDDP2(R0),F03;GET ARGUMENT286ATISIS;CLROF3;287CLROF3;CLEAR AGUMENT288STDP3,F03;CLEAR AGUMENT289CLROF3;CLEAR AGUMENT289CLROF3;CLEAR AGUMENT280CLROF4;FLAG281ABSOF0;ABSO282LUSIS;JUMP IF GUADRANT 1 OR 3283PLUSIS;JUMP IF GUADRANT 1 OR 3294CRPOF6;F1;CHECK IF X41;0295CTFCCSLECLEISIS;296BLECLSIS;ATAN(X)=P1/2-ATAN(1/X)297GT1SIS;F3;F4;CLMPART BIAS298CLMPOF3;F4;CAPACATAR299CLMPOF3;53;F364,0299CLNOF3;F4;F40UADRANT BIAS293CLNOF3;F4;F40UADRANT BIAS294CLNOF3;5;F4;COMPARE TAN(15) I X395CCCGGELISSIS						IF ARG2>0, BIAS=0
2761 INFSIDIPI2SID,F11RESULT=SIGN(ARG1)+P1/2279BGEEXISID+P12279BGEEXISID+P12280BGEEXISID+P12281BREXISID+P12282JJSET D J283DATANI SETD JCLEAR ATAN2 BIAS284CLRO FSJCLEAR ATAN2 BIAS285LDD#2(RD),F0J286GET ARGUMENT287CFCC JGET ARGUMENT288STD FJ,FDJFSatAR2 BIAS289CLRO FSJCLEAR GUADRANT J OR 3289ABSD FBJABSOK FBJ283PLUSIDI LDDH1.0F11284GFCCSTD285GTSD FJ,FDJFLAG286CLRO FSJCLEAR GUADRANT J OR 3289ABSD FBJABSOK FBJ293PLUSIDI LDDH1.0F111.0294GFCCR4JX>1.0, ADJUST SIGN FLAG295GTSDD FJ,FGJGTSDJS,FJGUADRANT BIAS296GTSDJS,FJGUADRANT BIAS297GTISDJS CCR4JX>1.0, ADJUST SIGN FLAG298DIVOFJ,F4J1.07X299LDDFJ,F4JTGAX299CDDFJ,F4JF3=0,0299CKPCSGE299CDDFG,F4J393CLEAS FJ,F4JF3=0,0394CHPOFG,F4J395GDDFG,F4J396GDDRT,F4J397LDDFG,F4J<	274		A2M5151	DIVD		ARG1/ARG2, SET FLOAT CC
277INFSISI LODPI2SIS,F17RESULTESIGN(ARG1)*P1/2278BGEEXISIS;+P1/2279BGEEXISIS;+P1/2280BGEEXISIS;+P1/2281BREXISIS;+P1/2282JSETD JSET DP MODE FOR FPU283DATANISETD JCLEAR ATAN2 BIAS284LDD\$2(RD),FUJGLEAR ATAN2 BIAS285LDD\$2(RD),FUJGET SIGN OF ARGUMENT286ATISISI QLRR4CLEAR SIGN FLAG287CFCCJUP IF GUADRANT BIAS289CLROF3,FDJ284ABSOF0,FDJ285CLROF3,FDJ286CLROF3,FDJ287CLROF3,FDJ288FUUSISIJUP IF GUADRANT 1 OR 3291ABSOF0,F1J283PLUSISIDD294CRPOF0,F1J295CFCC296BLE296GTISISI DEC297GTISISI DEC298DIVO299LDD291JUNC293CLRO294CLRO295CCC296BLE297GTISISI STD298DIVO299CLRO299CLRO291ADD291CLCA293CLRO294CLRO295CCC296BLE297GTISISI STD298	275			BR	AT15151	JOIN MAIN ROUTINE
279 BGE EXIST F87 FEGT ARGI 279 BGE EXISTS +P1/2 280 BR EXISTS +P1/2 281 BR EXISTS +P1/2 282 J CRO F31 CLEAR ATAN2 BIAS 284 CLRO F31 CLEAR ATAN2 BIAS 285 LDD P2(RD),F01 GET ARGUMENT 286 ATISTST (R 41 CLEAR GUADNANT 1 OR 3 287 STO F3,FD1 F58ATAN2 BIAS 288 SCRO F31 CLEAR GUADNANT 1 OR 3 289 CLRO F31 CLEAR GUADNANT 1 OR 3 291 ASSO F01 ABS(X) 292 INC F41 FLAC 293 PLUSIST JUMP 1F GUADRANT 1 OR 3 294 CLPO F0,F11 1.04 295 CFCC R41 X>1.0, ADJUST SIGN FLAG 296 GTISIST, F07 GUADRANT BIAS F36.0 297 GTISIST, F37 GUADRANT BIAS F36.0 298 DIVO F3.F41 SATAN(276		1			
275 BGE $EXISIS;$ $+PI/2$ 286NEGOFI; $+PI/2$ 281 BR EXISIS; $+PI/2$ 282J $CLRO$ FJ; $CLEAR ATAN2 BIAS$ 283 $DATANI$ SETOJ $CLEAR ATAN2 BIAS$ 284 CRO FJ; $CLEAR ATAN2 BIAS$ 285 LDD $eQ(RO),FU;$ GET ARGUMENT286ATISIS; CLR R4; $CLEAR OLDENT287GFC;JGET SIGN OF ARGUMENT288STDFJ;FD;FS=ATAN2 BIAS290BGEPLUSIS;JUMP IF GUADRANT BIAS291ABSFD;ABS(X)292INCR4;FLAG =293PLUSID;LDDH:0,F1;294CFCCGISID;DC295CFCCGUADRANT BIAS296BELLEISIS;297GTISID;DCR4,F1;298LDDFJ;F0;ATAN(X)=P1/2=ATAN(1/X)299LDDFJ;F0;ATAN(X)=P1/2=ATAN(1/X)201JLODP12SIS,F3;GUADRANT BIAS203CLNDFJ;F0;ATAN(X)=P1/2=ATAN(1/X)204CLNDFJ;F1;F0;ATAN(X)=P1/2=ATAN(1/X)205CFCCGET SISSIS,F3;F3=000206GEELISSIS;F3;F3=000207LDDF3,F4;CUADRANT BIAS208LDDF3,F4;CANCOT3=1,0209MULOR73SIS,F4;GUADRANT BIAS201DDDF$	277		INF \$151		PI2\$15,F11	· · · · · · · · · · · · · · · · · · ·
279BGEEXISIS $+P1/2$ 280NEGDFJJ $+P1/2$ 281BREXISISJ $+P1/2$ 282JCLRDFJJ $+P1/2$ 283DATANISETDJSET DP MODE FOR FPU284CLRDFJJCLEAR ATAN2 BIAS285ATISISI CLRR44CLEAR SIGN FLAG286ATISISI CLRR44CLEAR GUADRANT BIAS287STDFJ,FDJFSATAN2 BIAS289CLRDFJJCLEAR GUADRANT FIAS290BGEPLUSISJJUMP IF GUADRANT 1 OR 3291ABSOFGJABS(X)292INCR44FLAG =293PLUSIDI LOD#1.0/F111.0/294CHPDF0.F111.0/295CFCCINC296GTISISI DECR44297GTISISI DECR44298LDDF1.F01AAN(X)=P1/2-ATAN(1/X)299LDDF1.F01AAN(X)=P1/2-ATAN(1/X)201JLDDF3.F41F3=0.0202CLRDF3.F41F3=0.0203LDDF1.F01AAN(X)=P1/2-ATAN(1/X)204LDDF3.F51GTAN(15) I X205CFCCGTISISI, F513GUADRANT BIAS206CFCCGETS15.F14207LDDF1.F01X+00T3-1.0208LDDF0.F11F3.P1/6209MULOR7.S15.F11S4.M(15) I X201SUBOH4.0/F3.F11	278				eR31	
DREXISIS282j283DATANI SETO284CLRO285LOD286ATISIST CLR877CCC j288STD289CLRO290BGE291STD292F3,F07293CLRO294CLRO295CLRO296ATISIST CLR297RATANE298STD299CLRO293PLUSIST294CLRO295CLRO296GTISIST DEC297GTISIST DEC298CLRO299LDD291JUNP F294CHCC295CCC296BLE297CTISIST DEC298CLNO299LDD299LDD291STD299CLNO296CLNO297GTISIST DEC298CLNO299CLNO299CLNO299CLNO299CLNO299TISISTSP172291STD299CLNO291STD299CLNO290TISISTSP172291STD292F3293CLNO294CLNO295SUBO296CLNO297STST298CLNO298CLNO29				BGE	EXIS157	+P1/2
BREXISIS282J283DATANISETD284LCR0FJ285LDD $e2(R3),F00$ 286ATISISI CLR287CFCC288STD289CLRO289CLRO280BGE280STD281ABSO282FJ,FD7283FJ284STD284STD285CLRO286STD287CLEAR QUADRANT BIAS289BGE280BGE281ABSO282INC283PLUSISI294CMPO295CFCC296BLE297GTISIDI CEC298CLRO297GTISIDI CEC298CLDO298LDD299LDD299LDD291J301J302LEISISI303CLRO298CDO299LDD299LDD299LDD291STD301J302LEISISI303CLRO304CHPO305CFCC306CHPO307LDD308CLRO309NUD309NUD301J301J302LEISISI303CHPO3	280			NEGD	F11	=P1/2
282 j DATANI SETD j SET DP MODE FOR FPU 283 LDD #2(RD),FØ; GET ARGUMENT 285 LDD #2(RD),FØ; GET ARGUMENT 286 ATISISI CLR R4; CLEAR SIGN FLAG 286 CFCC j GET SIGN OF ARGUMENT 288 CLRO F3; CLEAR GUADRANT BIAS 289 CLRO F3; CLEAR GUADRANT 1 OR 3 289 ABSD FØ; ABSD 290 LNC R4; FLAG - 293 PLUSID; LDD #1.0+F1; 1.0 294 CMPD FØ;F1; 1.0 ABSD 295 GTISID; DEC R4; X>1.0, ADJUST SIGN FLAG 296 BLE LEISID; CMPD FJAG 297 GTISID; DEC R4; X>1.0, ADJUST SIGN FLAG 298 DIVD F0,F1; 1.0/X GUADRANT BIAS 298 DIVD F3,F4; F4=GUADRANT BIAS 390 CHPO TISISIS;F0; GUADRANT BIAS 391 ADDD F15SIS;F0				BR	EX1\$157	
283 DATANI SETO / CLEW ATAN2 BIAS 284 LOD \$2(R\$),F\$; CLEW ATAN2 BIAS 285 LOD \$2(R\$),F\$; GET ARGUMENT 286 ATISIS; CLR R4; CLEAR SIGN FLAG 287 GCT SIGN OF ARGUMENT STO F\$,F\$; 288 STO F\$,F\$; F\$=ATAN2 BIAS 289 CLRO F\$; JUMP IF GUADRANT BIAS 289 STO F\$,F\$; JUMP IF GUADRANT 1 OR 3 290 BGE PLUSIS; JUMP IF GUADRANT 1 OR 3 291 ABSO F\$; JUMP IF GUADRANT 1 OR 3 292 PLUSID; LOD #1.0,F1; 1.0 293 PLUSID; LOD #1.0,F1; 1.0 294 GT1SID; DEC R4; X>1.0, ADJUST SIGN FLAG 295 GT1SID; DEC R4; X>1.0, ADJUST SIGN FLAG 296 GLE ESIS; X>1.0, ADJUST SIGN FLAG 297 GT1SID; DEC R4; 298 DIDO F1,F0; 1.0/X 299 LOD F0,F1; GUA			1			
284CLROF35CLEARCLEAR ATAB 21AS285LDD $\varphi_2(RO), F01$ GET ARGUMENT286AT15151CLRR44CLEAR SIGN FLAG287CFCCJGET SIGN OF ARGUMENT288STDF3, F31CLEAR GUADRANT 81AS299CLROF33CLEAR GUADRANT 1 OR 3291AB30F01AB5(X)292INCR41FLAG293PLUS151LDD#1,0,F11294CMPDF0,F11CMECK IF X4=1,0295CFCC296BLELEIS151297GT1S151DECR41298LDDF1,F011,0'A299LDDF1,F011,0'A299LDDF1,F011,0'A299LDDF1,F01GAXANT BIAS290CHNOF0,F111,0'A293LDDF1,F01GAXANT BIAS294CMPOF3,F41F44QUADRANT BIAS295CFCCGAXANT BIAS296CHNOF3,F41F44QUADRANT BIAS297GT1S151STDF3,F41F349,0298LDDF1,F61COMPARE TAN(15) I X299LDDF1,F61F349,0291ABCCHNOF359,763293CLNOF351,57,63F379,76394CLNOF351,57,63F379,76395LDDF0,F11K+R00T3-1,0/(X+R00T3)316LDDF0,F21X317M			DATANI	SETD	n 👔 👘 ser	SET DP MODE FOR FPU
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$				CLRD	F31	CLEAR ATAN2 BIAS
286ATISISI CLRR41CLEAR SIGN FLAG287CFCCJGET SIGN FLAG288STD $F3, F01$ FS#ATAN2 BLAS289CLRO $F37$ CLEAR QUADRANT BLAS290BGEPLUSISJJUMP IF QUADRANT 1 OR 3291ABSOF01ABS(X)292INCR41FLAG293PLUSISI LDD#1.0*F111.0294CHPDF0.F11CHECK IF X<#1.0				LOD	02(R5),F0;	GET ARGUMENT
237 CFCC j GET SIGN OF ARGUMENT 288 STD F3,FDJ F5*ATAN2 BLAS 289 CLRO F3; CLEAR GUADHANT BLAS 280 BGE PLUSI5; JUMP IF GUADRANT 1 OR 3 281 ASSO FØ; ASS(X) 282 INC R4; FLAG 283 PLUSIDI LDD #1.0+F1; L4C 284 GHPD F0;F1; CHECK IF X<=1.0			AT15151	CLR	R41	CLEAR SIGN FLAG
286STD $\vec{F}3, \vec{F}5$ $\vec{F}5\pi \pi An 2 BIAS$ 289 $CLR0$ $\vec{F}3$ $CLEAR GUADRANT BIAS$ 280 BGE $PLUS157$ $JUMP$ IF 292 INC $R47$ $FLAG$ 293 $PLUS151$ LDD $H1.0F11$ 1.0 294 $CMP0$ $F0.F11$ $I.0$ 295 $CFCC$ 296 BLE $LE15157$ 297 $GT1S151$ DEC $R47$ $Z98$ $DIV0$ $F0.F11$ $I.07X$ DDV $F0.F17$ $I.07X$ $PU2815.F37$ 298 $DIV0$ $F0.F17$ $I.07X$ 299 LDD $F1.F07$ $ATAN(X) = PI/2 = ATAN(1/X)$ 298 $DIV0$ $F0.F17$ $I.07X$ 299 LDD $F1.F07$ $TAN(X) = PI/2 = ATAN(1/X)$ 298 $DIV0$ $F0.F17$ $I.07X$ 299 LDD $F1.F07$ $TAN(X) = PI/2 = ATAN(1/X)$ 298 $DIV0$ $F0.F17$ $I.07X$ 309 LDD $F1.F07$ $TAN(X) = PI/2 = ATAN(1/X)$ 300 CRO $TS15.F47$ $COMPARE TAN(15)$ 301 f $F0.F27$ $X = TAN(15)$ 303 $CLRO$ $F3.F47$ $F4=0UADRANT BIAS$ 304 CRO $TS15.F47$ $COMPARE TAN(15)$ 305 $CLRO$ $F3.F47$ $F3=0.0$ 306 $CFCC$ $S1.5.F47$ $TAN(15)$ 307 LDD $F0.F21$ $X = TAN(15)$ 318 LDD $F0.F21$ $X = R00T3=1.0$ 319 $ADDO$;	GET SIGN OF ARGUMENT
260 $CLR0$ F3i $CLEAR GUADRANT BIAS$ 201ASDF0iABS(X)202INCR4;FLAG =203PLUSIDILDDH1.0+F1i204CMPOF0;F1iCHECK205CFCC206BLELEIS15j207GTISIDIDEC208DIVOF0;F1i1.0/X209LDDF1;F0iATAN(X)=P1/2=ATAN(1/X)208DIVOF0;F1i1.0/X209LDDF1;F0iATAN(X)=P1/2=ATAN(1/X)208DIVOF3;F4iF4=QUADRANT BIAS209LDDF1;F0iATAN(X)=P1/2=ATAN(1/X)208DIVOF3;F4iF4=QUADRANT BIAS209LDDP12\$15;F3iWuADRANT BIAS300CHPOTISS15;F0iGUADRANT BIAS301JSameCHPO302LEISISISTDF3;F4i303CHPOTISS15;F0iCMPARE TAN(15) i X304CMPOTISS15;F0iX=R00T3=1.0305CFCCSameSame306BCELISS15;F0iX=R00T3=1.0317MULOF0;F0iX=R00T3=1.0318LDDF0;F2iX319XPUSISIMULOF0;F2iX310MULOF0;F1iF1:F1:X*2C(I)313JMULOF0;F1iF1:F1:X*2314L15S15ILDDF0;F1iF1:F1:X*2315MULOF0;F1iF1:F1:X*2C(I)<					F3,F51	FSEATAN2 BIAS
290 6GE PLUSI5; JUMP IF GUADRANT 1 OR 3 291 A850 F0; AB5(X) 292 INC R4; FLAG 293 PLUSID: LDD #1.0FI; 1.0 294 CMPD F0.F1; CHECK IF X<=1.0				CLRO	F31	CLEAR QUADRANT BIAS
291 ABS0 FØ; ABS(X) 292 INC R4; $FLAG = -$ 293 PLUSIDI LDD #1.0,FI; 1.0 294 CMPD FØ,FI; CHECK IF X<=1.0						JUMP IF QUADRANT 1 OR 3
292 INC $R4i$ $FLAG =$ 293 $PLUSIDI<$ LOD $w1.0,F1i$ 1.0 294 $CFCC$ BLE $LEIS15i$ 295 $GFCC$ BLE $LEIS15i$ 296 BLE $LEIS15i$ $CHECK$ IF $X < m1.0$ 297 $GTISIDI$ DEC $R4i$ $X > 1.0, ADJUST$ 298 $DIVD$ $F0,F1i$ $1.0/X$ 299 $DIVD$ $F0,F1i$ $1.0/X$ 299 LDD $F1,F0i$ $ATAN(X) = P1/2 = ATAN(1/X)$ 300 LDD $F1,F0i$ $ATAN(X) = P1/2 = ATAN(1/X)$ 301 i i GVX 302 $LEIS15i$ STD $F3,F4i$ $F4 = QUADRANT BIAS$ $GXAV$ GVX 303 $CHPD$ $TISS15,F0i$ 304 $CHPD$ $TISS15,F0i$ 305 GEE $LISS15i$ 306 GEE $LISS15i$ 307 LDD $P06iS15,F3i$ 308 LDD $F0,F1i$ 309 $MULD$ $RT3S15,F0i$ 310 $SUBD$ $\pi1.0F0i$ 311 $ADDD$ $RT3S15,F1i$ 312 $DIVD$ $F1,F0i$ 313 i 314 $L1SS15i$ LDD $F0,F2i$ X 315 $MULD$ $F0,F2i$ MV $W+F00,F0,F0i$ 314 $L1SS15i$ LDD DDD $R0,F0,F1i$ $COUNT$ OF $CDEFFICIENTS$ 314 LDD $F0,F1i$ $COUNT$ 315 MOV $\#6,Nii$ <td></td> <td></td> <td></td> <td>ABSD</td> <td>FØ;</td> <td>ABS(X)</td>				ABSD	FØ;	ABS(X)
293 $PLUSIDI LDD$ $41,0,F11$ $1,0$ 294 $CMPD$ $P0,P11$ $CMECK IF X <= 1,0$ 295 $CFCC$ 296 B_LE $EIS151$ 297 $GT1SIDI DEC$ $R41$ $X > 1,0, ADJUST SIGN FLAG$ 298 $DIVO$ $F0,F11$ $1,0/X$ 299 LDD $F1,F01$ $ATAN(X) = PI/2 = ATAN(1/X)$ 300 LDD $F1,F01$ $ATAN(X) = PI/2 = ATAN(1/X)$ 300 LDD $F1,F01$ $ATAN(X) = PI/2 = ATAN(1/X)$ 301 J $GUADRANT BIAS = PI/2$ 302 $LEIS151 STD$ $F3,F41$ $F4 = QUADRANT BIAS$ 303 $CLRO$ $F31$ $F3 = 0,0$ 304 $CMPO$ $T1SS15,F401$ $COMPARE TAN(15) I X$ 305 $CFCC$ $S06$ BEE $L15S157$ 306 BEE $L15S157$ $X <= TAN(153)$ 307 LDD $F0,F11$ $BA0,F01$ 308 LDD $F0,F11$ 309 $MULD$ $RT3S15,F401$ 310 $SUBD$ $H1,0,F01$ 311 $ADDD$ $RT3S15,F401$ 312 $OIVD$ $F1,F00$ 313 I 314 $L15S151$ LDD $F0,F21$ X 315 $MULD$ $F0,F11$ 316 MOV $\#0,F01,F11$ 317 MOV $\#0,F11$ 318 DDD $R00,F1,F11$ 319 $XPUS151$ MUD $F0,F11$ 320 DEC $R11,COPF11$ 321						FLAG -
294 CHPO F0,F1; CHECK IF X<=1.0 295 GFLS GFLS GFLS 296 BLE LEIS15; 297 GT1S15; DEC R4; X>1.0, ADJUST SIGN FLAG 298 DIVO F0,F1; 1.0/X 299 LDD F1;F0; ATAN(X)=P1/2=ATAN(1/X) 300 LDD P12S15;F3; GUADRANT BIAS=P1/2 301 j GUADRANT BIAS F3=0.0 301 j GUADRANT BIAS F3=0.0 302 LE1S1S; STD F3;F4; F4=QUADRANT BIAS 303 CLRD F3; F3=0.0 304 CMPO T15315;F0; COMPARE TAN(15) ; X 305 CFCC S0 F3=F1; 306 DEE L15515; X F3=P1/6 305 CFCC S0 F3=F1; 306 DEE L15515; X F3=F1; 307 LDD P16515; F0; F3=F1; 308 MDUD RT3515;F0; X=TAN(15) 310 SUBO #1,0;F0; X=ROOT3=1,0			PLUS151			
295 UFCC 296 BLE LE1s157 297 GT1S151 DEC R41 X>1.0, ADJUST SIGN FLAG 298 DIVD F0,F1; 1.0/X 299 LDD F1,F0; ATAN(X)=PI/2=ATAN(1/X) 300 LDD P12515,F3; GuadDRANT BIAS=FI/2 301 j GuadDRANT BIAS GuadDRANT BIAS 302 LE1S15; STD F3,F4; F4=QUADRANT BIAS 303 CLR0 F3; F3=60.0 GuadDRANT BIAS 304 CLR0 F3; F3=60.0 GuadDRANT BIAS 305 CFCC GeE L15S15; X<= TAN(15) i X				CMPD	F0,F1;	CHECK IF X<#1,0
296 BLE LE1s15;297GT1S1D: DECR4;X>1.0, ADJUST SIGN FLAG298DIVOF0,F1;1.07X299LDDF1,F0;ATAN(X)=PI/2=ATAN(1/X)300DDP12S15,F3;GUADRANT BIAS*PI/2301jGUADRANT BIAS302LE1S15;STDF3,F4;F4=QUADRANT BIAS303CLR0F3;F3=0.0304CUROT15s15,F0;COMPARE TAN(15); X305CFCCGGEL15s15;X<= TAN(15); X				CFCC		
297GT1\$15: DECR4; $X>1,0, ADJUST SIGN FLAG298DIVDF0,F1;1.07X299LDDF1,F0;ATAN(X)=P1/2=ATAN(1/X)300UDDP12S15,F3;GUADRANT BIAS=P1/2301;GUADRANT BIAS302LE1S15: STDF3,F4;F4=QUADRANT BIAS303CLRDF3;F3#0.0304CMPOT15s15,F0;COMPARE TAN(15) i X305CFCC306BGEL15S15;X<= TAN(15)$				BLE	LE15157	
298 DIVD F0,F1; 1.0/X 299 LDD F1,F0; ATAN(X)=P1/2=ATAN(1/X) 300 LDD P12515,F3; GUADRANT BIAS=P1/2 301 j GUADRANT BIAS GUADRANT BIAS 303 CHPD F3;F4; F4#GUADRANT BIAS 304 CHPD F3;F4; F4#GUADRANT BIAS 305 CLRD F3; F3#0,0 306 BGE L15515; X<= TAN(15) i X			GT15151		R41	X>1.0, ADJUST SIGN FLAG
299LDD $F1,F0;$ LDD $ATAN(X) = PI/2 = ATAN(1/X)$ $UDD PI2S15,F3;$ $ATAN(X) = PI/2 = ATAN(1/X)$ $UADRANT BIAS = PI/2301jjUDDPI2S15,F3;UADRANT BIAS = PI/2302LE1S15; STDF3,F4;F4 = QUADRANT BIASS03GMPOT15315,F0;COMPARE TAN(15)304CMPOT15315,F0;COMPARE TAN(15) ; X305CFCC306BGEL15S15;X <= TAN(15)307LDDF0,F1;308LDDF0,F1;309MULDRT3S15,F0;310SUBD*1.0,F0;311ADDDRT3S15,F1;312DIVOF1,F0;MULDRT3S15,F0;313j314L15S15; LDDF0,F2;MOV#6,F0;X + *2MOV#6,F0;X + *2316MOV#6,F1;MOV#0,F1;COUNT319XPDS15; MULDF0,F1;320ADDD(R0)+,F1;321ADDD(R0)+,F1;322BGTXPDS15; LOOP323MULDF2,F1;PI/6UR 0,0324ADDDF3,F1;PI/6UR 0,0325SUBDF4,F1;$						
300 LDD PI2S15,F3; GUADRANT BIAS=PI/2 301 j 302 LEIS15; STD F3,F4; F4=QUADRANT BIAS 303 CLRD F3; F3=0.0 304 CLRD F3; F3=0.0 304 CMPO T15515;F0; COMPARE TAN(15);X 305 CFCC Sample Sample 306 BGE L15515;F0; X<= TAN(15);X						
301 j 302 LE1S151 STD F3,F4; F4=QUADRANT BIAS 303 CLR0 F3; F3#0.0 304 CMPO T15S15,F0; COMPARE TAN(15) ; X 305 GE L15S15; X<= TAN(15)						
302 LE1S15: STD F3,F4; F4=QUADRANT BIAS 303 CLR0 F3; F3=6.0 304 CMP0 T15315,F0; COMPARE TAN(15) ; X 305 CFCC S06 BGE L15S15; X = TAN(15); X 306 BGE L15S15; X = TAN(15); F3=PI/6 307 LOD PI6s15,F4; F3=PI/6 308 LOD F0,F1; S1 309 MULD RT3s15,F0; X=R00T3=1.0 311 ADDD RT3s15,F1; X=R00T3 312 DIVO F1,F0; (X=R00T3=1.0)/(X=R00T3) 313 J X=X=X 314 L15S15; LDD F0,F2; X 315 MULD F0,F0; X=*2 316 MOV #FC0515,R0; PDINTER TO POLYNOMIAL CONSTANTS 317 MOV #8.,R1; COUNT OF COEFFICIENTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPUS15; MULD F0,F1; 320 DEC R1; COUNT				सन्द अस् वस	· · · · · · · · · · · · · · · · · · ·	
303 CLRD F3; F3#0.0 304 CMPO T15\$15,F0; COMPARE TAN(15) ; X 305 CFCC S06 SEE L15\$15; X<# TAN(15)			LF15151	STD	F3.F41	F4=QUADRANT BIAS
304 CMPO T15315,F0; COMPARE TAN(15) i X 305 CFCC 306 BGE L15515; X<# TAN(15)						
305 CFCC 306 BGE L15\$15; X<= TAN(15)						
306 BGE L15\$15; X<= TAN(15)						
307 LDD PI6815,F3; F3#PI/6 308 LDD F0,F1; 309 MULD RT3\$15,F0; 310 SUB0 #1.0,F0; X+ROOT3=1.0 311 ADDD RT3\$15,F1; X+ROOT3=1.0 311 ADDD RT3\$15,F1; X+ROOT3=1.0)/(X+ROOT3) 312 DIVD F1,F0; (X*ROOT3=1.0)/(X+ROOT3) 313 J X X+ROOT3=1.0)/(X+ROOT3) 314 L15\$15; LDD F0,F2; X 315 MULD F0,F0; X+*2 316 MULD F0,F1; COUNT OF CUEFFICIENTS 317 MOV #FCO\$15,R0; POINTER TO POLYNOMIAL CONSTANTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; OUNT 320 DEC R1; COUNT 321 ADDD (R0)+,F1; F1;# F1* X*+2 + C(I) 322 BGT XPD\$15; LOOP F1;# F1* X 324 ADDD F2,F1; F1;# F1*X 325 SUBD			a ser a s		1155151	X . TAN (15)
308 LDD F0,F1; 309 MULD RT3\$15,F0; 310 SUBD #1.0,F0; X+R00T3=1.0 311 ADDD RT3\$15,F1; X+R00T3 312 DIVD F1,F0; (X+R00T3=1.0)/(X+R00T3) 313 J X X+R00T3=1.0)/(X+R00T3) 313 J X+R00T3=1.0)/(X+R00T3) 314 L15\$15; LDD F0,F2; X 315 MULD F0,F2; X 316 MULD F0,F2; X 317 MULD F0,F2; X 318 L15\$15; LDD F0,F1; COUNT OF COEFFICIENTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; COUNT 320 DEC R1; COUNT 321 ADDD (R0)+,F1; F1;= F1* X*+2 + C(I) 322 BGT XPD\$15; LOOP MULD 323 MULD F2,F1; F1;= F1*X 324 ADD0 F3,F1; P1/6 UR 0.0 325 SUB0						
309 MULD RT3\$15,F0; 310 SUBD #1.0,F0; X+R00T3=1.0 311 ADDD RT3\$15,F1; X+R00T3 312 DIVD F1,F0; (X+R00T3=1.0)/(X+R00T3) 313 J 314 L15\$15; LDD F0,F2; X 315 MULD F0,F2; X 316 MULD F0,F2; X 317 MULD F0,F2; X 318 LDD F0,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; TITALIZE ACCUMULATOR 320 DEC R1; COUNT 321 ADDD (R0)+,F1; F1;# F1* X**2 + C(I) 322 BGT XPD\$15; LOOP 323 MULD F2,F1; F1*# F1*X 324 ADDD F3,F1; P1/6 UR 0.0 325 SUBD F4,F1; P(X)=GUAD BIAS						
310 SUBD #1.0,F0; X+RODT3=1.0 311 ADDD RT3\$15,F1; X+ROUT3 312 DIVD F1,F0; (X+ROUT3=1.0)/(X+ROOT3) 313 j (X+ROUT3=1.0)/(X+ROOT3) 313 j (X+ROUT3=1.0)/(X+ROOT3) 313 j (X+ROUT3=1.0)/(X+ROOT3) 313 j (X+ROUT3=1.0)/(X+ROOT3) 314 L15\$15; LDD F0,F2; X 315 MULD F0,F0; X+*2 316 MULD F0,F0; X+*2 316 MOV #6.R1; COUNT OF COEFFICIENTS 317 MOV #8.R1; COUNT OF COEFFICIENTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPUS15; MULD F0,F1; 321 DEC R1; COUNT 322 BGT XPDS15; LOOP 323 MULD F2,F1; F1;= F1+X 324 ADDD F3,F1; P1/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD BIAS						
311 ADDD RT3\$15,F1; X+R00T3 312 DIVD F1,F0; (X*R00T3=1.0)/(X+R00T3) 313 ; 314 L15\$15; LDD F0,F2; X 315 MULD F0,F0; X**2 316 MOV #FC0\$15,R0; PDINTER TO POLYNOMIAL CONSTANTS 317 MOV #8.,R1; COUNT OF CUEFFICIENTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; 320 DEC R1; COUNT 321 ADDD (R0)+,F1; F1;# F1* X**2 + C(I) 322 BGT XPD\$15; LOOP F1;# F1* X**2 + C(I) 323 MULD F2,F1; F1*# F1*X 324 ADDD F3,F1; P1/6 UR 0.0 325 SUBD F4,F1; P(X)=GUAD						X+R00T3+1.0
312 DIVO F1,F0; (X*RODT3=1.0)/(X+ROOT3) 313 ; 314 L15\$15; LDD F0,F2; X 315 MULD F0,F0; X+*2 316 MOV #FC0\$15,R0; PDINTER TO POLYNOMIAL CONSTANTS 317 MOV #FC0\$15,R0; POINTER TO POLYNOMIAL CONSTANTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; COUNT 320 DEC R1; COUNT 321 ADDD (R0)+,F1; F1;= F1*X**2 + C(I) 322 BGT XPD\$15; LOOP 323 MULD F2,F1; F1:= F1*X 324 ADDD F3,F1; PI/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD BIAS						
313 1 314 L15\$15; LDD F0,F2; X 315 MULD F0,F0; X**2 316 MOV #FCO\$15,R0; PDINTER TO POLYNOMIAL CONSTANTS 317 MOV #FCO\$15,R0; PDINTER TO POLYNOMIAL CONSTANTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; INITIALIZE ACCUMULATOR 319 XPU\$15; MULD F0,F1; COUNT 320 DEC R1; COUNT 321 ADDD (R0)+,F1; F1;# F1* X**2 + C(I) 322 BGT XPD\$15; LOOP 323 MULD F2,F1; F1;# F1* X**2 + C(I) 324 ADDD F3,F1; PI/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD BIAS						
314 L15\$15: LDD F0,F2; X 315 MULD F0,F0; X+*2 316 MOV #FC0\$15,R0; PDINTER TO POLYNOMIAL CONSTANTS 317 MOV #8.,R1; COUNT OF CUEFFICIENTS 318 LDD (R0)+,F1; INITIALIZE ACCUMULATOR 319 XPU\$15: MULD F0,F1; 320 DEC R1; COUNT 321 ADDD (R0)+,F1; F1:= F1*X**2 + C(I) 322 BGT XPD\$15; LOOP 323 MULD F2,F1; F1:= F1*X 324 ADDD F3,F1; PI/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD BIAS				****		a service and the service of the ser
315MULD $F0,F0;$ $X**2$ 316MOV#FCO\$15,R0;PDINTER TO POLYNOMIAL CONSTANTS317MOV#8.,R1;COUNT OF CUEFFICIENTS318LDD(R0)+,F1;INITIALIZE ACCUMULATOR319XPUS15:MULDF0,F1;320DECR1;COUNT321ADDD(R0)+,F1;F1:= F1*X**2 + C(I)322BGTXPDS15; LOOP323MULDF2,F1;F1:= F1*X324ADDDF3,F1;PI/6 UR 0.0325SUBDF4,F1;P(X)=QUAD BIAS			1158151	LDD	FØ.F21	i 🗙 sector de la construcción de la constr
316MOV#FCO\$15,R0;PDINTER TO POLYNOMIAL CONSTANTS317MOV#8.,R1;COUNT OF COEFFICIENTS318LDD(R0)+,F1;INITIALIZE ACCUMULATOR319XPU\$15;MULDF0,F1;320DECR1;COUNT321ADDD(R0)+,F1;F1;# F1* X**2 + C(I)322BGTXPD\$15; LOOP323MULDF2,F1;F1;# F1*X324ADDDF3,F1;PI/6 UR 0.0325SUBDF4,F1;P(X)=QUAD BIAS			#1-#4+			
317MOV $#8., R17$ COUNT OF CUEFFICIENTS 318 LDD $(R0)+,F17$ INITIALIZE ACCUMULATOR 319 XPDS151 MULDF0,F17INITIALIZE ACCUMULATOR 320 DECR17COUNT 321 ADDD $(R0)+,F17$ F17# F1* X**2 + C(I) 322 BGTXPDS157 LOOP 323 MULDF2,F17F17# F1*X 324 ADDDF3,F17PI/6 UR 0.0 325 SUBDF4,F17P(X)=QUAD BIAS						
318LDD $(R0)+,F1i$ INITIALIZEACCUMULATOR319XPD\$15:MULDF0,F1iINITIALIZEACCUMULATOR320DECR1iCOUNT321ADDD $(R0)+,F1i$ F1:= F1* X**2 + C(I)322BGTXPD\$15; LOOP323MULDF2,F1iF1:= F1*X324ADDDF3,F1iPI/6 UR 0.0325SUBDF4,F1iP(X)=QUAD						
319XPU\$15: MULOF0,F1; DECCOUNT 320 DECR1/COUNT 321 ADDD(R0)+,F1;F1:= F1* X**2 + C(I) 322 BGTXPD\$15; LOOP 323 MULDF2,F1;F1:= F1*X 324 ADDDF3,F1;PI/6 UR 0.0 325 SUBDF4,F1;P(X)=QUAD BIAS						
320 DEC R1/ COUNT 321 ADDD (R0)+,F1i F1i= F1*X**2 + C(I) 322 BGT XPD\$15; LOOP 323 MULD F2,F1i F1i= F1*X 324 ADDD F3,F1i PI/6 UR 0.0 325 SUBD F4,F1i P(X)=QUAD			XPUS151			स्थान्य 1 स्थित्स किन्द्र - गिता किन्द्र राज्य कार्य स
321 ADDD (R0)+,F1; F1:= F1* X**2 + C(I) 322 BGT XPD\$15; LOOP 323 MULD F2,F1; F1:= F1*X 324 ADDD F3,F1; PI/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD						COUNT
322 BGT XPD\$15; LOOP 323 MULD F2,F1; F1;= F1*X 324 ADDD F3,F1; PI/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD						
323 MULD F2,F1; F1*= F1*X 324 ADDD F3,F1; PI/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD						이 비 史 영 편 · 비 속 전 · 바이 전 편이가 이 편 함께 문 · · · · · · · · · · · · · · · · · ·
324 ADDD F3,F1; PI/6 UR 0.0 325 SUBD F4,F1; P(X)=QUAD BIAS			n ter e			Fit= Fi+X
325 SUBD F4, F1; P(X)=QUAD BIAS						
050 101 E41 (FOI OTAM LEVO						
	250			131	n n n n n a star a s	The state of the second s

78

*

\$

ō

327 328				BEQ NEGD	SG1\$157 F17	NO ADJUSTMENT NEGATE RESULT FOR (-1,0)&(1,INF)
329 330			SG15151	ADUD	F5,F1;	ATAN2 BIAS
331 332 333 334			ExI\$151	STD Mov Mov Mov	F1,=(SP); (SP)+,R0; (SP)+,R1; (SP)+,R2;	MOVE RESULT TO STACK AND THEN TO REGISTERS
335 336				MOV	(SP)+,R31 R51	EXIT
337			Harrister			
338 339 340) -		/ P1\$151	.WORD	040511,007732; 121041,064301;	PI
341 342 343	2		/ PI2\$151	.WORD .WORD	040311,007732; 121041,004301;	PI/2
344 345 346	5 5		1 T15\$151	.WORD	037611,030242; 172366,065261;	TAN (15)
347 348 349))		/ PI0\$15:	.WORD	040006,005221; 140553,115454;	PI/6
350 351 352) RT3\$15:	WORD	040335,131727; 041302,082524;	
353 354	2100		FCUS15:	.ENDC .word	037005,150707	1.0443895157187
	2106	150707 162300 163030		, WORD	162300,163930	
356 357	2110	137204	1	, WORD	137204,143233	1=,06483193510303
	2114 2116	004010	_	.WORD	004010,000413	
359 360	2120	037235 043002	,	, WORD	037235,043002	1.0767936896066
361	2124	027154 142446		, WORD	027154,142446	
362	2		1		1 1 M . 7	
363		137272 025671		.WORD	137272,025671	30909037114191074
364	2134	116412 065630		.WORD	116412,065630	
365			1	60 0 0 0		1.11111097898051048
	2142	037343 107047		.WORD	037343,147047	1.111103/030001040
367		023625 025401		.WORD	023625,025401	
368 369	2150	137422		.WORD	137422,044444	114285714102825545
370	2134	071335 116151		. WORD	071335,116151	
371		झ ल ज ज भ के	1			

79

SUTNO3 MACRO VR04=14 07=SEP=72 11:43 PAGE 20+

372	2160	037514	i i i i i i i i i i i i i i i i i i i	WORD	037514,146314	1,19999999998729448
· · · · ·		146314		s Killi mak da	a a straight a second straight	
373		146224		. WORD	146224,165650	
~2 ~7 A	5100	165650	•			
374		1 A A A A A A A A A A A A A A A A A A A		1996 - 1 996 - 19		
375	2170	137652		.WORD	137652,125252	1=,333333333333389930
		125252				
376		125252		. WORD	125252,113602	
	2176	113002				
377			1			
378	2200	040200	CON\$151	, WORD	040200,000000	1 99999999999999999
	2202	000000				
379	2204	000000		, WORD	000000,000000	
	2206	000000				
380			1			
381				. ENDC		

80

*

4

4

ò

(

			and the set of the first	and the figure filters.		
			TITLE	SDVD05		
			.IFDF	CNUS16		
		•	.GLOBL	SDVD, SEI		14 Ph Ph - 24 Ph 1 (Ph Ph 1) Ph
		<i>i</i> 1	20VD	INE DU	JOPE NIA	IDE ROUTINE
		į	SDVD	V005A		
		, ,				ITAL EQUIPMENT CORP., MAYNARD,
		1		IN THE PO		
0		1				COND ITEM ON THE STACK
1		1		DENOMIN		
2		1		STACK IN		PUTS IT ON TOP
3 4	14 12 10 12 10 12	1	RØ\$%Ø	STACK IN	INCTO L	
4 5	000000 000001		R1=%1			
5	000002		R28%2			
7	0000003		R3#%3			
8	000004		R4=%4			
9	000000		R5=%5			
8	000000		SP=%6			
1	000007		PC=%7		•	
2	000000		FØ=%0			
3	000001		F1=%1			
4	000010		0=8.			
5	000020		N=16.			
6	000020		Q=16.			
7			. IFDF	FPU		
8		SDVDS	, WORD	170011	11SETD	
9			.WORD	172526	11LDD	(SP)+,F1 JGET DIVISUR
0			.WORD	172426	1160	(SP)+,FØ ;GET DIVIDEND
1			. WORD	174401	11DIVD	F1,FØ /GET QUOTIENT
2			.WORD	174046	11STD	FØ,=(SP) JTO STACK
3			JMP	@(R4)+		
4			.ENOC			
5		di un bit ni 🖷	IFNDE	FPU		
6 12210	010445	SDVDI	MOV	R4,=(SP)		
7 12212	010545		MOV	R5,=(SP)		
8 12214	005000		CLR	RØ		
9 12216 Ø 1222Ø	005001		CLR	R1 R2		
0 12220	005002		CLR	R3		
1 12222			CLR			
3 12226			ASL	N+Ø=2(SF	»)	SHIFT NUMERATOR
. ¹ 8280	000010		177 - 1973		₹.	time mant ti stanskradijeta taniš2
1 10030	006110		ROL	ØSP	IGET NU	MERATOR SIGN
M 16242			CLR	= (SP)	,	
	005046		··· •			
5 12234			TST	USPJI		CHECK FOR 0.0 DENOMINATOR
5 12234	005766		TST	U(SP);		CHECK FOR 0.0 DENOMINATOR
5 12234 6 12236	005765 000010			•		
5 12234 6 12236 7 12242	005766 000010 001521		BEQ	DCHS16;	. ØSP	CHECK FOR 0.0 DENOMINATOR JUMP TO ERROR EXIT JGET NUMERATOR EXPONENT
5 12234 6 12236 7 12242	005766 000010 001521 156516			•	, @SP	JUMP TO ERROR EXIT
5 12234 6 12236 7 12242 8 12244	005765 000010 001521 156516 000021		BEQ BISB	DCH516; N+1(5P),		JUMP TO ERROR EXIT
5 12234 6 12236 7 12242 8 12244 9 12250	005766 000010 001521 156516 000021 001525		BEQ	DCH\$16; N+1(5P), ZER\$16	IJUMP I	JUMP TO ERROR EXIT JGET NUMERATOR EXPONENT
5 12234 6 12236 7 12242 8 12244 9 12250	005765 000010 001521 156516 000021 001526 156600		BEQ BISB BEQ	DCH516; N+1(5P),	IJUMP I	JUMP TO ERROR EXIT JGET NUMERATOR EXPONENT
5 12234 6 12236 7 12242 8 12244 9 12250 9 12252	005765 000010 001521 156516 000021 001526 156600 000020		BEQ BISB BEQ BISB	DCH\$16; N+1(5P), ZER\$16	IJUMP I	JUMP TO ERROR EXIT JGET NUMERATOR EXPONENT
12232 5 12234 6 12236 7 12242 8 12244 9 12250 0 12252 1 12256 2 12260	005765 000010 001521 156016 000021 001526 156000 000020 000020		BEQ BISB BEQ	DCH\$16; N+1(SP); ZER\$16 N(SP);R0	JUMP I	JUMP TO ERROR EXIT JGET NUMERATOR EXPONENT F NUMERATOR IS ZERO

SDVD05 MACRD VR04=14 07=SEP=72 11:43 PAGE 21+

54	12264	156600	BISB	N+3(5P),	RØ			
		000023	0000	NLOISPA				
55	122/0	156601	BISB	N+2(SP),	N 1			
		000055	44 . A . M					
56	12274		SWAB	R1				
57	12275	156001	BISB	N+5 (SP)	, K1			
		000025						
58	12302	156692	BISB	N+4 (SP),	, R2			
		000024						
59	12306	000302	SWAB	R2				
60	12310	156002	BISB	N+7 (SP)	, R2			
	an e e e	000027						
61	12314	156503	BISB	N+6 (SP)	, R3			
÷.	THE PLANE	000026					1997	
60	12320		SWAB	R3				
63	12322		ASL	D(SP)	ISHIFT	DENUMINATUR		
00	LEGEE	000010		~ ~ ~				
64	12326		ADC	2(SP)	IGET DE	ESULT SIGN		
04	LEDED			2 (01)				
66		000002	(° 1 43	D A				
	12332		CLR	R4	د ب ن	GET DIVISOR	EXPONEN	Ŧ
66	12334	156004	BISB	D+1(SP)	174	IGEL DIATOR	CAPUNEN	!
		000011	A		*******	LOF ENDONONES		
67		160416	SUB	R4, eSP		ACT EXPONENTS		
68	12342		SWAB	D(SP)	ILEFT .	JUSTIFY DENOMIN	AIUR	
		000010			an an 1 an 17 an 18			
69		000261	SEC		TINSER	T NORMAL BIT		
70	12350	006066	KOR	D(SP)				
		000010						
71	12354	116665	MOVB	D+3(SP),	D(SP)			
		000013						
		000010						
72	12362	116066	MOVB	0+2(SP)	.U+3(SP))		
	****	000012						
		000013						
73	12370		MOVE	D+5(5P)	D+2(SP))		
/ 4	1	000015	· • • •					
		000012						
7 4	10176		MOVB	D+4(SP),	DASISP	1 • • • •		
/4	16010	116668	ngya	Und (er)	I A TO LON			
		000014						
18 .07		000015	NS (15 17 13	N+7/501	0+4650			
15	12404	116666	MOVE	0+7(SP),	074(07.			Set a set
		000017						
		000014			··· · · · · · · · · · · · · · · · · ·			
76	12412	116960	MOVB	D+6(SP),	U+7 (SP)			
		000016						
		000017						
77	12420	105666	CLRB	D+6(SP)				
		000010						
78	12424	005065	CLR	Q(SP)	ICLEAR	QUOTIENT		
		000020	· · · ·				÷ '''	
79	12430	005066	CLR	Q+2(SP)				
		000022	ter an					$\mathcal{D}_{i}^{(1)} = \mathcal{D}_{i}^{(1)}$
80	12434		ÇLR	Q+4 (SP)				
	,	000024		· · · ·				
81	12440	020060	CMP	RU, D(SP)		JCUMPARE HIGH	I NUM AI	ND DEN
		000010						- Art. 196
82	12444	101042	BHI	DL w316	JUMP 1	IF DENOMINATOR	LŨW	
¥ 5.	* *** *** ** ***	• • • + ··· ·· ··	war tit pa	يتتريد روي , د شغيه يعني	5 mm 147 1		water	

82

⇒

(

(

 $\left(\right)$

¥

.

.

83 12446 103445 BLO DHISIS JUMP IF DENOMINATOR HIGH 84 12430 20166 CMP HIDTSIS JUMPARE LUW ORDER PARTS 608612 BHI DLMSIS BUD DHISIS JUMPARE LUW ORDER PARTS 608614 BHI DLMSIS BUD DHISIS JUMPARE LUW ORDER PARTS 608614 BHI DLMSIS BUD DHISIS JUMPARE LUW ORDER PARTS 608614 BHI DLMSIS BUD DHISIS JUMPARE LUW ORDER PARTS 608614 BHI DLMSIS BUD DHISIS JUMPARE LUW ORDER PARTS 612464 12463 BUD DHISIS JUMPARE LUW ORDER PARTS DMRSIS 612464 12470 DUMPARE BUMP EXPONENT DMRSIS DMRSIS DMRSIS 612542 004643 BH ECISIS PLTSIS PLTSIS DMRSIS DMRSIS DMRSIS DMRSIS DMRSIS DMRSIS DMSSIS						
34 12459 223166 CHP R1,0+2(SP) JCOMPARE LOW DRDER PARTS 85 12464 10136 HI DLMSI6 85 12464 10136 HI DLMSI6 87 12469 22266 CMP R2,0+4(SP) 88 12464 10132 HI DLMSI6 99 12474 10132 HI DLMSI6 99 12474 10132 BH DLMSI6 91 12474 10120 HI DLMSI6 92 12474 10120 BH DLMSI6 93 12502 00804 CLR R4 95 12504 024635 BR ECISI6 94 12522 004635 BR ECISI6 94 12524 026403 BR ECISI6 94 12522 004635 BR ECISI6 94 12524 026403 BR ECISI6 95 12526 025465 CLR FLTSI6 96 12524 02526	83 (2446	103446		81.0	DHIS16 JUMP IF DENOMINATOR HIGH
000012 00014 00000 00014 00000 00000 00014 00000 00000 00014 00000 00000 00014 00000 00000 00014 00000 00000 000150 00000 000000 000150 000000 000000 000150 000000 000000 011447 000000 000000 011447 000000 000000 011447 000000 0000000 011476 0000000 0000000 011476 0000000 0000000 011476 00000000 0000000 011476 000000000 00000000 011252 00000000000000000000000 000000000000000000000000000000000000						
85 12454 101356 BHI DLMS16 85 12400 122266 CMP H2,0+4(SP) 88 12464 10132 BHI DLMS16 89 12474 10132 BHI DLMS16 80 12474 10122 BHI DLMS16 91 12474 10126 BHI DLMS16 92 12476 02152 BNE DLMS16 93 12600 02516 MQ PLATA 94 12520 026516 MQ PLATA 95 12630 026405 BH DLMS16 95 12630 026405 BH FLTS16 95 12630 02700 DCMS16 MQ PLA05,RW 91 12474 010516 MQ PLA05,RW FERROR 5,S 91 1250 0264053 BH ECIS16 S 91 1252 0264053 BH ECIS16 S 91 1252 0264053 CHR RS S	-,,	9 me - 1 - 1 m-				
B7 12450 022266 CMP P2,0+4(SP) 88 12454 13132 BHI DLws16 99 12470 020466 CMP R3,0+6(SP) 98 12474 10122 BHI DLws16 91 12474 10250 BHI DLws16 91 12474 10425 BHI DLws16 91 1252 026465 SH FIS 92 1252 024635 BR EC1S16 94 12404 024045 BR EC1S16 95 1252 024050 CLR 4405,RØ JEKROR 5,S 106 2526 024050 CLR GP7+,SP1+ JFLUSH EXP AND SIGN 106 2506<	85 1	2454		,	BHI	
000014 00015 86 12470 12470 12470 12470 91 12474 101025 DH DH315 91 12474 101025 DH DH315 91 12474 101025 DH DH316 91 12474 101025 DH DLH316 92 12500 00016 INC #SP 94 12502 000465 DH DLH316 94 12502 000465 DH PHONENT 94 12504 000465 DH PHONENT 94 12504 00465 DH PHONENT 94 12504 00463 DH ECISI6 95 12520 005746 ECISI6 TST 91 12520 005746 ECISI6 TST 102 2520 000576 CLH QCH4057 103 2566 CLP QCH4157 PC 103 2566 CLH QCH424(SP) Q00014 104 2562 00007 DH RTNS16 107 2552 00007 <td>86</td> <td>2456</td> <td>103442</td> <td></td> <td>BLO</td> <td>DHI\$16</td>	86	2456	103442		BLO	DHI\$16
88 12464 101032 BHI DLwsi6 90 12470 20366 CMP R3,0*6(SP) 90 12470 101225 BHI DLwsi6 91 12474 101225 BHI DLwsi6 92 12476 00105 BHI DLwsi6 93 12500 005024 CR R4 94 12522 005004 CR R4 95 12500 012700 DCHSi61 MOV #1403,R0 JEKRUK 3,3 97 12512 00403 BR ELIS16 JEKRUK 3,3 98 12520 00576 ELS161 JEKRUK 3,80 JEKRUK 3,3 91 12520 00576 ELS161 JER CS3 101 2526 02740 UNUS161 MOV #4025,R0 JEKRUK 3,3 102 2520 00576 ELS161 SE CS4 OS100 102 2520 00576 CLR Q+2=4(SP) O007270 OS001 103 2530 005065 CLR <td>87 5</td> <td>2460</td> <td>020266</td> <td></td> <td>CMP</td> <td>R2,0+4(SP)</td>	87 5	2460	020266		CMP	R2,0+4(SP)
B9 12405 103436 BLD DHISI6 90 12474 161020 BHI DLWSI6 91 12500 085216 INC #SP 91 12502 08504 CLR R4 95 12504 004403 BR FLTSI6 96 12504 004403 BR ECISI6 97 12512 2044367 ECLSI61 TST 91 12620 044367 BR ECISI6 91 12620 044365 CLR (SP)+(SP)+ JFLUSH EXPAND SIGN 101 2252 04456 CLR QF2-4(SP) 040401 102 2530 04565 CLR QF4-4(SP) 04041 103 2534 045665 CLR QF4-4(SP) 04047 <td>-</td> <td></td> <td></td> <td></td> <td></td> <td></td>	-					
90 12470 020365 CMP R3,D+6(SP) 90 12476 02036 BME DLMS16 91 12476 02032 BNE DHS16 92 12502 020324 CLR R4 93 12502 020324 CLR R4 94 12502 020324 CLR R4 95 12504 02703 DCMS161 MOV #1403,R# JENROR 3,3 97 12512 020443 BR ECISI6 SE SE 98 12514 012700 UNUSI61 MOV #1403,R# JENROR 3,3 97 12512 020435 BR ECISI6 SE JENROR 5,8 91 1252 020445 ECISI6 TST -(SP) JFAKE SIGN 100 2522 020456 CLR R5,SERRA 020770 MONOSE 103 2534 020465 CLR Q+2=4(SP) 020477 020477 103 2544 02065 CLR Q+4=4(SP) 0200022 020022<						
000016 91 12474 101026 BHI DLWS16 92 12476 001032 BNE DHIS16 93 12500 005216 INC #SP FBUMP EXPONENT 94 12500 00504 CLR R4 95 12504 002465 BR FLTS16 95 12504 002463 BR FLTS16 95 12514 012700 DUNDS16; MOV #4005,RD JEMROR 5,8 96 12520 004043 BR ECIS16 BR JEMROR 5,8 97 12512 004043 BR ECIS16 BR JEMROR 5,8 99 12520 004056 CLR GRADE JEMROR 5,8 102 02520 ZEKS161 CLR GRADE JEMROR 5,8 102 02520 026402 CLR GRADE GRADE 102 0250 02602 CLR GRADE GRADE 103 2534 05066 CLR GRADE GRADE 104						
91 12474 101020 HI DLMS16 92 12500 00502 GAL UHIS16 94 12502 005004 LR R4 95 12506 012760 DCMS105 HOV H1403,RØ JEHROR 3,3 97 12512 006403 BR ELIS16 JEHROR 3,3 97 12512 006403 BR ELIS16 98 12514 012760 UNDS165 MOV #4005,RØ JEHROR 5,8 0014005 ECLS161 TST -(SP) JFAKE SIGN 100 200746 102 2520 00566 CLR QF9+,(SP)+ JFLUSH EXP AND SIGN 102 2530 05066 CLR QF4=4(SP) 00014 103 2544 05065 CLR QF4=4(SP) 000022 103 2540 005065 CLR QF4=4(SP) 000022 104 2550 00602 ROR R1 JTO ENSURE TMAT N=0 108 2554 005065 CLR QF4=4(SP) 000016 <	90	24/0			CMP	KJ, U*6 (SF)
92 12476 00132 SAE OHISI6 93 12500 005216 INC #SP JBUMP EXPONENT 94 12502 005604 CLR R4 95 12506 012700 DCMS16: MOV #1403,R0 JEMROR 3,3 001403 97 12512 00603 BR ECISI6 98 12514 012700 UNUSI6: MOV #4005,R0 JEMROR 5,8 004005 99 12520 005746 ECLSI6: TST -(SP) JFAKE SIGN 100 2522 004567 ECISI6: JSR R5,SERRA 007270 101 2526 022626 ZEMS16: CMP (SP)+,(SP)+ JFLUSH EXP AND SIGN 102 2522 005666 CLR Q+0=4(SP) 000014 103 2534 005665 CLR Q+2=4(SP) 000014 104 2540 005665 CLR Q+4=4(SP) 000020 105 2540 005665 CLR Q+4=4(SP) 000020 105 2540 005065 CLR Q+4=4(SP) 000020 105 2540 005065 CLR Q+6=4(SP) 000020 105 2550 006000 DLMS16: ROR R0 JFALVE DENOMINATUR (C=0) 106 2550 006000 DLMS16: ROR R0 JFALVE DENOMINATUR (C=0) 108 2550 006002 R0H R2 110 2550 006002 R0H R2 110 2550 006003 R0R R3 111 2550 005025 INC #SP JCOMPENSATE EXPONENT 112 2564 012705 DHISI6: MOV #9,,R5 JGO DO FIRST 9 QUOTIENT BITS 000015 113 2570 004767 JSR PC,DV1S16 0004767 JSR PC,DV1S16 0000765 TST R5 JSEE IF DONE 115 2600 005705 TST R5 JSEE IF DONE 115 2600 005705 TST R5 JSEE IF DONE 115 2600 005705 TST R5 JSEE IF DONE 116 2600 005705 TST R5 JSEE IF DONE 117 2602 00125 BNE FLISI6 JYES, REST OF NUMERATOR IS 0 118 2600 005705 TST R5 JSEE IF DONE 119 2610 004767 JSR PC,DV1S16 000020 HIS, R5 JGO DO 16 MORE BITS 119 2610 004767 JSR PC,DV1S16 119 2610 004767 JSR PC,DV1S16 110 2600 005765 TST PS PC,DV1S16 110 2600 005765	.				ធដ។	Di wete
93 12500 005216 INC #SP JBUMP EXPONENT 94 12502 005004 CLR R4 95 12506 012700 DCHSIBI MOV #1403,R0 JEHROR 3,3 97 12512 000403 BR ECISI6 98 12514 012700 UNUSI6: MOV #4005,R0 JEHROR 5,8 99 12520 005765 ECISI6 TST =(SP) JFAKE SIGN 100 2252 005765 CLSI6: IST =(SP) JFAKE SIGN 101 2252 005765 CLR Q+2-4(SP) Q0014 102 2530 005056 CLR Q+2-4(SP) Q0022 103 2534 005056 CLR Q+2-4(SP) Q0022 103 2544 005065 CLR Q+2-4(SP) Q0022 104 2550 005065 CLR Q+2-4(SP) Q0022 105 2544 005065 CLR Q+2-4(SP) Q0022 106 2550 005060 R0R						
94 12502 005004 CLR R4 95 12504 002455 BR FLTS16 96 12506 012780 DCHS16: MOV #1403,R0 /ERROR 3,3 001403 97 12512 2004033 BR EC1S16 98 12514 012780 UNUS16: MOV #4005,R0 /ERROR 5,8 99 12520 005746 ECLS16: TST -(SP) /FAKE SIGN 100 2522 004067 ECLS16: TST -(SP) /FAKE SIGN 100 2522 004067 ECLS16: CLR (SP) //SP)+ /FLUSH EXP AND SIGN 101 2526 022520 ZENS16: CLR (SP) 102 2530 005066 CLR (4+2=4(SP)) 103 2534 005066 CLR (4+2=4(SP)) 103 2534 005066 CLR (4+2=4(SP)) 103 2534 005066 CLR (4+2=4(SP)) 105 2544 005066 CLR (4+2=4(SP)) 105 2544 005066 CLR (4+6=4(SP)) 106 2550 006027 BR RTNS16 107 2552 006000 DL#S16: ROR R0 / HALVE DENOMINATOR (C=0) 108 2554 006002 ROR R1 /TO ENSURE THAT N <d 109 2566 006002 ROR R2 110 2506 006002 ROR R2 110 2506 006003 ROR R3 111 2526 006003 KOR R3 111 2550 006075 JSR PC,DY1S16 106 2570 004767 JSR PC,DY1S16 113 2570 005765 TST R5 /SEE IF DONE 114 2574 110466 MOVB R4,G(SP) /SAVE ALL MIGH ORDER 0 FRACTIO 115 2640 005765 TST R5 /SEE IF DONE 117 2622 005765 TST R5 /SEE IF DONE 117 2622 005765 TST R5 /SEE IF DONE 117 2622 00125 BNE FLIS16 /YE8, REST OF NJMERATOR IS 0 118 2640 01275 JSR PC,DY1S16 119 2610 005765 TST R5 /SEE IF DONE 117 2622 00125 BNE FLIS16 /YE8, REST OF NJMERATOR IS 0 118 2640 01275 JSR PC,DY1S16 119 2610 005765 TST R5 /SEE IF DONE 119 2610 005765 JSR PC,DY1S16 000020 119 2610 005765 TST R5 /SEE IF DONE 119 2610 005765 JSR PC,DY1S16 000020 119 2610 005765 MOV R4,0*2(SP) 119 2610 004767 JSR PC,DY1S16 000020 119 2610 004767 JSR PC,DY1S16 000156 120 2014 010466 MOV R4,0*2(SP)</d 						
95 12504 000465 BR FLTS16 96 12506 012700 DCMS161 MOV #1403,R0 JERROR 3,3 97 12512 200403 BR ECIS16 98 12514 012700 UNUSIG: MA4005,R0 JERROR 5,8 99 12520 005765 ECIS15 TST -(SP) JFAKE SIGN 100 2526 22502 ZENSIGE CLR Q+0-4(SP) Q007270 101 2526 22526 ZENSIGE CLR Q+0-4(SP) JFLUSH EXP AND SIGN 102 2530 005065 CLR Q+2-4(SP) Q0014 103 2534 005065 CLR Q+2-4(SP) 000020 Q00022 Q00022 Q Q 104 2540 005065 CLR Q+4-4(SP) 000020 Q00022 Q Q Q 105 2544 005065 CLR Q+4-4(SP) 000022 Q Q Q Q Q 105 2556 Q <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td></td<>						
96 12506 012700 DCHS161 MOV #1403,R0 JERROR 3,3 001403 97 12512 000403 BR EC1S16 98 12514 012700 UNUS161 MOV #4005,R0 JERROR 5,8 004005 99 12520 005766 ECLS161 TST =(SP) JFAKE SIGN 100 2522 004567 ECIS161 JSR R5,StRRA 007270 101 2566 02262 ZEKS161 CMP (SP)+,(SP)+ JFLUSH EXP AND SIGN 102 2530 005066 CLR Q+2=4(SP) 100 0015 104 2540 005066 CLR Q+2=4(SP) 100 0015 105 2544 005066 CLR Q+4=4(SP) 106 0250 005066 CLR Q+6=4(SP) 106 0250 005066 CLR Q+6=4(SP) 106 2550 005066 CLR Q+6=4(SP) 106 2550 005066 CLR Q+6=4(SP) 106 2550 005076 RR R1 JTO ENSURE THAT N=D 108 2554 005070 RR R1 JTO ENSURE THAT N=D 109 2556 006002 ROR R2 100 2560 006003 ROR R3 111 2560 006003 ROR R3 112 2560 006003 ROR R3 112 2560 006003 ROR R3 113 2570 004767 JSR PC,DV1S16 000016 114 2574 110466 MOVB R4,Q(SP) JSAVE ALL MIGH ORDER Q FRACTIO 115 260 005765 TST R5 JSEE IF DONE 116 2600 005765 TST R5 JSEE IF DONE 117 2602 005765 TST R5 JSEE IF DONE 119 2610 005767 JSR PC,DV1S16 000020 119 2610 005767 JSR PC,DV1S16 000020 119 2610 005767 JSR PC,DV1S16 119 2610 005767 JSR PC,DV1S16 120 2614 01276 MOV R4,0+2(SP) 120 2614 01366 MOV R4,0+2(SP)						
001403 BR EC1s16 97 12512 000403 BR EC1s16 98 12514 012700 UNDS16: MOV #4005,R0 JENROR 5,8 99 12522 004574 ECLS161 TST -(SP) JFAKE SIGN 100 2522 004574 ECLS161 JSR R5,SERRA 007270 UNDS16: MOV R5,SERRA 001252 004574 ECLS161 JSR R5,SERRA 001252 004566 CLR Q+0-4(SP) 1012 2530 005066 CLR Q+2-4(SP) 000014 Q00020 Q4-4-4(SP) 103 2534 005066 CLR Q+4-4(SP) 000012 000022 BR 104 2540 005066 CLR Q+4-4(SP) 000022 000022 BR 105 254 00600 UHS161 ROR R0 106 2550 006002 R0R R1 JTO ENSURE THAT N=0 109 2556 006003 KOR R3 110 2560 006003 KOR R3 111 2560 00601 INC #SP 112 2564 012705 DHIS161 MOV #9,R5 JGO DO FINST 9 QUOTIENT BITS 113 2570 004767 JSR PC,DV1516 000016 JEXCEPT NORMAL BIT <td></td> <td></td> <td></td> <td>DCHS161</td> <td></td> <td></td>				DCHS161		
97 12512 200403 BR EC1\$16 98 12514 012700 UND\$161 MOV #4005,R0 JEKROR 5,8 98 12520 004005 TST =(SP) JFAKE SIGN 100 2522 0040746 EC1\$161 TST =(SP) JFAKE SIGN 101 2526 022626 ZEK\$161 CMP (SP)*,(SP)+ JFLUSH EXP AND SIGN 102 2530 02566 CLR Q+2=4(SP) 000014 103 2534 005066 CLR Q+2=4(SP) 104 2540 005066 CLR Q+4=4(SP) 105 2544 005066 CLR Q+4=4(SP) 106 2554 005066 CLR Q+4=4(SP) 106 2554 005066 CLR Q+6=4(SP) 106 2554 005066 CLR Q+6=4(SP) 107 252 006000 UH\$\$161 ROR R1 JTO ENDMINATOR (C=0) 108 2554 005061 ROR R2 JEXEPN (C=0)	,					
98 12514 012700 UNUS16: MOV #4005,R0 ;EMROR 5,8 99 12520 003546 ECLS16: TST =(SP) /FAKE SIGN 100 2522 004567 ECLS16: JSR R5,SERRA 00770 (05770 101 2526 022526 ZEMS16: CMP (SP)+,(SP)+ /FLUSH EXP AND SIGN 000014 000014 103 2534 003066 CLR 000020 000016 000020 000020 000020 000020 000022 GR 000022 000022 106 2550 00000 UL#S16: ROR 107 2550 000000 RNR 108 2554 00000 UL#S16: ROR 109 2556 000000 ROR 109 2556 000000 ROR 110 2560 000000 ROR 110 2560 000000 ROR 111 2560 000000 ROR 112 2564 012705 DH1S16: MOV 113 2570 000176 INC 114 2574 110466 MOVB 115 IS270 00020 116 2600 005705 TST 117 2602 00175 IST 118 2570 00025 HNE 119 2610 00075 TST	97	2512			BR	EC1516
99 12520 005746 ECLS161 TST - (SP) /FAKE SIGN 100 2522 004567 ECLS161 JSR R5,SERRA 007270 101 2526 02262 ZEKS161 CMP (SP)+,(SP)+ /FLUSH EXP AND SIGN 102 2530 005066 CLR Q+2=4(SP) 000014 103 2534 005066 CLR Q+2=4(SP) 000020 000020 105 2544 005066 CLR Q+4=4(SP) 000020 105 2544 005066 CLR Q+4=4(SP) 000020 105 2544 005066 CLR Q+6=4(SP) 000022 106 2550 00000 DL#S161 ROR R0 HALVE DENOMINATOR (C=0) 108 2554 006001 ROR R1 /TO ENSURE THAT N=0 109 2556 006002 ROR R2 110 2560 005003 KOR R3 111 2562 005216 INC #SP /COMPENSATE EXPONENT 112 2564 012705 DHIS161 MOV #9.,N5 /GO DO FIRST 9 QUOTIENT BITS 000020 114 2574 110466 MOVB R4,Q(SP) /SAVE ALL HIGH ORDER Q FRACTIO 000020 115 115 116 2600 005705 TST R5 /SEE IF DONE 117 2602 00125 BNE FLIS16 /YES, REST OF NUMERATOR IS 0 118 2604 412705 MOV #16.,RS /GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1S16 110 2600 004767 JSR PC,DV1S16 0000020 119				UNUS16:	MOV	
100 2522 004567 ECISI6: JSR R5,SÈRRA 007270 101 2526 022626 ZERSI6: CMP (SP)+,(SP)+ ;FLUSH EXP AND SIGN 102 2530 005066 CLR Q+0-4(SP) 000016 103 2534 005066 CLR Q+2-4(SP) 000020 105 2544 005066 CLR Q+6-4(SP) 000022 106 2550 00002 RDR RC R1 ;TO ENSURE THAT NeD 109 2556 006002 RDR R2 110 2556 006002 RDR R2 110 2556 006002 RDR R2 111 2560 006003 RDR R3 111 2560 006003 RDR R3 113 2570 004767 JSR PC,DV1516 0000176 114 2574 110466 MOVB R4,Q(SP) JSAVE ALL MIGH ORDER Q FRACTIO 000020 ;EXCEPT NORMAL BIT 115 2640 005705 TST R5 ;SEE IF DONE 117 2602 001025 BNE FLIS16 ;YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16, R5 JGO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1515 119 2610 004767 JSR PC,DV1515 120 2614 010466 MOV #16, R4,Q(SP)				i de la composición de		
007270 101 2526 022626 ZER\$151 CMP (SP)+, (SP)+ ;FLUSH EXP AND SIGN 102 2530 005066 CLR Q+0-4(SP) 000016 104 2540 005066 CLR Q+2=4(SP) 000016 104 2540 005066 CLR Q+6=4(SP) 000022 105 2544 005066 CLR Q+6=4(SP) 000022 BR RTN\$16 107 2552 006000 DLW\$16; ROR R0 /HALVE DENOMINATOR (C=0) 108 2554 006000 DLW\$16; ROR R1 /TO ENSURE THAT N <d 109 2556 006002 ROR R2 100 2560 006003 ROR R3 111 2562 005216 INC \$SP /COMPENSATE EXPONENT 112 2564 012705 DH1\$16; MOV #9.,R5 /GO DO FIRST 9 QUOTIENT BITS 000011 113 2570 044767 JSR PC,DV1\$16 00076 114 2574 110466 MOVB R4,Q(SP) /SAVE ALL HIGH ORDER Q FRACTIO 000020 //EXCEPT NORMAL BIT 115 2600 005705 TST R5 /SEE IF DONE 116 2600 005705 TST R5 /SEE IF DONE 117 2602 00125 BNE FL1\$16 /YE\$, REST OF NUMERATOR IS 0 118 264 012705 MOV #16.,R5 /GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$15 119 2610 004767 JSR PC,DV1\$15 119 2610 005705 TST R5 /SEE IF DONE 119 2610 005705 TST R5 /SEE IF DONE 119 2610 004767 JSR PC,DV1\$15 119 2610 004767 JSR PC,DV1\$15 119 2610 004767 JSR PC,DV1\$15 119 2610 004767 JSR PC,DV1\$16 119 2610 004767 JSR PC,DV1\$15 120 2614 012765 MOV #16.,R5 /GO DO 16 MORE BITS 120 2614 012765 MOV R4,Q+2(SP) 000022</d 						
101 2526 02262 ZEK\$161 CMP (SP)+, (SP)+ (FLUSH EXP AND SIGN 102 2530 005066 CLR 0+0-4(SP) 103 2534 005066 CLR 0+2=4(SP) 104 2540 005066 CLR 0+4=4(SP) 105 254 005066 CLR 0+6=4(SP) 105 2550 00022 106 2550 000477 BR RTN\$16 107 2552 006000 DLW\$161 ROR R0 /HALVE DENOMINATOR (C=0) 108 2554 006001 ROR R1 /TO ENSURE THAT N <u 109 2556 006002 ROR R2 110 2560 006003 ROR R3 111 2552 005216 INC \$\$P /COMPENSATE EXPONENT 112 2564 012705 DH1\$161 MOV #9,,R5 /GO DO FIRST 9 QUOTIENT BITS 113 2570 004767 JSR PC,DV1\$16 114 2574 114466 MOVB R4,Q(SP) /SAVE ALL MIGH ORDER 0 FRACTIO 115 //EXCEPT NORMAL BIT 116 2600 005705 TST R5 /SEE IF DONE 117 2602 001025 BNE FL1\$16 /YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16, R5 /GO DO 16 MORE BITS 119 2610 004767 JSR PC,DV1\$16 119 2614 010466 MOV R4,Q+2(SP) 120 2614 010466 MOV R4,Q+2(SP)</u 	100	2522		EC1516:	JSR	R5, SERRA
102 2530 005066 CLR Q+0-4(SP) 000014 103 2534 005066 CLR Q+2-4(SP) 000016 104 2540 005065 CLR Q+4-4(SP) 000022 106 2550 000477 BR RTN\$16 107 2532 006000 UL#\$16: ROR R0 /HALVE DENOMINATOR (C=0) 108 2554 006001 ROR R1 /TO ENSURE THAT N <d 109 2556 006002 ROR R2 110 2560 006003 AOR R3 111 2562 005215 INC ØSP /COMPENSATE EXPONENT 112 2564 012705 DH1\$16: MOV #9,,R5 /GO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC,DV1\$10 0000176 114 2574 110466 MOVB R4,Q(SP) /SAVE ALL HIGH ORDER Q FRACTIO 000020 115 116 2600 005705 TST R5 /SEE IF DONE 117 2602 001025 BNE FL1\$16 /YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16,R5 /GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$10 000022 119 2610 004767 JSR PC,DV1\$10 000022 119 2610 004767 JSR PC,DV1\$10 000022 119 2610 004767 JSR PC,DY1\$10 000022 120 2614 010466 MOV R4,Q+2(SP) 000022</d 	-	_		· · · ·	a	
000014 103 2534 005065 CLR Q+2=4(SP) 000020 104 2540 005065 CLR Q+4=4(SP) 000022 105 2544 005066 CLR Q+6=4(SP) 000022 106 2550 0000477 BR RTN\$16 107 2552 006000 DL#\$16: ROR R0 / HALVE DENOMINATOR (C=0) 108 2554 006001 ROR R1 /TO ENSURE THAT N <d 109 2556 006002 ROR R2 110 2550 006003 ROR R3 111 2562 005216 INC #SP /COMPENSATE EXPONENT 112 2564 012705 DH1\$16: MOV #9.,R5 /GO DO FIRST 9 UUCTIENT BITS 000016 114 2574 110466 MOVB R4,Q(SP) /SAVE ALL MIGH ORDER Q FRACTIO 000020 115 116 2600 005705 TST R5 /SEE IF DONE 117 2602 001025 BNE FL1\$16 /YES, REST OF NUMERATOR IS Ø 118 2604 012705 MOV #16.,R5 /GO DO 16 MORE BITS 0000156 119 2610 004767 JSR PC,DY1\$15 119 2610 004767 JSR PC,DY1\$15 120 2614 010466 MOV R4,Q+2(SP) 000022</d 				2EK\$101		
103 2534 005066 CLR Q+2=4(SP) 000016 104 2540 005066 CLR Q+4=4(SP) 000022 105 2544 005066 CLR Q+6=4(SP) 000022 106 2550 000477 BR RTN\$16 107 2552 006000 DL#\$16: ROR R0 / HALVE DENOMINATOR (C=0) 108 2554 006001 ROR R1 /TO ENSURE THAT N <d 109 2556 006002 ROR R2 110 2560 006003 ROR R3 111 2562 005216 INC #SP /COMPENSATE EXPONENT 112 2564 012705 DH1\$16: MOV #9.,R5 /GO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC,DV1\$16 000076 114 2574 110466 MOVB R4,Q(SP) /SAVE ALL HIGH ORDER Q FRACTIO 000020 /EXCEPT NORMAL BIT 115 2600 005705 TST R5 /SEE IF DONE 117 2602 001025 ENE FL1\$16 /YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16.,R5 /GO DO 16 MORE BITS 0000156 119 2610 004767 JSR PC,DV1\$16 000020 119 2610 004767 JSR PC,DV1\$16 000022</d 	102	2530			LLK.	
000016 104 2540 005066 CLR Q+4=4(SP) 000020 000022 105 2544 000022 106 2550 000022 106 2550 000022 106 2554 000022 106 2554 006000 DL#si6: 107 BR RTN\$16 107 2552 006000 DL#si6: 108 2554 006002 R0R 109 2556 006002 R0R 110 2560 006003 R0R 111 2562 00526 INC 112 2564 012705 DH\$156: 112 2564 012705 DH\$156: 113 2570 004767 JSR 114 2574 110466 MOVB R4,G(SP) JSAVE ALL MIGH ORDER Q FRACTIO 115 Iff Iff Iff Iff Iff Iff 116 2600 005705 TST RS JSEE IF DONE 117 <	1	0534			C1 2	0+2-4(SP)
104 2540 005065 CLR Q+4=4(SP) 000020 105 2544 005066 CLR Q+6=4(SP) 000022 106 2550 000477 BR RTNs16 107 2552 006000 DL#s16: ROR R0 HALVE DENOMINATOR (C=0) 108 2554 006002 ROR R1 /TO ENSURE THAT N≤D 109 2556 006002 ROR R2 110 2560 006003 ROR R3 111 2562 005216 INC ØSP /COMPENSATE EXPONENT 112 2564 012705 DH1516: MOV #9,R5 /GO DO FIRST 9 QUOTIENT BITS 0000411 JSP PC,DV1516 000476 114 2574 110466 MOVB R4,Q(SP) /SAVE ALL HIGH ORDER Q FRACTIO 000020 //S SEE IF DONE 115 // 2602 00125 BNE FL1516 /YES, REST OF NUMERATOR IS 0 116 2600 005705 TST R5 /SEE IF DONE 117 2602 00125 BNE FL1516 /YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16,R5 /GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1516 000156 /// SR PC,DV1516	103	2004			w Ly N	
000020 105 2544 005065 CLR Q+6=4(\$P) 106 2550 000477 BR RTN\$16 107 2552 006000 DL#\$16: ROR R0 JHALVE DENOMINATOR (C=0) 108 2554 006001 ROR R1 JTO ENSURE THAT N <u< td=""> 109 2556 006002 ROR R2 110 2562 00503 ROR R2 111 2562 005215 INC \$</u<>	104	2540			CLR	0+4-4(SP)
105 2544 005066 CLR Q+6=4(SP) 000022 106 2550 000477 BR RTN\$16 107 2552 006000 DLW\$16; ROR R1 JTO ENSURE THAT N <d 109 2556 006002 ROR R2 110 2560 006003 ROR R3 111 2562 005216 INC \$\$P JCOMPENSATE EXPONENT 112 2564 012705 DH1\$16; MOV \$9,,R5 JGO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC,DV1\$16 000076 114 2574 110466 MOVB R4,Q(SP) JSAVE ALL MIGH ORDER Q FRACTIO 000020 115 117 2602 005705 TST R5 JSEE IF DONE 117 2602 001025 BNE FLI\$16 JYES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16,,R5 JGO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$16 000020 119 2610 004767 JSR PC,DV1\$16 000020 120 2614 010466 MOV #16,,R5 JGO DO 16 MORE BITS 000020 139 2610 004767 JSR PC,DV1\$16 000020 149 2610 004767 JSR PC,DV1\$16 000020 150 000220 151 000020 152 000020 153 000020 154 000025 BNE FLI\$16 JYES, REST OF NUMERATOR IS 0 164 012705 MOV #16,,R5 JGO DO 16 MORE BITS 000020 155 000020 156 000020 157 000020 158 000020 159 000020 150 000020 150</d 	104	60-0				
000022 106 2550 000477 BR RTN\$16 107 2552 006000 DL#\$16: ROR R0	105	2544			CLR	Q+6=4(SP)
107 2552 006000 DLW\$16; ROR R0 #HALVE DENOMINATOR (C#0) 108 2554 006001 ROR R1 #TO ENSURE THAT N <d< td=""> 109 2556 006002 ROR R2 110 2560 006003 ROR R3 111 2562 005216 INC 05P #COMPENSATE EXPONENT 112 2564 012705 DH1516; MOV #9, R5 #GO DO FIRST 9 GUOTIENT BITS 000011 INC 05P #COMPENSATE EXPONENT #ITS 113 2570 004767 JSR PC, DV1516 000176 I14 2574 110466 MOVB R4, Q(SP) #SAVE ALL HIGH ORDER Q FRACTIO 115 IEXCEPT NORMAL BIT #IS #EXCEPT NORMAL BIT #IS 116 2600 005705 TST R5 #SEE IF DONE 117 2602 001025 BNE FL1516 #YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16.,R5 #GO DO 16 MORE BITS 119 2610 004767 JSR</d<>						
108 2554 006001 ROR R1 JTO ENSURE THAT N <u< td=""> 109 2556 006002 ROR R2 110 2560 006003 ROR R3 111 2562 005216 INC ØSP JCOMPENSATE EXPONENT 112 2564 012705 DHIS16: MOV #9,,R5 JGO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC,DV1\$16 000176 114 2574 110466 MOVB R4,Q(SP) JSAVE ALL HIGH ORDER Q FRACTIO 115 </u<>	106	2550	000477		BR	RTN\$16
109 2556 006002 RUR R2 110 2560 006003 ROR R3 111 2562 005216 INC @SP JCOMPENSATE EXPONENT 112 2564 012705 DH1\$16: MOV #9,R5 JGO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC,DV1\$16 000176 114 2574 110466 MOVB R4,Q(SP) JSAVE ALL HIGH ORDER Q FRACTIO 115	107	2552	000000	DLWS16:	ROR	
110 2560 006003 ROR R3 111 2562 005216 INC #SP #COMPENSATE EXPONENT 112 2564 012705 DH1516; MOV #9, R5 #GO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC, DV1\$16 000176 000176 000176 #SAVE ALL HIGH ORDER Q FRACTIO 115 EXCEPT NORMAL BIT #EXCEPT NORMAL BIT 116 2600 005705 TST R5 #SEE IF DONE 117 2602 001025 BNE FLIS16 #YES, REST OF NUMERATOR IS Ø 118 2604 012705 MOV #16, R5 #GO DO 16 MORE BITS 119 2610 604767 JSR PC, DV1\$16 000156 120 2614 010466 MOV R4, Q+2(SP) 000022 EXCEPT MOV R4, Q+2(SP) #00022					ROR	
111 2562 005216 INC #SP #COMPENSATE EXPONENT 112 2564 012705 DH1516; MOV #9,R5 #GO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC,DV1516 0000176 000176 114 2574 110466 MOVB R4,Q(SP) #SAVE ALL HIGH ORDER Q FRACTIO 115 000020 #EXCEPT NORMAL BIT #IGH ORDER Q FRACTIO 116 2600 005705 TST R5 #SEE IF DONE 117 2602 001025 BNE FL1516 #YES, REST OF NUMERATOR IS Ø 118 2604 012705 MOV #16,R5 #GO DO 16 MORE BITS 119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010465 MOV R4,Q+2(SP) 000022	109	2556	006002		ROR	
112 2564 012705 DH1\$16: MOV #9,,R5 /GO DO FIRST 9 QUOTIENT BITS 000011 113 2570 004767 JSR PC,DV1\$16 000176 114 2574 110466 MOVB R4,Q(SP) /SAVE ALL HIGH ORDER Q FRACTIO 000020 115 116 2600 005705 TST R5 /SEE IF DONE 117 2602 001025 BNE FL1\$16 /YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16,,R5 /GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010466 MOV R4,Q+2(SP) 000022						
000011 113 2570 004767 JSR PC,DV1\$10 000176 114 2574 110466 MOVB R4,Q(SP) ISAVE ALL HIGH ORDER Q FRACTIO 000020 115 116 2600 005705 TST R5 ISEE IF DONE 117 2602 001025 BNE FL1\$16 IYES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16.,R5 IGO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010466 MOV R4,Q+2(SP) 000022						
113 2570 004767 JSR PC,DV1\$16 000176 MOVB R4,Q(SP) ISAVE ALL HIGH ORDER Q FRACTIO 114 2574 110466 MOVB R4,Q(SP) ISAVE ALL HIGH ORDER Q FRACTIO 115 116 2600 005705 TST R5 ISEE IF DONE 117 2602 001025 BNE FL1\$16 IYES, REST OF NUMERATOR IS Ø 118 2604 012705 MOV #16.,R5 IGO DO 16 MORE BITS 119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010466 MOV R4,Q+2(SP) 000022	112	2564		DH1516:	MOV	#9,,R5 IGO DO FIRST 9 QUOLIENT BITS
000176 114 2574 110466 MOVB R4,Q(SP) ISAVE ALL HIGH ORDER Q FRACTIO 115 isoud200 iexcept Normal Bit 116 2600 005705 TST R5 ISEE IF DONE 117 2602 001025 BNE FL1516 IYES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16.,R5 IGO DO 16 MORE BITS 119 2610 004767 JSR PC,DV1\$16 000156 MOV R4,Q+2(SP) 000022 MOV R4,Q+2(SP)		0			1-0	
114 2574 110466 MOVB R4,Q(SP) /SAVE ALL HIGH ORDER Q FRACTIO 115	113	25/0			124	PC, DV1510
000020 ;EXCEPT NORMAL BIT 115 116 2600 005705 TST R5 ;SEE IF DONE 117 2602 001025 BNE FL1516 ;YES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16.,R5 ;GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$16 120 2614 010466 MOV R4,Q+2(SP) 000022 FE FE		057 A	000170		MOVA	PALOISPI ISAVE ALL HIGH OPDED D FRACTION
115 ;EXCEPT NORMAL BIT 116 2600 005705 TST R5 ;SEE IF DONE 117 2602 001025 BNE FL1516 ;YES, REST OF NUMERATOR IS Ø 118 2604 012705 MOV #16.,R5 ;GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010466 MOV R4,Q+2(SP) 000022	114	20/4			NUYD	Reflect
116 2600 005705 TST R5 JSEE IF DONE 117 2602 001025 BNE FL1516 JYES, REST OF NUMERATOR IS 0 118 2604 012705 MOV #16.,R5 JGO DO 16 MORE BITS 000020 000156 JSR PC,DV1\$16 120 2614 010466 MOV R4,Q+2(SP) 000022 000022 000022 000022	115		0000020			FXCEPT NORMAL BIT
117 2602 001025 118 2604 012705 000020 119 2610 004767 000156 120 2614 010466 MOV R4,0+2(SP) 000022		2600	005705		TST	
118 2604 012705 MOV #16,,R5 7GO DO 16 MORE BITS 000020 119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010465 MOV R4,Q+2(SP) 000022						
000020 119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010466 MOV R4,G+2(SP) 000022						
119 2610 004767 JSR PC,DV1\$16 000156 120 2614 010466 MOV R4,Q+2(SP) 000022	~ • -					
000156 120 2614 010465 MOV R4,Q+2(SP) 000022	119	2610			JSR	PC, DV1\$16
000022	···					
	120	2614			MOV	R4, Q+2(SP)
121 2620 005705 TST R5						
122 2622 001015 BNE FL1816						
123 2624 012705 MOV #16,,R5	123	2024	012/05		MUV	ng #19, j KO ta basa kanala na kanala na Na kanala na

\$DVD05 MACRU VR04=14 07=SEP=72 11:43 PAGE 21+

¢

¢.

(

\$

Þ

SDVD05 MACRU VR04=14 07=SEP=72 11:43 PAGE 21+

			000020	j .		
	124	2630	004767		JSR	PC, DY1516
	÷		000136			
	125	2634	1010460		MOV	R4, G+4(SP)
						a com for a first from the second
		ne a in	000024		167 av 196	e 🔄 🔄 e serie de la companya de
			005705		TST	R5 Carter and
	127	2642	001005		BNE	FL1\$16
	128	2044	012705		Mav	#16, /R5
			000020			
	129	2650	004767		JSR	PC, DV1516
	a m ·		000116			
	a (2.04	OFF.			F _ m	
			000401		BR	FLTS16
e -	131	2656	005004	FL1516:		RA ICLEAR LOWEST ORDER QUOTIENT
				FLTS161	MOV	(SP)+,R5 JPUSH UP EXPONENT
	133	2662	062705		ADD	#200,R5 JADD IN EXCESS 200
			000200			그는 사람이 가지 않는 것은 것이 같이 많이 많이 많이 많이 많다. 것이 같아요. 이 것이 같아요.
	134	2666	003712		BLE	UND\$16 JUNDERFLOW
		2670			CMP	#377,R5
			000377		A GUY	
	A 115 AT	0.01				
			002433		BLT	OVRS16 JOVERFLOW
	137	2676	110566		MOVB	R5,0+1=2(SP) JINSERT EXPONENT IN RESLT
			000017		•	
	138	2702	000020	SGNS161	ROR	(SP) + /INSERT QUOTIENT SIGN
		2704			ROR	Q+0+4(SP)
			000014			
		2710	006066	4	000	040 A(0)
	140	e / .∎.¥)			ROR	Q+2+4(SP)
			000016			
	141	2714	006066		ROR	Q+4=4(SP)
			000020			
	142	2720	006004		ROR	RA
		2722			ADC	R4 IROUND
	144				ADC	
	****	6/64			AUC	Q+4-4(SP)
			000020			
	145	2100	005566		ADC	Q+2=4(SP)
			000016			
	146	2734	005566		ADC	Q+0+4(SP)
			000014			그는 사람이 집에 가지 않는 것이 아니는 아이에게 가지 않는 것이 아니는 것이 아니는 것이 아니는 것이 같아. 이 아이는 것이 아니는 않는 않아. 아니는 것이 아니는 아니는 것이 아니는 아니는 것이 아니. 것이 아니는 것이 아니. 것이 아니는 것이 아니는 것이 아니는 것이 아니. 것이 아니는 것이 아니. 것이 아니는 것이 아니는 아니 아니는 것이 아니. 것이 아니는 것이 아니는 것이 아니. 것이 아니는 것이 아니. 것이 아니는 것이 아니는 것이 아니. 것이 아니는 것이 아니. 것이 아니는 것이 아니는 것이 아니. 아니는 것이 아니 아니. 아니는 것이 아니. 아니는 것이 아니는 것이 아니. 아니는 것이 아니. 아니는 아니는 것이 아니 아니는 것이 아니
	147	2740	010466		MOV	R4, Q+6=4(SP) JINSERT LOW ORDER FRACTION
	÷		000022			CALA CLAIMENT ATURENT AND DUREY LUNCTION
	1 4 9	3741	103406		Net	
					BCS	01116
			102405	M	BVS	QV1316
				RTNS161	MOV	(SP)+,R5
	151	2752	012604		MOV	(SP)+,R4
	152	2754	062706		ADD	#8., SP IFLUSH FIRST ARGUMENT
		•	000010		2000 TTT	いってます 山下 こうたく機能です にんきやてす 一位時間度が動作す
	153	2760			TMD	
				GUIRAAA	JMP	@(R4)+
				OV13161		- (SP) / FAKE EXP
	122	2704	012700	OVRS161	MOY	#2003, R0 ; ERROR 3, 4
			002003			
			000653		BR	ECL516
	157	2772	006304	Dv1516:	ASL	R4 ISHIFT QUOTIENT
			006303		ASL	R3 ISHIFT NUMERATOR
			006102		ROL	R2
			006101			
					ROL	
			006100		ROL	RØ
			103420		BCS	GUS10 JGUARANTEED TO GU
	163	3006	026500		CMP	D+0+2(SP),R0 JCOMPARE HIGH DIVISOR AND DIVIDE

SDVD05 MACRO VR04=14 07=SEP=72 11:45 PAGE 21+

s

É

3012	101034		BHI	NG0816	FJUMP IF DIVISOR BIGGER
				G0\$16	JJUMP IF DIVISOR SMALLER
3016	026601				
3022			BHI	NG0516	
					P). 82
3032			BHI	NG0\$16	
					2), D3,
****			A cit.	0.04#(0	
3040			841	NICOSIA	
3044	001400				JJUMP IF NUMERATUR =DENOMINATOR
		605161			
VUNC		000101	000	0+0+=(3	P),R3 INFN-D
3050			800	00	
					53] /h./h.
0000	··· ···		300	U#4+4(S	PJ, R2
1.5.			0		
••					
30/0			SUB	0+2+2(\$	Pl,R1 and the second
	· • •		÷		
3076		•	SUB	0+0+2(S	P), RØ
		к			
			INC	R4	INSERT QUOTIENT BIT
3104	005305	NGU\$161	DEC	RS	ICOUNT LOOP
3106			BGT	DV1516	
3110	000207		RTS .	PC	
3112	005204	NOUS16:	INC	R4	FINSERT LAST 1 BIT IN QUOTIENT
· · ·	esta is A an A		BR	EQ1516	יין איז
3114	000401				
3114 3116	000401	EQ2\$16:	ASL	R4	FINISH OUT QUOTIENT WITH AIS
		EQ2\$16: EQ1\$16:			FINISH OUT QUOTIENT WITH 015
3116	006304		ASL Dec	R 4 R 5	FINISH OUT QUOTIENT WITH O'S
3116 3120 3122	006304 005305		ASL DEC BGT	R4 R5 Eu2\$16	
3116 3120 3122 3124	006304 005305 003375 005205	EQ1816:	ASL DEC BGT INC	R4 R5 EU2\$16 R5	FLAG NO MORE NUMERATOR
3116 3120 3122	006304 005305 003375 005205	EQ1816:	ASL DEC BGT	R4 R5 Eu2\$16	
	3014 3016 3022 3024 3026 3022 3026 3026 3026 3034 3034 3034 3034 3034 3034 3034 3054 305	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	3014 103414 3016 026601 000014 3022 101030 3024 103410 3026 026002 000016 030216 3034 103404 3035 026003 000020 00020 3042 101020 3044 001422 3045 166003 60502 000020 3042 101020 3044 001422 3045 166003 60502 000020 3054 105602 000020 3056 3056 025602 3057 166502 000016 3070 3064 005601 3070 166601 000014 3076 3074 00500 3075 166502 000012 3102 3102 005204 3104 005305 3105 005204 3112 005204 <td>3012 101034 BHI 3014 103414 BLD 3016 026601 CMP 3022 101030 BHI 3024 103414 BLD 3022 101030 BHI 3024 103410 BLD 3026 026602 CMP 000016 BHI 3034 3034 103404 BLD 3035 026603 CMP 000020 BHI 3042 101020 BHI 3044 001422 BEQ 3045 166603 G0\$163 3046 166603 G0\$163 3054 00020 BHI 3045 166603 G0\$163 3054 005500 SBC 3055 025602 SBC 3056 025602 SBC 3056 025600 SBC 3060 166601 SUB 000016 SBC SUB 3076 166601 SUB 000012 SUB<td>3012 101034 BHI NG0516 3014 103414 BL0 G0516 3016 026601 CMP D+2+2(S 3022 101030 BHI NG0516 3024 103410 BL0 GU516 3026 026502 CMP D+4+2(S 302016 S032 101024 BHI NG0516 3034 103404 BL0 GU516 GU516 3035 026503 CMP D+6+2(S 000020 3042 101020 BHI NG0516 S0516 3044 001422 BEQ NQ0516 3044 001422 BEQ NQ0516 3045 166603 G0516 SUB D+6+2(S 000020 SBC R2 3054 005601 SBC R1 3056 025602 SUB D+4+2(S 000016 SBC R0 3056 025602 SUB D+2+2(S 000016 SBC R0 SBC 3070 1666</td></td>	3012 101034 BHI 3014 103414 BLD 3016 026601 CMP 3022 101030 BHI 3024 103414 BLD 3022 101030 BHI 3024 103410 BLD 3026 026602 CMP 000016 BHI 3034 3034 103404 BLD 3035 026603 CMP 000020 BHI 3042 101020 BHI 3044 001422 BEQ 3045 166603 G0\$163 3046 166603 G0\$163 3054 00020 BHI 3045 166603 G0\$163 3054 005500 SBC 3055 025602 SBC 3056 025602 SBC 3056 025600 SBC 3060 166601 SUB 000016 SBC SUB 3076 166601 SUB 000012 SUB <td>3012 101034 BHI NG0516 3014 103414 BL0 G0516 3016 026601 CMP D+2+2(S 3022 101030 BHI NG0516 3024 103410 BL0 GU516 3026 026502 CMP D+4+2(S 302016 S032 101024 BHI NG0516 3034 103404 BL0 GU516 GU516 3035 026503 CMP D+6+2(S 000020 3042 101020 BHI NG0516 S0516 3044 001422 BEQ NQ0516 3044 001422 BEQ NQ0516 3045 166603 G0516 SUB D+6+2(S 000020 SBC R2 3054 005601 SBC R1 3056 025602 SUB D+4+2(S 000016 SBC R0 3056 025602 SUB D+2+2(S 000016 SBC R0 SBC 3070 1666</td>	3012 101034 BHI NG0516 3014 103414 BL0 G0516 3016 026601 CMP D+2+2(S 3022 101030 BHI NG0516 3024 103410 BL0 GU516 3026 026502 CMP D+4+2(S 302016 S032 101024 BHI NG0516 3034 103404 BL0 GU516 GU516 3035 026503 CMP D+6+2(S 000020 3042 101020 BHI NG0516 S0516 3044 001422 BEQ NQ0516 3044 001422 BEQ NQ0516 3045 166603 G0516 SUB D+6+2(S 000020 SBC R2 3054 005601 SBC R1 3056 025602 SUB D+4+2(S 000016 SBC R0 3056 025602 SUB D+2+2(S 000016 SBC R0 SBC 3070 1666

				.TITLE .IFDF	SDV103 CND817	
				GLOBL	SDVISER	
			3			INTEGER DIVIDE ROUTINE
			1			ייזין או איז
			1	SOVI	V003A	
			1			
			1	COPYRIG	HT 1971.	DIGITAL EQUIPMENT CORP. MAYNARD, MASS.
			1	CALLED	IN THE PO	LISH MODE WITH THE NUMERATOR AT 2(SP)
0			2		DENOMINA	
1			1			GER QUOTIENT ØSP.
2		000000		RØ=%Ø	· · · · · · · · · · · · · · · · · · ·	
3		000001		R1=%1		
4		000002		R2=%2		
5		000003		N3=%3		
6		000004		R4=%4	×	
7		000005		R5#%5		
8		000000		SP=%6		
9		177304		MQ=1773	ð 4	
ø				. IFNDF	EAE&MULD	IV
	13130	005000	SDVI	CLR	RØ	ICLEAR RESULT SIGN
		012001		MOV	(SP)+,R1	
		003003		BGT	P1517	JJUMP IF DENOMINATOR PLUS
		001443		BEQ	CHKS17	ICAN'T DIVIDE BY ZERD
		005200		INC	RØ	INOTE #
		005401		NEG	R1	
7		011003	P15171	MOV	eSP,R3	JGET NUMERATOR
	- ,	003003	1 8 7 8 7 7	BGT	P2517	JJMP IF NUMERATOR PLUS
		001434		BEQ	ZERS17	JJUMP IF IT IS ZERO
		005200		INC	RØ	ISET RESULT SIGN
		005403		NEG	R3	
		010446	P2\$171	MOV	R4,=(SP)	
		012704	~~~.	MOV	#8,,K4	ISET FOR 8 ITERATIONS
Ú.	10108	000010			a a trans	in personal Γ Γ από το που τοποιετικό ο φορατική παν Γ
٨	13164	005002		CLR	R2	ICLEAR HIGH ORDER DIVIDEND
		000303		SWAB	R3	ITEST HIGH ORDER NUMERATOR
		001402		BEQ	DIVS17	JUMP IF HIGH ORDER QUOTIENT IS 0
		006304		ASL	R4	TWE NEED ALL 16 ITERATIONS
		000303		SWAB	RJ	JUNDU THE ABOVE SWAB
			OIVS171	ASL	R3	JOOUBLE DIVIDEND
		006102	₩# \$ \$ ₽ \$ (\$	ROL	R2	
		001405		BEQ	LUPS17	IJUMP IF NO CHANCE THIS TIME
		005203		INC	RJ	JASSUME IT WILL GO. INSERT QUOTIENT BIT
	3M	160102		SUB		ITRIAL STEP
		103002		BHIS	LUPS17	IOK
		060102		ADD	R1, R2	JDIVIDEND NOT BIG ENDUGH YET
		005303		DEC	R3	TAKE OUT QUOTIENT BIT
7	13216	005304	LUPS171		R4	 Construction of the second se Second second sec second second sec
		003360	सम्बद्धाः प्रस्तिति हि	BGT		IGO AGAIN
		012504		MOV	(SP)+,R4	
		005403		NEG	R3	ITEST FOR NEGMAX
		006200		ASR		JGET RESULT SIGN
		103402		BCS	P3\$17	JUMP IF =
		005403		NEG	R3	FANSWER IS POSITIVE
		102404		BVS		JUMP IF ANSWER IS -NEGMAX
			835171	MOV		JOUTPUT RESULT
		010316 000134	- 0 W 1 / 1	JMP	@(R4)+	JRETURN
	1 1 1 1 1 1 1 1 1 1 1 1	000100		₩ FEF	· · · · · · · · · · · · · · · · · · ·	हा उन 1899 उ पसर75 t र

86

(

SDV103 MACRO VR04=14 07-SEP=72 11:43 PAGE 224

ŧ.

57	13242 005016	2ER\$171	CLR	OSP FRESULT IS Ø
58	13244 000134		JMP	@(R4)+
59			. ENDC	
60		1	SOVI FO	R THE EAE
61			. IFOF	EĂE
62		SOVI	MOV	#MQ, R0 PPDINT TO MQ
63			MOV	(SP)+,R1 JGET DIVISOR
64			BEQ	CHK\$17 JJUMP IF DIVISION BY 0
65			MOV	(SP)+, ORU ; DIVIDEND TO MQ
66			TST	- (RØ) JSKIP AC
67			MOV	R1,=(R0) JDIVISOR TO DIV
68			CMP	(RØ)+,(RØ)+ /POINT TO MQ
69			MOV	ORD,=(SP) JGET QUOTIENT
70			JMP	@(R4)+ IRETURN TO USER
71			ENDC	
72		1		R MUL/DIV
73		•	. IFDF	MULDIV
74		SOVII	MOV	2(SP),R1 JGET LOW ORDER DIVIDEND
75			WORD	006700 IJSEX RØ JEXTEND SIGN
76			WORD	071026 /;DIV (SP)+,R0 /DIVIDE
77			MOV	RØ, øSP JPUSH QUOTIENT
78			BCS	CHK\$17 JJUMP IF ERROR
79			JMP	@(R4)+
80			ENDC	
81	13246 004567	CHKS171	JSR	R5, SERR JERROR 3,5
· • •	006534			
82	13252 000134		JMP	@ (R4)+
83	13254 003		BYTE	3
84	13255 005		BYTE	5
85			. ENDC	

SOVR08 MACRO VR04=14 07=5EP=72 11:43 PAGE 23

		7	•TITLE •IFDF •GLOBL \$DVR ==	SDVR08 CNDS18 SDVR,SE THE RE		DE RO	UTINE				
		1	SDVR	VØØ8A							
)]	COPYRIG	HT 1971,	1972 01	GITAL	EQUIP	MENT C	ORP.,	MAYNARD,	MA
		1	C.1.) ED		- 	1005					
			THE NUM	IN THE P ERATOR I	S THE S	AFCOND	TTEM	ON THE	STACK		
		<i>i</i>	AND THE	DENOMIN	ATOR IS	NN T	OP.				
		2	TAKES T	HE QUUTI	FNT AND	PUTS	IT ON	TOP			
		1	OF THE	STACK IN	THEIR	PLACE					•
	000000	•	R0=x0								
	100000		R1=%1								
	000002		R2=%2								
	000003		R3#%3								
	000004		R4=%4								
	000005		R5=%5								
	000006		5P=%6								
	000007		PC=%7								
	177304		MQ=1773	04							
	177312		NOR#177	312							
	177314		LSH#177	314							
	177316		ASH=177	310							
	000000		FØ#%0								
	000001		F1=%1								
	000010		Ŭ≡8,								
	000014		N=12.								
	000014		Q=12.								
			. IFOF	FPU		-					
		SOVRI	.WORD	170001	IISET		1. E.I			DIVISOR	
			WORD	172526	JILDF)+,F1				1
			, WORD	172426	TILDE	* *)+,FØ	NTVTNE		OTATORNE	,
			, WORD	174401	//DIVE		ге / 	DIVIDE		IENT TO	STO
			, WORD	174048	TISTE	r 0 /			19001		010
			JMP	@(R4)+							
			.ENDC	E CI I							
12060	0104×4	\$nv#!	.IFNDF	FPU Rà , = (sp	1						
	010446	a U Y R I	MOV	R5, - (SP							
	010548		CLR	R0,413F	1						
	005000		CLR	RU R1							
	005001 005040		CLR CLR	=(SP)							
	005366		ASL	N+0=2(S	P)	ISH	IFT NU	IMERATO	R		
LVE V	0000012		- W		·						
13274	006116		ROL	ØSP	IGET *	NUMERA	TOR SI	LGN			
	005046		CLR	= (SP)							
	005766		1 ST	U(SP);		CHE	CK FUF	X Ø,Ø D	ENOMIN	ATOR	
******	000010		-	· · · · ·							
13304	001456		BEQ	DCHS18;							
	156516		BISB	N+1 (SP)	,@SP	1 GE	T NUME	RATOR	EXPONE	NT	
	000015			1 mm							
13312	001451		BEQ	ZERSIS		IF NU	MERAT	IR IS Z	ERO		
	156500		BISB	N(SP),R	P X						

88

(

Ć

(

(

(

ŧ,

¢

		•	000014		
	55	13320		SWAB	RØ JUEFT JUSTIFY NUMERATOR FRACTION
		13322	000261	SEC	FINSERT NORMAL BIT
		13324	006000	ROR	RØ
	58	13326	156600	8158	N+3 (SP), K0
	90	TAGEA	000017		
	15 A			0+60	NILO (SO) 81
	23	10005	156501	BISB	N+2(SP),R1
			000016		
		13336		SWAB	R1. State of the second st
	61	13340	005002	CLR	R2
	62	13342	005003	CLR	R3
	63	13344	006366	ASL	D(SP) /SHIFT DENOMINATOR
			000010		
	64	13350	025566	ADC	2(SP) /GET RESULT SIGN
			000002		
	65	13354	156502	BISB	D+1(SP),R2 JGET DIVISUR EXPONENT
			000011		
	66	13360	160216	SUB	R2,05P ISUBTRACT EXPONENTS
		13362		CLR	R2
	68	13364		BISB	D(SP),R2 JGET HIGH URDER FRACTION
		10044	000010		
	en	12270		SAAB	R2
		13370			FINSERT NORMAL BIT
			000261	SEC	
	-	13374	006002	ROR	R2
	72	133/6	156602	BISB	0+3(SP),R2
			000013		
÷,	73	13402	156603	BISB	D+2(SP),R3
			000012		
	74	13405	000303	SWAB	RJ
۰.	75			. IFOF	EAEIMULDIV
	76			CLC	JENSURE NUM, AND DENOM. +
	77			ROR	RØJ
	78			ROR	RI JLOW ORDER RI AND R3 ARE 0
	79			ROR	
	80	1. A.		ROK	R3
	81		-0	ENDC	RØ, RZ JCOMPARE HIGH NUMERATOR AND DENOMINATOR
			020002	CMP	
	83	19412	103440	BLO	
	84			IFNDF	EAESMULDIV
	85		101034	BHI	DLWS18 JJUMP IF DENOMINATUR LOW
			020103	CMP	R1,R3 ICOMPARE LOW ORDER PARTS
			101032	BHI	DLW\$18
	88	13422		BNE	DHI518
	89	13424	005066	CLR	Q(SP) JQUOTIENT FRACTION IS 1
			000014		
	90	13430	005216	INC	PSP JBUMP EXPONENT
			005005	CLR	R5
			000445	BR	FLT\$18
	93	****	स्त स्ट ड्रेंग ¹¹ र मन पर	ENDC	े स्ट्रिय के प्राण्ड के 1999 के 1999 क
				.IFDF	EAEIMULDIV
	94				DLWS18 JJUMP IF DENOMINATOR LOW OR SAME
	95			BHIS	NEMBRO INCHL TE REMANTINUIAN PAN AN ANAR
	96		المراجع	.ENDC	10511 1001
			022026 ZERS181	CMP	(SP)+, (SP)+ ;FLUSH EXP AND SIGN
			000415	BR	EC1\$18
	99	13442	005726 DCH5181	TST	(SP) + JFLUSH EXP
	100	3444		MOV	#4003,R0 JERROR 3,8
			004003		

SDVR08 MACRO VR04-14 07-SEP-72 11:43 PAGE 23+

101	3450	000406		BR	ECL\$18
			OV1518:		-(SP) /FAKE SIGN
103			OVR5181	MOV	#3003, R0 JERROR 3,6
***		003003	*******	· · 🗰 ·	a na manana ang kanana ang kanana Kanana ang kanana ang ka
104	3460	000402		BR	ECLS18
			UNDS181		#1405,R0 JERROR 5,3
100	ONOS		ONA 2141	HQ Y	Atagalus trucky ata
1.00		001405	Emi aios	* * *	40 BAL 481 116 (1 A 7 MA)
	-		ECLS181		(SP)+ /FLUSH SIGN
107	34/0	004567		JSR.	R5, SERRA
		006322		* . *	
108	3474		EC1818:	CLR	Q+Q=4(SP) ;RETURN Ø
		000010			
109	3500	005066		CLR	Q+2=4(SP)
		000012			
110	3504	000445		BR	RTN\$18
111	3506	006000	QLW\$18:		RØ JHALVE NUMERATOR (C=0)
112	3510	005001		ROR	R1 ITO ENSURE THAT N <d< td=""></d<>
113	3512	005216		INC	@SP /COMPENSATE EXPONENT
114				. IFNDF	EAESMULDIV
	3514	012704	DHIS181		#9, H4 JGO DO FIRST 9 QUOTIENT BITS
** *		000011			
116	3520	004767		JSR	PC, DV1518
	17 26 77 36	000104			
117	3524	110560		MOVB	R5,Q(SP) ;SAVE ALL HIGH DRDER Q FRACTION
* * *	****	000014			াৰ উপৰ দেশৰ হ'ব বিজেপ বিজেপ বিজেপে
118		000014			;EXCEPT NORMAL BIT
	3530	005704		TST	R4 /SEE IF DONE
120	3532	001402		BEQ	NTØS18 INØ, NUMERATOR NUT Ø
121	3534		ne iz		
		005005		CLR	
122		000404		BR	FLTS18
123	9940		NTØ5181	MOV	#16,,R4 JGO DO 16 MORE BITS
		000020			
124	3544			JSR	PC, DV1\$18
		000060		111 A	
125				. ENDC	
126				. IFOF	EAEIMULDIV
127			DH15181	CLC	
128				ROR	R3 JENSURE LOW HALF DENOM, +
129				ROR	RØ ISCALE NUMERATOR FOR FIXED PT, DIVIDE
130				ROR	\mathbb{R}^{1} . The second secon
131				.ENDC	
132				IFOF	EAE
133				MOV	#MQ,R5 /POINT TO MQ
134				MOV	R1,0R5 INUMERATOR TO AC,MQ
135				MOV	R0,=(R5)
136				MOV	R2,=(R5) /(A+S+B)/C
137				TST	(R5) + JPOINT TO AC
138				MOV	(R5)+,R1 KEEP REMAINDER
139				MOV	(R5) +, R4 IKEEP QUOTIENT
140				MOV	R3,0K5 /GET Q+D
141				TST	-(R5) PROINT TO MQ
142				ASR	R1 ISCALE R
143				SUB	R1,=(R5) /Q+D=R
				DEC	P#ASH
144					
145				MQV	$R_{2,=}(R_{5})$; $(Q*D=R)/C$
146				CMP	(R5)+, (R5)+ MQ
147				NEG	PR5

90

(

				MOV	#2,0#ASH IMULT BY 4
148				ADD	#2,0#ASH
149				CLR	Ø#NOK INORMALIZE
150					##NOR, #SP JAPPLY TO EXPONENT
151				SUB	##6, ##LSH /POSITION NORMAL BIT
152				MOV	
153				MOV	
154				MOV	eR5, K5
155				. ENDC	Nation of The State of the Stat
156				.IFDF	MULDIV
157				MOV	RØ, R4 INUMERATOR TO DIVÍDEND
158				MOV	R1,R5
159				WURD	071402 // DIV R2,R4 /(A+S+B)/C
160				MOV	R5,R1 /SAVE REMAINDER
161				MOV	R4, RU ISAVE QUOTIENT
162				WORD	070403 11 MUL R3,R4 IGET G*D
163					RIGHT ISCALE R SECTORE AND SECTORE SECTORES
164				SUB	R1,R4 J0*D*R
165		4 A 3			073427,=1 ;; ASHC #=1,R4 ;SCALE
166		٠		WORD	071402 11 DIV R2,R4 1GET (0+D-R)/C
167				NEG	R4 / (R+Q+D)/C
168		1997 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 - 1998 -	en de la Maria Antonio	.WORD	073427,=14, ;; ASHC #14,,R4 ;UNSCALE
169				ADD	R0,R4 10+(R=0+D)*S/C
170			NBT5181	WORD	073427,1 11 ASHC #1,R4 1SHIFT
171				BMI	NBIS18 ICHECK FOR NORMAL BIT
172				DEC	PSP ICOMPENSATE EXPONENT
173				BR	NBTS18 JGO AGAIN
174			NB15181		073427,=7 ;;ASHC #=8,R4 ;ALIGN FRACTION
175				MOV	R4,Q(SP) ISTORE HIGH ORDER
176				ENDC	
177	3550	012604	FLT\$181	MOV	(SP)+,R4 JPUSH UP EXPONENT
178	3552	062704		ADD	#200,R4 JADD IN EXCESS 200
		000200			
179	3556	003741		BLE	UNDS18 JUNDERFLOW
180	3500	022704		СМР	#377,R4
		000377			
181	3564	002733		BLT	OVRS18 JOVERFLOW
182	3566	110466		MOVB	R4,Q+1=2(SP) #INSERT EXPONENT IN RESULT
		000013			
183	3572	006026	SGN\$181	ROR	(SP) + FINSERT QUOTIENT SIGN
184	3574	006066		ROR	Q+0-4(SP)
		000010			
185	3600	006005		ROR	Rb
200 11		005505		ADC	R5 IROUND
		005566		ADC	Q+0=4(SP)
		000010			
188	3610	010560		MOV	R5,Q+2=4(SP) ;INSERT LOW ORDER FRACTION
		000012			
189	3614	103715		BCS	OV1518
		102715		BVS	0V1\$18
			RTNS181	MOV	(SP)+,R5
		012004	28 E 13 MB W M B	MOV	(SP) +, R4
		022626		CMP	(SP)+, (SP)+ /FLUSH FIRST ARGUMENT
		000134		JMP	@(R4)+
195	- 4 - 4 4	~~~~~		.IFNDF	EAE8MULDIV
	3630	006305	DV15181	ASL	R5 ISHIFT QUOTIENT
		006301	*****	ASL	R1 ISHIFT NUMERATOR
		006100		ROL	RØ
120	~0~*	000400		·· • • •	

SDVR08 MACRO VR04=14 07=SEP=72 11:43 PAGE 23+



92

ТŔ

SDXP05 MACRO VR04=14 07=SEP=72 11:43 PAGE 24

(

s

4

			.TITLE .IFDF	\$DXP05 CND\$19
		1	DEXP	VUUSA
		I COPYR		1,1972 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MAS
		1	.GLOBL	DEXPISERRAT
			. IFNDF	FPU
			.GLOBL .ENDC	\$ADD, \$SBD, \$MLD, \$DVD, \$ID, \$DI, \$POLSH, \$POPR4;
		j		TRAN DEXP FUNCTION
		1		SEQUENCEI
		1	JSR	R5, DEXP
		1	BR .WORD	ARGUMENT ADDRESS
		1 4 1	a static a	
			RETURNS	E**ARG IN RØ = R3.
		1	D:	
	000000	1		
	000001		R1=%1 R2=%2	
	000002 000003	8	R3#%3	
	000004		R4=%4	
I	000005		R5=%5	
1	000000		SP=26	
	000000		F0=%0	
1	000001		F1=%1	
1	000002		F2=%2	
	000003		F32%3	
		DEVD	. IFOF	FPU @2(R5),RØ; GET HIGH ORDER ARG
		DEXPI	.ENDC	02(R5),R0; GET HIGH ORDER ARG
-			.IFNDF	FPU
13710	010546	DEXPI	MOV	R5, (SP) ISAVE RETURN
13712	016504		MOV	2(R5),R4 JGET ARG POINTER
	000002			
	011400		MOV	ØR4,RØ IGET HIGH URDER ARG
	1. 12 m 12 m 4		ENDC	
13720			BGT CMP	POS819
13722	141062		L MP	R0,#141662 ARG IS =
13726			BHI	ZERS19 JJUMP IF ARG <88.7
13730			BR	SMTS19 JJUMP TO TEST SMALL MAGNITUDE ARG
		P05\$191		R0, #41660
	041060			
13736			BHI	OVR\$19 JJUMP IF ARG >87
		SMT\$19;		RØ JDUMP SIGN
13742			CMP	R0, #43000
• 2 * A #	043000		ain	ONES19 JJUMP IF ARG MAGNITUDE <2++=60
13746	103444		BLO .IFNDF	DNES19 JJUMP IF ARG MAGNITUDE <2++=60 FPU
13750	162706		SUB	#20., SP IGET WORK SPACE
14146	000024			na wa wa wa ka ini manawa ki ini manakina. 27 ki filomiona
13754	062704		ADD	#8,,R4 /POINT TO LOW ORDER ARG
	000010			
13760			MOV	- (R4) - (SP) ; PUSH ARG

SDXP05 MACRO VR04=14 07=SEP=72 11:43 PAGE 24+

52	13752	014446		MOV	= (R4), = (SI	?)				
53	13764			MOV	- (R4),=(S)					
54	13766	014446		MOV	= (R4) , = (SI	?)				
55	13770	012746		MOV	#013761,=		1 PUSH	L0G2(E)		
	•. · · ·	013761								
56	13774	012746		MOV	#024534,7	(SP)				
		024534								
57	14000	012740		MOV	#125073,=	(SP)				
÷,		125073			· · · · · · · · · · · · ·					
58	14004	012746		MOV	#40270,=(SP)				
••	9 , 19, 19, 19, 19, 19, 19, 19, 19, 19, 19	040270								
59	14010			JSR	R4, SPOLSH		;ENTER	PULISH	MODE	
	A . P. H. P.	005630								
60	14014			WORD	SMLD P	Y = X + L OG	2(E)			
61	*-	0144541		WORD	DUP519					
62		0176401		WORD		INT (X+L	0G2(E))		
63	14022			WORD	ADJS19	₩ 1 1 7 7 7 4 9				
64	14024			, WORD		Z=INT(X	*L0G2(E)),Y>#	0) Z=Z-1,	Y<0
65		0007001		WORD	SSBD					
		0143621		WORD)#16+(X	*L0G2(E) -FLOA	f(2))	
67		0144541		WORD		2 COPIE			· · · · · ·	
68		0176401		WORD	501		-			
69		0144021		WÜRD		SAVE IN	TEGER	PART OF	2**Y	
70		0160461		WORD		E=O=INT				
		0007001		WORD	\$580,0165		1E/16			
r .		0143701		Mar da ann da mha		• •				
72	14046			WORD	DUPS19, DUI	P\$19	JGET 3	COPIES		
r 164	14050						,			
73		0161461		WORD	SMLD 1	E+E				
		0175761		WORD	SPOPR4		1808 F	*E TO RI	FGS	
		0141161		WORD	UPL519					
	14000		ONE \$191		#40200,RU		PFSIII	T IS 1.		
10	1-0-0	040200	6N#915.	n u v	HAREADIUA		I CHOOL			
77	14054			BR	21519					
78	14066		OVR\$191		#1004,R0		FERROR	4.2		
70	14040	001004			******		y m (77 F 98		
79	14072			BR	ECL\$19					
80	14074		ZERS191	MOV	#2005,RØ		IERROR	5.4		
00	1-01-4	002005	AC	11 0 t			.	V F H		
8.	14100		ECL\$191	158	R5, SERRA					
• Î	1-1-00	005712		₩ ₩ 11 .	12 M B. H. H. H. 17 17 14					
82	14104	005000		CLR	RØ 11	RESULT	15 0			
83	14106		Z15191	CLR	R1					
84	14110	005002	*****	CLR	R2					
85	14112	005002		CLR	RJ	· · · · ·			·	
		0000000		BR	007519					
87	14116		UPLS191	MOV	#033343,*	(99)	PUSH	P3#7.21	350341084	4819083
07	14190	033343	ML#875.	11 4 1	ABOOMMON.	¥₩8.4.0	Friedrik	। अत्य साम्रा•के∖ ्		
2 a	14122			MOV	#015345,=	(SP)				
88	14165	012/48		119 T -	40108801×	rat ¥				
8 0	14125	015545		MOV	#152405,=	(SP)				
89	14125			i u y	#176#621 ···	(ar)				
^		152405		MAN	4010716 -	1001				
90	14132			MOV	#040746,=	1941				
A -		040746								
91	· · · · · ·		1	Man	04 (00)					
92	14136			MOV	R3,-(SP)					
93	14140	010246		MOV	R2,=(SP)					

94

(

:

SDXP05 MACRO VR04=14 07=SEP=72 11143 PAGE 24+

Ų

Þ.

94	14142	010146	MOV	R1,=(SP)	
	14144	010040	MOV	RU,=(SP)	
96	9 m 9 m 9	भा सा स्थान्त्र गा पर 1			
		a10746	MOV	#153703, = (SP)	PUSH P1#,057761135831801928
9/ 3	14140	012746	e V V	#100, 001 (017	
		153703			
98 (14152	012740	MOV	#153011, - (SP)	
		153011			
99	14156	012746	MOV	#113360, = (SP)	
		113360			
100	4162	012746	MOV	#037154, = (SP)	and the second
100	4146	037154			
101		1	M O LI	07 (00)	
		010340	MOV	R3,=(SP)	ar an
103		010246	MOV	R2,=(SP)	
104	4172	010146	MOV	R1, = (SP)	$M_{\rm eff} = \frac{1}{2} \left[\frac{1}{2} $
105	4174	010040	MOV	R0,-(SP)	A second s
106					
	A . 7 6	012745	MOV	#171042,=(SP)	JPUSH Q0#20,813771196523036297
101	41.0		r i Qi Y	HTS THHEF CONST	
a	4-0-	171042	MANU	4077777 - FBB	
108	4202	012746	MOV	#074433,=(SP)	
		074433			
109	4206	012746	MOV	#101232,=(SP)	
		101232			
110	4212	012746	MOV	#041246, "(SP)	
	·	041245		•	
111	4014	004167	JSR	R4, SPOLSH	
112	-510	004467	4 G W	eration for the second	$\Delta = \frac{1}{2} \left[\frac{1}{$
_		005422			ANTIPLAN TO WORK COMPE
113	4222	0007041	. WORD	SADD, AUP\$19	JARE+E+Q0 TO WORK SPACE
	4224	0144101			
114	4220	0161461	, WORD	SMLD, SADD, SMLD	18#E+(E+E+P1+P0)
	4230				1. キャーション しゃく 感じ 彼ら之来 (なららか) シー
	4232	0161461			
115	4234	0144701	WORD	TWC319 JOUPLIC	ATE A AND B
				SADD, ABP\$19	JA+B TO WORD SPACE
110	4236	0007041	WORD	AVOANDLATA	
	4240	0144321		i an an an an An An An	ing series and the series of t
117	4242	0007001	. WORD	SSBD, SDVU	1 (A+B) / (A=B)
	4244	0122101			
118		0142501	WORD	SCLS19 JAPPLY	SCALE FACTORS
	4250	012705 SCL\$19:	-	#RT2\$19+8,,R5	POINT TO POWERS OF 2
* * 3	~ ~ ~ 0		राध्यत् .		n normania e en la maria mentra m En la mante en la maria de la maria de la mante en l
4	Antes	Ø145541	A C C		n an an Anna an
150	4204	006268 ASR\$191	7 C M	8. (SP) /SHIFT	We can assume the second se
		000010			a ann an Ann
		103010	BCC		F BIT IS OFF - contactor and - contactor
		014546	MQV	=(R5),=(SP)	;PUSH 2**((2**N)+D/16)
		014546	MOV	=(R5),=(SP)	
		014546	MOV	=(R5),=(SP)	
		014546	MOV	-(R5), -(SP)	
				R4, SPOLSH	
150	42/2	004467	JSR	Kwlar Arau	
		005346	·		ANTI MANIN DU LONIA MAMMANA ANA
127	4276	0161461	, WORD	SMLD,ASR\$19	IMULTIPLY BY ABOVE FACTOR AND
	4300	0142541			
128		001403 NML5191	BEQ	SC1519	
		162705	SUB	#8, K5 /POINT	TO NEXT POWER OF 2
				ाण्ड्र इत्या प्राप्त स्वाधिण है। संस्थान	
A 113 A	A 13 4 ~	000010	80	AS2#10	
130		000761 012600 SC1519;	BR	ASR519 (SP)+,RØ	POP RESULT
			HT 1 1 1		

SDXP05 MACRO VR04=14 07=SEP=72 11:43 PAGE 24+

132	4314	012601		MOV	(SP)+,R1
133	4310	012002		MOV	(SP)+,R2
134	4320	012603		MOV	(SP)+,R3
135	4322	005726		TST	(SP) + JFLUSH D
				MOV	(SP)+,R4 JGET Z
136	4324	012004			
137	4326	000304		SWAB	RA THING THE CHONENT MODIFIED
138	4330	105004		CLRB	R4 IMAKE INTO EXPONENT MODIFIER
139	4332	006204		ASR	R4
140	4334	060400		ADD.	R4, RØ JAPPLY TO RESULT
141	4336	100653		BMI	OVRS19 JJUMP IF OVERFLOW
142	4340	012005	OUTS191	MOV	(SP)+,R5 JPOP RETURN
143	4342		· · · · · · · · · · · · · · · · · · ·	RTS	R5 FRETURN TO USER
144			1		
145	4344	005775	ADJS191	TST	#2(R5) ITEST X
140	***				A set of the set of
	4 11 16 10	000002		BGE	ARNS19 JJUMP IF +
146	4350	002001			
147	4352	005316		DEC	
148	4354	011666	ARN\$191	MOV	ØSP,28,(SF) ISAVE Z AS AN INTEGER
		000034			
149	4360	000134		JMP	@ (R4) +
150			1		
151	4352	062716	M165191	ADU	#1000,0SP J16* STACK ITEM
		001000	,		
152	4366			JMP	0(R4)+ 1
153		Dec.e.	· •	* *	
154	4370	162716	0165191	SUB	#1000,0SP J1/16+STACK ITEM
194	407 Ø		A144431	900	n e e e e e e e e e e e e e e e e e e e
		001000		401	DERSIG JJUMP IF NO UNDERFLOW
155	4374	100001		BPL	
156	4376	005016		CLR	
157	4400	000134	0683191	JMP	@ (R4) +
158			1		
159	4402	011060	05V5191	MOV	@SP,26.(SP) ;SAVE D AS AN INTEGER
		000032			
160	4406	000134		JMP	l 🖉 🕻 🕅 👌 🕂 - Calder en la companya de la
161	· .		3 1 10 1 14		
162	4410	012066	AUPS191	MOV	(SP)+,38,(SP) ;A TO WORK SPACE
		000046			
163	4414			MOV	(SP)+,38.(SP)
100		000046			
	1120			MOV	(SP)+,38,(SP)
164	4420			1 S 🖌 🔻	И факульт Килики, как и странование со составляется с составляется с составляется с составляется с составляетс С Килики и Килики и Килики и составляется с составляется с составляется с составляется с составляется с составл
		000046		MAN	(60) + 38 (60)
165	4424	012066		MOV	(SP)+,38,(SP)
		000046		·	
166	4430	000134		JMP	€(R4)+ second and a second s
167			n h aga shirin ta		
168	4432	012666	ABP5191	MOV	(SP)+,22.(SP) IMOVE A+B TO WORD SPACE
		000026			
169	4436	012666		MOV	(SP)+,22.(SP)
		000026			
170	4442			MOV	(SP)+,22.(SP)
• 7 %		000026		· •	
171	4445			MOV	(SP)+,22. (SP)
171				··· v •	JAIA Lmww.r.aca
	4450	000026		TWD .	6 (8 4) -
172	4452	000134		JMP	@ (R4) +
173			1	1. m 1.	
174	4454		DUP\$19:	MOV	6(SP),=(SP) JDUPLICATE STACK ITEM
		000000		a se a de se da	

SDXP05 MACRO VR04=14 07=SEP=72 11144 PAGE 24+

(

依

(

(

 $[\mathbf{C}_{i}]$

i.

(

175	4400	016645		MOV	6(SP),=(SP)			
		000005						
176	4464	016546		MOV	6(SP),=(SP)			
*		000000		A				
177	4470	015645		MOV	6(SP),=(SP)			
		000005						
178	4474			JMP	@(R4)+			
179			1	-				
180	4176	012700	TWC5191	MOV	#8.,RØ JEIGHT	ITEMS		
100		000010	10000					
	4502		TW15191	MOV	14.(SP),=(SP)	IDUPLICATE 2 D	OUBLES	
181		000016	17184877					
182	4536	005300		DEC	RØ			
	4506			BGT	TW1519			
183	4510	003374		JMP	# (R4) +			
184	4015	000134	•	M Let.				
185								
186	A # 8 A	GAGDEE		WORD	040205,002363,0	31771.157145	12++1/2	
187		040265		SHURD	0402-010-200012		· · · · · · · · · · · · · · · · · · ·	
		002363						
		631771						
		157145			040010 013760 B	BOSIS 114951	12**1/4	
188	4524	040230		.WORD	040230,033760,0	000101104401	15****	
		033760						
	4530	···· •						
		134251					1044149	
189				, WORD	040213,112701,1	61752,105/2/	12**1/8	
	4536	112701						
	4540	161752						
	4542			and a second second			·	
190		040205	RT25191	, WORD	040205,125303,0	03/14,0441/3	12*+1/16	
	4546	125303						• • •
	4520	063714						
	4592	044173			and the second sec			
191				,ENDC		• .		
192			1					
193				. IFDF	FPU		Out DO	
194				SETO	1	DOUBLE PRECISI		
195				SETI	1	SHORT INTEGERS		
196				MOV	#FC0\$19, NØ1	POINTER TO CON	ISTANTS	
197				LDD	\$2(R5),F21	GET ARGUMENT	04522	
198				MODO	(RØ)+,F21	F2#FRACT (X+LOG		
199		ene patro		STCOL	F3,R41	Z=INT(X+LOG2(E		
200				TSTO	F21		*	
201			1.	CFCC				4.5
202				BGE	M165197	TEST F2	e kúzen	
 203				ADDD	#1.0+F2;	MAKE F2 POSITI		1000 A.
204				DEC	R41	AND ADJUST Z#2	(=1	
205			1					
206			M16519:	MODD	#16.0,F27	F2#FRACT(16+()		UAT(Z))
207				STCDI	F3, R31	D=INT (10+(,	• • •	
208				DIVD	#16,0,F21	E=F2/16		
209				LDD	F2,F3;			
210				MULD	F3,F3;	E*E		
211			1					
212			• •	LOD	F3,F1;			
213				ADDD	(RØ)+,F17	A=E+E+QU		
214				MULD	(RØ)+,F31			
					· · · · · · · · · · · · · · · · · · ·			

SDXP05 MACRO	VR04-14	07=SEP=72	11:40	PAUE 24+
--------------	---------	-----------	-------	----------

	•				
		ADDD	10014 531		
		MULD	(R0)+,F31 F2,F31	B=(E+E+P1 + P0)	
				DATEACANT + LAN	₹ <u>C</u> , s is the light
		LOD	F1, FØ;		
		ADDD	F3, FU1	A+B	× + 1
		5UB0	F3,F1;	A=B	
		DIVD	F1, F0;	(A+B)/(A=B)	
	1		a katika kati	* *	
	SCLS191	ASR	R31	SHIFT D	
	AA#012:	BCC	NML\$197		
				NILL TTOL N BY DAD	* AE O
		MULD	(RØ)+,FØ7	MULTIPLY BY ROO	
		BR	SCL\$191		. · · ·
	NMLS191		SC1519/		
		ADU	#8.,RØ1	POINT TO NEXT R	OOT OF 2
		BR	SCL\$191		
	1				
	SC15191	STD	F0,=(SP)1	MOVE RESULT TO	STACK
		MOV	(SP)+, R01	AND THENCE TO R	
		MOV	(SP)+,R1/	्रत्यान् प्रथम (देख्या∰क 1,444,144,	
		MOV	(SP)+,R21		
		MOV	(SP)+,R31	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
		SWAB	R41	CONVERT Z TO EX	PONENT MODIFI
		CLRB	R41		
	n an star Start an	ASR	RAL STATE STATE STATE		
		ADD	R4, RØ1	APPLY TO RESULT	
		BMI	OVR5191	JUMP IF OVERFLO	
		RTS	R51	EXIT	
	•	1.10			in the second
	0.5.4.0.4	Nou	44400 m		
	ONES191	MOV	#40200,R0	RESULT IS 1.	
		BR	Z1519		
	OVRS191	MOV	#1004,R0	JERROR 4,2	
and the second		BR	ECLS19		$\sum_{i=1}^{n-1} x_i ^2 = \sum_{i=1}^{n-1} x_i ^2 = \sum_{i$
	ZER\$191	MOV	#2005,RØ	JERROR 5,4	and the second second
	ECLS191	JSR	R5, SERRA	• •	
		CLR	RØ IRESULT	18 0	
	Z1519:	CLR	R1	** *	
	60 J W J W P				
		CLR	R2		
		CLR	R3		
		RTS	R51	EXIT	
	1				
	1 Martin 1997	ORDER-DE	PENDENT CONSTAN	rs .	
					าง
· · · ·	n an	RØ POINT	S AT NEXT CONST.	ANT TH LENG ARROTT	
		RØ POINT	IS AT NEXT CONST	ANT IN FFW VEROLI	
•) Fr0eto:				
	, FC0\$191		40270,125073,024		LUG2(E)
	; FC0\$191 ;	, WORD	40270,125073,024	4534,0137611	LUG2(E)
) FC05191 I	, WORD	40270,125073,02 041246,101232,07	4534,013761 <i>1</i> 74433,171042;	L0G2(E) Q0
	; FCO\$191 ;	,WORD .WORD .WORD	40270,125073,024 041246,101232,07 037154,113360,15	4534,0137611 74433,171042; 53011,153703;	LUG2(E) Qø P1
	FC0\$191	, WORD	40270,125073,024 041246,101232,07 037154,113360,15 040746,152405,01	4534,0137611 74433,171042; 53011,153703; 15345,033343;	L0G2(E) Q0
	FC0\$191	,WORD .WORD .WORD	40270,125073,024 041246,101232,07 037154,113360,15	4534,0137611 74433,171042; 53011,153703; 15345,033343;	LUG2(E) Qø P1
	FC0\$191	WORD WORD WORD WORD	40270,125073,024 041246,101232,07 037154,113360,15 040746,152405,01 040205,125303,00	4534,013761) 74433,171042; 53011,153703; 15345,033343; 53714,044174;	LUG2(E) Q0 P1 P0 2**1/16
		WORD WORD WORD WORD WORD WORD	40270,125073,024 041246,101232,07 037154,113360,15 040746,152405,07 040205,125303,06 040213,112701,16	4534,013761; 74433,171042; 53011,153703; 15345,033343; 53714,044173; 51752,105727;	LUG2(E) Q0 P1 P0 2**1/16 2**1/8
		WORD WORD WORD WORD WORD WORD WORD	40270,125073,024 041246,101232,07 037154,113360,15 040746,152405,07 040205,125303,06 040213,112701,16 040230,033760,05	4534,013761; 74433,171042; 53011,153703; 15345,033343; 53714,044173; 51752,105727; 50615,134251;	LUG2(E) Q0 P1 P0 2**1/16 2**1/8 2**1/4
		WORD WORD WORD WORD WORD WORD WORD WORD	40270,125073,024 041246,101232,07 037154,113360,15 040746,152405,07 040205,125303,06 040213,112701,16	4534,013761; 74433,171042; 53011,153703; 15345,033343; 53714,044173; 51752,105727; 50615,134251;	LUG2(E) Q0 P1 P0 2**1/16 2**1/8
		WORD WORD WORD WORD WORD WORD WORD	40270,125073,024 041246,101232,07 037154,113360,15 040746,152405,07 040205,125303,06 040213,112701,16 040230,033760,05	4534,013761; 74433,171042; 53011,153703; 15345,033343; 53714,044173; 51752,105727; 50615,134251;	LUG2(E) Q0 P1 P0 2**1/16 2**1/8 2**1/4

SEX	P04 1	ACRU VI	204=14 07	7=SEP=72	11143 PAGE 25 99
				TITLE	SEXP04
				. IFOF	CND\$20
2			1		
1			I and	EXP	VØØ4A
j			1 CAPVE	1041 497	1,1972 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MA
				TAULT TAN	TITALE RTATIVE EMATINEL AMOLACHITOMS INTERNATION
	•		•		
)				.GLOBL	EXP, SERRAJ
0				. IFNDF	FPU
12				.GLOBL .ENDC	SADR, SSBR, SMLR, SDVR, SIR, SRI, SPOLSH;
3				EXP	THE REAL EXPONENTIATION ROUTINE
4		a ²⁴	1	CALLING	
5			t i i shekara	JSR	R5,EXP
6			1 Carlos	BR	
7			j	WORD	ARG ADDRESS
8			1 A I		
9			7	RETURNS	EXPONENTIAL IN RØ AND R1.
0			a transformation and the second	12	
1		000000		ROFXO	
2		000001		R1=%1 R2=%2	
3 4		000002		RSEXS	
5		000004		R4=%4	
6		0000005		R5=%5	
7		000006		SP#%6	
8		000007		PC=%7	
9		000000		F0=%0	
Ø		000001		F1=%1	
1		000002		F2=%2	
2		000003	E.B.	F3#%3	ALART ACET LOODUFNT BATUTED
3	14554	016504	C X T I	MQV	2(R5),R4 JGET ARGUMENT POINTER
	14660	000002		MOV	ØR4,RØ JGET HIGH ORDER ARG
4 5	14502	011400 003004		BGT	PUSS20 JJUMP IF ARG +
	14564	020027	(2011) - A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A	CMP	RØ,#141662
Ŷ	1.40.44	141662			αν το κατά ματαγματικά την παραγματική την παραγματική το την παραγματική την παραγματική την παραγματική την Για το παραγματική την παραγματική την παραγματική την παραγματική την παραγματική την παραγματική την παραγματι
7	14570	101146		BHI	ZER\$20 JJUMP IF EXPONENT < +88.7
		000403		BR	SMTS20
9	14574	020027	PD\$\$201	CMP	RØ, #41660
		041660			
		101137		BHI	OVR520 JJUMP IF EXPONENT > 87
		006300	SMT\$201		RØ JDUMP SIGN
Z	14004	020027		CMP	RØ,#63000
78	14610	063000 103527		BLO	ONES20 JJUMP IF EXPONENT MAGNITUDE < 2**-28
3	14010	100451		IFNDF	FPU
	14612	005746		TST	-(SP) ISAVE SPACE FOR SCALE
		005046		CLR	=(SP) /PUSH A 1.
		012746		MOV	#40200,=(SP)
		040200			
8	14622	016446		MOV	2(RA),=(SP) JGET LOW ORDER ARGUMENT
		000002		14 m 1 *	
1.25	1 4676	011446		MOV	PR4,=(SP) /HIGH ORDER
		016446		MOV	2(R4), - (SP) INEED TWO COPIES OF IT

SEXP04 MACRO VR04=14 07=SEP=72 11:43 PAGE 25+

			Ar m fr	
		011446	MOV	PR4, = (SP)
52	14636	004467	JSR	R4, SPOLSH JENTER PULISH MODE
-		005002		NUCER BOUND I BOOLCS
53	200	0147321	.WORD	PLES20 /PUSH LOG2(E)
54	14644		. WORD	SMLR
55	14646		WURD	SRI JFIX LOG2(E) *X
56	14650		.WORD	ESVS20 ISAVE EXPONENT SCALE
57		0160621	. WORD	SIR JFLOAT IT
58	14654		.WORD	PLE320 /PUSH LOG2(E)
59	14656		.WORD	\$DVR
60	14600		. WORD	SSBR TOUGH CONSTRUCT FORCETANTS
		0147521	. WORD	CFR\$20 JPUSH CONTINUED FRACTION CONSTANTS
		0171621	WORD	SMLR IY+Y
63	14656		WORD	SADR JB1+Y+Y
	14670		. WORD	SDVR /A1/(B1+Y+Y)
	14672		.WORD	SADR JY+A1/(B1+Y+Y)
		0020101	.WORD	SADR 1 AU+Y+A1/(B1+Y+Y)
67		0132551	.WURD	\$DVR JY/(A0+Y+A1/(B1+Y+Y))
68		0147121	.WORD	INC320 $J=2+Y/(A0+Y+A1/(B1+Y+Y))$
	14702		. WORD	SADR /1-2+Y/
	14704		. WORD	DUPS20 JOUPLICATE IT
		0171621	, WORD	SMLR /(1-2+Y/)++2
	A	0150461	NORD	SCL\$20 JEXIT POLISH MODE AND SCALE RESULT
73	14712		ADD	#100200,@SP ;MULTIPLY BY -2.0
. .		100200	1	
•	14716	000134	JMP	@(R4)+ JGO BACK TO LIST
75		1		
76	14720	016646 DUP\$201	MOV	2(SP),-(SP) JOUPLICATE STACK ITEM
-		000002		
77	14724		MOV	2(SP), = (SP)
-		000002	10	
	147.30	000134	JMP	● (R4) + 1 of the second state of the second
79		ALATIE BIEROMI	MOV	#125073, -(SP) / PUSH LOG2(E)
80	14/32	012746 PLE5201	n U V	ATSONO1- (OL) ILAOU PRASCI
	1 1 4 3 4	125073 012746	MOV	#40270, . (SP)
81	14/00		THU Y	HANSLO MARCH
8 0	1 47 40	040270 000134	JMP	@(R4)+
	14/42	000104	M FIF	
83	14744	011665 ESV\$201	MOV	OSP, 10. (SP) ISAVE EXPONENT SCALE
0.4	1 4/ 44	011065 ESV\$201 000012		content to the resource of the
8	14780		TMD	A [A 5] A
	14/50	000134	JMP	• (R4) + and a second secon
86		DORTAR PRESDOT	้ออเ	OSP ISHIFT MODIFIED ARG
		006116 CFR\$201	ROL	
		006100	ROL	
89	14/20	162716	SUB	#400,0SP ;DIVIDE BY 2.
0.0	4.4760	000400	BIAS	ZFR\$20 JUNDERFLOW, MAKE ARG 0
90		101430	BLUS Ror	RØ JGET SIGN BACK
		006000 006016	ROR	ØSP
		011000	MOV	ØSP,RØ JGET MODIFIED ARGUMENT
		016601	MOV	2(SP),R1 JIN REGISTERS
37 44	74148	000002	9 7 9 7 1	THE PARTY AND THE REPORT OF A REPORT OF
05	14776	012746	MOV	#036002,=(SP) ;PUSH =12.01501675 *********
30	1-110	036602	5 J S 🖉 🔻	Lenner (1997) Lenner Martin Mithellen (1997) Lenner Martin Mithellen (1997)
96	15002		MOV	#141100, -(SP)
50	TORAL	141100	• • *** * •	ার্জনাক্রান্ট্রিসাটি বিশ্ববিদ্যালয় বিশ্ববিদ্যালয় বিশ্ববিদ্যালয় বিশ্ববিদ্যালয় বিশ্ববিদ্যালয় বিশ্ববিদ্যালয় ব
		▲ ○ ★ 부 환 평		

 $(\)$

 $\left(\right)$

 $\left(\right)$

ø

di.

(

÷£

		010146		MOV	R1,=(SP)	
		010046		MOV	RØ,=(SP)	· · · · · · · · · · · · · · · · · · ·
99 1	5012	012740		MOV	#071571, - (SP)	;PUSH 601,8042667 ************
		071571				
100	5016	012746		MOV	#042426,-(SP)	
		042420				
101	5022	012746		MOV	#056133, = (SP)	;PUSH 60.0901907 *********
s		056133				
102	5026	012746		MOV	#041560,=(SP)	
		041560				
103	5032	010146		MOV	R1,=(SP)	
104	5034	010040		MOV	RØ,=(SP)	
105	5036	010146		MOV	R1,=(SP)	
4	5040	010046		MOV	R0, . (SP)	
		000134		JMP	@(R4)+	
108				. ENDC		
109			1	38 · • · · · · · · · · ·		
110			•	, IFDF	FPU	
111				SETD	1	DOUBLE PRECISION ARGUMENT REDUCT
112				SETI	1	SHORT INTEGERS
113				MOV	#FC0\$20, R0;	POINTER TO CONSTANTS
114				LOCFD	0R4, F21	GET ARGUMENT
115			1997 - 19	MODD	(RØ)*,F21	F2=FRACT(X*LOG2(E))
116				STCOI	F3,R4;	R4=INT (X*LOG2(E))
117				LDD	#1,0,F0;	F0=1.0
118				DIVD	(RØ)+,F21	Y=F2/(2+L0G2(E))
119				SETF		t man a mandra and an a san a san a san a an a a san a s
120				LDCOP	F2,F21	REST IN SINGLE PRECISION
121				CFCC		TEST FOR UNDERFLOW
122				BEQ	SC15201	APPROXIMATION RESULT IS 1.0
				LDF	F2,F3;	HERITATION HAIR DEVENT
123				MULF	F3,F31	Y + Y
124 125				ADDF	(RØ)+,F31	B1+Y+Y
125				LOF	(RØ)+,F17	
127				DIVF	F3,F1;	A1/(B1+Y*Y)
				ADDF	F2,F1;	MAR WATE SI
128				ADDF	(RØ)+,F11	AØ+Y+A1/(B1+Y+Y)
129					F1,F2;	Y/(A0+Y+A1/(B1+Y+Y))
130				DIVF		IN CWRALLAWIN CRITICALI
131				MULF Subf	#2,0,F21 F2,F0;	1=2+Y/
132						
133			SC1520:	MULF	FØ,FØ; FØ,-(SP)/	(1=2*Y/)**2 Move approximation to stack
134			3613661		F0/4(3F)/	HAAR REFUGATION TO ALVON
135				.ENDC		
136			1	TENAD	E D.I	
137			700000	.IFNDF	FPU (PP)	PELICH AFART ADD
	0044	022626	ZFR\$201	CMP	(SP)+, (SP)+	JFLUSH CFRACT ARG
139			;		RESULT	15 1.
140				. ENDC		
		012600	SCLS201	MOV	(SP)+,R01	GET APPROXIMATION RESULT
142	5050	012601		MOV	(SP)+,R1/	
				. IFNDF	FPU	
143						GET INT(X*LOG2(E))
144	5052	012604		MOV	(SP) +, R41	where with the same water a result of
144 145				. ENDC		
144 145 146	5054	000304		.ENDC Swab	R41	MAKE INTO EXPONENT MODIFIER
144 145 146	5054			.ENDC Swab Clrb	R4; R4;	
144 145 146 147 148	5054 5056 5060	000304		.ENDC Swab	R41	

101

SEXP04 MACRO VR04=14 07=SEP=72 11143 PAGE 25+

							met une Alteres - autor for energies and
150		100405		BMI	OVR\$201		TEST OVERFLOW
151	5066	000205		RTS	R51		
152			-)				
153	5070	005001	ONES201	CLR	R1		
154	5072	012700		MOV	#40200,F	20	JEXP(TINY) = 1.
-		040200					
155	5076	000205		RTS	R5		
156	5100	012700	0VR\$201	MOV	#2404, RI	7	JERROR 4,5
100	0100	002404	આ મુખ્ય છે. આ સાથ મ	· · · • • •			
157	5104	000402		BR	ECL320		
	*		ZEN\$201	MOV	#2405,R	7	FERROR 5,5
158	5106	012700	45.0059 s	- CIQ Y	******	V	training the second
	F . 4 .	002405		tes	05 5500		
159	5112	004567	ECL\$201	JSR	R5, SERR	A	
a 1875 i.u.		004700		6 1 8	na	IRETURN	6 (A)
160	5116	005000		CLR	RØ	IREIURN	0
161	5120	005001		CLR	R1		
162	5122	000205		RTS	R 5		
163			1				
164				.IFDF	FPU		
165		*	the second second	ORDER-DE	EPENDENT	CONSTANT	r S
166			j				
167			FCUS201	, WORD	040270,	1250731	LOG2(E) DOUBLE PRECISION
168				, WORD	024534,6	0137611	
169			1				
170				WORD	040470,	1250731	2+LOG2(E) DOUBLE PRECISION
171				WORD	024534,0	013761;	
172			5				
173				WORD	041560.0	3561331	B1=60.0901907
174			1		****		
			•	.WORD	042426,0	715711	A1=601,8042667
175				* VAUN	8484491	******	ास्त्री ः स्थान्तरं मित्री होत् हो स्थान्त्र स्थानसंस्थान्त्र
176			,		141100,0	336609:	A0==12.01501675
177				.WORD	14114011	UNDUUE)	WRIE WERE ANTALA
178				ENDC			
179				. ENDC			

SFC	L02 N	ACRO VR	804=14 07	7=SEP=72	11:43 P/	VGE 26			105
1				.TITLE	SFCL02				
2				IFOF	CN0521				
3			1						
4			1	SFCALL	V002A				
5			1				WENT COO	DODATION	
5		e stragar.	COPYRI	IGHT 197	1, 019117	IL ENVIP	MENT CUR	PORALLUN,	MAYNARD, MAS
7 ·			1	CI (10)	SFCALL				
5			.,	.GLOBL	SPUACE	NE FOR	CALLING	SINGLE ARG	FORTRAN
0			2					TRAN FUNCT	
11			1		SEQUENCE				
2			,	MOV	ARG POIN				
3			,	MOV	HFUNCTIC		R4		
4			,	JSR	PC, SFCAL				
5			;		ROUMENT		t e e le port		
6		000000	•	RUEXO					
7		000004		R48%4					
8		000005		R5=%5					
9		000006		SP=%6					
20	15124	012740		MOV	#RET\$21	, - (SP)	PUSH S	FCALL RETU	RN
		015146	t i i i i i i i i i i i i i i i i i i i			f 24 -		A A	
21	15130	012746		MOV	#137#=(8	3M)	JJMP	PPC	
		000137						105	
		010546		MOV	R5,=(SP;		I WORD	ARG	
23	15136	012746		MQV	#401/=(577	1BR	.+4	
		000401		Mark 14	60 n5		IJSR	R5, FUNCT	
		010005		MOV	SP,R5		JJSK	NOFFUNCT	
		004014	DOTODI	JSR	RØ, ØR4	1611164	CALL		
26	10146		RET\$211	ADD	#8.,5P	IFLUSH			
. 7		000010		JMP	@(SP)+	PETION	TO USER	WITH ARG	ON STACK
7	10102	000136	en de la composition de la composition En esta de la composition de la composit	W PUP	. ₩. L OPA [™]		ICT (ARG)		
28			•	.ENDC		HAR FUR	1941 S 111 M 2	417 THY # 4.	

¢

SFIX03 MACRO VR04=14 07=SEP=72 11:43 PAGE 27

123			•	.TITLE .IFDF	SFIXØ3 CNDS22		en e		
4			1	IFIX	V003A			an Maria ang ang ang ang ang ang ang ang ang an	4
5 6 7			COPYR1	GHT 1971	1972 DIGIT	AL EQUIPH	ENT: CORPORAT:	ION, MAYNARD,	MAS
8 9			n Na Sanan		IFIX/SRI/SP Ran IFIX Fu		an a		1
10			14 - 27 1	CALLING JSR	SEQUENCE: R5, IFIX	and the second			÷
12			1	BR	A				
13			/ / A 1	WORD	ARGUMENT AD	DRESS			
15 16			1		THE TRUNCAT	and contraction of the			
17			1	ARQUILIN	I NO NU TUIC	DEN TU NO		international de la companya de la c La companya de la comp	
18		000000		R0=%0					
19		000004		R4=%4					
20		000000		R5=%5					
21 22		0000006	ተሮግኑባ	SP=%6 MOV	2(R5),R4	ICET	ARG ADDRESS		
66	10194	000002	4F # ^ #	rių y	CINNINA	. / 9-1	AND ADDREDS		
23	15160			MOV	2(R4), = (SP)	PUS	HARG		
		000002							
	15164			Mav	#R4,=(SP)				
25			RND\$221	JSR	R4, \$POLSH	7ENT	ER POLISH MOI	DE Constant Anna	
		004452		1 5 1 5 1 5					
26		0176501		. WORD	SRI, UPL522	THE FIRM	NCATE AND FI	Xen el contra de entre de la contra de la co	
		0151761		44 - A 14	AGANA BA	• • •	TUREADA ADA	1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	
27			UPLS221		(SP)+,RØ		INTEGER RESI Aller	YL I Charles de la Const	
28 29	12500	000205	· · · · ·	RTS .ENDC	R5 JRE				

SFLT02 MACRO VR04=1	4 07=SEP=72 11143	PAGE 28 105
1 2 3	.TITLE SFLT02 .IFDF CND\$23	
4 1	FLOAT V002A	
5 5 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	PYRIGHT 1971, DIGI	TAL EQUIPMENT CORPORATION, MAYNARD, MASS
8 9 1	,GLOBL FLOAT, Float the fo Calling sequen	SIR, SPOLSH, SPOPR3 RTRAN FLOAT FUNCTION
10 / 11 / 12 /	JSR R5,FLO	
13 / 14 /Ai		S UF INTEGER
15 <i>J</i> 16 <i>J</i>	RETURNS REAL EU USES SIR.	DUIVALENT IN RØ AND R1.
17 7 18 000000	R0=%0 R1=%1	
19 000001 20 000004 21 000005	R45%4 R5#%5	
22 000006 23 15202 017546 FLUA	5P=%6	- (SP) JGET ARGUMENT ON STACK
000002 24 15206 004467	JSR R4, SPOI	
004432 25 15212 0160621	WORD SIR	FCALL SIR TO CONVERT TO REAL
26 15214 017610 27 15216 015220	WORD SPOPRS WORD UPLS23	POP RESULT TO REGS
28 15220 000205 UPLS		FRETURN TO CALLER

ç

SICIU2 MACRO VR04=14 07=SEP=72 11:43 PAGE 29

1				TITLE	SICI02
2				.IFOF	CND\$24
3			1	SICI	V002A
5			3	****	
6			COPYRI	GHT 1971	,1972 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MAS
7			1		
8				,GLOBL	SICI, SOCI
8 9			1	SOCI	ASCIL TO OCTAL CONVERSION
10)	SICI	ASCII TO INTEGER CONVERSION
11			1		SEQUENCE
12			1	PUSH	CHARACTER FIELD START
13			I.	PUSH	CHARACTER FIELD LENGTH PC,SICI OR BOCI
14			1	JSR Returns	a second a second a second de la
15 16		000000	· ·	ROEXO	and to the manual statements
17		000001		R1=%1	
18		000002		R2=%2	
19		000000		SP=%6	
20		000007		PC=%7	
	15222	012745	\$061:	MOV	#67, = (SP) /SET OCTAL FLAGS
		000067	· · · · · ·	Ph. se	and a strange of the
22	15226	000402	a w f1 + 4	BR	GU\$24 #471/=(SP) /SET DECIMAL FLAGS
23	19290	012746	21611	MOV	#471,=(SP) /SET DECIMAL FLAGS
24	15234	000471	G0\$24:	MOV	R1,=(SP) /SAVE R1
25	15236		90 46 48	MOV	8. (SP), R1 JGET STRING START
20	10500	000010			
26	15242	056566		ADD	6(SP),8,(SP) / JGET END+1
	10 10 10 10 10 10 10 10 10 10 10 10 10 1	000006			
		000010			
27	15250	016666		MOV	4(SP),6(SP) IFIDDLE RETURN POINTER
		000004			
		000000			
28	15256	010055		Mav	RU, 4(SP) ISAVE RO
- 0	. 155 m 165 m	000004		54 (* 14	R2,=(SP) / SAVE R2
29		010246		MOV CLR	R2,=(SP) ISAVE R2 =(SP) ICLEAR SIGN
	15264	005046 005000		CLR	RØ JELEAR WORK SPACE
			STT\$241	MOVB	(R1)+,R2 /GET NEXT CHAR.
		042702		BIC	#177600, R2
~ ~	ang ing mangar ang	177600			
34	15276	120227		CMPB	R2,#1 1
	-	000040			
		001004		BNE	SGSS24 JJUMP IF NOT BLANK
36	15304	020166		CMP	R1,12,(SP)
.		000014		0. •	STT\$24 JJUMP IF MORE TO SCAN
		002767		BLT	STT\$24 JJUMP IF MORE TO SCAN SGN\$24 JDONE
38	10012	000454	SG53241	BR	7(SP); IF OCTAL CONVERSION
33	10014	103/80	0043641	1010	র উপের রাজে বিজে বিজে বিজে বিজে বিজে বিজে বিজে বি
Δīλ	15320	001002		BNE	SN1524; DO NOT PERMIT SIGNS
		005216		INC	OSP; OCTAL - FAKE THE SIGN BIT
		000420		BR	NCK\$241 GO PROCESS THE DIGIT
			SN15241		R2,#1+1
• 🖛		000053			
44	15332	001441		BEQ	FLD\$24 JJUMP IF +
					•

SICI02 MACRO VR04=14 07=SEP=72 11:43 PAGE 29+

K.

÷

Ģ

ø

45	15334	120227		CMPB	R2,#!-	
		000055				
46	15340	001012		BNE	NCK\$24 JJUMP IF NOT -	
47	15342	005216		INC	PSP ISET SIGN .	
48		000434		BR	FLD\$24	a da an
			NXT5241		(R1)+,R2 JGET NEXT CHAR.	
		042702		BIC	#177000,R2	
00	10000	177600		~**	#1// #001.4	
F 4				MAB D	00 41 B	
ô1	10004	120227		CMPB	R2,#1 J	
		000040		trans		
52	15360	001002		BNE	NCKS24 JJUMP IF NOT BLANK	
53	15362	112702		MQVB	#60,R2 JBLANK #ZERO	
		000060				
54	15366	120227	NCKS241	CMPB	R2,410	
-		000060				
55	15370	002440		BLT	ERRS24 JJUMP IF TOO SMALL	
		120266		СМРВ	R2,6(SP)	
90	100/ 4	000000		write o	n m i O sol i	
K 49				0.07	сарьод тнима те тал ате .	
		003035		BGT	ERRS24 JJUMP IF TOO BIG	
58	19402	162702		SUB	#60,R2 IMAKE NUMERIC	
		000000				
59	15406	105766		TSTB	7(SP) FOCTAL OR BINARY	
		000007				
60	15412	001435		BEQ	OCL524	
61	15414	006300		ASL	RØ JRØ#BASE*RØ+R2	
		102426		BVS	ERR\$24	
		160002		SUB	RØ, RZ	
		006300		ASL	RØ	
		102423		BVS	ERR\$24	
		006300		ASL	RO	
		102421		BVS	ERR\$24	
		160200		SUB	R2, R0	
		102417		BVS	ERR\$24	
70	15436	020166	FLU5241	CMP	R1,12,(SP)	
		600014				
71	15442	002741		BLT	NXTS24 JJUMP IF MORE FIELD TO :	SCAN
72	15444	006026	SGN\$241	<i>p</i>	(SP) + ITEST SIGN	
		103403		BCS	DNES24 JJUMP IF -	
		005400		NEG	RØ IMAKE +	
		102411		BVS	NGMS24 JJUMP IF -NEGMAX	
		000241	0.1.5 0 0 4 s	CLC	ISET SUCCESS FLAG	
			DNES241		(SP)+,R2 IRESTORE R2	
		012601		MOV	(SP)+,R1 /RESTORE R1	
		006126		ROL	(SP) + JFLUSH FLAG AND SET C B)	IT IF ERROR
80	15464	010066		MOV	RØ,4(SP) JRETURN RESULT	
		000004				
81	15470	012500		MOV	(SP)+,RØ	
		000207		RTS	PC	
			ERK\$241	TST	(SP) + JFLUSH SIGN	
					RØ	
		005166	91 0 110 069		4(SP) /SET ERROR FLAG	
00	10360			COM	ALVEY FORE CARVA PLAN	
n 4		000004		0.0	ONE AD A	
86	10504	000764		BR	DNE\$24	
87			1	61 m 1		
			OCL\$24:		RO; SHIFT 3 BITS LEFT,	
		103771		BCS	ERR\$247 CHECKING AS YOU GO	
90	15512	006100		ROL	RØT	

107

SICI02 MACRO VR04-14 07-SEP-72 11:43 PAGE 29+

91	15514	103767	BCS	ERRS241		
92	15516	006100	ROL	RØJ		
		103765	BCS	ERR\$24;	$e_{i} = e_{i} \chi_{i} \cdot e_{i} + e_{i} \chi_{i} \cdot e_{i} = e_{i} \chi_{i} \cdot e_{i} + e_{i} \chi_{i} $	2
94	15522	060200	ADD	R2, RØ;	ADD IN THE	DIGIT
95	15524	000744	BR	FL03241	DO NEXT	
96	-	14°	. ENDC			

the second se

ŧċ,

Set.5

1

÷.,

61C	002 M	ACRO VA	04-14 07	7-SEP=72	11143 PAGE 30	109	
		l Nasar ang P				And the second	
				.TITLE	SIC002		
2				. IFDF	CN0325		
			1	\$1C0	ASUDV		
))			1				
			; COPYR	IGHT 197	, DIGITAL EQUIP	MENT CORPORATION, MAYNARD, MA	ASS
5 7 9		te Arto	J. Anton				
5				GLOBL	SICO, SOCU		
0			1	\$000	DCTAL TO ASCII	CONVERSION	
1			1	\$100	INTEGER TO ASCI	I CONVERSION	
2			1	· · · · · · · · · · · · · · · · · · ·	SEQUENCE:	хаж т т () кі	
3				PUSH	FIELD START LOC FIELD LENGTH	ALLUN	
, 4			1	PUSH PUSH VAI			-
5			3	JSR	C,SICO (OR SOC	(0)	
,7			1	ERROR W	L RETURN WITH	C BIT SET ON	
8			1		R2, R3 ARE DEST	ROYED	
9	•	000000		RØ=%0	an an trainin an Arraige. An Arraige an Arraige a		
0		190900	a fearth is	R1#%1			
21		000002		R2=%2 R3=%3			
22		000003		R4=%4			
24	ž.	000000		SP=%6			
25		000007		PC=%7			-
26	15526	012700	SOCOI	MOV	HOCT\$25-REL\$25	RØ JPUINT TO OCTAL TA	BLE
	· ·	000166		_	n an		
27	15532			BR	GU\$25	RØ JPUINT TO DECIMAL	TAR
28	15534	012700	SICOI	MOV	#DEC\$25=REL\$25	TRD FFATIAL TO OCUTINE	
	15547	000154 010446	G0\$251	MOV	R4,=(SP)		
29 30		016003		MOV	8, (SP), R3	JGET FIELD START	
	1	000010					
31	15546	016602		MOV	6.(SP),R2	IGET FIELD LENGTH	
	i in the	000006	254			IF LENGTH NOT NEG	
		002003		BGE	-	The Private Hol Hea	
		005002 005066		CLR CLR	R2 6(SP)		
24	10000	000000		νer μ _μ ε τ			
35	15562		LP55251	MOV	4, (SP), R4	JGET VALUE TO BE CONVERTED	
	•	000004					
36	15556	012746		MOV	#1 ,=(SP)	ICLEAR SIGN	
	. 5 11 7 0	000040		C M D	RØ,#UCTS25-RELS	\$25 JCHECK IF DOING OC	TAL
37	100/2	020027 000166		CMP	RUJAVLIJAJAKEL		
18	15576	001405		BEQ	PUSS25 IYES, (GIVE MAGNITUDE RESULT	
		005704		TST	R4		
		002003		BGE	POSS25 IJUMP		
41	15604	005404		NEG		BSOLUTE VALUE	
42	15606	012716		MOV	#1-, PSP ISAVE		
4 79		000055	DASCART	r. 0	-(SP) ISET F	ENCE	
		005040 060700	P055251	ADD	PČ,RŰ		
	15616	0-0100	RELS251	n w W	• 		
46	15616	005710	TST\$251	TST	e R Ø		
47	15620	001416	· · · · · · ·	BEQ	MOVS25 JJUMP	IF ALL POWERS DONE	
48	15622	005001		CLR	R1		

(

Ģ

SIC002 MACRO VR04=14 07=SEP=72 11143 PAGE 30+

49 15624 161004 SUBS251 PRØ,R4 ISEE IF CURRENT POWER WILL GO AGAIN SUB BAC\$25 50 15626 103402 BLO 51 15630 005201 INC R1 JBUMP DIGIT BR SUB\$25 52 15632 000774 ITOD MUCH, BACK UP (RØ)+,R4 53 15634 062004 BAC\$251 ADD TST R1 54 15636 005701 JUMP IF DIGIT NOT Ø 55 15640 001002 BNE NZES25 56 15642 005716 TST PSP 57 15644 001764 BEQ TSTS25 JJUMP IF NO NON-ZERO DIGITS YET #60,R1 58 15646 062701 NZES251 ADD ICONVERT TO ASCII 000060 MOV R1,=(SP) 59 15652 010146 TST525 60 15654 000760 BR **JPOINT TO FIELD END** 61 15656 060203 MOV\$251 ADD R2, R3 #60,R4 ICONVERT LEAST SIGNIFICANT DIGIT ADD 62 15660 062704 000000 63 15664 110443 MOVB R4, ... (R3) 64 15666 005302 DCRs25: DEC R2 FULS25 JJUMP IF COUNT EXHAUSTED 65 15670 003410 BLE IMOVE DIGIT (SP)+,=(R3) MOVE 66 15672 112043 DCRS25 JJUMP IF NOT THE FENCE BNE 67 15674 001374 (SP)+, 0R3 IMOVE OUT THE SIGN 68 15676 112613 MOVB 69 15700 005302 FILS251 DEC R2 ONES25 JJUMP IF FIELD FILLED 70 15702 001410 BEQ #1 ,=(R3) 71 15704 112743 MOVE IMOVE IN LEADING BLANKS 000040 72 15710 000773 BR F1L\$25 73 15712 005726 FULS25: TST (SP)+ ERR\$25 INUMBER TOO BIG FOR FIELD 74 15714 001011 BNE #1 , (SP)+ 75 15716 022726 CMP 000040 BNE STS525-4. JUMP IF NO ROOM FOR -76 15722 001011 (SP)+,R4 77 15724 012604 DNES251 MOV 78 15726 012666 MOV (SP)+,4(SP) IMOVE RETURN UP 000004 (SP)+ IFLUSH VALUE 79 15732 005728 TST IFLUSH FLAG AND SET C BIT ON IF ERROR (SP)+ 80 15734 006126 ROL 81 15736 000207 RTS PC (SP) + 82 15740 005726 ERR\$251 TST ERR\$25 83 15742 001376 BNE (SP) + **IFLUSH SIGN** TST 84 15744 005726 8. (SP), R3 85 15746 016603 MOV 000010 JFILL FIELD WITH * 86 15752 112723 ST\$\$251 MOVE #1+, (R3)+ 000052 DEC 6(SP) 87 15756 005366 000000 IJUMP IF MORE TO DO BGT STS525 88 15762 003373 IFLAG ERROR 6(SP) COM 89 15764 005166 000000 DNES25 90 15770 000755 BR 91 15772 023420 DEC\$251 .WORD 10000,,1000,,100,,10,,0 15774 001750 15776 000144 16000 000012 10002 000000 100000,10000,1000,100,10,0 92 16004 100000 OCT\$251 .WORD

110

1

-
///

16	500	6	Ø	1	Ø	Ø	Ø	0
10	01	Ø	Û	Ø	1	Ø	Ø	Ø
10	01	2	Ø	Ø	Ø	1	Ø	ð
10	501	4	Ø	Ø	Ø	Ø	1	Ø
10	501	6	0	Ø	Ø	Ø	Ø	Ø

• ENOC

.

93

2.

Ģ

SINT02 MACRO VR04=14 07=SEP=72 11143 PAGE 31

1				TITLE	SINTU2		
¥.				.IFOF	CND\$26		
3			1	INT	V002A		
5			1				
6			J COMYR.	IGHT 197	1, DIGITAL EQUIP	MENI CURPURATI	ON, MAYNARD, MASS
8			,	GLOBL	INT, IDINT, SRI, S	POLSH	,
9			1 .	THE FOR	TRAN INT AND IDI	NT FUNCTIONS,	
10			1		SEQUENCE		
11			1	JSR	R5, INT (OR IDI	NT)	
12			i -	BR	A	· · · · ·	
13			i i		ARGUMENT ADDRES	S	
14			JAI			-	
15			2	RETURNS	INTEGER EQUIVAL	ENT IN RØ.	
16			1	USES SR			
17			i i	*****	- •		
18		000000	•	R0=%0			
19		000004		R48%4			
20		000005		R5#%5			
21		0000006		SPEXO			
	16020	0000000	INTE	WF TAU			
		016504	IDÍNTI	MOV	2(R5),R4	JGET ARGUMENT	ADDRESS
£ Ų	140.00	000002	₩₩ <i>₩</i> 1819		E TOOL LEA		La Maria de Maria
24	16024			MOV	2(R4),=(\$P)	PUSH LOW ORC	FR PEAL PART
64	10004	000002		19 9 4	C (UM/) = (YP)		
06	16030			MOV	@R4,=(SP)	HIGH ORDER	
	79	004467		JSR	R4, SPOLSH	ICALL SRI TO	CONVERT TO
26	10005			M SK	RAIDULOU		
		003606		unan	EGT IN EQS	INTEGER	
2/		017650		.WORD	SRI, UPL526	Faulence.	
	16040			Nov	ARDA DA		25 CHL #
	10042		UPL\$261		(SP)+,R0	POP INTEGER	REQUEI
29	16044	000205		RTS	R 5		
30				. ENDC			

IR	84 M	ACRO VE	804-14 0	7=SEP=72	11:43 PA	GE 32			/13	
				n an an an Araba Ar			i ya			
				.TITLE	SIR04	·	· · ·			
2				. IFOF	CND\$27					
5				,GLOBL	SIR,SID					
l .			1	INTEGER	TO REAL	CONVERS	JUNA			
5 5 7 4			1	SIR	V004A					
0		000000))))))	CONVERT	IS A FU	AL FORM	D ON THE	IPMENT COR TOP OF THE Eturn IT A	P., MAYNARD Stack S the top	. , М
3		000001		R1=%1						
4		000002		R2#%2						
5		000003		R3=%3						
6 7		000004		848%4 588%6						
/ 8		000000		MQ#17730	A A					
9		177312		NOR = 1773			14			
0		000000	and the second	F0=%0						
1		~~~~~		. IFDF	FPU		1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -			
2		,	SIUI	SETDI						12.3
3				BR	IDI\$27					
4			SIRI	SETF	1				• • • •	
5			IDIS271	SETI	1		SHORT I	NTEGERS		
6				LOCIF	(SP)+,F0	1 .	CONVERT			
7				STF	F0,=(SP)	1	RESULT	TO STACK		
8				JMP	@(R4)+					
9				. ENDC						
0				, IFNDF	FPU		- milinii	an. P≏iitskiaersionas esaña		
· •	16046		SIDI	MOV	esp,=(SP		FUSH A	RGUMENT DO	WN	
		011646		MOV	PSP, *(SP	I. SALENA	I OWEST O	RDER DOUBL	c	
3	10002	005066		CLR	2(SP)	ILLEAR	LONESI U	RUEN VOUDL	5	
		000002		CLR	4 (SP)					
4	10000	000004		Set light 1.5	A CAL Y					
15	16062	005046	STRE	CLR	=(SP)	IMAKE F	ROOM FOR	RESULT		
		016601		MOV	2(SP),R1		JGET IN	TEGER ARGU	MENT	
-		000002								
57	16070	003002		BGT	PUSS27					
		001424		BEQ	ZER\$27					
		005401		NEG	R1		BSOLUTE V	ALUE		
	16076	006146	P055271		= (SP)	ISAVE S	DIGN			
12	16100	012702		.IFNDF Mov	EAE #220;R2	IGET M	AX. POSSI	BLE EXPONE	NT +1	
		000220								
13				ENDC	5					
44			1	EAE CODI	EAE					
45 46				MOV	4217.00	IGET M	AX. POSST	BLE EXPONE	NT	
47				ENDC	17 M & F F FL M	e an an i ste	িলায় সেলাসায়	अस्य प्रसार अस्य राष्ट्र पर रष्ट्रीयस्य		
	16104	105066 000004		CLRB	4(SP)	ICLEAR	LOWEST O	RDER FRACT	ION	
19	16110	0000 01 4	NOM\$271							
50	****		17 M 1 . W. W. P. F.	. IFNDF	EAE					
				(A) A A A A A A A A A A A A A A A A A A						
	16110	006101		KOL	R1		FOR NORMA IF FOUND	LBIT		

€

Þ

SIR04 MACRO VR04=14 07=SEP=72 11143 PAGE 32+

53	16114	005302		DEC	R2	IDECREASE EXPONENT
54	10116	000774		BR	NOMS27	ITRY AGAIN
55				.ENDC		
56			1	EAE CODE		
57			•		EAE	
58					#MQ,R3	POINT TO MO
59				CLR	0R3	en entre for scola
60						
61					R1,=(R3)	
62					#NOR RO	
63				CLR	erø	INORMALIZE FRACTION
				SUB	(RØ)+,R2	
64						ISHIFT OUT NORMAL BIT BY LSH
65					PR3, K1	IRESULT TO R1
66	· · · · · ·			. ENDC		
67	16120	110166 000005	N005271	MOVB	R1,5(SP)) ISAVE LOW ORDER FRACTION
68	16124			CLRB	R1	
69		150201				COMBINE EXPONENT AND HIGH ORDER FRACTIO
70	16130				R1	ICOMBINE EXPONENT AND HIGH ORDER FRACTIO
71	16132	006026				JGET SIGN
72	16134	006001				FINSERT SIGN IN RESULT
		106066			3(SP)	LINDERI STAN IN VESOFI
	14140	000003		NUND	01077	
7 4	10142			MOV	01 450	
	16144		7500000			JOUTPUT RESULT
	10144	000134	ZER\$271		@(R4)+	
76			÷	.ENDC		
77		and the second		.ENDC		

Y Kaji

۲.

Ć

P

1 2 3 4			1	.TITLE .IFOF .GLOBL \$MLD	SMLD05 CND\$28 SMLD/SERRA THE DOUBLE MULTIPLY ROUTINE
5	<i>,</i>			SMLD	V005A
7 8 9			/ 7 7	CALLED	GHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. In Polish mode.
10			1.000	REPLACE	ES THE TOP TWO DOUBLES ON THE STACK
11			1 .	WITH TH	HEIR PRODUCT.
12		000000		R0=%0	
13		000001		R1=%1	
14		000002		R2=%2	
15		000005		R3=%3	
16		000004		R4=%4	
17		000005		R5=%5	
18		0000006		5P=%6	
19		000007		PC=%7	and the second
20		177304		MQ=1773	304
21		000010		A=8,	
22		000020		B=16.	
23		000014		RESLITE1	
24		000002		SIGN=2	
25		000000		F0=%0	
26				. IFOF	FPU AND A DESCRIPTION OF A DESCRIPTION OF A DESCRIPTION OF A DESCRIPTION O
27			SMLDI	.WORD	170011 JISETD
28	<i>0</i>			.WORD	172426 IJLDD (SP)+,FØ JGET OPERAND
29				, WORD	171026 11MULD (SP)+,FØ 1PRODUCT
30			- 4 - ¹	WORD	174046 IISTD F0,-(SP) IPRODUCT TO STAC
31				JMP	@(R4)+
32				ENDC	804
33				. IFNDF	
34	16146	010446	SMLDI	MOV	R4,=(SP)
35	16150	010546		MOV	R5,=(SP) A+0=4(SP) ;SHIFT MULTIPLICAND
36	16152	006366		ASL	A+0=4(SP) ;SHIFT MULTIPLICAND
AN AN		000004		0.01	(CON) SUEED STON
37	16156			ROL	=(SP) /KEEP SIGN =(SP) /CLEAR EXPONENT
	16160	005046		CLR	
39	16162	116615		MOVB	A+1(SP), #SP JKEEP MULTIPLICAND EXPONENT
4.3		000011		8en -	ZER528 JJUMP IF ANSWER IS ZERD
	16106	001436		BEQ	ZER\$28 JJUMP IF ANSWER IS ZERO A(SP),A+1(SP) JSHIFT FRACTION LEFT
41	101/0	116565		MOVB	ALOPYTAATION JOUTEL LUNALION PELL
		000010 000011			
4.0	18175			SEC	FINSERT NURMAL BIT
	16176	000261		ROR	A (2 B)
43	10500	000010		nyņ.	
	45004			Mova	A+3(SP), A(SP)
44	10204	116666		MOVB	MARY AND ARY STREET
		000010			
A 14	16010			SWAB	A+2(\$P)
40	16212	000012		. # A A A	n na 1999 i 🗰 🖗 grafi 🗶 an an an an ann an an ann an ann an ann
46	16216	116666		MOVB	A+5(SP), A+2(SP)
	****0	000015		ः भूत र भ्य	1917年後期1月1日第二日の1月1日には、1917年後期1日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年年月日には、1917年月月日には、1917年月月日日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月日には、1917年月月月日に日には、1917年月月日に日に日日に日日に日に日に日日日に日日日に日に日に日日日に日日日に日に日日に日日日に日日に日日に日
		000012			
47	16224	000366		SWAB	A+4(5P)
~ *	*	****		ज्ज्ञ न १२०५ २	

SMLD05 MACRO VR04=14 07=SEP=72 11:43 PAGE 33+

		000014		e e	
48	10230	116666		MOVB	A+7 (SP), A+4 (SP)
		000017			and a second
		000014		and the second second	a de la companya de l A companya de la comp
49	16236	000365		SWAB	A+6(\$P)
	÷	000016			
50	16242	105066		CLRB.	A+6(SP) IMAKE ROOM FOR EXTRA BITS
-		000016			
61	16246	006366		ASL	B(SP) /SHIFT HIGH MULTIPLIER
¥ 4	1 4 6 7 V	000020			
80	16050			ADC	SIGN(SP) JGET PRODUCT SIGN
25	10505	005566		ADC	atanial) tart undanal arau
		000002			
53	16256	105768		TSTB	B+1(SP)
		000021			
54	15262	001003		BNE	NNZSZ8 JJUMP IF NOT ZERO
55	16264	022626	ZER\$281	CMP	(SP)+, (SP)+ JFLUSH SIGN AND EXPONENT
		000167	ZE1\$281	JMP	ZE2\$28
	* * ** *	000334		* *	
57	16070	005000	NNZ5281	CIR	RØ ICLEAR PRODUCT
				CLR	R1
58	10514	005001			
59				. IFNDF	EAE&MULDIV
60		005002		CLR	R2
61	10300	005003		CLR	R3
62	16302	005005		CLR	R5 ICLEAR C BIT OVERFLOW CATCHER
63	16304	006066		ROR	B(SP) /SIGN IS NOW Ø
,		000020			
64	16310	012746		MQV	#16./=(SP) //PUSH ITERATION COUNT
		000020			이 같은 것은 사람이 있는 것을 수 있는 것이 있는 것이 있는 것이 있는 것이 있는 것이 있다. 이 가지 않는 것이 있는 것 같은 것이 같은 것이 있는 것이 같은 것이 있는 것이 있는 것이 있는 것이 있는 것이 있는 것이 있는 것이 없는 것
65	15314	016604	• · ·	MOV	B+6+2(SP),R4 JGET LOWEST ORDER MULTIPLER
03	10014			1101	्रिक्रिल ६४०१.२१९२२ २०२१ स्वतः स्वतः अ स्टानस्य न्यानस्य राजस्य माञ्चलाः । स्वतः
		000030		Den	B6ZS28 JJUMP IF NO BITS HERE
		001404		BEQ	
67	16322	004767		JSR	PC, MT0\$28
		000410			
68	16326	012716		MOV	#16, / OSP / RESTORE COUNT
		000020			
69	16332	016604	8625281	MOV	B+4+2(SP),R4 JGET NEXT LOWEST FRACTION
	-	000026			
70	16336	001003		BNE	B4N528 JJUMP IF WORK TO DU
		005766		TST	B+6+2(SP)
ſ .	****	000030			
				SEA	B4Z\$28 JJUMP IF NO PRODUCT YET
12	10344	001406		BEQ	
73	10346		84N\$281	JSR	PC, MT2\$28
		000360			na sense de la companya de la compa Nota de la companya d
74	16352	004767		JSR	PC, MLTS28 JONE BIT FULL PRECISION
		000262			
75	16356	012716		MOV	#16, /@SP
		000020			이 같은 것 같은
76	16362	016504	8425281	MOV	B+2+2(SP),R4 JGET NEXT TO HIGHEST ORDER FRACT
	1+0+6	000024			
99	1 KIRE	001005		BNE	B2N\$28
					B+4+2(SP)
/ 0	10010	005766		TST	Dia managera de la companya de la co
-		000026		D	0.01/eDe
		001003		BNE	B2N\$28
80	16376	005766		TST	B+6+2(\$P)
		000030			
81	15402	001402		BEQ	B2Z\$28
82	16404	004767	82N5281		PC,MLTS28

116

\$MLD05 MACRU VR04=14 07=SEP=72 11:43 PAGE 33+

む

(

·(?--

Þ:

		000230	0.070000	MON	8+0+2(SP)	۵.	10FT	нтон	noneo	Q 7 7 Q
63	10410		0249401	PIQ V	DTUTE(OF)	1.12.14	10-1	0100	WR.WER	0113
94		000022		MOV	#7,05P 1	THERE	ARE	IV SF	VEN DE	THEM
84	10414	000007			HIJEWE I	1 FT bes 13 bes			r San Lat. and S	8 3,7 540 7 1
	16100	0000007		JSR	PC,MLTS28					
00	10440	000214		2 3 N	LANDAIAC.					
86	16121	604757		JSR	PC, MT1528		160 D	O THE	NORMA	I STT
60	10464	000214		Q 0 (c	e âl cei îne-	,		, and a second second		
87	16430	005726		TST	(SP)+ 1	FLUSH	ITERAT	IDN C	OUNT	
		062604		ADD	(SP)+,R4		TADD	EXPUN	ENTS	
89	4 7 7 7 6			ENDC	, , ,		• • • •			
90				IFDF	EAEIMULDI	V.				
91				CLR	R4					
92				BISB	8+1 (SP), R			EXPUN		
93				ADD	R4,05P 1					
94				MOVB		')	/ INSE	RT NO	RMAL B	IT
95				ROR	B(SP)		1			
96				SWAB		LEFT J	USTIFY	FRAC	TION	
97				MOVB	B+3(SP),8					
98				SWAB	B+2(SP)					
99					B+5(5P),8	+2(SP)				
100			-	SWAB	B+4(5P)					
101				MOVA	8+7 (SP),8	+4(5P)				
102				SWAB	8+6(5P)					
103					8+6(\$P)					
104				ENDC						
105				.IFDF		(57) # 1 1 #	10 M A			
106				MOV	NMQ, K4 J		ru mu			
107			<i>11</i>		A(SP),=(S B+6+2(SP)	4 D A				
108				MOV JSR	R5,EMUS20		1961	WT AD N		
109 110				MOV	(SP)+,R2		RESI		PRODU	CT
111				MOV	(SP)+,R3		1			Y
112				MOV	A+2(SP),=	(SP)				
113				MOV	B+4+2(SP)		IGET	A2+83		
114				JSR	R5, EMUS28					
115					(SP)+,R2		ADD	TO PR	ODUCT	
116				ADC	RI					
117				ADD	(SP)+,R3					
118				ADC	R2					
119				ADC	R1					
120				MQV	A+4(SP),-	(SP)				
121				MOV	B+2+2(SP)	, OR4	JGET	A3+85		
122				JSR	R5, EMUS28					
123				ADD	(SP) + , R2	$\frac{1}{2} = 1 + 1$				
124				ADC	R1					
125				ADD	(SP) + , R3					
126				AOÇ	R2					
127				ADC	R1					
128		1997 - 19		MOV	A+6(SP),"					
129				MOV	8+0+2(SP)		JGLT	A4+B1		
130				JSR	R5,EMUS28					
131				ADD	(SP)+,R2					
132				ADC	R1					
133				ADD	(SP)+,R3-					
134				ADC	R2					

135		ADC	R1	
130		MOV		BY 2**16
137		MOV	R1,R2	
138		CLR	R1	
139		MOV	A(SP),=(38)	
140		MOV	B+4+2(SP),0R4	1GET A1+83
141		JSR	R5, EMUS28	
142		ADD	(SP)+,R2	
143		ADC	R1	
10 -		ADD		
144			(SP)+,R3	
145		ADC	R2	
140		ADC		
147		MOV	A+2(SP), = (SP)	and an in the
148		MOV	B+2+2(SP),0R4	JGET A2+82
149		JSR	R5, EMUS28	4 - 1 - 4
150		ADD	(SP)+,R2	
151		ADC	R1	
152		ADD	(SP)+,R3	
153		ADC	R2	
154		ADC	R1	
155		MOV	A+4(SP),=(SP)	
156		MOV	B+0+2(SP),0R4	JGET A3+81 -
157		JSR	R5, EMUS28	
158		ADD	(SP)+,R2	
159		ADC	R1	
160		ADD	(SP)+,R3	
161		ADC	R2	
162		ADC	R1	
163		MOV	A(SP) = (SP)	The second se
164		MOV	B+2+2(SP), #R4	IGET A1+B2
165		JSR	R5,EMU\$28	
166		ADD	(SP)+,R1	
167		ADC	RØ	
168		ADŬ	(SP)+,R2	
169		ADC	R1	
170		ADC	RØ	
171		MOV	A+2(SP),=(SP)	
172		MOV	8+0+2(SP),0R4	JGET A2+B1
173		JSR	R5,EMUS28	, JEI / E;
174		ADD	(SP)+,R1	
175		ADC	RU	
176		ADD	(SP)+,R2	
177		ADC	R1	
178		ADC	RØ	
179		MOV	A(SP) = (SP)	
180		MOV	8+0+2(SP),0R4	JGET A1+B1
181		JSR	R5, EMU\$28	
182		ADD	(SP)+, R0	
183		ADD	(SP)+,R1	
184	1. 	ADC	RØ	
185		MOV	(SP)+,R4	JGET SUM OF EXPONENTS
186		.ENDC	an a	रूल्यान्ताः साम्रहताः अत्या समादितः दश्वराष्ठस्योते हेर्भवी
			MUL O TY	
187		.IFDF	MULDIV	
188		MOV	A(SP),=(SP)	and the said a
189		MOV	B+6+2(SP),R4	IGET A1+84
190		JSR	PC, EMUS28	
191		MOV	R4,R2 JRESULT	TO PRODUCT

118

*

#

æ

47

 \boldsymbol{v}

		14-14		
192		MOV	R5, R3	
193		MOV	A+2(SP),=(SP)	
194		MOV	B+4+2(SP),R4	1GET A2+83
195		JSR	PC, EMUS28	All and All an I have an
196		ADD	R4,R2 JADD TO	PRODUCT
197		ADC	R1	
198		ADD	R5, R3	
199		ADC	R2	
200		ADC	R1	
201		MOV	A+4(SP),=(SP)	
202		MOV	B+2+2(SP),R4	16ET A3+82
203	•	JSR	PC, EMUS28	
204		ADD	R4, R2	
205		ADC	R1	
206		ADD	R5, R3	
207		ADC	R2	
208		ADC	R1	
		MOV		
209			A+6(SP),=(SP)	1067 AA+84
210		MOV	8+0+2(SP),R4	JGET A4+B1
211		JSR	PC, EMUS28	
212		ADD	R4, R2	
213		ADC	R1	
214		ADD	R5,R3	
215		ADC	R2	
216		ADC	R1	
217		MOV	R2,R3 IDIVIDE	BY 2**16
218		MOV	R1,R2	
219		CLR	R1	
220		MOV	A(SP),=(SP)	
221	We get the second s	MOV	B+4+2(SP),R4	IGET A1+83
222		JSR	PC, EMUS28	
223		ADD	R4, R2	
224		ADC	R1	100 C
225		ADD	R5, R3	
226		ADC	R2	
227		ADC	R1	• . · · · · · · · · · · · · · · · · · ·
228		MOV	A+2(SP),=(SP)	
229		MOV	B+2+2(SP),R4	IGET A2+82
230		JSR	PC, EMUS28	14-1 767-6
		ADU	R4,R2	
231				
232		ADC	R1	 x x
233		ADD	R5,R3	
234		ADC	R2	
235		ADC	R1	
236		MOV	A+4(SP),=(SP)	
237		MOV	B+0+2(SP),R4	IGET A3+B1
238		JSR	PC, EMUS28	
239		ADD	R4, R2	
240		ADC	R1	
241		ADD	R5,R3	
242		ADC	R2	
243		ADC	R1	
244		MOV	A(SP),=(SP)	
245		MOV	B+2+2 (SP),R4	IGET A1+82
246		JSR	PC, EMUS28	
247		ADD	R4, R1	
248		ADC	RØ	
1994 - 1997 - 19		- * *		

				100	R5, R2	
249				ADD	R1	
250				ADC		
251				ADC	RØ	-(99)
252				MOV	A+2(SP);	
253				MOV	8+0+2(SP	
254				JSR	PC, EMUS2	
255				ADD	R4, R1	
256				ADC	RØ	
257				ADD	R5, R2	
258				ADC	R1	
259				ADC	RØ	
260				MOV	A(SP),=((SP)
				MOV	8+0+2 (SF	
261				JSR	PC, EMUSA	
262				ADD	R4, R0	
263						
264				ADD	R5, R1	
265	1			ADC	RØ	JGET SUM OF EXPONENTS
266	i			MOV	(SP)+,R4	A JGET SUM OF EXPONENTS
267				, ENDC		
260		006303		ASL	R 3	ISHIFT OUT NORMAL BIT
269		006102		ROL	R2	
270		006101		ROL	R1	
271		006100	فر	ROL	RØ	
1 Star 1 14				BCS	NOMS28	JUMP IF IT WAS FOUND
272		103405		ASL	R3	
273		006303				
274		006102		ROL	R2	
275		006101		ROL	R1	IMUST HAVE GOT IT NOW
276		006100		KOL	RØ	a strangen provide state and strangen a
277	6456	005304		DEC	R4	JADJUST EXPONENT
278	6460	162704	NOM5281	SUB	#200,R4	ITAKE OUT ONE OF THE EXCESS 128'S
		000200				
279	6464	603453		BLE	UND\$28	JUMP IF UNDERFLOW
280		022704		CMP	#377,R4	
		000377				
281	6472	002445		BLT	OVR\$28	JUMP IF OVERFLOW
282		105003		CLRB	R3	
		150203		BISB	R2,R3	ISHIFT FRACTION RIGHT
283		-	2 ¹	SWAB	R3	
284		000303		CLRB	R2	
285		105002				
286		150102		BISB	R1, R2	
287		000302		SWAB	R2	
286	6510	105001		CLRB	R1	
289	6512	150001		8158	RØ,R1	
290	6514	000301		SWAB	R1	
29		105000		CLRB	RØ	
292		150400	14 A	BISB	R4, RØ	
29		000300		SWAB	RØ	
294		006026		ROR	(SP)+	JGET PRODUCT SIGN
295		006000		ROR	RØ	INSERT IT IN RESULT
		· · · · · · · · · · · · · · · · · · ·		ROR	R1	a da se no se
296		006001		ROR	R2	
297		006002				
298				ROR	R3	FROUND RESULT
299		005503	-5.	ADC	R3	LUNNIN LPAFI
301		005502		ADC	R2	
30	0542	005501		ADC	R1	
30	2 6544	005000		ADC	RØ	காட்டான் அடை திர்சுறிறார் திரி 15 நாகிராக(க
30		103410		BCS	OV1\$28	JUMP IF OVERFLOW ON ROUND

12

*

-1

ŵ.

SMLD05 MACRO VR04=14 07=SEP=72 11:43 PAGE 33+

<u>ان</u>

¥

			102415		BVS	011528
	305	0552	010066	001\$28:	MOV	RØ, HESLT(SP) / PUT DUT ANSWER
			000014			
	306	6556	010166		MOV	R1,RESLT+2(SP)
			000016			
	307	6562	010266		MOV	R2, RESLT+4(SP)
			000020	an an the second se		
	308	6566	010366		MOV	R3,RESLT+6(SP)
			000022			
			012005		MOV	(SP)+,R5
	310	6574	012604		MOV	(SP)+,R4
	311	6576	062706		ADD	#8., SP IFLUSH TOP ARGUMENT
			000010			
			000134	$(F_{i}) = i F_{i}$	JMP.	Ø(RA)+ IRETURN
	313	6604	005746	OV1\$281	TST	=(SP) /FAKE SIGN
	314	6606	012700	0VR\$281	MOV	#5003,R0 /ERROR 3,10
			005003			
			000402		BR	ECLS28
	316	6614	012700	UNU\$28:	MOV	#3005,R0 JEKROR 5,6
			003005			
	317	6620	005725	ECL\$281	TST	(SP)+ /FLUSH SIGN
			004567		JSR	R5, SERRA JCALL ERROR
			003170			
	319	6626		ZE23281	CLR	RØ JCLEAR HIGH ORDER RESULT
			005001		CLR	R1 JCLEAR LOW ORDER
			005002		CLR	R2
			005003		CLR	R3
			000745		BR	OUTS28
	324	- 4 - 0			IFNUF	EAESMULDIV
		6640	006204	MLTS28:		R4 ITEST NEXT MULTIPLIER BIT
			103022	IN TOWAR	BCC	X0\$28 JUMP IF IT IS 0
	327			MT1528:		A+6+4(SP),R3 ADD IN MULTIPLICAND
	021	VU 74	0000000	111 A DEO 4		ATOTALOFYING JADU IN DUBITELLAND
	209	5650	000022		A 75 P	R2
			005501		ADC	R1 sector of which is the sector sector.
			005500		ADC	Roman Contraction and the second s
	331		005505		ADC	R5 ISAVE OVERFLOW
• •	332	0000	066602		ADD	A+4+4(SP),R2
	3 7 7		000020			
			005501		ADC	R1 Contraction of the second sec
		6666			ADC	
			005505		ADC	R5
	335	6672	066601	1. J. S.	ADD	A+2+4(SP),R1
	70 or 70	10 x 10 x	000016		4 474	
			005500		ADC	RØ
		6700	005505		ADC	R5
	339	6702	066600		ADD	A+0+4(SP), R0
			000014			
	340	6706	005505		ADC	R5
	341	6710	006205	X03281	ASR	R5 IRECOVER OVERFLOW IF ANY
	342	6712	006000		ROR	RU INOW SHIFT PRODUCT
	343	6714	006001		ROR	R1 - Contraction of the second s
			006002		ROR	R2
			006003		ROR	R3
			005366		DEC	2(SP) /COUNT LOOP
		- , "" 144	000002			nama ny kaodim-paositra dia mampina mandritra dia mang manana dia kaodim-paositra dia kaodim-paositra dia kaodi Ny INSEE dia mampina dia kaodim-paositra dia kaodim-paositra dia kaodim-paositra dia kaodim-paositra dia kaodim- Ny INSEE dia mampina dia kaodim-paositra dia kaodim-paositra dia kaodim-paositra dia kaodim-paositra dia kaodim-
	347	6726	003344		BGT	MLTS28 JAGAIN PLEASE
					÷ ·	

121

SMLD05 MACRO VR04=14 07=SEP=72 11143 PAGE 33+

348	6730	000207		RTS	PC	IRETURN TO CALLER
349	6732		MT25281	DEC	2(SP)	100 ONLY 15 BITS THIS PASS
		000002				
350	6736		MT05281	ASR	R4	ITEST NEXT MULTIPLIER BIT
351		103007		BCC	X00528	JJUMP IF Ø
		066501		ADD	A+2+4 (SF	
	****	000016				
787	6716			ADC	RØ	
353		005500 005505		ADC	R5	
354					A+0+4(SF	
355	0702	066600		ADD	Waffaarfou	r 🖌 👔 🕅 🕼 🖓 🖓 🖉
		000014		1		
356	1 11 40			ADC	R5	a watan ana ang mana
357	6760	··· · · · · · ·	X005281	ASK	R5	FRECOVER ANY OVERFLOW
358	6762			ROR	RØ	
359	6764			ROR	R1	
360	6766	006002		ROR	R2	
361	6770	006003		ROR	RJ	
362	6772	005366		DEC	2(SP)	ICOUNT LOOP
		000002				
363	6776	003357		BGT	MTØS28	
364	7000	000207		RTS	PC	FRETURN TO CALLER
365				. ENDC		
366			·	. IFOF	EAE	
367	4		EMUS28:	CLR	PSP	ICLEAR PRODUCT
368				TST	PR4	
369				BEG	MZ\$28	JJUMP IF MULTIPLIER Ø
370				BGT	MPLS28	
371				TST	2(SP)	JTEST MULTIPLICAND
372			- 	BEQ	MZ\$28	IJUMP IF 0
373				BGT	MNGS28	JJUMP IF ++=
374				ADD	(R4)+,#8	
375			a sa kara ta bata kara	ADD	2(SP),05	
376				BR	EML\$28	
			MPLS281	TST	2(SP)	ITEST MULTIPLICAND
377			n r w g e o e	BEQ	MZ\$28	JJUMP IF Ø
378					MLQ328	JUMP IF •
379				BGT		
380				ADD	(RA)+,05	
381			NUC-00-	BR	EMLS28	
382			MNGS281	ADD	2(SP),08	
383			ML03281	TST	(R4)+	POINT TO MUL
384			EMLS281	MOV	2(SP), 01	
385				MOV	=(R4),21	
386				ADD	=(R4),08	
387				TST	(R4)+	POINT TO MQ
388				JMP	PR5	IRETURN
389			MZ\$281	CLR	2(SP)	IRETURN Ø
390				JMP	@R5	
391				.ENDC		$\frac{1}{2} = \frac{1}{2} \left[\frac{1}{2} \left[$
392				. IFOF	MULDIV	
393			EMUS281	CLR	=(SP)	ICLEAR HIGH PRODUCT
394				TST	R4	ITEST MULTIPLICAND
395				BEQ	MZ528	JUMP IF Ø
396				BGT	MPL528	1 A second se
397				TST	4 (SP)	JTEST MULTIPLIER
398				BEQ	MZ\$28	JJUMP IF 0
399				BGT	MN1528	
400				BR	MNG528	
				w • •	• / · 5	

122

(

SMLD05 MACRO VR04=14 07=SEP=72 11:43 PAGE 33+

ъ

ŧ

16-

ø

401	MPLS281	TST	4(SP) JTEST	MULTIPLIER
402		BEQ	the second se	IF Ø
403		BGT	MLQ\$28 1+	
404		ADD	R4,05P	
405		BR	MLQ328	
	MNG\$281	ADD	R4,05P	
	MN15281	ADD	4(SP),05P	
	ML45281	WORD	070466,4	JIMUL 4(SP),R4 JGET PRO
	MDNS281	ADD	(SP)+,R4	JADD IN HIGH URDER PARTS
410		MOV	(SP)+, @SP	JFLUSH MULTIPLIER
411		RTS	PC IRETUR	
	MZ 528 :	CLR	R4 IRESUL	
	C C C C C C			
413		CLR	R5	
414		BR	MUNS28	
415		. ENDC		
416		ENDC		
417		ENDC		

SML105 MACRO VR04=14 07=SEP=72 11:43 PAGE 34

				TITLE	SMLIU5		
1							
2				स्त गर 🖓	CNUS29		
3				.GLOBL	SMLI, SER		911 N
4			1	SMLI	- INTEGE	R MULTIPLY	
S.			1				
5 6 7 8 9			<u>'</u>	* MI *	V005A		
0			1	SMLI	Annaw .		
7			the production of the second				
8			1 1	COPYRIGH	T 1971,	DIGITAL EQUIPMET CORP., MAYNARD,	mage.
ā			j	CALLED T	N THE PO	LISH MODE	
				REPLACE	THE TWO	INTEGERS ON THE TOP OF THE STACK	
10			/		1006.159 76 60600		
11			1		IR PRODU	💆 🎼 an	
12		0000000		RØ#%0			
13		000001		R1#%1			
		000002		R2=%2		n an entre tre sector de la construcción de la construcción de la construcción de la construcción de la constru Nota de la construcción de la constru	
14							
15		000003		R3=X3			
16		000004		R4=%4			
17		000005		R5=%5			
18		000000		SPEX6			
				SR\$29=17	7314		
19		177311					
20		177304		MG=17730			
21				. IFNDF	EAESMULD) I V	
	17000	005000	SMLII	ČLR	RØ	ICLEAR PRODUCT SIGN	
			And the state of the state	MOV	(SP)+,R1		
23		012601				JJUMP IF +	
24		003003		BGT	P1529	TUUBT AF T Tuub Te Angure Te Yead	
25	17010	001455		BEG	ZERS29	JUMP IF ANSWER IS ZERO	
		005200		INC	RØ	INOTE -	
		005401		NEG	R1		
			848001	MOV	PSP, K3	JGET MULTIPLIER	
28		011003	P15291			ter ter filter	
29	17020	003003		BGT	P2\$29		
30	17022	001450		BEQ	ZERS29		
		005200		INC	RØ	FORM RESULT SIGN	
91	11064			NEG	R3		
		005403				ISAVE R4	
33	17030	010446	P25291	MOV	R4, . (SP)		
34	17032	012704		MOV	#8,,K4	ISET UP FOR LOW EIGHT BITS	
		000010					
	17036			CMP	R1,RS		
	17036					JUMP IF MULTIPLIER SMALLER	
36	17040	002003		BGE	CLR529	·····································	
37	17042	010102		MOV	R1, R2	FIF NOT MAKE IT SO	
38		010301		MOV	R3,R1		
		010203		MOV	R2, R3		
33	1/040	010000	a. 8. on.		R2	ICLEAR HIGH ORDER PRODUCT	
40	17090	002005	CLR\$291			1999 - 19	
41	17052	006002	MULS291	ROR	R2	ISHIFT PRODUCT	
		006003		ROR	R3		
		103001		BCC	CYC\$29	JUMP IF MULTIPLIER BIT IS Ø	
				ADD	R1, R2	JADD IN MULTIPLICAND	
44	1/000	000102				COUNT LOOP	
45	17062	005304	CYCS291	DEC	R4	ICOUNI COOP	
46	17064	003372		BGT	MULS29		
		012604		MÖV	(SP)+,R	4 JRESTORE R4	
				TSTB	R3	ITEST HIGH MULTIPLIER	
		105703			OVR\$29	JUMP IF MULTIPLIER NOT GONE	
49	17072	001026		BNE		"专业规则" 电F 则带输出电压器器的 "这些人,不是这种 专业会议的 的复数内门间带 的复数目光	
50	17074	150203		BISB	R2, R3	IMOVE PRODUCT RIGHT	
51	17076	000303		SWAB	RJ		
		105002		CLRB	R2		
					RZ		ъ.
		000302		SWAB		TONE LART SHIFT	
54	17104	006202		ASR	R2	JONE LAST SHIFT	
55	17106	001020		BNE	OVRS29	JUMP IF PRODUCT EXCEEDS 15 BITS	
S.S.	17110	006003		ROR	R3		
90	71 T 80	******			- 179		

SML105 MACRO VR04=14 07=SEP=72 11:43 PAGE 34+

5	7 17112	005403		NEG	R3	IMAKE -		
	8 17114	100015		BPL	0VR\$29		TOO BIG	
-	9 17116	006000		ROR	RØ		DUCT SIGN	
		103402		BCS	OUTS29	JJUMP IF		
-	1 17122	005403		NEG	R3	IMAKE +		
	2 17124	102411		BVS	OVRS29	V CARLE -		
-			0070000	MOV	R3,05P	IMOVE OU	T RESULT	
6		010316	OUT\$291	JMP	@(R4)+	IRETURN		
	4 17130	000134	NOMODO	NEG	R3		R OCTAL 10000	0
	5 17132	005403	NGM5291		OVR329	JUMP IF		2
	6 17134	102005		BVC Ror	RU		R NEGATIVE RE	siii T
-	7 17136	006000					CAN HANDLE T	
6		103772		BCS	OUTS29			H+3
	9 17142		700.001	BR	OVR529 ØSP	IOVERFLO		
		005015	ZER\$291	CLR			KUVULI	
7		000134		JMP	@(R4)+	IRETURN		
7				ENDC				
7			7		DE FOR T	HE EAC		
7				.IFOF	EAE	1078 VO	ABBBEEB	
7			SMLII	MOV	#MQ, KØ	IGET MO		0.40
7				MOV	(SP)+,(IMULTIPLIER T	
7				MOV	(SP)+,0		IMULTIPLICAND	
- 7	8			MOV	-(RØ),=		PRODUCT TO S	TACK
7	75			BITB	#2,5852	9		
8	Ø			BEQ				SINGLE PRECISION
8	1			JMP	@(R4)+	IRETURN	TO USER	
8	2			. ENDC				
8	3		1	SMLI FO	R THE MU	LDIV		•
8	4			. IFDF	MULDIV			
8	5		SMLII	MOV	(SP)+,R	Ø	IMOVE MULTIPL	IER
8	6			WORD	070026	11MUL	(SP)+, R0	IMULTIPLY
8				MOV	R1,-(SP)	IPUSH PRODUCT	
8				BCS	OVRS29	JJUMP IF	OVERFLOW	
8	9			JMP	@(R4)+			
9				.ENDC				
9		005016	UVK\$291	CLR	(SP)	RETURN	0	
9		004567		JSR		JERROR 3		
	≖a atra,≊"R⊎	002630		····			▼ 5 ¹¹	
۵	3 17156	000134		JMP	@(R4)+			
0	4 17160	000.04		BYTE	3			
	5 17161	016		BYTE	14.			
9	• • • •	~ * 4		ENDC	19 °. 19			
3	u					• • • • • •		

129

ιto,

SMLR05 MACRO VR04=14 07=SEP=72 11:43 PAGE 35

			*****		•				
				SMLR05					
				CND\$30					
				SMLR, SER					
		1	SMLR	THE REAL	MULTI	PLY ROUTINE			a tha an Area
		1							
		·	18 14 1 79 1 . ····	UDBEA					
		1	SMLR	V005A					
		1				· · · · · · · · · · · · · · · · · · ·			4.00
		1	COPYRIGH	T 1971,	DIGITA	L EQUIPMENT	LORP.,	MAYNARU,	MASS.
		1	-						
		1	CALLED I	N POLISH	H MODE.				
		1	REPLACES	THE TOP	THO RI	EALS ON THE	STACK		승규님은 가격
		ì	WITH THE						1 - A - A - A - A - A - A - A - A - A -
	000000		ROAXO						1994 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -
	000001		R1#%1						1
	000002		R2#%2						
	000003		R3=%3						
	000004		R4=%4						
	000005		R5#%5						
	000006		SP=%6						
	000007		PC=X7						
	177304		MQ=17730	4					
	177311		SR#17731						
	177314	A second	LSH=1773						
				1 m					
	000000	A C3	FØRXØ						
	000010	Aso.							
	000014	B#12,							
	000010	RESLIES.							
	000002	SIGN#2							
			. IFDF	FPU					
		SHLRI	WORD	170001	11SETF				
		WITH T	WORD	172426	11LDF	(SP)+,FØ		IGET MULT	TPI TCAN
						(SP)+,FØ		MULTIPLY	
			.WORD	171026	17MULF				
			.WORD	174046	IISTF	FØ,=(SP)		PRODUCT	IN SINC
			JMP	@(R4)+					
			.ENDC						
			. IFNDF	FPU					
17162	010446	SMLRI	MOV	R4,-(SP))				
	010546		MOV	R5, - (SP)					ay the table of the
** * * *	******		. IFNDF	EAESMULC					
17160	01 66 A A			A+0-4(SF					
1/100	016602		MOV	WARA COL	1. J N 6				
	000004								
17172	006302		ASL	R2		MULTIPLICA	NV		
17174	006146		ROL	= (SP)	IKEEP				
17176	005046		CLR	•(\$P)	ICLEAR	EXPONENT			
	000302		SWAB	R2					
	110216		MOVB	R2,05P	IKEEP	MULTIPLICAN	D EXPON	ENT	
	001507		BEQ	ZE1830	JUMP	IF ANSWER I	S ZERO		
1720A			SEC	ज्या स्थापन् भी ^{भा} नेको प्		T NORMAL BI			
			ROR	R2	1. 16 (1. 04 mm) J	1 1790-11790 Million	1		
17206			- n U n						
17206	006002								
17206 17210 17212	006002 105002		CLRB	R2	10 m				
17206 17210 17212	006002 105002 156602			A+3 (SP)	R2				
17206 17210 17212 17214	006002 105002 156602 000013		CLRB BISB	A+3(5P),	R2				
17206 17210 17212 17214	006002 105002 156602 000013		CLRB		R2				
17206 17210 17212 17214 17224	006002 105002 156602 000013 005003		CLRB BISB CLR	A+3(5P), R3					
17206 17210 17212 17212 17214	006002 105002 156602 000013		CLRB BISB	A+3(5P),					

126

(

SMLR05 MACRO VR04=14 07=SEP=72 11:43 PAGE 35+

"c

÷¢

ISHIFT HIGH MULTIPLIER B(SP) ASL 55 17230 006366 000014 JGET PRODUCT SIGN 56 17234 005566 ADC SIGN(SP) 000002 TSTB 8+1 (SP) 57 17240 105766 000015 BEQ ZE1530 JJUMP IF ZERO 58 17244 001467 ROR B(SP) **ISIGN IS NOW ZERO** 59 17246 006066 000014 CLR RØ. ICLEAR PRODUCT 60 17252 005000 61 17254 005001 CLR R1 JGET LOW ORDER MULTIPLIER 62 17255 016504 B+2(SP),R4 MOV 000016 B22530 BEQ 63 17262 001406 64 17264 012705 82N\$30: MOV #15. / R5 000017 JSR PC, MT0530 65 17270 004767 000220 JOD LAST LOW BIT FULL PRECISION 66 17274 004767 PC, MLT\$30 JSR 000160 JGET HIGH ORDER BITS B(SP),R4 67 17300 016604 B225301 MOV 000014 ITHERE ARE ONLY SEVEN OF THEM MOV 47,R5 68 17304 012705 000007 PC, MLTS30 JSR 69 17310 004767 000144 IGO DO THE NORMAL BIT 70 17314 004767 JSR PC, MT1530 000144 ADD (SP)+,R4 JADD EXPONENTS 71 17320 062604 .ENOC 72 EAE CODE 73 1 . IFDF EAE 74 (A1+A2+2++=10) + (B1+B2+2++=10)75 1 #MQ,R4 JPOINT TO MQ MOV 76 JGET LEADING BIT MOV #100000,K5 77 ILOW ORDER B TO MQ 78 MOV B+2=4(SP),0R4 B+0-4(SP),=(R4) JHIGH TO AC 79 MOV ZERSJØ JJUMP IF Ø BEQ 80 IGET SIGN INC PHLSH 81 RORB Ø#SR 82 -(SP) ISAVE IT ROL 83 (R4)+,=(SP) ISAVE EXPONENT MOV 84 0SP IRIGHT JUSTIFY IT CLRB 85 **e**SP SWAB 86 IMOVE FRACTION LEFT #7,0#LSH MOV 87 ISAVE B2 0R4,=(SP) MOV 88 JINSERT NORMAL BIT R5,=(R4) BIS 89 (R4) + = (SP)ISAVE B1 MOV 90 ILOW ORDER A TO MQ A+2+4(SP),0R4 MOV 91 A+0+4(SP),=(R4) ;HIGH TO AC ZE2530 ;JUMP IF 0 MOV 92 BEQ 93 IGET SIGN INC 0#LSH 94 RORB O#SR 95 IGET RESULT SIGN AOC 6(\$P) 96 IGET EXPONENT MOV PR4,R3 97 CLRB R3 98 R3 SWAB 99

127

100			ADD	R3,4(SP)				XPONENTS	
101			MOV	#7,0#LSH				FRACTION	
102			MOV	(R4)+,R2		ISAVE			
103			BIS			RT NORMAI			
104			CLR	RØ	ICLEAR	R PRODUC	r. , .		
105			CLR	RI					
106			MOV	(R4)+,R3		ISAVE	A2		
107			BNE	AZNSJØ				1 - 1 - E	
108			TST	= (R4)	1 POTN	TO MQ			
109			BR	A22530		r cut			
				esp, era					
110		A2N5301					* ***		
111			CMP	• (R4) , = (ne j ::	VIO COND	TTU AC	* (1.5)	
112			ADD	R3,084	IAZ.	E'S CUMP	LUNKELI	100	
113			TST	RJ					
114			BPL	A2P\$30		· · · · · · · · · · · · · · · · · · ·			
115			ADD	ØSP, ØR4	181. (
116		A2P\$301	MOV	(R4)+,R1		1HIGH	PRODUCT	TO R1	
117		A225301	MOV	2(SP), (R	(4) +	182 TI	D MQ		
118			BNE	B2N\$30					
119			TST	= (R4)	IPOTN.	TO MO			
			BR	822530				5 G	
120		OMNOTO					a tha a starte		
121		82N\$301		R2,0R4		32*A1			
122		an the sheet	CMP	=(R4),=(TPUIN	T TO AC		
23			ADD	2(SP), 06	4	182.	CORRECTI	QN	
24			TST	2(SP)					
25			BPL	82P\$30	JUMP	IF 82 +			
26	1		ADD	R2,0R4	7A1. 0	CORRECTIO	JN .		
27		B2P\$301	ADD	(R4)+,R1		THIGH	PRODUCT	TO R1	
28			ADC	RØ					
		8245301		R2, (R4) +	1.1.1.N	JA1 TO	า ผู้เวิ		
29		05-9001				· · · · · ·	· · · · · · · · · · · · · · · · · · ·		
30			ADD	R2, RØ					
31			MOV	PSP, PR4		1 # 01			
32			ADD	(SP)+,R0				4.	
33			ADD	=(R4),R1		 A 199 			
34			ADC	RØ					
35			ADD	= (R4), RØ		JAC+RI	3		
36			TST	(SP)+		32			
37			MOV	(SP)+,R4			SUM OF EX	XPONENTS	
38			ENDC						
		•	MUL/DIV	000					
39		1							
40		-	.IFDF	MULDIV	0 n A	A			
41		l I a state		2**=10)*(
42			MOV	8+2=4(SP			IRDER B		
43			MOV	8+0-4(SP	1,R4	;HIGH	ORDER		
44			BEQ	ZERSJØ					
45			NORD	073427,1		11	ASHC	#1,R4	IGET SIG
40			ROL		ISAVE				
			MOV	R4,-(SP)			EXPONEN	T	
47				• • •		JONYL	EV-PUEN	•	
48			CLRB	PSP State	-				
49			SWAB			JUSTIFY		1. Mar - A	- 1 E m
50			WORD	073427,7		11	ASHC	#7,R4	ILEFT JU
51			MOV	R5,=(SP)		ISAVE			
52			BIS	#100000,	R4	INSE	IT NORMAL	. BIT	
53				R4,=(SP)		ISAVE			
54			MOV	A+2+4(SP		IGET A			
55			MOV	A+0+4(SP		IGET A			
56				ZE2830				Ø	
				~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~			, tiby bar¥##	*	

# 128

(

10

Ð.

157				WORD	073287,1			11		ASHC	#1,R2	2 /GET	SIG
158				ADÇ	6(SP)	IGET							
159				MOV	R2, R0	IGET	EXP	ONEN	T				
160				CLRB	RØ								
161			er tre di	SWAB	RØ								
162				ADD	R0,4(SP)			IGET	SUM		PONENTS	3	
163				.WORD	073227,7	,		11		ASHC	#7,R2	2 JGET	A 1
164				BIS	#100000,	R2				NORMAL	BIT		
165				CLR	RØ	ICLE	AR A	CCUM	ULAT	OR			
166				CLR	R1								
167				TST	RJ	ICHE	CK A	2					
168				BEQ	A22530	JUM	P IF	0					
169				WORD	070403	11	ļ	MUL		R3,R4	IGET	A2+81	
170				ADD	R3, R4								
171				TST	RJ								
172				BPL	A2P\$30	JUM	P IF	A2 -	+				
173				ADD	ØSP,R4	781							
174			A2P5301		R4, R1			***1					
175			A225301	MOV	2(SP),R4			182		ULTIPL	IFR		
176			~~~~	BEQ	B22530	JUM			1	· · · · · · · · · · · · · · · · · · ·			
177				WORD	070402	11		MUL		R2, R4	IGET	A1+82	
178				ADD	2(SP),R4		-			· · · · · · · · · · · · · · · · · · ·	<b>1</b>	raja naraa .	
				TST									
179					2(SP)			80.	<b>.</b>				
180				BPL		JUM							
181			80	ADD	R2, R4	FA1							
182			82P\$301	ADD	R4, R1	FA1*	P5+5	***1	0				
183				ADC	RØ		-						
184			822\$30;		R2, R4	FA1	TO M	ULTI	PLIE	ĸ			
185				ADD	R2, RØ	_					· · · · · · · · · · · · · · · · · · ·		
186				. WORD	070416	11.		MUL		esp,R4	JGET	A1+81	
187				ADD	(SP)+, R0								
188				ADD	R5,R1	ILOW	ORD	ERA	1*B1				
189				ADC	RØ								
190				ADD	R4, RØ	THIG		DER	A1*B	1			
191				TST	(SP) +	IPOP							
192				MOV	(SP)+,R4			IGET	SUM	OF EX	PONENTS	3	
193				. ENDC									
194	7322	006101		ROL	R1	ISHI	FT O	UTIN	ORMA	LBIT			
195	7324	006100		ROL	RØ								
		103403		BCS	NOMSSO	JUM	PIF	IT	WAS	FOUND			
		006101		ROL	R1								
198	7332	006100		ROL	RØ	IMUS	T HA	VE G	DT I	T NOW			
		005304		DEC		TADJ							
			NOM\$30:		#200,R4						XCESS 1	2815	
		000200			,		·· **						
201	7342	003436		BLE	UNDS30	JUM	P IF	UNDI	ERFL	ÛW			
		022704		CMP	#377,R4	र, जन आसे हैं। े		<b>677</b> 3		····· •• •			
5 V 6	. <b>.</b>	000377		चन र ३ १	- <b>THE TO BOOM</b>								
201	7350	002427		BLT	OVR\$30	JUM	PIP	0VF1	REIN	W			
		105001		CLRB	R1	r www.tii	• • •	₩. [™] , ⁱ in l					
		150001		BISB	RØ,R1								
		1000301											
				SWAB	R1								
		105000		CLRB	RU.								
		150400		BISB	R4, R0								
		000300		SWAB	RØ		a D e l			<b>N</b> i			
		006026		ROR	· ·	JGET							
211	13/0	006000		ROR	RØ	INS	<u>c k</u> T	T. T	NRE	JULT			

#### SMLR05 MACRO VR04=14 07=SEP=72 11:43 PAGE 35+

212	7372	006001		ROR	R1
213	7374	005501		ADC	n 🖁 🛔 an a' shekara na shekara ta shekar
214	7376	005500		ADC	RØ
215	7400	103414		BCS	OV1830 JJUMP IF OVERFLOW ON ROUND
	7402	102413		BVS	011330
	7404	010066	OUTS30:	MOV	RØ, RESLT(SP) / PUT OUT ANSWER
<b>E</b> 17	1464		001300	TENNY .	
0.00	7	000010		MON	04 DRCI 740/003
510	/410	010166	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -	MOV	R1,RESLT+2(SP)
		000012		li mir	ADDAL BE
219	7414			MOV	(SP)+,R5
220		012604		MOV	(SP)+,R4
221		022020		CMP	(SP)+, (SP)+ JFLUSH TOP ARGUMENT
222	7422	000134		JMP	@(RA)+ IRETURN
223				.IFDF	EAEIMULDIV
224			ZE2\$301	CMP	(SP)+, (SP)+ /POP 81,82
225				ENDC	
226	7424	022626	ZE15301	CMP	(SP)+, (SP)+ IPOP SIGN AND EXPONENT
227	7426	000411		BR	ZERSJØ
228	7430		OVR\$301	TST	(SP) + JFLUSH SIGN
229			OV15301	MOV	#6003,R0 JERROR 3,12
100 Big 17	1 - 1 - <b>1</b> - <b>1</b>	006003			
230	7436	000403		BR	ECL\$30
231	7440		UNDS301	MOV	#3405, RØ JERROR 5,7
	,	003405	48.69 <b>8</b> 8 8 8 9		
232	7 4 4 4	005725		TST	(SP) + JFLUSH SIGN
			ECLS301		R5,SERRA JCALL ERROR
~ ~ ~	1 4 4 4	002344			
074	7452		ZERS301	CLR	RØ JCLEAR RESULT
			4ENJUUN		
	7454			CLR	
	1420	000752		BR	OUTS JØ
237				IFNDF	EAE&MULDIV
238		006204	MLTS301	ASR	R4 ITEST NEXT MULTIPLIER BIT
239		103004	· · · · · · · · ·	BCC	X0530 JUMP IF IT IS Ø
240	7464	060301	MT1\$301	ADD	R3, R1
241	7466	005500		ADC	RØ
	7470	103406		BCS	CUVS30
243	7472	060200		ADD	R2, R0
244	7474	006000	XØ\$301	ROR	RØ INDW SHIFT PRODUCT
245	7475	006001		ROR	n R1 - Constant and a second secon
246	7500	005305		DEC	R5 ICOUNT LOOP
247	7502	003366		BGT	MLTSJØ JAGAIN PLEASE
	7504	000207		RTS	PC JRETURN TO CALLER
249	7506		COV\$301	ADD	R2, RØ IFIRST ADD OVERFLOWED RØ
250	7510	000261		SEC	ISHOW THIS OVERFLOW TO SHIFT
251	7512	000770		BR	X0\$30
252	7514		MT0\$301	ASR	R4 IREDUCED PRECISION MULTIPLY
253		103001	1. <b>1</b> . <b>1</b>	BCC	X00530
	7520	050200		ADD	R2, R0 JUSE ONLY HIGH ORDER MULTIPLICAND
255	7522		X005301	ROR	RØ
256	7524	006001		ROR	R1
257	7525	005305		DEC	R [®]
258	7530	003371		BGT	MTOS\$0
	7532			RTS	
259	1048	<u>nnnen</u> i		ENDC	F ¥
260					
261				. ENDC	
262				.ENDC	

130

4

ſ

(

# SNEG02 MACRO VR04=14 07=SEP=72 11143 PAUE 36

Ħ

Þ

1			•	•TITLE •IFOF	SNEGØ2 CNDS31					
3			7	SNEG	V002A					
5 6			COPYRI	IGHT 197	1,1972 D	IGITAL EQUIPM	ENT CORPORAT	ION, MAY	YNARD, I	MAS
7 8 9 10 11			7 7 7	INTEGER CALLED	, REAL A In the P	GR, SNGD, SERR ND DOUBLE PRE ULISH MODE, M ON TOP OF T		IDN.		
12		000004		R4=%4						
13		000005		R5=%5						
14		000006		5P # % 6		1				
15	17534	005416	SNGII	NEG	PSP	INEGATE AN I				3.9
16	17536	102406		BVS	0VR\$31		000			
17	17540	000134		JMP.	@(R4)+	IRETURN	and a specie			1.1
18	17542		SNGRI	$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{i$	e di terre di di	and the first states of				× .
19	17542	005716	SNUDI	TST	PSP		and the second			<i>U</i> .,
	17544			BEQ	ZERSJI	IJUMP IF Ø T				
21	17546	062716	a di serie da	ADD	#100000	, SP JINV	ERT FLUATING	SIGN		
		100000					and the second			т.
22	17552		ZER\$311		@(R4)+	e 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997	and the second	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -		*
23	17554	004567	OVN\$311	JSR	R5, SERR	JERROR 3,11		s state in the	1999 - Alexandre († 1997) 1997 - Alexandre († 1997)	sa San
		002225		$\{a_{i}\}_{i=1}^{n} = \{a_{i}\}_{i=1}^{n} \in \mathbb{N}$			and the second	Des Vert	: 4° - 5	ź, i
	17560	000134		JMP	@(R4)+	4	1			
25	17562	003		.BYTE	3		a she a she a she	ate toy fill.		43
26	17503	013		BYTE	11.					
27				ENDC						1.1

#### MACRU VR04=14 07=8EP=72 11143 PAGE 37 SPHR07

			.TITLE .IFOF	SPHRØ7 CNDS32				
		1	SPSHR	V007A				14 
		1			i la delete de la delete			
and the second second		ICUPYRIC	GHT 1971	, DIGITAL E	QUIPMENT CO	RPORATION,	MAYNARD,	MASS
		1		en di tan 🏶 tan tan ber	సారిగా ఇద్దాలను కో			
	000000	Rø	3	<b>X 0</b> 1 (2013) - Maria Maria	an a			
2	000001	81	n <mark>a</mark> e por por és	*1				
1	000002	R2	<b>\$</b>	×2				
2	000003	R3		x3	(1999年)。 第二章 1997年(1997年)			
3	000004	RA	8	×4				
4	000005	RB		%5	the second s			
5	000000	SP	<b>E</b>	%6			¥	
5				×7				
				A/ **				
	000007						a ser a s	4
7	00000/	1	DUTINE P		WD. OR FOUR	ITEMS ON	THE STACK	
7 8	00000/	1		LACES ONE. T				
7 8 9	000007	1	FROM TH	LACES ONE, T E REGISTERS	RØ-R3, IT	IS USED AF	TER FUNCT	
7 8 9 0	000007	1	FROM TH	LACES ONE. T	RØ-R3, IT	IS USED AF	TER FUNCT	
7 8 9 0 1	0000001	1	FROM TH CALLS T	LACES ONE, T E REGISTERS O PLACE THE	R0-R3, IT Function Re	IS USED AF Sult on Th	TER FUNCT	-
7 8 9 0 1 2	00000X	ITHIS RU	FROM TH	LACES ONE, T E REGISTERS	R0-R3, IT Function Re	IS USED AF Sult on Th	TER FUNCT	-
7 8 9 0 1 2 3 17564		) )THIS R( ) ) SpSHR51	FROM TH CALLS T ,GLOBL	LACES ONE, T E REGISTERS O PLACE THE SPSHR5, SPSH	R0-R3, IT Function Re R4,\$P\$HR3,\$	IS USED AF Sult on Th PSHR2, SPSH	TER FUNCT IE STACK IR1	-
7 8 9 1 2 3 17564 4 17564	010346	) )THIS R( ) ) SPSHR51	FROM TH CALLS T .GLOBL Mov	LACES ONE, T E REGISTERS O PLACE THE SPSHR5, SPSH R3,=(SP)	R0-R3, IT Function Re R4,\$P\$HR3,\$	IS USED AF Sult on Th	TER FUNCT IE STACK IR1	-
7 8 9 1 2 3 17564 4 17564 5 17566	010346 010246	) )THIS R( ) SPSHR51 SPSHR41	FROM TH CALLS T ,GLOBL Mov Mov	LACES ONE, T E REGISTERS O PLACE THE SPSHR5, SPSH R3,=(SP) R2,=(SP)	RØ=R3. IT Function re R4,\$pshR3,\$ ;push	IS USED AF Sult on Th PSHR2, SPSH Four Words	TER FUNCT IE STACK IR1	
7 8 9 1 2 3 17564 4 17564 5 17566 6 17570	010346	J JTHIS R( J J SpShR51 SpShR51 SpShR31	FROM TH CALLS T .GLOBL Mov	LACES ONE, T E REGISTERS O PLACE THE SPSHR5, SPSH R3,=(SP)	RØ=R3. IT Function re R4,\$pshR3,\$ ;push	IS USED AF Sult on Th PSHR2, SPSH	TER FUNCT IE STACK IR1	
7 8 9 1 2 3 17564 4 17564 5 17566 6 17570 7 17572	010346 010246 010146	/ /THIS R( / / / / / / / / / / / / / / / / / / /	FROM TH CALLS T ,GLOBL Mov Mov Mov	LACES ONE, T E REGISTERS O PLACE THE SPSHR5, SPSH R3, = (SP) R2, = (SP) R1, = (SP)	RØ=R3. IT Function Re R4,\$PSHR3,\$ ;PUSH ;PUSH	IS USED AF Sult on Th PSHR2, \$PSH Four words Two words	TER FUNCT IE STACK IR1	
7 8 9 1 2 3 17564 4 17564 5 17566 6 17570 7 17572 8 17572	010346 010246 010146 010046	J JTHIS R( J J SpShR51 SpShR51 SpShR31	FROM TH CALLS T .GLOBL MOV MOV MOV	LACES ONE, T E REGISTERS O PLACE THE SPSHR5, SPSH R3, = (SP) R2, = (SP) R1, = (SP) R0, = (SP)	RØ=R3. IT Function Re R4,\$PSHR3,\$ ;PUSH ;PUSH	IS USED AF Sult on Th PSHR2, SPSH Four Words	TER FUNCT IE STACK IR1	
7 8 9 1 2 3 17564 4 17564 5 17566 6 17570 7 17572	010346 010246 010146 010046	/ /THIS R( / / / / / / / / / / / / / / / / / / /	FROM TH CALLS T ,GLOBL Mov Mov Mov	LACES ONE, T E REGISTERS O PLACE THE SPSHR5, SPSH R3, = (SP) R2, = (SP) R1, = (SP)	RØ=R3. IT Function Re R4,\$PSHR3,\$ ;PUSH ;PUSH	IS USED AF Sult on Th PSHR2, \$PSH Four words Two words	TER FUNCT IE STACK IR1	-

			.TITLE .IFOF	SPPRØ4 CND343				
			SPOPR5	V004A				
		1 COPYR:	IGHT 19	71, DIGITAL	. EQUIPMEN	IT CORPORATIO	IN, MAYNARD, MASS	3
	000000	1 R0=%0			n an			
3	000001	R1=%1	· · · ·					
		R2=%2					•	
2	000003	R3=X3						
5	000004	R43%4						
1	000005	R5=%5						
)		SP#%6				tan ing kanalan sa		
<b>b</b>	000007	PC=%7						
		1					T	
3		THIS				ITEMS FROM		
3		THIS	AND PL	ACES THEM J	N REGISTE	RS RØ=R3, I	T IS USED IN EX	
8 9 0		I THIS I	AND PL	ACES THEM J	N REGISTE	RS RØ=R3, I		
9 9 0		; ; THIS ; ; ; ;	AND PL Functi	ACES THEM J ONS TO RETU	N REGISTE	RS RØ=R3, I Inction value	T IS USED IN EX	
3 9 0 1 2		; THIS ; ; THIS ; ; ;	AND PL	ACES THEM J ONS TO RETU	N REGISTE	RS RØ=R3, I Inction value	T IS USED IN EX	
3 2 2 3		7 7 7 7	AND PL Functi	ACES THEM J ONS TO RETU	N REGISTE	RS RØ=R3, I Inction value	T IS USED IN EX	
3 2 2 3 4 17576	N12500	1 1 1 5pupr51	AND PL FUNCTI .globu	ACES THEM I ONS TO RETU Sporks, sp	N REGISTE JRN THE FU Popr4,\$pop	RS R0=R3, I Inction value R3	T IS USED IN EX In the registe	
3 2 3 4 17576 5 17576	012600	7 7 7 7	AND PL Functi .globl Mov	ACES THEM I ONS TO RETU Spopns, Sp (Sp)+, RØ	N REGISTE JRN THE FU Popr4,\$pop	RS RØ=R3, I Inction value	T IS USED IN EX In the registe	
3 2 3 4 17576 5 17576 5 17676 5 17600	012601	1 1 1 5pupr51	AND PL FUNCTI .GLOBL Mov Mov	ACES THEM I ONS TO RETU SPOPR5, SP (SP)+, R0 (SP)+, R1	N REGISTE JRN THE FU Popr4,\$pop	RS R0=R3, I Inction value R3	T IS USED IN EX In the registe	
3 4 17576 5 17576 5 17600 7 17602	012601 012602	1 1 1 5pupr51	AND PL FUNCTI .GLOBL MOV MOV MOV	ACES THEM I ONS TO RETU SPOPR5, SF (SP)+, RØ (SP)+, R1 (SP)+, R2	N REGISTE JRN THE FU Popr4,\$pop	RS R0=R3, I Inction value R3	T IS USED IN EX In the registe	
3 4 17576 5 17576 5 17600 7 17602 3 17604	012601 012602 012603	1 1 1 5pupr51	AND PL FUNCTI •GLOBL MOV MOV MOV MOV	ACES THEM I ONS TO RETU SPOPR5, SF (SP)+, RØ (SP)+, R1 (SP)+, R2 (SP)+, R3	N REGISTE JRN THE FU Popr4,\$pop	RS R0=R3, I Inction value R3	T IS USED IN EX In the registe	
3       4       17576       5       17576       5       17602       3       17604       9       17606	012501 012602 012603 000134	I I Spupr51 Spupr41	AND PL FUNCTI •GLOBL MOV MOV MOV MOV JMP	ACES THEM I ONS TO RETU SPOPR5, SF (SP)+, R0 (SP)+, R1 (SP)+, R2 (SP)+, R3 0(R4)+	N REGISTE JRN THE FU Popr4,\$pop ;p	RS R0=R3. I Inction value R3 VOP Four Word	T IS USED IN EX IN THE REGISTE	
3         4         17576         5         17576         5         17602         8         17604         9         17605         17606         17605	012601 012602 012603 000134 012600	1 1 1 5pupr51	AND PL FUNCTI •GLOBL MOV MOV MOV MOV JMP MOV	ACES THEM I ONS TO RETU SPOPR5, SF (SP)+, R0 (SP)+, R1 (SP)+, R2 (SP)+, R3 $\Phi$ (R4)+ (SP)+, R0	N REGISTE JRN THE FU Popr4,\$pop ;p	RS R0=R3, I Inction value R3	T IS USED IN EX IN THE REGISTE	
8         9         1         12         3         4       17576         5       17576         6       17602         8       17604         9       17606         0       17610         1       17612	012501 012502 012603 000134 012500 012501	I I Spupr51 Spupr41	AND PL FUNCTI GLOBL MOV MOV MOV JMP MOV MOV	ACES THEM I ONS TO RETU SPOPR5, SF (SP)+, RØ (SP)+, R1 (SP)+, R2 (SP)+, R3 0(R4)+ (SP)+, RØ (SP)+, RØ	N REGISTE JRN THE FU Popr4,\$pop ;p	RS R0=R3. I Inction value R3 VOP Four Word	T IS USED IN EX IN THE REGISTE	
5 17576 6 17600 7 17602 8 17604 9 17606 0 17610	012601 012602 012603 000134 012600	I I Spupr51 Spupr41	AND PL FUNCTI •GLOBL MOV MOV MOV MOV JMP MOV	ACES THEM I ONS TO RETU SPOPR5, SF (SP)+, R0 (SP)+, R1 (SP)+, R2 (SP)+, R3 $\Phi$ (R4)+ (SP)+, R0	N REGISTE JRN THE FU Popr4,\$pop ;p	RS R0=R3. I Inction value R3 VOP Four Word	T IS USED IN EX IN THE REGISTE	

ŝ

Þ

ø

#### MACRU VR04-14 07-SEP-72 11143 PAGE 39 SRD02

1				.TITLE	SRDØZ					
2				, IFOF	CNDS34					
3				GLOBL	SRD					
4			1	SRD	THE REAL	. TO DOUB	LE PRECISI	ON CON	VERTER	•
5			1							
6 7			i 1 an Suiteiri 1	3RD	V002A	an an thailte				
â			1	COPYRIGH	IT 1971,	DIGITAL	EQUIPMENT	CORP	MAYNARD,	MASS.
å			1		LEROS TO		STACK ITEN			
10						PRECISION			· · · ·	
11		000004	,	R4=%4						
12				SPEXO						
		000006								10 C 10 C
13		000000		F0=X0						
14		000001		F1#%1						
15				. IFDF	FPU					
16			SRDI	.WORD	170011	IISETD	1001 04			ADC
17				. WORD	177426	ILDCFD	(SP)+,F0		CONVERT	ANG
18				WORD	174046	11STD	$FØ_{1} = (SP)$	<u>,</u>		
19				JMP	@(R4)+					
20			and the second second	, ENDC	24	, r. <u>.</u> .				
21				IFNDF	FPU					
22	17616	016646	SRDI	MOV	2(SP),-	(\$P)	IMOVE LOW	ORDER	PART	
	-	000002	· •							
23	17622	016646		MOV	2(SP),=	(SP)	IMOVE HIGH	I ORDER	PART	
		000002								
24	17626	005066		CLR	4(SP)	<b>IINSERT</b>	TRAILING Z	EROS		
<b></b>		000004	× .	··· ···						
25	17632			CLR	6(SP)		14 °			
εņ	TLOAR			<b>* 1</b>	* <b>}</b> * * * *		t			
• •		000006		140	A(34]-					
26	1/090	000134		JMP	@(R4)+		1 A. A. A.			
27				.ENDC		$(1,1)^{(1)} \mapsto (1,1)^{(1)} \mapsto (1,1)^{(1)}$	1.1			
28				.ENDC					A state of the second	

134

æ

-

# SRI04 MACRO VR04=14 07=SEP=72 11:43 PAGE 40

(

÷C.

(

(

Fi.

ø

				SRT04								
					4600						i k	
				3K1,4U1,	PERK							
		1	REAL TU	INTEGER	LUNVER	SIUN.						
		1										
		1	SRI	VOODA								
		1					1 e					
		1	CORVETG	HT 1071.1	972 01	GTTAL	EQUI	PMENT	COR	P., M	AYNAR	0. M
		f 8	ABCIMEN	τις Αιθή		non	FALS	UMBER	0.0	THE	TAP	
					i w la la la in		Sin Priling 17	A M LUM CLU	<b>WIN</b>	1.135	1 1 1	
		1	UP THE	STACK	6		-		n Char	-	1.67	
		J. S.				IT 1T	TU AN	INTE	GER	ON TI	15	
		1	TOP OF	THE STACK								
	000000		R0=%0									
				* 2 ¹								
	-											
						1 - C.						
					· · · · · · · · · · · · · · · · · · ·	16 - A						
				1 1 1 1 A L			$\sum_{i=1}^{n}  A_i  = \sum_{i=1}^{n}  A_i  = \sum_{i$					
	000000											
	177304		MQ=1773	04								
	177314		LSH=177	314								
			FDFXD									
	****			FPU								
		Sole				nnu	RIFE	RECTS	TON			2 C
		40.41	· · · ·	DTDA AR.			Set Spe Set 11		<b>* * * * *</b>			
				KT02021				. ()				
				$\mathbf{J}^{\mathrm{exp}}$ and $\mathbf{J}^{\mathrm{exp}}$								
		RIU\$351	SETI	🕴 e 🖉 e e e e e e e e e e e e e e e e e					5			
			LOD	(SP)+,F0	1	GET	ARGU	IMENT				44 - 194 19
			STCOI	F0.=(SP)	1	CON	VERT	TO ST	ACK	-1		
						RET	URN					
				CON						$\{(i,j)\} \in \mathcal{C}$		
• 7 m A m	-	* = T =			501.	₩ 23 (i	NPATE	. <b>•</b> 0 6	E AL	FARM		
1/040		2011	MUY	(34)+15(	2611		NUMIC	10 6	<u>c</u> r <u></u>	r ukny	• •	
17644	012066		MOV	(SP)+,2(	SP);							1
	000002											
17650		3911	CI.R	R2	ICLEAR	WORK	SPAC	E				
									LI MEN	Ŧ		
								ale mito	Ungn	t ·		
						IGN						
			ROL	=(SP)								
17664	110103		MOVB	R1, R3	IGET H	IGH O	RDER	FRACT	ION			
					IGET P	XPONE	NT					
					1 MC 444 5 244		/ <b>, , ,</b>					
1.0.5			<i>च भू भ</i>									
	000201		ä. <b>≆</b>	750010	JUMP	TC 77	T C 4	00 0				
	002433		BLT	ZER\$35	VUUMP.	TL T1	T2	40 S.	· A 🛏 🛴			
			BEQ	DNES35								
17700	001413											
17700	001413 022701		CMP	#15,/R1								
17700	022701		CMP	#15,/K1								
17700 17702	022701 000017				JUMP	IF IT	15 1	00 B1	G			
17700 17702 17706	022701 000017 002422		BLT	QVRS35	IJUMP JEOEM					FR F		N
17700 17702 17706 17710	022701 000017 002422 000303		BLT Swab	078335 R3	IJUMP IFORM					ER FF	RACTIU	N
17700 17702 17706 17710 17712	022701 000017 002422 000303 105003		BLT Swab Clrb	QVR335 R3 R3	IFORM					ER FF	RACTIO	N
17700 17702 17706 17710 17712	022701 000017 002422 000303		BLT Swab	078335 R3	IFORM					ERF	RACTIŬ	N
	17644 17650 17652 17654 17656 17660 17662 17664 17666 17670	177304 177314 000306 000306 17640 012666 000002 17650 005002 17650 005002 17652 005202 17652 005202 17654 012601 17656 006116 17663 006101 17664 110103 17666 105001 17670 000301 17672 162701	000001 000003 000003 000005 000005 000006 177304 177314 000006 \$DI: \$RI: RID\$35: 17640 012666 000002 17640 012666 000002 17650 005002 \$RI: 17650 005002 \$RI: 17650 005002 \$RI: 17656 006116 17668 105001 17666 105001 17670 000301 17672 162701	J       SRI         J       COPYRIG         J       DF THE         J       TRUNCAT         J       TOP OF         000000       R0=%0         0000001       R1=%1         0000002       R2=%2         0000003       R3=%3         0000004       R4=%4         000005       R5=%5         000006       SP=%6         177304       MG=1773         177314       LSH=177         000006       F0=%0         .IFDF       SDI:         SET0       BR         \$RI:       SETP         ND935:       SETI         LD0       STC0I         JMP       .ENDC         .IFNDF       17640         12666       SDI:         MOV       000002         17654       012666         MOV       000002         17654       012666         MOV       000002         17654       012601         MOV       000002         17654       012601         17650       006101         17654       12601         17654	. IFOF CND\$35 .GLOBL \$KI,\$DI, FREAL TO INTEGER FREAL TO INTEGER	. IFOF CND\$35 .GLOBL \$RI,\$DI,\$ERR ; REAL TO INTEGER CONVER ; COPYRIGHT 1971,1972 DI ; ARGUMENT IS A DOUBLE W ; OF THE STACK. ; TRUNCATE IT AND CONVER ; TOP OF THE STACK. 000000 R0=%0 000001 R1=%1 000002 R2=%2 000000 R0=%0 000000 R3=%3 000000 R3=%3 0000000 R3=%3 000000 R3=%3 0000000 R3=%3 0000000 R3=%3 0000000 R3=%3 000000 R3=%3 00000	.IFOF CNDS35 .GLOBL SKI,SDI,JERR ; REAL TO INTEGER CONVERSION. ; ; SRI V004A ; ; COPYRIGHT 1971,1972 DIGITAL ; ARGUMENT IS A DOUBLE WORD R ; UF THE STACK. ; TRUNCATE IT AND CONVERT IT ; TOP OF THE STACK. 000000 R0#20 000001 R1=21 000002 R2=22 000000 R0#20 000000 R3=23 000000 R3=23 000000 R3=23 000000 R3=25 000000 SP=26 177304 MQ=177304 177314 LSH=177314 000000 F9=20 .IFOF FPU SDI: SETO ; DOU GR RIDS35; SRI: SETF ; SIN RIDS35: SETI ; DOU GR RIDS35; SRI: SETF ; SIN RIDS35: SETI ; SHO LOD (SP)+,F0; GET STCOI F0,-(SP); CON JMP @(R4)+; RET .ENDC .IFNDF FPU 17640 012666 SDI: MOV (SP)+,2(SP); TRU 000002 17654 012666 SDI: CLR R2 ;CLEAR WORK 17654 012661 MOV (SP)+,2(SP); TRU 000002 17654 012601 MOV (SP)+,2(SP); TRU 17656 006101 ROL R2 ;SET UP NOR 17654 012601 MOV (SP)+,R1 ;GET 17656 006101 ROL R1 JAND 17656 006101 ROL R1 JAND 17656 105001 CLRB R1 17670 000301 SWAB R1,R3 ;GET HIGH O 17667 105001 SWAB R1,R3 ;GET HIGH O 17667 105001 SWAB R1,R3 ;GET HIGH O	.IFOF CND356 .GLOBL \$RI, 5DI, 3ERR ; HEAL TO INTEGER CONVERSION. ; SRI V004A ; ; COPYRIGHT 19/1,1972 DIGITAL EQUI ; ARGUMENT IS A DOUBLE WORD REAL N ; UF THE STACK. ; TRUNCATE IT AND CONVERT IT TO AN ; UP THE STACK. ; TRUNCATE IT AND CONVERT IT TO AN ; TOP OF THE STACK. ; 000000 R0=X0 ; 000000 R3=X3 ; 000000 R3 ; 000000 R3=X3 ; 000000 R3 ; 000000 R3	.IFOF CND355 .GLOBL SRI, SDI, JERR ; MEAL TO INTEGER CONVERSION. ; MEAL TO INTEGER CONVERSION. ; SRI V004A ; COPYRIGHT 19/1,1972 DIGITAL EQUIPMENT ; ARGUMENT IS A DOUBLE WORD REAL NUMBER ; UF THE STACK. ; TRUNCATE IT AND CONVERT IT TO AN INTE ; TOP OF THE STACK. 000000 R0=X0 000001 R1=X1 000002 R2=X2 000000 R3=X3 000000 R3=X3 000000 R3=X3 000000 R3=X5 000000 R3=X5 000000 R3=X5 000000 R3=X5 000000 R3=X5 000000 R3=X5 000000 SP=X5 000000 SP=X5 0000000 SP=X5 0000000 SP=X5 SP=X5 0000000 SP=X5 SP=X5 0000000 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=X5 SP=	. IFOF CNDS45 . GLOBL SRI, DD1, JERR ; REAL TO INTEGER CONVERSION. ; REAL TO INTEGER CONVERSION. ; SRI V004A ; COPYRIGHT 1971, 1972 DIGITAL EQUIPMENT COR ; ARGUMENT IS A DOUBLE WORD REAL NUMBER ON ; UF THE STACK. ; TRUNCATE IT AND CONVERT IT TO AN INTEGER ; TQUCATE IT AND CONVERT IT TO AN INTEGER ; TQUCAUE R2=22 000000 R0=20 000000 R0=20 000000 R3=23 000000 R3=23 0000000 R3=23 000000	. IFOF CNOS35 . GLOBL SKI, SDI, SER . GLOPL SKI, SDI, SER . MEAL TO INTEGER CONVERSION. . MEAL TO INTEGER CONVERSION. . MEAL TO INTEGER CONVERSION. . SRI V004A . COPYRIGHT 19/1,1972 DIGITAL EQUIPMENT CORP., f . ARGUMENT IS A DUUBLE WORD REAL NUMBER ON THE . UP THE STACK. . UP OF THE STACK. . IF OF FPU . SDI: SETO ; DOUBLE PRECISION . SRI: SETF ; SINGLE PRECISION . BR RID335; . SRI: SETF ; SINGLE PRECISION . BR RID335; . SRI: SETF ; SINGLE PRECISION . SRI: SETF ; SINGLE PRECISION	. IFOF CN0535 . GLODS SK1, SD1, SERR ; HEAL TO INTEGER CONVERSION. ; SRI V004A ; COPYRIGHT 19/1,1972 DIGITAL EQUIPMENT CORP., MAYNAR ; ARGUMENT IS A DOUBLE WORD REAL NUMBER ON THE TOP ; UF THE STACK. ; ARGUMENT IS A DOUBLE WORD REAL NUMBER ON THE TOP ; UF THE STACK. ; TRUNCATE IT AND CONVERT IT TO AN INTEGEN ON THE ; TOP OF THE STACK. 000000 R0=X0 000000 R0=X0 000000 R0=X0 000000 R0=X0 000000 R0=X0 000000 R0=X0 000000 SP=X0 177304 MG=177304 177314 LSH=177314 000000 F0=X0 . IFOF FPU SD11 SETD ; DOUBLE PRECISION BR R10335; SRI1 SETF ; SINGLE PHECISION BR R10335; SRI1 SETF ; SINGLE PHECISION STOOL F0,-(SP); CONVERT TO STACK PHECISION STOOL F0,-(SP); CONVERT TO STACK PHECISION STOOL F0,-(SP); CONVERT TO STACK PHECISION STOOL F0,-(SP); SINGLE PHECISION STOOL F0,-(SP);

# SRI04 MACRO VR04=14 07=SEP=72 11:43 PAGE 40+

53 54 55 56 57 58	17722	006103 006102 005301 003374	DEC\$351	.IFNDF Rol Rol Dec Bgt .ENDC	EAE&MULL R3 R2 R1 SFT\$35	JGET NE JGET NE JGO AGA	SE EXPONENT	
59			🗜 goran ar					
60			1	EAE CODI				
61				.IFDF	EAE	P. (5) (5) # A1 W	*A MA	
62	· · ·			MOV	#MQ,RØ	IPOINT	TO MQ 'FRACTION	
63				MOV	R3, eKO		FRACILUN	
64				MOV	R2,=(R0)		ISHIFT LEF	T
65				MOV	R1, PHLSH	ר ז מרכווו <b>ז</b>	TO REG	•
66				.ENDC	BRAINS	FREOVE		
67			•		CODE			
68 69			1	.IFDF	MULDIV			
70				WORD	073201	IJASHC	R1, R2	
71				ENDC	******	T & CARLESS		
72	17730	005402	DNE5351	NEG	R2	IMAKE .		t x - 1
73		102406		BVS	NGM535		F POSSIBLE	NEGMAX
74		003007		BGT	OVR\$35			15 BITS
7.5	17736	006026	SGNS351	· · ·	(SP)+	IGET SI		
76		103401		BCS	OUTS 45		F .	
77		005402		NEG	R2	1- RESL	LT	
78		010216	OUT\$351	MOV	R2,0SP	ISTORE	INTEGER RES	
79		000134		JMP	@ (R4)+	IRETURN	N TO CALLER	
80		006026	NGMS351	ROR	(SP)*			
81		103774		BCS	OUT\$35		RESULT TO B	
82		005746	OVR\$351	TST	= (SP)	IFAKE S	BIGN	
83	17756	004567		JSR	R5, SERR	IERROR	3,22	
		002024			A.,		1	
84	17762	000401		BR	ZERS35			
85	17704	003		BYTE	3			
86	17765	026		BYTE	22.	الاسترادية		
87	17766	005002	ZER\$351	CLR	R2	JANSWEF	R IS Ø	
88	17770	000762		BR	SGNS35			
89				ENDC				
90				.ENOC				

136

).

-

*

#### \$SGL02 MACRU VR04=14 07=SEP=72 11:43 PAGE 41

£.

r.

<u>, -</u>:

 $\mathbf{r}$ 

1				.TITLE	\$SGL02				
23			•	. IFOF	CND336	a			
4			;	SNGL	VØØZA				
5			1					-	· · · · · · · · · · · · · · · · · · ·
6 7			I COPYR	IGHT 197	1, DIGIT	AL EQUIPM	IENI CORPU	RATION,	MAYNARD, MASS
8				GLOBL	SNGLISE	RK			
9			1			L FUNCTIC	IN		
10			1	CALLING	SEQUÊNCI				
11			1	JSR	R5, SNGL	1. 12 1. 1.			
12			1	BR	A				
13			11 11 11 1	.WORD	ARGUMEN	T ADDRESS	En en state		
14			7 A I			e egitetet i t			
15			7				INDED TO S	INGLE	
16			1	PRECISO	N REAL FO	DRMAT IN	RØ, R1,		
17			1						
18		000000		R0=%0					
19		000001		fi 1 = %1		Carl March 1			
20		000004		R4=%4			それにすいた。		
21		000000		R5≡%5			1		
22	17772	016504	SNGLI	MOV	2(R5),R4	4	JGET ADDR	ESS	
		000002		1. A. L.		-	a she ta an	17 80 Fr er 25	
23	-	012400		MOV	(R4)+, R0		JGET HIGH		
24	20000			MOV	(R4)+,R1		IGET LOW	ORDER	
25	20002	··· ··· •		MOV		JGET NEX			
26		006104		ROL	R4	IGET ROU			
27				ADC	R1	IROUND R	EAL		
28	20010			ADC	RØ				
29		103402		BCS	OVR\$36	JJUMP IF	OVERFLOW	UN ROUI	ND
		102401		BVS	OVR\$36		and the second sec		
		000205		RTS	R5		TO CALLER		
32	20050		OVR\$361	JSR	R5, SERK	TERROR 4	12		
-		001762							
		000205		RTS					
	20020	004		BYTE	4		. «		
35	20027	014		BYTE	12.				
36				.ENDC					
							2		

137

.

#### SSIN04

04 MACRO VR04=14 07=SEP=72 11:43 PAGE 42

				.TITLE	\$SIN04
				.IFOF	CNU\$37
			,	SINCOS	VØØ4A
			COPYR	IGHT 197	1,1972 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MAS
			;	.GLOBL	SIN, COS;
				. IFNDF	
				.GLOBL .ENDC	SADR, SMLR, SSBR, SDVR, SINTR, SPOLSH
			!	SIN CALLING	COS THE REAL SIN AND COSINE FUNCTIONS
				JSR	R5,SIN (OR COS)
			1	BR	
			1	WORD	ARG ADDRESS
			141		
			1	RETURNS	SIN UR CUS OF ARG IN RØ AND R1
		000000		RØ=%0	
		000001		R1=%1	
		000002		R2=%2	
		000003		K35%3	
		000004		R4=%4	
		000000		R5=%5	a service and a service of the servic A service of the servic
		0000007		PC=%7	
		000000		FØ=XØ	
		000001		F1=%1	
		000002		F2=%2	
		000003		F3=%3	
				IFNDF	ter <b>FPU</b> se de la constant de la const La constant de la cons
21	0030	016504	COSI	MOV	2(R5),R4 JGET ARGUMENT ADDRESS
		000002			
	0034			CLR	-(SP) IMAKE ROOM FOR QUADRANT FLAG
21	0036			MOV	2(R4),=(SP) ;PUSH ARGUMENT
Ä		000002		MMI	ADA - (00)
	0042 0044	011445		MOV Mov	@R4,=(SP) #007733,=(SP) ;PUSH PI/2
61	01074	007733		1104	HONNOOD (GET DEADLETES
20	0050			MOV	#040311,=(SP)
		040311			
21	0054	004467		JSR	R4, SPOLSH JENTER POLISH MODE
		001564			
		002010		.WORD	\$ADR; SNC\$37 ; CUS(X) = \$IN(X+PI/2)
		020100		NAU	
24	0004	016504	⇒Ti4∎	MOV	2(R5),R4 JGET ARGUMENT ADDRESS
90	2020	000002 005046		CLR	- (SP) IMAKE ROOM FOR QUADRANT FLAG
		016446		MOV	2(R4), #(SP)
, <b>1</b>	rs∕ Re	000002		- ' আরে ই	and a constant of the second
21	0076	011445		MOV	PR4,=(SP) ;PUSH ARGUMENT
			SNC\$37:		ØSP IREMOVE AND SAVE SIGN
		006066		ROR	4(SP) FIN QUADRANT FLAG
		000004			
		006016		ROR	
21	0110	012746		MOV	#007733,=(SP)
		007733			

138

**A**.

(

)

€{

×.

4

48	20114	012746	MOV	#040711,=(SP)
49	20120		JSR	R4, SPOLSH JENTER POLISH MODE
En	00.04	001520		90VD 2V/001
50	20124		WORD	SOVR JX/2PI
51		0202221	, WORD	DUPS37 12 COPIES
52	20130		. HORD	SINTH /INT(X/2PI)
53		0020041	. WORD	SSBR /FRACT(X/2PI)
54	20134	0202341	. WORD	X4337 14*FRACT(X/2PI)
55	20136	0202221	. WORD	DUP337 12 COPIES
56		0031421	WORD	SINTR /INT(4+FRACT(X/2PI))
57		0202461	WORD	QUD337 ISAVE INT()
		0020041	WORD	SSBR JY=FRACT(4*FRACT(X/2PI))
58				
59		0202541	. WORD	QST\$37 IREDUCE Y TO (=1,1)
60		0202221Q55371	. WORD	DUP\$37 12 COPIES
61		0202221	. WORD	OUP\$37 13 COPIES
62		0171621	.WORD	SMLR IY+Y
63	20156	0203241	.WORD	PLY\$37 JPUSH COEFFICIENTS
64	20100	0171621	.WORD	SMLR /A4*Y**2
65	20102	0020101	. WORD	SADR 1A4*Y**2+A3
		0171621	WORD	SMLR
67		0020101	NURD	SADR
68		0171621	WORD	SMLR
		0020101	WORD	SADR
69				
70		0171621	, WORD	\$MLR
71		0020101	.WORD	SADR
72	20200	0171621	.WORD	SMLR 1((((A4+Z+A3)+Z+A3)+Z+A2)+Z
73				1+A1)+Z+A0)+Z Z=Y+Y
74		0202041	. WORD	RTN\$37
75	20204	012600 RTN5371	MOV	(SP)+,RØ ;POP HIGH URDER RESULT
76	20206	012001	MOV	(SP)*,R1
77	20210	005725	TST	(SP) + JPOP QUADRANT FLAG
78	20212	002002	BGE	RT1507 JJUMP IF ARGUMENT WAS +
79		062700	ADD	#100000, R0 /SIN(-X) =-SIN(X)
, -		100000		
80	20220		RTS	R5 JBACK TO CALLER
81	6.2.5.40			
82	20222	016646 DUP\$37:	MOV	2(SP),=(SP) ;DUPLICATE STACK ITEM
02	20265		1.04	COLINE AND AND AND ALTER
A 33	00000	000002	há chi há	0.0001 - (80)
83	20226		MOV	2(SP), = (SP)
		000002	<b>b</b>	
84	20232	000134	JMP	@(R4)+
85		and the state of the second second		
		005716 X45371	TST	OSP JCHECK FOR Ø FRACTION
87	20236	001762	BEQ	RTNS37 JUUIT NOW
88	20240	105266	INCH	1(SP) IQUADRUPLE STACK ITEM
		000001		
89	20244	000134	JMP	@(R4)+
90	den all med	1		
91	20246	051666 QUUS37:	BIS	@SP,8,(SP) ISAVE QUADRANT NUMBER
- +	fa ≌ fa TV	000010		ப்பட்டது. தாரையுள்ள கூட்டிய தான்ப் நான்பு கொண்டு கொண்டுகள் காராது. தொண்டுகள் தார்கள் பிரியில் காராது குறையுக்கு
92	20252	000134	JMP	@(R4)+
	6 4 <b>6 4 6</b>			• Spectra Rg, Theorem 2010 and the second s
93	00064	105766 QST\$37:	TSTB	4(SP) JTEST QUADRANT
94	20204		1910	an Penal State Marker and Marker and States
~~	AMPES	000004	DEA	ATTAIN THIND TO STORT AD TUTAN ANAS
95	and	001413	BEQ	Q13537 JJUMP IF FIRST OR THIRD QUAD
96	50505	062716	ADD	#100000, OSP INEGATE STACK ITEM

/39

# SSIND4 MACRO VR04=14 07=SEP=72 11:43 PAGE 42+

-

		100000					
97		005045		CLR		A FLUATING 1.	
98	20270	012746		MOV	#40200,=(SP)		
		040200					
99	20274	004467		JSR	R4, SPOLSH	JENTER POLISH	
	6 M 6 1 4	001344					
					8.00 ACU\$3.7	• V = 4 _ V	
100		002010		, WORD	SADR, QSR\$37	7×=1×	and the second second
		0203041		$(R_{\rm eff}) = \frac{1}{2} \left( \frac{1}{2} - \frac{1}{2} \right) \left( 1$			
101	0304	012704	QSR\$371	MQV	#QSE\$37, K4	POINT BACK INTO LIST	
		020150					
102	0310		0135371	ASRB	5(SP) /TEST	QUADRANT	
101	. VORU						1997 - A. C. S.
4 49 13	-	000005					
103		· · · · · · · · · ·	1			TH PINGT OD SPONDA	
		103002		BCC		IF FIRST OR SECOND	
105	0316	062716		ADD	#100000, •SP	INEGATE STACK ITEM	
		100000					
106	0322		QUTS371	JMP	@(R4)+		
107			in an an grant an				
			∦ 101 ¥∞%272∎	him hi	1993 + DO	CAVE VIV	
108			PLY5371		(SP)+,R0	ISAVE Y+Y	
105		012601		MOV	(SP)+,R1		
110	0330	012/02		MOV	#CON\$37+4,R2	IPUINT TO LIST OF COEF	FICIENTS
		020404			•		
111	0334	012703		MOV	#5,R3		
* * 4		000005					
				6 A	P1 1 4 1 1 1 1 1		
11		000402		BR	PY1537	- Millord M. M	
113			PY25371		$R1_{i} = (SP)$	PUSH Y+Y	
114	0344	010046		MOV	$RØ_{,-}(SP)$		
115	0346	014246	PY18371	MOV	=(R2),=(SP)		
116		014246	•	MOV	= (R2), = (SP)		
117		005303		DEC		T COEFFICIENTS	
				BGT	PY2537		
					FIZMW/		
118	0354	003372					
115	0355			JMP	@(R4)+		
110	0355						
119	0356			JMP			
119 120 121	0356		1	JMP • ENDC	@(R4)+		
119 120 121 122	0356		) CnS1	JMP • ENDC • IFOF	@(R4)+ FPU	DOUBLE PRECISION FP	
119 120 121 122 123	) 0356 ) }		7 Cosi	JMP .ENDC .IFDF SETD	@(R4)+ FPU J	DOUBLE PRECISION FP	
119 120 120 120 122 123	0356 1 2		7 Cosi	JMP •ENDC •IFOF SETD LDCFD	@(R4)+ FPU ; @2(R5),FØ;	GET ARGUMENT	
119 120 121 122 123 124	0356 1 2 3		) Cosi	JMP .ENDC .IFOF SETD LDCFD ADDD	@(R4)+ FPU ; @2(R5),FØ; PI2\$37,FØ;		
$   \begin{array}{c}     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     2 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     1 \\     $	) 0356 ) 2			JMP •ENDC •IFDF SETD LDCFD ADDD BR	@(R4)+ FPU ; @2(R5),FØ;	GET ARGUMENT COS(X) = SIN(X+PI/2)	
119 120 121 122 123 124	) 0356 ) 2		I Cosi Sini	JMP •ENDC •IFDF SETD LDCFD ADDD BR •SETD	@(R4)+ FPU 1 @2(R5),FØ; PI2\$37,FØ; SNC\$37; 1	GET ARGUMENT COS(X) = SIN(X+PI/2) Double precison FP	
	0 0356 1 2 3 4 5 5			JMP •ENDC •IFDF SETD LDCFD ADDD BR	@(R4)+ FPU ; @2(R5),FØ; PI2\$37,FØ;	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT	
	) 0356 ) } }		SINI	JMP .ENDC .IFOF SETD LOCFD ADDD BR SETD LDCFD	<pre>@(R4)+ FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT	
	) 0356 ) ; ; ; ; ; ; ; ; ; ; ;			JMP •ENDC •IFOF SETD LOCFD ADDD BR SETD LDCFD SETI	<pre>@(R4)+ FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) Double pregison fp Get argument Short integers	
	) 0356 ) ; ; ; ; ; ; ; ; ; ; ; ; ;		SINI	JMP •ENDC •IFOF SETD LOCFD BR SETD LOCFD SETI MOV	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PREGISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS	
	) 0356 ) ; ; ; ; ; ; ; ; ; ;		SINI	JMP .ENDC .IFOF SETD LOCFD BR SETD LOCFD SETI MOV CLR	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,K0; R4;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PREGISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG	
	) 0356 ) 2 5 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7		SINI	JMP .ENDC .IFOF SETD LDCFD BR SETD LDCFD SETI MOV CLR CFCC	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0; R4; ;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PREGISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS	
	) 0356 ) 2 5 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7		SINI	JMP .ENDC .IFOF SETD LOCFD BR SETD LOCFD SETI MOV CLR	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,K0; R4;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PREGISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT	
	) 0356 ) 2 5 5 7 3 9 9		SINI	JMP .ENDC .IFOF SETD LDCFD BR SETD LDCFD SETI MOV CLR CFCC	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0; R4; ;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PREGISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG	
			SINI	JMP .ENDC .IFOF SETD LDCFD ADDD BR SETD LDCFD SETI MOV CLR CFCC BGE INC	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0; R4; ; P0\$\$37; R4;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PREGISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT	
			SIN: SNG\$37:	JMP • ENDC • IFOF SETD LDCFD BR SETD LDCFD SETI MOV CLR CFCC BGE INC ABSD	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0; R4; ; P0\$\$37; R4; F0;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN	
			SINI	JMP .ENDC .IFDF SETD LDCFD ADDD BR SETD LDCFD SETI MOV CLR CFCC BGE INC ABSD DIVD	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,K0; R4; ; P0\$\$37; R4; F0; (R0) +,F0;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2)	
			SIN: SNG\$37:	JMP .ENDC .IFDF SETD LDCFD ADDD BR SETD LDCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0; R4; ; P0\$\$37; R4; F0;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI)	
			SIN: SNG\$37:	JMP .ENDC .IFOF SETD LOCFD BR SETD LOCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD SETF	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,K0; R4; ; P0\$\$37; R4; F0; (R0)+,F0; ; #0,25,F0; ;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI) SINGLE PRECISION FP	
			SIN: SNG\$37:	JMP • ENDC • IFOF SETD LOCFD ADDD BR SETD LOCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD SETF LOCDF	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,K0; R4; ; P0\$\$37; R4; F0; (R0) +,F0;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI)	
			SIN: SNG\$37:	JMP .ENDC .IFOF SETD LOCFD ADDD BR SETD LOCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD SETF LOCDF CFCC	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,K0; R4; ; P0\$\$37; R4; F0; (R0)+,F0; ; #0,25,F0; ;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI) SINGLE PRECISION FP CONVERT ARGUMENT	
			SIN: SNG\$37:	JMP • ENDC • IFOF SETD LOCFD ADDD BR SETD LOCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD SETF LOCDF	<pre>@(R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37;; ; @2(R5),F0; ; #FC0\$37,R0; R4; ; P0\$\$37; R4; F0; (R0)+,F0; #0,F0; ; F0,F0;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI) SINGLE PRECISION FP CONVERT ARGUMENT CHECK FOR Ø FRACTION	
			SIN: SNG\$37:	JMP • ENDC • IFOF SETD LOCFD BR SETD LDCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD SETF LDCDF CFCC BEG	<pre>@ (R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0; R4; ; PU\$\$37; R4; F0; (R0)+,F0; ; F0,F0; ; F0,F0; ; RTN\$37;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI) SINGLE PRECISION FP CONVERT ARGUMENT	
			SIN: SNG\$37:	JMP • ENDC • IFOF SETD LOCFD BR TD LDCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD SETF LDCFD SETO MODD SETF LDCFD MODD SETO MODD SETO MODD SETO MODD SETO MODD SETO MODO SETO MODO SETO MODO SETO MODO SETO MODO SETO MOV CLR CFCC BGE MODO SETO MODO SETO MODO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO SETO	<pre># (R4) + # FPU # # # # # # # # # # # # # # # # # # #</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI) SINGLE PRECISION FP CONVERT ARGUMENT CHECK FOR Ø FRACTION FØ=FRACT(4+FRACT(X/2P)	
			SIN: SNG\$37:	JMP • ENDC • IFOF SETD LOCFD BR SETD LDCFD SETI MOV CLR CFCC BGE INC ABSD DIVD MODD SETF LDCDF CFCC BEG	<pre>@ (R4) + FPU ; @2(R5),F0; P12\$37,F0; SNC\$37; ; @2(R5),F0; ; #FC0\$37,R0; R4; ; PU\$\$37; R4; F0; (R0)+,F0; ; F0,F0; ; F0,F0; ; RTN\$37;</pre>	GET ARGUMENT COS(X) = SIN(X+PI/2) DOUBLE PRECISON FP GET ARGUMENT SHORT INTEGERS POINTER TO CONSTANTS SIGN FLAG: + ARG GET SIGN OF ARGUMENT SIGN FLAG: = ARG REMOVE ARGUMENT SIGN X/(PI/2) FØ=FRACT(X/2PI) SINGLE PRECISION FP CONVERT ARGUMENT CHECK FOR Ø FRACTION	

140

#### \$SIN04 MACRO VR04=14 07=SEP=72 11:43 PAGE 42+

145				BCC	0133371	JUMP IF FIRST O	R THIRD QUAD
146				NEGP	FOI		
147				ADUF	#1.0,F0;	Y=1.0-X	
148			013537:	ROR	R11		
149				BCC	012537;	JUMP IF FIRST O	R SECOND QUAD
150		7		NEGF	FUI	Y = -Y	
151				· · · · · · · · · · · · · · · · · · ·			
			012\$37:	LOF	F0,F2;		
152			MI5301.		F2,F21	Z=Y++2	
153				HULF		COUNT OF CONSTA	NTO END DALV
154				MOV	#4,811		
155				LDF	(R0)+,F11	INITIALIZE ACCU	MULAIOR
156			XPU\$371	MULF	F2,F1;	-	
157		1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	1	DEC		COUN	
158				ADDF	(RØ)+,F17	F11= Z*F1 + C(I	)
159				BGT	XPD5371	LOUP	
160				MULF	F1,F0;	FØ:= Y+F1	
161				TST	R41	TEST SIGN FLAG	
162				BEQ	RTN3371		
163				NEGF	FØT	SIN(=X) = =SIN(	X 3
164			RTN\$371	STF	FU,=(SP)1	MOVE RESULT TO	
-				MOV	(SP)+, RØ7		NCE TO RØ,R1
165						minur friu	NAC IN UNIUS
166				MOV	(SP)+,R11	10 M #	
167				RTS	R5;	EXIT	
168			1		in in general states and		
169			FCU\$371				
170			P125371	.WORD	040311,0077321	PI/2 (DOUBLE PR	ECISION)
171				.WORD	121041,0543021		
172			<b>1</b>				
173			2	ORDERED	EPENDENT CONSTAN	TS	
174			1			•	
175				. ENDC			
							the product of the pr
176	13 7 6 0		s Maria de La Agrico	.WORD	035036,153672;	00015148419	
177		035036		. NURU	00200011000151	*000101404*3	
	0362	153672					
178			<b>)</b>			12 mm + 12 mm - 17 41 15 mm - 18	
179	0364	136231		, WORD	136231,0231431	.00467376557	
	0366	023143					
180			7				
181	0370	037243		.WORD	037243,032130;	.0796896793	
7		032130					
182	~ .		1				
183	0374	140045	•	.WORD	140045,0567417	.645963711	
* 0 4	0376			च रा क∄्झा ्	்து சன்து பாது தன்றுமாற்றுக்கு இந்த	मुझी के प्राप्त के मान्य प्राप्त है। इसके उनके	a tha an thair an thai
184	0010	5997 MI	1				
	مى مەنەر يەرىخى	- A- 4 - 4		u non	040311.007733.	1,570796318	
185			CON\$371	• HURU	040311,007733;	18418198414	
	0402	007733		معر بر الم	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -		
186				.ENDC			
							the second se

141

A

ф

# STNH02 MACRU VR04=14 07=SEP=72 11:43 PAGE 43

1				TITLE	STNH02
2				. IFDF	CND\$48
3				GLOBL	TANH, EXP, SADR, SSBR, SMLR, SDVR, SFCALL
4				GLOBL	SPOLSH, SPSHR3
5		•	1		TRAN TANH FUNCTION
6			1	TANH	V002A
7			1		IT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS,
			1		
8				CALLING	SEQUNCE
9			l 🖡 🔬 🖓 🖓	JSR	R5, TANH
10			1	BR	A second s second second se Second second s Second second se
11			1	WORD	ARGUMENT ADDRESS
12			TAT		
13			1.	RETURNS	(EXP(2+ARG) =1)/(EXP(2+ARG)+1) IN R0,R1,
14			5 <b>1</b> - 5 - 5		
15		000000	,	R0=%0	
				- J	
16		000001		R18%1	
17		000004		R48%4	
18		000005		R5≡%5	
19		000006		5P = %6	
20		000007		PC=%7	
21	20404	010546	TANH	MOV	R5,=(SP) ;SAVE RETURN POINTER
22	20406	016505		MOV	2(R5),R5 JGET ARG AUDRESS
		000002			
23	20412	011500		MOV	ØRS, NØ IGET HIGH URDER ARG
24		001554		BEQ	ZERSUS JJUMP IF ARGED
25		006300		ASL	RO
26		105000		CLRB	RØ
-					
27	20422	000300		SWAB	
28	20424	020027		CMP	RØ,#205
		000205			الله الله الله الله الله الله الله الله
29	20430	002410		BLT	STESJ8 /JUMP IF ABS(ARG) <16.
30	20432	012700		MOV	#40200,R0 ;ANSWER IS 1.*SIGN(ARG)
	•	040200			
31	20436	005001		CLR	R1 And
		005715		TST	OR5 ITEST ARG SIGN
		002052		BGE	007538
				ADD	#100000,R0 ;MAKE =1.
34	20444	062700		AUU	47884881 198 1995
19 ath		100000		u n	0117#30
35	20450			BR	QUT\$38
36	20472		STES381	CMP	RU,#177
		000177			
		003007		BGT	TAN\$38 JJUMP IF >1/2
38	20460	020027		CMP	R0, #164
		000164			
39	20464	002043	24	BGE	SML\$38 JUSE CONTINUED FRACTION FOR THIS RANGE
		016501		MOV	2(R5),R1
79	*******	000002			
<b>A</b> 4	03170	0110002		MOV	<pre>@R5,RØ JIF ABS(X)&lt;2*+=12, LET TANH=X</pre>
				BR	OUTSUB
		000435	** • • • • • • •		
43	20476		TANS381	MOV	2(R5),=(SP) ;PUSH 2+ARG ON STACK
		000002			
		011546		MOV	QR5,=(SP)
45	20504	062716		ADD	#200,05P ;DOUBLE ARG
		00200			
46	20510	010605		MOV	SP, R5 ISET UP CALL TO EXP, ARG POINTER
		012704		MOV	HEXP,R4 JPOINT TO EXP
	*****	014554	1	- <b>**</b> *	
		<i>ઌ</i> ૢ ૡઌઌૡ	•		

5 T N	IHNS I	MACRU	VR04=14 K	7=357=72	11140 FAGE 40	<b>.</b>				19	<u>ر</u>
48	20516	00476	7	JSR	PC,SFCALL						
-0	20090	17440									
49	00520	01014		MOV	R1,=(SP)	;	PUSH	E**2ARG			
				MOV	RØ,=(SP)		• • •				
60		010040			-(SP) JPUSH	ы 1					
		00504		CLR		11 <b>A B</b>					
52	20530	01274		MOV	#40200,-(SP)						
		040201					-	C			
53	20534	01014	6	MOV	R1,=(SP)	e - 1 🖡	PUSH	E**2ARG			
54	20535	01004	5	MOV	RØ,=(SP)						
55		00504		CLR	= (SP)						
56		01274		MOV	#40200,-(SP)	1	PUSH	1.			
	6.99.16	040201									
a 19	OBBAR	00446		JSR	R4, SPOLSH	f	GET	(E++2X =1)	/(E*+2X	+1)	
57	20040			6 30	NALM CHO.	<b> </b>		Allow a set of all a			<u>j.</u> 6
	a sa an ƙ	00107		5. M m m		100 B	nvo :	101 \$ 3 8			
58		00200		. WORD	\$58R, UP538, \$/	AUNIO	UYKE	OFLUJA			4.1
		02075		and the second of				a sha ga cafee			÷
	20556	002011	1 1	· · · · ·				an iti ya sh			
	20500	01325	61								
		02056							1. N. 1.		
50			UPL\$381	MOV	(SP)+,RØ	1	POP	RESULT	a de la composición d		
50		01260		MOV	(SP)+,R1					1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	, <b>.</b>
					(SP)+,R5	1 <b>.</b>	REST	ORE RETURN	g se		1
61		01260				URN T					
62				RTS							÷ •
53	205/4	01650		MOV	2(R5),R1	· · · · •	GET	ARU			
		00000	2		A CARLES AND A CARL						
ŝ4	20600	01150	0	MOV	ers, Rø			• • • • • • • • • • • • • • • • • • • •			
55				JSK	R4, SPOLSH						
		00103			and the second second				N 6 4		
66	ONANA	01757		. WORD	SPSHR3, SPSHR	3. \$PS	HR3.	SMLR, XSQ53	8	IGET	XI
<i>~ ~</i>		01757							1	· · · · · · · · · · · · · · · · · · ·	
		01757									2.
		01716									
	20616					_					
67	20620	61664	6 XSQ\$381	MOV	2(SP),=(SP)	)	GET	X SQUARE			
		00000	2								
68	20624	01664	6	MOV	2(SP), = (SP)						
	7786	00000		<b>•</b> ·							
69	20630			JSR	R4, SPOLSH						
03	20000			000	D. L. B. M.						
		00101			P35538, SADR, 1	-	10	1 CFT - 11 P	NUMERAT	ent	
10		02073		. WORD	Paaskalauki	014-90	10	10H1 UP	NUCENNI	<b>W</b> IN	
		00201									
	20640	02066	ó 1								
71	20642	01757	91	. WORD	SPSHR3, P45\$31	8,\$PS	SHR3,	SDVR, SADR,	SADR, SDV	/R	
		02071		-							
		01757									
		01325									
		00201									
		00201									
		01325									
72		00200		. WURD	SSBRISHLR, UP	1 2 3 8					
	20662	01716	21								
		02056									
73	*******		1	THE ABO	VE COMPUTES X	(1=(0	(++35	)/(Y+45	+105.	. / ()))	
74			1	WHERE Y				*********			
	ORESE	81##R	0 ONES38		4(SP),RØ	9	GET	XSQUARE AG	AIN		
10	<0000	01600		± 11 <b>9</b> 1	- 280 - 1988			। र म्ला स्थान गर्मा विकास 	> t <del>ga</del> 7 ³		
19 ana		00000		MOV	6(SP),R1						
16	206/2	01600	1	MOV	o Couvint						

STNH02 MACRO VR04=14 07=SEP=72 11143 PAGE 43+

¥)

G

 $\left( \right)$ 

14

¢

(

143

STNH02 MACRO VR04=14 07=SEP=72 11:43 PAGE 43+

		000006			· · · · · · · · · · · · · · · · · · ·		
77	20676	005066		CLR	6(SP) /INSERT	A 1.	
		000000		· · · · · .			
78	20702	012766		MOV	#40200,4(SP)		
		040200					
		000004					n de la companya de
79	20710	000134		JMP	@(R4)+		
80	20712	012746	P455381	MOV	#136237,=(SP)	; PUSH	45.1842
		136237					
81	20716	012746		MOV	#41464,=(SP)		
		041464					
82	20722	012746	P155381	MOV	#165707,=(SP)	<b>IPUSH</b>	105,4605
		165707			- 		
83	20726	012746		MOV	#41722,=(SP)		
		041/22					
84	20732	000134		JMP	Ø(R4)+		
85	20734	012746	P355381	MOV	#116457,=(SP)	IPUSH	35,1535
		116457					
86	20740	012746		MOV	#41414,=(SP)		
		041414	. 4 Ter		• · · · · · · · · · · · · · · · · · · ·		
87	20744	000134		JMP	Ø (R4) +		
88	20745	005000	ZER\$381	CLR	RØ		
89	20750	005001		CLR	R1		
90	20752	000700		BR	OUTSJ8		
91			1				
92	20754	012066	UP\$381	MOV	(SP)+,10.(SP)	IMOVE	STACK ITEM
		000012				а. ¹	
93	20760	012066		MOV	$(SP) + , 10 \cdot (SP)$		
	Ng Sala	000012	1999 A. 1998	a san			
94	20704	000134		JMP	@ (R4)+		
95			1				
96				. ENOC			

UP

۰.0

7

×.

SATN03 MACRU VR04-14 07-SEP-72 11:43 PAGE 44

N.

б.

Å,

Q

			TITLE	SATNØ3
		1	.IFOF	CND\$49
		1	ATAN	VØØ3A
		COPYRI	GHT 1971	1, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MAS
		1		
			.GLOBL	ATAN, ATAN2;
			.IFNDF	FPU SADR, SSBK, SMLR, SDVR, SPOLSH, SPOPR31
			.GLOBL .ENDC	and 1200 in the later of all all all of
		1		TRAN ATAN AND ATAN2 FUNCTIONS
		1	JSR JSR	SEQUENCE FOR ATAN: R5,ATAN
		<b>j</b>	BR	A
		1	WORD	ARGUMENT ADDRESS
		JAT .	RETURNS	ARCTAN(ARG) IN RU AND R1.
		1		
		1	CALLING JSR	SEQUENCE FOR ATAN2: R5,ATAN2
		1	BR	A
		1	. WORD	ARGUMENT 1 ADDRESS
		1	.WORD	ARGUMENT 2 ADDRESS
		/ A 1	RETURNS	ACRTAN(ARG1/ARG2) IN RØ AND R1.
		7	IF ABS(	ARG1/ARG2) > 2++24, THE RESULT IS
		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	IF ABS() SIGN(ARC	ARG1/ARG2 > 2++24, THE RESULT IS G1)+PI/2.
		7 7 7 7	IF ABS() SIGN(ARC	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
		7 7 7 7 7	IF ABS() Sign(AR) IF ARG2 Sign(AR)	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000000 000000	7 7 7 7 7	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0=%0	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000000 000001 000002	7 7 7 7	IF ABS() Sign(AR) IF ARG2 Sign(AR)	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003	7 7 7 7 7	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003 000003	7 7 7 7	IF ABS( SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3 R4#%4	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003 000003 000004 000005 000005	7 7 7 7 7	IF ABS( SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3 R4#%4 R5#%5 SP#%6	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003 000004 000004 000005 000005 000005		IF ABS( SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3 R4#%4 R5#%5 SP=%6 F0#%0	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003 000004 000005 000005 000005 000000 000000		IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3 R4#%4 R5#%3 SP#%6 F0#%0 F1#%1	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003 000004 000005 000005 000000 000000 000002 000002 000003		IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3 R4#%4 R5#%2 SP=%6 F0#%0 F1#%1 F2#%2 F3#%3	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003 000004 000005 000005 000005 0000005 0000005 000000		IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3 R4#%4 R5#%2 F0#%0 F1#%1 F2#%2 F1#%1 F2#%3 F4#%4	ARG1/ARG2) > 2++24, THE RESULT IS G1)+PI/2, <0 THE RESULT IS ARCTAN(ARG1/ARG2) +
	000001 000002 000003 000004 000005 000005 000000 000000 000002 000002 000003		IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%3 R4#%4 R5#%2 SP=%6 F0#%0 F1#%1 F2#%2 F3#%3	ARG1/ARG2) > 2+*24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI.
20706	000001 000002 000003 000004 000000 000000 000000 000000 000000	ATAN21	IF ABS(/ SIGN(ARC IF ARG2 SIGN(ARC R0=%0 R1=%1 R2=%2 R3=%3 R4=%4 R5=%5 SP=%6 F0=%0 F1=%1 F2=%2 F1=%1 F2=%2 F4=%4 F5=%5 sIFNDF CLR	ARG1/ARG2) > 2+*24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI. FPU -(SP) JCLEAR SIGN FLAG
20706 20770	000001 000002 000003 000004 000005 000000 000000 000000 000000 000000	ATAN2 I	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0=%0 R1=%1 R2=%2 R3=%%3 R4=%4 R5=%5 SP=%6 F0=%0 F1=%1 F2=%2 F1=%2 F1=%2 F1=%2 F5=%5 .IFNDF CLR CLR	ARG1/ARG2) > 2+*24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI. FPU -(SP) JCLEAR SIGN FLAG -(SP) JCLEAR ATAN2 BIAS
20706 20770 20772	000001 000002 000003 000004 000005 000000 000000 000000 000002 000003 000003 000003 000003 000003 000003 000003 000004 000005 005046 005046	ATAN21	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0=%0 R1=%1 R2=%2 R3=%%3 R4=%4 R5=%5 SP=%6 F0=%0 F1=%1 F2=%2 F3=%3 F4=%4 F5=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5 SF=%5	ARG1/ARG2) > 2+*24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI. (1)*PI. FPU =(SP) JCLEAR SIGN FLAG =(SP) JCLEAR ATAN2 BIAS =(SP)
20766 20770 20772 20774 20775	000001 000002 000003 000005 000005 000000 000000 000000 000002 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000003 000000	ATAN21	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0=%0 R1=%1 R2=%%3 R4=%%3 F%5=%%6 F0=%%6 F0=%%6 F0=%%6 F1=%%1 F2=%%6 F1=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3	ARG1/ARG2) > 2+*24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI. (1)*PI. (SP) /CLEAR SIGN FLAG =(SP) /CLEAR ATAN2 BIAS =(SP) (SP) /CLEAR QUADRANT BIAS =(SP)
20706 20770 20772 20774 20776 21000	0000001         0000002         0000003         0000003         0000005         0000000         0000001         0000002         0000002         0000002         0000003         0000003         0000002         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         0000003         000003         000003         0000046         0000046         0000046         0000046         0000046         0000046         0000046         0000046         0	ATAN21	IF ABS(/ SIGN(ARC IF ARG2 SIGN(ARC R0=%0 R1=%1 R2=%2 R3=%3 R4=%4 R5=%5 SP=%6 F0=%0 F1=%1 F2=%2 F3=%3 F4=%4 F5=%5 .IFNDF CLR CLR CLR CLR	ARG1/ARG2) > 2+*24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI. (1)*PI. (SP) JCLEAR SIGN FLAG =(SP) JCLEAR ATAN2 BIAS =(SP) =(SP) JCLEAR QUADRANT BIAS
20706 20770 20772 20774 20776 21000	0000001         0000002         0000003         0000003         0000005         0000000         0000001         0000002         0000002         0000002         0000003         0000003         0000004         0000003         0000003         0000003         0000003         0000003         005046         005046         005046         005046         005046         005046         005046         005046	ATAN21	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0=%0 R1=%1 R2=%%3 R4=%%3 F%5=%%6 F0=%%6 F0=%%6 F0=%%6 F1=%%1 F2=%%6 F1=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3=%%6 F3	ARG1/ARG2) > 2+*24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI. (1)*PI. (SP) /CLEAR SIGN FLAG =(SP) /CLEAR ATAN2 BIAS =(SP) (SP) /CLEAR QUADRANT BIAS =(SP)
20706 20770 20772 20774 20776 21000 21004	000001         000002         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000002         000002         000002         000002         000002	ATAN2I	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0#%0 R1#%1 R2#%2 R3#%%3 R5#%%3 F1#%2 F3#%3 F1#%2 F1#%2 F1#%2 F3#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F5#%3 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2 F1#%2	ARG1/ARG2) > 2**24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*P1. FPU =(SP) /CLEAR SIGN FLAG =(SP) /CLEAR ATAN2 BIAS =(SP) =(SP) /CLEAR QUADRANT BIAS =(SP) =(SP) /CLEAR QUADRANT BIAS =(SP) 2(R5),R4 /GET FIRST ARG ADDRESS 2(R4),=(SP) /GET FIRST ARG
20706 20770 20772 20774 20776 21000 21004	000001         000002         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         000003         005046         005046         005046         005046         005046         005042         016446         000022         015446         000022         016446         000022         015446	ATANZI	IF ABS() SIGN(ARC IF ARG2 SIGN(ARC R0=%0 R1=%2 R3=%2 R3=%2 R3=%2 F1=%2 F3=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F5=%2 F	ARG1/ARG2) > 2**24, THE RESULT IS G1)*PI/2. <0 THE RESULT IS ARCTAN(ARG1/ARG2) + G1)*PI. G1)*PI. FPU -(SP) JCLEAR SIGN FLAG -(SP) JCLEAR ATAN2 BIAS -(SP) JCLEAR ATAN2 BIAS -(SP) JCLEAR QUADRANT BIAS -(SP) 2(R5J,R4 JGET FIRST ARG ADDRESS

#### SATN03 MACRO VR04=14 07=SEP=72 11:45 PAGE 44+

JGET SECOND ARG ADDRESS 56 21014 016504 MOV 4(R5),R4 000004 IGET SECOND ARG MOV 2(R4), = (SP)57 21020 016446 000002 58 21024 011446 MOV @R4, = (SP) OSP,R1 MOV JARG2 TO R1 59 21026 011601 JUMP IF DENOMINATOR IS Ø INF\$39 BEQ 60 21030 001437 IGET ABS VAL ARGI 61 21032 006300 ASL RØ IGET EXPONENT 62 21034 105000 CLRB RØ SWAB RØ 63 21036 000300 R1 64 21040 006301 ASL CLRB R1 JGET EXPONENT ARG2 65 21042 105001 SWAB R1 66 21044 000301 JGET EXPONENT DIFFERENCE 67 21046 160100 SUB R1, R0 #26, , RØ ICHECK MAGNITUDE 68 21050 022700 CMP 000032 BLT INF\$39 ITREAT AS INFINITY 69 21054 002425 R4, SPOLSH 70 21056 004467 DIV\$391 JSR 000562 . WORD SUVR, UPL\$39 JGET ARG1/ARG2 71 21062 0132561 21064 0210661 72 21066 005775 UPL\$391 TST 04(R5) /IF ARG2 >0, BIAS #0 000004 ATES39 JIF ARG2<0, BIAS#SIGN(ARG1)*PI 73 21072 002014 BGE 1PI 74 21074 012766 MOV #040511,8.(SP) 040511 000010 75 21102 012766 #007733,10. (SP) MOV 007733 000012 76 21110 005775 TST 02(R5) ITEST ARG1 000002 77 21114 002003 BGE ATES39 ADD #100000,8.(SP) 1 - PI 78 21116 062766 100000 000010 ISET CODES 79 21124 005716 ATE5391 TST PSP 80 21126 000426 AT1539 JJOIN MAIN ROUTINE BR #18, SP FLUSH STACK 81 21130 062706 INF\$391 ADD 000022 JANS = SIGN(ARG1)+PI/2 MOV #040311,RØ 82 21134 012700 040311 #007733,K1 83 21140 012701 MOV 007733 ITEST ARG1 TST 02(R5) 84 21144 005775 000002 JJUMP IF +PI/2 85 21150 002002 BGE INR\$39 1-PI/2 86 21152 062700 ADD #100000,R0 100000 INR\$391 RTS 87 21156 000205 R5 **TRETURN TO USER** 88 89 21160 005046 ATANE CLR . (SP) ICLEAR SIGN FLAG CLR . (SP) ICLEAR ATAN2 BIAS 90 21162 005046 CLR - (SP) 91 21164 005046 92 21166 005046 CLR - (SP) ICLEAR QUADRANT BIAS 93 21170 005046 CLR . (SP) 2(R5),R4 94 21172 016504 MOV JGET ARG ADDRESS

146

.

*

籔
SATN03 MACRU VR04=14 07=SEP=72 11:43 PAGE 44+

С.

Ċ

C

000002 IGET LOW ORDER ARG 95 21176 016446 MOV 2(R4),=(SP) 000002 @R4,=(SP) JGET HIGH ORDER 96 21202 011446 MOV PLUSSO JJUMP IF QUADRANT 1 OR 3 97 21204 002004 AT1\$391 BGE 98 21206 062716 #100000,0SP JGET ABS VALUE ADD 100000 12. (SP) IFLAG -99 21212 005266 INC 000014 100 1216 021027 PLUS39: CMP PSP, #40200 JCHECK IF <1. 040200 IJUMP IF <1. LE1539 101 1222 103431 610 102 1224 003003 BGT GT1539 1>1. ICHECK LOW ORDER 103 1226 005766 2(SP) TST 000002 104 1232 001423 BEQ LE1\$39 1=1. 105 1234 012766 GT1539: MOV #140311,4(SP) 1-P1/2 140311 000004 #007733,6(SP) JATAN (X) PPI/2=ATAN (1/X) 106 1242 012/66 MOV 007733 000006 DEC 12. (SP) LADJUST SIGN 107 1250 005366 000014 108 1254 016646 IMOVE ARG DOWN MOV 2(SP),=(SP) 000002 109 1260 016646 2(SP), = (SP)MQV 000002 110 1264 012766 #40200.4(SP) ; INSERT 1. MOV 040200 000004 111 1272 005060 CLR 6 (SP) 000006 R4, SPOLSH ICOMPUTE 1./X 112 1276 004467 JSR. 000342 113 1302 0132561 . WORD SDVRILE1\$39 1304 0213061 2(SP),=(SP) IMOVE ARG DOWN 114 1306 016646 LE15391 MOV 000002 2(SP),=(SP) 115 1312 016646 MOV 000002 4(SP) 116 1316 005066 CLR 000004 117 1322 005066 CLR 6(SP) 0000006 CMP @SP,#037611 1TAN(15) 118 1326 021627 037011 BLO L15539 JJUMP IF LESS THAN TAN(15) 119 1332 103445 120 1334 101004 BHI TNSS39 JJUMP IF > 121 1336 026627 CMP 2(SP),#030243 000002 030243 122 1344 101440 BLUS L15\$39 **JINSERT PI/6** 123 1346 012766 TNS\$391 MOV #040006,4(SP) 040006 000004 MOV 4005222,0(SP) 124 1354 012/60

147

SATN03 MACRO VR04-14 07-SEP-72 11:43 PAGE 44+

		005222		
		000006		
		011600	MOV	ØSP, RØ JARG TO REGS
126	1364	016601	MOV	2(SP),R1
		000002		
127	1370	012746	MOV	#131727,=(SP) ;PUSH =RUOT 3
		131727		
128	1374	012746	MOV	#140335, - (SP)
		140335		
129	1400	010146	MOV	R1,=(SP)
		010046	MOV	RØ,=(SP) /PUSH ARG
131	<b>.</b>	005046	CLR	=(SP) /PUSH 1.
132	1406	012746	MOV	#40200,=(SP)
		040200		n an the second sec Second second
133	1412	012746	MOV	#131727, *(SP) ; PUSH ROUT3
		131727		
134	1416	012746	MOV	#040335,=(SP)
		040335		
135	1422	010146	MOV	R1,=(SP) ;PUSH ARG
136			MOV	R0,=(SP)
137	1426	004467	JSH	R4, SPOLSH / TRANSFORM ARG
		000212		
138		₽		(ROOT3+X=1)/(ROOT3 +X)
139	1432	0171621	, WORD	\$MLR, \$\$BR, UP\$39, \$\$BR, \$DVR, L15\$39
	1434	0020041		
		0215361		
	1440	0020041		
	1442	0132561		
	1444	0214461		
		011000 L155391	MOV	ØSP, RØ IGET ARG
141	1450	016601	MOV	2(SP),R1
		000002		
142	1454	010146	MOV	R1,=(SP) JGET THREE COPIES
143	1456	010046	MOV	R0, = (SP)
144	1400	010146	MOV	R1,=(SP)
145	1462	010046	MOV	R0,=(SP)
146	1464	004467	JSR	R4, SPOLSH
	-	000154		
147	1470	0171621	WORD	SMLR JGET ARG*+2
		0215501	WORD	PLYS39 ISET UP COEFFICIENTS
149		0171621	. WORD	SMLR, SAUR, SMLR, SADR, SMLR, SADR
	1476	0020101		
	1500	0171621		
	1502	0020101		
	1504	0171621		
	1506	0020101		
150	1510	0171621	, WORD	SMLR, SAUR, SMLR, SAUR
	1512	0020101		
	1514	0171621		ne en e
	1516	0020101		
		0020101	.WORD	SADR /P(X)+0 IF X<#1, P(X)=PI/2 IF X>1
<i>y</i> .		0216041	. WORD	SGN3J9 /ADJUST SIGN
153	1524	0020101	. WORD	SADR FADD ATAN2 BIAS
154	1526	0176101	. WORD	SPOPR3, EXIS39 JPOP RESULT TO REGS
	1530	0215321		
155		005726 EX15391	TST	(SP) + JPOP SIGN FLAG
156	1534	000205	RTS	R5 JRETURN TO USER

148

(

÷.

.

(

Ú.

<u>24</u>

(

# SATN03 MACRO VR04=14 07=SEP=72 11143 PAGE 44+

ip-

(

(

È

Æ;

157		stan ing a sa 🖡 📲 🔤 👘		and a second and a s	and a second
158	1536	012666 UP\$391	MOV	(SP)*,10.(SP)	IMUVE STACK ITEM UP
		000012	1.0 Ma 1.4	4000 + 40 4000	
159	1542	012666	MOV	(SP)+,10.(SP)	
	at anna at 100	000012	Y at Th	0.00.43	
160		000134	JMP	@(R4)+	
161		Juna Radi Di Malica	45 /2 14	1001+ 8A	
162		012500 PLY5391	MOV	(SP)+,R0	JPOP POLY ARG
163		012601	MOV	(SP)+,R1 #CON\$39+4,R2	POINT TO COEFFICIENT TABLE
164	1554	012702 021644	HUY. S	HUUHUUUU	teatist in achertentist funder
165	1560	012703	MOV	#5,83 1LOOP 5	
100	1040	000005			
166	1564	and the second	BR	PY1539	
167			MOV	R1,=(SP)	PUSH ARG
168			MOV	R0,=(SP)	
169			MOV	-(R2),-(SP)	PUSH CONSTANT
170		014246	MOV	-(R2),-(SP)	
171			DEC	R3 ICOUNT	
172	1600	003372	BGT	PY2539	
173	1602	000134	JMP	@(R4)+	
174		1			
175	1604	005766 SGN\$391	TST	8.(SP) ICHECK	SIGN FLAG
		000010			
		001402	BEQ	SG1539	na ann an Anna ann an Anna an A
177	1612	062716	ADU	#100000, •SP	INEGATE RESULT FOR (-1,0) & (1,1
		100000			
		000134 SG1539:	JMP	@(R4)+	and the second
179			.ENDC		
182			<b></b>	Meridian 1	
181		5 <b>- 5 - 5</b>	.IFOF	FPU	SET FP MODE FOR FPU
182		ATAN21	SETF Mov	1 2(R5),R31	ADDRESS OF ARG1
183			MOV	4(R5),R41	ADDRESS OF ARG2
185			MOV	eR3, R01	HIGH ORDER ARG1
186			MOV	0R4, R17	HIGH ORDER ARG2
187			BEQ	INF5391	JUMP IF DENOMINATOR Ø
188		•	ASL	RUI	יישר אותיייניער אותי איז איז איז איז איז איז איז איז איז אי
189			CLRB	RUI	and the second
190			SWAB	RØJ	EXPONENT OF ARG1
191		and a strategy of the second	ASL	R1;	
192			CLRB	R17	
193			SWAB	R11	EXPONENT OF ARG2
194			SUB	R1, R0;	GET EXPONENT DIFFERENCE
195	i	$\mu_{1} = 1$ ( $\lambda_{1} = 0$	CMP	#26,1RØ1	CHECK MAGNITUDE
196	<b>)</b>	a na si	BLT	INF\$39;	TREAT AS INFINITE
197	r .		LDF	PI\$39,F31	INITIALIZE BIAS=PI
198			LDF	PR3, FOI	GET ARG1
199			CFCC	· · · ·	
200			BGE	A1P5391	JUMP IF ARG1>0
201			NEGF	F31	BIAS=SIGN(ARG1)+PI
202		A1P5391	LDF	@R4, F1;	GET ARG2
203			CFCC	1040301	
204			BLT	A2M3391	TE ARCAND RTAS-A
205		A in hit or if it a	CLRF	F31	IF ARG2>0, BIAS#0 Arg1/Arg2, set float CC
206		A2M5391	DIVF	F1, F0;	JOIN MAIN ROUTINE
207			BR	AT13391	医医骨髓 印刷集合 计图验生命接触

	1			
	INF\$391	LDF	P12539, F1;	RESULT=SIGN(ARG1)+PI/2
	HIT DWW R	TST	eR31	TEST ARGI
		BGE	EX1539;	+PI/2
		NEGF	F11	-PI/2
		BR	EX1839;	
		PR.	FUT0A31	
	/ /	20170		SET FP MODE FOR FPU
	ATANE	SETF		
		CLRF	F3;	CLEAR ATAN2 BIAS
		LDF	02(R5),F01	GET ARGUMENT
	AT15391		R41	CLEAR SIGN FLAG
		CFCC	1 Alexandre States	GET SIGN OF ARGUMENT
		STF	F3,FD1	FSFATAN2 BIAS
		CLRF	F31	CLEAR QUADRANT BIAS
		BGE	PLU\$391	JUMP IF QUADRANT 1 OR 3
		ABSF	FØJ	ABS(X)
		INC	R41	FLAG -
	PLUS391	LDF	#1,0+F11	1. <b>1. 0</b>
		CMPF	FØ,F1;	CHECK IF X<=1.0
		CFCC		
		BLE	LE15091	
	GT15391	DEC	R41	X>1.0, ADJUST SIGN FLAG
		DIVF	F0,F1;	1.0/X
		LDF	F1,F0;	ATAN(X) =PI/2=ATAN(1/X)
		LDF	P12539,F31	QUADRANT BIAS=PI/2
	1		ि समय प्रमाण गर्भ भाषा प्राप्त के गाँउ का	
the second s	LE15391	STF	F3,F41	F4=QUADRANT BIAS
		CLRF	FSI	F3=0,0
		CMPF	T15539,F0;	COMPARE TAN(15) : X
		CFCC	1129031601	
			1 1 8 8 3 8 .	X48 TAN (15)
		BGE	L15539;	
	an a	LDF	P16539,F31	F3=F1/6
		LOF	F0, P1;	
		MULF	RT3539, FØ1	and the second
		SUBF	#1,0,F07	X+R00T3=1.0
		ADDF	RT3539,F1;	X+RODT3
and the second		DIVE	F1,F0;	(X*ROOT3=1.0)/(X+ROOT3)
	1			
	L155391	LOF	F0,F2;	X
		MULF	F0,F0;	X**2
		MOV	#FC0\$39,R0;	POINTER TO POLYNOMIAL CONSTANTS
		MOV	#4,R1;	COUNT OF CUEFFICIENTS
		LOF	(RØ)+,F17	INITIALIZE ACCUMULATOR
na sa	XPD\$391	MULF	FU,F1;	
		DEC	R1/	COUNT
		ADDF	(RØ)+,F11	F1## F1# X##2 * C(I)
	a she she shin. A	BGT	XPD\$391 LOOP	
	나는 옷을 가려서 있는다.	MULF	F2,F1;	F1#= F1+X
		ADDF	F3,F1;	PI/6 OR 0.0
		SUBF	F4,F1;	P(X)-QUAD BIAS
		TST	R41	TEST SIGN FLAG
		BEQ	SG18397	NO ADJUSTMENT
	an	NEGF	F17	NEGATE RESULT FOR (-1,0)&(1,INF)
	SG15391	ADDF	F5,F1;	ATAN2 BIAS
	I sign a spec			
	EX1\$391	STF	F1,=(SP)/	MOVE RESULT TO STACK
	<b>H V H D V D H</b>	MOV	(SP)+,R01	AND THEN TO REGISTERS

150

ŝ

SATN03 MACRO VR04=14 07=SEP=72 11:43 PAGE 44+

15

đ,

4

Ø

265				MOV	(SP)+,R11	
266				RTS	R51	EXIT
267			7			
268	4.55		P15391	, WORD	040511,007733;	PI
269			P125391	. WORD	040311,007733;	PI/2
270			T155391	. WURD	037611,030243;	TAN (15)
271			P16\$391	. WURD	040006,0052221	P1/6
272			RT35391	WORD	040335,131727;	ROUTS
273				.ENDC		
274	1620	037305	FCU\$391	.WURD	037305,035302	1.0963034789
	1622	035302				
275	1624	137421		.WORD	137421,056514	11419574624
	1626	056514				
276	1630	037514		.WORD	037514,143333	1.1999773201
	1632	143333			•	•
277	1634	137052		, WORD	137652,125244	1=.3333331319
	1636	125244				
278	1640	040200	CON\$391	. NORD	040200,000000	1,99999999999
	1642	000000	•			
279			1			
280				. ENDC		

		.TITLE .GLOBL	SPOL07 SPOLSH		
		) ; spolsh - IS C		T IS DESIRED TO EN	TER POLISH
		I MODE FR	OM INFLING CODE. Be called via a	JSR R4, SPOLSH	
		I SPOLSH	V007A		
		/ / COPYRIGHT 197	1, 1972, DIGITAL	EQUIPMENT CORPORA	TION, MAYNARD
	000004				
	000000	SPEXO			
		J THE FOLLOWING	ENTRY POINT IS	A DIRTY TRICK TO D	EFINE
		THE OTS	VERSION NUMBER		P TO ALLOW
			CULAR LINK.		
				S TO TELL US WHAT	VERSION HE
		IS USIN	G, ALL IS NOT LO		
21644		.GLOBL Sv20A1	SV20A JTHIS I	S VERSION 20A	
		1	(50)+	IDELETE JUNK FROM	STACK
21644 21646	005725 000134	SPULSH: TST JMP	(SP)+ @(R4)+	JWEIRE NOW IN POL	
		I THE FOLLOWING	CONDITIONALIZED	ENTRY POINTS IDEN	TIFY THE ASSE
			USED IN THIS VE	RSION OF THE OTS,	
		.IFOF .GLOBL	SEAE		
		SEAEI			
		. ENDC	man (Calabas) (1 m. Tak		
		. IFDF	EISIMULDIV		
		.GLOBL	SEIS		
		,ENDC			
		. IFOF	FPU		
		.GLOBL	SFPU		
		SFPUI			
		, ENOC			
		. IFOF	FIS		
		.GLOBL	SFIS		
		SFISI			
		.ENDC	Bev		
		.IFOF .GLOBL	RSX SRSX		
		SRSXI	<b>•</b> • • • •		

重

1				.TITLE	\$ SQ T 1/2 3
2				IFOF	CND\$41
3			1		
Å			1	SORT	V003A
5			i -		
6			I COPYR	IGHT 197	1, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MA
7			1		
8				CLARI	SQRT, SERR;
9				.GLOBL	PPU
10				, IFNDF	SADR, SDVR, SPOLSH;
11				.GLOBL .ENDC	gwaktgat, falwfallt
12				SQRT	THE REAL SQUARE ROOT FUNCTION
13			<i>y</i>		
14			1		SEQUENCE R5, SQRT
15				J-SR Here	
16				BR	
17			1	#ARG	
18			1 A 1	OFTHONS	THE SOULDE DOAT IN DO AND PA
19				REIURNO	THE SQUARE ROOT IN RØ AND R1.
20		adminan	*	Bri - 9/14	
21		000000		RØ=%0	
22		000001		R19%1 R49%4	
23		000004		R4=%4	
24		000005		R5#%5	
25		000000		5P=%6	
26		000000		FØ=%0	
27		000001		F1=%1	
28		000002		F2=%2	
29				.IFDF	FPU
30			SORTI	MOV	02(R5),R1; GET HIGH ORDER ARGUMENT
31				.ENDC	
32	01660		CAUT!	. IFNDF	FPU PS _ (SP)
33		010546	SUNI	MOV	R5,=(SP) 2(R5),R5
34	51025	016505		MQV	STRONING TOTT ARGAINAT ADDIEGO
30		000002		MAN	ØR5,R1 JGET HIGH ORDER ARGUMENT
	51050	011501		MOV	@R5,R1 /GET HIGH ORDER ARGUMENT
36		1 (3 M A 4 4		.ENDC	ERR\$41 JERROR IF ARGUMENT NEGATIVE
		100443		BMI BEQ	ERR\$41 JERROR IF ARGUMENT NEGATIVE Zer\$41 JFAST Exit if Zero
	51005	001446		, IFNDF	FPU
39	01460	410745		MOV	#3,=(SP) ;PUSH ITERATION COUNT
4 Ø	21004	012746 000003		sa 🖓 A	AATAFOLD - LEADIE TIEEELTDIE POOLI
A 4		0000000		.ENDC	
41	0147-	TARA CON 4		ASR	R1 FORM INITIAL ESTIMATE
42	210/0	006201 062701		ADD	#20100,R1
	61016	020100			FT ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩
A 4	01676			CLR	. (SP) JUSE ONLY HIGH ORDER PARTS FIRST
		005046		MOV	RI, (SP) ; CAUSE ADD AND DIVIDE ARE
	€ ¥ / ₩Ø	819140		.IFNDF	FPU
46	01420	11 DA 12 DA 4 4			• (SP) IFASTER THAT WAY
		005046		CLR S	
		011546		MOV	0K5, = (SP) _(SP)
		005046		CLR	₩(SP) Di _icpi
		010146	م د ادر مو دنور با	MOV	RÍ,=(SP) R4,\$POLSM
31	21/12		LUPS411	JSR	R4, SPOLSH JENTER POLISH MODE
F		177720		unne	
<b>þ</b> 2		013256		. WORD	SDVR,SADR,UPLS41 J(X/E+E)
	21720	002010	t.		

(

## \$\$QT03 MACRO VR04=14 07=5EP=72 11:43 PAGE 46+

		021724			-		· · · · · · · · · · · · · · · · · · ·	
53	21724		UPLS411	SUB	#200,05P	,	; (X/E+E)/2	
		000200						
54	21730	005366		DEC	4(SP)	ICOUNT	_00P	
•		000004						
5 6	A1734			BEQ	OUTS41			
55	21734	001410					JUSE LOW ORDER P	
56	21736	016546		MOV	2(85),=(	[ ] ]	JUGE LUN UNDER P	"MICLO
		000002					i i internet i internet de la companya de la	
57	21742	011546		MOV	@K5,=(SP		TOO FROM NOW ON	4
58	21744	016646		MOV	6(SP),=(	(SP)		
		000000						
59	21750	016545		MOV	6(SP),=(	(SP)		
		000000						
60	21754	000756		BR	LUPS41	JGO FOR	ANOTHER ITERATIC	) N
61	21756	012000	OUTS411	MOV	(SP)+, RE	)	JGET RESULT INTO	
62	21700	012601	· · · · · · · · · · · · · · · · · · ·	MOV	(SP)+,R1			
63	21762	005726		TST	(SP) +		ERATION COUNTER	
64	21764	012505	RTNS411	MOV	(SP)+,85			
			01040444	RTS			TO CALLER	
65	21706	000205	Provide Add		R5			
66	21770	004567	ERK\$411	JSR	R5, SERR	IERRUR A	4/14	
		000012		53 m				
67	21774	000773		BR	RTN\$41			
68	21775	004		BYTE	4			
69	21777	013		BYTE	11.			
70	22000	005000	ZER5411	CLR	RØ			
71	22002	005001		CLR	R1			
72	22004	000767		BR	RTN\$41			
73				.ENDC				
74				IFOF	FPU			
				Mov	#3,R01		ITERATION COUNT	
75							SINGLE PRECISION	. 50
76				SETF	1			
77				LDF	(SP)+,F0		GET INITIAL ESTI	TAIL
78				LOF	02(R5),F	- 21	GET X	
79			1					
80			LUPS411	LOF	F0,F1;		E=EI	
81				LDF	F2,FØ;		X	
82				DIVF	F1, F01		X/E	
83				ADDF	F1, FU;		X/E+E	1999 - B. 1999 - A.
84				DEC	RØJ		COUNT	
85				DIVE	#2.0,F01	n de la company	E1=(X/E+E)/2	
86				BGT	LUP\$411			
87			1		and and and the first			
88			e	STF	FØ,=(SP)	1 2	RESULT 1	TO STACK
				MOV	(SP)+,R0		AND THENCE TO RE	
89				MOV	(SP)+,R1		त्याप्रका राष्ट्रकार्यकार्थका रिक्स विद्यु	F F 15 A
90						1. P	EXIT	
91				RTS	R51		Im A R I	
92			I	Lon	ne elenna	•	EDDAD 4 14	
93			ERR\$411	JSR	R5, SERR;	ī	ERROR 4,11	
94				RTS	R51		EXIT	
95				BYTE	4			
96				BYTE	11.			
97			ZER\$411	CLR	RØJ			
98				CLR	R17	an an ta		
99				RTS	R51			
100	<u>,                                     </u>			.ENDC				
101				ENDC				
÷۲.	•			in and is π4, 444.				

154

£

٠.

 $\left( \right)$ 

œ

C

(

 $\left( \right)$ 

6

£

12				.TITLE .GLOBL	SERRØ1	RA, SERVE	r <b>c</b> 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997
3		000000		R0 = X0			•
4		000005		R5 = %5			
		000000		5P = %6			<ul> <li>A start of the sta</li></ul>
5 6 7		000007		PC = %7			
7 8					R HANDLE		
9			1				ROL TO THE USER'S ERROR
10			1.000		IF ANY.		T ACTION IS TO HALT.
11			1				F HIS ERROR ROUTINE Vec'. Control Is passed
12			🕴 i kaupana	TO FUE	AL LOCATI Address I	N SEDVER	VIA A JSR PC, #SERVEC.
13 14			;				IN THE ERROR CODE.
15 16			J	CALLING	SEQUENCE		
17					100		
18			. <b>1</b>		JSR BR	R5,SERR- A	
19 20			1		BYTE	ERROR CL	158
20			2		BYTE	ERROR NL	
22			1	A:	9	*****	
23			•				
24			1	ŨR:			
25							
20			1		MOV	HERRNUM,	RÓ
27			1		JSR	R5, SERRA	
28						_	
29		010046	SERRI	MOV	RØ, - (SP)		SAVE RØ
30	22010	015500		MOV	2(R5),RØ	1	GET ERROR CLASS/NUMBER
-		000002		i'l in			
		000401		BR	ERB\$43		A 115
		010048		MOV	RØ,=(SP)		SAVE RO
	22020	100007	ER55431	.IFNDF CMPB	CLASS51 RØ, #51		DEFINE TO GET WARNINGS CLASS 5 (WARNING)?
34	22020	120027		LMED	KØ1491		CTW00 0 (WWKWING)1
7.6	00004	000005 001402		BEG	IGN5431		IGNORE IF SO
35	66069	00140 <b>6</b>		.ENDC	TANG-91		VALATE TL AA
37	00006	004777		JSR	PC, @SERV	Fr.	CALL USER ERR ROUTINE
97	55050	000004		* 311	r wy ww ch y	-*/	ine manana bertekan minist tibelekan bertekan. 
38	22032	012000	IGN\$43:	MOV	(SP)+,R0	1	RESTORE RØ
		000205	ज्या स्था वि	RTS	R51	• •	RETURN TO ERROR ROUTINE
			SERVECT	.WORD	HLT8431		ADDR OF USER ERR ROUTINE
41	22040	000000	HLT\$431				DEFAULT: HALT
		000775		BR	HLT\$431		HARD STOP

### SLOR01 MACRO VR04=14 07=SEP=72 11:43 PAGE 48

4 ×	100004 100005	•TITLE \$LOR01 •IFDF CND\$44&CND\$42 R4=%4 \$P=%6	
5 6 7	1	LOAD FLAC - SINGLE PRECISION	
	12666 SLUR	MOV (SP)+,2(SP); MOVE OPERAND TO RESULT LOC	
	12666	MOV (SP)+,2(SP)	
	060134	JMP @(R4)+; POLISH RETURN .ENOC	

156

⑲

).

Ś.

Ŧ

#### MACRO VR04=14 07=SEP=72 11:43 PAGE 49 SLDD01

12				.TITLE .IFDF	SLDD01 CND345&CND342	
3		000000		ROFXO		
4		000004		R4=%4		
5		000006		SPE%6		
6						
7			1	LOAD FL	AC - DOUBLE PREC	CISION
9	022056	010600	SIDDE	MOV	SP, RØ;	COPY STACK POINTER
	22060	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		ADD	#8.,R0;	CALC ADDR OF RESULT
11		012620		MOV	(SP)+,(R0)+; (SP)+,(R0)+	MOVE OPRAND TO RESULT LOC
	22070		gan a goran	MOV	(SP)+,(R0)+ (SP)+,(R0)+	
15	22074			JMP . ENOC	@(R4)+;	POLISH RETURN

فد

C.

 $\mathcal{Q}$ 

T.

12			.TITLE .IFDF	55TRØ1 CND5468CND542	
3		000000	K0=%0		
4		1000001	R1=%1		
5	•	000002	R2=%2		
6		000003	R3=%3		
7		000004	R4=X4		
8		000005	R5=%5		
9		000006	SP=%6		
10		000007	PC=X7		the second se
11					
12 13			STORE FI	LAC - SINGLE PRE	
14	22078	012705 \$STR:	MOV	#FAC\$42,851	GET ADDRESS OF FLAC
		0004321			
15	22102	005766	TST	30(SP)1	TEST FOR STACK MODE
		000030			
16	22106	001415	BEQ	STK\$461	BRANCH IF NOT
17	22110	005066	CLR	30(SP)1	CLEAR STACK MODE FLAG
		000030			CADU OFICH AATMEEA
18		010000	MOV	SP,RU;	COPY STACK POINTER
		010601	MOV	SP,R1	
		022121	CMP	(R1)+, (R1)+;	R1 = R1 + 4 Loop count
21	22122		MOV	#15,R21	
		000013	MOV	(DI)+ (D0)++	MOVE UP STACK TO MAKE ROOM
22			MOV	(R1)+, (R0)+)	NUTE OF STRUK TO MAKE ROOM
	22130	005302	DEC	R2	
24	22132	001375	BNE	LPS40	መልተበለኑ
25				TS TO OPERAND LU	STORE THE FLAC
26	22134		MOV	(R5)+,(R0)+;	STAKE THE LEAD
27		012520	MOV	(R5)+,(R0)+	POLISH RETURN
28	22140		JMP	@(R4)+; (88)+ (80)++	STORE THE FLAC
29	22142		MOV	(R5) +, (R0) + j	STAKE THE LEVE
30		012520	MOV	(R5)+,(R0)+ (SP)+,(SP)+;	POP OPERAND OFF THE STACK
31		022626	CMP		EAL ALEVANA ALL IDE OLARN
32	22150	000134	JMP	@(R4)+	
33			.ENDÇ		

# 158

2

Ľ

### SSTD01 MACRO VR04=14 07=SEP=72 11143 PAGE 52

ſ

**S***

 $\langle \hat{Z} \rangle$ 

Ľ

.TITLE \$STD01 1 ž CND\$478CND\$42 . IFDF 3 K0=%0 000000 4 000001 R1#%1 R2=%2 5 000002 6 000003 K3#%3 7 R4=%4 000004 8 R5=%5 000000 9 000006 SP#%6 000007 PC=%7 10 11 12 22152 012705 \$STD1 MOV #FAC\$42,85 0004321 13 22156 005766 TST 34(SP)1 TEST FOR STACK MODE 000034 BEQ 14 22162 001420 STK\$471 BRANCH IF NOT 15 22164 005066 CLR 34(SP) 000034 16 22170 010600 MOV SP,RØ 17 22172 010001 MOV SP,R1 18 22174 062701 ADD #10,R1 000010 19 22200 012702 MOV #15,82 000013 (R1)+, (R0)+ 20 22204 012120 LP\$47; MOV DEC R2 21 22200 005302 LP\$47 22 22210 001375 BNE STORE THE FLAC MOV (R5)+, (R0)+; 23 22212 012520 24 22214 012520 MOV (R5)+,(R0)+ 25 22216 012520 MOV (R5) +, (R0) +MOV (R5)+, (RØ)+ 26 22220 012520 27 22222 000134 JMP @(R4)+; RETURN 28 22224 012520 STK\$471 MOV (R5)+,(R0)+ MOV (R5) +, (R0) + 29 22226 012520 30 22230 012520 MOV (R5)+,(R0)+ MOV 31 22232 012520 (R5)+, (R0)+ ADD #10, SP; POP OPERAND 32 22234 062706 000010 JMP 33 22240 000134 @(R4)+ . ENDC 34 TITLE FPMP11 FLOATING POINT & MATH PACKAGE 35 36 37 0000011 . END

FPMP11 FLUATING POINT & MATH PA MACRU VR04-14 07-SEP-72 11:43 PAGE 52+ / SYMBOL TABLE

A =	600010		ABP\$19	014432R		AC s	177302
ADJ519	Ø14344R		ADR\$42	000202R		AINT	003124RG
AI154	003150R		ALOG	002550RG		ALOG10	002544RG
ARN\$19	014354R	- 4	ASH =			ASL 511	007574R
	003216R		ASR\$19	014254R		AS1511	007552R
ASLS4							011072R
ATAN	021160RG		ATAN2	Ø20766RG		ATES15	
ATES39	021124R		AT1515	011204R		AT1539	021204R
AUPS19	014410R		A1 🛢	MC 200 Mile 241 and 144		A1Z51	001046R
A2 #	000010		A2158	003656R		A2N\$1	001054R
A2N82	002106R		6 =			BAC\$25	Ø15634R
BEXP =			BT9\$1	001606R		87952	002464R
			82 8			B2N528	016404R
				016410R		B22530	017300R
B2N\$30	017264R		B22328				
B4N528	016346R		842528	Ø16302R		B6Z\$28	Ø16332R
B9A\$1	001666R		89A\$2	002404R		CAD\$42	000476R
CAR542	000452R		CFR520	014752R		CHK\$17	013246R
CLE\$9	005144R		CLR\$29	017050R		CMD342	000466R
CMF 542	000254R		CMR\$42	000442R		CM1542	000264R
CNDs1 =			CND\$10=			CNDS11=	000001
							000001
CND\$12			CND\$133				
CND\$15			CN0316=			CND\$17=	000001
CND\$18=			CND519=			CND32	000001
CND\$20	000001		CND\$21₽	000001		CND\$22=	000001
CND523	000001		CND\$24=	000001		CND\$25=	000001
CND\$26=			CND327=			CND\$28=	000001
CND\$29=			CND\$3 .			CND\$30=	000001
CND\$31			CND532=	000001		CND\$33=	000001
			CND\$35=			CND\$36=	000001
CND\$34							
CND\$37=			CND\$38=			CNUS39=	
CND54	000001		CND541=			CND\$42=	
CNDS44=	000001		CND\$45∓	000001		CND546=	000001
CND\$47	000001		CND\$5	000001		CNDS6 #	000001
CNDS7 #			CNDS8 =	000001		CND\$9 =	000001
CNV\$8	003444R		CONSIU	007440R	•	CONS13	010444R
CONS15	Ø12200R		CON\$3	003120R		CON\$37	020400R
			COS	020030RG		COV\$30	017506R
CON\$39	021640R						000022
CYC\$29	017062R		C1 =	and the second		C2 =	· · · · ·
C23\$11	007514R		D =	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -		DATAN	Ø11136RG
DATANZ	010666RG		DBLE	003416RG		DCHS16	Ø12506R
DCH318	013442R		DCOS	007674RG		DCRS25	Ø15666R
DEC\$25	015772R		DEC\$35	017724R		DEXP	013710RG
DGS\$9	006552R		DG159	006570R		DG259	000602R
06339	006610R		DHIS16	012504R		DHIS18	013514R
-		•				DIVS15	011010R
DIGITS			DIG\$9	006632R			
DIV\$17	013176R		DIV339	021056R		DIVS8	004310R
DIV\$9	005306R		DLOG	006654RG		DLOG10	006650RG
DLWS16	Ø12552R		DLWS18	013506R		DMODE =	100000
DNES11	007604R		DNES24	015456R		DNE\$25	Ø15724R
DNES35	017730R		DNES4	003226R		DNES9	000022R
DN154	003234R		DOUBLES	000001		DPRSB	004610R
	007640R		DR2\$12	007606R		DSIN	007752RG
DR1512						DUPS10	007246R
DSQRT	010454RG		DSV\$19	014402R			
DUP\$13	010154R		DUP\$19	014454R		DUPS20	Ø1472ØR
DUP\$3	003014R		DUP\$37	020222R		DV1310	012772R
DV1518	013630R		DV138	004326R		DV159	005344R
DV238	004314R		Dv259	005350R		DV3\$8	004346R
DV458	004366R		D1 =	000014		01058	004024R
9 · · 98 C	· · · · · · · · · · · ·						

160

(

£

Ð

(

2

Ħ

FPMP11 FLOATING POINT & MATH PA MACRU VR04-14 07-SEP-72 11143 PAGE 52+ Symbol Table

E

C T

Ľ

2

016519	014370R		02 = 000024	D6R\$19 014400R	
ECKS1	001232R		ECK\$2 002212R	ECL\$16 012520R	
ECLS18	013466R		ECL519 014100R	ECL\$20 015112R	
	016620R		ECL\$30 017446R	EC1\$16 Ø12522R	
ECLS28					
EC1\$18	013474R		EEXP = 000006	EFES9 005734R	
EFTS9	005646R		EF158 004176R	EF258 004204R	
END #	000044	. A	ENM58 004212R	EN188 004224R	
EN258	004274R		EQ1516 013120R	E01518 013700R	
EQ2516	013116R		EQ2318 013676R	ER5543 022020R	
ERF			ER0\$10 007352R	ERR510 007130R	
	010644R		ERR\$24 015474R	ERR\$25 0157408	
ERRS14				ERR\$42 000422R	
ERRSJ	003070R		ERR\$41 021770R		
ERRSB	004074R		ERR\$9 006276R	ESIGN = 000010	
ESV\$20	014744R		EXAS1 001122R	EXA52 002144R	
EXCS8	004114R		EXI510 007312R	EXIS15 011770R	
EXIS3	003040R		EXIS39 021532R	EXP 014554RG	
EXP\$9	005770R		EXT38 004130R	EX159 005774R	
EX259	006012R		EX359 006000R	FAC\$42 000432R	
	012100R		FC0139 021620R	FFES9 005464R	
FCOS15				FF489 005564R	
FFTS9	005460R		FF339 005600R		
FF559	005544R		FF639 005626R	FIL\$25 015700R	
FLOS24	Ø15436R		FLD\$8 003712R	 FLOAT 015202RG	
FLTS16	012660R		FLT\$18 013550R	FLT38 004454R	
FL1516	012656R		FPS35 003202R	FP556 003360R	
FULS25	015712R		FØ = = % 0 0 0 0 0	F1 =x000001	
			F3 =%000003	F4 =%000004	
	2000002				
	%000005	· · · · · ·	GO\$16 013046R	G0\$18 Ø13654R	
GO\$24			1.1.5.25 0.1.5540.00	LITI 6 1 M	
	015234R		G0\$25 Ø1554ØR	GT1\$15 011250R	
GT1539	021234R		HLT\$43 022040R	IDINT 016020RG	
GT1539	021234R		HLT\$43 022040R		
GT1539 IFIX	021234R 015154RG		HLTS43 022040R Igns43 022032R	IDINT 016020RG INC\$20 014712R	
GT1539 IFIX INF515	021234R 015154RG 011076R		HLT\$43 022040R IGN\$43 022032R INF\$39 021130R	IDINT 016020RG INC\$20 014712R INR\$15 011134R	
GT1539 IFIX INF\$15 INR\$39	021234R 015154RG 011076R 021156R		HLT\$43 022040R IGN\$43 022032R INF\$39 021130R INT 016020RG	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R	
GT1539 IFIX INF515 INR539 ISR59	021234R 015154RG 011076R 021156R 006526R		HLT\$43 022040R IGN\$43 022032R INF\$39 021130R INT 016020RG L = 000024	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 006506R LENGTH= 000042	
GT1539 IFIX INF\$15 INR\$39 ISR\$9 LE1\$15	021234R 015154RG 011076R 021156R 006526R 011356R		HLT\$43 022040R IGN\$43 022032R INF\$39 021130R INT 016020RG L = 000024 LE1\$39 021306R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R	
GT1839 IFIX INF\$15 INR\$39 ISR\$9 LE1815 LGT\$10	021234R 015154RG 011076R 021156R 006526R 011356R 007342R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INT       016020RG         L       = 000024         LE1\$39       021306R         LGT\$3       003000R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R	
GT1339 IFIX INF\$15 INR\$39 ISR\$9 LE1\$15 LGT\$10 LOG\$3	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INT       016020RG         L       # 000024         LE1\$39       021306R         LGT\$3       003000R         LP\$\$25       015562R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 006506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 ISR\$9 LE1515 LGT\$10 LOG\$3 LP\$47	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R 022204R		HLT\$43 022040R IGN\$43 022032R INF\$39 021130R INT 016020RG L = 000024 LE1\$39 021306R LGT\$3 003000R LPS\$25 015552R LSH = 177314	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R	
GT1339 IFIX INF\$15 INR\$39 ISR\$9 LE1\$15 LGT\$10 LOG\$3	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R		HLT\$43 022040R IGN\$43 022032R INF\$39 021130R INT 016020RG L = 000024 LE1\$39 021306R LGT\$3 003000R LPS\$25 015502R LSH = 177314 LUP\$41 021712R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 ISR\$9 LE1515 LGT\$10 LOG\$3 LP\$47	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R 022204R		HLT\$43 022040R IGN\$43 022032R INF\$39 021130R INT 016020RG L = 000024 LE1\$39 021306R LGT\$3 003000R LPS\$25 015552R LSH = 177314	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 ISR\$9 LE1515 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R 022204R 022204R 013216R 021446R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015502R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 ISR\$9 LE1515 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30	021234R 015154RG 021156R 026526R 011356R 007342R 002552R 022204R 022204R 013216R 021446R 017460R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015502R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML5\$8 004742R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 ISR\$9 LE1515 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30 ML8\$9	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R 022204R 022204R 013216R 021446R 017460R 017460R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015502R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MOV\$25       015656R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH# 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML5\$8 004742R MQ = 177304	
GT1339 IFIX INF\$15 INF\$15 INF\$15 ISF\$9 LE1515 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30 ML8\$9 MT0\$28	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R 022204R 013216R 013216R 021446R 017460R 017460R 016736R		HLT\$43022040RIGN\$43022032RINF\$39021130RINT016020RGL#000024LE1\$39021306RLGT\$3003000RLPS\$25015502RLSH#177314LUP\$41021712RMDV\$8004036RML1\$9005204RMOV\$25015656RMT0\$30017514R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 006506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML5\$8 004742R MQ = 177304	
GT1339 IFIX INF\$15 INF\$15 INF\$15 IST\$19 LGT\$19 LGT\$19 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30 ML8\$9 MT0\$28 MT0\$28 MT1\$30	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R 022204R 013216R 021446R 017460R 016736R 017464R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INT       016020RG         L       # 000024         LE1\$39       021306R         LGT\$3       003000R         LP\$\$25       015502R         LSH       # 177314         LUP\$\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MT0\$30       017514R         MT2\$28       016732R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 006506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML5\$8 004742R MQ = 177304 MT1\$28 016644R MUL\$29 017052R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 IST\$19 LE1515 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30 ML839 MT0\$28 MT0\$28 MT1\$30 MUL\$8	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 002552R 022204R 013216R 013216R 021446R 017460R 016736R 016736R 017464R 017464R		HLTS43       022040R         IGNS43       022032R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015562R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 006506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML5\$8 004742R MG = 177304 MT1\$28 016644R MUL\$29 017052R M16\$19 014362R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 LGT\$10 LGT\$10 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30 ML8\$9 MT0\$28 MT0\$28 MT0\$28 MT1\$30 MUL\$8 M45\$9	921234R 915154RG 911976R 921156R 926526R 911356R 907342R 902552R 922294R 913216R 921446R 913216R 915736R 916736R 916736R 916736R 916736R 916736R 916736R		HLTS43       022040R         IGNS43       022032R         INF\$39       021130R         INF\$39       02130R         INT       016020RG         L       00024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015562R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         MJ\$38       004762R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 006506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML5\$8 004742R MG = 177304 MT1\$28 016644R MUL\$29 017052R M16\$19 014362R M51\$9 006156R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 IST\$19 LE1515 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30 ML839 MT0\$28 MT0\$28 MT1\$30 MUL\$8	921234R 915154RG 911976R 921156R 926526R 911356R 907342R 902552R 922294R 913216R 921446R 917469R 916736R 916736R 917464R 916736R 917464R 916736R 917464R 916736R 917464R 916736R		HLTS43       022040R         IGNS43       022032R         INF\$39       021130R         INF\$39       02130R         INT       016020RG         L       00024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015562R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         M54\$8       004674R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH=       000042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         L15\$15       011642R         ML5\$8       004742R         MQ       177304         MT1\$28       016644R         MUL\$29       017052R         M16\$19       014362R         M51\$9       026156R	
GT1339 IFIX INF\$15 INF\$15 INF\$15 LGT\$10 LGT\$10 LGT\$10 LOG\$3 LP\$47 LUP\$17 L15\$39 MLT\$30 ML8\$9 MT0\$28 MT0\$28 MT0\$28 MT1\$30 MUL\$8 M45\$9	921234R 915154RG 911976R 921156R 926526R 911356R 907342R 902552R 922294R 913216R 921446R 913216R 915736R 916736R 916736R 916736R 916736R 916736R 916736R		HLTS43       022040R         IGNS43       022032R         INF\$39       021130R         INF\$39       02130R         INT       016020RG         L       00024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015562R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         MJ\$38       004762R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 006506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML5\$8 004742R MG = 177304 MT1\$28 016644R MUL\$29 017052R M16\$19 014362R M51\$9 006156R	
GT1339 IFIX INF\$15 INR\$39 ISR\$9 LE1515 LGT\$10 LOG\$37 LUP\$17 L15\$30 ML15\$30 ML0\$28 MT0\$28 MT1530 MUL\$8 M45\$9 M55\$8	021234R 015154RG 011076R 021156R 02526R 011356R 007342R 0025204R 022204R 013246R 021446R 017460R 017464R 016736R 017464R 016736R 017464R 017464R 017464R 017464R 017464R 017464R 017464R 017464R		HLT\$43022040RIGN\$43022032RINF\$39021130RINT016020RGL00024LE1\$39021306RLGT\$3003000RLP\$\$25015502RLSH177314LUP\$41021712RMD\$8004036RML1\$9005204RMO\$25015656RMT0\$30017514RMT2\$28016732RMUL\$9005240RM54\$8004674RM54\$8004674R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH=       000042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         L15\$15       011642R         ML5\$8       004742R         MQ       177304         MT1\$28       016644R         MUL\$29       017052R         M16\$19       014362R         M51\$9       026156R	
GT1339 IFIX INF\$15 INR\$39 ISR\$9 LE1\$15 LGT\$10 LCP\$47 LUP\$17 LUP\$17 ML839 MT0\$28 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT1\$300 MT2\$300 MT1\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$300 MT2\$30	021234R 015154RG 011076R 021156R 006526R 011356R 007342R 0025204R 022204R 013446R 017460R 017464R 016736R 017464R 016736R 017464R 017464R 017464R 017464R 017464R 006136R 006172R 006174R		HLT\$43022040RIGN\$43022032RINF\$39021130RINT016020RGL00024LE1\$39021306RLGT\$3003000RLGT\$3003000RLSH177314LUP\$41021712RMDV\$8004036RML1\$9005204RMOV\$25015656RMT0\$30017514RMT2\$28016732RMUL\$9005240RM54\$8004674RM54\$8004674RM54\$2016356R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH=       000042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         LI5\$15       011642R         ML5\$8       004742R         MQ       177304         MT1\$28       016644R         MUL\$29       017052R         M16\$19       014362R         M51\$9       026156R         M54\$9       000050R         N       000014         NCK\$8       003614R	
GT1339 IFIX INF\$15 INR\$39 ISR\$9 LE1\$15 LGT\$10 LP\$47 L15\$30 ML0\$30 ML0\$30 MT1\$30 MT1\$30 MT1\$30 MT1\$30 M52\$9 M55\$8 N55\$8 NAD\$42 NCP\$2	021234R 015154RG 011076R 021156R 021156R 0113526R 0113342R 0025204R 022204R 0123446R 012446R 0174646R 0174646R 0174646R 0174646R 0174646R 0174646R 0174648 005136R 005136R 006172R 006172R 006172R 0062254R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LGT\$3       003000R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         M54\$8       004702R         M54\$8       004674R         M54\$8       006254R         NCK\$24       015366R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH=       000042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         L15\$15       011642R         ML5\$8       004742R         MQ       177304         MT1\$28       016644R         MUL\$29       017052R         M16\$19       014362R         M51\$9       026156R         N       000014         NCK\$8       003614R         NE\$55       003326R	
GT1339 IFIX INF\$15 INR\$39 ISR\$9 ISR\$9 ISR\$9 ISF30 LGT\$10 LP\$47 LUP\$530 ML0\$30 ML0\$30 MT1530 MT1530 MT1530 M52\$9 M52\$9 M52\$9 NAD\$2 S55\$8 NAD\$2 NEG\$6	921234R 915154RG 911976R 921156R 921526R 9113526R 91133526R 9023204 9022204 9132428 9214468 917469 8017468 917468 917468 917468 917468 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 9051368 905135 9051368 9051368 9051368 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 905135 90515 905135 905135 905135 905135 905135 905135 90735 905135 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90735 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 90757 907577 90757 90757 90757 90757 90757 90		HLT\$43022040RIGN\$43022032RINF\$39021130RINF\$3902130RINT016020RGL00024LE1\$39021306RLGT\$3003000RLGT\$3003000RLSH177314LUP\$41021712RMDV\$8004036RML1\$9005204RMDV\$25015656RMT0\$30017514RMT2\$28016732RMUL\$9005240RM54\$8004674RM54\$8004674RM81\$9006254RNCK\$24015306RNEG\$42000252RNFL\$1001502R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH=       00042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         L15\$15       011642R         ML5\$8       004742R         MQ       177304         M15\$29       017052R         M16\$19       014362R         M51\$9       026156R         M54\$9       006050R         N       003614R         NCK\$8       003614R         NEG\$5       003326R         NGM\$24       015476R	
GT1339 IFIX INF\$15 INR\$39 ISR\$9 ISR\$9 ISE1510 LGT\$10 LP\$530 ML05530 ML05530 MT1530 MT1530 MT1530 MT1539 M5239 M5239 M5239 M5239 NAD\$29 NAD\$29 NAD\$29 NGM329	921234R 915154RG 911976R 921156R 921526R 9113526R 907342R 9022294R 9022294R 91214469R 91214469R 91214468R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 915466R 9051348R 905134R 905134R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 905254R 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527 90527		HLT\$43022040RIGN\$43022032RINF\$39021130RINF\$3902130RINT016020RGL000024LE1\$39021306RLGT\$3003000RLGT\$3003000RLSH177314LUP\$41021712RMDV\$8004036RML1\$9005204RMDV\$25015656RMT0\$30017514RMT2\$28016732RMUL\$9005240RM54\$8004674RM54\$8004674RM81\$9006254RNCK\$24015306RNEG\$42000252RNFL\$1001502RNGM\$35017750R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH=       00042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         L15\$15       011642R         ML5\$8       004742R         MQ       177304         MUL\$29       017052R         M16\$19       014362R         M51\$9       026156R         M54\$9       000050R         N       000014         NCK\$8       003614R         NEG\$5       003326R         NGM\$24       015476R         NGD\$16       013104R	
GT1339 IFIX INF\$15 INR\$39 ISR\$39 ISR\$39 ISC315 IGT\$10 LCP\$530 ML05530 ML05530 ML05530 MT15530 MT15530 M52\$9 M52\$9 N555\$8 NAD\$2 S55\$8 NAD\$2 S55\$8 NAD\$29 NGQ\$18	921234R 915154RG 911076R 921556R 921556R 911356R 907342R 9022204R 9022204R 9124460R 9124460R 9124460R 91574646R 9157466R 9157466R 9157466R 9065132R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LGT\$3       003000R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MU\$25       015656R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         M54\$8       004674R         M54\$8       004674R         M54\$8       004252R         NCK\$24       015306R         NEG\$42       000252R         NFL\$1       001502R         NGM\$35       017750R         NML\$19       014302R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML588 004742R MQ = 177304 MT1\$28 016644R MUL\$29 017052R M16\$19 014362R M51\$9 006050R N = 000014 N54\$9 006050R N = 000014 NCK\$8 003614R NEG\$5 003326R NGM\$24 015476R NGD\$16 013104R NNZ\$28 016272R	
GT1339 IFIX INF\$15 INR\$39 ISR\$39 ISR\$39 ISC315 IGT\$10 LCP\$530 ML05530 ML05530 ML05530 MT15530 MT15530 M525588 N525588 N525588 N525588 N525588 N525588 N525588 N525588 N525588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52588 N52	921234R 915154RG 911076R 921556R 921556R 911356R 907342R 9022204R 9022204R 9022204R 9022204R 9022204R 9124460R 9124460R 915460R 915460R 915460R 915460R 915460R 915460R 915460R 9065132R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 906514R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 9		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LF\$30       021706R         LGT\$3       003000R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MC\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         M54\$8       004762R         M54\$8       004674R         M54\$8       004674R         M54\$8       004252R         NCK\$24       015306R         NEG\$42       000252R         NFL\$1       001502R         NGM\$35       017750R         NML\$19       014302R         NN2\$9       005176R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML588 004742R MQ = 177304 MT1\$28 016644R MUL\$29 017052R M16\$19 014362R M51\$9 006050R N = 000014 NC\$\$8 003614R NEG\$5 003326R NGM\$24 015476R NG0\$16 013104R NN2\$28 016272R N0D\$1 001476R	
GT1339 IFIX INF\$15 INR\$39 ISR\$39 ISR\$39 ISC315 IGT\$10 LCP\$530 ML05530 ML05530 ML05530 MT15530 MT15530 M52\$9 M52\$9 N555\$8 NAD\$2 S55\$8 NAD\$2 S55\$8 NAD\$29 NGQ\$18	921234R 915154RG 911076R 921556R 921556R 911356R 907342R 9022204R 9022204R 9124460R 9124460R 9124460R 91574646R 9157466R 9157466R 9157466R 9065132R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 9065134R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 906524R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624R 90624		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LGT\$3       003000R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MU\$25       015656R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         M54\$8       004674R         M54\$8       004674R         M54\$8       004252R         NCK\$24       015306R         NEG\$42       000252R         NFL\$1       001502R         NGM\$35       017750R         NML\$19       014302R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML588 004742R MQ = 177304 MT1\$28 016644R MUL\$29 017052R M16\$19 014362R M51\$9 006050R N = 000014 NCK\$8 003614R NEG\$5 003326R NGM\$24 015476R NGD\$16 013104R NNZ\$28 016272R NOD\$1 001476R ND\$9 005402R	
GT1339 IFIX INF\$15 INF\$159 ISE1510 LGT\$37 LGT\$37 LUP\$\$39 ML05\$39 ML15\$9 ML15\$9 ML15\$9 ML15\$9 ML15\$9 ML15\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9	921234R 915154RG 911976R 921156R 926526R 911356R 902526R 9025204R 9022206R 922206R 922206R 9132146R 92146R 913246R 915526R 913246R 915526R 913256R 913246R 915526R 913246R 915526R 913246R 915526R 913246R 915526R 913246R 915526R 915526R 913246R 915526R 905134R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       003000R         LF\$30       021706R         LGT\$3       003000R         LSH       177314         LUP\$41       021712R         MDV\$8       004036R         ML1\$9       005204R         MC\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         M54\$8       004762R         M54\$8       004674R         M54\$8       004674R         M54\$8       004252R         NCK\$24       015306R         NEG\$42       000252R         NFL\$1       001502R         NGM\$35       017750R         NML\$19       014302R         NN2\$9       005176R	IDINT 016020RG INC\$20 014712R INR\$15 011134R ISN\$9 005506R LENGTH= 000042 LFT\$8 005010R LOG\$10 006656R LP\$46 022126R LUP\$14 010542R L15\$15 011642R MLT\$28 016640R ML588 004742R MQ = 177304 MT1\$28 016644R MUL\$29 017052R M16\$19 014362R M51\$9 006050R N = 000014 NC\$\$8 003614R NEG\$5 003326R NGM\$24 015476R NG0\$16 013104R NN2\$28 016272R N0D\$1 001476R	
GT1339 IFIX INF\$15 INF\$159 ISE1510 LGT\$37 LGT\$37 LUP\$\$39 ML05\$39 ML05\$39 ML15\$39 ML15\$39 ML15\$39 ML15\$39 ML15\$39 ML15\$39 ML15\$39 ML15\$39 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$	921234R 915154RG 911076R 921556R 921556R 91133526R 90252542R 90222216R 90222216R 90222216R 90222216R 90222216R 90222216R 90222216R 90222246R 90222246R 90222246R 90222246R 90222246R 9022246R 90257466R 90061724R 90061724R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 9002248R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 900224R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90024R 90		HLTS43       022040R         IGNS43       022032R         INF\$39       021130R         INF\$39       021306R         INF\$39       021306R         LE1\$39       023000R         LGT\$3       000024         LE1\$39       021306R         LGT\$3       003000R         LPS\$25       015502R         LSH       #         MDV\$8       004036R         ML1\$9       005204R         MD\$28       004702R         MUL\$9       005240R         M54\$8       004674R         M81\$9       006254R         NCK\$24       015306R         NEG\$42       000252R         NFL\$1       001502R         NML\$19       014302R         NNZ\$9       005176R         NO\$28       016450R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH#       000042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         L15\$15       011642R         L15\$15       011642R         MLT\$28       016640R         ML5\$8       004742R         MQ       177304         MT1\$28       016644R         MUL\$29       017052R         M16\$19       014362R         M51\$9       026156R         N54\$9       000014         NCK\$8       003614R         NEG\$55       003326R         NGM\$24       015476R         NQ\$16       013104R         NN2\$28       016272R         NOD\$1       001476R         NO\$1       001476R         NO\$1       001476R         NO\$29       05402R	
GT1339 IFIX INF\$15 INF\$159 ISE1510 LGT\$37 LGT\$37 LUP\$\$39 ML05\$39 ML15\$9 ML15\$9 ML15\$9 ML15\$9 ML15\$9 ML15\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9 ML25\$9	921234R 915154RG 911976R 921156R 926526R 911356R 902526R 9025204R 9022206R 922206R 922206R 9132146R 92146R 913246R 915526R 913246R 915526R 913256R 913246R 915526R 913246R 915526R 913246R 915526R 913246R 915526R 913246R 915526R 915526R 913246R 915526R 905134R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R 90524R		HLT\$43       022040R         IGN\$43       022032R         INF\$39       021130R         INT       016020RG         L       000024         LE1\$39       021306R         LGT\$3       000024         LE1\$39       021306R         LGT\$3       003000R         LP\$\$25       015502R         LSH       =         MDV\$8       004036R         ML1\$9       005204R         MOV\$25       015656R         MT0\$30       017514R         MT2\$28       016732R         MUL\$9       005240R         M54\$8       004702R         M54\$8       004252R         NCK\$24       015366R         NEG\$42       000252R         NFL\$1       001502R         NG\$35       017750R         NML\$19       014302R         NNZ\$9       005176R         N00\$27       016120R	IDINT       016020RG         INC\$20       014712R         INR\$15       011134R         ISN\$9       006506R         LENGTH#       000042         LFT\$8       005010R         LOG\$10       006656R         LP\$46       022126R         LUP\$14       010542R         L15\$15       011642R         L15\$15       011642R         MLT\$28       016640R         ML5\$8       004742R         MQ       177304         MT1\$28       016644R         MUL\$29       017052R         M16\$19       014362R         M51\$9       020156R         N51\$9       020014         NCK\$8       003614R         NEG\$55       003326R         NGM\$24       015476R         NQU\$16       013104R         NN2\$28       016272R         NOD\$1       001476R         NOD\$1       001476R         NOD\$1       001476R         NOD\$1       001476R         NOD\$30       017336R	

FPMP11 FLUATING POINT & MATH PA MACRU VR04-14 07-SEP-72 11:43 PAGE 52+ SYMBOL TABLE

NT381B       DIJSGRR       NUKEND       DUBAD       NITS24       DIJSGR         NXT88       CU3574R       NZES25       015646R       CL524       015067R         OTTS25       CU5064R       ONTS19       014340R       ONTS20       015120       01522R         OUTS25       V15552R       OUTS29       017148       OUTS20       001314       01522R         OUTS25       V1744R       OUTS28       V1556       02276R       OUTS29       017144       00152R         OUTS25       V1744R       OUTS28       V1556       02276R       OUTS14       01564R         OUTS26       V1744R       OUTS28       01564R       OVR12       007652R       OVR12       007652R         OVR16       014340R       OVR12       004670R       OVR32       004354R         OVR31       017454R       OVR32       004354R       OVR32       004354R         OVR34       024426R       PL520       014742R       PL526       044792R       PL536       033408R         OVR34       024426R       PL520       014742R       PL536       03402R       PL526       0440472R       PL536       044678       011348R       01568R       011568R       015628R								
NYTES         D01574R         N2E325         015646R         OCL324         0115064R         ONE319         0144090R         ONE320         015776R           DUT314         016526R         UT319         014346R         OUT320         017246R         OUT320         017246R           DUT314         016526R         UT329         017126R         OUT320         017246R         OUT320         0174246R           OUT320         017424R         UUT350         02436R         OUT320         017424R         01552R           OUT320         017424R         UUT320         017424R         01552R         017424R           OUT320         01744R         UUT320         017424R         01556R         01576R           OUT320         013456R         04419         044670R         048328         01756R           OVR320         017430R         047430R         047450R         04752R         P         000420         PC         \$20020R         04336R           OVR320         017428R         PL5320         014728R         PL5320         04336R         PL5320         043406R           OVR320         01226R         PL530         004202R         PL5313         03300R         PL5330         021550R </td <td>NT0518</td> <td>013540R</td> <td></td> <td>NUMENDE</td> <td>000000</td> <td></td> <td>NXTS24</td> <td>015346R</td>	NT0518	013540R		NUMENDE	000000		NXTS24	015346R
OCT 255         DIG 00 AR         ORE 519         DI 4000R         DNE 520         DNE 528           DNE 538         020556R         UT2 542         000170R         UUTS1         00151         001762         022426R           DUTS14         01552R         UUTS29         017126R         UUTS20         02426R           DUTS28         01552R         UUTS29         017126R         UUTS20         02426R           DUTS14         01552R         UUTS38         026570R         UVTS1         021541         0217404R           DUTS1         01350R         UVTS2         00240R         UVTS1         00156R         UVTS1         00156R           OVRS10         01430AR         UVTS1         00456R         UVTS2         024236R           UVRS20         017430R         UVRS31         017554R         UVTS2         024357R           UVRS20         017432R         P         =         000020         UVTS2         024356R           UVRS20         017432R         UVRS31         01752R         DUTS28         016604R           VVRS20         017432R         PLS20         014732R         PLS26         003406R           VIS30         017430R         PLS20         014732R				-				
DNESSB         0208564         07232         00170R         OUTSI         00152         001520           DUTS14         0165264         OUTS19         0143468         OUTS2         017264         OUTS2         017444           OUTS3         017744R         OUTS2         017126R         OUTS3         017444         OUTS3         017444           OUTS3         00322R         OUTS6         003406R         OVFS1         01564R           OVRS1         001566R         OVFS1         01564R         OVFS1         012754           OVRS2         013404R         OVFS1         012754R         OVFS2         012754R           OVRS2         01402R         OVFS1         014056R         OVFS2         012754R           OVRS2         017430R         OVFS2         01472R         PLMS3         017754R           OVRS3         017432R         P         000420         PC         *X004087           PLUS1         01202R         PLS2         01472R         PLMS42         003354R           PLS42         00022R         PLS5         03356R         PD154         010302R           PLS42         00022R         PLS43         003226R         PLS42         000226R								
OUTS14         OUTS29         OUTS19         OUTS29         OUTS26         OUTS29         OUTS26         OUTS29         OUTS26         OUTS29         OUTS26         OUTS29         OUTS26         OUTS29         OUTS26         OUTS26         OUTS29         OUTS26         OUTS26         OUTS29         OUTS26         OUTS29         OUTS29 <thouts29< th=""> <thouts29< th=""> <thouts29< td="" th<=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></thouts29<></thouts29<></thouts29<>								
OUTS28         0UTS28         0UTS28         0UTS28         0UTS28         0UTS38         0UTS38         0UTS38         0UTS41         02T36           OUTS3         0UTS38         0UTS38         0UTS41         02T36         0UTS41         02T36           OUTS3         0UTS41         0UTS41         02T56R         0UTS41         02T56R           OVRS16         01432A         0VRS12         007632R         0VRS2         002336R           OVRS16         01432A         0VRS2         01526R         0VRS2         002336R           OVRS20         014742R         0VRS2         01528         01757AR           OVRS30         017432R         0VRS3         01752AR         0VRS3         01752AR           OVRS30         017432R         P         000020         PC <ts3< td="">         007656R           OVIS16         01762R         PLS2         01472R         PLS2         004336R           PLS32         01472R         PLS2         000220         PC<ts3< td="">         00336R           PLS32         01022R         PLS2         000220         PC<ts3< td="">         00326R           PLS32         00020R         PLS32         00326R         PLS33         00326R</ts3<></ts3<></ts3<>								
OUTS35         01734R         OUTS36         020570R         OUTS41         021756R           OUTS35         01734R         UUTS6         003400R         OVFS1         001564R           OVRS16         014506R         OVFS16         014564R         OVFS16         0127564R           OVRS16         014304R         OVFS19         014056R         OVFS29         017150R           OVRS16         017430R         OVRS11         017534R         OVRS29         017150R           OVRS16         017430R         OVRS11         013452R         OVRS29         017150R           OVRS16         01702R         OVRS11         013412R         PUS28         016604R           OVRS16         01702R         PUS28         014742R         PUS28         016040R           OVIS16         01202R         PLS20         014742R         PLS38         004020           OVIS16         01202R         PLS20         014742R         PLS38         004002           PLUS15         01202R         PLS10         007270R         PL233         004020           PUS15         014002         POS18         014374R         PL539         014374R           005227         01567R         P1137 </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
Burss         Bussler         Burss         <		·						
OVRS1         ØURS12         ØURS12 <thøur32< th="">         ØURS12         <thøur32< th="">         ØURS12</thøur32<></thøur32<>								
ÖVRS1B         ÖLKS1G         ÖLKS1G         ÖLKS1G         ÖLKS1G         ÖLKS1G         ÖLKS2G         ÖLKS2G <thölks2g< th=""> <thøls2g< th="">         ÖLKS2G<td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></thøls2g<></thölks2g<>								
UVRS20         015100R         UVRS26         016506R         UVRS35         017554R           UVRS30         017430R         UVRS31         017554R         UVRS35         017554R           UVRS30         017430R         UVRS31         017554R         UVRS35         017554R           UVRS30         017450R         UVRS31         013452R         UVRS35         017554R           UVRS30         017420R         UVRS31         013452R         UVRS35         016664R           UVRS30         017420R         PLS30         014432R         PLMS42         003354R           PLUS15         01216R         PLMS42         00340R         PLS30         003224R         PLS30         003224R           PLIS42         00022R         PLS31         010322R         PLS33         00326R         POINT =         00002         POS22         01507R         PVS13         010322R           PUS15         012042R         PY1837         020347R         PLS32         014574R         POS22         01572R           PVS131         010322R         PY133         010322R         PY133         010322R         PY133         010322R           PY1331         010327R         PY1339         021572R								
UVR\$30       017430R       UVR\$31       017554R       UVR\$35       01754R         UVR\$36       02020R       UVR\$6       04677R       UV1312       007656R         UV1\$10       01754R       P       00020       PC       \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$	OVRS18							
UVR306         U20020H         UVR38         UA677R         UVI312         UA7656R           UVI316         U12702R         U1318         U13452R         DVI328         U16604R           UVI316         U17432R         P         U00020         PC         %2004027           PCX38         U40202R         PL550         U14732R         PL7542         U00326R           PLS51         U1216R         PLU339         U21216R         PL753         U00202R           PL1342         U00372R         PL250         U0777R         PL253         U00402R           PUIS12         U00372R         PL251         U00372R         PU2520         U14574R           PUS25         U15012R         PU537         U16076R         PR4513         U10328R           PV1315         U12042R         PY137         U2032R         PY2537         U20342R           PY2310         U10312R         PY2517         U13174R         P1529         U17016R           PY3517         U13230R         P1517         U13144R         P1529         U20342R           PY2339         U21556R         P1517         U13144R         P1529         U201703R           P3536         U2072R         P2537	OVR\$20	015100R		OVR\$28				20 EX
Ovisic         012702R         0visis         01318         013452R         0visis         01528         01604R           Ovisis         014338         01432R         P         # 000020         PC         #X000027           PCXs8         004052R         PLs20         014332R         PLm542         000334R           PLS55         013330R         PL%52         004320R         PLs20         014732R         PL%53         004020R           PLysis         011216R         PLu339         021216R         PL%53         004022F           PLysis         01202R         PL3210         007272R         PL233         004022F           PLysis         012042R         PL52210         007272R         PL233         004022F           POINTE         000002         PD5519         013732R         PO5200         014574R           POS225         015012R         PS527         016076R         Pv1513         010324R           PY2513         012042R         P1537         020346R         Pv1539         021572R           PV2513         012042R         P1537         020346R         P1533         020712R           Q2517         013144R         P1529         017030R         P3517	OVR\$30	017430R		OVR 331				
UNISS         UT         UT <thu< td=""><td>OVRS36</td><td>020020R</td><td></td><td>OVRSB</td><td>004670R</td><td></td><td></td><td></td></thu<>	OVRS36	020020R		OVRSB	004670R			
Qijija       0/1432R       P       # 000020       PC       #%00007         PCK86       0/4052R       PLES20       0/14732R       PLM542       0/0354R         PLS342       0/0026R       PLS55       0/03330R       PLS56       0/03020R         PLS42       0/0027R       PLS50       0/0330R       PLS56       0/0300R         PL1542       0/002R       PLS51       0/07270R       PL253       0/0326R         PUINTL       0/0002       PDS17       0/04002       PDS17       0/04002         POINTL       0/04002       PDS17       0/04076R       PY1813       0/0422R         PY1815       0/2042R       PY1813       0/14574R       P1839       0/21572R         PY2813       0/0312R       PY2815       0/2042R       PY2837       0/2337       0/2042R         PY2839       0/2156R       P1517       0/13144R       P15838       0/20712R       0/20458R <t< td=""><td>OV1\$16</td><td>012762R</td><td></td><td>OV1\$18</td><td>013452R</td><td></td><td></td><td>Ø16604R</td></t<>	OV1\$16	012762R		OV1\$18	013452R			Ø16604R
PCK\$8 $0 = 0 \le 0 \ge 0$ $147 \le 2R$ PL \$20 $0147 \le 2R$ PL \$42 $00 \le 3542$ PL \$35 $00 \le 350R$ PL \$36 $00 \le 350R$ PL \$56 $00 \le 350R$ PL \$56 $00 \le 350R$ PL \$15 $0112 \le 2R$ PL \$13 $0212 \le 2R$ PL \$13 $01 \le 300R$ PL \$15 $0112 \le 30R$ PL \$15 $012 \ge 2R$ PL \$10 $00727R$ PL \$23 $00 \le 2R$ $00 \le 2R$ $00 \le 2R$ PL \$15 $012 \ge 2R$ PL \$10 $00727R$ PL \$23 $00 \le 2R$ $00 \le 2R$ PL \$15 $012 \ge 2R$ PL \$10 $00727R$ PL \$23 $012 \le 2R$ PL \$15 $012 \le 2R$ PL \$17 $0137 \le 2R$ PC \$23 $014574R$ PD \$25 \$15 \le 12RPD \$25 \$17 $012 \le 2R$ PT \$13 $01032R$ PT \$25 \$15 \$12 \$12 \$12 \$10 $012 \le 2R$ PT \$13 $01032R$ PT \$13PT \$23 \$10 $012 \le 2R$ PT \$137 $020346R$ PT \$139 $021572R$ PT \$23 \$10 $012 \ge 2R$ PT \$137 $02034R$ PT \$253 $017016R$ PT \$23 \$10 $012 \le 2R$ PT \$137 $02034R$ PT \$253 $017016R$ PT \$23 \$10 $012 \ge 2R$ PT \$137 $02034R$ PT \$253 $02712R$ PT \$23 \$10 $012 \ge R$ PT \$253 $02712R$ $02342R$ PT \$253PT \$23 \$10 $012 \ge R$ PT \$253 $02072R$ PT \$253PT \$23 \$10 $012 \ge R$ PT \$253 $02072R$ PT \$253PT \$23 \$10 $012 \ge R$ PT \$253 $02072R$ PT \$253PT \$13 $012 \ge R$ PT				P s	000020		PC #	8000007
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$				PLES20	014732R		PLM542	000354R
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$							PLS\$6	003406R
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$								
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$								
PHODE         =         040000         PNT\$8         003766R         POINT =         000002           PIINTL#         000002         POS\$19         013732R         POS\$20         014574R           POS\$25         015612R         POS\$27         016076R         PR4513         010134R           PTF\$8         004000R         PT\$42         000076R         PY1\$13         010322R           PY1\$15         012042R         PY1\$37         020346R         PY1\$39         021572R           PY2\$13         010312R         PY2\$17         013144R         P1\$29         017016R           PY2\$39         021526R         P3538         020734R         P4538         020712R           Q         #         00014         Q5533         02034R         Q5537         020342R           Q1537         02024R         Q3553         02034R         Q5337         02046R           Q15337         02024R         Q00513         010252R         Q15313         010264R           Q15337         02024R         RD59         006432R         RD59         006402R           Q15337         02020R         RIT\$8         00502R         RIT\$9         006402R           Q15337         020								
POINTL# $00002$ POS\$19 $013732R$ POS\$20 $014574R$ POS\$25 $015612R$ POS\$27 $016076R$ PR4\$13 $010134R$ PTF\$8 $000076R$ PY1\$13 $010322R$ PY1\$15 $012042R$ PY1\$37 $020346R$ PY1\$13 $010322R$ PY2\$13 $010312R$ PY2\$15 $01202R$ PY1\$39 $021572R$ PY2\$13 $010312R$ PY2\$15 $01202R$ PY2\$37 $020342R$ PY2\$13 $010312R$ PY2\$17 $013144R$ P1\$29 $017016R$ P1\$538 $02072R$ P2\$538 $020734R$ P45\$38 $020712R$ Q $*000014$ QS\$13 $01020R$ QS\$13 $010224R$ QS\$13 $01026R$ QS\$37 $02034R$ QU\$133 $10224R$ QS\$13 $01027R$ QU\$33 $10224R$ QU\$133 $010224R$ QU\$13 $01027R$ QU\$32RRD\$9 $006432R$ R0F\$9 $006402R$ RES10 $00714R$ RESULT# $000036$ RET\$21 $01516R$ RES11#00010RESULT# $000036$ RET\$21 $01546R$ RN\$22 $00520R$ RT\$11 $007554R$ R0F\$4 $03204R$ RN\$22 $01516R$ RT\$11 $007542R$ RT\$1513 $010250R$ RT\$1516 $01320R$ RT\$1516<							1000	
$PO3$25$ $015612R$ $PO3$27$ $016076R$ $PR4$13$ $010134R$ $PT58$ $004060R$ $PT$42$ $00076R$ $PY1$13$ $010322R$ $PY1$15$ $012042R$ $PY1$37$ $020346R$ $PY1$13$ $010322R$ $PY2$13$ $010312R$ $PY2$15$ $012042R$ $PY1$37$ $020346R$ $PY1$37$ $PY2$39$ $021556R$ $P1517$ $013144R$ $P1529$ $017016R$ $P15$38$ $020722R$ $P2$17$ $013156R$ $P2329$ $017030R$ $P3$17$ $013236R$ $P3538$ $020734R$ $P45338$ $020712R$ $Q$ $\pi$ $000014$ $QSE513$ $010000R$ $QSE537$ $020150R$ $QST$37$ $020254R$ $QU0513$ $010246R$ $QU537$ $020246R$ $QUT$37$ $020252R$ $Q15313$ $010224R$ $QU124R$ $QT537$ $020254R$ $QU0537$ $Q06402R$ $RES13$ $01026R$ $RC510$ $007144R$ $REG53$ $002742R$ $REL$25$ $01516R$ $RES17$ $000210$ $RESUTF$ $00036$ $RET$2101516RRES17000220RRT188002742RRT1542003204RRN0$2201516RROR$11007564RROR$4003204RRTN$1001326RRT154200322RRT154200322RRTN$13010140RRTN$14010640RRTN$41021764RRT$13301020RRT1537020204RRT154200322RRT$1$								
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	······	· · · · · · · · · · · · ·						
PY1515       012042R       PY1537       020346R       PY1539       021572R         PY2513       010312R       PY2515       012032R       PY2537       020342R         PY2539       021566R       P1517       013144R       P1529       017016R         PY2539       021566R       P1517       013144R       P1529       017016R         P15368       020722R       P2517       013156R       P229       017030R         P3517       013236R       P35538       020734R       P45538       020712R         Q       # 000014       WSE513       010050R       QSE337       02024R         QST537       020254R       QUT537       020342R       QU537       02024R         QUT513       010276R       QUT537       020342R       RD59       00642R         QUT513       010276R       QUT537       020322R       R13513       010264R         Q13537       020310R       RD59       00642R       RDF59       006632R         REGS10       007144R       REGS3       002742R       R1T59       006634R         RET542       000220R       R1758       00502R       R1754       002204R         RTN513       01140R								
PY2813010312RPY2815012032RPY2837020342RPY2839021566RP1817013144RP1829017016RP1538020722RP2817013156RP2829017030RP3817013236RP353802074RP4538020712RQ $=$ 000014QSE813010000RQSE837020150RQSR513010260RQSR537020344RQSE537020246RQUTS13010276RQUTS37020224RQUD813010216RQUTS13010276RQUTS37020246RRD59006422RREG510007144RREG53002742RREL52501566RRESLT000020RIT58005022RRIT5900634RRND520156RROS11007542RRIT542000320RRTN51301440RRTN514010640RRTN514010250RRTN516013620RRTN51702024RRTN516012750RRTN516013620RRT518013766RRT1513010150RRTN516013620RRT518013766RRT1513010150RRT1537020220RRT2519014544RRT2542000322RRUD59006340RRU159006476RRU2590064644RU359006500RR0 $= $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$								
PY2539 $021566R$ P1\$17 $013144R$ P1\$29 $017016R$ P15\$38 $020722R$ $P2$17$ $013156R$ $P2$29$ $017030R$ P3517 $013236R$ $P35338$ $020734R$ $P45338$ $020712R$ Q* $000014$ QSE\$13 $01000R$ QSE\$37 $020172R$ QSR\$13 $010250R$ QSR\$37 $020304R$ QST\$13 $010224R$ QST\$37 $020224R$ QUD\$13 $010216R$ QUD\$37 $020246R$ QUT\$13 $010276R$ QUT\$37 $020322R$ Q13313 $01026R$ Q13537 $020310R$ RDD\$9 $006432R$ RDF\$9 $006402R$ REG\$10 $007144R$ REG\$3 $002742R$ REL\$25 $01516R$ RESLT* $000010$ RESULT= $000036$ RET\$21 $015146R$ RESLT* $000022R$ R158 $000632R$ RD\$9 $006402R$ RN\$2 $015166R$ RO\$11 $007542R$ RT\$9 $006632R$ RN\$2 $015166R$ RO\$11 $007542R$ RT\$1542 $00332R$ RTN\$13 $010140R$ RT\$1513 $010754R$ RT\$1542 $00332R$ RT\$1513 $010126R$ RT\$1513 $010150R$ RT\$1542 $000322R$ RT\$1514 $010540R$ RT\$1513 $010150R$ RT\$1542 $000322R$ RU\$259 $006340R$ RU\$259 $006476R$ RU\$259 $006464R$ RU\$259 $006500R$ R3 $=x000000$ R4 $=x000004$ R5 $=x00002$ R3 $=x000026$ SCK\$19 $014$								
P15\$36020722RP2\$17013156RP2\$29017030RP3\$17013236RP35338020734RP45\$38020712R0 $=$ 000014055338020734RP45\$38020712R0 $=$ 00001405513010060R05537020150R05813010260R05837020304R05533010224R01533010276R001533010216R000537020246R015337020310RRD59006432RRD59006402RREG\$10007144RREG\$3002742RREL\$25015166RRESLT $=$ 000010RESULT=00036RET\$21015146RRESLT $=$ 004664RR158005022RRIT\$9006634RRNN\$2015166RR07\$11007564RR07\$4003204RRTN\$13010140RR1\$11007564RRT\$1\$12003204RRTN\$16013620RRT\$1\$13010460RRT\$1\$14010640RRT\$1\$10013620RRT\$1\$13010150RRT\$1\$13010150RRT1\$37020220RR12519014544RR1\$2\$42003228RUD\$9006540RR0 $=$ \$00000R1 $=$ \$00000R1\$37020220RR12519014544RR1\$2\$42000322RR1\$39006540RR0 $=$ \$00000R1 $=$ \$00000R2 $=$ \$000002R3 $=$ \$000006R1 $=$ \$000004R5 $=$ \$000005 $=$ \$000006R4 $=$ \$000004R5 $=$ \$0000								
P3\$17013236RP35\$38020734RP45\$38020712R0=000014GSE\$13010050RGSE\$37020150RGSR\$13010250RGSR\$37020304RGSE\$37020150RGST\$37020254RGUD\$13010216RGUD\$37020246RGUT\$13010276RGUT\$37020322RG13\$13010264RGUT\$13010276RGUT\$37020322RG13\$13010264RGUT\$14REG\$10007144RREG\$3002742RREL\$25RESLT=00010RESULT#00036RET\$21015146RRESLT=00020RRIT\$8005022RRIT\$9006634RRN\$22015166RROR\$11007564RROR\$400204RRN\$8004664RRT1\$14010640RRTN\$16012750RRTN\$18013620RRTN\$14010640RRTN\$16012750RRT1\$1301040RRT\$1510010222RRT\$6400322RRU\$9006340RRT\$1516013706RRT\$1513010150RRT1\$37020220RRT\$2519014544RRT\$2542000322RRU\$9006340RRU\$9006476RRU\$9006464KRU\$9006372RSCL\$20015046RSCL\$3007232RSCL\$20015046RSCL\$3007232RSCL\$19014250RSCL\$20015046RSCL\$3007232RSCL\$8003722RSCL\$20015046RSCL\$3002003R4\$200004 <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>								
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	P15\$38							
QSR\$13 $010260R$ QSR\$37 $020304R$ QST\$13 $010224R$ QST\$37 $020254R$ QUD\$13 $010216R$ QUD\$37 $020246R$ QUT\$13 $010276R$ QUT\$37 $020322R$ $Q13513$ $010264R$ Q13\$37 $020310R$ RDD\$9 $006432R$ RDF\$9 $006402R$ REG\$10 $007144R$ REG\$3 $002742R$ REL\$25 $015616R$ RESLT= $000036$ RET\$21 $015146R$ RES21RES1= $000220R$ RIT\$8 $005022R$ RIT\$9 $006634R$ RN\$22 $015166R$ ROR\$11 $007564R$ ROR\$4 $003204R$ RN\$6 $004664R$ RT\$11 $007542R$ RT\$42 $000332R$ RTN\$13 $01040R$ RTN\$14 $010640R$ RTN\$41 $021764R$ RTN\$16 $01326R$ RT\$516 $013726R$ RT\$133 $010150R$ RT\$137 $020220R$ RT\$518 $013726R$ RT\$133 $010150R$ RT\$137 $020220R$ RT\$2519 $014544R$ RT\$242 $000322R$ RU\$99 $006340R$ RU\$99 $006476R$ RU\$99 $006464$ RU\$99 $006500R$ R\$ $= $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$	P3\$17	Ø13236R		P35538	020734R			
QST\$37 $020254R$ $QUD$13$ $010216R$ $QUD$37$ $020246R$ QUT\$13 $010276R$ $QUT$37$ $020322R$ $Q13$13$ $010264R$ Q13\$37 $020310R$ RDD\$9 $006432R$ RDF\$9 $006402R$ REG\$10 $007144R$ REG\$3 $002742R$ REL\$25 $015616R$ RESLT = $00010$ RESULT= $00036$ RET\$21 $01146R$ RES2 015166RROR\$11 $007564R$ ROR\$4 $003204R$ RN\$2 $015166R$ ROR\$11 $007564R$ ROR\$4 $003204R$ RN\$6 $004664R$ RT1\$11 $007564R$ RT1\$42 $0003204R$ RN\$6 $004664R$ RT1\$11 $007564R$ RT1\$42 $0003204R$ RN\$6 $004664R$ RT1\$11 $007564R$ RT1\$42 $0003204R$ RTN\$13 $010140R$ RTN\$14 $010640R$ RTN\$41 $02176R$ RTN\$13 $010140R$ RTN\$14 $010640R$ RTN\$41 $01050R$ RTN\$16 $01326R$ RT\$518 $013706R$ RT1\$13 $010150R$ RT1\$37 $020220R$ RT\$2\$19 $014544R$ RT2\$42 $000322R$ RUD\$9 $006340R$ RU1\$9 $006476R$ RU2\$9 $006464R$ RU3\$9 $006500R$ R0 $=000026$ SCK\$1 $001200R$ SC \$2 $002176R$ SCL\$10 $007232R$ SCL\$19 $014250R$ SC \$2 $002176R$ SCL\$3 $003000R$ SCL\$8 $003722R$ SC \$2 $002176R$ SCL\$3 $002002$ SFL\$2 $002306R$ SF \$2 <td< td=""><td>Q . Ŧ</td><td>000014</td><td></td><td>QSES13</td><td>010060R</td><td></td><td>QSE\$37</td><td>020150R</td></td<>	Q . Ŧ	000014		QSES13	010060R		QSE\$37	020150R
QST\$37 $020254R$ $0U0$13$ $010216R$ $0U0$37$ $020246R$ QUT\$13 $010276R$ $QUT$37$ $020322R$ $Q13$13$ $010264R$ Q13\$37 $020310R$ $RDD$9$ $006432R$ $RDF$9$ $006402R$ REG\$10 $007144R$ REG\$3 $002742R$ REL\$25 $015616R$ RESLT = $00010$ RESULT= $000366$ RET\$21 $01146R$ RET\$42 $000220R$ RIT\$8 $005022R$ RIT\$9 $006634R$ RNU\$22 $015166R$ ROR\$11 $007564R$ ROR\$4 $003204R$ RNN\$6 $004664R$ RT1\$11 $007564R$ RTN\$42 $000322R$ RTN\$13 $010140R$ RTN\$14 $010640R$ RTN\$41 $021750R$ RTN\$13 $010140R$ RTN\$14 $010640R$ RTN\$41 $021750R$ RTN\$16 $01326R$ RTS\$18 $013746R$ RTN\$41 $021750R$ RTN\$16 $01326R$ RTS\$18 $013746R$ RT1\$13 $010150R$ RT1\$37 $020220R$ RT2\$19 $014544R$ RT2\$42 $000322R$ RUD\$9 $006340R$ RU1\$9 $006476R$ RU2\$9 $006464R$ RU3\$9 $006500R$ R3 $x000000$ R1 $x3000004$ R5 $x3000002$ R3 $x000000$ R1 $x3000004$ R5 $x000002$ R3 $x000000$ R4 $x3000004$ R5 $x000005$ $s 00026$ SCK\$1 $001200R$ SCK\$2 $00176R$ SCL\$3 $003000R$ SCL\$8 $003524R$ SCL\$10 $007232R$ SCL\$8 <td>QSR\$13</td> <td>010260R</td> <td></td> <td>QSR\$37</td> <td>020304R</td> <td></td> <td>QST\$13</td> <td>010224R</td>	QSR\$13	010260R		QSR\$37	020304R		QST\$13	010224R
QUT\$13       010276R       QUT\$37       020322R       Q13\$13       010264R         Q13\$37       020310R       RDD\$9       006432R       RDF\$9       006402R         REG\$10       007144R       REG\$3       002742R       REL\$25       015616R         RESLT #       000036       RET\$21       015146R       RET\$21       015146R         RESLT #       000020R       RIT\$8       005022R       RIT\$9       00634R         RND\$22       015166R       ROR\$11       007564R       ROR\$4       00230R         RN\$6       004664R       RI\$11       007542R       RT\$1\$42       000320R         RT\$13       010140R       RT\$14       010640R       RT\$1\$42       000320R         RT\$13       010140R       RT\$14       010640R       RT\$1\$42       000320R         RT\$13       010140R       RT\$14       010640R       RT\$1\$13       010150R         RT\$13       010126R       RT\$1\$10       01706R       RT\$1\$13       010150R         RT\$157       020220R       RT2\$19       014544R       RT2\$42       000322R         RUD\$9       006500R       R0       #2000000       R1       #2000004         R2       #2000002					010216R		QUD\$37	020246R
Q13\$37       020310R       RDD\$9       006432R       RDF\$9       006402R         REG\$10       007144R       REG\$3       002742R       REL\$25       015616R         RESLT #       00020R       RIT\$8       00502R       RIT\$9       006634R         RND\$22       015166R       ROR\$1       007564R       ROR\$4       00302R         RN\$6       004664R       RI\$11       007564R       ROR\$4       00303R         RN\$6       004664R       RI\$11       007564R       ROR\$4       00303R         RTN\$13       010140R       RTN\$14       010640R       RTN\$16       012750R         RTN\$13       010140R       RTN\$14       010640R       RTN\$16       012750R         RTN\$13       013126R       RT\$\$18       013706R       RT\$1\$13       010150R         RT\$1\$37       020220R       RT\$2\$19       014544R       RT\$2\$42       000322R         RUD\$9       006340R       RU1\$9       006476R       RU2\$9       006464R         RU3\$9       006500R       R0       #X000000       R1       #X000001         R2       #X000002       R3       #X000000       R1       #X000001         R2       #X0000002       R3 </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>								
REG\$10       007144R       REG\$3       002742R       REL\$25       015616R         RESLT #       000010       RESULT#       000036       RET\$21       015146R         RET\$42       000220R       RIT\$8       005022R       RIT\$9       006634R         RN\$22       015166R       ROR\$11       007564R       ROR\$4       003204R         RN\$6       004664R       RR1\$11       007564R       ROR\$4       003204R         RN\$6       004664R       RR1\$11       007564R       ROR\$4       003204R         RN\$6       004664R       RR1\$11       007542R       RT\$1\$42       00032R         RT\$13       010140R       RT\$1\$14       010640R       RT\$1\$42       00032R         RT\$14       010640R       RT\$1\$13       010150R       RT\$1\$13       010150R         RT\$1517       020220R       RT\$1\$13       010150R       RT\$1\$13       010150R         RT\$1\$37       020220R       RT2\$19       014544R       RT\$2\$42       000322R         RU\$9       006340R       RU\$1\$9       006476R       RU\$2\$9       006464R         RU\$39       006500R       R\$0       #2000002       R\$1       #2000004         R5       #2000002<								
RESLT # 000010       RESULT# 000036       RET\$21       015146R         RET\$42       000220R       RIT\$8       005022R       RIT\$9       006634R         RND\$22       015166R       ROR\$11       007564R       ROR\$4       003204R         RN\$8       004664R       RI\$11       007564R       ROR\$4       003204R         RN\$8       004664R       RI\$11       007564R       ROR\$4       003204R         RN\$13       010140R       RT\$14       010640R       RT\$1542       00032R         RT\$13       012140R       RT\$14       010640R       RT\$1542       00032R         RT\$13       012126R       RT\$37       020204R       RT\$\$41       021764R         RT\$316       013126R       RT\$518       013796R       RT\$1513       014150R         RT\$1537       020220R       RT2\$19       014544R       RT2\$42       00322R         RUD\$9       065340R       RU1\$9       066476R       RU2\$9       066464R         RU3\$9       065300R       R0       #2000003       R4       #2000004         R2       #2000002       R3       #2000003       R4       #2000004         R2       #20000005       \$       #0002322R						, ¹		
RET\$42       000220R       RIT\$8       005022R       RIT\$9       00634R         RND\$22       015166R       ROR\$11       007564R       ROR\$4       003204R         RRN\$8       004664R       RR1\$11       007542R       RT\$42       000322R         RTN\$13       010140R       RTN\$14       010640R       RTN\$16       012750R         RTN\$18       013620R       RTN\$14       010640R       RTN\$16       012750R         RTN\$18       013620R       RTN\$14       010640R       RTN\$16       012750R         RTN\$18       013620R       RT\$1513       010150R       RT\$1513       010150R         RT\$1517       020220R       RT\$2519       014544R       RT\$2542       00322R         RUD\$9       006340R       RU\$1\$9       006476R       RU\$259       006464k         RU\$39       006500R       R\$0       =%00000       R\$1       =%00000         R2       =%00002       R\$3       =%00002       R\$1       =%00000         R2       =%00002       R\$3       =%000026       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R								
RND\$22       015166R       ROR\$11       007564R       ROR\$4       003204R         RRN\$6       004664R       RR1\$11       007542R       RT1\$42       00032R         RTN\$13       010140R       RTN\$14       010640R       RTN\$16       012750R         RTN\$18       013620R       RTN\$14       010640R       RTN\$16       012750R         RTN\$16       013620R       RTN\$37       020204R       RTN\$41       021764R         RT\$150       013126R       RT\$518       013796R       RT1\$13       010150R         RT\$1537       020220R       RT2\$19       014544R       RT2\$42       000322R         RUD\$9       006340R       RU1\$9       006476R       RU2\$9       006464R         RU3\$9       006500R       R0       =%00000       R1       =%000004         R2       =%000002       R3       =%000003       R4       =%000004         R5       =%000005       S       =000266       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R       SCL\$3       003000R       SCL\$8       003742R         SFD\$1       001414R       SFD								
RRN\$8       004664R       RR1\$11       007542R       RTI\$42       000332R         RTN\$13       010140R       RTN\$14       010640R       RTN\$16       012750R         RTN\$18       013620R       RTN\$37       020204R       RTN\$16       012750R         RTN\$18       013620R       RTN\$37       020204R       RTN\$41       021764R         RTS\$16       013126R       RTS\$18       013706R       RT1\$13       010150R         RT1\$37       020220R       RT2\$19       014544R       RT2\$42       000322R         RUD\$9       005340R       RU1\$9       006476R       RU2\$9       006464R         RU3\$9       005500R       R0       =%000000       R1       =%000001         R2       =%000002       R3       =%000003       R4       =%000004         R5       =%000005       S       =000026       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R       SCL\$3       003040R       SCL\$8       003722R         SCN\$8       003524R       SC1\$19       014312R       SC1\$8       003742R         SFD\$1       001266R       SFR								
RTN\$13       010140R       RTN\$14       010640R       RTN\$16       012750R         RTN\$16       013620R       RTN\$37       020204R       RTN\$41       021764R         RTS\$16       013126R       RTS\$18       013706R       RT1\$13       010150R         RT1\$37       020220R       RT2\$19       014544R       RT2\$42       00322R         RUD\$9       006340R       RU1\$9       006476R       RU2\$9       006464R         RU3\$9       006500R       R0       =%00000       R1       =%00001         R2       =%000005       S       =000026       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R       SCL\$10       007232R       SCL\$19       014250R         SCN\$8       003524R       SC1\$19       014312R       SC1\$8       003742R         SFD\$1       001414R       SFD\$2       00236R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002306R         SFD\$1       001414R       SFD\$2       00236R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2		01010UN						
RTN\$18013620RRTN\$37020204RRTN\$41021764RRTS\$16013126RRTS\$18013706RRT1\$13010150RRT1\$37020220RRT2\$19014544RRT2\$42000322RRUD\$9006340RRU1\$9006476RRU2\$9066464kRU3\$9006500RR0=%000000R1=%00001R2=%000002R3=%000003R4=%00004R5=%000005S=000026SCK\$1001200RSCK\$2002176RSCL\$10007232RSCL\$19014250RSCL\$20015046RSCL\$3003000RSCL\$8003722RSCN\$8003524RSC1\$19014312RSC1\$8003742RSFD\$1001266RSFR\$2002336RSFL\$2002306RSFR\$1001266RSFR\$2002236RSFL\$2002306RSFT\$2002216RSFT\$35017720RSF0\$2002326RSF1\$1001304RSGN\$15012064RSGN\$16012702RSGN\$1801304RSGN\$24015444RSGN\$35017736RSGN\$18013572RSGN\$24015314RSG\$58003552R								
RTS\$16013126RRTS\$18013706RRT1\$13010150RRT1\$37020220RRT2\$19014544RRT2\$42000322RRUD\$9006340RRU1\$9006476RRU2\$9006464RRU3\$9006500RR0=%000000R1=%00001R2=%000002R3=%000003R4=%00004R5=%00005S=000026SCK\$1001200RSCK\$2002176RSCL\$10007232RSCL\$19014250RSCL\$20015046RSCL\$3003000RSCL\$8003722RSCN\$8003524RSC1\$19014312RSC1\$6003742RSFD\$1001414RSFD\$2002336RSFL\$2002306RSFR\$1001266RSFR\$2002236RSFL\$2002326RSFT\$2002216RSFT\$35017720RSF0\$2002326RSF1\$1001304RSGN\$15012064RSGN\$16012702RSGN\$18013572RSGN\$24015314RSGS\$8003552RSGN\$18013572RSGN\$24015314RSGS\$8003552R								
RT1\$37       020220R       RT2\$19       014544R       RT2\$42       000322R         RUD\$9       006340R       RU1\$9       006476R       RU2\$9       006464R         RU3\$9       006500R       R0       =%00000       R1       =%00001         R2       =%000002       R3       =%000003       R4       =%00004         R5       =%00005       S       =000026       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R       SCL\$3       003000R       SCL\$8       003722R         SCN\$8       003524R       SCL\$19       014312R       SC1\$8       003742R         SFD\$1       001414R       SFD\$2       002336R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002326R         SFT\$2       002216R       SFT\$35       017720R       SF0\$2       002326R         SF1\$1       001304R       SGN\$15 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>								
RUD\$9       ØØ634ØR       RU1\$9       ØØ6476R       RU2\$9       ØØ6464R         RU3\$9       ØØ6500R       RØ       #X00000       R1       #X00001         R2       #X00002       R3       #X000003       R4       #X000004         R5       #X000005       S       #000026       SCK\$1       Ø01200R         SCK\$2       Ø02176R       SCL\$10       ØØ7232R       SCL\$19       Ø14250R         SCK\$2       Ø02524R       SCL\$3       ØØ3000R       SCL\$8       ØØ3742R         SFD\$1       Ø01414R       SFD\$2       ØØ2336R       SFL\$2       ØØ2306R         SFR\$1       Ø01266R       SFR\$2       ØØ2236R       SFL\$2       ØØ2326R         SFT\$2       ØØ2216R       SFT\$35       Ø17720R       SFØ\$2       ØØ2326R         SF1\$1       ØØ1304R       SGN\$15								
RU359       006500R       R0       #X00000       R1       #X00001         R2       #X000005       R3       #X000003       R4       #X000004         R5       #X000005       S       #000026       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCK\$2       002176R       SCL\$20       015046R       SCL\$20       003722R         SCN\$8       003524R       SCL\$19       014312R       SC1\$80       003742R         SFD\$1       001414R       SFD\$2       002336R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002326R         SFT\$2       002216R       SFT\$35       017720R       SF0\$2       002326R         SF1\$1       001304R       SGN\$15								
R2       #X000002       R3       #X000003       R4       #X000004         R5       #X000005       S       #000026       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R       SCL\$3       003000R       SCL\$8       003722R         SCN\$8       003524R       SC1\$19       014312R       SC1\$8       003742R         SFD\$1       001414R       SFD\$2       002336R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002326R         SFT\$2       002216R       SFT\$35       017720R       SF0\$2       002326R         SF1\$1       001304R       SGN\$15       012064R       SGN\$16       012702R         SGN\$18       013572R       SGN\$24       015444R       SGN\$35       017736R         SGN\$39       021604R       SGS\$24       015314R       SG\$58       003552R								
R5       #X000005       S       # 000026       SCK\$1       001200R         SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R       SCL\$3       003000R       SCL\$8       003722R         SCN\$8       003524R       SCL\$19       014312R       SCL\$8       003742R         SFD\$1       001414R       SFD\$2       002336R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002306R         SFT\$2       002216R       SFT\$1       001236R       SFL\$2       002326R         SFT\$2       002216R       SFT\$35       017720R       SF0\$2       002326R         SF1\$1       001304R       SGN\$15       012064R       SGN\$16       012702R         SGN\$18       013572R       SGN\$24       015444R       SGN\$35       017736R         SGN\$39       021604R       SG\$24       015314R       SG\$528       003552R								-
SCK\$2       002176R       SCL\$10       007232R       SCL\$19       014250R         SCL\$20       015046R       SCL\$3       003000R       SCL\$8       003722R         SCN\$8       003524R       SC1\$19       014312R       SCL\$8       003742R         SFD\$1       001414R       SFD\$2       002356R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002326R         SFT\$2       002216R       SFT\$35       017720R       SF0\$2       002326R         SF1\$1       001304R       SGN\$15       012054R       SGN\$16       012702R         SGN\$18       013572R       SGN\$24       015444R       SGN\$35       017736R         SGN\$39       021604R       SG\$24       015314R       SG\$528       003552R	R2 = 7	(000002						
SCL\$20       015046R       SCL\$3       003000R       SCL\$8       003722R         SCN\$8       003524R       SC1\$19       014312R       SC1\$8       003742R         SFD\$1       001414R       SFD\$2       002356R       SFL\$2       002306R         SFR\$1       001266R       SFR\$2       002236R       SFL\$2       002306R         SFT\$2       002216R       SFR\$2       002236R       SF0\$2       002326R         SFT\$2       002216R       SFT\$35       017720R       SF0\$2       002326R         SF1\$1       001304R       SGN\$15       012054R       SGN\$16       012702R         SGN\$18       013572R       SGN\$24       015444R       SGN\$35       017736R         SGN\$39       021604R       SGS\$24       015314R       SG\$528       003552R	R5 =%	000005			000026			
SCL\$20         Ø15046R         SCL\$3         Ø03000R         SCL\$8         Ø03722R           SCN\$8         Ø03524R         SC1\$19         Ø14312R         SC1\$88         Ø03742R           SFD\$1         Ø01414R         SFD\$2         Ø02336R         SFL\$2         Ø02306R           SFR\$1         Ø01266R         SFR\$2         Ø02236R         SFL\$2         Ø02306R           SFT\$2         Ø02216R         SFR\$2         Ø02236R         SF0\$2         Ø02326R           SFT\$2         Ø02216R         SFT\$35         Ø17720R         SF0\$2         Ø02326R           SF1\$1         Ø01304R         SGN\$15         Ø12054R         SGN\$16         Ø12702R           SGN\$18         Ø13572R         SGN\$24         Ø15444R         SGN\$35         Ø17736R           SGN\$39         Ø21604R         SGS\$24         Ø15314R         SG\$588         Ø03552R	SCK\$2	Ø02176R	•	SCL 510				
SCN\$8         003524R         SC1\$19         014312R         SC1\$88         003742R           SFD\$1         001414R         SFD\$2         002336R         SFL\$2         002306R           SFR\$1         001266R         SFR\$2         002236R         SFL\$2         002306R           SFR\$1         001266R         SFR\$2         002236R         SFL\$2         002306R           SFT\$2         002216R         SFR\$2         002236R         SF0\$2         002326R           SF1\$1         001304R         SGN\$15         012064R         SGN\$16         012702R           SGN\$18         013572R         SGN\$24         015444R         SGN\$35         017736R           SGN\$39         021604R         SGS\$24         015314R         SGS\$58         003552R		015046R		SCL \$3	003000R			
SFD\$1         001414R         SFD\$2         002336R         SFL\$2         002306R           SFR\$1         001266R         SFR\$2         002236R         SFT\$1         001236R           SFR\$1         00126R         SFR\$2         002236R         SFT\$1         001236R           SFT\$2         002216R         SFT\$35         017720R         SF0\$2         002326R           SF1\$1         001304R         SGN\$15         012064R         SGN\$16         012702R           SGN\$18         013572R         SGN\$24         015444R         SGN\$35         017736R           SGN\$39         021604R         SGS\$24         015314R         SGS\$52         003552R				SC1519	014312R		SC158	003742R
SFR\$1         001266R         SFR\$2         002236R         SFT\$1         001236R           SFT\$2         002216R         SFT\$35         017720R         SF0\$2         002326R           SFT\$1         001304R         SGN\$15         012064R         SGN\$16         012702R           SGN\$18         013572R         SGN\$24         015444R         SGN\$35         017736R           SGN\$39         021604R         SGS\$24         015314R         SGS\$58         003552R					002336R		SFLS2	002306R
SFT\$2         ØØ2216R         SFT\$35         Ø17720R         SFØ\$2         ØØ2326R           SF1\$1         ØØ1304R         SGN\$15         Ø12064R         SGN\$16         Ø12702R           SGN\$18         Ø13572R         SGN\$24         Ø15444R         SGN\$35         Ø17736R           SGN\$39         Ø21604R         SGS\$24         Ø15314R         SGS\$58         Ø03552R								001236R
SF1\$1         001304R         SGN\$15         012064R         SGN\$16         012702R           SGN\$18         013572R         SGN\$24         015444R         SGN\$35         017736R           SGN\$39         021604R         SGS\$24         015314R         SGS\$88         003552R							SFØS2	
SGN\$18         Ø13572R         SGN\$24         Ø15444R         SGN\$35         Ø17736R           SGN\$39         Ø21604R         SGS\$24         Ø15314R         SGS\$88         Ø03552R								
SGN\$39 021604R SGS\$24 015314R SGS\$8 003552R				1				
ுகையாருள் கூறு மானதற்றும் மன்றும் மாதியாம் குட்டுக்கு பட்ட கூட்டும் குட்டுக்கு இன்றைக்கு இட								
άδτατό κτοδιόυ ο όπταλο δοτούμοι ομιαττικώς ορογιά								
	991910	MICOLOU		-91905	56148VI		₩711 M/##	

162

()

È

2

( )

(

5

\$7

FPMP11 FLUATING POINT & MATH PA MACRU VR04-14 07-SEP-72 11:43 PAGE 52+ 163 SYMBUL TABLE

T

(

Q

Ŷ

	SHF 54 Sin	603202R 020064Rg	SIGN = SINGLE=	000002 000001	SIGNS = SL831	000000 001346R		
	SMES5	003266R	SMES6	003364R	SML\$38	020574R		
	SMT519	013740R	SMT\$20	014602R	SNC\$13	010000R	÷	- <u>-</u>
e! .	SNCS37	020100R	SNGL	017772RG	SN1824	015326R		
		6000006	SPC39	006532R	SORT	021650RG		
		177311	SRL52	002274R	SR529 =			
	SR8\$1	001400R	START .	000044	STCS10	007166R		
	STC33	002760R	STE\$38	020452R	STK310	007156R		
	STK\$3	002754R	STK\$42	000122R	STK\$46	022142R		
	STKS47	022224R	STM542	000376R	STR\$2	002416R		
	STR\$8	004564R	STS\$25	Ø15752R	STS59	006322R		
	STT524	015270R	ST4342	000140R	ST0342	000160R		
	SUBS1	001630R	SUB\$2	002446R	SUB\$25	015624R		
	\$1651		-58A51	001376R	TANH	020404RG		
	TAN538	020476R	T81542	000500R		000042		
	TNSS15	011462R	TN5539	021346R	TRAPH	000000RG		
	TSTS25	015616R	TWC519	Ø14476R	Tw1519	014502R		
	TYPE #	000014	UNDS1	001612R	UND316	Ø12514R		
	UNDS18	013402R	UND\$2	002536R	UND328	010614R		
	UND\$30	Ø17440R	UND\$8	004670R	UNFS1	001602R	· ·	
	UNF52	002526R	UPC342	000115R	UPLS14	010554R		
	UPLS15	011020R	UPL\$19	014116R	UPLS22	015176R		
	UPL\$23	015220R	UPL\$26	016042R	UPL\$38	020564R		
	UPLS39	021066R	UPL841	021724R	UPS10	007210R		
	UP\$15	Ø11776R	UPS3	002706R	UP\$38	020754R		
	UP\$39	021536R	UTS51	ØØ1576R	UTSS2	002522R		
	XCO\$9	005102R	XC189	005122R	XPD310	607054R		
	XPD\$13	010072R	XPD315	011712R	XSQS38	020620R		
	XØ328	016710R	X0530	017474R	X00528	015760R		
	X00530	Ø17522R	X4513	010176R	X4\$37	020234R		
	ZERS1	001622R	ZER\$13	010210R	ZERS14	010654R		
	ZERS16	Ø12526R	ZER817	013242R	ZER\$18	013436R		
•	ZERS19	014074R	ZERS2	002540R	ZERS20	015106R		
	ZER \$ 27	016144R	ZER\$28	Ø16254R	ZER\$29	017144R		
	ZERSJØ	Ø17452R	ZER\$31	Ø17552R	ZER\$35	017766R		
	ZER\$38	020746R	ZERS41	022000R	ZER S8	004100R		
ì.	ZE1528	016266R	ZE1830	017424R	ZE2\$28	016626R		
	ZFR\$20	015044R	27551	001710R	27552	002502R		
	ZT151	001760R	27251	001754R	Z1519	014106R		
	SADD	000704RG	SADR	002010RG	SCMD	003242RG		
	SCMR	003340RG	SDCI	003442RG	SDCO	005046RG		
	SDI	Ø1764ØRG	SDINT	007450RG	SOR	007622RG		
	\$ DVD	012210RG	5DVI	013130RG	SDVR	013256RG		
	SECO	005076RG	SERR	022006RG	SERRA	022016RG		
	SERVEC	022036RG	SFCALL	Ø15124RG	SFCO	005042RG		
	SGCO	005034RG	SICI	015230RG	\$100	015534RG		
	SID	016046RG	SINTR	003142RG	SIR	016062RG		
	SLDD	022056R	SLOR	022044R	SMLD	016146RG		
p L	SMLI	017002RG	5MLR	017162RG	SNGD	017542RG		
	SNGI	017534RG	SNGR	Ø17542RG	SOCI	015222RG		
	SOCO	015526RG	SPOLSH	021644RG	SPOPRS	017610RG		
	SPOPR4	017576RG	SPOPR5	017576RG	SPSHR1	017572RG		
	SPSHR2	Ø17572RG	SPSHR3	017570RG	SPSHR4	017564RG		
	SPSHR5	Ø17564RG	\$RCI	003454RG	SRD	017616RG		
	SRI	017650RG	\$SBD	000700RG	\$SBR	002004RG		
1	SSTD	022152R	STR	022076R	SV20A	021644RG		
		a carta carta						

FPMP11 FLUATING POINT & MATH PA MACRU VR04-14 07-SEP=72 11143 PAGE 52+ 164 SYMBUL TABLE

K.

17

(

1

Ş.

j)

, ABS, 000000 000 022242 001

ERRORS DETECTED: 0 FREE CURE: 14863, WORDS ,LP:<PR:,UT1:FPMP,MAC

#### INDEX

Absolute loader, 3-22, A-1, A-3, A-4 Addressing modes in TRAP calls, 3-2 ASCII conversion routines, 2-3, 3-12, 3-14 errors, 3-14 Assembler PAL-11S, B-1 error codes, B-3 Assembly instructions, conditional, 2-2 Assembly switch tape, 3-17, 3-20

33

1

1

2

Binary exponent, 1-3
Binary form of TRAP instruction, 3-4
Blanks in input conversion
 routines, 2-4
Bootstrap loader, 3-22, A-1, A-2,
 A-3

Calling conventions, 3-6 CLASS5 switch, 3-20 Conditional assembly instructions,2-2 source codes, 3-17, 3-19 Condition codes set by TRAP handler, 3-5 Conversion routines, 2-2 through 2-5

Data formats of conversion routines 2-3 Decimal position, 2-4 Division, 1-2 Double precision package, 2-2 storage, 1-3 switch, 3-2 D type conversion, 2-3, 2-5 scale factor, 2-4

Entry points, 3-1 Errors ASCII conversion routine, 3-14 assembly, 3-21 3-20 CLASS5, floating point operations, 3-15 FPMP-11 code, 3-16 PAL-11S assembler, B-3 user error handling routines, 3-15 user floating point, 3-1 E type conversion, 2-3 scale factor, 2-4 Excess 128 code, 1-3 Exponent in floating point notation, 1-2

Fixed point notation, 1-2 FLAC see Floating point accumulator Floating point accumulator (FLAC), 3-1 restrictions, 3-1 Floating point compare, 3-5 Floating point notation, 1-1, 1-2 double precision, 1-3 normalized, 1-2 single precision, 1-3 storage, 1-3 FORTRAN library functions, 3-10 FORTRAN operating time system, D-2 1-2 Fractional representation, F type conversion, 2-3 scale factor, 2-4

Global entry points of FPMP-11, D-1
.GLOBL assembler directive, 3-2
G type output conversions, 2-3, 2-5
scale factor, 2-4

Hardware option switches, 3-17, 3-18 Hardware requirements, 1-1

Immediate mode, 3-2
Input conversion routines, blanks
 in, 2-4
Integer conversion, 2-5
I type conversion, 2-3, 2-4

J5RR mode, 3-6, 3-10 storage of results, 3-11 JPC mode subroutine call, 3-12, 3-13 LINK 11S linker, 3-22, C-1 error halts, C-6 error procedure and messages, C-3 fatal errors, C-4 non-fatal errors, C-4 restarting, C-3 Loaders, 3-22, A-1 Loading instructions, FPMP-11 package, 3-22

Modes in TRAP calls, 3-2 Multiplication, 1-2

Negative exponent, 1-2 Normalized numbers, 1-2, 1-4 Number storage, 1-3

One-word integer conversion, 2-5 OTS routines, 3-1, D-2 called directly, 3-8 non-OTS routines, D-8 O type conversion, 2-3, 2-4 PAL-11S Assembler, B-1 error codes, B-3 operating procedures, B-1 Polish mode, 3-6 calls, 3-8 through 3-10 @R0 mode, 3-2 3-2 Relative mode, Routines accessed by TRAP handler, D-8 Sample program, 4-1 Scale factor, 2-4 Single precision package, 2-1
switch, 3-20 word storage, 1-3 Software requirements, 1-1 Source listing, E-1 Source tapes, 2-2 Special packages, creation of, 3-17 Stack mode, 3-2 Storage of floating-point numbers, 1-3 Switch tape assembly, 3-17 Switches CLASS5, 3-20 double precision, 3-20 single precision, 3-20 Summary FPMP-11 routines, D-1 Symbol table size limitations, 3-21 Transcendental functions, 3-20 TRAP calls addressing modes, 3-2 binary form, 3-4 source form, 3-4 TRAP handler, 3-2 for OTS routines, 3-1 routines, D-8 user routines, 3-6, 3-7 Trigonometric functions, 3-20 Two-word integer conversion, 2-5

User error handling routines, 3-15

ť

Y

1

### HOW TO OBTAIN SOFTWARE INFORMATION

#### SOFTWARE NEWSLETTERS, MAILING LIST

The Software Communications Group, located at corporate headquarters in Maynard, publishes software newsletters for the various DIGITAL products. <u>Newsletters</u> are published monthly, and keep the user informed about customer software problems and solutions, new software products, documentation corrections, as well as programming notes and techniques.

There are two similar levels of service:

The Software Dispatch

The Digital Software News

The Software Dispatch is part of the Software Maintenance Service. This service applies to the following software products:

PDP-9/15 RSX-11D DOS/BATCH RSTS-E DECsystem-10

ुने

T

1

Ŕ

A Digital Software News for the PDP-11 and a Digital Software News for the PDP-8/12 are available to any customer who has purchased PDP-11 or PDP-8/12 software.

A collection of existing problems and solutions for a given software system is published periodically. A customer receives this publication with his initial software kit with the delivery of his system. This collection would be either a Software Dispatch Review or Software Performance Summary depending on the system ordered.

A mailing list of users who receive software newsletters is also maintained by Software Communications. Users must sign-up for the newsletter they desire. This can be done by either completing the form supplied with the Review or Summary or by writing to:

> Software Communications P.O. Box F Maynard, Massachusetts 01754

#### SOFTWARE PROBLEMS

Questions or problems relating to DIGITAL's software should be reported as follows:

#### North and South American Submitters:

Upon completion of Software Performance Report (SPR) form remove last copy and send remainder to:

Software Communications P.O. Box F Maynard, Massachusetts 01754

The acknowledgement copy will be returned along with a blank SPR form upon receipt. The acknowledgement will contain a DIGITAL assigned SPR number. The SPR number or the preprinted number should be referenced in any future correspondence. Additional SPR forms may be obtained from the above address.

#### All International Submitters:

Upon completion of the SPR form, reserve the last copy and send the remainder to the SPR Center in the nearest DIGITAL office. SPR forms are also available from our SPR Centers.

#### PROGRAMS AND MANUALS

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center.

Digital Equipment Corporation Software Distribution Center 146 Main Street Maynard, Massachusetts 01754 Digital Equipment Corporation Software Distribution Center 1400 Terra Bella Mountain View, California 94043

f

7

5

Outside of the United States, orders should be directed to the nearest Digital Field Sales Office or representative.

#### USERS SOCIETY

DECUS, Digital Equipment Computers Users Society, maintains a user exchange center for user-written programs and technical application information. The Library contains approximately 1,900 programs for all DIGITAL computer lines. Executive routines, editors, debuggers, special functions, games, maintenance and various other classes of programs are available.

DECUS Program Library Catalogs are routinely updated and contain lists and abstracts of all programs according to computer line:

- PDP-8, FOCAL-8, BASIC-8, PDP-12
- . PDP-7/9, 9, 15
- . PDP-11, RSTS-11
- . PDP-6/10, 10

Forms and information on acquiring and submitting programs to the DECUS Library may be obtained from the DECUS office.

In addition to the catalogs, DECUS also publishes the following:

DECUSCOPE

-The Society's technical newsletter, published bi-monthly, aimed at facilitating the interchange of technical information among users of DIGITAL computers and at disseminating news items concerning the Society. Circulation reached 19,000 in May, 1974.

PROCEEDINGS OF<br/>THE DIGITAL-Contains technical papers presented at DECUS Symposia<br/>held twice a year in the United States, once a yearEQUIPMENT USERS<br/>SOCIETYin Europe, Australia, and Canada.

MINUTES OF THE -A report of the DECsystem-10 sessions held at the two DECsystem-10 United States DECUS Symposia.

COPY-N-Mail -A monthly mailed communique among DECsystem-10 users.

LUG/SIG

SESSIONS

-Mailing of Local User Group (LUG) and Special Interest Group (SIG) communique, aimed at providing closer communication among users of a specific product or application.

Further information on the DECUS Library, publications, and other DECUS activities is available from the DECUS offices listed below:

DECUS Digital Equipment Corporation 146 Main Street Maynard, Massachusetts 01754 DECUS EUROPE Digital Equipment Corp. International (Europe) P.O. Box 340 1211 Geneva 26 Switzerland

FPMP-11 USER'S MANUAL DEC-11-NFPMA-C-D

#### READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Report (SPR) form (see the HOW TO OBTAIN SOFTWARE INFORMATION page).

Did you find errors in this manual? If so, specify by page.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

Please indicate the type of user/reader that you most nearly represent.

Higher-level language programmer

Occasional programmer (experienced)

User with little programming experience

Student programmer

T

1

ease cut along this line

١Ē

1

Non-programmer interested in computer concepts and capabilities

Name		Date				
Organization						
organización <u></u>						
Street						
$\sum_{i=1}^{n} e_{i} e_{i} = \frac{1}{2} \sum_{i=1}^{n} e_{i} = \frac{1}{2} \sum_{i=1$						
City	State		Zip Code_			
hand the stand of the second second			or			
			Country			
If you do not require a	written reply, p	lease ch	eck here.			

FIRST CLASS PERMIT NO. 33 MAYNARD, MASS. Ç

D

2

1

BUSINESS REPLY MAIL NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:



Software Communications P. O. Box F Maynard, Massachusetts 01754

— — — Fold Here – – – – –

----- Do Not Tear - Fold Here and Staple ------



DIGITAL EQUIPMENT CORPORATION MAYNARD, MASSACHUSETTS 01754