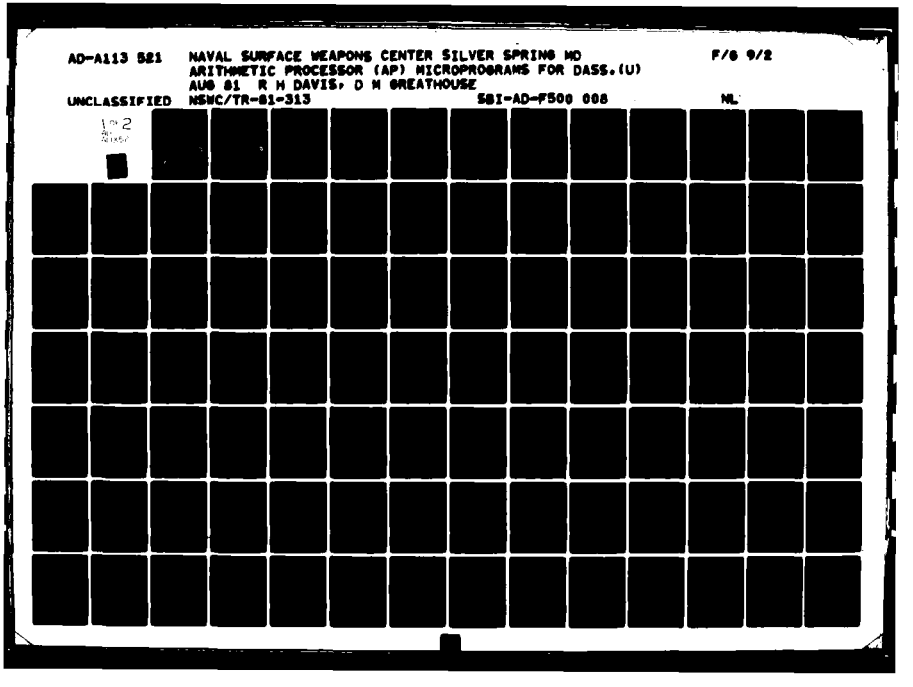
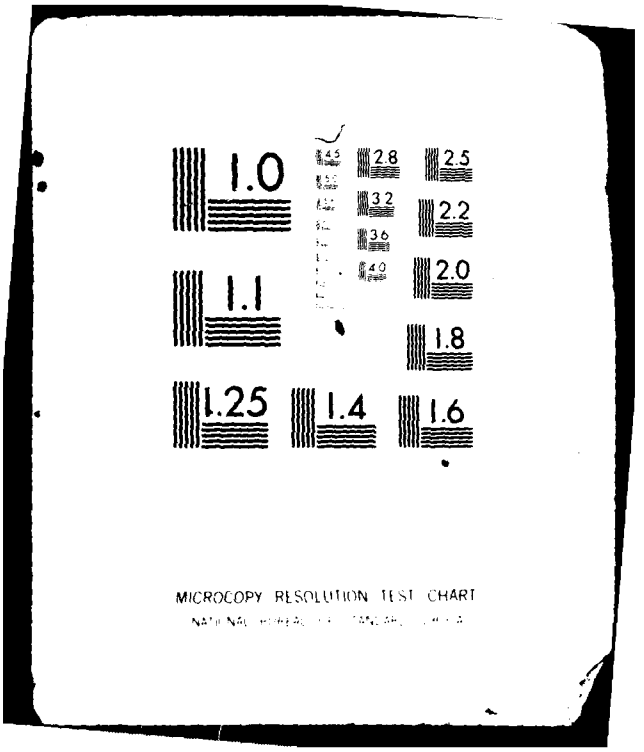

AD-A113 521
NAVAL SURFACE WEAPONS CENTER SILVER SPRING MD
F/G 9/2
ARITHMETIC PROCESSOR (AP) MICROPROGRAMS FOR DASS.(U)
AUG 81 R H DAVIS, D H BREATHOUSE
UNCLASSIFIED
NSWC/TR-81-313
SBI-AD-F500 008
NL


1.0
1.1
1.25
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

(11)

NSWC TR 81-313

ARITHMETIC PROCESSOR (AP) MICROPROGRAMS FOR DASS

BY ROBERT H. DAVIS DOROTHY M. GREATHOUSE

UNDERWATER SYSTEMS DEPARTMENT

4 AUGUST 1981

NAVAL SURFACE WEAPONS CENTER

Dahlgren, Virginia 22448 • Silver Spring, Maryland 20910

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NSWC TR 81-313	2. GOVT ACCESSION NO. AD-A113 521	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ARITHMETIC PROCESSOR (AP) MICROPROGRAMS FOR DASS		5. TYPE OF REPORT & PERIOD COVERED Final - 1976-1979 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert H. Davis Dorothy M. Greathouse		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Surface Weapons Center White Oak Silver Spring, Maryland 20910		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64261, 01102, A5495490/ 004D/1W11020000, 1U34DF
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Department of the Navy Washington, D. C. 20361		12. REPORT DATE 4 August 1981 13. NUMBER OF PAGES 130
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Arithmetic Processor (AP)
Microprogram (MACRO)
Advanced Signal Processor (ASP)
Digital Acoustic Sensor Simulator (DASS)
Digital Signal Processing
Digital Signal Generation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the Arithmetic Processor microprograms developed for the passive synthesis portion of the Digital Acoustic Sensor Simulator (DASS) program.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This report describes the Arithmetic Processor (AP) microprograms (or MACROS) developed for the passive synthesis portion of the Digital Acoustic Sensor Simulator (DASS) program. These microprograms are written for the Arithmetic Processor of the AN/UYS-1 Advanced Signal Processor (ASP) built by IBM, Inc. Approximately half of these programs are highly specialized for the DASS application and are of limited general use. The others have more general application and should be of interest to other ASP users.

The work described herein was sponsored by Naval Air Systems Command, NAIR-370, under Airtask 5495490/004D/1W11020000.

F. B. SANCHEZ
By direction

CONTENTS

Chapter		Page
1	INTRODUCTION	1-1
2	BROADBAND AMPLITUDE (BBAMP) MACRO	2-1
3	DISCRETE LINE SPECTRA GENERATION (DSCLN) MACRO	3-1
4	SINGLE REAL FFT (SRFFT) MACRO	4-1
5	SCALE AND ADD TWO ARRAYS (SCLA2) MACRO	5-1
6	ASYNCHRONOUS SAMPLE, SCALE AND SUM (ASSSS) MACRO	6-1
7	SCALE AND ADD THREE ARRAYS (SCLA3) MACRO	7-1
8	FILTER TWO AND SUM THREE ARRAYS (F22S3) MACRO	8-1
9	DEMODULATED NOISE (DEMON) MACRO	9-1
APPENDIX A	AP MICROPROGRAM SOURCE LISTINGS	A-1

ILLUSTRATIONS

Figure		Page
3-1	CONSTRUCTION OF HANNING APPROXIMATION	3-2
3-2	DSCLN ILLUSTRATION OF TERMINOLOGY	3-3
3-3	COMPARISON OF APPROXIMATION AND EXACT HANNING FUNCTION	3-5
6-1	ASSSS INPHASE MAIN LOOP FLOW	6-4
6-2	ASSSS OUTPHASE MAIN LOOP FLOW	6-5

TABLES

Table		Page
2-1	BBAMP AP MICROPROGRAM SCALING	2-5
2-2	BBAMP AEC REGISTER MAP	2-6
2-3	BBAMP AE REGISTER MAP	2-7
2-4	BBAMP/DSCLN WORKING STORE MAP	2-8
2-5	BBAMP PROGRAM CODING CHART	2-9
3-1	DSCLN AP MICROPROGRAM SCALING	3-10
3-2	DSCLN AEC REGISTER MAP	3-13
3-3	DSCLN AE REGISTER MAP	3-14
3-4	DSCLN PROGRAM CODING CHART	3-15
4-1	SRFFT AP MICROPROGRAM SCALING	4-5
4-2	SRFFT AEC REGISTER MAP	4-6
4-3	SRFFT AE REGISTER MAP	4-7
4-4	SRFFT WORKING STORE MAP	4-8
4-5	SRFFT PROGRAM CODING CHART	4-9
5-1	SCLA2 AP MICROPROGRAM SCALING	5-4
5-2	SCLA2 AEC REGISTER MAP	5-5
5-3	SCLA2 AE REGISTER MAP	5-6
5-4	SCLA2 WORKING STORE MAP	5-7
5-5	SCLA2 PROGRAM CODING CHART	5-8
6-1	ASSSS AP MICROPROGRAM SCALING	6-11
6-2	ASSSS AEC REGISTER MAP	6-12
6-3	ASSSS AE REGISTER MAP	6-13
6-4	ASSSS WORKING STORE MAP	6-14
6-5	ASSSS PROGRAM CODING CHART	6-15
7-1	SCLA3 AP MICROPROGRAM SCALING	7-4
7-2	SCLA3 AEC REGISTER MAP	7-5

TABLES (Cont.)

Table		Page
7-3	SCLA3 AE REGISTER MAP	7-6
7-4	SCLA3 WORKING STORE MAP	7-7
7-5	SCLA3 PROGRAM CODING CHART.	7-8
8-1	F22S3 AP MICROPROGRAM SCALING	8-4
8-2	F22S3 AEC REGISTER MAP.	8-5
8-3	F22S3 AE REGISTER MAP	8-6
8-4	F22S3 WORKING STORE MAP	8-7
8-5	F22S3 PROGRAM CODING CHART.	8-8
9-1	DEMON AP MICROPROGRAM SCALING	9-5
9-2	DEMON AEC REGISTER MAP.	9-6
9-3	DEMON AE REGISTER MAP	9-7
9-4	DEMON WORKING STORE MAP	9-8
9-5	DEMON PROGRAM CODING CHART.	9-9

CHAPTER 1

INTRODUCTION

This report describes the Arithmetic Processor (AP) microprograms (or MACROS) developed for the passive synthesis portion of the Digital Acoustic Sensor Simulator (DASS) program. These microprograms are written for the Arithmetic Processor of the AN/UYS-1 Advanced Signal Processor (ASP) built by IBM, Inc. Approximately half of these programs are highly specialized for the DASS application and are of limited general use. The others have more general application and should be of interest to other ASP users. A brief description of each MACRO is given below. A complete description of each MACRO is provided in its corresponding chapter. Chapter format, for the most part, conforms to a previous IBM publication.[1] Each chapter contains a functional description of the algorithm implemented by the microprogram, a mathematical description giving the exact computations performed by the microcode, a description of the microcode implementation from the viewpoint of sequencing and control, and tabulation of register usage and program statistics. In addition, each chapter contains detailed charts for:

a) Program coding and timing
b) Arithmetic Element Controller (AEC) and Arithmetic Element (AE) register assignments
c) Working Store (WS) layout of input and output data
d) Scaling analysis of input, intermediate and output data.

The program coding charts are a fixed-format representation of the sequence of micro-instructions for each microprogram. Each line on the chart represents the events occurring during one 100 ns microstep. The register assignment charts list the computational elements which are stored in each AE register during microprogram execution. The Working Store layouts illustrate the WS data organization used in DASS, although other equivalent layouts could be used. The scaling analysis

[1] Proteus AU Microprogram Design Document, Code Ident 6259764, IBM, Inc., 01 Aug 1976.

charts list each partial result of the computations and give its fixed-point scaling in terms of the assumed position of the binary point. An Fx scaling indicates that the partial result is a 32-bit (fullword) number with the binary point x bits to the left (x>0) or right (x<0) of the midpoint of the word (16 bits from either end). An Hx scaling indicates a 16-bit (halfword) number with an assumed binary point x bits to the left (x>0) or right (x<0) of the right end of the halfword.

Appendix A contains the source listings for each microprogram.

BBAMP (BROADBAND AMPLITUDE)

BBAMP weights an array of complex spectral components by a set of spectral amplitudes and convolves the result with the spectrum of a sine pulse. This MACRO is peculiar to DASS and has limited application to other problems.

DSCLN (DISCRETE LINE SPECTRA GENERATION)

DSCLN generates the appropriate spectral components for discrete line components of arbitrary frequency, amplitude and phase and adds these spectral components to the Broadband spectrum generated by BBAMP. This MACRO is also peculiar to DASS and has limited application to other problems.

SRFFT (SINGLE REAL FFT)

SRFFT performs the additional pass or operation necessary, in addition to the basic (complex to complex) FFT algorithm, to effect transforms between real (time) arrays and single sided complex (frequency) arrays. This MACRO should be of general interest and applicability.

SCLA2 (SCALE AND ADD TWO ARRAYS)

SCLA2 provides the combination of two arrays, X and Y, to produce an output array Z. Provision is made for independent scaling of each of the two input arrays. This MACRO is of general applicability.

ASSSS (ASYNCHRONOUS SAMPLE, SCALE AND SUM)

ASSSS resamples an input array at a rate which is non-integrally related to the origin sample rate. Linear interpolation is performed to reduce aliasing. Resampled data is then scaled and added to a second array. This MACRO is of some, but not extensive, general usefulness.

SCLA3 (SCALE AND ADD THREE ARRAYS)

SCLA3 provides the combination of three arrays X, Y, and Z to produce an output array W. Provision is made for independent scaling of each of the three input arrays. The MACRO is of general applicability.

F22S3 (FILTER TWO AND SUM THREE ARRAYS)

F22S3 is intended for filtering each of two inputs and adding the sum of the filter outputs to a third input. Each of the two filters is a two-pole recursive filter with independent parameters. This MACRO may be of use in other applications.

DEMON (DEMODULATED NOISE)

DEMON is used to generate two signal components, each consisting of a broadband signal modulated by a periodic signal. The MACRO additionally combines the two components with a third input to produce a composite output signal. This MACRO is of some, but not extensive, general usefulness.

CHAPTER 2

BROADBAND AMPLITUDE (BBAMP) MACRO

INTRODUCTION

The Broadband spectra generation algorithm is described in detail in a previous publication.[2] Briefly, the broadband spectra is described by an array A of 860 numbers representing the spectral amplitude at 860 frequencies evenly spaced across the band of interest. The 860 frequencies are given by:

$$f_n = f_c(n - \tfrac{1}{2}); \; n = 1, 2, \ldots, 860$$

where f_c is the DFT cell width. Each specific spectral array B is generated by the following algorithm:

$$BR_i = \tfrac{1}{2}\,(A_i RR_i - A_{i+1} RR_{i+1})$$

$$BI_i = \tfrac{1}{2}\,(A_i RI_i - A_{i+1} RI_{i+1})$$

$$BR_0 = BI_0 = 0$$

A_{860} must be 0

$BR_i = BI_i = 0$ for i greater than 860

where BR_i and BI_i are the real and imaginary components of the ith cell of the output spectral array B, i=0 to 1023, and RR_i and RI_i are independent samples from a Gaussian random process with zero mean and variance of $\frac{1}{2}$.

The DFT (single sided complex to real) of the B array produces a 2048 sample time waveform segment with a sine pulse weighting. Successive segments generated in this manner are overlapped by 1024 samples and combined to form the output broadband time waveform. Since the segments are uncorrelated (all RR's, RI's are independent) the overlapped segments add incoherently to yield a random function having the desired average spectral behavior and stationary first order

[2]DAVIS, R. H., "Synthesis of Steady-State Signal Components by an All-Digital System", NOLTR 74-215, Naval Ordnance Laboratory, (Now Naval Surface Weapons Center) 05 Dec 1974.

statistics.

FUNCTIONAL DESCRIPTION

BBAMP MACRO consists of four steps to obtain the non-zero broadband spectrum. The first step is to generate a "variator" factor which is a continuous, piecewise linear function defined over the non-zero amplitudes. This function is produced by beginning at each breakpoint with the value of the function and adding an increment (signed) for each successive value until the next breakpoint is reached. The second step is to multiply the AMP element by the variator value generated as described above to produce the amplitude for that spectral cell. This amplitude is then used in the third step to scale the two components of the corresponding RN (Random Number) array element to produce the "half cell" frequency components. The final step is to form the output array element by linearly combining the i^{th} half cell components with the previous ($(i-1)^{st}$) half cell components.

MATHEMATICAL DESCRIPTION

Phase 1: $CAV_i = CAV_{i-1} + DAV_j$

where CAV_{i-1} is the variator function value at the last cell and DAV_j is the increment value for the (current) j^{th} linear segment. If i corresponds to the $j+1^{st}$ breakpoint,

$$CAV_i = AV_{j+1}$$

where AV is the initial value for the j+1 segment obtained from the VAR buffer.

Phase 2: $VA_i = CAV_i * A_i$

where A_i is the AMP value for the i^{th} cell, and VA_i is the variated amplitude.

Phase 3: $BHR_i = VA_i * RR_i$

$BHI_i = VA_i * RI_i$

where RR_i and RI_i are the real and imaginary values, respectively, of the i^{th} RN element, and BHR_i and BHI_i are the properly scaled components for the "half-cell" frequencies.

Phase 4: $BR_i = BHR_i - BHR_{i-1}$

$BI_i = BHI_i - BHI_{i-1}$

where BR_i and BI_i are the final output components of the i^{th} element of the BB array.

IMPLEMENTATION

BBAMP is implemented with a ten-instruction main loop which processes one

pair of A's (amplitudes), R's (complex random numbers) and B's (output broadband spectral cells). An eighteen-instruction preamble loads the pipe such that two valid B's are produced on each pass through the main loop. An eight-instruction segment of code contained within the main loop is skipped except when a new set of variator parameters is required for a new linear segment. The variator array is stored as two 32-bit word pairs - the first word contains the initial variator value for the i^{th} corresponding linear segment in the high order 16 bits, and a count parameter determining the number of pairs of amplitudes for which the segment is to apply in the low order 16 bits. The second word contains the increment value for the segment. The count parameter is interpreted as the number of amplitude pairs - 1, and thus a count of zero produces a segment of length 2. The first count parameter is interpreted by the preamble as number of pairs - 2; and therefore, the minimum length of the first segment is 4. The count is tested for zero and decremented each pass through the main loop. The embedded variator function change code is executed if the zero test is met.

The inside loop is controlled by BNZ8. An outside loop controlled by BNZ9 allows arrays of greater than 512 points to be processed. BBAMP is intended to be combined with DSCLN in a SUPER MACRO, and returns to the SUPER MACRO Code via BR4 upon completion.

SCALING

See Table 2-1.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 2-2.

ADDRESS REGISTERS

AR0 - The increment code is initialized to 1. The address portion is initialized to the address of the AMP buffer.

AR1 - The increment code is initialized to 1. The address portion is initialized to the address of the VAR buffer.

AR2 - The increment code is initialized to 1. The address portion is initialized to the address of the RN buffer.

AR3 - The increment code is initialized to 1. The address portion is initialized to the address of the output BB buffer. An initial dummy read with a "subtract increment" directive effectively initializes the address portion to BB-1.

INCREMENT REGISTERS

INC 0, 1, 2, 3 - Wrap code must be 1024.

BRANCH REGISTERS

BR4 - Unconditional branch to return to SUPER MACRO at completion of BBAMP. If BBAMP and DSCLN are not combined by a SUPER MACRO, BR4 should branch to STOP.

BCR0 - Conditional branch over the variator change portion of the main loop. Conditioned on the sign of the incremented count CT.

BNZ8 - Branch register controlling the number of times the inner loop is executed.

BNZ9 - Branch register controlling the number of times the outer loop is executed. Total number of data pairs processed is product of inner times outer loop executions. BNZ9 allows processing of arrays longer than 512 points.

SINE/COSINE DESTINATION REGISTERS

Not applicable.

DECIMATE REGISTER

Not applicable.

SCALE FACTOR REGISTERS

SFOH - All True/Direct.

SFOL - Left, center, and right prescalers - True/Direct, postscaler - True/Left 4.

ARITHMETIC ELEMENT REGISTER MAP

See Table 2-3.

WORKING STORE MAP

See Table 2-4.

PROGRAM CODING CHART

See Table 2-5.

TABLE 2-1 BBAMP AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
AV	H12		
CT	H0		
DV	F12		
A		H12	
R		H12	
INTERMEDIATE RESULTS:			
DAV	F12		
CAV	F12	H12	
VA		H12	
BHR, BHI	F8		
OUTPUTS:			
BR, BI			H12

TABLE 2-2 BBAMP AEC REGISTER MAP

Incr Code Address Register

	0	3 4 15
0	1	A(AMP)
1	1	A(VF)
2	1	A(RN)
3	1	A(BB)
4		
5		
6		
7		

WS Wrap Increment Register

	0	3 4 15
0	Ø	
1	Ø	
2	Ø	
3	Ø	
4		
5		
6		
7		

Memory Address Register

BBAMP

Sin/Cos Destination Register

	AE 0	AE 1	AE 2	AE 3
0				
1				
2				
3				

Decimate Register

Count	Reset

Comments

N = number of spectral cells to process.

BR/BCR Unconditional/ Conditional Branch

Branch Address

	0 15
0	BB2
1	
2	
3	
4	STOP
5	
6	
7	

BNZ Branch and Count Registers

	0 7	8 15	16 21	22 31
8	$\frac{N}{4} - 1$	$\frac{N}{4} - 1$	X	BB1
9	1	1	X	BB1
A			X	
B			X	
C			X	
D			X	
E			X	
F			X	

TABLE 2-3 BBAMP AE REGISTER MAP

MLR

ØH	M1,CAV1	CAV2	ØL
1H	VA1	VA2	1L
2H	AV1	CT1	2L
3H	DUMMY	READ	3L

MRR

ØH	M1, A1	A2	ØL
1H	RR1	RI1	1L
2H	RR2	RI2	2L
3H			3L

ALIR

ØH	BH R1		ØL
1H	BH I1		1L
2H	AV	CT	2L
3H	D V		3L

ACIR

ØH	P1,	BHR2	ØL
1H	BHI2		1L
2H			2L
3H			3L

ARIR

Ø	TEMP
1	
2	
3	

AEOR

ØH			ØL
1H	BR1	BI1	1L
2H	BR2	BI2	2L
3H			3L

SCALE FACTOR REGISTERS

AEØ

ØH	Ø	Ø	Ø	Ø	Ø	Ø	Ø	3	ØL
1H									1L
2H									2L
3H									3L

AE1

ØH									ØL
1H									1L
2H									2L
3H									3L

ALOR

Ø	
1	
2	
3	

ACOR

Ø	CAV
1	CT
2	
3	

AROR

Ø	DAV
1	P1
2	
3	

TABLE 2-4 BBAMP/DSCLN WORKING STORE MAP

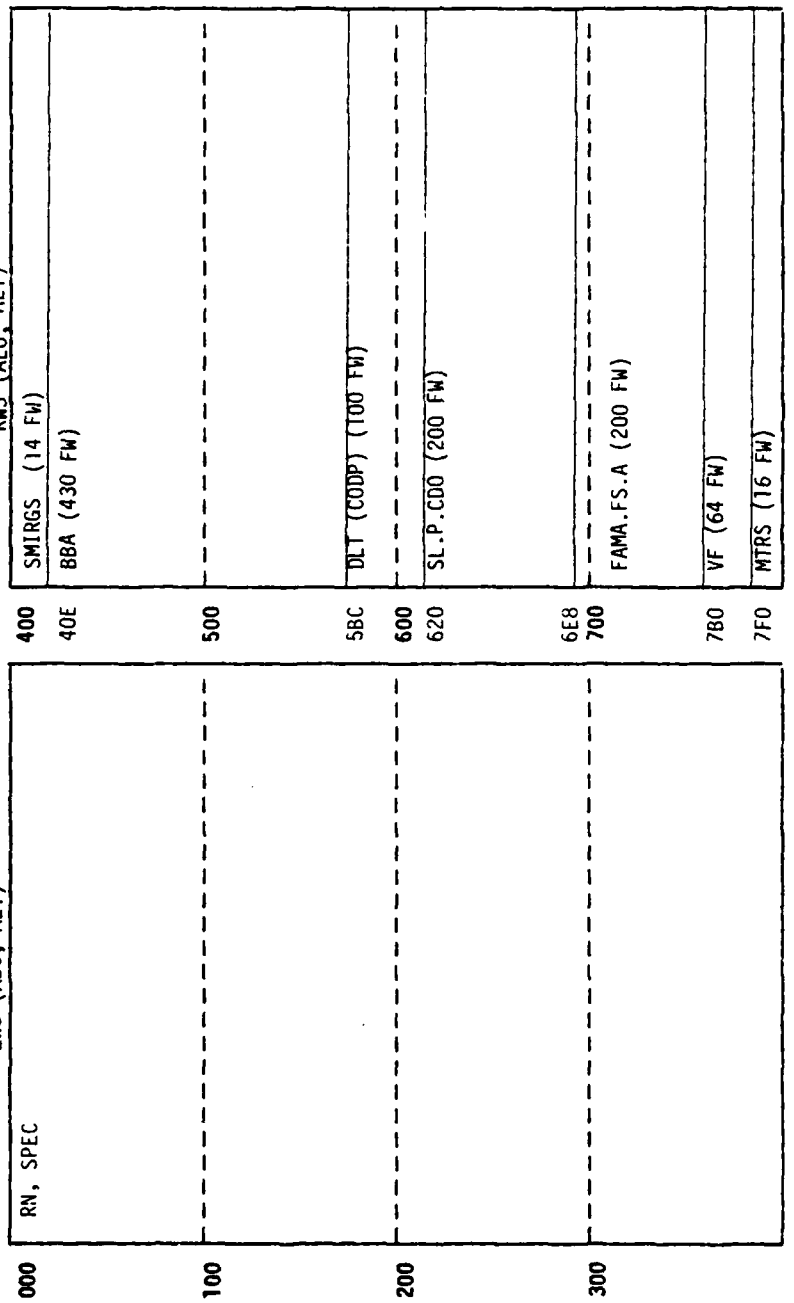


TABLE 2-5 BBAMP PROGRAM CODING CHART

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	BBAMP		AR1 +Ø							M1 = NOT Ø L				
			READ AV1,CT1											
			AR1 +1	M1 ØH						M1 = NOT Ø L				
			READ DV1	AV1,CT1 2			AV1,CT1 2							
			ARØ +Ø		M1 ØH									
			READ A1, A2			P1 = M1*M1	DV1 3							
										DAV = DV1 ØH				
					A1,A2 Ø			P1 Ø					DAV Ø	
			AR3 -1			VA1 = AV1*A1				CAV2=AV1+DAV ØH				
			DUMMY READ							P1 = LO(P1) ØH		CAV2 Ø		
			AR2 +Ø	CAV2 ØL					VA1 Ø	CT = -CT1 ØH R			P1 1	
			READ R1	DUMMY 3		VA2=CAV2*A2				VA1 ØL		CT 1		
			AR2 +1											
			READ R2	VA1 1H	R1 1				VA2 Ø	R				
			ARØ +1			BHR1=VA1*RR1				VA2 ØL				
			READ A3,A4		R2 2	BHI1=VA1*RI1				CAV3=CAV2+DAV ØH				
				VA2 1L			BHR1 Ø			R		CAV3 Ø		
				CAV3 ØH	A3,A4 Ø	BHR2=VA2*RR2	BHI1 1			CAV4=CAV3+DAV ØH R				
						VA3=CAV3*A3				B1 = Ø ØH		CAV4 Ø		
				CAV4 ØL		BHI2=VA2*RI2		BHR2 Ø		B1 = Ø ØH R				

TABLE 2-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	BB1		AR2 +1			VA4=CAV4*A4			VA3 Ø	CT = CT + P1 ØH R				BR1 1H
			READ R3					BHI2 1		SETCS VA3 ØL R		CT 1		BI1 1L
			AR2 +1						VA4 Ø	BR2=BHR2-BHR1 ØL				
	BCRØ		READ R4	VA3 1H	R3 1					VA4 ØL R				
			ARØ +1			BHR3=VA3*RR3				BI2=BHI2-BHI1 ØL R				BR2 2H
			READ A5,A6	VA4 1L	R4 2	BHI3=VA3*RI3				CAV5=CAV4+DAV ØH R				
			AR1 +1							R				BI2 2L
			READ AV,CNT		A5,A6 Ø									
			AR1 +1											
			READ DV				AV,CNT 2							
							DV 3			CT = -CNT ØH				
						BHR3=VA3*RR3				DAV = DV ØH		CT 1		
						BHI3=VA3*RI3				CAV = AV ØH			DAV Ø	
	BB2		AR3 +1				BHR3 Ø			R		CAV5 Ø		BI2 2L
	BNZ8		WRITE B1	CAV5 ØH	A5,A6 Ø	BHR4=VA4*RR4	BHI3 1			CAV6=CAV5+DAV ØH R				
			AR3 +1			VA5=CAV5*A5				BR3=BHR3-BHR2 ØL		CAV6 Ø		
			WRITE B2	CAV6 ØL		BHI4=VA4*RI4		BHR4 Ø		BI3=BHI3-BHI2 ØL R				

TABLE 2-5 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD		ALOR	ACOR	AROR	AEOR
											R				BR3 11
	BNZ9										R				BI3 1L
						VA5=CAV5*A5									
						BHI4=VA4*RI4				BI3=BHI3-BHI2	ØL				
						NOP									
	BR4														
						NOP									
						NOP									

CHAPTER 3

DISCRETE LINE SPECTRA GENERATION (DSCLN) MACRO

INTRODUCTION

The discrete line spectra generation algorithm is described in detail in a previous publication.[3] Briefly, for each discrete line to be added to the spectrum of each 2048 sample time waveform generated, complex components are added to six frequency cells - three on each side of the desired frequency. The magnitudes and phases of the six complex components are computed from the desired frequency and amplitude for the segment to be generated and from the frequency and phase of the preceding segment.

The exact amplitude weighting for the six components is given by the periodic Hanning function.[4] For DSCLN MACRO, a three-segment approximation is used based on sine and cosine functions derived from the fractional part of the desired frequency. Figure 3-1 illustrates the use of quarter-cycle and half-cycle trig functions to generate the desired weighting function. In Figure 3-2, for a desired spectral line of amplitude A at C + D (C, the next lower cell index and D, the fractional distance to the specified line position) the following computations are performed to obtain the magnitudes for the six cell components.

For cell c - 2:

$$M_{c-2} = A(.035*\sin(\pi D)*\cos(\frac{\pi D}{2}))$$

$$= A(.07*\sin(\frac{\pi D}{2})*\cos^2(\frac{\pi D}{2}))$$

For cell c - 1:

$$M_{c-1} = A(.5 - .5*\sin(\frac{\pi D}{2}) + .026*\sin(\pi D))$$

$$= A(.5 - .5*\sin(\frac{\pi D}{2}) + .052*\sin(\frac{\pi D}{2})*\cos(\frac{\pi D}{2}))$$

---

[3] DAVIS, R. H., "Synthesis of Steady-State Signal Components by an All-Digital System", NOLTR 74-215, Naval Ordnance Laboratory (now Naval Surface Weapons Center), 05 Dec 1974.

[4] Ibid.

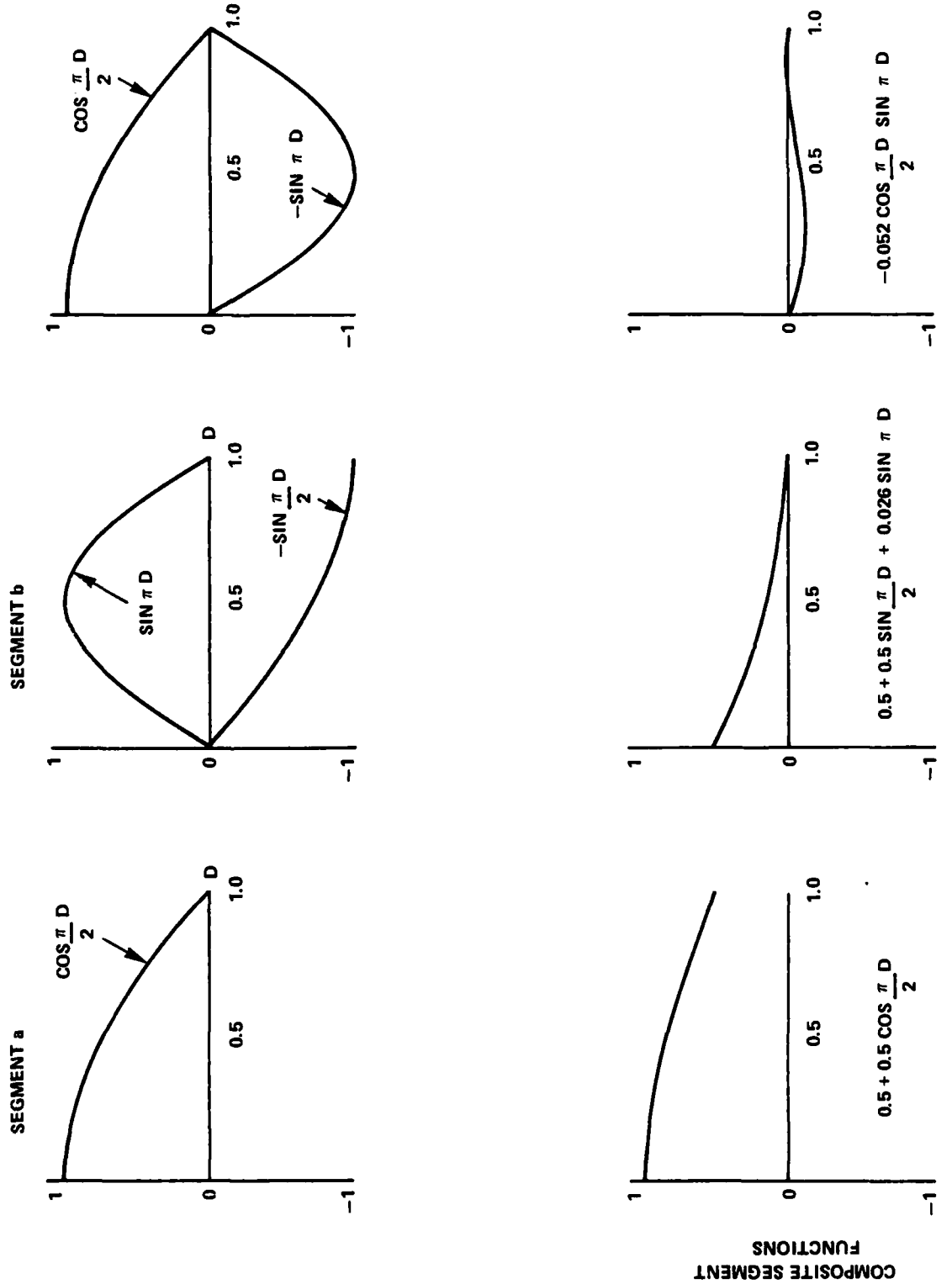


FIGURE 3-1 CONSTRUCTION OF HANNING APPROXIMATION

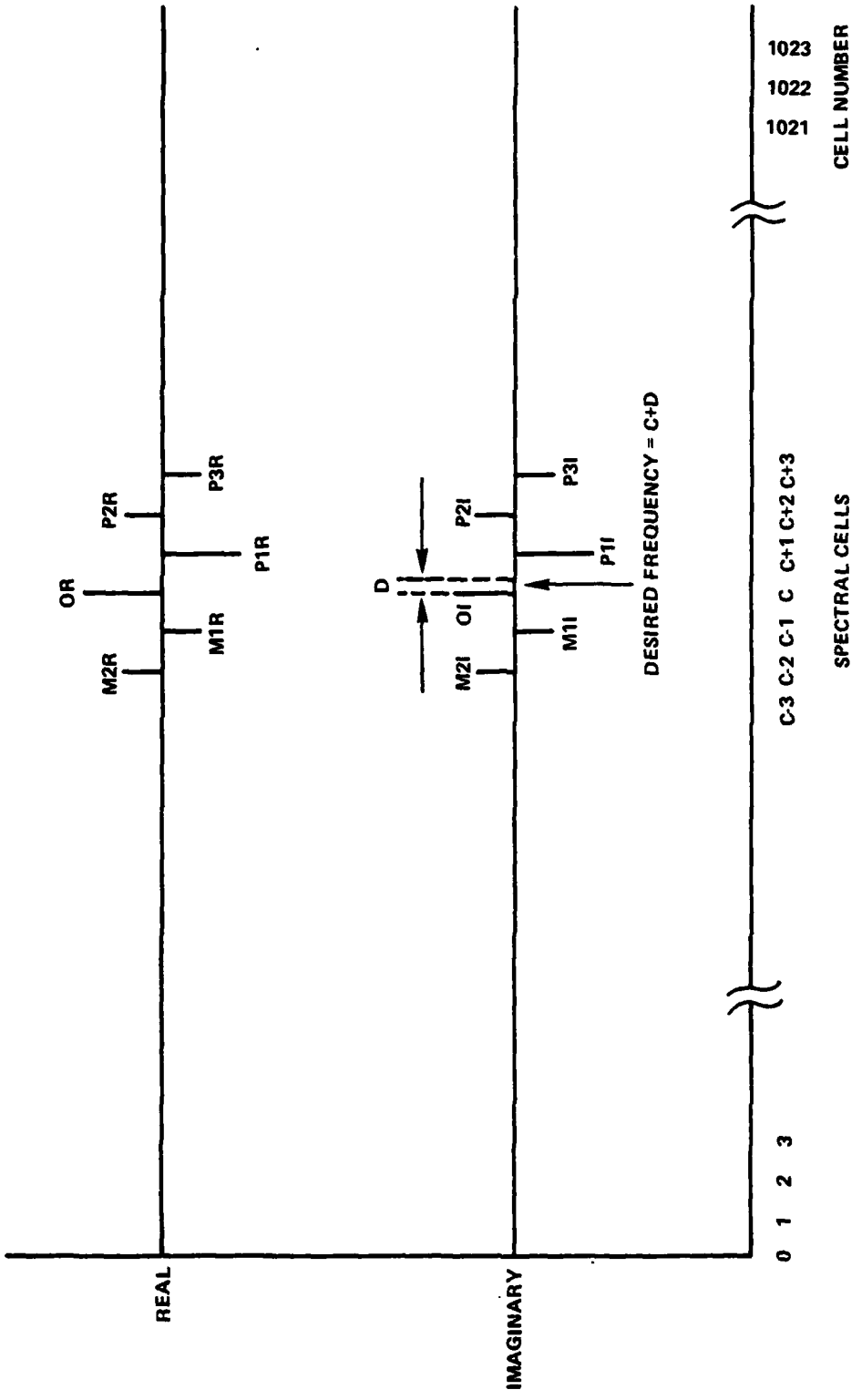


FIGURE 3-2 DSCLN ILLUSTRATION OF TERMINOLOGY

For cell c:

$$M_c = A(.5 + .5*\cos(\frac{\pi D}{2}))$$

For cell c + 1:

$$M_{c+1} = A(.5 + .5*\sin(\frac{\pi D}{2}))$$

For cell c + 2:

$$M_{c+2} = A(.5 - .5\cos(\frac{\pi D}{2}) + .052*\sin(\frac{\pi D}{2})*\cos(\frac{\pi D}{2}))$$

And for cell c + 3:

$$M_{c+3} = A(.035*\sin(\pi D)*\cos(\frac{\pi D}{2}))$$
$$= A(.07(\sin^2(\frac{\pi D}{2})*\cos(\frac{\pi D}{2}))).$$

Given D, the $\sin(\frac{\pi D}{2})$ and $\cos(\frac{\pi D}{2})$ are computed and combined to form the above functions. For a desired phase P, the component phases are set to -P, -P, P, -P, P, P.

Figure 3-3 is a BASIC generated plot comparing the three-segment approximation to the exact Hanning Function.

FUNCTIONAL DESCRIPTION

DSCLN MACRO consists of two phases of computation for each discrete line. The first phase computes the center frequency, adds the FM perturbation to obtain the instantaneous frequency, and computes the initial phase to be used for the next iteration of DSCLN. The first phase also computes the modulated amplitude to be used during the second phase computations of the current iteration. The second phase computes the six complex spectral components for each line based on the instantaneous frequency and phase stored in the Discrete Line Table (DLT) input data on the amplitude just obtained during Phase 1. The six components are then added to the appropriate cells of the spectrum being constructed.

MATHEMATICAL DESCRIPTION

For the i^{th} line during the j^{th} iteration, the following computations are performed:

Phase 1: $CDO_{i,\ j+2} = CDO_{i,\ j+1} + SL_i$

where $CDO_{i,\ j}$ is the center frequency for the j^{th} iteration and SL_i is the slew rate for the i^{th} line. Note that this computation produces the center frequency CDO_i valid for the second following iteration

$$FSM_{i,\ j+1} = FS_i * FM_{i,\ j+1}$$

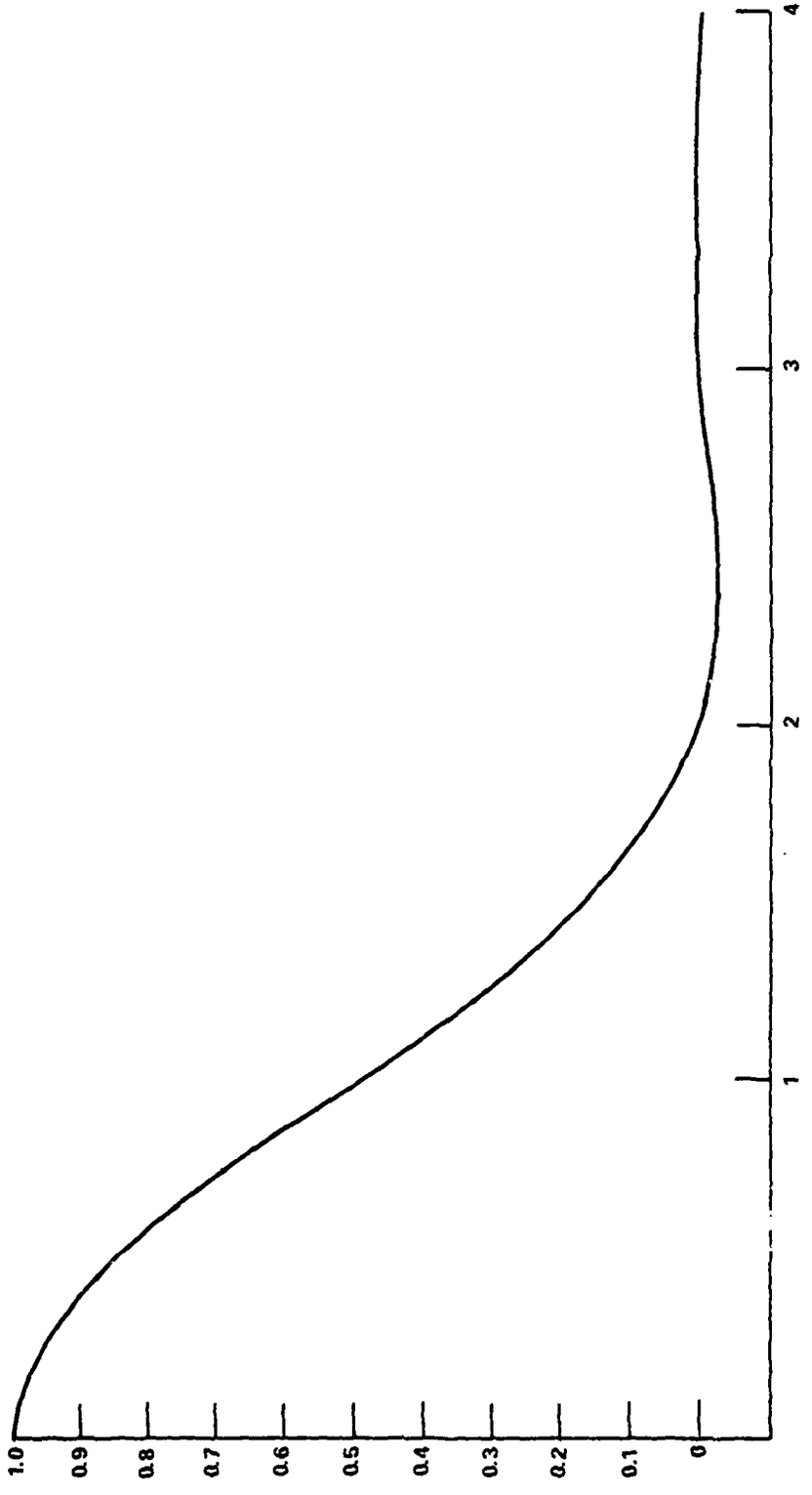


FIGURE 3-3 COMPARISON OF APPROXIMATION AND EXACT HANNING FUNCTION

where $FSM_{i,\ j+1}$ is the appropriately scaled FM perturbation for the i^{th} line during the $(j+1)^{st}$ iteration.

$$P_{i,\ j+1} = LO(P_{i,\ j} + \frac{3}{4}CD_{i,\ j} - \frac{1}{4}CD_{i,\ j+1})$$

where $CD_{i,\ j}$ is the instantaneous frequency for the j^{th} iteration.

$$A_{i,\ j} = A_i * AM_{i,\ j}$$

where A_i is the unmodulated line amplitude for the i^{th} line, and $AM_{i,\ j}$ is the amplitude modulation factor for the i^{th} line during the j^{th} iteration.

$$CODP_{i,\ j+1} = [CDO_{i,\ j+1} + FSM_{i,\ j+1}]$$

Integer part plus fraction part divided by 4 - where the integer part (H.O. 16 bits) is the center cell number C and the fractional part (L.O. 16 bits) is used as an angle argument ϕD (in BAM) in computing the Hanning coefficients.

Phase 2:

Let A = $A_{i,\ j}$

K = $\cos(2\pi * P)$

Z = $\sin(2\pi * P)$

C = $\cos(2\pi * \phi D)$

S = $\sin(2\pi * \phi D)$

7 = .07

5 = .052

BM2 (R,I) = Initial contents of cell C-2

BM1 (R,I) = Initial contents of cell C-1

BO (R,I) = Initial contents of cell C

BP1 (R,I) = Initial contents of cell C+1

BP2 (R,I) = Initial contents of cell C+2

BP3 (R,I) = Initial contents of cell C+3

M2 (R,I) = Modified contents of cell C-2

M1 (R,I) = Modified contents of cell C-1

B (R,I) = Modified contents of cell C

P1 (R,I) = Modified contents of cell C+1

P2 (R,I) = Modified contents of cell C+2

P3 (R,I) = Modified contents of cell C+3

The following computations are performed:

M2R = BM2R - 7CCSAK

M2I = BM2I - 7CCSAZ

M1R = BM1R + 1/2(ASK-AK) - 5SCAK
M1I = BM1I + 1/2(ASZ-AZ) - 5SCAZ
OR = BOR + 1/2(AK+ACK)
OI = BOI + 1/2(AZ+ACZ)
PIR = BP1R - 1/2(AK+ASK)
P1I = BP1I - 1/2(AZ+ASZ)
P2R = BP2R + 1/2(ACK-AK) + 5SCAK
P2I = BP2I + 1/2(ACZ-AZ) + 5SCAZ
P3R = BP3R + 7CSSAK
P3I = BP3I + 7CSSAZ

IMPLEMENTATION

DSCLN is implemented with a 48-instruction main loop which processes one set of Minor Frame (MNF) computations for one line. An eighteen-instruction preamble generates the parameters 7 and 5 and two auxilaries, M1 and P1, used in the main loop, and initializes the pipe for the computations for the first line. The loop count is controlled by BNZ.A, which should be set to the number of lines to be processed minus one. Since DSCLN is the last AP program to be executed under the SUPER MACRO, SM1, a halt is executed after completion.

Since intermediate results CDO, CODP and P are updated in Working Store by DSCLN, this area of WS must be returned to Bulk Store upon completion of DSCLN.

SCALING

See Table 3-1.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 3-2.

ADDRESS REGISTERS

AR0 - The initial value is set to $\cos^{-1}(7)$ = 27C8. During main loop computations AR0 contains the spectral cell addresses C-2, ..., C+3.

AR1 - The initial value is set to $\sin^{-1}(5)$ = 2809. During main loop computations AR1 contains the fraction ΦD from which the coefficients C and S are obtained.

AR2 - During main loop computations AR2 is loaded with FMA and used to access the MTR functions for FM values.

AR3 - During main loop computations AR3 is loaded with AMA and used to access the MTR functions for AM values.

AR4 - The increment code is initialized to 1. The address portion is initialized to the starting address of the SLP, CDO data area.

AR5 - The increment code is initialized to 1. The address portion is initialized to the starting address of the FAMA, FSA data area.

AR6 - The increment code is initialized to 1. The address portion is initialized to the starting address of the CODP data area.

INCREMENT REGISTERS

INCR0 - 1024 Wrap. The increment value is set to -5.

INCR1 - 1024 Wrap. The increment value is set to -2.

INCR2 - 1024 Wrap. The increment value is set to +2.

INCR3 - 1024 Wrap.

INCR4 - 1024 Wrap. The increment value is set to +2.

INCR5 - 1024 Wrap.

INCR6 - 1024 Wrap.

BRANCH REGISTERS

BNZA - Branch and Count Register controlling the number of executions of the main loop. Count and Reset fields are initialized to N-1, where N is the number of lines to be processed. The Branch Address is DSCLP.

SINE/COSINE DESTINATION REGISTERS

SCD0 - Destine Cosine only to AE0.

SCD1 - Destine Cosine/Sine to AE0.

SCD2 - No-op (00).

SCD3 - Destine Cosine/Sine to AE0.

DECIMATE REGISTER

Not Applicable.

SCALE FACTOR REGISTERS

SF0H - All True-Direct.

SF0L - Left and center prescalers - True/Right 1, right prescaler and postscaler - True/Direct.

SF1H - Left prescaler - True/Right 4, center and right prescalers - True/Right 2, postscaler True/Left 2.

SF1L - All prescalers - True/Direct, postscaler - True/Left 4.

SF2H - Left prescaler - True/Right 1, center and right prescalers - True/Direct, postscaler - True/Left 1.

SF2L - All prescalers - True/Direct, postscaler - True/Left 2.

SF3H - Left prescaler - True/Right 2, center prescaler - True/Direct, right prescaler - True/Right 1, postscaler - True/Left 1.

SF3L - Left and center prescalers - True/Right 2, right prescaler - True/Direct, postscaler - True/Left 2.

ARITHMETIC ELEMENT REGISTER MAP

See Table 3-3.

WORKING STORE MAP

See Table 2-4. (BBAMP and DSCLN are combined.)

PROGRAM CODING CHART

See Table 3-4.

TABLE 3-1 DSCLN AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
CO/DP	H0, H14		
CDO	F0		
SL	H16 (L.O.HW)		
P	H16 (L.O.HW)		
A		H12	
FM		H12	
AM		H15	
FS		H8	
7		H17	
5		H18	
BM2R, BM2I	H12		
BM1R, BM1I	H12		
BOR, BOI	H12		
BP1R, BP1I	H12		
BP2R, BP2I	H12		
BP3R, BP3I	H12		
INTERMEDIATE RESULTS:			
S, C		H14	
Z, K		H14	
7, 5		H18	
A		H12	
7C		H16	
5S		H16	
CS		H14	

TABLE 3-1 (CONTINUED)

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
CS		H14	
AK	F10	H12	
AZ	F10	H12	
5SC		H15	
AKS	F10		
AZS	F10		
AKC	F10		
AZC	F10		
7CAK		H14	
7CAZ		H14	
5SCAK	F11		
5SCAZ	F11		
7CCSAK	F12		
7CCSAZ	F12		
7SSCAK	F12		
7SSCAZ	F12		
T1, T2	F11		
T3, T4	F11		
FSM	F4		
D4	F0		
SL	F0		
PP	F0		
NCD4/NCD	F0		
CDP4	F0		
NC	F-15		
CC	F-1		

TABLE 3-1 (CONTINUED)

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
FSM4	F0		
OUTPUTS:			
M2R, M2I			H12
M1R, M1I			H12
0R, 0I			H12
P1R, P1I			H12
P2R, P2I			H12
P3R, P3I			H12

TABLE 3-2 DSCLN AEC REGISTER MAP

Incr Code Address Register

	0	3 4 15
0		27C8=Cos^{-1}(7)
1		2809=Sin^{-1}(5)
2		
3		
4	1	620=A(SL,P,CD0)
5	1	6E8=A(FAMA,FSA)
6	1	5BC=A(C0DP)
7		

WS Wrap Increment Register

	0	3 4 15
0	0	FFB=-5
1	0	FFE=-2
2	0	002
3	0	
4	0	002
5	0	0
6	0	0
7		

Memory Address Register

DSCLN

Sin/Cos Destination Register

	AE 0	AE 1	AE 2	AE 3
0	10	00		
1	11	00		
2	00	00		
3	11	00		

Decimate Register

Count	Reset

Comments

N = # of lines.

BR/BCR Unconditional/ Conditional Branch

Branch Address

	0 15
0	
1	
2	
3	
4	
5	
6	
7	

BNZ Branch and Count Registers

	0 7	8 15	16 21	22 31
8			X	
9			X	
A	N-1	N-1	X	DSCLP
B			X	
C			X	
D			X	
E			X	
F			X	

TABLE 3-3 DSCLN AE REGISTER MAP

MLR

ØH	FS	A	ØL
1H	P1	5SC,NCD	1L
2H	7CAK,7C	7CAZ,5S	2L
3H	7,C,P	5,S,SL	3L

MRR

ØH	C,FM	S,FM	ØL
1H	K,AM	Z,AM	1L
2H	AK,7	AZ,5	2L
3H	CS	SS,P1	3L

ALIR

ØH	BM2, BP2, NC		ØL
1H	BM1	BP3	1L
2H	BØ	FSM	2L
3H	BP1	PP	3L

ACIR

ØH	7CCSAK, CDØ, TEMP		ØL
1H	7CCSAZ, CØDP, TEMP		1L
2H	5SCAK, 7SSCAK, TEMP		2L
3H	5SCAZ, 7SSCAZ, P		3L

ARIR

Ø	ASK
1	ASZ
2	ACK
3	ACZ

AEOR

ØH	M2R,P2R	M2I,P2I	ØL
1H	M1R,P3R,CØ	M1I,P3I,DØ	1L
2H	ØR,C	ØI,D4	2L
3H	P1R,SL	P1I,P	3L

SCALE FACTOR REGISTERS

AEØ

ØH	Ø	Ø	Ø	Ø	1	1	Ø	Ø	ØL
1H	3	2	2	2	Ø	Ø	Ø	3	1L
2H	1	Ø	Ø	1	Ø	Ø	Ø	2	2L
3H	2	Ø	1	1	2	2	Ø	2	3L

AE1

ØH									ØL
1H									1L
2H									2L
3H									3L

ALOR

Ø	CØDP
1	P1
2	7
3	CDØ

ACOR

Ø	AK,CD
1	AZ
2	5
3	P1,LM

AROR

Ø	T1,D4,M1
1	T2
2	T3,CD4
3	T4,CC

TABLE 3-4 DSCLN PROGRAM CODING CHART

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD		ALOR	ACOR	AROR	AEOR
	DSCLN		+0 AR5												
		CS(1)	READ FAMA							LOG = 0	L				
			+1 AR5							LOG = 0	L			0 M1	
			READ FS,A		0H M1										
			+0 AR6	1H M1											
		C(0)	READ CODP	0 FS,A	2H DUMMY	P1 = M1*M1									
			+0 AR4												
			READ SL,P		2L 5		0 P1	1 CODP							
			+0 AR2							CODP	OH				
			READ FM	3 SL,P	2H 7					P1 = LO(P1)	OH				
			+1 AR4							P1 = LO(P1)	OH	0 CODP	3 P1		
			READ CDO	1H P1	0 FM					LM = M1 + P1(R2)	3L				
					3L P1	7 = 7*P1						1 P1	3 LM		
						5 = 5*P1		0 CDO		D4 = CODP∧LM	L				
			+0 AR3			FSM = FS*FM		2 7		CDO	OH			0 D4	
			READ AM			SL = P1*SL		3 5		7 = LO(7)(L1)	2H				
							2 FSM			5 = LO(5)	OH	3 CDO			
					1 AM			2 SL		FSM4-FSM(R4)	1H	2 7	5		
														1 FSM4	
						PP = P*P1				NCD4=(CDO+FSM4)(R2)	1H				

TABLE 3-4 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	DSCLP		AR6 +0							CDO = CDO + SL OH			NDC4 2	
			READ CD	NCD 1L			PP 3			CDP4=H(CODP)+D4 3L				
			AR4 +0			NC = P1*NCD				OD=NCD4∆LM L		CDP4 0		CDO 1
			WRITE CDO							CC=PP-CDP4-D4 OL				
			AR4 -1				NC 0			C=CDO+FSM OH		CC 1		OD 2
		CS(1)	READ SL,P							P= $\frac{L(NC)}{2}$ +CC $\frac{NCD}{4}$ 3H				
			AR6 +0							SL=LO(SL) OH				C 2H
			WRITE CD							7 OH				P 3
			AR4 +0			A = A*AM				5 OH				SL 3H
		CS(3)	WRITE SL,P	7 3H	C OH									
			AR0 -2	5 3L		7C = 7*C		A 0						
			READ BM2		S OL	C = P1*C				A 1L				
			AR0 +1			S = P1*S		7C 0						
			READ BM1	A 0L	K 1H	5S = 5*S	BM2 0	C 1		7C OH R				
			AR0 +1			AK = A*K		S 0		C = LO(C) OH				
			READ B0	7C 2H	Z 1L		BM1 1	5S 1		S = LO(S) OH R				
			AR0 +1	C 3H		AZ = A*Z		AK 0		5S OH				
			READ BP1	S 3L			B0 2			AK 2L				
			AR5 +2	5S 2L		CS = C*S		AZ 0		R		AK 0		
			READ FAMA		AK 2H	5SC = 5S*C	BP1 3			AZ 2L R				
			AR5 +1			7CAK = 7C*AK		CS 0				AZ 1		
			READ FSA		AZ 2L	SS = S*S		5SC 1		CS 2L R				

TABLE 3-4 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
			+0 AR2			7CAZ=7C*AZ		0 7CAK		5SC OH				
			READ FM	0 FS,A	3H CS	AKS=AK*S		1 SS		7CAK 2L R				
			+0 AR3	1L 5SC		AZS=AZ*S		0 7CAZ		SS 2L R				
			READ AM	2H 7CAK	0 FM	AKC=AK*C			0 ASK	7CAZ 2L R				
			+2 AR4		3L SS	7CCSAK=7CAK*CS			1 ASZ	T1=(ASK-AK)/2 2H R				
			READ SL,P	2L 7CAZ	1 AM	AZC=AZ*C			2 ACK	T2=(ASZ-AZ)/2 2H R			0 T1	
			+1 AR4			7CCSAZ=7CAZ*CS		0 7CCSAK		T3=(AK-ACK)/2 2H			1 T2	
			READ CDO	3 SL,P					3 ACZ	M2R=BM2R-7CCSAK OH			2 T3	
			+1 AR6			5SCAK=5SC*AK		1 7CCSAZ		T4=(AZ-ACZ)/2 2H				
			READ CD					0 CDO		M2I=BM2I-7CCSAZ OH R			3 T4	0H M2R
			+1 ARO			5SCAZ=5SC*AZ		2 5SCAK		CDO OH				
			READ BP2					1 CD		M1R=BM1R-5SCAK+T1 2H R				0L M2I
			+1 ARO					3 5SCAZ		CD OH	3 CDO			
			READ BP3				0 BP2			M1I=BM1I-5SCAZ+T2 OH R				1H M1R
			-5 ARO							OR=BOR+AK/2+ACK/2 2H	0 CODP			
			WRITE M2				1 BP3			OI=BOI+AZ/2+ACZ/2 2H R				1L M1I
			+1 ARO							P1R= BP1R-AK/2-ASK/2 2H R				2H OR
			WRITE M1							P1I BP1I-AZ/2-ASZ/2 2H R				2L OI
			+1 ARO			7SSCAK=7CAK*SS				P2R= BP2R+5SCAK+T3 2H R				3H P1R
			WRITE 0			7SSCAZ=7CAZ*SS				P2I= BP2I+5SCAZ+T4 2H R				3L P1I
			+1 ARO			FSM=FS*FM		2 7SSCAK		P1 OH R				0H P2R
			WRITE P1					3 7SSCAZ		P3R= BP3R+7SSCAK OH R				0L P2I

TABLE 3-4 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
			+1 ARO		3L P1		2 FSM			P3I= BP3I+7SSCAZ 0H				
	BNZA		WRITE P2			SL=P1*SL				FSM4=FSM (R4) 1H R				P3R 1H
			+1 ARO							D4=CODP/LM L R			FSM4 1	P3I 1L
			WRITE P3			PP=P1*P		2 SL		NCD4=CDO+FSM4 1H			0 D4	
						NOP								
						STOP								

CHAPTER 4

SINGLE REAL FFT (SRFFT) MACRO

FUNCTIONAL DESCRIPTION

The SRFFT MACRO performs the complex multiplications required to allow the FFT to effect the transformation from real time series to single-sided spectra. SRFFT is implemented as a separate pass which must be executed on a single-sided spectrum before bit reversal, or as a last step after a forward transform and bit reversal to produce a single-sided spectrum. For N = 1024, 2048 time samples are produced from 1024 spectral samples or vice-versa, and the six passes (1 SRFFT, 5 FFT4) require 2.46 ms.

The computation involves pairs of complex samples from symmetrical locations in the data array, i. e., outputs $F_n + F_{N-n}$ are derived from inputs G_n and G_{N-n}. The computation requires the exponential $e^{\frac{j2\pi n}{2N}}$, which is generated by the Sin/Cos Generator.

MATHEMATICAL DESCRIPTION

The input array is denoted by G_n, n=0, ..., N-1. The output array is denoted by F_n, n=0, ..., N-1. The computations are broken into the following steps, where GR_i, GI_i, FR_i, FI_i, etc., denote the real and imaginary parts of $G_i + F_i$, respectively, and C_i and S_i the real and imaginary parts of the complex exponential $\exp(j2\pi \frac{i}{2N})$:

$$CR_i = (GR_{n-i} - GR_i)/2$$
$$CI_i = (GI_{N-i} - GI_i)/2$$
$$SCR_i = S_i * CR_i$$
$$CCR_i = C_i * CR_i$$
$$CCI_i = C_i * CI_i$$
$$SCI_i = S_i * CI_i$$
$$AR_i = (GR_{N-i} + GR_i)/2$$
$$AI_i = (GI_{N-i} + GI_i)/2$$

$$FR_i = (SCR_i + CCI_i + AR_i/4)*4$$
$$FI_i = (SCI_i - CCR_i + AI_i/4)*4$$
$$FR_{N-i} = (-SCR_i - CCI_i + AR_i/4)*4$$
$$FI_{N-i} = (SCI_i - CCR_i - AI_i/4)*4$$

IMPLEMENTATION

The SRFFT MACRO consists of a sixteen-instruction prologue and an eight-instruction main loop. During the prologue, two minus-one's are created and placed in MLR.2H and MRR.1H to allow data to be recycled through the multiplier with only a sign change. The (N-i) output address counter AR.5 is artificially incremented by one to allow the first output to be effected in the main loop with a decrement associated with it. Also unique to the prologue is the computation for F_0, which for a forward transform is FR0 = GR0/2 + GI0/2, and for an inverse transform is $FR_0 = GR_0$, $FI_0 = GR_0$. These are selectable by SF2H and SF2L.

Scaling is applied in the computation of AR, AI, CR, and CI (SF0L) to effect the overall factor of 1/2 that is required from input to output. In addition, SF1H is used to compensate for the effective scaling index difference of two that is introduced when CR and CI are multiplied by C and S. If an additional net scaling is desired, it may be effected by reducing the number of postscaler shifts in SF1H.

The main loop count control is based on BNZ8, with an outside loop counter BNZ9 to permit operation on arrays larger than 512 points.

SCALING

See Table 4-1.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 4-2.

ADDRESS REGISTERS

AR0 - The initial value is set to zero, including increment code.

AR2 - The increment code is set to 1. The address portion is initialized to the starting address of the Input Array.

AR3 - The increment code is set to 1. The address portion is initialized to the last address of the Input Array.

AR4 - The increment code is set to 1. The address portion is initialized to the starting address of the Output Array.

AR5 - The increment code is set to 1. The address portion is initialized to the last address of the Output Array.

INCREMENT REGISTERS

INC0 - Set to 1/2N as BAM angle increment.

INC2, 3, 4, 5 - Wrap code must be set to 1024.

BRANCH REGISTERS

BNZB - Branch and Count Register controlling the number of executions of the inner loop. Count and Reset fields are initialized to N/4-2 and N/4-1, respectively, where N is the array size. The Branch Address is SRFLP.

BNZC - Branch and Count Register controlling the number of executions of the outer loop. Count and Reset fields are initialized to 1. The Branch Address is SRFLP.

SINE/COSINE DESTINATION REGISTERS

SCD0 - Destine Sine/Cosine to AE0 and AE1.

DECIMATE REGISTERS

Not applicable.

SCALE FACTOR REGISTERS

SF0H - All True/Direct.

SF0L - Left, center and right prescalers - True/Right 1, postscaler - True/Direct.

SF1H - Left prescaler - True/Direct, center prescaler - True/Right 2, right prescaler - True/Direct, postscaler - True/Left 2.

SF2H - Forward Transform: Left and center prescalers - True/Right 1, right prescaler and postscaler - True/Direct.
Inverse Transform: Left and center prescalers - Inhibit, right prescaler and postscaler - True/Direct.

SF2L - Forward Transform: Left prescaler - True/Direct, center and right prescalers - Inhibit, postscaler - True/Direct.
Inverse Transform: Left prescaler - True/Direct, center and right prescalers - Inhibit, postscaler - True/Direct.

ARITHMETIC ELEMENT REGISTER MAP

See Table 4-3.

WORKING STORE MAP

See Table 4-4.

PROGRAM CODING CHART

See Table 4-5.

TABLE 4-1 SRFFT AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
GR, GI	H12	H12	
GRN, GIN	H12	H12	
C, S		H14	
INTERMEDIATE RESULTS:			
MGRN, MGIN	H12 (L.O.HW)		
MGR, MGI	H12 (L.O.HW)		
CR, CI		H12	
AR, AI	F12		
CCR, CCI	F10		
SCR, SCI	F10		
OUTPUTS:			
FR, FI			H12
FRN, FIN			H12

TABLE 4-2 SRFFT AEC REGISTER MAP

Incr Code Address Register

	0	3 4 15
0	Ø	ØØØ
1		
2	1	A(INPUT) = I
3	1	A(EOI) = I+N-1
4	1	A(OUTPUT) = O
5	1	A(EOO) = O+N-1
6		
7		

WS Wrap Increment Register

	0	3 4 15
0		$\frac{1}{2N}$
1		
2	Ø	
3	Ø	
4	Ø	
5	Ø	
6		
7		

Memory Address Register

SRFFT

Sin/Cos Destination Register

	AE 0	AE 1	AE 2	AE 3
0	1 1	1 1		
1	Ø Ø	Ø Ø		
2				
3				

Decimate Register

Count	Reset

Comments

N = Number of Complex Points

EOI = End of Input
EOO = End of Output

BR/BCR Unconditional/ Conditional Branch

Branch Address

	0 15
0	
1	
2	
3	
4	
5	
6	
7	

BNZ Branch and Count Registers

	0 7	8 15	16 21	22 31
8				
9				
A				
B	$\frac{N}{4}$ - 2	$\frac{N}{4}$ - 1		SRFLP
C	1	1		SRFLP
D				
E				
F				

Count Reset Spare Branch Address

TABLE 4-3 SRFFT AE REGISTER MAP

MLR

ØH	GR	GI	ØL
1H	CR	CI	1L
2H	M1		2L
3H	DUMMY READ		3L

MRR

ØH	C	S	ØL
1H	M1		1L
2H	GRN	GIN	2L
3H			3L

ALIR

ØH	GR	GI	ØL
1H	MGR		1L
2H	SCR/MGI		2L
3H	SC	I	3L

ACIR

ØH	GRN	GIN	ØL
1H	MGRN		1L
2H	CCR/MGIN		2L
3H	CC	I	3L

ARIR

Ø	
1	
2	
3	

AEOR

ØH	FR	FI	ØL
1H	FRN	FIN	1L
2H			2L
3H			3L

SCALE FACTOR REGISTERS

AEØ

ØH	Ø Ø Ø Ø	1 1 1 Ø	ØL
1H	Ø 2 Ø 2		1L
Forward / Inverse 2H	14 14 Ø Ø / Ø Ø Ø Ø	4 4 Ø	2L
3H			3L

AE1

ØH	Ø Ø Ø Ø	1 1 1 Ø	ØL
1H	Ø 2 Ø 2		1L
2H	14 14 Ø Ø / Ø Ø Ø Ø	4 4 Ø	2L
3H			3L

ALOR

Ø	
1	
2	
3	

ACOR

Ø	AR
1	AI
2	
3	

AROR

Ø	GIØ
1	
2	
3	

TABLE 4-4 SRFFT WORKING STORE MAP

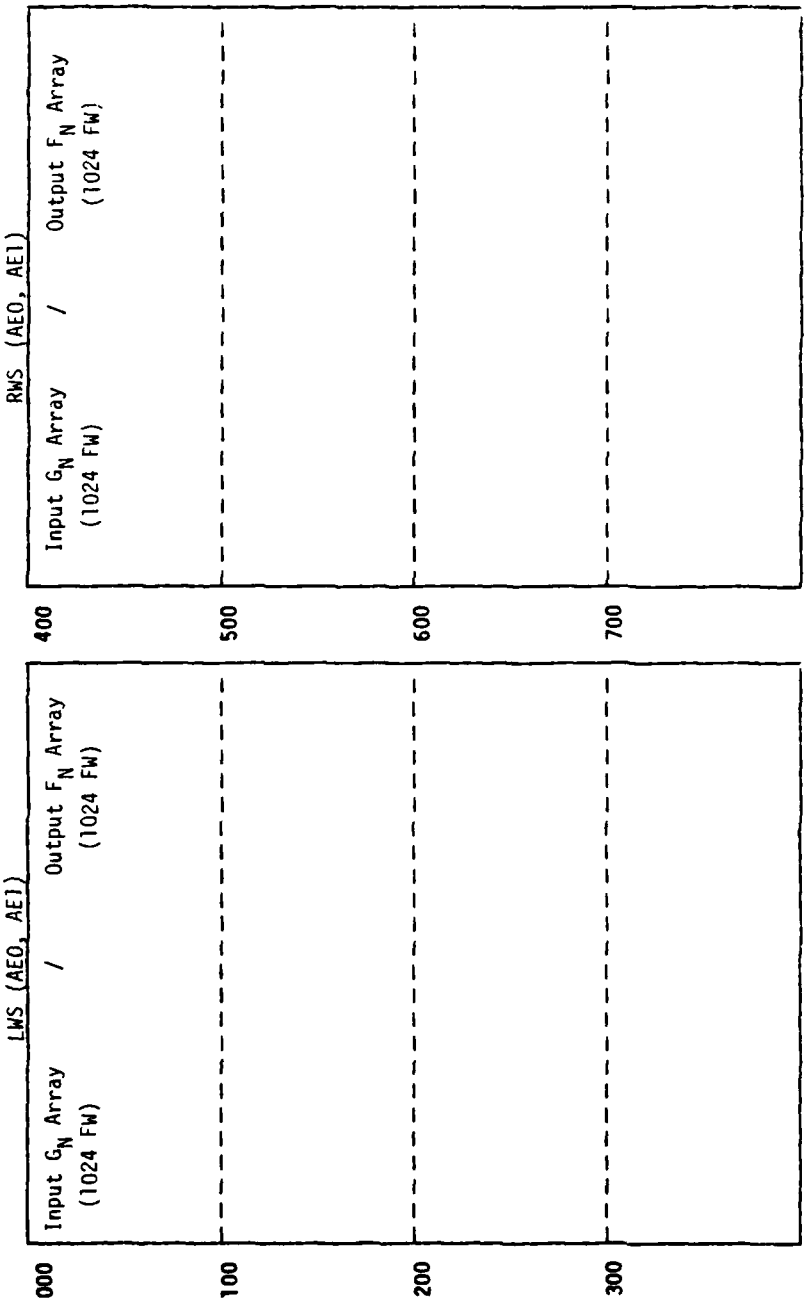


TABLE 4-5 SRFFT PROGRAM CODING CHART

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD		ALOR	ACOR	AROR	AEOR
	SRFFT		+Ø AR2							LOG = 0	L				
			READ GØ												
			+1 AR2	2H M1						LOG = 0	L				
		CS(1)	READ G1				1 GØ								
			+Ø AR3		1H M1					GIØ=L(GØ)	ØH				
			READ GN1	Ø G1			2 G1			FIØ = GRØ	2L			Ø GIØ	
			+1 AR5							FRØ= GRØ or (GRØ+GIØ)/2	2H				
			READ DUMMY		ØH C			Ø GN1							ØL FIØ
			+1 AR2							CI1 =(GIN1+GI1)/2	ØL				ØH FRØ
			READ G2	3 DUMMY	ØL S					AR1=(GRN1+GR1)/2	ØL				
			-1 AR3	1L CI						CR1=(GRN1-GR1)/2	ØL R			Ø AR1	
			READ GN2	Ø G2		CCI1 = C*CI1	Ø G2			AI1=(GIN1-GI1)/2	ØL				
			+0 AR4	1H CR		SCI1 = S*CI1					R			1 AI1	
			WRITE FØ		2 GN2	SCR1 = S*CR1		Ø GN2	3 CCI		ØL				
		CS(2)				CCR1 = C*CR1	3 SCI1			CR2=(GRN2-GR2)/2	ØL				
						MGR2 = M*GR2	2 SCR1			CI2=(GIN2+GI2)/2	ØL				

TABLE 4-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
			+1	1H					2	1H				
	SRFLP		AR2	CR		MGRN=MI*GRN			CCR	FR=SCR+CCI+AR R				
			READ	1L	ØH		1			1H				
			G	CI	C	MGIN=MI*GIN	MGR			FI=SCI-CCR+AI R				
			-1					1		1H				ØH
			AR3			MGI=MI*GI		MGRN		FRN=-SCR-CCI+AR R				FR
			READ	Ø	ØL		Ø	2		1H				ØL
			GN		S	CCI=C*CI	G	MGIN		FIN=SCI-CCR-AI R				FI
			+1				2			ØL				1H
			AR4			SCI=S*CI	MGI			AR=-MGR-MGRN R				FRN
			WRITE		2			Ø	3	ØL		Ø		1L
	BNZB	CS	F		GN	SCR=S*CR		GN	CCI	AI=MGI-MGIN R		AR		FIN
			-1				3			ØL		1		
			AR5			CCR=C*CR	SCI			CR=GRN-GR		AI		
			WRITE				2			ØL				
			FN			MGR=MI*GR	SCR			CI=GIN+GI				
						NOP								
	BNZC													
						CCR=C*CR								
										ØL				
						MGR=MI*GR				CI=GIN+GI				
						NOP								
						STOP								

CHAPTER 5

SCALE AND ADD TWO ARRAYS (SCLA2) MACRO

FUNCTIONAL DESCRIPTION

SCLA2 scales an array X by a constant A, scales an array Y by a constant B, and combines the two scaled arrays on an element-by-element basis to form an output array Z. X, Y, and Z must be of the same number of elements N. The scaling coefficients A and B are obtained via the Sin/Cos Generator in order to avoid the requirement for two words of Working Store (WS). These must therefore be loaded from the corresponding CFCB as the ARCCOS (A) and the ARCCOS (B), respectively.

MATHEMATICAL DESCRIPTION

$$Z_i = A * X_i + B * Y_i; \quad i = 1, 2, \ldots, N$$

where A and B are two scalers, and X_i, Y_i and Z_i are the i^{th} elements of arrays X, Y and Z, respectively.

IMPLEMENTATION

SCLA2 is implemented with a six-instruction main loop which processes one pair of X's and one pair of Y's to form one pair of Z's. A twelve-instruction preamble obtains the coefficients A and B via the Sin/Cos Generator and loads the pipe with the first pairs of X's and Y's. Each pass through the main loop then stores one pair of Z's in the output array. Since the maximum count is 255, or 256 passes through the loop, an outside loop is required to permit array sizes N greater than 512 elements. The inside loop is controlled by BNZ8 and the outside loop is controlled by BNZ9.

SCALING

See Table 5-1.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 5-2.

ADDRESS REGISTERS

AR0 - ARCCOS(A), i. e., AR0 is set to an angle α such that $\cos(\alpha) = A$.

AR1 - ARCCOS(B).

AR2 - The increment code is initialized to one. The address is initialized to the first address of the X array.

AR3 - The increment code is initialized to one. The address is initialized to the first address of the Y array.

AR4 - The increment code is initialized to one. The address is initialized to the first address of the Z array.

INCREMENT REGISTERS

Not applicable. Wrap codes must be 1024 for INC 2, 3, 4.

BRANCH REGISTERS

BNZ8 - Branch and Count register controlling the inner loop. Reset and Count fields are initialized to N/M-1, where N is the number of elements in each of the arrays, and M is the number of times the outside loop is executed. Branch Address is SC2LP.

BNZ9 - Branch and Count register controlling the number of executions of the outer loop. Reset and Count fields are initialized to M-1. Branch Address is SC2LP.

SINE/COSINE DESTINATION REGISTERS

SCD0 - Destine Cosine only to both AE0 and AE1.

SCD1 - Destine Cosine only to both AE0 and AE1.

DECIMATE REGISTER

Not applicable.

SCALE FACTOR REGISTERS

SF0H - Left, center and right prescalers - True/Direct. Postscaler - True/Left 2.

ARITHMETIC ELEMENT REGISTER MAP

See Table 5-3.

WORKING STORE MAP

See Table 5-4.

PROGRAM CODING CHART

See Table 5-5.

TABLE 5-1 SCLA2 AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
A, B		H14	
XH, XL		H12	
YH, YL		H12	
INTERMEDIATE RESULTS:			
AXH, AXL	F10		
BYH, BYL	F10		
OUTPUTS:			
ZH, ZL			H12

TABLE 5-2 SCLA2 AEC REGISTER MAP

	Incr Code	Address Register	WS Wrap	Increment Register
	0 3 4 15		0 3 4 15	
0	Ø	ØØØ	Ø	
1	Ø	ØØØ	Ø	
2	1	A(X)	Ø	
3	1	A(Y)	Ø	
4	1	A(Z)	Ø	
5				
6				
7				

Memory Address Register

SCLA2

Sin/Cos Destination Register

	AE 0	AE 1	AE 2	AE 3
0	1,0	1,0		
1	1,0	1,0		
2				
3				

Decimate Register

Count	Reset

Comments

N = # of data points to be processed.

BR/BCR Unconditional/ Conditional Branch

	Branch Address
	0 15
0	
1	
2	
3	
4	
5	
6	
7	

BNZ Branch and Count Registers

	Count	Reset	Spare	Branch Address
	0 7 8 15	16 21	22 31	
8	$N_{/4-1}$	$N_{/4-1}$		SC2LP
9	1	1		SC2LP
A				
B				
C				
D				
E				
F				

TABLE 5-3 SCLA2 AE REGISTER MAP

MLR

ØH	XH	XL	ØL
1H	YH	YL	1L
2H	DUMMY		2L
3H			3L

MRR

ØH	A	B	ØL
1H			1L
2H			2L
3H			3L

ALIR

ØH	AX H	ØL
1H	AX L	1L
2H		2L
3H		3L

ACIR

ØH	BX H	ØL
1H	BX L	1L
2H		2L
3H		3L

ARIR

Ø	
1	
2	
3	

AEOR

ØH	ZH	ZL	ØL
1H			1L
2H			2L
3H			3L

SCALE FACTOR REGISTERS

AEØ

ØH	Ø Ø Ø 2		ØL
1H			1L
2H			2L
3H			3L

AE1

ØH	Ø Ø Ø 2		ØL
1H			1L
2H			2L
3H			3L

ALOR

Ø	
1	
2	
3	

ACOR

Ø	
1	
2	
3	

AROR

Ø	
1	
2	
3	

TABLE 5-4 SCLA2 WORKING STORE MAP

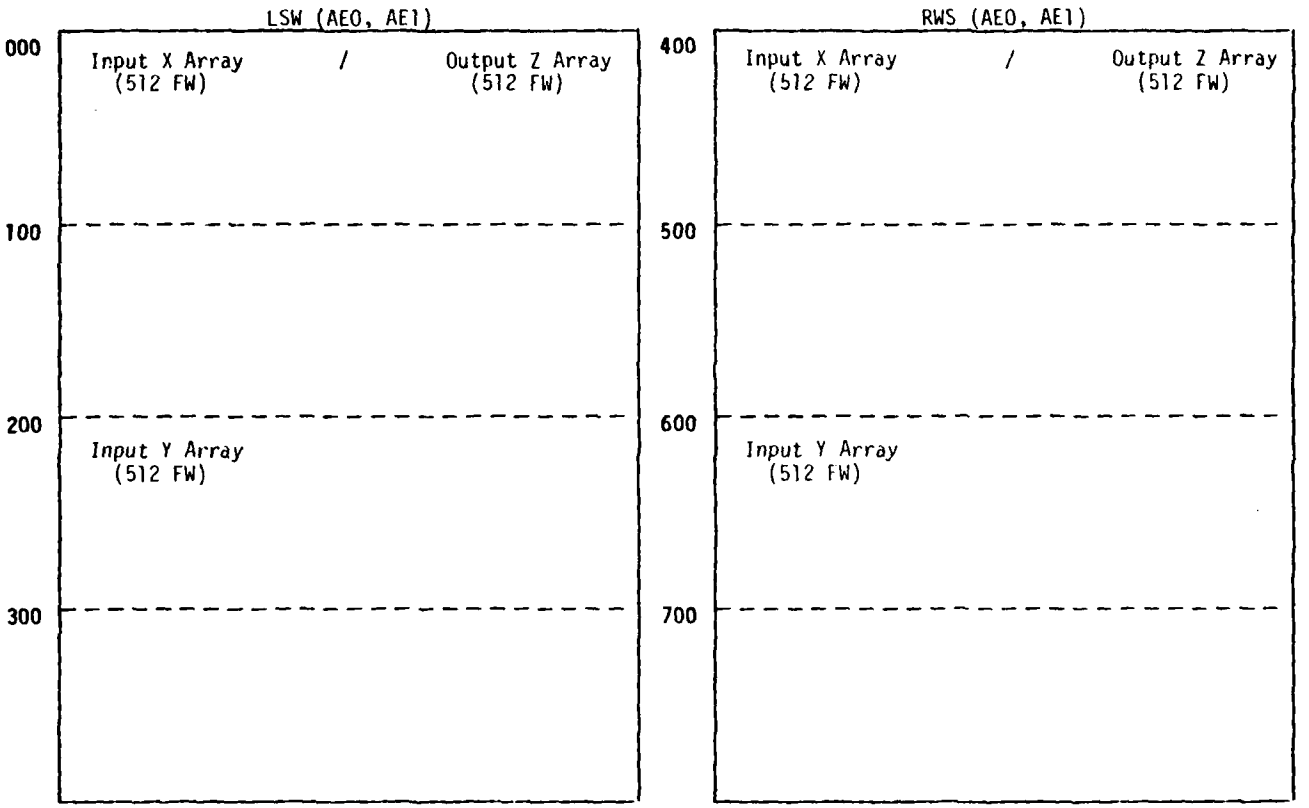


TABLE 5-5 SCLA2 PROGRAM CODING CHART

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	SCLA2		AR2 +Ø			NOP								
		C(Ø)	READ X											
			AR3 +Ø			NOP								
		C(1)	READ Y	X Ø										
						NOP								
				Y 1	A ØH	NOP								
			AR4 -1			AXH = A*XH								
			READ Z		B ØL	AXL = A*XL								
			AR2 +1			BYH = B*YH	AXH Ø							
			READ X			BYL = B*YL	AXL 1							
			AR3 +1					BYH Ø						
			READ Y	X Ø				BYL 1						
						AXH = A*XH				ZH = AXH+BYH ØH				
				Y 1		AXL = A*XL				ZL = AXL+BYL ØH				
	SC2LP		AR2 +1			BYH = B*YH	AXH Ø							ZH ØH
			READ X			BYL = B*YL	AXL 1							ZL ØL
			AR3 +1					BYH Ø						
	BNZ8		READ Y	X Ø				BYL 1						
			AR4 +1			AXH = A*XH				ZH = AXH+BYH ØH				
			WRITE Z	Y 1		AXL = A*XL				ZL = AXL+BYL ØH				

TABLE 5-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD		ALOR	ACOR	AROR	AEOR	
															ZH	ØH
	BNZ9															
						AXH=A*XH										
						AXL=A*XL				ZL=AXL+BYL	ØH					
						NOP										
						STOP										

CHAPTER 6

ASYNCHRONOUS SAMPLE, SCALE AND SUM (ASSSS) MACRO

FUNCTIONAL DESCRIPTION

The ASSSS MACRO performs the dopplered resampling of the input signal D for each multipath arrival at the sensor. Linear interpolation is employed to provide some protection against aliasing, and is equivalent to passing the signal through a filter with a frequency response of:

$$H(f) = \frac{\sin^2(\frac{\pi f}{f_s})}{(\frac{\pi f}{f_s})^2}$$

before resampling. The parameter f_s is the sample rate of the input or source signal. The resampled signal is multiplied by a gain parameter A and added to a Sum array S. Since there is not a one-to-one correspondence between the samples of the input signal D and the output, to add N new samples to the Sum array may require more or less than N samples of D. The resampling may start at an arbitrary phase (or fractional position F, $0 \leq F < 1$) with respect to the input samples, but the MACRO is intended for application only where the resample rate is relatively close to the original rate, say $\pm$ 10% maximum.

MATHEMATICAL DESCRIPTION

Phase 1: Apply amplitude factor to source input samples

```
AH = A * DH
AL = A * DL
```

where A is the amplitude factor and DH and DL are the input samples in the high and low halves of the Working Store (WS) word.

Phase 2: Form interpolation fractions for next pair of outputs.

```
F1 = F2 + D + 0C
F3 = -F2 - D + 1C
F2 = F1 + D + 0C
F4 = -F1 - D + 1C
```

where D is the resample rate parameter, F1 and F2 are the resample fractions, F3 and F4 are the complements of F1 and F2, respectively, with respect to unity, and 0C and 1C are conditional "ones" that are used to constrain the F's to interval 0-1.

Phase 3: Form the products of the scaled inputs with the interpolation fractions.

1AH = F1 * AH

2AL = F2 * AL

3AL = F3 * AL

4AH = F4 * AH

Phase 4: Combine outputs of Phase 3 with sample values from Sum array to form final results.

For Inphase loop:

OH = SH + 3AL + 1AH

OL = SL + 4AH + 2AL

For Outphase loop:

OH = SH + 4AH + 2AL

OL = SL + 3AL + 1AH

IMPLEMENTATION

The ASSSS MACRO consists of two main loops, each eight instructions long, eight transition sequences for the eight possible ways that transitions between the two major loops can occur, plus some initialization and finalization code. If the block size is 1024 samples (512 words), both the input array and the output array may exceed 512 words, and the MACRO must operate from one half of Working Store to the other half. Smaller block sizes may be handled within one side of WS.

The two main loops are designated the "Inphase" loop and the "Outphase" loop, depending on how the input array and output array are related. Actually, a more reasonable nomenclature might be "leading" and "lagging" loops, since for the "inphase" or "leading" loop, the two samples in an output data word are derived from the two samples in the input data word plus one contribution from the previous input data word. For the "outphase" or "lagging" loop, the output data samples are derived from the two samples in the input data word plus one contribution from the following input data word.

Since the transitions will occur at rates less than one in twenty or more times through the main loops, the timing is nominally determined by the eight steps per word for each of the main loops. For nominally 512 executions of one or the other of the main loops (1024 samples), the approximate execution time for the ASSSS MACRO is .41 ms.

The operation of the two main loops is illustrated in Figures 6-1 and 6-2. Each figure follows the two samples in one input word through the pipeline to the three output samples that they affect. Since the outputs contain terms that involve triple products of an interpolation fraction, an amplitude and the input data sample, two full trips through the pipeline are required, beginning with a read of the input word in the first trip through the loop and ending with the write of the last affected output in the fourth following pass through the loop. On the first pass through the pipeline, the input data samples are multiplied by the amplitude factor. Approximately concurrent with these first products being passed through the adder, the four appropriate interpolation fractions are generated and passed to the multiplier. On the second pass through the pipeline, the scaled samples are multiplied by the interpolation fractions and these triple products are combined with the corresponding Sum array samples to form the final composite output.

At the time the interpolation fractions F_i are formed, a test is made to determine whether they have progressed out of the range $0 \leq F_i < 1$. Depending on which of the two fractions F1 or F2 have exceeded the valid range, and whether the increment D is positive or negative, the program deviates through one of eight transition routines to resume operation in the alternate (inphase or outphase) loop. These transition routines re-adjust the appropriate interpolation fractions to the valid range and compute the one or two output values that do not conform to the procedures in the main loops. The following gives a brief description of each of the transition sequences.

PI001 In to Out On F1, D positive

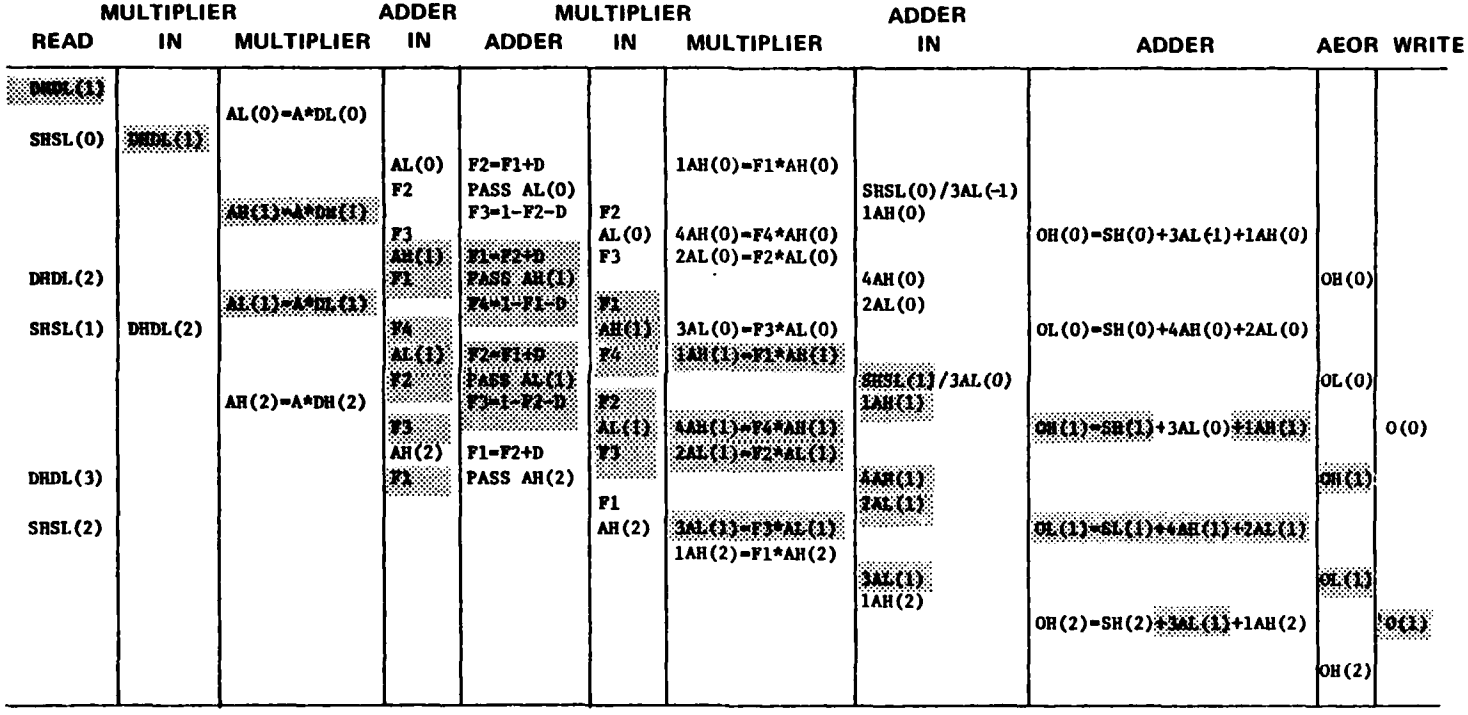
At the time the branch is taken, the following quantities have been erroneously computed:

F1 = -1 + e F3 = 1 - e - 2D

F2 = e + D F4 = 1 - e - D

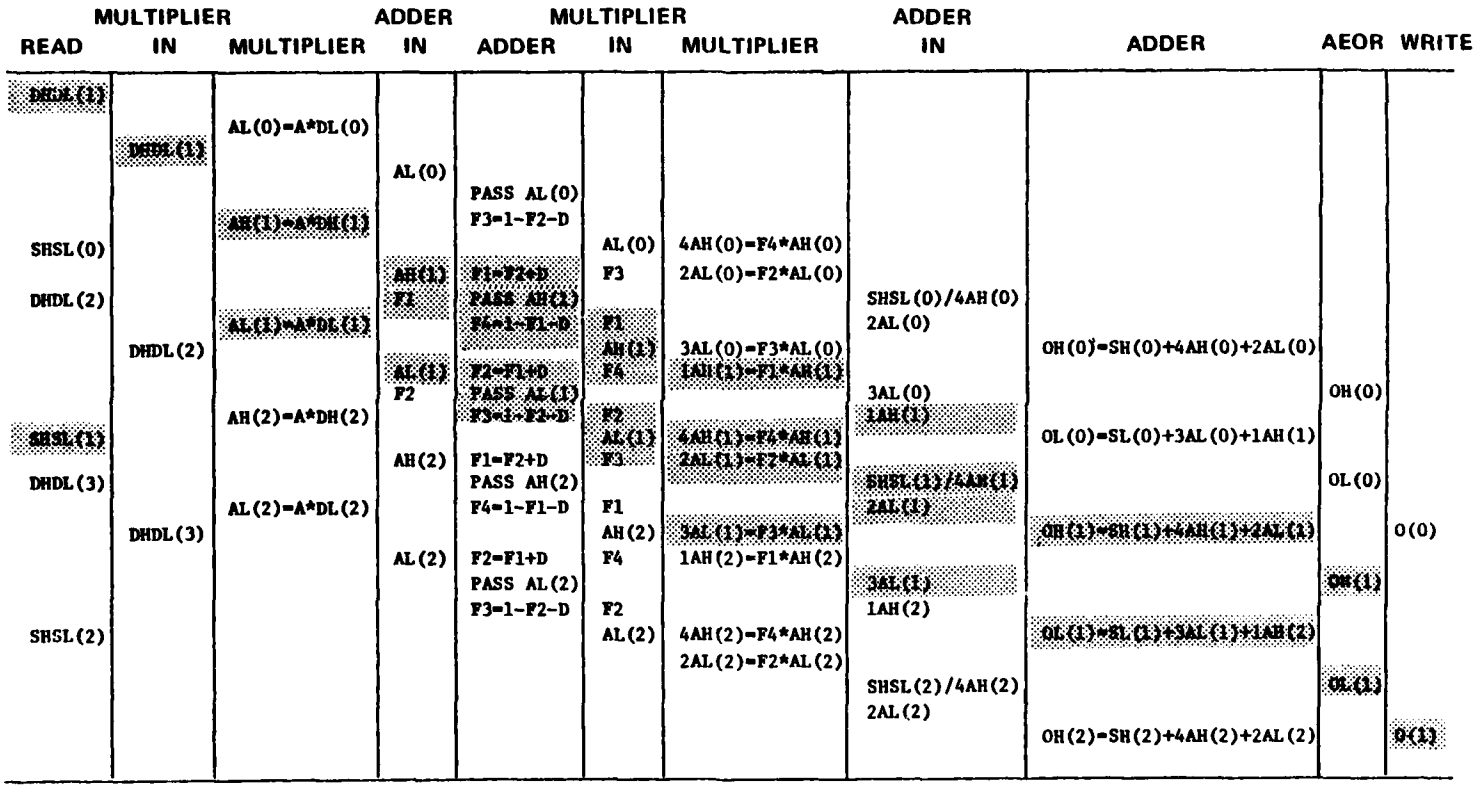
where D is the amount the fraction F is incremented each sample, and e is a small number between 0 and D. The following two transition outputs are computed:

OH = SH = e * AL + (1-e) * AH (AH from current input)



NOTE: SHADED AREA IS ONE CYCLE THROUGH THE INPHASE LOOP

FIGURE 6-1 ASSSS INPHASE MAIN LOOP FLOW



NOTE: SHADED AREA IS ONE CYCLE THROUGH THE OUTPHASE LOOP

FIGURE 6-2 ASSSS OUTPHASE MAIN LOOP FLOW

and OL = SL + (e+D) * AH + (1-e-D) * AL (AH from next input)

which, using the fractional quantities already computed, become

OH = SH + F1 * AL - F1 * AH + AL

and OL = SL - F2 * AH - F4 * AL

Finally, the next interpolation fractions are computed

F2 = e + 2D F4 = 1 - e - 2D

and computation resumes in the Outphase loop at MOPL2.

PIO02 In to Out On F2

At the time the branch is taken, the following quantities have been erroneously computed:

F2 = -1 + e F4 = -e

F1 = e + D F3 = 1 - e - D

The transitional outputs become

OL = SL + e * AH + (1-e) * AL (previous AL)

or OL = SL - F4 * AH - F2 * AL

and OH = SH + (e+D) * AL + (1-e-D) * AH (current AL)

or OH = SH + F1 * AL + F3 * AH

The adjusted interpolation fractions are:

F2 = e + 2D F4 = 1 - e - 2D

Computation resumes at MOPL1 in the Outphase loop.

POI01 Out to In On F1

At the time the branch is taken, the following quantities have been erroneously computed:

F1 = -1 + e F3 = -e

F2 = e + D F4 = 1 - e - D

The transition output is

OL = SL + e * AL + (1-e) * AH

or OL = SL - F3 * AL - F1 * AH

and the adjusted interpolation fractions are

F1 = e + D F3 = 1 - e - D

Computation resumes at MIPL1 in the Inphase loop.

POI02 Out to In On F2

Erroneously computed fractions are:

F2 = -1 + e F4 = 1 - e - 2D

F1 = e + D F3 = 1 - e - D

The transition output is:

OH = SH - e * AH + (1-e) * AL (AL from previous input)

or OH = SH + F2 * AH + AH - F2 * AL

The adjusted interpolation fractions are:

F1 = e + 2D F3 = 1 - e - 2D

Computations resume at MIPL2 in the Inphase loop.

The four transition sequences for negative fraction increment D all produce transition outputs which are not exact. Each of the outputs is in error by a term involving a fraction factor that has a magnitude of (-2D) or less. Since by definition D is much less than unity, this term can safely be ignored. The alternative would have required long transition sequences involving "backing up" the computations to obtain data which is no longer in the pipeline.

MI001 In to Out On F1, D negative

At the time the branch is taken, the following quantities have been erroneously computed:

F1 = -e F2 = 1 - e + D

F3 = -1 + e F4 = e - D

The transition outputs are:

OH = SH - F3 * AL

= SH + (1-e) * AL

(OH = SH + (1-e) * AL + e * AH is exact)

and

OL = SL + F2 * AH

= SL + (1-e+D) * AH

(OL = SL + (1-e+D) * AH + (e-D) * AL is exact).

The adjusted interpolation fractions are:

F4 = e - 2D F3 = e - 3D

F2 = 1 - e + 2D

Computation resumes at MOPL1 in the Outphase loop.

MI002 In to Out On F2, D negative

At the time the branch is taken, the following quantities have been erroneously computed:

F2 = -e F4 = -1 + e

F1 = 1 - e + D F3 = e - D

The transition output is:

OL = SL - F4 * AH

= SL + (1-e) * AH

(OL = SL + (1-e) * AH + e * AL is exact).

The adjusted interpolation fractions are:

F2 = 1 - e + D F3 = e - 2D

Computation resumes at MOPL1 in the Outphase loop.

MOI01 Out to In On F1, D negative

At the time the branch is taken the following quantities have been erroneously computed:

F1 = -e F3 = -1 + e

F2 = 1 - e + D F4 = e - D

The transition output is:

OL = SL - F3 * AL

= SL + (1-e) * AL

(OL = SL + (1-e) * AL + e * AH is exact).

The adjusted interpolation fractions are:

F4 = e - 2D

F1 = 1 - e + D

Computation resumes at MIPL1 in the Inphase loop.

MI002 Out to In On F2, D negative

At the time the branch is taken the following quantities have been erroneously computed:

F4 = -1 + e F2 = -e

F3 = e - D F1 = 1 - e + D

F4 = e - 2D

The transition outputs are:

OH = SH - 4AH

= SH + (1-e) * AH

(OH = SH + (1-e) * AH + e * AL is exact)

and OL = SL + 1AL

= SL + (1-e+D) * AL

(OL = SL + (1-e+D) * AL + (e-D) * AH is exact).

The adjusted interpolation fractions are:

F1 = 1 - e + 2D

F4 = e - 3D

Computation resumes at MIPL1 in the Inphase loop.

SCALING

See Table 6-1.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 6-2.

ADDRESS REGISTERS

AR0 - The increment code is set to one. The address field is initialized to the location of the gain parameter A.

AR1 - The increment code is set to one. The address field is initialized to the location of the F and D parameters.

AR2 - The increment code is set to one. The address field is initialized to the location of the resampled input.

AR3 - The increment code is set to one. The address field is initialized to the location of the sum input.

AR4 - The increment code is set to one. The address field is initialized to the location of the resampled output.

INCREMENT REGISTERS

Not applicable. Wrap codes must be 1024 for INC 2, 3, 4.

BRANCH REGISTERS

BCR0 - Conditional branch. For D positive, branch to PI001 on F1. For D negative, branch to MI001 on F1.

BCR1 - Conditional branch. For D positive, branch to PI002 on F2. For D negative, branch to MI002 on F2.

BR2 - Unconditional branch to MIPL2.

BR3 - Unconditional branch to MOPL2.

BCR4 - Conditional branch. For D positive, branch to POI01 on F1. For D negative, branch to MOI01 on F1.

BCR5 - Conditional branch. For D positive, branch to POI02 on F2. For D negative, branch to MOI02 on F2.

BR6 - Unconditional branch to MIPL1.

BR7 - Unconditional branch to MOPL1.

BNZ8 - Branch and Count register controlling the number of executions of

the Inphase processing loop. The Branch Address is MIPLP.

BNZ9 - Branch and Count register controlling the number of executions of the Outphase processing loop. The Branch Address is MOPLP.

BNZA - Unconditional branch to MIPLO.

BNZB - Unconditional branch to MOPLO.

BNZC - Unconditional branch to STOP.

SINE/COSINE DESTINATION REGISTERS

Not applicable.

DECIMATE REGISTER

Not applicable.

SCALE FACTOR REGISTERS

SFOH - All True/Direct.

SFOL - All prescalers - True/Direct. Postscaler - True/Left 1.

SF1H - Left prescaler - True/Right 4; center and right prescalers - True/Direct; postscaler - True/Left 4.

ARITHMETIC ELEMENT REGISTER MAP

See Table 6-3.

WORKING STORE MAP

See Table 6-4.

PROGRAM CODING CHART

See Table 6-5.

TABLE 6-1 ASSSS AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
A		H12	
F	F15		
D	F15		
DH, DL		H12	
SH, SL	H12		
INTERMEDIATE RESULTS:			
AH, AL	F8	H12	
F1, F2	F15	H15	
F3, F4	F15	H15	
1AH, 2AL	F11		
3AH, 4AL	F11		
OT	F12		
1AL, 2AH	F11		
OUTPUTS:			
OH, OL			H12

TABLE 6-2 ASSSS AEC REGISTER MAP

	Incr Code	Address Register (0 3 4 15)	WS Wrap	Increment Register (0 3 4 15)
0	1	A(A)	Ø	
1	1	A(F, D)	Ø	
2	1	A (Resample Input)	Ø	
3	1	A (Sum Input)	Ø	
4	1	A (Output)	Ø	
5				
6				
7				

Memory Address Register

MLPO or MLPI

Sin/Cos Destination Register

	AE 0	AE 1	AE 2	AE 3
0				
1				
2				
3				

Decimate Register

Count	Reset

BR/BCR Unconditional/ Conditional Branch — Branch Address (0 15)

0	(I001)
1	(I002)
2	MIPL2
3	MOPL2
4	(OI01)
5	(OI02)
6	MIPL1
7	MOPL1

BNZ Branch and Count Registers

	Count (0 7)	Reset (8 15)	Spare (16 21)	Branch Address (22 31)
8	CI	CI	X	MIPLP
9	CO	CO	X	MOPLP
A	X'FF'	X'FF'	X	MIPLØ
B	X'FF'	X'FF'	X	MOPLØ
C	X'FF'	X'FF'	X	STOP
D			X	
E			X	
F			X	

Comments

For Out Phase start,
MAR = MLPO
For In Phase start,
MAR = MLPI

For D Positive,

BRØ = PI001
BR1 = PI002
BR4 = POI01
BR5 = POI02

For D Negative

BRØ = MI001
BR1 = MI002
BR4 = MOI01
BR5 = MOI02

TABLE 6-3 ASSSS AE REGISTER MAP

MLR

ØH	DH	DL	ØL
1H	F1	F3	1L
2H	F2	F4	2L
3H	DAL		3L

MRR

ØH	AH	AL	ØL
1H	A	-1	1L
2H			2L
3H			3L

ALIR

ØH	DH |	DL	ØL
1H	SH |	SL	1L
2H	F |		2L
3H	D |		3L

ACIR

ØH	1AH		ØL
1H	2AL		1L
2H	A |	-1	2L
3H	1AH		3L

ARIR

Ø	3AL/1AL
1	4AH/2AH
2	AH
3	AL/FAL

AEOR

ØH	OH |	OL	ØL
1H	|		1L
2H	|		2L
3H	|		3L

SCALE FACTOR REGISTERS

AEØ

ØH	Ø | Ø | Ø | Ø	Ø | Ø | Ø | 3	ØL
1H	3 | Ø | Ø | 3	| | |	1L
2H	| | |	| | |	2L
3H	| | |	| | |	3L

AE1

ØH	Ø | Ø | Ø | Ø	Ø | Ø | Ø | 3	ØL
1H	3 | Ø | Ø | 3	| | |	1L
2H	| | |	| | |	2L
3H	| | |	| | |	3L

ALOR

Ø	OT
1	
2	
3	

ACOR

Ø	F1
1	F2
2	F3
3	F4

AROR

Ø	Ø
1	ONE
2	ONE
3	Ø

TABLE 6-4 ASSSS WORKING STORE MAP

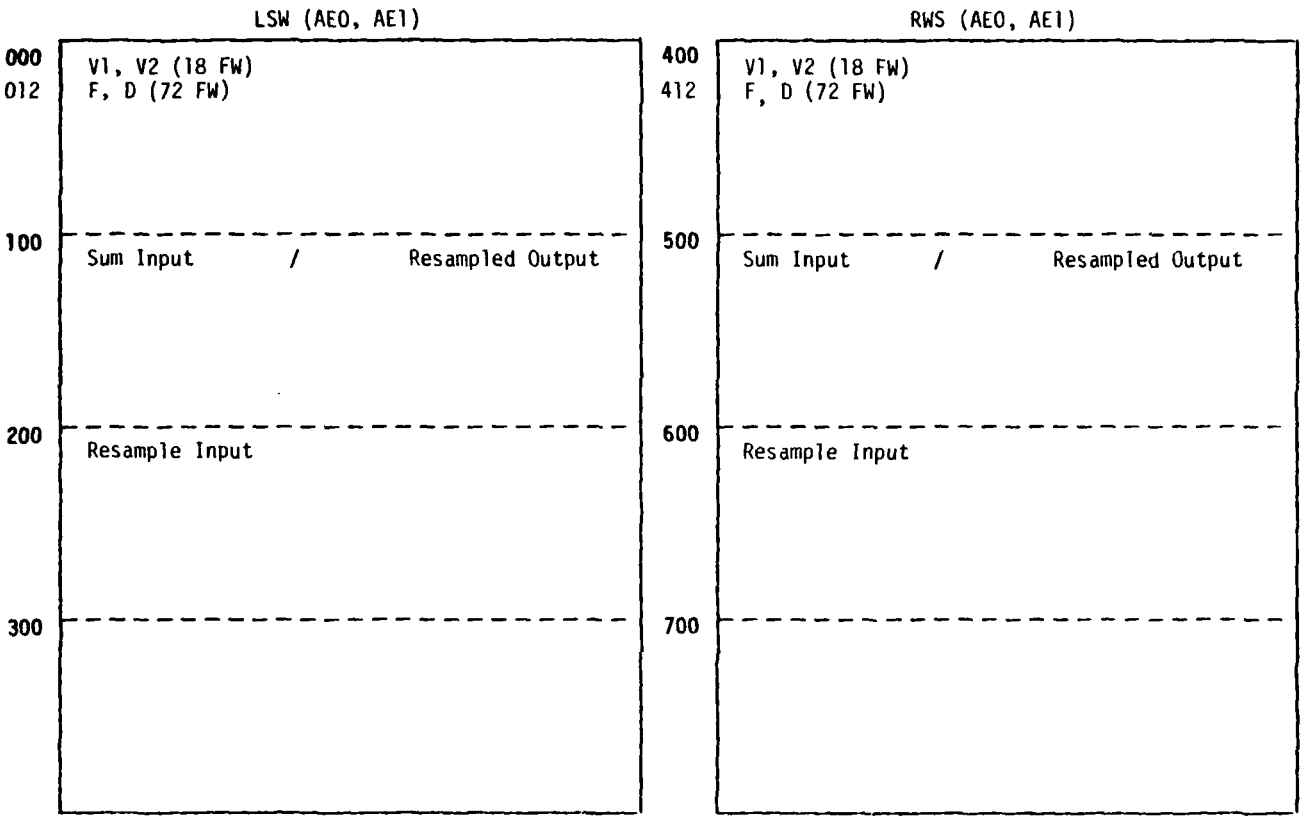


TABLE 6-5 ASSSS PROGRAM CODING CHART

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	MLP1		AR0 +0			NOP								
			READ A,-1											
			AR1 +0			NOP								
			READ F		A,-1 1			A,-1 1		Ø = Inhibit ØH				
			AR2 +0							Ø = Inhibit ØH				
			READ DHDL				F 2			Ø = Inhibit ØH	Ø Ø		Ø Ø	
			AR2 +1							ONE = -(-1)-Ø-Ø ØH			Ø 3	
			READ DHDL	DHDL Ø			DHDL Ø			ONE = -(-1)-Ø-Ø ØH			ONE 1	
			AR1 +1			AL = A*DL				ØH			ONE 2	
			READ D	DHDL Ø			DHDL Ø			ØH				
						AH=A*DH			AL 3	F3=ONE-F ØH				
							D 3			AL ØL		F3 2		
			AR3 +0	F3 1L					AH 2	F1 = F ØH R				
			READ SHSL		AL ØL					SETCS AH ØL R		E1 Ø		
			AR2 +1	F1 1H		AL=A*DL				F4=-F1-D+1C ØH R				
			READ DHDL		AH ØH	3AL=F3*AL	SHSL 1			R		F4 3		
			AR4 -1	F4 2L		1AH=F1*AH			AL 3	F2=F1+D ØH R				
	BNZA MIPLØ		READ DUMMY	DHDL Ø			DHDL Ø		3AL Ø	SETCS AL ØL		F2 1		
				F2 2H		AH=A*DH		1AH Ø		F3=-F2-D+1C ØH R				
	BCR1 1002			DUMMY 3	AL ØL	4AH=F4*AH				ØH=SH+1AH+3AL TH R		F3 2		

TABLE 6-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD		ALOR	ACOR	AROR	AEOR
	MLPO		AR0 +0							Ø = Inhibit	ØH				
			READ A,-1							Ø = Inhibit	ØH				
			AR2 +0								ØH	Ø Ø		Ø Ø	
			READ DHDL		A-1 1			A,-1 1						Ø 3	
			AR1 +0							ONE = -(-1)-Ø-Ø	ØH				
			READ F	DHDL Ø			DHDL Ø			ONE = -(-1)-Ø-Ø	ØH			ONE 1	
			AR1 +1			AH = A*DH								ONE 2	
			READ D				F 2								
			AR2 +1			AL = A*DL			AH 2	F4 = ONE - F	ØH				
			READ DHDL				D 3			AH	ØL		F4 3		
			AR4 -1	F4 2L					AL 3	F2 = F	ØH R				
			READ DUMMY	DHDL Ø	AH ØH		DHDL Ø			AL SETCS	ØL R		F2 1		
			AR3 +0	F2 2H		AH = A*DH				F3 = -F2-D+ ONE	ØH R				
			READ SHSL		AL ØL	4AH =F4*AH		DUMMY 3			R		F3 2		
			AR2 +1	F3 1L		2AL = F2*AL			AH 2	F1 = F2 + D	ØH R				
	BNZB MOPLØ		READ DHDL				SHSL 1		4AH 1	AH SETCS	ØL		F1 Ø		
				F1 1H		AL = A*DL		2AL 1		F4 = -F1-D+1C	ØH R				
	BCR4 0101			DHDL Ø	AH ØH	3AL = F3*AL	DHDL Ø			OH = SH+2AL+4AH	TH R		F4 3		

TABLE 6-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	MIPLP		AR4 +1	F2 2H		AH = A*DH		1AH Ø		F3 = -F2-D+1C ØH R				
	BCR1 1002		WRITE OHOL		AL ØL	4AH = F4*AH				OH = SH+1AH+3AL 1H R		F3 2		
	MIPLØ		AR2 +1	F3 1L		2AL = F2*AL			AH 2	F1 = F2+D+ØC ØH R				
			READ DHDL						4AH 1	AH SETCS ØL R		F1 Ø		OH ØH
	MIPL3		AR3 +1	F1 1H		AL = A*DL		2AL 1		F4 = -F1-D+1C ØH R				
	BNZ8 MIPLP		READ SHSL	DHDL Ø	AH ØH	3AL = F3*AL	DHDL Ø			OL = SL+2AL+4AH 1H R		F4 3		
	MIPL1			F4 2L		1AH = F1*AH			AL 3	F2 = F1+D+ØC ØH R				
	BCRØ 1001						SHSL 1		3AL Ø	AL SETCS ØL R		F2 1		OL ØL
			AR4 +1	F2 2H		NOP								
	BNZC STOP		WRITE OHOL											
						NOP								
						STOP								

TABLE 6-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	MOPLP		AR4 +1	F1 1H		AL = A*DL		2AL 1		F4 = F1-D+1C ØH R				
	BCR4 0101		WRITE OHOL	DHDL Ø	AH ØH	3AL = F3*AL	DHDL Ø			OH = SH+2AL+4AH 1H R		F4 3		
	MOPLØ			F4 2L		1AH = F1*AH			AL 3	F2 = F1 +D+ØC ØH R				
									3AL Ø	AL SETCS ØL R		F2 1		OH ØH
	MOPL2		AR3 +1	F2 2H		AH = A*DH		1AH Ø		F3 = -F2-D+1C ØH R				
	BNZ9 MOPLP		READ SHSL		AL ØL	4AH = F4*AH				OL = SL+1AH+3AL 1H R		F3 2		
	MOPL1		AR2 +1	F3 1L		2AL = F2*AL			AH 2	F1=F2+D+ØC ØH R				
	BCR5 0102		READ DHDL				SHSL 1		4AH 1	AH SETCS ØL R		F1 Ø		OL ØL
			AR4 +1	F1 1H		NOP								
	BNZC STOP		WRITE OHOL	DHDL Ø			DHDL Ø							
						NOP								
						STOP								

TABLE 6-5 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	P1001		AR2 +1	F3 1L		1AL = F1*AL			AH 2	OT = SH+AL ØL				
			READ DHDL							AH ØL				
						AL = A*DL			1AL Ø	F4 = F3 ØH	OT Ø			
	BR3 MOPL2			DHDL Ø	AH ØH	3AL = F4*AL	DHDL Ø			OH = OT-1AH+1AL 1H R		F4 3		
			F4	2L		1AH = F2*AH			AL 3	F2 = F2+D ØH				
									3AL Ø	AL ØL R		F2 1		OH ØH
	P1002			F1 1H		AL = A*DL		2AL 1						
				DHDL Ø	AH ØH		DHDL Ø			F4 = F3 ØH R				
						4AH = F4*AH			AL 3	F2 = F1 ØH		F4 3		
	BR7 MOPL1		F4	2L						AL ØL		F2 1		
			AR3 +1 F2	2H		AH = A*DH			4AH 1	F3 = -F2-D+ONE ØH				
	BNZ9 MOPLP		READ SHSL		AL ØL	4AH = F4*AH				OL = SL-2AL-4AH 1H R		F3 2		

TABLE 6-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	POI01			F2 2H		AH = A*OH		AH Ø						
					AL ØL					F3 = F4 ØH R				
			AR2 +1			3AL = F3*AL			AH 2	F1 = F2 ØH		F3 2		
	BR6 MIPL1		READ DHDL	F3 1L						AH ØL		F1 Ø		
			AR3 +1	F1 1H		AL = A*DL			3AL Ø	F4 = -F1-D+ONE ØH				
	BNZ8 MIPLP		READ SHSL	DHDL Ø	AH ØH	3AL = F3*AL	DHDL Ø			ØL = SL-1AH-3AL 1H R		F4 3		
	POI02			F4 2L		2AH = F2*AH			AL 3	OT = SH+AH 1H				
										AL ØL				
						AH = A*DH			2AH 1	F3 = F4 ØH	OT Ø			
	BR2 MIPL2				AL ØL	4AH = F3*AH				OH = OT-2AL+2AH 1H R		F3 2		
			AR2 +1	F3 1L		2AL = F1*AL			AH 2	F1 = F1+D ØH				
			READ DHDL							AH ØL R		F1 Ø		OH ØH

TABLE 6-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	MI001			1L F3						1H OH = SH-3AL				
										ØH F4 = -F2-D+ONE				
						1AH = F2*AH				ØH F2 = F2+D				ØH OH
	BR7 MOPL1			2L F4						ØH F2 = F2+D		1 F2		
			+1 AR3			AH = A*DH		Ø 1AH		ØH F3 = -F2-D+ONE				
	BNZ9 MOPLP		READ SHSL	2H F2		4AH = F4*AH				1H OL = SL+1AH		2 F3		
	MI002		-1 AR2 READ DHDL	1H F1										
										ØH F2 = F1				
	BR7 MOPL1			Ø DHDL			Ø DHDL					1 F2		
			+1 AR3	2H F2		AH = A*DH				ØH F3 = -F1-D+ONE				
	BNZ9 MOPLP		READ SHSL			4AH = F3*AH				1H OL = SL-4AH		2 F3		

TABLE 6-5 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	MOI01			2H F2						ØH				
										ØL DAL = -AL				
	BR6 MIPL1			3H DAL						ØK F1 = F2				
			+1 AR3 READ SHSL			AL = -1*DAL				ØH F4 = F4-D		Ø F1		
	BNZ8 MIPLP			1H F1		3AL = F4*AL				1H OL = SL-3AL		3 F4		
	MOI02			2L F4					3 AL	1H OH = SH-4AH				ØH OH
										ØL DAL = -AL				
						1AL = F1*AL				ØH R F1 = F1+D		Ø F1		
	BR6 MIPL1			3H DAL										
			+1 AR3 READ SHSL	1H F1		AL = DAL*-1			Ø 1AL	ØH F4 = F4-D				
	BNZ8 MIPLP					3AL = F4*AL				1H OL = SL+1AH+1AL		3 F4		

CHAPTER 7

SCALE AND ADD THREE ARRAYS (SCLA3) MACRO

FUNCTIONAL DESCRIPTION

SCLA3 MACRO scales an array X by a constant A, scales an array Y by a constant B, scales an array Z by a constant C, and combines the three scaled arrays on an element-by-element basis to form an output array W. X, Y, Z, and W must be of the same number of elements N.

SCLA3 is intended to provide arbitrary linear mixing of three inputs, and for convenience, the coefficients A, B, and C are accessed by the AE from Working Store. Separate address registers are used for the four arrays, and the output may overwrite any of the three inputs or the scaling parameters A, B, and C.

MATHEMATICAL DESCRIPTION

$$W_i = A * X_i + B * Y_i + C * Z_i; \quad i = 1, 2, \ldots, N$$

where A, B, and C are scalers, and X_i, Y_i, Z_i and W_i are the i^{th} elements of arrays X, Y, Z and W, respectively.

IMPLEMENTATION

SCLA3 is implemented with an eight-instruction main loop which processes one pair of X's, one pair of Y's and one pair of Z's to form one pair of W's. A twelve-instruction preamble loads the three coefficients from Working Store using Address Register AR0, and loads the pipe with the first pairs of inputs from the three input arrays. In addition, the preamble decrements the output Address Register AR4 by one to allow the first output to be written into the initial value of AR4. Each pass through the main loop then stores one pair of W's in the output array. The main loop is controlled by BNZ9.

SCALING

See Table 7-1.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 7-2.

ADDRESS REGISTERS

AR0 - The increment code is set to 1. The address field is initialized to the location of the parameters A, B, and C.

AR1 - The increment code is set to 1. The address field is initialized to the first address of the X array.

AR2 - The increment code is set to 1. The address field is initialized to the first address of the Y array.

AR3 - The increment code is set to 1. The address field is initialized to the first address of the Z array.

AR4 - The increment code is set to 1. The address field is initialized to the first address of the W array.

INCREMENT REGISTERS

Not applicable. Wrap codes must be 1024 for INC 0 to 4.

BRANCH REGISTERS

BNZ9 - Branch and Count Register controlling the main loop. Reset and Count fields are initialized to N/2-1, where N is the number of elements in each of the arrays. Branch Address field is set to SC3LP.

SINE/COSINE DESTINATION REGISTERS

Not applicable.

DECIMATE REGISTER

Not applicable.

SCALE FACTOR REGISTERS

SFOL - Left, center and right prescalers - True/Direct. Postscaler - True/Left 4.

ARITHMETIC ELEMENT REGISTER MAP

See Table 7-3.

WORKING STORE MAP

See Table 7-4.

PROGRAM CODING CHART

See Table 7-5.

TABLE 7-1 SCLA3 AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
A, B, C		H12	
X, Y, Z		H12	
INTERMEDIATE RESULTS:			
AX	F8		
BY	F8		
CZ	F8		
OUTPUTS:			
W			H12

TABLE 7-2 SCLA3 AEC REGISTER MAP

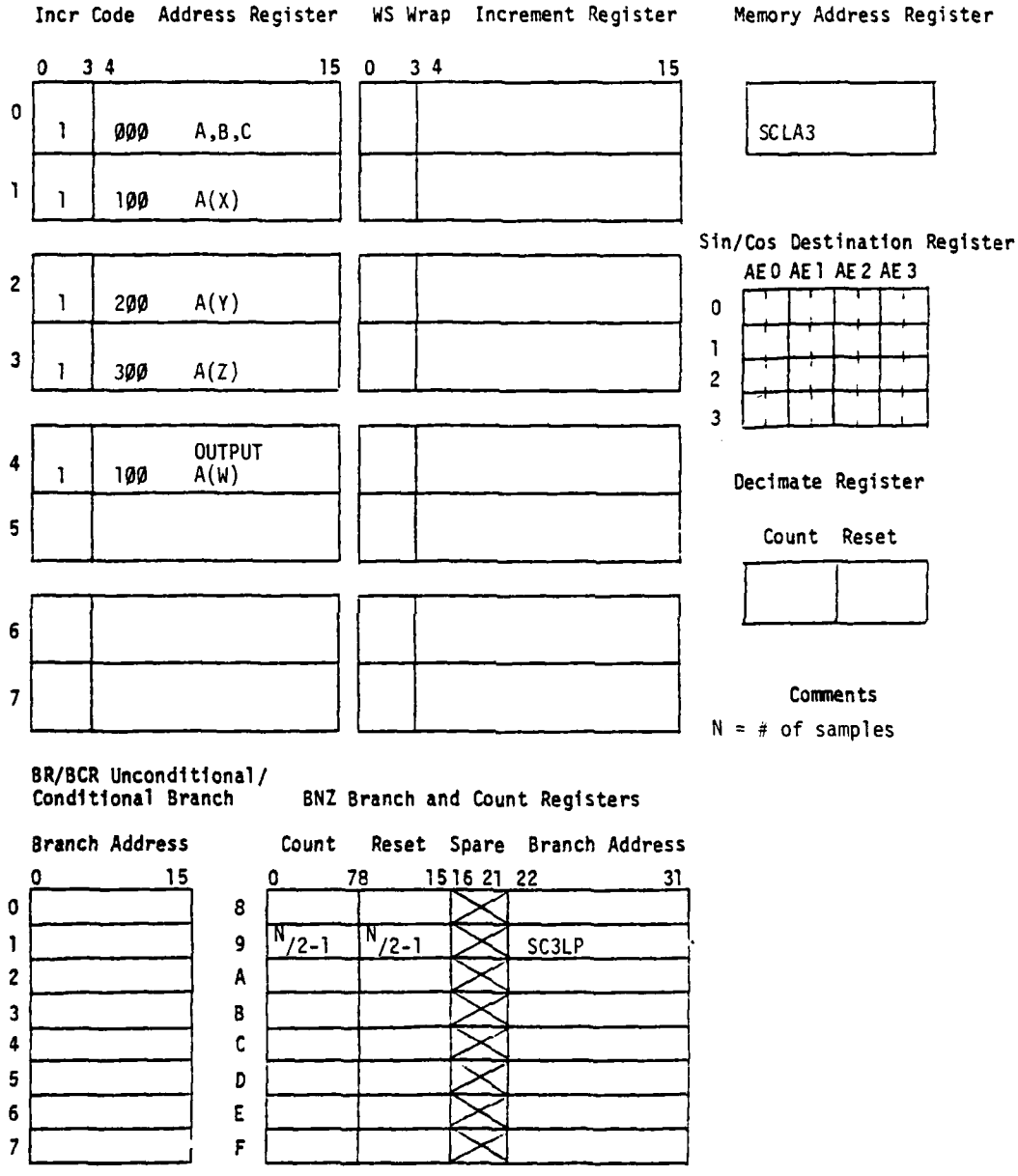


TABLE 7-3 SCLA3 AE REGISTER MAP

MLR

ØH	X1	X2	ØL
1H	Y1	Y2	1L
2H	Z1	Z2	2L
3H			3L

MRR

ØH	A	B	ØL
1H	C		1L
2H			2L
3H			3L

ALIR

ØH	AX|1	ØL
1H	AX|2	1L
2H	DUMMY	2L
3H	|	3L

ACIR

ØH	BY|1	ØL
1H	BY|2	1L
2H	|	2L
3H	|	3L

ARIR

Ø	CZ1
1	CZ2
2	
3	

AEOR

ØH	W1	W2	ØL
1H			1L
2H			2L
3H			3L

SCALE FACTOR REGISTERS

AEØ

ØH	Ø | Ø | Ø | 3	| | |	ØL
1H	| | |	| | |	1L
2H	| | |	| | |	2L
3H	| | |	| | |	3L

AE1

ØH	| | |	| | |	ØL
1H	| | |	| | |	1L
2H	| | |	| | |	2L
3H	| | |	| | |	3L

ALOR

Ø	
1	
2	
3	

ACOR

Ø	
1	
2	
3	

AROR

Ø	
1	
2	
3	

TABLE 7-4 SCLA3 WORKING STORE MAP

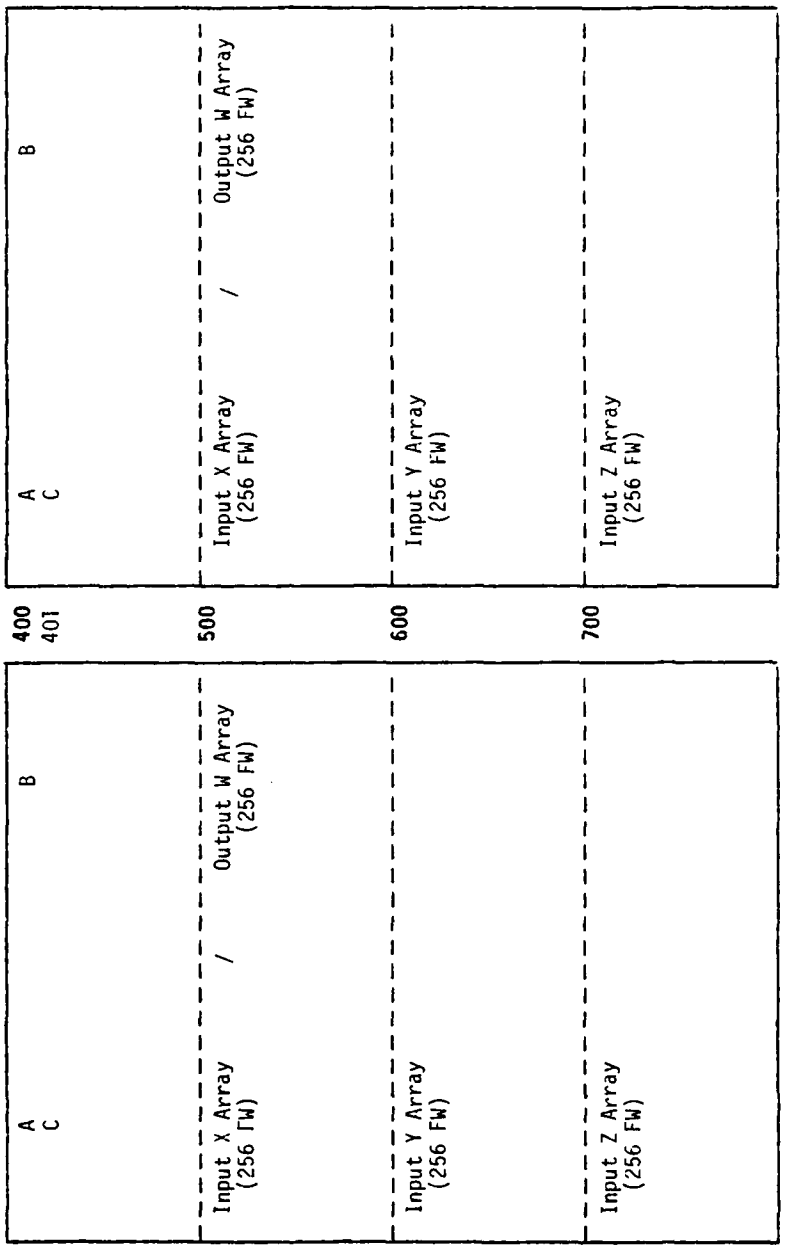


TABLE 7-5 SCLA3 PROGRAM CODING CHART

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
			+Ø											
	SCLA3		AR0			NOP								
			READ A,B											
			+1		Ø									
			AR0		A,B	NOP								
			READ C											
			+Ø		1									
			AR1		C	NOP								
			READ X1,X2											
			+Ø	Ø										
			AR2	X1,X2		NOP								
			READ Y1,Y2			AX1 = A*X1								
			+Ø	1										
			AR3	Y1,Y2		AX2 = A*X2								
			READ Z1,Z2			BY1 = B*Y1	Ø AX1							
			-1	2			1							
			AR4	Z1,Z2		BY2 = B*Y2	AX2							
			READ					Ø						
	SC3LP		DUMMY			CZ1 = C*Z1		BY1						
			+1				2	1						
			AR1			CZ2 = C*Z2	DUMMY	BY2						
			READ X3,X4						Ø					
			+1	Ø					CZ1 1	ØH				
			AR2	X3,X4					CZ2	W1=AX1+BY1+CZ1				
			READ Y3,Y4			AX3 = A*X3				ØH W2=AX2+BY2+CZ2				ØH
			+1	1										
	BNZ9		AR3	Y3,Y4		AX4 = A*X4				R				W1
			READ				Ø							ØL
			Z3,Z4			BY3 = B*Y3	AX3			R				W2
			+1	2			1							
			AR4	Z3,Z4		BY4 = B*Y4	AX4							
			WRITE W1,W2			NOP								
						STOP								

CHAPTER 8

FILTER TWO AND SUM THREE ARRAYS (F22S3) MACRO

FUNCTIONAL DESCRIPTION

F22S3 MACRO provides filtering of two input arrays X and Y to produce filter outputs F and G, respectively, and then combines F and G with a third array S to form a new output array S. The two filters for the X and Y inputs are two-pole recursive with separately specifiable gain and pole position parameters. The parameters and the initial states of the two filters are obtained from Working Store. The final states of the filters are returned to the same locations in Working Store. Separate Address Registers are provided for each of the input arrays and the output array. The output array may overwrite any of the input arrays or the parameters, but not the filter states. All arrays must be of the same number of elements N. The output S may be passed through a final filter consisting of a single zero at Z = +1 (i. e., a zero at zero frequency). This option is controlled by a Scale Factor Registers 1H and 1L, with a "0101, 0010" giving no zero and a "0111, 0111" enabling the filter.

MATHEMATICAL DESCRIPTION

The two filtering operations are:

$$F_i = U * X_i + A * F_{i-1} + B * F_{i-2}; \quad i = 1, \ldots, N$$

and

$$G_i = V * Y_i + C * G_{i-1} + D * G_{i-2}; \quad i = 1, \ldots, N$$

F_0, G_0, F_{-1}, and G_{-1} are initialized by the preamble.

The summing operation is:

$$S_i = S_i + F_i + G_i; \quad i = 1, \ldots, N.$$

X_i, Y_i, and S_i are the i^{th} elements of the X, Y, and S arrays respectively, and F_i and G_i are partial results resident only in the AE registers. F_{N-1}, G_{N-1}, F_N and G_N are returned to Working Store, overwriting the initial states.

IMPLEMENTATION

F22S3 is implemented with a twelve-instruction main loop, a twenty-two instruction preamble and a six-instruction clean-up sequence. The preamble loads the parameters from Working Store via Address Register AR1, loads the initial filter states via Address Register AR5, and loads the pipe with the first pairs of X's, Y's and S's. Each pass through the main loop then produces a pair of filtered F's, a pair of filtered G's, and combines these with a pair of input S's to produce a pair of output S's. The clean-up sequence returns the final filter states to Working Store and stops. The main loop is controlled by BNZ8.

SCALING

See Table 8-1.

The filter coefficients A, B, C and D are treated as H14 numbers. The gain coefficients U and V are treated as H12 numbers. Assuming the inputs X and Y and the filter states F and G to be H0, the partial results AF, BF, CG and DG then are scaled F14, and the modified inputs UX and VY are scaled F12. SFOL is used in forming new filter states and shifts the AF, BF, CG and DG terms right by two on input. Outputs saved in ACOR and AROR do not pass through the postscaler, and, therefore, are F12. Outputs passed back to the multiplier are shifted left by four to regain a scaling of H0.

The summing operation uses SFOH which right shifts the S input four to agree with the F and G inputs, and shifts the output left four to achieve a final output scaling of H0. The filter states passed to the output to be saved for the next execution of F22S3 are also shifted left four by SFOH.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 8-2.

ADDRESS REGISTERS

AR0 - The increment code is set to 1. The address field is initialized to the beginning of the input S array.

AR1 - The increment code is set to 1. The address field is initialized to the beginning of the parameters block.

AR2 - The increment code is set to 1. The address field is initialized to the beginning of the input X array.

AR3 - The increment code is set to 1. The address field is initialized to the beginning of the input Y array.

AR4 - The increment code is set to 1. The address field is initialized to the beginning of the output S array.

AR5 - The increment code is set to 1. The address field is initialized to the beginning of the filter states block.

INCREMENT REGISTERS

INC0 through 5 - The wrap field should be set to 1024.

BRANCH REGISTERS

BNZ8 - Branch and Count Register controlling the main loop. Count and Reset fields are initialized to N/2-1, where N is the number of elements in each of the arrays. The Branch Address field is set to F22LP.

SINE/COSINE DESTINATION REGISTERS

Not applicable.

DECIMATE REGISTER

Not applicable.

SCALE FACTOR REGISTERS

SF0H - Left prescaler - True/Right 4, Center and Right prescalers - True/Direct, Postscaler - True/Left 4.

SF0L - Left prescaler - True/Direct, Center and Right prescalers - True/Right 2, Postscaler - True/Left 4.

SF1H - Left and center prescalers - True/Direct, Right prescaler - True/Right 1 or Inhibit, Postscaler - True/Left 4.

SF1L - Left prescaler - True/Direct, Center prescaler - True/Right 1 or Inhibit, Right prescaler - True/Direct, Postscaler - True/Left 4.

ARITHMETIC ELEMENT REGISTER MAP

See Table 8-3.

WORKING STORE MAP

See Table 8-4.

PROGRAM CODING CHART

See Table 8-5.

TABLE 8-1 F22S3 AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
FH, FL		H12	
GH, GL		H12	
U, V		H12	
A, C		H14	
B, D		H14	
XH, XL		H12	
YH, YL		H12	
SH, SL	H12		
SSH, SSL	H12		
INTERMEDIATE RESULTS:			
UXH, UXL	F8		
VYH, VYL	F8		
AFH, AFL	F10		
BFH, BFL	F10		
CGH, CGL	F10		
DGH, DGL	F10		
FH, FL	F8	H12	
GH, GL	F8	H12	
SSH, SSL	F8		
OUTPUTS:			
FHO, FLO			H12
GHO, GLO			H12
SHO, SLO			H12

TABLE 8-2 F22S3 AEC REGISTER MAP

Incr Code Address Register

	0	3 4	15
0	1	S's (Input)	
1	1	Parameter (AC, BD, HV)	
2	1	X's	
3	1	Y's	
4	1	S's (Output)	
5	1	States (FC, GO, SSO)	
6			
7			

WS Wrap Increment Register

	0	3 4	15
0			
1			
2			
3			
4			
5			
6			
7			

Memory Address Register

F22S3

Sin/Cos Destination Register

	AE 0	AE 1	AE 2	AE 3
0				
1				
2				
3				

Decimate Register

Count	Reset

Comments
N = # of Data Samples

BR/BCR Unconditional/
Conditional Branch

Branch Address

	0	15
0		
1		
2		
3		
4		
5		
6		
7		

BNZ Branch and Count Registers

	0	7 8	15 16	21 22	31
8	$N/2-1$	$N/2-1$	X	F22LP	
9			X		
A			X		
B			X		
C			X		
D			X		
E			X		
F			X		

TABLE 8-3 F22S3 AE REGISTER MAP

MLR

ØH	XH	XL	ØL
1H	YH	YL	1L
2H	A	C	2L
3H	B	D	3L

MRR

ØH	U	V	ØL
1H			1L
2H	FH	FL	2L
3H	GH	GL	3L

ALIR

ØH	SH	SL	ØL
1H	SSH	SSL	1L
2H	U	X	2L
3H	V	Y	3L

ACIR

ØH	Dummy		ØL
1H			1L
2H	A	F	2L
3H	C	G	3L

ARIR

Ø	
1	
2	BF
3	DG

AEOR

ØH	SHO	SLO	ØL
1H	[illegible]	FLO	1L
2H	GHO	GLO	2L
3H		SSL	3L

SCALE FACTOR REGISTERS

AEØ

ØH	3 Ø Ø 3	Ø 2 2 3	ØL
1H	Ø Ø 1/4 3	Ø 1/4 Ø 3	1L
2H			2L
3H			3L

AE1

ØH	3 Ø Ø 3	Ø 2 2 3	ØL
1H	Ø Ø 1/4 3	Ø 1/4 Ø 3	1L
2H			2L
3H			3L

ALOR

Ø	
1	
2	
3	

ACOR

Ø	SSH
1	FH
2	FL
3	

AROR

Ø	SSL
1	GH
2	GL
3	

TABLE 8-4 F22S3 WORKING STORE MAP

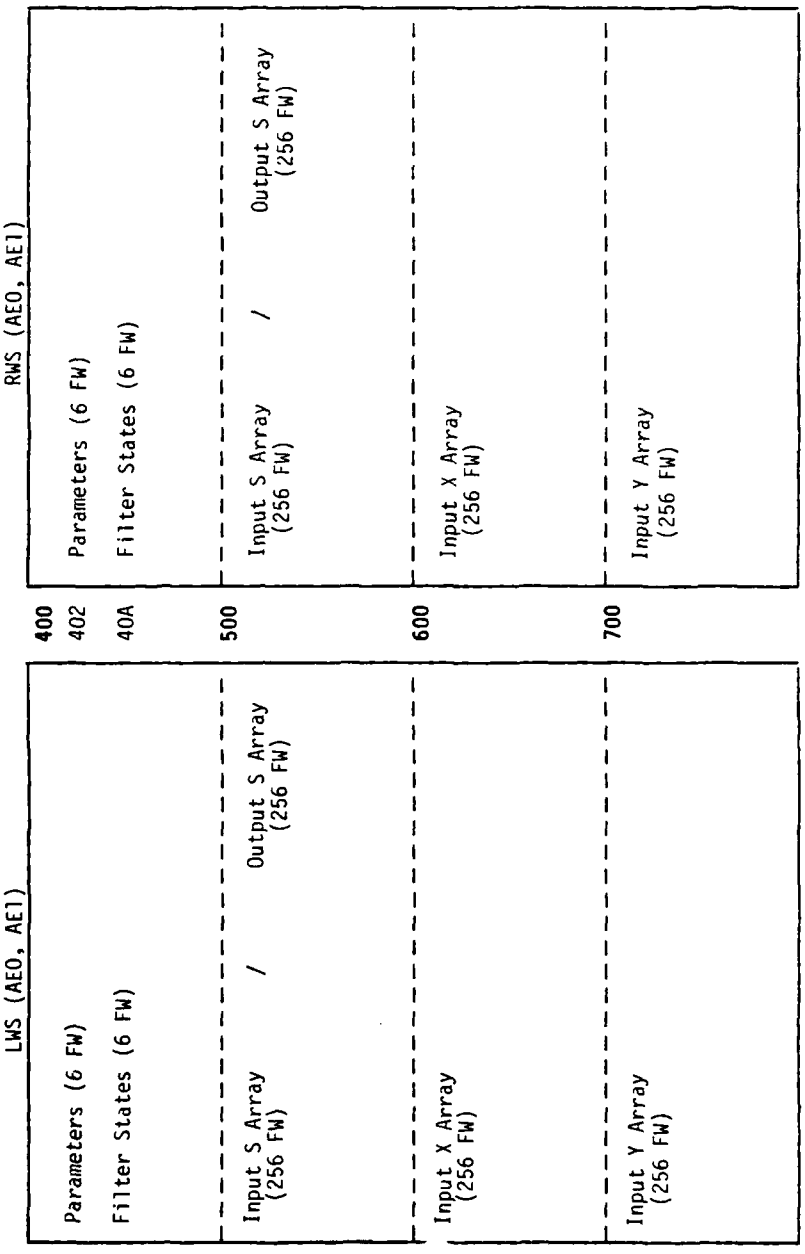
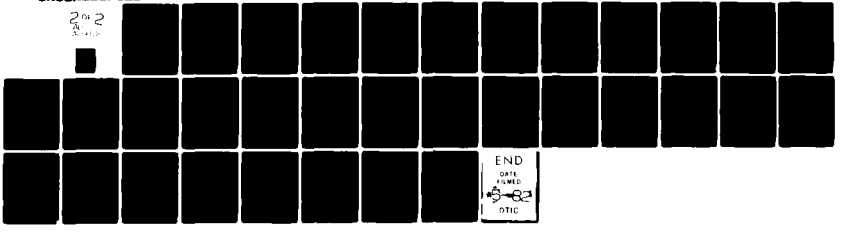


TABLE 8-5 F22S3 PROGRAM CODING CHART

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	F22S3		+Ø AR5											
			READ F0											
			+1 AR5											
			READ GO		FH,FL 2									
			+0 AR1											
			READ A,C		GH,GL 3									
			+1 AR1											
			READ B,D	A,C 2										
			+1 AR1			AFL=A*FL								
			READ U,V	B,D 3		CGL=C*GL								
			+Ø AR2			BFH=B*FH		AFL 2						
			READ XH,XL		U,V Ø	DGH=D*GH		CGL 3						
			+Ø AR3						BFH 2					
			READ YH,YL	XH,XL Ø					DGH 3					
			+1 AR5			UXH=U*XH								
			READ SSQ	YH,YL 1										
			+Ø AR0			VYH=V*YH	UXH 2							
			READ Cd,Sd				SSH,SSL 1			FH=UXH+AFL+BFH ØL				
			-1 AR4			BFL=B*FL	VYH 3			PASS SSL ØH		FH 1		
			READ DUMMY		FH 2H	DGL=D*GL	SH,SL Ø			R			SSL Ø	
			+1 AR2			UXL=U*XL			BFL 2	GH=VYH+CGL+DGH ØL R				
			READ XH,XL			VYL=V*YL		DUMMY Ø	DGL 3	FHO=FH ØH			GH 1	

AD-A113 821 NAVAL SURFACE WEAPONS CENTER SILVER SPRING MD F/G 9/2
ARITHMETIC PROCESSOR (AP) MICROPROGRAMS FOR DASS.(U)
AUG 81 R H DAVIS, D M GREATHOUSE
UNCLASSIFIED NSWC/TR-81-313 SBI-AD-F500 008 NL



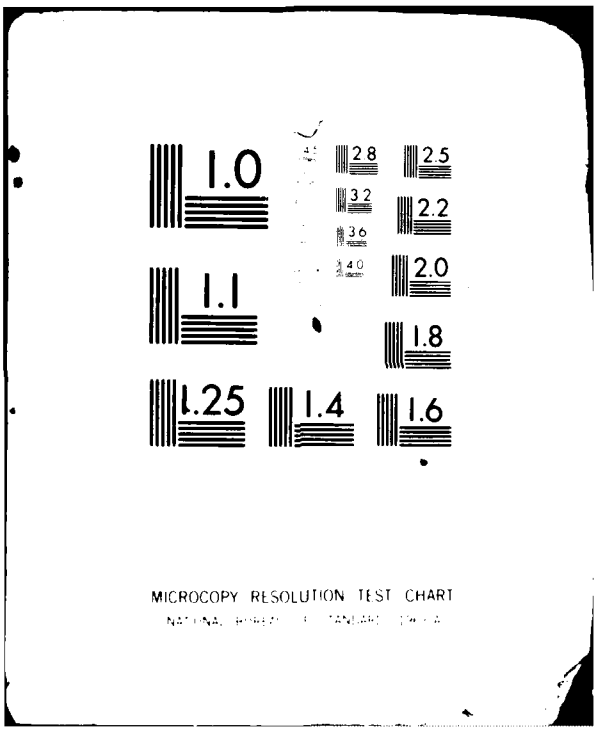

1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

TABLE 8-5 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	F22LP		AR3 +1		GH 3H	AFH=A*FH	UXL 2			SSH=SH+FH+GH ØH R				
			READ YH,YL	XH,XL Ø		CGH=C*GH	VYL 3			GHO=GH ØH R		SSH Ø		FHO 1H
						BFH=B*FH		AFH 2		SHO=SSH(-SSL) 1H R				
				YH,YL 1		DGH=D*GH		CGH 3		FL=UXL+AFH+BFL ØL R				GHO 2H
						UXH=U*XH			BFH 2	GL=VYL+CGH+DGL ØL R		FL 2		SHO ØH
					FL 2L	VYH=V*YH			DGH 3	FLO=FL ØH R			GL 2	
			ARØ +1		GL 3L	AFL=A*FL	UXH 2			SSL=SL+FL+GL ØH R				
			READ SH,SL			CGL=C*GL	VYH 3			GLO=GL ØH R			SSL Ø	FLO 1L
			AR2 +1			BFL=B*FL		AFL 2		SLO=SSL(-SSH) 1L R				
	BNZ8		READ XH,XL			DGL=D*GL	SH,SL Ø	CGL 3		FH=UXH+AFL+BFH ØL R				GLO 2L
			AR4 +1			UXL=U*XL			BFL 2	GH=VYH+CGL+DGH ØL R		FH 1		SLO ØL
			WRITE SO	XH,XL Ø	FH 2H	VYL=V*YL			DGL 3	FHO=FH ØH R			GH 1	
										PASS SSL ØH				
						NOP								
			AR5 +Ø											SSL 3L
			WRITE SSO											
			AR5 -1			NOP								
			WRITE GO											
			AR5 -1			NOP								
			WRITE FO											
						NOP								
						STOP								

CHAPTER 9

DEMODULATED NOISE (DEMON) MACRO

FUNCTIONAL DESCRIPTION

The DEMON MACRO generates two channels of modulated broadband noise and adds the two on a sample-by-sample basis to a composite channel. Two channels of white noise are input to the MACRO which filters each with a two-pole recursive filter to provide some spectral shaping of the broadband noise. The modulating function for each channel is derived by truncating a sinusoid with arbitrary amplitude, mean and frequency.

MATHEMATICAL DESCRIPTION

DEMON computations are conveniently grouped into three phases:

a) Generate Modulating Functions

b) Generate Filtered Noises

c) Form Composite Output Signal

Phase 1

For the first DEMON channel (AE0):

S_{1i} = COS(AR0) AR0 = AR0 + INC0

For the first channel (AE1):

S_{1i} = COS(AR2) AR2 = AR2 + INC2

where i is the time index.

The sinusoid is scaled by K

$$KS_{1i} = K_1 * S_{1i}$$

and then offset by $L_1 - T_1$ (Level - Threshold) to form a test parameter P_{1i}.

The modulating function value for the i^{th} sample is then given by

$$DM_{1i} = \begin{cases} T_1; & P_{1i} \leq 0 \\ KS_{1i} + L_i & P_{1i} > 0 \end{cases}$$

Similarly, for the second DEMON channel (AE0) and the second channel (AE1)

$$S_{2i} = \text{COS}(AR1) \qquad AR1 = AR1 + INC1$$
$$S_{2i} = \text{COS}(AR3) \qquad AR3 = AR3 + INC3$$
$$KS_{2i} = K_2 * S_{2i}$$
$$DM_{2i} = \begin{cases} T_2; & P_{2i} \leq 0 \\ KS_{2i} + L_2; & P_{2i} > 0 \end{cases}$$

Phase 2

For the first channel, the input white noise samples are passed through a two-pole recursive filter

$$Y_{1i} = A_1 * X_{1i} + B * Y_{1(i-1)} + C * Y_{1(i-2)}$$

where A_1 is a gain coefficient, B is equal to the TWOR filter coefficient and C is equal to the MRSQ coefficient. Similarly, for the second channel

$$Y_{2i} = A_2 * X_{2i} + B * Y_{2(i-1)} + C * Y_{2(i-2)}.$$

Phase 3

The filtered noise channels are multiplied by the modulating functions

$$D_{1i} = DM_{1i} * Y_{1i}$$
$$D_{2i} = DM_{2i} * Y_{2i}$$

and combined with the composite channel to form the final output

$$TG_i = TG_i + D_{1i} + D_{2i}.$$

IMPLEMENTATION

DEMON is implemented with a 20-instruction main loop, a 30-instruction preamble, and a 6-instruction clean-up. The preamble loads the parameters and filter states from Working Store (WS) via Address Registers AR6 and AR7, generates the first two points of each periodic modulating signal via Address Registers AR0 through AR3, loads the first pair of channel inputs via Address Register AR4, and loads the first pair of target data via Address Register AR5. Since the target output overwrites the target input, a dummy read of -1 on AR5 is executed. Each pass through the main loop produces two pairs of filtered Y's which are multiplied by the corresponding modified modulating signals. The products are then added to the target signal to produce a composite output target signal. The cleanup sequence returns the final filter states to Working Store and stops. The main loop is

controlled by BNZF.

SCALING

See Table 9-1.

ARITHMETIC ELEMENT CONTROLLER (AEC) REGISTERS

See Table 9-2.

ADDRESS REGISTERS

AR0 - Phase address for first DEMON channel (AE0).
AR1 - Phase address for second DEMON channel (AE0).
AR2 - Phase address for first DEMON channel (AE1).
AR3 - Phase address for second DEMON channel (AE1).
AR4 - The increment code is set to 1. The address field is initialized to the beginning of the input X array.
AR5 - The increment code is set to 1. The address field is initialized to the beginning of the input TG array.
AR6 - The increment code is set to 1. The address field is initialized to the beginning of the parameters array.
AR7 - The increment code is set to 1. The address field is initialized to the beginning of the filter states array.
INC0 - Frequency increment for the first DEMON channel (AE0).
INC1 - Frequency increment for the second DEMON channel (AE0).
INC2 - Frequency increment for the first DEMON channel (AE1).
INC3 - Frequency increment for the second DEMON channel (AE1).
INC4, 6, 7 - Wrap code must be set to 1024.
INC5 - Wrap code - 1024, increment field - +2.

BRANCH REGISTERS

BNZF - Branch and Count register controlling the number of executions of the main loop. Count and Reset fields are set to N/2-1, where N is the number of elements in the target array. The Branch Address is DMNLP.

SINE/COSINE DESTINATION REGISTERS

SCD01 - Destine Cosine only to AE0.
SCD23 - Destine Cosine only to AE1.

DECIMATE REGISTER

Not applicable.

SCALE FACTOR REGISTERS

SFOH - Left prescaler - True/Right 4, center and right prescalers - True/Direct, postscaler - True/Left 4.

SFOL - Left and center prescalers - True/Right 2, right prescaler - True/Direct, postscaler - True/Left 4.

SF1H - Left and center prescalers - True/Right 2, right prescaler - True/Direct, postscaler - True/Left 2.

SF1L - All - True/Direct.

ARITHMETIC ELEMENT REGISTER MAP

See Table 9-3.

WORKING STORE MAP

See Table 9-4.

PROGRAM CODING CHART

See Table 9-5.

TABLE 9-1 DEMON AP MICROPROGRAM SCALING

LABEL	ADDER IN	MULTIPLIER IN	OUTPUT
INPUTS:			
A1, A2		H12	
K1, K2		H12	
Y11, Y12		H12	
Y21, Y22		H12	
X11, X12		H13	
X21, X22		H13	
S		H14	
B, C		H14	
TG1, TG2	H7		
L1, T1	H12		
L2, T2	H12		
INTERMEDIATE RESULTS:			
AX1, AX2	F9		
BY'	F10		
CY1", CY2"	F10		
D1, D2	F8		
KS	F10		
Y		H12	
DM	H12		
P	F12		
OUTPUTS:			
Y11, Y12			H12
Y21, Y22			H12
TG1, TG2			H7

TABLE 9-2 DEMON AEC REGISTER MAP

Incr Code Address Register

	0 3	4 15
0		ANGLE Ø1
1		ANGLE Ø2
2		ANGLE 11
3		ANGLE 12
4	1	X1, X2
5	1	T
6	1	Parameters
7	1	States

WS Wrap Increment Register

	0 3	4 15
0		FREQ Ø1
1		FREQ Ø2
2		FREQ 11
3		FREQ 12
4	Ø	
5	Ø	+2
6	Ø	
7	Ø	

Memory Address Register

DEMON

Sin/Cos Destination Register

	AE0	AE1	AE2	AE3		
0	1	Ø	Ø	Ø		
1	1	Ø	Ø	Ø		
2	Ø	Ø	1	Ø		
3	Ø	Ø	1	Ø		

Decimate Register

Count	Reset

Comments

BR/BCR Unconditional/ Conditional Branch

Branch Address

	0 15
0	
1	
2	
3	
4	
5	
6	
7	

BNZ Branch and Count Registers

	Count 0 7	Reset 8 15	Spare 16 21	Branch Address 22 31
8				
9				
A				
B				
C				
D				
E				
F	FF	FF		DMNLP

TABLE 9-3 DEMON AE REGISTER MAP

MLR

ØH	K1	A1	ØL
1H	K2	A2	1L
2H	Y11	Y12	2L
3H	Y21	Y22	3L

MRR

ØH	S	DM	ØL
1H	B	C	1L
2H	X11	X_{12}	2L
3H	X21	X22	3L

ALIR

ØH	B | Y'		ØL
1H	TG1	TG2	1L
2H	L1	T1	2L
3H	L2	T2	3L

ACIR

ØH	C | Y1"	ØL
1H	C | Y2"	1L
2H	D | 1	2L
3H	DUMMY	3L

ARIR

Ø	AX1
1	AX2
2	KS
3	D2

AEOR

ØH	Y11	Y12	ØL
1H	Y21	Y22	1L
2H	TG1	TG2	2L
3H			3L

SCALE FACTOR REGISTERS

AEØ

ØH	3 | 0 | Ø | 3	2 | 2 | Ø | 3	ØL
1H	2 | 2 | Ø | 2	Ø | Ø | Ø | Ø	1L
2H			2L
3H			3L

AE1

ØH	3 | Ø | Ø | 3	2 | 2 | Ø | 3	ØL
1H	2 | 2 | Ø | 2	Ø | Ø | Ø | Ø	1L
2H			2L
3H			3L

ALOR

Ø	Y11
1	Y21
2	Y12
3	Y22

ACOR

Ø	T1
1	T2
2	
3	

AROR

Ø	P
1	Ø
2	
3	

TABLE 9-4 DEMON WORKING STORE MAP

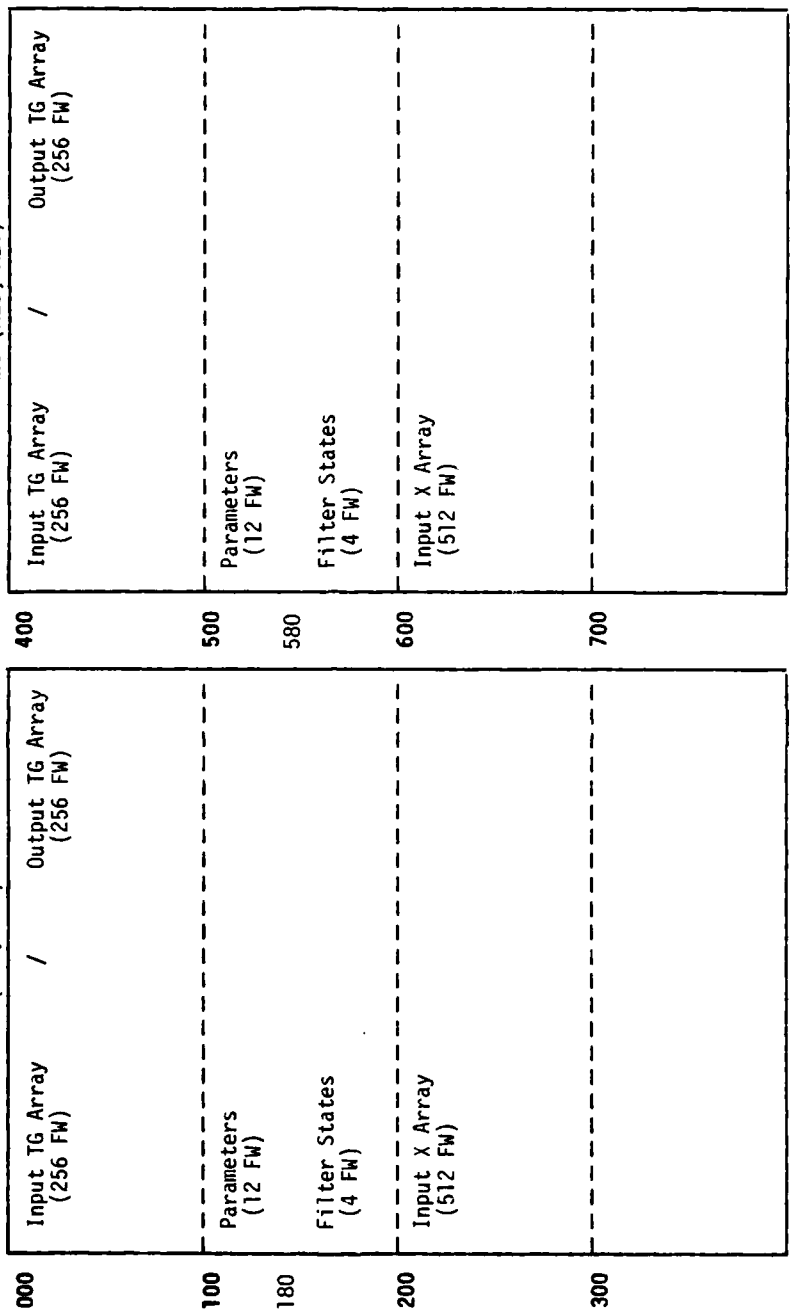


TABLE 9-5 DEMON PROGRAM CODING CHART

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
	DEMON		AR6 +0			NOP								
			READ L1,T1											
			AR6 +1			NOP								
			READ K1,A1				L1,T1 2							
			AR6 +1							Pass T1 1L				
			READ B1,C1	X1,A1 0								T1 0		
			AR6 +1											
			READ L2,T2		B1,C1 1									
			AR4 +0											
		S(01)	READ X11,X12				L2,T2 3							
			AR7 +0							Pass T2 1L				
		S(11)	READ Y11,Y12		X11,X12 2							T2 1		
			AR6 +1											
			READ K2,A2	Y11,Y12 2	S11 ØH									
			AR4 +1											
		S(02)	READ X21,X22	K2,A2 1	S11 ØH	AX11 = A1=X11								
			AR7 +1			CY11 = C1*Y11								
		S(12)	READ Y21,Y22		X21,X22 3	BY12 = B*Y12			AX11 Ø					
						KS11 = K1*S11		CY11 Ø						
				Y21,Y22	S21 ØH		BY12 Ø							
			AR5 +0						KS11 2					
		S(01)	READ TG1,TG2		S21 ØH	CY21 = C1*Y21								

TABLE 9-5 (CONTINUED)

#	LABEL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
			AR5 -1			KS21 = K2*S21				P11 = L1-T1+KS11 1H			SETCS Ø P11	
		S(11)	READ DUMMY			BY22 = B*Y22	TG1,TG2 1	CY21 1		Y11 = BY12+CY11 +AX11 ØL				
						AX21 = A2*X21			KS21 2	DM11 = T1+(P11)C 1L				
		S(02)		Y11 2H	S12 ØH	CY12 = C*Y12	BY22 Ø	DUMMY 3		R	Y11 Ø			
					DM11 ØL	CY22 = C*Y22			AX21 1	P21 = L2-T2+KS21 1H				
		S(12)			S12 ØH	D11 = DM11*Y11		CY12 Ø		Y21 = BY22+CY21 +AX21 ØL			SETCS Ø P21	
						KS12 = K1*S12		CY22 1		DM21=T2+(P21)C 1L				
				Y21 3H	S22 ØH	BY11 = B*Y11		D11 2		PASS Y11 ØH R	Y21 1			
	DMNLP		AR4 +1		DM21 ØL	AX12 = A1*X12			KS12 2					
		S(01)	READ X11,X12		S22 ØH	D21 = DM21*Y21	BY11 Ø	•		P12= L1-T1+KS12 1H				Y11 ØH
						KS22 = K2*S22			AX12 Ø	PASS Y21 ØH			SETCS Ø P12	
		S(11)			X11,XT2 2	BY21 = B*Y21			D21 3	Y12 = BY11+CY12 +AX12 ØL				
						AX22 = A2*X22			KS22 2	DM12 = T1+(P12)C 1L				Y21 1H
		S(02)		Y12 2L	S11 ØH	CY11 = C*Y11	BY21 Ø			P22 = L2-T2+KS22 1H R	Y12 2			
					DM12 ØL	CY21 = C*Y21			AX22 1	TG1 = TG1+D11+D21 1L			SETCS Ø P22	
		S(12)			S11 ØH	D12 = DM12*Y12		CY11 Ø		Y22 = BY21+CY22 +AX22 ØL				
						KS11 = K1*S11		CY21 1		DM22 = T2+(P22)C 1L R				TG1 2H
				Y22 3L	S21 ØH	BY12 = B*Y12		D12 2		PASS Y12 ØH R	Y22 3			
			AR4 +1		DM22 ØL	AX11 = A1*X11			KS11 2					
		S(01)	READ X21,X22		S21 ØH	D22 = DM22*Y22	BY12 Ø			P11 = L1-T1+KS11 1H				Y12 ØL
						KS21 = K2*S21			AX11 Ø	PASS Y22 ØH			SETCS Ø P11	
		S(11)			X21 X22 3	BY22 = B*Y22			D22 3	Y11 = BY12+CY11 +AX11 ØL				

TABLE 9-5 (CONTINUED)

#	LAREL BRANCH	SIN COS	READ WRITE	MLR	MRR	MULTIPLY	ALIR	ACIR	ARIR	ADD	ALOR	ACOR	AROR	AEOR
			AR5 +2			AX21 = A2*X21			KS21 2	DM11 = T1+(P11)C 1L				Y22 1L
		S(02)	READ TG1,TG2	Y11 2H	S12 ØH	CY12 = C*Y12	BY22 Ø			P21 =L2-T1+KS21 1H R	Y11 Ø			
					DM11 ØL	CY22 = C*Y22			AX21 1	TG2 = TG2+D12+D22 1L			SETCS P21 Ø	
	BNZF	S(12)			S12 ØH	D11 = DM11*Y11	TG1,TG2 1	CY12 Ø		BY22+CY21 Y21 = +AX21 ØL				
			AR5 -1			KS12 = K1*S12		CY22 1		DM21=T2+(P21)C 1L R				TG2 2L
			WRITE TG1,TG2	Y21 3H	S22 ØH	BY11 = B*Y11		D11 2		PASS Y11 ØH R	Y21 1			
			AR7 -1			NOP								
			WRITE Y11,Y12											
			AR7 +1			NOP								
			WRITE Y21,Y22											
						NOP								
						STOP								

APPENDIX A

AP MICROPROGRAM SOURCE LISTINGS

```
      BBBBBBBBB    BBBBBBBBB      BBBBBBBBB     BB       BB    BBBBBBBBB
      BBBBBBBBBBB  BBBBBBBBBBB    BBBBBBBBBB    BBBB   BBBB    BBBBBBBBBB
      BB       BB  BB       BB    BB       BB   BB BBBB BB    BB       BB
      BB       BB  BB       BB    BB       BB   BB  BB  BB    BB       BB
      BBBBBBBBB    BBBBBBBBB      BB       BB   BB  BB  BB    BBBBBBBBB
      BBBBBBBBB    BBBBBBBBB      BB       BB   BB      BB    BBBBBBBBB
      BB       BB  BB       BB    BBBBBBBBBBB   BB      BB    BB
      BB       BB  BB       BB    BB       BB   BB      BB    BB
      BBBBBBBBBBB  BBBBBBBBBBB    BB       BB   BB      BB    BB
      BBBBBBBBB    BBBBBBBBB      BB       BB   BB      BB    BB

*COMDECK BBAMP
*       BB GENERATION 12/05/75 RHD
*
*       18 STEP PREAMBLE
BBAMP   AECW    LOG=0                   M1 = NOT 0                      A
        AECW2   RDWA=MLR.2.ALIR.2.AR.1.  READ AV1, COUNT1               *B
                TOMLR.0H                M1
        AECW    LOG=0                   M1 = NOT 0                      0
        AECW2   RDWB=ALIR.3.AR.1.AAR.   READ DV1                        *1
                TOMRR.0H                M1
        AECW    SF.0H                   NOP                             2
        AECW2   RDWB=MRR.0.AR.0.        READ A1.A2                      *3
                MLR.0H.MRR.0H           P1 = M1*M1
        AECW    ALIR.3.SF.0H            DAV = DV1                       4
        AECW2   TOACIR.0.TOAROR.0       P1. DAV                         5
        AECW    MLR.2H.MRR.0H.          VA1 = AV1*A1                    *6
                ALIR.2H.AROR.0.SF.0H    CAV2 = AV1 + DAV
        AECW2   RDWB=MLR.3.AR.3.SAR.    BACK UP AR3                     *7
                ACIR.0L.SF.0H.          P1 = LOW PART                   *
                TOACOR.0.TOMLR.0L       CAV2
        AECW    ALIR.2LC.SF.0H.         CT = -COUNT                     *8
                TOARIR.0.               VA1                             *
                TOAROR.1.ROUND          P1. ROUND CAV2
        AECW2   RDWB=MRR.1.AR.2.        READ R1                         *9
                MLR.0L.MRR.0L.          VA2=CAV2.A2                     *
                TOACOR.1.               CT                              *
                ARIR.0.SF.0L            PASS VA1
        AECW    TOMLR.1H                VA1                             10
        AECW2   RDWB=MRR.2.AR.2.AAR.    READ R2                         *11
                TOARIR.0.ROUND          VA2.ROUND VA1
        AECW    MLR.1H.MRR.1H.          BHR1=VA1.RR1                    *12
                ARIR.0.SF.0L            PASS VA2
        AECW2   RDWB=MRR.0.AR.0.AAR.    READ A3.A4                      *13
                MLR.1H.MRR.1L.          BHI1=VA1.RI1                    *
                ACOR.0.AROR.0.SF.0H.    CAV3=CAV2+DAV                   *
                TOMLR.1L                VA2
        AECW    TOALIR.0.               BHR1                            *14
                TOACOR.0.TOMLR.0H.ROUND CAV3.ROUND VA2
        AECW2   MLR.1L.MRR.2H.          BHR2=VA2.RR2                    *15
                TOALIR.1.ROUND.         BHI1.ROUND CAV3                 *
                ACOR.0.AROR.0.SF.0H     CAV4=CAV3+DAV
        AECW    MLR.0H.MRR.0H.          VA3=CAV3.A3                     *16
                TOACOR.0.TOMLR.0L       CAV4
        AECW2   MLR.1L.MRR.2L.          BHI2=VA2.RI2                    *17
                TOACIR.0.TOAEOR.1.ROUND BHR2.B1=0.ROUND CAV4
```

```
*
*       10 INSTRUCTION MAIN LOOP WITH EMBEDDED
*       8 INSTRUCTION VARIATOR FUNCTION CHANGE
BB1     AECW    MLR.0L,MRR.0L,          VA4=CAV4,A4             *18
                TOARIR.0,               VA3                     *
                ACOR.1,AROR.1,SF.0H,    CT=CT+P1                *
                TOAEOR.1L,ROUND         BI1, ROUND BR1
        AECW2   RDWR=MRR.1,AR.2,AAR,    READ R3                 *19
                TOACIR.1,ROUND,         BHI2,ROUND BI1          *
                ARIR.0,SF.0L,           PASS VA3                *
                TOACOR.1,SETCS          CT,SET STATUS6
        AECW    TOARIR.0,               VA4                     *20
                ACIR.0,ALIR.0C,SF.0L,   BR2=BHR2-BHR1           *
                TOMLR.1H                VA3
        AECW2   RDWR=MRR.2,AR.2,AAR,    READ R4                 *21
                ARIR.0,SF.0L,           PASS VA4                *
                TOAEOR.2H,ROUND,        BR2,ROUND VA3           *
                BCR.0                   BRANCH ON ST6
        AECW    MLR.1H,MRR.1H,          BHR3=VA3,RR3            *22
                TOMLR.1L,ROUND,         VA4,ROUND BR2           *
                ALIR.1,ACIR.1C,SF.0L    BI2=BHI1-BHI2
        AECW2   RDWR=MRR.0,AR.0,AAR,    READ A5,A6              *23
                MLR.1H,MRR.1L,          BHI3=VA3,RI3            *
                TOAEOR.2L,ROUND,        BI2,ROUND VA4           *
                ACOR.0,AROR.0,SF.0H     CAV5=CAV4+DAV
*       VARIATOR FUNCTION CHANGE
*       EXECUTES ONLY WHEN CT = 0
        AECW    SF.0H,ROUND             NOP,ROUND BI2            24
        AECW2   RDWR=ALIR.2,AR.1,AAR    READ AV,COUNT            25
        AECW    SF.0H                   NOP                      26
        AECW2   RDWR=ALIR.3,AR.1,AAR    READ DV                  27
        AECW    SF.0H                   NOP                      28
        AECW2   ALIR.2LC,SF.0H          CT = -COUNT              29
        AECW    MLR.1H,MRR.1H,          BHR3=VA3*RR3            *30
                ALIR.3,SF.0H,           DAV = DV                *
                TOACOR.1                CT
        AECW2   MLR.1H,MRR.1L,          BHI3=VA3*RI3            *31
                ALIR.2H,SF.0H,          CAV = AV                *
                TOAROR.0                DV
*       CONTINUATION OF MAIN LOOP
BB2     AECW    TOALIR.0,ROUND,         BHR3, ROUND BI2         *32
                TOACOR.0,TOMLR.0H       CAV5
        AECW2   MLR.1L,MRR.2H,          BHR4=VA4,RR4            *33
                TOALIR.1,ROUND,         BHI3,ROUND CAV5         *
                ACOR.0,AROR.0,SF.0H,    CAV6=CAV5+DAV           *
                WRM=AEOR.1,AR.3,AAR,    WRITE B1                *
                BNZ.8                   BR TO BB1
        AECW    MLR.0H,MRR.0H,          VA5=CAV5,A5             *34
                TOACOR.0,TOMLR.0L,      CAV6                    *
                ALIR.0,ACIR.0C,SF.0L    BR3=BHR3-BHR2
        AECW2   MLR.1L,MRR.2L,          BHI4=VA4,RI4            *35
                TOAEOR.1H,ROUND,        BR3,ROUND CAV6          *
                ALIR.1C,ACIR.1,SF.0L,   BI3=BHI2-BHI3           *
                TOACIR.0,               BHR4                    *
                WRM=AEOR.2,AR.3,AAR     WRITE B2
*
*       OUTSIDE LOOP FOR SIZE GREATER THAN 512
        AECW    TOAEOR.1L,ROUND         BI3,ROUND BR3            36
        AECW2   BNZ.9,ROUND             BR TO BB1,ROUND BI3      37
        AECW    MLR.0H,MRR.0H           VA5=CAV5,A5              38
        AECW2   MLR.1L,MRR.2L,          BHI4=VA4,RI4            *39
                ALIR.1C,ACIR.1,SF.0L    BI3=BHI2-BHI3
*
        AECW    SF.0H                   NOP
        AECW2   BR.4                    BRANCH TO SM11
        AECW    SF.0H                   NOP
        AECW2   SF.0H                   NOP CONTINUE
```

```
 DDDDDDDDD     DDDDDDDD     DDDDDDDD    DD          DD       DD
 DDDDDDDDDD    DDDDDDDDD    DDDDDDDDD   DD          DDD      DD
 DD       DD   DD       D   DD       D  DD          DD D     DD
 DD       DD   DD           DD          DD          DD DD    DD
 DD       DD   DDDDDDDDD    DD          DD          DD  DD   DD
 DD       DD    DDDDDDDDD   DD          DD          DD   DD  DD
 DD       DD          DD   DD          DD          DD    DD DD
 DD       DD   D      DD   DD       D  DD          DD     D DD
 DDDDDDDDDD    DDDDDDDDD   DDDDDDDDD   DDDDDDDDDD  DD       DDD
 DDDDDDDDD      DDDDDDD    DDDDDDDD    DDDDDDDDD   DD        DD
```

```
*COMDECK DSCLN
DSCLN   AECW    SF.0H                           NOP                     -2
        AECW2   LOG=0,                          CREATE MINUS 1          *-1
                RDWA=AR.2,AR.5,                 READ FAMA               *
                SCMR.2H=AR.1                    GET (5( PARAMETER
        AECW    LOG=0,TOMRR.0H,TOARDR.0         DES M1                   0
        AECW2   RDWB=MLR.0,AR.5,AAR,            READ FS,A               *1
                TOMLR.1H                        DES M1
        AECW    SF.0H                           NOP                      2
        AECW2   RDWA=ACIR.1,AR.6,               READ CODP               *3
                SCMR.2H=AR.0,                   GET (7( PARAMETER       *
                MLR.1H,MRR.0H                   P1=M1*M1
        AECW    SF.0H                           NOP                      4
        AECW2   RDWB=MLR.3,AR.4,TOALIR.0        READ SL,P,DES P1         5
        AECW    ACIR.1,SF.0H                    PASS CODP                6
        AECW2   ALIR.0L,SF.0H,TOALOR.0,         P1=LO(P1), DES CD4      *7
                RDWB=MRR.0,AR.2                 READ FM
        AECW    ALIR.0L,SF.0H,TOMLR.1H,TOACOR.3 P1=LO(P1),DES P1         8
        AECW2   TOALOR.1,TOMRR.3L,              DES P1, P1              *9
                ACOR.3,AROR.0,SF.3L,            LM=M1+P1(R2)            *
                RDWA=ACIR.0,AR.4,AAR            READ CD0
        AECW    MLR.1H,MRR.2H,                  7=7*P1                  *A
                TOACOR.3                        DES LM
        AECW2   MLR.1H,MRR.2L,                  5=5*P1                  *B
                ALOR.0,ACOR.3,LOG=N             D4=CODP AND LM
        AECW    TOACIR.2,TOARDR.0,              DES 7,DES D4            *C
                MLR.0H,MRR.0H,                  FSM=FS*FM               *
                ACIR.0,SF.0H                    PASS CD0
        AECW2   MLR.3H,MRR.3L,TOACIR.3,         SL=P1*SL, DES 5         *D
                ACIR.2L,SF.2H,TOALOR.3,         7=LO(7)*2,DES CD0       *
                RDWB=MRR.1,AR.3                 READ AM
        AECW    TOALIR.2,                       DES FSM                 *E
                ACIR.3L,SF.0H,TOALOR.2          5=LO(5), DES 7
        AECW2   TOACIR.2,TOACOR.2,              DES SL, 5               *F
                ALIR.2,SF.1H                    FSM4=FSM(R4)
        AECW    TOARDR.1                        DES FSM4                 G
        AECW2   MLR.3L,MRR.3L,                  PP=P*P1                 *H
                ACIR.0,ARDR.1,SF.1H             NCD4=(CD0+FSM4)(R2)
DSCLP   AECW    TOARDR.2,TOMLR.1L,              DES NCD4,NCD            *10
                ALOR.3,ACIR.2,SF.0H             CD0=CD0+SL
        AECW2   TOAFOR.1,                       DES CD0                 *11
                ACIR.1H,ARDR.0,SF.3L,           CDP4=M(CODP)+D4         *
                RDWB=AR.0,AR.6,TOALIR.3         READ CD, DES PP
        AECW    ACOR.3,ARDR.2,LOG=N,            OD=CD AND LM            *12
                TOACOR.0,                       DES CDP4                *
```

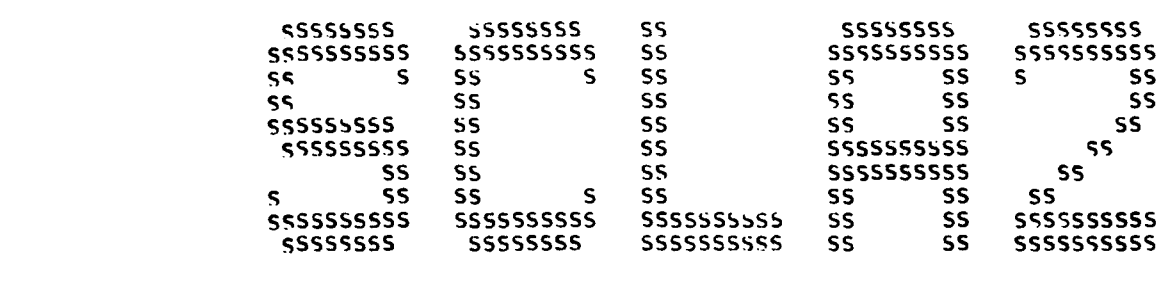
```
        MLR.1L.MRR.3L                    NC=P1*NCD
AECW2   ALIR.3.ACOR.0C.AROR.0C.SF.0L.    CC=PP-CDP4-D4          *13
        TOAFOR.2.                        DES 0D                 *
        WRM=AEOR.1.AR.4                  WRITE CD0
AECW    AROR.1.ACIR.0.SF.0H.             C=CD0+FSM4             *14
        TOALIR.0.TOACOR.1                DES NC. CC
AECW2   ALIR.0L.ACOR.1.AROR.2.SF.3H.     P=L(NC)/2+CC+NCD4      *15
        RDWR=AR.2.AR.4.SAR.              READ SL. P             *
        SCMR.0H=AR.1.TOAEOR.2H           COS.SIN (D), DES C
AECW    ACIR.2L.SF.0H.TOAEOR.3           SL=LO(SL). DES P        16
AECW2   ALOR.2.SF.0H.TOAEOR.3H.          PASS 7. DES SL         *17
        WRM=AEOR.2.AR.6                  WRITE CD
AECW    MLR.0L.MRR.1H.                   A=A*AM                 *18
        ACOR.2.SF.0H.TOMLR.3H            PASS 5. DES 7
AECW2   TOMLR.3L.                        DES 5                  *19
        WRM=AEOR.3.AR.4.                 WRITE SL. P            *
        SCMR.1H=AR.3                     COS.SIN (P)
AECW    MLR.3H.MRR.0H.TOACIR.0           7*C. DES A              1A
AECW2   MLR.1H.MRR.0H.                   P1*C=C                 *1B
        ACIR.0.SF.1L.                    PASS A                 *
        RDWR=ALIR.0.AR.0.AIRO            READ BM2
AECW    MLR.1H.MRR.0L.TOACIR.0.          S=P1*S. DES 7C         *1C
        TOMLR.0L                         DES A
AECW2   MLR.3L.MRR.0L.TOACIR.1.          S*S. DES C             *1D
        ACIR.0A.SF.0H.ROUND.             PASS 7C. RND A         *
        RDWR=ALIR.1.AR.0.AAR             READ BM1
AECW    MLR.0L.MRR.1H.TOACIR.0.          A*K. DES S             *1E
        ACIR.1LA.SF.0H.TOMLR.2H          C=LO(C). DES 7C
AECW2   ACIR.0LA.SF.0H.ROUND.            S=LO(S). RND 7C        *1F
        TOMLR.3H.TOACIR.1.               DES C. 5S              *
        RDWR=ALIR.2.AR.0.AAR             READ B0
AECW    MLR.0L.MRR.1L.TOACIR.0.          A*Z. DES AK            *20
        ACIR.1A.SF.0H.TOMLR.3L           PASS 5S. DES S
AECW2   ACIR.0.SF.2L.TOMLR.2L.           PASS AK. DES 5S        *21
        RDWR=ALIR.3.AR.0.AAR             READ BP1
AECW    MLR.3H.MRR.0L.TOACIR.0.          C*S. DES AZ            *22
        TOACOR.0.TOMRR.2H.ROUND          DES AK. RND 5S
AECW2   MLR.2L.MRR.0H.ROUND.             5S*C. ROUND AK         *23
        ACIR.0.SF.2L.                    PASS AZ                *
        RDWR=AR.2.AR.5.AIRE              READ FMA. AMA
AECW    MLR.2H.MRR.2H.TOACIR.0.          7C*AK. DES CS          *24
        TOACOR.1.TOMRR.2L                DES AZ
AECW2   MLR.3L.MRR.0L.TOACIR.1.          S*S. DES 5SC           *25
        ACIR.0A.SF.2L.ROUND.             PASS CS. RND AZ        *
        RDWR=MLR.0.AR.5.AAR              READ FS. A
AECW    MLR.2H.MRR.2L.TOACIR.0.          7C*AZ. DES 7CAK        *26
        ACIR.1.SF.2H.TOMRR.3H            PASS 5SC. DES CS
AECW2   MLR.3L.MRR.2H.TOACIR.1.          AK*S. DES 5S           *27
        ACIR.0.SF.2L.ROUND.              PASS 7CAK. RND CS      *
        TOMLR.1L.                        DES 5SC                *
        RDWR=MRR.0.AR.2                  READ FM
AECW    MLR.3L.MRR.2L.TOACIR.0.          AZ*S. DES 7CAZ         *28
        ACIR.1A.SF.2L.TOMLR.2H.ROUND     PASS 5S.DES 7CAK.R 5SC
AECW2   MLR.3H.MRR.2H.TOARIR.0.          AK*C. DES ASK          *29
        ACIR.0.SF.2L.TOMRR.3L.ROUND.     PAS 7CAZ.DES 5S.R 7CAK*
        RDWR=MRR.1.AR.3                  READ AM
AECW    MLR.2H.MRR.3H.TOARIR.1.          7CAK*CS. DES ASZ       *2A
        ACOR.0C.ARIR.0.SF.2H.            T1=(ASK-AK)/2          *
```

```
        TOMLR,2L,ROUND                  DES 7CAZ, RND SS
AECW2   MLR,3H,MRR,2L,TOARIR,2,         AZ*C, DES ACK          *28
        ACOR,1C,ARIR,1,SF,2H,           T2=(ASZ-AZ)/2          *
        TOAROR,0,ROUND,                 DES T1, RND 7CAZ       *
        RDWR=MLR,3,AR,4,AIRE            READ SL, P
AECW    MLR,2L,MRR,3H,TOACIR,0,         7CAZ*CS,DES 7CCSAK     *2C
        ACOR,0,ARIR,2C,SF,2H,           T3=(AK-ACK)/2          *
        TOAROR,1                        DES T2
AECW2   ALIR,0H,ACIR,0C,SF,0H,          M2R=BM2R-7CCSAK        *2D
        TOARIR,3,TOAROR,2,              DES ACZ, T3            *
        RDWR=ACIR,0,AR,4,AAR            READ CD0
AECW    MLR,1L,MRR,2H,TOACIR,1,         5SC*AK,DES 7CCSAZ      *2E
        ACOR,1,ARIR,3C,SF,2H,           T4=(AZ-ACZ)/2          *
        TOAEOR,0H                       DES M2R
AECW2   ALIR,0L,ACIR,1C,SF,0H,          M2I=BM2I-7CCSAZ        *2F
        TOAROR,3,ROUND,                 DES T4,RND M2R         *
        RDWR=ACIR,1,AR,6,AAR            READ CD
AECW    MLR,1L,MRR,2L,TOACIR,2,         5SC*AZ, DES 5SCAK      *30
        ACIR,0,SF,0H,TOAEOR,0L          PASS CD0, DES M2I
AECW2   ALIR,1H,ACIR,2C,AROR,0,         M1R=BM1R-5SCAK+T1      *31
        SF,2H,TOALOR,3,ROUND,           DES CD0, RND M2I       *
        RDWR=ALIR,0,AR,0,AAR            READ BP2
AECW    ACIR,1,SF,0H,TOACIR,3,TOAEOR,1H PASS CD,DES 5SCAZ,M1R  32
AECW2   ALIR,1L,ACIR,3C,AROR,1,SF,2H,   M1I=BM1I-5SCAZ+T2      *33
        TOALOR,0,ROUND,                 DES CD4,RND M1R        *
        RDWR=ALIR,1,AR,0,AAR            READ BP3
AECW    ALIR,2H,ACOR,0,ARIR,2,SF,2H,    0R=B0R+AK/2+ACK/2      *34
        TOAEOR,1L                       DES M1I
AECW2   ALIR,2L,ACOR,1,ARIR,3,SF,2H,    0I=B0I+AZ/2+ACZ/2      *35
        TOAEOR,2H,ROUND,                DES 0R, RND M1I        *
        WRM=AEOR,0,AR,0,AIRE            WRITE M2
AECW    ALIR,3H,ACOR,0C,ARIR,0C,SF,2H,  P1R=BP1R-AK/2-ASK/2    *36
        TOAEOR,2L,ROUND                 DES 0I, RND 0R
AECW2   ALIR,3L,ACOR,1C,ARIR,1C,SF,2H,  P1I=BP1I-AZ/2-ASZ/2    *37
        TOAEOR,3H,ROUND,                DES P1R, RND 0I        *
        WRM=AEOR,1,AR,0,AAR             WRITE M1
AECW    MLR,2H,MRR,3L,                  7CAK*SS                *38
        ALIR,0H,ACIR,2,AROR,2,SF,2H,    P2R=BP2R+5SCAK+T3      *
        TOAEOR,3L,ROUND                 DES P1I, RND P1R
AECW2   MLR,2L,MRR,3L,                  7CAZ*SS                *39
        ALIR,0L,ACIR,3,AROR,3,SF,2H,    P2I=BP2I+5SCAZ+T4      *
        TOAEOR,0H,ROUND,                DES P2R, RND P1I       *
        WRM=AEOR,2,AR,0,AAR             WRITE 0
AECW    ALOR,1,SF,0H,TOACIR,2,          PASS P1, DES 7SSCAK    *3A
        MLR,0H,MRR,0H,                  FSM=FS*FM              *
        TOAEOR,0L,ROUND                 DES P2I, RND P2R
AECW2   ALIR,1H,ACIR,2,SF,0H,           P3R=BP3R+7SSCAK        *3B
        TOACIR,3,TOMRR,3L,ROUND,        DES 7SSCAZ,P1,RND P2I  *
        WRM=AEOR,3,AR,0,AAR             WRITE P1
AECW    ALIR,1L,ACIR,3,SF,0H,           P3I=BP3I+7SSCAZ        *3C
        TOALIR,2,TOAEOR,1H              DES FSM,P3R
AECW2   MLR,3H,MRR,3L,                  SL=P1*SL               *3D
        TOAEOR,1L,ROUND,                DES P3I, RND P3R       *
        WRM=AEOR,0,AR,0,AAR,            WRITE P2               *
        ALIR,2,SF,1H,                   FSM4=FSM(R4)           *
        BNZ,A                           BR ON CT TO USCLP
AECW    ALOR,0,ACOR,3,LOG=N,            D4=CODP AND LM         *3E
        TOAROR,1,ROUND                  DES FSM4, RND P3I
AECW2   TOACIR,2,TOAROR,0,              DES SL, D4             *3F
        ACIR,0,AROR,1,SF,1H,            NCD4=CD0+FSM4          *
        MLR,3L,MRR,3L,                  PP=P1*P                *
        WRM=AEOR,1,AR,0,AAR             WRITE P3
AECW    SF,0H                           NOP                     40
AECW2   STOP                                                    41
```

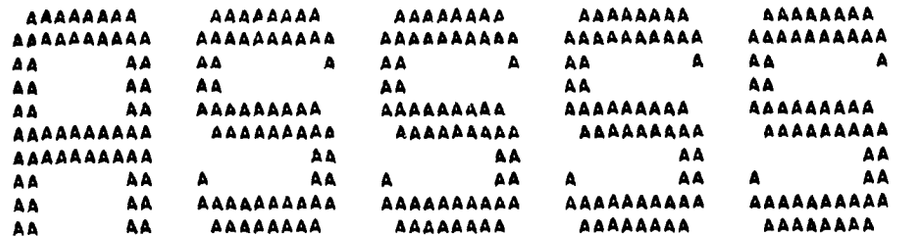
```
 SSSSSSSS    SSSSSSSSS    SSSSSSSSSS   SSSSSSSSSS   SSSSSSSSSS
SSSSSSSSSS   SSSSSSSSSS   SSSSSSSSSS   SSSSSSSSSS   SSSSSSSSSS
SS       S   SS      SS   SS           SS               SS
SS           SS      SS   SS           SS               SS
SSSSSSSSS    SSSSSSSSSS   SSSSSS       SSSSSS           SS
 SSSSSSSSS   SSSSSSSSS    SSSSSS       SSSSSS           SS
        SS   SS   SS      SS           SS               SS
S       SS   SS    SS     SS           SS               SS
SSSSSSSSSS   SS     SS    SS           SS               SS
 SSSSSSSS    SS      SS   SS           SS               SS
```

```
*COMDECK SRFFT
SRFFT   AECW    LOG=0                           CREATE ALL ONES         0
        AECW2   TOMLR.2H.                       M1=MINUS 1             *1
                RDWR=ALIR.1.AR.2                READ G0
        AECW    LOG=0                           NEED TWO M1'S           2
        AECW2   TOMRR.1H.                       M1                     *3
                RDWA=MLR.0.ALIR.2.AR.2.AAR.     READ G1                *
                SCMR.0=AR.0INC                  GET COS.SIN (INC)
        AECW    ALIR.1H.SF.0H                   FR0=GR0                 4
        AECW2   ALIR.1H.SF.0H.                  FI0=GR0                *5
                RDWR=ACIR.0.AR.3.               READ GN1               *
                TOAEOR.0H                       FR0
        AECW    TOAFOR.0L                       FI0                     6
        AECW2   RDWA=MLR.3.AR.5.AAR             DUMMY READ FOR FN       7
        AECW    ALIR.2L.ACIR.0L.SF.0L           CI1=(GIN1+GI1)/2        8
        AECW2   TOMLR.1L.                       CI1                    *9
                ALIR.2H.ACIR.0H.SF.0L.          AR1=(GRN1+GR1)/2       *
                RDWA=MLR.0.ALIR.0.AR.2.AAR      READ G2
        AECW    TOACOR.0.ROUND.                 AR1. ROUND CI1         *A
                ALIR.2HC.ACIR.0H.SF.0L          CR1=(GRN1-GR1)/2
        AECW2   TOMLR.1H.                       CR1                    *B
                MLR.1L.MRR.0H.                  CCI1=C*CI1             *
                ALIR.2LC.ACIR.0L.SF.0L.         AI1=(GIN1-GI1)/2       *
                RDWA=MRR.2.ACIR.0.AR.3.SAR      READ GN2
        AECW    TOACOR.1.ROUND.                 AI1. ROUND CR1         *C
                MLR.1L.MRR.0L                   SCI1=S*CI1
        AECW2   TOARIR.3.                       CCR1                   *D
                MLR.1H.MRR.0L.                  SCR1=S*CR1             *
                WRM=AEOR.0.AR.4.                WRITE F0               *
                SCMR.0=AR.0INC                  COS.SIN (2*INC)
        AECW    TOALIR.3.                       SCI1                   *E
                MLR.1H.MRR.0H.                  CCR1=C*CR1             *
                ALIR.0HC.ACIR.0H.SF.0L          CR2=(GRN2-GR2)/2
        AECW2   TOALIR.2.TOMLR.1H.              SCR1.CR2               *F
                ALIR.0L.ACIR.0L.SF.0L.          CI2=(GIN2+GI2)/2       *
                MLR.0H.MRR.1H                   MGR2=M1*GR2
SRFLP   AECW    TOARIR.2.TOMLR.1L.ROUND.        CCR.CI.ROUND CR        *10
                MLR.2H.MRR.2H.                  MGRN=M1*GRN            *
                ALIR.2.ARIR.3.ACOR.0.SF.1H      FR=SCR+CCI+AR
        AECW2   TOALIR.1.TOAEOR.0H.ROUND.       MGR. FR. ROUND CI      *11
                MLR.2H.MRR.2L.                  MGIN=M1*GIN            *
                ALIR.3.ARIR.2C.ACOR.1.SF.1H.    FI=SCI-CCR+AI          *
                RDWA=MLR.0.ALIR.0.AR.2.AAR      READ G
        AECW    TOACIR.1.TOAEOR.0L.ROUND.       MGRN. FI. ROUND FR     *12
                MLR.0L.MRR.1H.                  MGI=M1*GI              *
```

```
        ALIR,2C,AMIR,3C,ACOR,0,SF,1H      FRN=-SCR-CCI+AR
AECW2   TOACIR,2,TOAEOR,1H,ROUND,         MGIN, FRN, ROUND FI    *13
        MLR,1L,MRR,0H,                    CCI=C*CI               *
        ALIR,3,ARIR,2C,ACOR,1C,SF,1H,     FIN=SCI-CCR-AI         *
        RDWA=MRR,2,ACIR,0,AR,3,SAR        READ GN
AECW    TOALIR,2,TOAEOR,1L,ROUND,         MGI,FIN,ROUND FRN      *14
        MLR,1L,MRR,0L,                    SCI=S*CI               *
        ALIR,1LC,ACIR,1LC,SF,0L           AR=(-MGR-MGRN)/2
AECW2   TOARIR,3,TOACOR,0,ROUND,          CCI,AR,ROUND FIN       *15
        MLR,1H,MRR,0L,                    SCR=S*CR               *
        ALIR,2L,ACIR,2LC,SF,0L,           AI=(MGI-MGIN)/2        *
        WRM=AEOR,0,AR,4,AAR,              WRITE F                *
        SCMR,0=AR,0INC,BNZ,B              COS,SIN,LOOP ON BNZB
AECW    TOALIR,3,TOACOR,1,                SCI,AI                 *16
        MLR,1H,MRR,0H,                    CCR=C*CR               *
        ALIR,0HC,ACIR,0H,SF,0L            CR=(GRN-GR)/2
AECW2   TOALIR,2,TOMLR,1H,                SCR,CR                 *17
        MLR,0H,MRR,1H,                    MGR=M1*GR              *
        ALIR,0L,ACIR,0L,SF,0L,            CI=(GIN+GI)/2          *
        WRM=AEOR,1,AR,5,SAR               WRITE FN
AECW    SF,0H                             NOP                     18
AECW2   BNZ,C                             OUTSIDE LOOP            19
AECW    MLR,1H,MRR,0H                     CCR=C*CR                1A
AECW2   MLR,0H,MRR,1H,                    MGR=M1*GR              *1B
        ALIR,0L,ACIR,0L,SF,0L             CI=(GIN+GI)/2
AECW    SF,0H                             NOP                     1C
AECW2   STOP                              STOP                    1D
```



```
*COMDECK SCLA2
SCLA2   AECW    SF.0H                       NOP                          0
        AECW2   RDWB=MLR.0,AR.2,            READ 1ST PAIR OF X(S        *1
                SCMR.0H=AR.0                GENERATE PARAMETER A
        AECW    SF.0H                       NOP                          2
        AECW2   RDWB=MLR.1,AR.3,            READ 1ST PAIR OF Y(S        *3
                SCMR.0L=AR.1                GENERATE PARAMETER B
        AECW    SF.0H                       NOP
        AECW2   SF.0H                       NOP
        AECW    MLR.0H,MRR.0H               AXH=A*XH                     4
        AECW2   MLR.0L,MRR.0H,              AXL=A*XL                    *5
                RDWB=MLR.2,AR.4,SAR         BACK UP Z POINTER
        AECW    TOALIR.0,                   AXH                         *6
                MLR.1H,MRR.0L               BYH=B*YH
        AECW2   TOALIR.1,                   AXL                         *7
                MLR.1L,MRR.0L,              BYL=B*YL                    *
                RDWB=MLR.0,AR.2,AAR         READ NEXT PAIR OF X(S
        AECW    TOACIR.0                    BYH                          8
        AECW2   RDWB=MLR.1,AR.3,AAR,        READ NEXT PAIR OF Y(X       *9
                TOACIR.1                    BYL
        AECW    MLR.0H,MRR.0H,              AXH=A*XH                    *A
                ALIR.0,ACIR.0,SF.0H         ZH=AXH+BYH
        AECW2   MLR.0L,MRR.0H,              AXL=A*XL                    *B
                ALIR.1,ACIR.1,SF.0H,        ZL=AXL+BYL                  *
                TOAEOR.0H                   ZH
SC2LP   AECW    MLR.1H,MRR.0L,              BYH=B*YH                    *C
                TOAEOR.0L,TOALIR.0          ZL, AXH
        AECW2   RDWB=MLR.0,AR.2,AAR,        READ NEXT PAIR OF X(S       *D
                MLR.1L,MRR.0L,              BYL=B*YL                    *
                TOALIR.1                    AXL
        AECW    TOACIR.0                    BYH                          E
        AECW2   RDWB=MLR.1,AR.3,AAR,        READ NEXT PAIR OF Y(S       *F
                TOACIR.1,BNZ.8              RYL, BRANCH TO SC2LP
        AECW    MLR.0H,MRR.0H,              AXH=A*XH                    *10
                ALIR.0H,ACIR.0H,SF.0H       ZH=AXH+BYH
        AECW2   WRM=AEOR.0,AR.4,AAR,        WRITE A PAIR OF Z(S         *11
                MLR.0L,MRR.0H,              AXL=A*XL                    *
                ALIR.1,ACIR.1,SF.0H,        ZL=AXL+BYL                  *
                TOAEOR.0H                   ZH
        AECW    SF.0H                       NOP                         12
        AECW2   BNZ.9                       OUTSIDE LOOP                13
        AECW    MLR.0H,MRR.0H               AXH=A*XH                    14
        AECW2   MLR.0L,MRR.0H,              AXL=A*XL                    *15
                ALIR.1,ACIR.1,SF.0H         ZL=AXL+BYL
        AECW    SF.0H                       NOP                         16

        AECW2   STOP                        STOP                        17
```



```
*COMDECK ASSSS
*       MLPTH - MULTIPATH RESAMPLE,SCALE AND ADD
*                                                   RHO 01/26/76
*       IN PHASE START SEQUENCE
MLPI    AECW    SF.0H                               NOP                     0
        AECW2   RDWA=MRR.1,ACIR.1,AR.0              READ A, -1              1
        AECW    SF.0H                               NOP                     2
        AECW2   RDWB=ALIR.2,AR.1                    READ F                  3
        AECW    TOALOR.0,SF.0H                      0 = INHIBIT             4
        AECW2   RDWA=MLR.0,ALIR.0,AR.2,             READ DHDL (1)          *5
                TOAROR.0,SF.0H                      0 TO 0C
        AECW    TOAROR.3,                           0 TO 1C                *6
                ALOR.0C,AROR.0C,ACIR.1LC,SF.0H      ONE=-1 -EPS
        AECW2   RDWA=MLR.0,ALIR.0,AR.2,AAR,         READ DHDL (2)          *7
                TOAROR.1,                           ONE TO 0C              *
                ALOR.0C,AROR.0C,ACIR.1LC,SF.0H      ONE=-1 -EPS
        AECW    MLR.0L,MRR.1H,                      AL = A*DL              *8
                TOAROR.2                            ONE TO 1C
        AECW2   RDWA=ALIR.3,AR.1,AAR                READ D                  9
        AECW    MLR.0H,MRR.1H,                      AH = A*DH              *10
                TOARIR.3,                           AL                     *
                ALIR.2C,AROR.1,SF.0H                F3 = ONE-F
        AECW2   ARIR.3,SF.0L,                       PASS AL                *11
                TOACOR.2,TOMLR.1L                   F3
        AECW    TOARIR.2,TOMRR.0L,                  AH,AL                  *12
                ALIR.2,SF.0H                        F1 = F
        AECW2   RDWA=ALIR.1,AR.3,                   READ SHSL(1)           *13
                ARIR.2,SF.0L,ROUND,                 PASS AH,ROUND AL       *
                TOACOR.0,TOMLR.1H,SETCS             F1,SET STATUS 6
        AECW    MLR.0L,MRR.1H,                      AL=A*DL                *14
                ALIR.3C,ACOR.0C,AROR.2TS,SF.0H,     F4=-F1-D+1C            *
                TOMRR.0H                            AH
        AECW2   RDWA=MLR.0,ALIR.0,AR.2,AAR,         READ DHDL(3)           *15
                MLR.1L,MRR.0L,ROUND,                3AL=F3*AL,ROUND AH     *
                TOACOR.3,TOMLR.2L                   F4
        AECW    MLR.1H,MRR.0H,                      1AH=F1*AH              *16
                TOARIR.3,                           AL                     *
                ALIR.3,ACOR.0,SF.0H                 F2=F1+D
        AECW2   RDWB=MLR.3,AR.4,SAR,                DUMMY READ             *17
                TOARIR.0,                           3AL                    *
                ARIR.3,SF.0L,SETCS,                 PASS AL,SET STATUS     *
                TOACOR.1,TOMLR.2H,                  F2                     *
                BNZ,A                               BRANCH TO MIPL0
        AECW    MLR.0H,MRR.1H,                      AH=A*DH                *18
                TOACIR.0,TOMRR.0L,                  1AH,AL                 *
```

```
                ALIR,3C,ACOR,1C,AROR,2TS,SF,0H   F3=-F2-D+1C
        AECW2   MLR,2L,MRR,0H,                   4AH=F4*AH                *19
                ALIR,1H,ACIR,0,ARIR,0,SF,1H,     0H=SH+1AH+3AL            *
                TOACOR,2,TOMLR,1L,ROUND,         F3,ROUND AL              *
                BCR,1                            BRANCH TOI002 ON F2
*
*
*       OUT PHASE START SEQUENCE
MLPO    AECW    SF,0H                            NOP                       20
        AECW2   RDWA=MRR,1,ACIR,1,AR,0,          READ A, -1               *21
                TOALOR,0,SF,0H                   0=INHIBIT
        AECW    SF,0H,TOAROR,0                   0TO 0C                    22
        AECW2   RDWA=MLR,0,ALIR,0,AR,2,          READ DHDL(1)             *23
                TOAROR,3                         0 TO 1C
        AECW    ALOR,0C,AROR,0C,ACIR,1LC,SF,0H   ONE=-1 -EPS               24
        AECW2   RDWB=ALIR,2,AR,1,                READ F                   *25
                ALOR,0C,AROR,0C,ACIR,1LC,SF,0H,  ONE=-1 -EPS              *
                TOAROR,1                         ONE TO 0C
        AECW    MLR,0H,MRR,1H,                   AH=A*DH                  *26
                TOAROR,2                         ONE TO 1C
        AECW2   RDWB=ALIR,3,AR,1,AAR             READ D                    27
        AECW    MLR,0L,MRR,1H,                   AL=A*DL                  *28
                TOARIR,2,                        AH                       *
                ALIR,2C,AROR,1,SF,0H             F4=ONE-F
        AECW2   RDWA=MLR,0,ALIR,0,AR,2,AAR,      READ DHDL(2)             *29
                ARIR,2,SF,0L,                    PASS AH                  *
                TOACOR,3,TOMLR,2L                F4
        AECW    TOARIR,3,TOMRR,0H,               AL,AH                    *30
                ALIR,2,SF,0H                     F2=F
        AECW2   RDWB=ACIR,3,AR,4,SAR,            DUMMY READ               *31
                ARIR,3,SF,0L,ROUND,              PASS AL,ROUND AH         *
                TOACOR,1,TOMLR,2H,SETCS          F2
        AECW    MLR,0H,MRR,1H,                   AH=A*DH                  *32
                ALIR,3C,ACOR,1C,AROR,2,SF,0H,    F3=-F2-D+ONE             *
                TOMRR,0L                         AL
        AECW2   RDWB=ALIR,1,AR,3,                READ SHSL(1)             *33
                MLR,2L,MRR,0H,ROUND,             4AH=F4*AH,ROUND AL       *
                TOACOR,2,TOMLR,1L                F3
        AECW    MLR,2H,MRR,0L,                   2AL=F2*AL                *34
                TOARIR,2,                        AH                       *
                ALIR,3,ACOR,1,SF,0H              F1=F2+D
        AECW2   RDWB=MLR,0,ALIR,0,AR,2,AAR,      READ DHDL(3)             *35
                TOARIR,1,                        4AH                      *
                ARIR,2,SF,0L,                    PASS AH                  *
                TOACOR,0,TOMLR,1H,SETCS,         F1                       *
                BNZ,B                            BRANCH TO MOPL0
        AECW    MLR,0L,MRR,1H,                   AL=A*DL                  *36
                TOACIR,1,                        2AL                      *
                ALIR,3C,ACOR,0C,AROR,2TS,SF,0H,  F4=-D-F1+1C              *
                TOMRR,0H                         AH
        AECW2   MLR,1L,MRR,0L,                   3AL=F3*AL                *37
                ALIR,1H,ACIR,1,ARIR,1,SF,1H,     0H=SH+2AL+4AH            *
                TOMLR,2L,TOACOR,3,ROUND,         F4,ROUND                 *
                BCR,4                            BRANCH TO OIO1 ON F1
*
*       INPHASE PROCESSING LOOP
MIPLP   AECW    MLR,0H,MRR,1H,                   AH=A*DH                  *38
                TOACIR,0,                        1AH                      *
```

```
                ALIR,3C,ACOR,1C,AROR,2TS,SF,0H,   F3=-D-F2+1C              *
                TOMRR,0L                          AL
          AECW2 MLR,2L,MRR,0H,                    4AH=F4*AH                *39
                ALIR,1H,ACIR,0,ARIR,0,SF,1H,      0H=SH+1AH+3AL            *
                TOACOR,2,TOMLR,1L,ROUND,          F3,ROUND AL              *
                WRM=AEOR,0,AR,4,AAR,              WRITE OUTPUT 0H0L        *
                BCR,1                             BRANCH ON F2 TO I002
MIPL0     AECW  MLR,2H,MRR,0L,                    2AL=F2*AL                *40
                TOARIR,2,                         AH                       *
                ALIR,3,ACOR,1,AROR,0TS,SF,0H,     F1=D+F2+0C               *
                TOAEOR,0H                         0H
          AECW2 RDWA=MLR,0,ALIR,0,AR,2,AAR,       READ 0H0L                *41
                TOARIR,1,ROUND,                   4AH,ROUND 0H             *
                ARIR,2,SF,0L,                     PASS AH                  *
                TOACOR,0,TOMLR,1H,SETCS           F1
MIPL2     AECW  MLR,0L,MRR,1H,                    AL=A*0L                  *42
                TOACIR,1,                         2AL                      *
                ALIR,3C,ACOR,0C,AROR,2TS,SF,0H,   F4=-D-F1+1C              *
                TOMRR,0H                          AH
          AECW2 RDWA=ALIR,1,AR,3,AAR,             READ SHSL                *43
                MLR,1L,MRR,0L,ROUND,              3AL=F3*AL,ROUND AH       *
                ALIR,1L,ACIR,1,ARIR,1,SF,1H,      0L=SL+2AL+4AH            *
                TOACOR,3,TOMLR,2L,                F4                       *
                BNZ,8                             BRANCH ON CT TO MIPLP
MIPL1     AECW  MLR,1H,MRR,0H,                    1AH=F1*AH                *44
                TOARIR,3,                         AL                       *
                ALIR,3,ACOR,0,AROR,0TS,SF,0H,     F2=D+F1+0C               *
                TOAEOR,0L                         0L
          AECW2 TOARIR,0,                         3AL                      *45
                ARIR,3,SF,0L,ROUND,               PASS AL,ROUND 0L         *
                TOACOR,1,TOMLR,2H,SETCS,          F2                       *
                BCR,0                             BRANCH ON F1 TO I001
          AECW  SF,0H                             NOP                       46
          AECW2 WRM=AEOR,0,AR,4,AAR,              WRITE LAST OUTPUT        *47
                BNZ,C                             BRANCH TO STOP
STOP      AECW  SF,0H                             NOP                       48
          AECW2 STOP                              STOP                      49
*
*         OUT PHASE PROCESSING LOOP
MOPLP     AECW  MLR,0L,MRR,1H,                    AL=A*0L                  *50
                TOACIR,1,                         2AL                      *
                ALIR,3C,ACOR,0C,AROR,2TS,SF,0H,   F4=-D-F1+1C              *
                TOMRR,0H                          AH
          AECW2 MLR,1L,MRR,0L,ROUND,              3AL=F3*AL,ROUND AH       *51
                ALIR,1H,ACIR,1,ARIR,1,SF,1H,      0H=SH+2AL+4AH            *
                TOACOR,3,TOMLR,2L,                F4                       *
                WRM=AEOR,0,AR,4,AAR,              WRITE OUTPUT 0           *
                BCR,4                             BRANCH TO 0I01 ON F1
MOPL0     AECW  MLR,1H,MRR,0H,                    1AH=F1*AH                *52
                TOARIR,3,                         AL                       *
                ALIR,3,ACOR,0,AROR,0,S,SF,0H,     F2=F1+D+0C               *
                TOAEOR,0H                         0H
          AECW2 TOARIR,0,ROUND,                   3AL,ROUND 0H             *53
                ARIR,3,SF,0L,                     PASS AL                  *
                TOACOR,1,TOMLR,2H,SETCS           F2
MOPL2     AECW  MLR,0H,MRR,1H,                    AH=A*0H                  *54
                TOACIR,0,                         1AH                      *
                ALIR,3C,ACOR,1C,AROR,2TS,SF,0H,   F3=-D-F2+1C              *
```

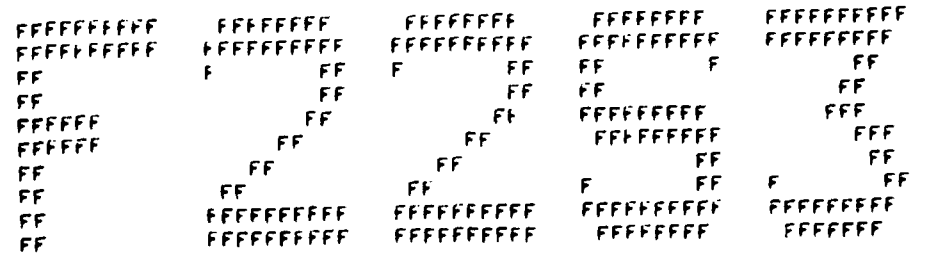
```
                TOMRR.0L                        AL
        AECW2   RDWA=ALIR.1,AH.3,AAR,           READ SHSL               *55
                MLR.2L,MRR.0H,ROUND,            4AH=F4*AH,ROUND AL      *
                ALIR.1L,ACIR.0,ARIR.0,SF.1H,    0L=SL+1AH+3AL           *
                TOACOR.2,TOMLR.1L,              F3                      *
                BNZ.9                           BRANCH ON CT TO MOPLP
MOPL1   AECW    MLR.2H,MRR.0L,                  2AL=F2*AL               *56
                TOARIR.2,                       AH                      *
                ALIR.3,ACOR.1,AROR.0TS,SF.0H,   F1=F2+0+0C              *
                TOAFOR.0L                       0L
        AECW2   RDWA=MLR.0,ALIR.0,AR.2,AAR,     READ DH0L               *57
                TOARIR.1,ROUND,                 4AH,ROUND 0L            *
                ARIR.2,SF.0L,                   PASS AH                 *
                TOACOR.0,TOMLR.1H,SETCS,        F1                      *
                BCR.5                           BRANCH TO 0102 ON F2
        AECW    SF.0H                           NOP                      58
        AECW2   WRM=AEOR.0,AR.4,AAR,            WRITE LAST OUTPUT       *59
                BNZ.C                           BRANCH TO STOP
        AECW    SF.0H                           NOP                      60
        AECW2   STOP                            STOP                     61
*
*       IN TO OUT TRANITIONS, 0 POSITIVE
PI001   AECW    MLR.1H,MRR.0L,                  1AL=F1*AL               *62
                TOARIR.2,                       AH                      *
                ALIR.1H,ARIR.3,SF.1H            0T=SH+AL
        AECW2   RDWA=MLR.0,ALIR.0,AR.2,AAR,     READ DH0L               *63
                ARIR.2,SF.0L,                   PASS AH                 *
                TOALOR.0                        0T
        AECW    MLR.0L,MRR.1H,                  AL=A*DL                 *64
                TOARIR.0,                       1AL                     *
                ACOR.2,SF.0H,                   F4=F3                   *
                TOMRR.0H                        AH
        AECW2   MLR.2L,MRR.0L,ROUND,            3AL=F4*AL,ROUND AH      *65
                ALOR.0,ACIR.0C,ARIR.0,SF.1H,    0H=0T-1AH+1AL           *
                TOACOR.3,TOMLR.2L,              F4                      *
                BR.3                            BRANCH TO MOPL2
        AECW    MLR.2H,MRR.0H,                  1AH=F2*AH               *66
                TOARIR.3,                       AL                      *
                ALIR.3,ACOR.1,SF.0H,            F2=F2+0                 *
                TOAFOR.0H                       0H
        AECW2   TOARIR.0,                       3AL                     *67
                ARIR.3,SF.0L,ROUND,             PASS AL,ROUND 0H        *
                TOACOR.1,TOMLR.2H               F2
*
PI002   AECW    MLR.0L,MRR.1H,                  AL=A*DL                 *68
                TOACIR.1,                       2AL                     *
                TOMRR.0H                        AH
        AECW2   ACOR.2,SF.0H,ROUND              F4=F3, ROUND AH          69
        AECW    MLR.2L,MRR.0H,                  4AH=F4*AH               *70
                TOARIR.3,TOACOR.3,TOMLR.2L,     AL, F4                  *
                ACOR.0,SF.0H                    F2=F1
        AECW2   ARIR.3,SF.0L,                   PASS AL                 *71
                TOACOR.1,TOMLR.2H,              F2                      *
                BR.7                            BRANCH TO MOPL1
        AECW    MLR.0H,MRR.1H,                  AH=A*0H                 *72
                TOARIR.1,                       4AH                     *
                ALIR.3C,ACOR.1C,AROR.1,SF.0H,   F3=-0-F2+ONE            *
                TOMRR.0L                        AL
```

```
         AECW2   RDWR=ALIR.1,AR.3,AAR,          READ SHSL              *73
                 MLR.2L,MRR.0H,ROUND,           4AH=F4*AH,ROUND AL     *
                 ALIR.1L,ACIR.1C,ARIR.1C,SF.1H, 0L=SL-2AL-4AH          *
                 TOACOR.2,TOMLR.1L,             F3                     *
                 BNZ,9                          BRANCH ON CT TO MOPLP
*
*
*        OUT TO IN TRANSITIONS, D POSITIVE
P0I01    AECW    MLR.0H,MRR.1H,                 AH=A*0H                *74
                 TOACIR.0,                      1AH                    *
                 TOMRR.0L                       AL
         AECW2   ACOR.3,SF.0H,ROUND             F3=F4,ROUND AL          75
         AECW    MLR.1L,MRR.0L,                 3AL=F3*AL              *76
                 TOARIR.2,                      AH                     *
                 TOACOR.2,TOMLR.1L,             F3                     *
                 ACOR.1,SF.0H                   F1=F2
         AECW2   RDWA=MLR.0,ALIR.0,AR.2,AAR,    READ DHDL              *77
                 ARIR.2,SF.0L,                  PASS AH                *
                 TOACOR.0,TOMLR.1H,             F1                     *
                 BR.6                           BRANCH TO MIPL1
         AECW    MLR.0L,MRR.1H,                 AL=A*0L                *78
                 TOARIR.0,                      3AL                    *
                 ALIR.3C,ACOR.0C,AROR.1,SF.0H,  F4=-D-F1+ONE           *
                 TOMRR.0H                       AH
         AECW2   RDWB=ALIR.1,AR.3,AAR,          READ SHSL              *79
                 MLR.1L,MRR.0L,ROUND,           3AL=F3*AL,ROUND AH     *
                 ALIR.1L,ACIR.0C,ARIR.0C,SF.1H, 0L=SL-1AH-3AL          *
                 TOACOR.3,TOMLR.2L,             F4                     *
                 BNZ,8                          BRANCH ON CT TO MIPLP
*
P0I02    AECW    MLR.2H,MRR.0H,                 2AH=F2*AH              *80
                 TOARIR.3,                      AL                     *
                 ALIR.1H,ARIR.2,SF.1H           0T=SH+AH
         AECW2   TOALOR.0,                      0T                     *81
                 ARIR.3,SF.0L                   PASS AL
         AECW    MLR.0H,MRR.1H,                 AH=A*0H                *82
                 TOARIR.1,                      2AH                    *
                 TOMRR.0L,                      AL                     *
                 ACOR.3,SF.0H                   F3=F4
         AECW2   MLR.1L,MRR.0H,ROUND,           4AH=F3*AH,ROUND AL     *83
                 ALOR.0,ACIR.1C,ARIR.1,SF.1H,   0H=0T-2AL+2AH          *
                 TOACOR.2,TOMLR.1L,             F3                     *
                 BR,2                           BRANCH TO MIPL2
         AECW    MLR.1H,MRR.0L,                 2AL=F1*AL              *84
                 TOARIR.2,                      AH                     *
                 ALIR.3,ACOR.0,SF.0H,           F1=F1+D                *
                 TOAEOR.0H                      0H
         AECW2   RDWA=MLR.0,ALIR.0,AR.2,AAR,    READ DHDL              *85
                 ARIR.2,SF.0L,ROUND,            ROUND 0H, PASS AH      *
                 TOARIR.1,TOACOR.0,TOMLR.1H     F1
*
*        IN TO OUT TRANSITIONS, D NEGATIVE
MI001    AECW    ALIR.1H,ARIR.0C,SF.1H          0H=SH-3AL               86
         AECW2   TOAEOR.0H,                     0H                     *87
                 ALIR.3C,ACOR.1C,AROR.1,SF.0H   F4=-D-F2+1
         AECW    ALIR.3,ACOR.1,SF.0H,           F2=D+F2                *88
                 MLR.2H,MRR.0H,                 1AH=F2*AH              *
                 TOMLR.2L,ROUND                 F4,ROUND 0H
```

```
         AECW2   ALIR.3,ACOR.1,SF.0H,          F2=D+F2                 *89
                 TOACOR.1,                     F2                      *
                 BR.7                          BRANCH TO MOPL1
         AECW    MLR.0H,MRR.1H,                AH=A*DH                 *90
                 ALIR.3C,ACOR.1C,ARDR.1,SF.0H, F3=-D-F2+ONE            *
                 TOACIR.0,                     1AH                     *
                 TOMLR.2H                      F2
         AECW2   RDWR=ALIR.1,AR.3,AAR,         READ SHSL               *91
                 MLR.2L,MRR.0H,                4AH=F4*AH               *
                 ALIR.1L,ACIR.0,SF.1H,         0L=SL+1AH               *
                 TOACOR.2,TOMLR.1L,            F3                      *
                 BNZ.9                         BRANCH ON CT TO MOPLP
*
MI002    AECW    SF.0H                         NOP                      92
         AECW2   RDWR=MLR.0,ALIR.0,AR.2,SAR    BACKUP DHDL AR.2         93
         AECW    ACOR.0,SF.0H                  F2=F1                    94
         AECW2   TOACOR.1,TOMLR.2H,            F2                      *95
                 BR.7                          BRANCH TO MOPL1
         AECW    MLR.0H,MRR.1H,                AH=A*DH                 *96
                 ALIR.3C,ACOR.0C,ARDR.1,SF.0H  F3=-D-F1+ONE
         AECW2   RDWR=ALIR.1,AR.3,AAR,         READ SHSL               *97
                 MLR.1L,MRR.0H,                4AH=F3*AH               *
                 ALIR.1L,ARIR.1C,SF.1H,        0L=SL-4AH               *
                 TOACOR.2,TOMLR.1L,            F3                      *
                 BNZ.9                         BRANCH ON CT TO MOPLP
*
*        OUT TO IN TRANSITIONS, D NEGATIVE
MOI01    AECW    SF.0H                         NOOP                     98
         AECW2   ARIR.3C,SF.0L                 DLT=-AL                  99
         AECW    TOMLR.3H                      DLT                      100
         AECW2   ACOR.1,SF.0H,                 F1=F2                   *101
                 BR.6                          BRANCH TO MIPL1
         AECW    MLR.3H,MRR.1L,                AL=-1*DLT               *102
                 ALIR.3C,ACOR.3,SF.0H,         F4=F4-D                 *
                 TOACOR.0,TOMLR.1H             F1
         AECW2   MLR.2L,MRR.0L,                3AL=F4*AL               *103
                 ALIR.1L,ACIR.1,ARIR.0C,SF.1H, 0L=SL+2AL-3AL           *
                 TOACOR.3,TOMLR.2L,            F4                      *
                 RDWR=ALIR.1,AR.3,AAR,         READ SHSL               *
                 BNZ.8                         BRANCH ON COUNT MIPLP
*
MOI02    AECW    TOARIR.3,                     AL                      *104
                 ALIR.1H,ACIR.0,ARIR.1C,SF.1H  0H=SH+1AH-4AH
         AECW2   ARIR.3C,SF.0L,                DLT=-AL                 *105
                 TOAFOR.0H                     0H
         AECW    MLR.1H,MRR.0L,                1AL=F1*AL               *106
                 ALIR.3,ACOR.0,SF.0H,          F1=F1+D                 *
                 TOMLR.3H,ROUND                DLT,ROUND 0H
         AECW2   TOACOR.0,TOMLR.1H,            F1                      *107
                 BR.6                          BRANCH TO MIPL1
         AECW    MLR.3H,MRR.1L,                AL=DLT*-1               *108
                 TOARIR.0,                     1AL                     *
                 ALIR.3C,ACOR.3,SF.0H          F4=F4-D
         AECW2   RDWR=ALIR.1,AR.3,AAR,         READ SHSL               *109
                 MLR.2L,MRR.0L,                3AL=F4*AL               *
                 ALIR.1L,ACIR.0,ARIR.0,SF.1H,  0L=SL+1AH+1AL           *
                 TOACOR.3,TOMLR.2L,            F4                      *
                 BNZ.8                         BRANCH ON CT TO MIPLP
```

```
 SSSSSSS     SSSSSSS    SS           SSSSSSS    SSSSSSSSS
SSSSSSSSS   SSSSSSSSS   SS          SSSSSSSSS   SSSSSSSSS
SS       S  SS       S  SS          SS      SS      SS
SS          SS          SS          SS      SS     SS
SSSSSSSSS   SS          SS          SS      SS    SSS
 SSSSSSSSS  SS          SS          SSSSSSSSS      SSS
        SS  SS          SS          SSSSSSSSS      SS
S       SS  SS       S  SS          SS      SS  S      SS
SSSSSSSSS   SSSSSSSSS   SSSSSSSSS   SS      SS  SSSSSSSSS
 SSSSSSS     SSSSSSS    SSSSSSSSS   SS      SS   SSSSSS
```

```
*COMDECK SCLA3
SCLA3    AECW     SF.0H                          NOP                    0
         AECW2    RDWB=MRR.0.AR.0                READ A.B               1
         AECW     SF.0H                          NOP                    2
         AECW2    RDWB=MRR.1.AR.0.AAR            READ C                 3
         AECW     SF.0H                          NOP                    4
         AECW2    RDWB=MLR.0.AR.1                READ X1.X2             5
         AECW     SF.0H                          NOP                    6
         AECW2    RDWB=MLR.1.AR.2                READ Y1.Y2             7
         AECW     MLR.0H.MRR.0H                  AX1=A*X1               8
         AECW2    RDWB=MLR.2.AR.3.               READ Z1.Z2             *9
                  MLR.0L.MRR.0H                  AX2=A*X2
         AECW     MLR.1H.MRR.0L.TOALIR.0         BY1=B*Y1. DES AX1      A
         AECW2    RDWB=ALIR.2.AR.4.SAR.          DUMMY READ             *B
                  MLR.1L.MRR.0L.TOALIR.1         BY2=B*Y2. DES AX2
SC3LP    AECW     MLR.2H.MRR.1H.TOACIR.0         CZ1=C*Z1. DES BY1      C
         AECW2    RDWB=MLR.0.AR.1.AAR.           READ X3.X4             *D
                  MLR.2L.MRR.1H.TOACIR.1         CZ2=C*Z2. DES BY2
         AECW     TOARIR.0                       DES CZ1                E
         AECW2    RDWB=MLR.1.AR.2.AAR.           READ Y3.Y4             *F
                  TOARIR.1.SF.0H.                DES CZ2                *
                  ALIR.0.ACIR.0.ARIR.0           W1=AX1+BY1+CZ1
         AECW     MLR.0H.MRR.0H.TOAEOR.0H.       AX3=A*X3. DES W1       *10
                  ALIR.1.ACIR.1.ARIR.1.SF.0H     W2=AX2+BY2+CZ2
         AECW2    RDWB=MLR.2.AR.3.AAR.ROUND.     READ Z3.Z4.ROUND W1    *11
                  MLR.0L.MRR.0H.                 AX4=A*X4               *
                  SF.0H.TOAEOR.0L.BN7.9          DES W2. LOOP
         AECW     MLR.1H.MRR.0L.TOALIR.0.        BY3=B*Y3. DES AX3      *12
                  ROUND                          ROUND W2
         AECW2    WRM=AEOR.0.AR.4.AAR.           WRITE W1.W2            *13
                  MLR.1L.MRR.0L.TOALIR.1         BY4=B*Y4. DES AX4
         AECW     SF.0H                          NOP                    14
         AECW2    STOP                           STOP                   15
```



```
*COMDECK F22S3
F22S3   AECW    SF.0H
        AECW2   RDWB=MRR.2.AR.5
        AECW    SF.0H
        AECW2   RDWB=MRR.3.AR.5.AAR
        AECW    SF.0H
        AECW2   RDWB=MLR.2.AR.1
        AECW    SF.0H
        AECW2   RDWB=MLR.3.AR.1.AAR
        AECW    MLR.2H.MRR.2L
        AECW2   RDWB=MRR.0.AR.1.AAR.
                MLR.2L.MRR.3L
        AECW    MLR.3H.MRR.2H.TOACIR.2
        AECW2   RDWB=MLR.0.AR.2.
                MLR.3L.MRR.3H.TOACIR.3
        AECW    TOARIR.2
        AECW2   RDWB=MLR.1.AR.3.
                TOARIR.3
        AECW    MLR.0H.MRR.0H
        AECW2   RDWB=ALIR.1.AR.5.AAR
        AECW    MLR.1H.MRR.0L.TOALIR.2
        AECW2   RDWB=ALIR.0.AR.0.
                ALIR.2.ACIR.2.ARIR.2.SF.0L
        AECW    MLR.3H.MRR.2L.TOALIR.3.
                ALIR.1L.SF.0H.
                TOACOR.1.TOMRR.2H
        AECW2   RDWB=ACIR.0.AR.4.SAR.
                MLR.3L.MRR.3L.
                TOAROR.0.ROUND
        AECW    MLR.0L.MRR.0H.TOARIR.2.
                ALIR.3.ACIR.3.ARIR.3.SF.0L.
                ROUND
        AECW2   RDWB=MLR.0.AR.2.AAR.
                MLR.1L.MRR.0L.TOARIR.3.
                ACOR.1.SF.0H.
                TOAROR.1.TOMRR.3H
F22LP   AECW    MLR.2H.MRR.2H.TOALIR.2.
                ALIR.0H.ACOR.1.AROR.1.SF.0H.
                TOAEOR.1H.ROUND
        AECW2   RDWB=MLR.1.AR.3.AAR.
                MLR.2L.MRR.3H.TOALIR.3.
                AROR.1.SF.0H.
                TOACOR.0.ROUND
        AECW    MLR.3H.MRR.2H.TOACIR.2.
                ACOR.0.AROR.0C.SF.1H.
```

```
NOP                             0
READ FH.FL                      1
NOP                             2
READ GH.GL                      3
NOP                             4
READ A.C                        5
NOP                             6
READ B.D                        7
AFL=A*FL                        8
READ U.V                        *9
CGL=C*GL
BFH=B*FH. DES AFL               A
READ XH.XL                      *B
DGH=D*GH. DES CGL
DES BFH                         C
READ YH.YL                      *D
DES DGH
UXH=U*XH                        E
READ SSH.SSL                    F
VYH=V*YH. DES UXH               10
READ SH.SL                      *11
FH=UXH+AFL+BFH
BFL=B*FL. DES VYH               *12
PASS SSL                        *
DES FH
DUMMY READ ON OUTPUT            *13
DGL=D*GL                        *
DES SSL. RND FH
UXL=U*XL. DES BFL               *14
GH=VYH+CGL+DGH                  *
ROUND SSL
READ XH.XL                      *15
VYL=V*YL. DES DGL               *
FHO=FH                          *
DES GH
AFH=A*FH. DES UXL               *16
SSH=SH+FH+GH                    *
DES FHO. RND GH
READ YH.YL                      *17
CGH=C*GH. DES VYL               *
GHO=GH                          *
DES SSH. RND FHO
BFH=B*FH. DES AFH               *18
SHO=SSH(-SSL)                   *
```

```
        TOAEOR,2H,ROUND          DES GHO, RND SSH
AECW2   MLR,3L,MRR,3H,TOACIR,3,  DGY=D*GH, DES CGH      *19
        ALIR,2,ACIR,2,ARIR,2,SF,0L,  FL=UXL+AFH+HFL     *
        TOAEOR,0H,ROUND          DES SHO, RND GHO
AECW    MLR,0H,MRR,0H,TOARIR,2,  UXH=U*XH, DES BFH      *1A
        ALIR,3,ACIR,3,ARIR,3,SF,0L,  GL=VYL+CGH+DGL     *
        TOACOR,2,TOMRR,2L,ROUND  DES FL, RND SHO
AECW2   MLR,1H,MRR,0L,TOARIR,3,  VYH=V*YH, DES DGH      *1B
        ACOR,2,SF,0H,            FLO=FL                 *
        TOARCR,2,TOMRR,3L,ROUND  DES GL, RND FL
AECW    MLR,2H,MRR,2L,TOALIR,2,  AFL=A*FL, DES UXH      *1C
        ALIR,0L,ACOR,2,AROR,2,SF,0H,  SSL=SL+FL+GL      *
        TOAEOR,1L,ROUND          DES FLO, RND GL
AECW2   RDWR=ALIR,0,AR,0,AAR,    READ SH,SL             *1D
        MLR,2L,MRR,3L,TOALIR,3,  CGL=C*GL, DES VYH      *
        AROR,2,SF,0H,            GLO=GL                 *
        TOAROR,0,ROUND           DES SSL, RND FLO
AECW    MLR,3H,MRR,2L,TOACIR,2,  HFL=H*FL, DES AFL      *1E
        AROR,0,ACOR,0C,SF,1L,    SLO=SSL(-SSH)          *
        TOAEOR,2L,ROUND          DES GLO, RND SSL
AECW2   RDWR=MLR,0,AR,2,AAR,     READ XH,XL             *1F
        MLR,3L,MRR,3L,TOACIR,3,  DGL=D*GL, DES CGL      *
        ALIR,2,ACIR,2,ARIR,2,SF,0L,  FH=UXH+AFL+BFH     *
        TOAEOR,0L,ROUND,BNZ,B    DES SLO, RND GLO, BR
AECW    MLR,0L,MRR,0H,TOARIR,2,  UXL=U*XL, DES BFL      *20
        ALIR,3,ACIR,3,ARIR,3,SF,0L,  GH=VYH+CGL+DGH     *
        TOACOR,1,TOMRR,2H,ROUND  DES FH, RND SLO
AECW2   WRM=AEOR,0,AR,4,AAR,     WRITE SHO,SLO          *21
        MLR,1L,MRR,0L,TOARIR,3,  VYL=V*YL, DES DGL      *
        ACOR,1,SF,0H,            FHO=FH                 *
        TOARCR,1,TOMRR,3H,ROUND  DES GH, RND FH
AECW    AROR,0,SF,0H             PASS SSL                22
AECW2   TOAEOR,3L                DES SSL                 23
AECW    SF,0H                    NOP                     24
AECW2   WRM=AEOR,3,AR,5          WRITE SSO               25
AECW    SF,0H                    NOP                     26
AECW2   WRM=AEOR,2,AR,5,SAR      WRITE GO                27
AECW    SF,0H                    NOP                     28
AECW2   WRM=AEOR,1,AR,5,SAR      WRITE FO                29
AECW    SF,0H                    NOP                     2A
AECW2   STOP                     STOP                    2B
```

```
DDDDDDDDD     DDDDDDDDDDD   DD       DD     DDDDDDDD     DD        DD
DDDDDDDDDDD   DDDDDDDDDDD   DDDD   DDDD   DDDDDDDDDD     DDD       DD
DD        DD  DD            DD DDDD DD   DD         DD   DD D      DD
DD        DD  DD            DD  DD  DD   DD         DD   DD DD     DD
DD        DD  DDDDDD        DD  DD  DD   DD         DD   DD  DD    DD
DD        DD  DDDDDD        DD      DD   DD         DD   DD   DD   DD
DD        DD  DD            DD      DD   DD         DD   DD    DD  DD
DD        DD  DD            DD      DD   DD         DD   DD     D  DD
DDDDDDDDDDD   DDDDDDDDDD    DD      DD   DDDDDDDDDDD   DD       DDD
DDDDDDDDD     DDDDDDDDDD    DD      DD    DDDDDDDD     DD        DD
```

```
*COMDECK DEMON
DEMON    AECW     SF.0H                              NOP                          0
         AECW2    RDWR=ALIR.2,AR.6                   READ L1,T1                   1
         AECW     SF.0H                              NOP                          2
         AECW2    RDWR=MLR.0,AR.6,AAR                READ K1,A1                   3
         AECW     ALIR.2L,SF.1L                      PASS T1                      4
         AECW2    RDWR=MRR.1,AR.6,AAR,TOACOR.0       READ B,C, DES T1             5
         AECW     SF.0H                              NOP                          6
         AECW2    RDWR=ALIR.3,AR.6,AAR               READ L2,T2                   7
         AECW     SF.0H                              NOP                          8
         AECW2    RDWR=MRR.2,AR.4,                   READ X11,X12                 *9
                  SCMR.0H=AR.0                       COS (S11) AE0
         AECW     ALIR.3L,SF.1L                      PASS T2                      A
         AECW2    RDWR=MLR.2,AR.7,TOACOR.1,          READ Y11,Y12, DES T2         *B
                  SCMR.0H=AR.2                       COS (S11) AE1
         AECW     SF.0H                              NOP                          C
         AECW2    RDWR=MLR.1,AR.6,AAR                READ K2,A2                   D
         AECW     SF.0H                              NOP                          F
         AECW2    RDWR=MRR.3,AR.4,AAR,               READ X21,X22                 *F
                  SCMR.0H=AR.1,                      COS (S21) AE0                *
                  MLR.0L,MRR.2H                      AX11=A1*X11
         AECW     MLR.2H,MRR.1L                      CY11=C*Y11                   10
         AECW2    RDWR=MLR.3,AR.7,AAR,               READ Y21,Y22                 *11
                  SCMR.0H=AR.3,                      COS (S21) AE1                *
                  MLR.2L,MRR.1H,TOARIR.0             BY12=B*Y12, DES AX11
         AECW     MLR.0H,MRR.0H,TOACIR.0             KS11=K1*S11, DES CY11        12
         AECW2    TOALIR.0                           DES BY12                     13
         AECW     TOARIR.2                           DES KS11                     14
         AECW2    RDWR=ALIR.1,AR.5,                  READ TG1,TG2                 *15
                  SCMR.0H=AR.0INC,                   COS (S12) AE0                *
                  MLR.3H,MRR.1L                      CY21=C*Y21
         AECW     MLR.1H,MRR.0H,                     KS21=K2*S21                  *16
                  ALIR.2H,ACOR.0C,ARIR.2,SF.1H       P11=L1-T1+KS11
         AECW2    RDWR=ACIR.3,AR.5,SAR,              DUMMY READ ON OUTPUT         *17
                  SCMR.0H=AR.2INC,                   COS (S12) AE1                *
                  MLR.3L,MRR.1H,TOACIR.1,            BY22=B*Y22, DES CY21         *
                  ALIR.0,ACIR.0,ARIR.0,SF.0L,        Y11=BY12+CY11+AX11           *
                  TOAROR.0,SETCS                     DES P11
         AECW     MLR.1L,MRR.3H,TOARIR.2,            AX21=A2*X21, DES KS21        *18
                  ACOR.0,AROR.0TS,SF.1L,             DM11=T1+(P11)C               *
                  TOALOR.0,TOMLR.2H                  DES Y11
         AECW2    SCMR.0H=AR.1INC,                   COS (S22) AE0                *19
                  MLR.2L,MRR.1L,TOALIR.0,            CY12=C*Y12, DES BY22         *
                  TOMRR.0L,ROUND                     DES DM11, RND Y11
         AECW     MLR.3L,MRR.1L,TOARIR.1,            CY22=C*Y22, DES AX21         *1A
```

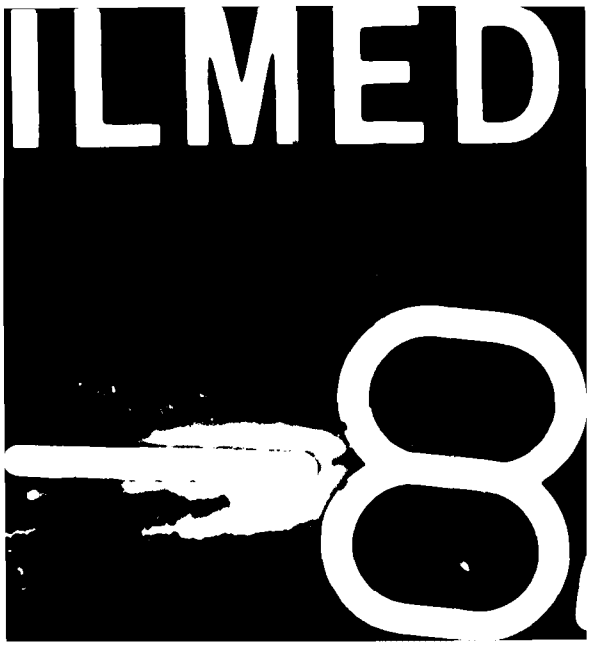
```
                ALIR,3H,ACOR,1C,ARIR,2,SF,1H
          AECW2 SCMR,0H=AR,3INC,
                MLR,2H,MRR,0L,TOACIR,0,
                ALIR,0,ACIR,1,ARIR,1,SF,0L,
                TOAROR,0,SETCS
          AECW  MLR,0H,MRR,0H,TOACIR,1,
                ACOR,1,AROR,0TS,SF,1L,
                TOALOR,1,TOMLR,3H
          AECW2 MLR,2H,MRR,1H,TOACIR,2,
                ALOR,0,SF,0H,
                TOMRR,0L,ROUND
OMNLP     AECW  MLR,0L,MRR,2L,TOARIR,2,
                TOAEOR,0H
          AECW2 RDWR=MRR,2,AR,4,AAR,
                SCMR,0H=AR,0INC,
                MLR,3H,MRR,0L,TOALIR,0,
                ALIR,2H,ACOR,0C,ARIR,2,SF,1H
          AECW  MLR,1H,MRR,0H,TOARIR,0,
                ALOR,1,SF,0H,TOAROR,0,SETCS
          AECW2 SCMR,0H=AR,2INC,
                MLR,3H,MRR,1H,TOARIR,3,
                ALIR,0,ACIR,0,ARIR,0,SF,0L,
                TOAEOR,1H
          AECW  MLR,1L,MRR,3L,TOARIR,2,
                ACOR,0,AROR,0TS,SF,1L,
                TOALOR,2,TOMLR,2L
          AECW2 SCMR,0H=AR,1INC,
                MLR,2H,MRR,1L,TOALIR,0,
                ALIR,3H,ACOR,1C,ARIR,2,SF,1H,
                TOMRR,0L,ROUND
          AECW  MLR,3H,MRR,1L,TOARIR,1,
                ALIR,1H,ACIR,2,ARIR,3,SF,2H,
                TOAROR,0,SETCS
          AECW2 SCMR,0H=AR,3INC,
                MLR,2L,MRR,0L,TOACIR,0,
                ALIR,0,ACIR,1,ARIR,1,SF,0L,
                TOAFOR,2H
          AECW  MLR,0H,MRR,0H,TOACIR,1,
                ACOR,1,AROR,0TS,SF,1L,
                TOALOR,3,TOMLR,3L,ROUND
          AECW2 MLR,2L,MRR,1H,TOACIR,2,
                ALOR,2,SF,0H,
                TOMRR,0L,ROUND
          AECW  MLR,0L,MRR,2H,TOARIR,2,
                TOAEOR,0L
          AECW2 RDWR=MRR,3,AR,4,AAR,
                SCMR,0H=AR,0INC,
                MLR,3L,MRR,0L,TOALIR,0,
                ALIR,2H,ACOR,0C,ARIR,2,SF,1H
          AECW  MLR,1H,MRR,0H,TOARIR,0,
                ALOR,3,SF,0H,TOAROR,0,SETCS
          AECW2 SCMR,0H=AR,2INC,
                MLR,3L,MRR,1H,TOARIR,3,
                ALIR,0,ACIR,0,ARIR,0,SF,0L,
                TOAFOR,1L
          AECW  MLR,1L,MRR,3H,TOARIR,2,
                ACOR,0,AROR,0TS,SF,1L,
                TOALOR,0,TOMLR,2H

P21=L2-T2*KS21
COS (S22) AE1                    *1B
D11=DM11*Y11,DES CY12            *
Y21=BY22+CY21*AX21               *
DES P21
KS12=K1*S12, DES CY22            *1C
DM21=T2+(P21)C                   *
DES Y21
BY11=H*Y11, DES D11              *1D
PAS Y11                          *
DES DM21, RND Y21
AX12=A1*X12, DES KS12            *1E
DES Y11
READ X11,X12                     *1F
COS (S11) AE0                    *
D21=DM21*Y21,DES BY11            *
P12=L1-T1*KS12
KS22=K2*S22, DES AX12            *20
PASS Y21, DES P12
COS (S11) AE1                    *21
BY21=H*Y21, DES D21              *
Y12=BY11+CY12*AX12               *
DES Y21
AX22=A2*X22, DES KS22            *22
DM12=T1+(P12)C                   *
DES Y12
COS (S21) AE0                    *23
CY11=C*Y11, DES BY21             *
P22=L2-T2*KS22                   *
DES DM12, RND Y12
CY21=C*Y21, DES AX22             *24
TG1=TG1+D11+D21                  *
DES P22
COS (S21) AE1                    *25
D12=DM12*Y12,DES CY11            *
Y22=BY21+CY22*AX22               *
DES TG1
KS11=K1*S11, DES CY21            *26
DM22=T2+(P22)C                   *
DES Y22, RND TG1
BY12=H*Y12, DES D12              *27
PASS Y12                         *
DES DM22, RND Y22
AX11=A1*X11, DES KS11            *29
DES Y12
READ X21,X22                     *29
COS (S12) AE0                    *
D22=DM22*Y22,DES BY12            *
P11=L1-T1*KS11
KS21=K2*S21, DES AX11            *2A
PASS Y22, DES P11
COS (S12) AE1                    *2B
BY22=H*Y22, DES D22              *
Y11=BY12+CY11+AX11               *
DES Y22
AX21=A2*X21, DES KS21            *2C
DM11=T1+(P11)C                   *
DES Y11
```

```
AECW2   POWR=ALIR.1,AR.5,AIRO,
        SCMR.0H=AR.1INC,
        MLR.2L,MRR.1L,TOALIR.0,
        ALIR.3H,ACOR.1C,ARIR.2,SF.1H,
        TOMRR.0L,ROUND
AECW    MLR.3L,MRR.1L,TOARIR.1,
        ALIR.1L,ACIR.2,ARIR.3,SF.2H,
        TOAROR.0,SETCS
AECW2   SCMR.0H=AR.3INC,
        MLR.2H,MRR.0L,TOACIR.0,
        ALIR.0,ACIR.1,ARIR.1,SF.0L,
        TOAFOR.2L,BNZ.F
AECW    MLR.0H,MRR.0H,TOACIR.1,
        ACOR.1,AROR.0TS,SF.1L,
        TOALOR.1,TOMLR.3H,ROUND
AECW2   WRM=AEOR.2,AR.5,SAR,
        MLR.2H,MRR.1H,TOACIR.2,
        ALOR.0,SF.0H,
        TOMRR.0L,ROUND
AECW    SF.0H
AECW2   WRM=AEOR.0,AR.7,SAR
AECW    SF.0H
AECW2   WRM=AEOR.1,AR.7,AAR
AECW    SF.0H
AECW2   STOP
```

```
        READ TG1,TG2                *20
        COS (S22) AE0               *
        CY12=C*Y12, DES BY22        *
        P21=L2-T2*KS21              *
        DES DM11, RND Y11
        CY22=C*Y22, DES AX21        *2E
        TG2=TG2+D12+D22             *
        DES P21
        COS (S22) AE1               *2F
        D11=DM11*Y11, DES CY12*
        Y21=BY22+CY21+AX21          *
        DES TG2, BR TO DMNLP
        KS12=K1*S12, DES CY22       *30
        DM21=T2+(P21)C              *
        DES Y21, RND TG2
        WRITE TG1,TG2               *31
        BY11=B*Y11, DES D11         *
        PASS Y11                    *
        DES DM21, RND Y21
        NOP                          32
        WRITE STATES Y11,Y12         33
        NOP                          34
        WRITE STATES Y21,Y22         35
        NOP                          36
        STOP                         37
```

DISTRIBUTION

	Copies
Manager, ASW Systems Project Office Department of the Navy Attn: LCDR G. Watson, ASW-131 Washington, D. C. 20362	1
Commander Naval Air Systems Command Department of the Navy Attn: R. Farris, NAVAIR 413 A. Stone, NAVAIR 370 Washington, D. C. 20361	1 1
Commander Naval Air Development Center Attn: M. Effont, Code 2032 Warminster, Pennsylvania 18974	1
Commanding Officer Naval Training Equipment Center Attn: Paul Jackson, Code N223 Orlando, Florida 32813	1
TRW 7600 Colshire Drive Attn: William Payne McLean, Virginia 22101	1
Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	12


ILMED
8