

AD-A102 278

GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV F/6 17/9
ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING, VOLUME II. PAR--ETC(U)
JUN 76 R J HANCOCK, F H CLEVELAND F30602-73-C-0380

UNCLASSIFIED

RADC-TR-76-186-VOL-2-PT-1 NL

1 of 4

AD

4-1278

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

11

2

LEVEL

III

(9)

AD A102278

RADC-TR-76-186, Vol II, Pt 1
Final Technical Report
June 1976

Vol 3 - A030572
VI-PSI - A030555



ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING
(Computer Program Documentation)

General Dynamic

DTIC
ELECTE
S JUL 31 1981 D

Approved for public release;
distribution unlimited.

Laboratory Directors' Fund Number 01707310
with Proj. 6512 funds added.

THIS DOCUMENT IS BEST QUALITY PRACTICABLE
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

DTIC FILE COPY

81 7 31 015

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

APPROVED:

John C. Cleary
JOHN C. CLEARY
Project Engineer

APPROVED:

Moses A. Diab
MOSES A. DIAB
Acting Technical Director
Surveillance Division

FOR THE COMMANDER:

John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

Initially funded under RADC Laboratory Directors' Funds with Proj. 6512 funds added.

Do not return this copy; retain or destroy.

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

19 TR-76-186-VOL-2-PT-1

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC TR-76-186 Vol II, Pt 1	2. GOVT ACCESSION NO. AD-A202278	3. RECIPIENT'S CATALOG NUMBER 2278
4. TITLE AND SUBTITLE ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING Computer Program Documentation		5. TYPE OF REPORT Final Technical Report
6. AUTHOR(s) Robert J. Hancock Fred H. Cleveland		7. PERFORMING ORG. REPORT NUMBER N/A
8. CONTRACT OR GRANT NUMBER(s) F30602-73-C-0380		
9. PERFORMING ORGANIZATION NAME AND ADDRESS General Dynamics Convair Aerospace Division P O Box 748, Ft Worth TX 76101		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101P LDF 01707310 1773
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (OCSA) Griffiss AFB NY 13441		12. REPORT DATE June 1976
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		13. NUMBER OF PAGES 380 12 344
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		16. DECLASSIFICATION/DOWNGRADING N/A 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) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: John C. Cleary (OCSA) Initially funded under RADC Laboratory Directors' Funds with Proj. 6512 funds added.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Radar Simulation Computer Modeling		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This effort is concerned with the development and implementation of a set of digital computer programs that will augment the RADC digital computer radar simulation model procured under Contr F30602-72-C-0393 (01707201). The computer programs shall consist of a sequence of subroutines that correspond to separate functions such as a chaff model, target model, propagation effects and clutter model. The original radar simulation model will be expanded to include a bistatic capability and will include ECM and phase coded pulse compression re-		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ceiver techniques. In addition, an interactive system has been designed for the simulation. Using an interactive display, an engineer would be able to understand what is happening by being able to observe results at several intermediate points in the problem. A picture is worth a thousand words. For example, an antenna pattern or waveform response to a target is more meaningful than a long table of numerical listings. Parts of the simulation were used by RADC for Deep Space Surveillance Radar (DSSR) waveform analysis, generating antenna patterns and tradeoffs involving phase shifter bit-size for the Advanced Space Defense Program (ASDP). The RADC radar simulation model is being used to support Seek Sail, Cobra Judy, Digital Coded Radar and Seek Sentry.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

This report contains a large percentage of machine-produced copy which is not of the highest printing quality but because of economical consideration, it was determined in the best interest of the government that they be used in this publication.

This report contains Vol I, Pt 1 (Sections 1-7 and 9-10) (Pages 1-1 thru 1-5, 2-1 thru 2-24, 3-1 thru 3-35, 4-1 thru 4-23, 5-1 thru 5-6, 6-1 thru 6-39, 7-1 thru 7-30, 9-1 thru 9-3 and 10-1 thru 10-2).

Vol I, Pt 2 contains Section 8 (Pages 8-1 thru 8-174).

Vol I, Pt 3 contains Section 8 (Pages 8-175 thru 8-418).

Vol II, Pt 1 contains Sections 1 - 8 and 10 & 11 (Pages 1-1, 2-1 thru 2-24, 3-1 thru 3-15, 4-1 thru 4-137, 5-1 thru 5-16, 6-1 thru 6-44, 7-1, 8-1 thru 8-26, 10-1 thru 10-4 and 11-1 thru 11-2).

Vol II, Pt 2 contains Sections 9 and 10 (Pages 9-1 thru 9-234 and Pages 10-1 thru 10-4).

Vol III contains Sections 1 thru 6 (Pages 1-1 thru 1-2, 2-1 thru 2-22, 3-1 thru 3-53, 4-1 thru 4-141, 5-1 thru 5-3 and 6-1).

Vol IV, Pt 1 contains Appendices A-K and Appendix M.

Vol IV, Pt 2 contains Appendix L.

Accession For	
NRIS	☒
DTIC	☐
USCIB	☐
JAN 1971	
Bul	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A	53

TABLE OF CONTENTS

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
	List of Figures	vi
1	Introduction	1-1
2	Data Initialization Controller	2-1
3	Simulation Controller	3-1
4	Stimulus/Transfer Function Modules	4-1
	ATOD	4-2
	CDIGFL	4-6
	CFAR	4-10
	CGEN	3-3, Vol III
	CGENSF	3-9, Vol III
	CLUTTR	4-14
	DCFAR	4-20
	DFT	4-24
	DIGFIL	4-30
	DIGFSF	4-34
	DIGTFL	4-41
	ECM	4-45
	FGENXY	4-51
	FILT	4-60
	HET	4-64
	HWDET	4-65
	INGTOR	4-70
	IONOS	4-74
	LAMPCP	4-80
	MTIFLT	4-83
	NONLIN	4-91
	PHDEC	4-94
	PHENC	4-100
	PXFRM	3-12, Vol III
	RDIGFL	4-105
	RECF	3-50, Vol III
	RNDARY	4-109
	SHIFT	4-113
	SWPINT	4-118
	TARGET	4-122

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
	TSARY	3-27, Vol III
	WEITRE	4-129
	WVGUID	4-135
5	Connection Modules	5-1
	CONV	5-2
	ZFFT	5-7
6	Peripheral Modules	6-1
	CLINT	6-2
	CUMDIS	6-11
	ERGYCP	6-16
	PLOTTR	6-18
	PLTFMT	6-23
	PTLIST	6-28
	RTOPDB	6-35
	SCANNR	6-40
	SPCAVG	3-46, Vol III
7	Supervisory Modules	7-1
	ANTARY	3-17, Vol III
	TSRPAT	3-42, Vol III
8	Subordinate Modules	8-1
	ABORT	8-2
	ANTINT	8-4
	AZGAIN/ELGAIN	8-8
	BLOCK DATA	8-10
	DBLKX	8-12
	IBOOL	8-14
	IFLD	8-15
	IPACK	8-16
	PACK	8-18
	RRAND	8-21

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
9	RADSIM Computer Program Flow Charts	9-1
	(Contained in Volume II, Part 2)	
10	Index	10-1
11	Recursive Computation of Sine/Cosine Pairs	11-1
ADDENDA		
4	Stimulus/Transfer Function Modules	
	CHAFF	, Vol I, Pt 3
	DIPOLE	, Vol I, Pt 3
	SCYL	, Vol I, Pt 3
6	Peripheral Modules	
	(Bistatic Antenna Initialization)	, Vol I, Pt 3
	ERRPRC	, Vol I, Pt 3
8	Subordinate Modules	
	BISTGT	, Vol I, Pt 3
	GAM	, Vol I, Pt 3
	BESS	, Vol I, Pt 3
	EXPI	, Vol I, Pt 3

LIST OF FIGURES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE</u>
2-1	Data Initialization Load Module Functional Block Diagram	2-2
3-1	Simulation Load Module Functional Block Diagram	3-2
CDIGFL-1	Block Diagram of CDIGFL/CDFNCL	4-8
CFAR-1	Block Diagram of CFAR	4-11
CLUTTR-1	Clutter Model Geometry	4-15
DCFAR-1	Block Diagram of DCFAR	4-22
DIGFIL-1	Block Diagram of DIGFIL/DIGFNC	4-32
DIGFSF-1	Block Diagram of DIGFSF	4-37
DIGFSF-2	Combination Filter Zero Locations in the Z-Plane	4-38
DIGFSF-3	Block Diagram of a Complex Resonator	4-39
DIGTFL-1	Block Diagram of DIGTFL	4-43
ECM-1	Block Diagram of ECM	4-48
FGENXY-1	Output Waveform Envelopes	4-55
HET-1	Block Diagram of SSB Modulator	4-64a
INGTOR-1	Block Diagram of INGTOR	4-72
MTIFLT-1	Block Diagram of MTIFLT/MTINCL (Floating Point Arithmetic)	4-87
MTIFLT-2	Block Diagram of MTIFLT/MTINCL (Integer Arithmetic)	4-88
PHDEC-1	Block Diagram of a Tapped Delay Line Phase Decoder	4-97
NONLIN-1	Example of Non Linear Trans. Function	4-92a
PHDEC-2	Surface Wave Device Diagrams	4-98

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE</u>
PHENC-1	Block Diagram of Pseudo- Random Phase Code Generators	4-103
RDIGFL-1	Block Diagram of RDIGFL/RDFNCL	4-107
SWPINT-1	Block Diagram of SWPINT/NCSWPI	4-120
TARGET-1	Target Model Geometry	4-126
CLINT-1	Clutter Model Geometry	6-5/6-6
11-1	Z-Plane Representation of the Digital Oscillator	11-1
11-2	Digital Oscillator Block Diagram	11-2

SECTION 1

INTRODUCTION

This volume of the final technical report contains the descriptions of the computer programs and subprograms which constitute the Radar System Simulation Model.

The simulation model computer program is divided into two distinct phases: (1) Data initialization activity, and (2) Simulation activity. In terms of software, each phase is composed of one main and a group of subprograms.

The subprograms are divided into five distinct groups: Stimulus/transfer function modules, connection modules, peripheral modules, supervisory modules and subordinate modules. These groups are more fully described in the sections devoted to each group.

The flow charts and cross reference tables for the entire Radar System Simulation Model are located in Part 2 of this volume.

The module descriptions, program listings, flow charts and cross reference tables have all been cross indexed and are located in Section 10. This section has been included in both parts of the volume for convenience.

SECTION 2

SIMULATION DATA LOADER EXECUTIVE (MAIN - 1)

The simulation data loader serves as the interface between the user and the segment of the computer program which performs the simulation. This segmentation was necessary since input data in a format convenient to the user must be converted to a form suitable for use by the simulation modules. The punch cards which define the simulation to be performed are of two types: simulation control cards and module parameter data cards. The simulation control cards determine what operations are to be performed in the simulation activity. This includes not only the scheduling of modules for execution but also the movement of data to and from temporary storage, and the modification of parameters for multiple executions of a simulation model configuration. Each control card with the exception of ENDPAS, ENDCFG, and ENDJOB is converted into a control word by the control word generator and placed in the control word block. Each control word contains the code number of the operation to be performed, the module to be executed and data set reference number if required. The module parameter data cards define the parameters of the simulation modules to be executed in the simulation. Each module requiring input data has a unique name-list which contains the input parameters.

Figure 2-1 is a block diagram of the Initializer load module. In the block diagram the data flow paths are shown as solid lines and control paths are shown as dashed lines. Control normally is retained by the Simulation Data Loader Executive but control transfer does occur to the subprograms CLINT and PHENC. The blocks containing the letter M represent data transfers between two storage areas. Arrows are used to indicate the direction of data flow. The blocks containing C&M perform a search operation to determine if the word stored in a buffer is a member of a reference dictionary. If the search is successful, the data in the buffer is transferred to the storage area designated by an arrow. The function served by certain blocks in the diagram are evident from their titles. Those blocks requiring further explanation are discussed in the following paragraphs.

The Module Parameter Namelist Dictionary serves the function of directing the input data to the proper location within the Module/System Parameter Table. Appendix B contains a list of the parameters in each namelist and the storage location assigned to each parameter. The namelist name is the same as the Module Reference Number, i.e., the namelist for module 101 is NL101.

The Module Reference Number Dictionary is a cross reference between Module Reference Numbers and the overhead operations which must be accomplished prior to execution of a module. For example, to generate a phase encoded waveform requires that the user supplied data be preprocessed by subroutine PHENC.

The Command Word Dictionary is a cross reference between input command words and the operations to be performed in the simulation. The control word is entered in columns 1 through 6 of the simulation control card. The following is a list of the command words and the corresponding operation initiated by each:

<u>EXEC</u>	This command word schedules execution of the module corresponding to the number appearing in columns 15 through 17 of the control card. No input data cards are required.
<u>LDEXEC</u>	This command word is the same as EXEC with the additional requirement that input data is read. The namelist name for entering the data is the same as the module reference number contained in columns 15 through 17 of the simulation control card.
<u>MODIFY</u>	This command word causes the data loaded in a previous step to be modified. This is typically used to change parameters when multiple simulation passes are to be made. The step number to be changed is entered in columns 10 through 12 of the simulation control card, right adjusted. The Module Reference Number is entered in columns 15 through 17 of the simulation control card.
<u>ENDPAS</u>	This command word signifies the end of a pass through a simulation configuration.

<u>ENDCFG</u>	This command word signifies the end of a configuration.
<u>ENDJOB</u>	This command word signifies the end of a simulation job.
<u>STOREX</u>	This command word causes the contents of the XT signal storage array to be stored on a temporary data set, usually a disc file. The Data Set Reference Number is entered in columns 11 and 12 of the simulation control card.
<u>STOREY</u>	Same as STOREX except the contents of signal storage array YT are stored.
<u>STOREA</u>	Same as STOREX except the contents of auxiliary storage array XA are stored.
<u>STOREB</u>	Same as STOREX except the contents of auxiliary storage array XB are stored.
<u>LOAD X</u>	This command word causes the data located in a temporary file to be loaded into the signal storage array XT. The Data Set Reference Number is entered in columns 11 and 12 of the simulation control card.
<u>LOAD Y</u>	Same as LOAD X except the signal storage array YT is loaded.
<u>LOAD A</u>	Same as LOAD X except the auxiliary storage array XA is loaded.
<u>LOAD B</u>	Same as LOAD X except the auxiliary storage array XB is loaded.
<u>REWIND</u>	This command word causes a temporary file to be rewound. For a disc file this moves the access pointer to the beginning of the data set. The Data Set Reference Number is entered in columns 11 and 12 of the simulation control card.

The output from the data loader are stored on temporary storage files. Data Set Reference Number 11 contains the control word block which controls the simulation activity and the module/system parameter tables used to define the modules and system characteristics. Data Set Reference Number 2 contains the clutter scatterer parameters.

Flow Chart: Page 9-2

Cross Reference Table: Page 9-206

02/11/75

INLET LISTING

AUTOFLOW CHART SET - FWL/SCL RAUSIM

FOR THIS MODULE

(CONTINUED)

2-6

LINE NO.

CONTENTS

1	COMMON/CLK1/ USCAT(5000)	
	COMMON /CLK2/ ITEMP(500),IEXEC(500)	
	COMMON/CLKND/ XNUM(141)	
2		
3		
4	DIMENSION ANTA(2,75) , ANTEL(2,75)	
5	EXPENSIVE WTX(3,100) , WT(3,100) , FZERU(2,50) , FFCLE(2,50),	010
6	VIGAND(1,9),TEN(2,50)	
7		
8		
9	DIMENSION FCOM(1300), DEC(3,32),IPY(6)	
10		
11	DIMENSION ISCAT(3,100)	
12	DIMENSION ISFLS(100)	
13		
14	DIMENSION DETH(3,100)	
15		
16	DIMENSION HAP(3,100)	
17		
18	DIMENSION F81(11),FSAM(2,100),FBCLEF(2,25),FFCULF(2,25)	
19		
20		
21	CONTRACT VIL*100,VIL*100,GBL*100	
22	ENTER ENCLX, ENCFAS, ENCLF, ENCLFG, STORX, STORLY, STORCA,	025
23	* ENCLY, ROWING, CUTX, CUTY, CUTA, CUTB	
24	ENTER CLTAX,CLTAY,CLTARX,CLTARY,BLANK,KNIP	
25	DORE LANCDA,JOETART,JVEL,JHEF(0),JFC	
26	JVAL JFHC,JJEDM,JMAZ,JHGT,JEKE,JFMDW,JFW	
27		
28		
29	DATA EXEC/EXEC' / , LEXIC/LEXEC' / , ENOPAS/ENOPAS' / ,	125
30	ENCLND/ENCLND' / , ENCLFG/ENCLFG' / , MODIFY/MODIFY' / ,	126
31	STORX/STORX' / , STORLY/STORLY' / , STORCA/STORCA' / ,	127
32	STORLY/STORLY' / , LEAD X/LEAD X' / , LEAD Y/LEAD Y' / ,	128
33	CUTX/CUTX' / , CUTA/CUTA' / , CUTB/CUTB' / ,	129

29 * COUT X/'COUT X'/ , COUT Y/'COUT Y'/ , COUT Z/'COUT Z'/ , 130 2-7
30 * COUT U/'COUT U'/ , INPUT X/'INPUT X'/ , INPUT Y/'INPUT Y'/ , 131
31 * INPUT A/'INPUT A'/ , INPUT E/'INPUT E'/ , SKIP/'SKIP '/
32 DATA NCDEX/'NCDEX'/ , CLEAR X/'CLEAR X'/ , CLEAR Y/'CLEAR Y'/ , 360
33 * CLEAR Z/'CLEAR Z'/ , CLEAR E/'CLEAR E'/ , CLEAR A/'/

34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

103 * (ITEMP(65), NDRF),(ITEMP(66), NACIL) 093 2-9
 104 * (ITEMP(65), IL),(ITEMP(66), IL) 094
 105 * (ITEMP(201), VILL),(ITEMP(201),VILL),
 106 * (ITEMP(201), GUBU)
 107 C 097
 108 EQUIVALENCE (ITEMP(67), KTEH),(ITEMP(67), IFQ(1,1)) 098
 109 C
 110 EQUIVALENCE (ITEMP(68), FFO),(ITEMP(69), FFI) 100
 111 * (ITEMP(70), FFI),(ITEMP(71), FFI) 101
 112 EQUIVALENCE (ITEMP(68), FFI),(ITEMP(69), FFI), 1031
 113 * (ITEMP(70), FFI),(ITEMP(71), FFI) 1032
 114 EQUIVALENCE (ITEMP(160), IFFON),(ITEMP(161), IFFOD) 102
 115 * (ITEMP(162), IFFIN),(ITEMP(163), IFFIO) 103
 116 * (ITEMP(164), IFFIN),(ITEMP(165), IFFIO) 104
 117 * (ITEMP(166), IFFIN),(ITEMP(167), IFFIO) 105
 118 * (ITEMP(168), MDEDE),(ITEMP(169), NEITDE) 106
 119 EQUIVALENCE (ITEMP(160), IFFON),(ITEMP(161), IFFOD), 1081
 120 * (ITEMP(162), IFFIN),(ITEMP(163), IFFIO), 1082
 121 * (ITEMP(164), IFFIN),(ITEMP(165), IFFIO), 1083
 122 * (ITEMP(166), IFFIN),(ITEMP(167), IFFIO) 1084
 123 C
 124 EQUIVALENCE (ITEMP(72), NZ),(ITEMP(73), NF) 108
 125 * (ITEMP(74), SF),(ITEMP(201), FZERO(1,1)), 109
 126 * (ITEMP(301), FPULL(1,1)) 110
 127 C 111
 128 EQUIVALENCE (ITEMP(75), FJCK) 112
 129 C 113
 130 EQUIVALENCE (ITEMP(76), NTSAP),(ITEMP(77), DX) 114
 131 * (ITEMP(78), THEIAS),(ITEMP(79), NKUWS) 115

00000000

00000000

AUTOFLOW DEFEAT SET - FWD/SCL - FASIM

2-10

00000000

00000000

CONTENTS

116	*	(ITEMP(101), NPSOC) , (ITEMP(102), NPSOC) ,	116
117	*	(ITEMP(103), NPSK) , (ITEMP(104), NPSK) ,	117
118	*	(ITEMP(105), INCKM) , (ITEMP(106), NPSLS) ,	118
119	*	(ITEMP(107), IDLISL) , (ITEMP(108), ANGLE) ,	119
120	*	(ITEMP(109), NGGELS) , (ITEMP(110), TCGNLM) ,	120
121	*	(ITEMP(111), TIMESB) , (ITEMP(112), MODTSK) ,	121
122	*	(ITEMP(113), REDEEL) , (ITEMP(114), REDEEL) ,	122
123	*	(ITEMP(115), IMETAR) ,	123
124	*	(ITEMP(116), TSKUS(1)) ,	124
125	*	(ITEMP(117), TOLMUN) ,	125
126	*	(ITEMP(118), TOLMUN) ,	126
127	*	(ITEMP(119), ESTRI) ,	127
128	*	(ITEMP(120), ENKIP) , (ITEMP(121), ENKIP) ,	128
129	*	(ITEMP(122), NPTX) , (ITEMP(123), SPW) ,	129
130	*	(ITEMP(124), NSUPP) , (ITEMP(125), SWTIM) ,	130
131	*	(ITEMP(126), KISTIM) , (ITEMP(127), FALTIM) ,	131
132	*	(ITEMP(128), TSTART) , (ITEMP(129), PCDL(1)) ,	132
133	*	(ITEMP(130), WTX(1,1)) , (ITEMP(131), OPRI) ,	133
134	*	(ITEMP(132), VPEAK) ,	134
135	*	(ITEMP(133), TCDL(1)) , (ITEMP(134), MODLPHI) ,	135
136	*	(ITEMP(135), NSK) , (ITEMP(136), IPY(1)) ,	136
137	*	(ITEMP(137), NTAPS) , (ITEMP(138), PCDL(1)) ,	137
138	*	(ITEMP(139), TAPSPC) , (ITEMP(140), INPTF) ,	138
139	*	(ITEMP(141), ITAP(1,1)) , (ITEMP(142), UCU(1,1)) ,	139
140	*	(ITEMP(143), XLSB) , (ITEMP(144), NPTIS) ,	140
141	*	(ITEMP(145), INOFF) , (ITEMP(146), ALLES) ,	141
142	*	(ITEMP(147), INOFF) , (ITEMP(148), ALLES) ,	142

187 EQUIVALENCE (ITEMP(106), HTOI),(ITEMP(107), RIGI) 137 2-11
188 * (ITEMP(108), ANGTOT),(ITEMP(109), TURINT) 138
189 * (ITEMP(110), KSIM),(ITEMP(111), NSCAT) 139
190 * (ITEMP(112), TGTVEL),(ITEMP(201), TSCAT(1,1)) 140
191 EQUIVALENCE (ITEMP(170), NCELL),(ITEMP(195), TAVG)
192 EQUIVALENCE (ITEMP(145), GAIN)
193 EQUIVALENCE (ITEMP(180), SHTO),(ITEMP(181), SHPHAS) 142
194 * , (ITEMP(182), TJIT) 143
195 144
196 EQUIVALENCE (ITEMP(146), CFREQ), (ITEMP(147), XWLENG),
197 * (ITEMP(148), SEDENS)
198 EQUIVALENCE (ITEMP(149), RADIUS), (ITEMP(190), NSAM).
199 * (ITEMP(197), NUF2), (ITEMP(201), FSAM(1,1))
200 EQUIVALENCE (ITEMP(199), NSEC), (ITEMP(201), FFCDEF(1,1)).
201 * (ITEMP(251), FFCDEF(1,1))
202 EQUIVALENCE (ITEMP(171), JRNC),
203 * (ITEMP(172), JKSIM), (ITEMP(173), JMAZ),
204 * (ITEMP(174), JHGT), (ITEMP(175), JERP),
205 * (ITEMP(176), JFMBW), (ITEMP(177), JPW),
206 * (ITEMP(178), JFO), (ITEMP(179), JSTAKT),
207 * (ITEMP(156), JVEL), (ITEMP(157), JPERCU)
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

000000

000000

ARTIFLOW CHART SET - FWC/SCL RAUSIM

2-12

000000

0000

CONTENTS

000	NAMELIST/NE01/ ASSET,ASSET,ASSET	
000	NAMELIST/NE02/ SIMON,DETTIN,NIME,ILUMF	
000	NAMELIST/NE03/ NO,FSHIFT,ICINV,ICFOR,SIMON,NOMPT	150
000	NAMELIST/NE04/ NO,FSHIFT,ICINV,ICFOR,SIMON,NOMPT	
000	NAMELIST/NE05/ ILIM,ELIM,NHIST,NCFACK,NDFACK	
000	NAMELIST/NE06/ ILIM,ELIM,NHIST,NCFACK,NDFACK	
000	NAMELIST/NE07/ NCFACK	
000	NAMELIST/NE08/ NCFACK	
000	NAMELIST/NE09/ NCFACK	
000	NAMELIST/NE10/ NCFACK	
000	NAMELIST/NE11/ NCFACK	
000	NAMELIST/NE12/ NCFACK	
000	NAMELIST/NE13/ NCFACK	
000	NAMELIST/NE14/ NCFACK	
000	NAMELIST/NE15/ NCFACK	
000	NAMELIST/NE16/ NCFACK	
000	NAMELIST/NE17/ NCFACK	
000	NAMELIST/NE18/ NCFACK	
000	NAMELIST/NE19/ NCFACK	
000	NAMELIST/NE20/ NCFACK	
000	NAMELIST/NE21/ NCFACK	
000	NAMELIST/NE22/ NCFACK	
000	NAMELIST/NE23/ NCFACK	
000	NAMELIST/NE24/ NCFACK	
000	NAMELIST/NE25/ NCFACK	
000	NAMELIST/NE26/ NCFACK	
000	NAMELIST/NE27/ NCFACK	
000	NAMELIST/NE28/ NCFACK	
000	NAMELIST/NE29/ NCFACK	
000	NAMELIST/NE30/ NCFACK	
000	NAMELIST/NE31/ NCFACK	
000	NAMELIST/NE32/ NCFACK	
000	NAMELIST/NE33/ NCFACK	
000	NAMELIST/NE34/ NCFACK	
000	NAMELIST/NE35/ NCFACK	
000	NAMELIST/NE36/ NCFACK	
000	NAMELIST/NE37/ NCFACK	
000	NAMELIST/NE38/ NCFACK	
000	NAMELIST/NE39/ NCFACK	
000	NAMELIST/NE40/ NCFACK	
000	NAMELIST/NE41/ NCFACK	
000	NAMELIST/NE42/ NCFACK	
000	NAMELIST/NE43/ NCFACK	
000	NAMELIST/NE44/ NCFACK	
000	NAMELIST/NE45/ NCFACK	
000	NAMELIST/NE46/ NCFACK	
000	NAMELIST/NE47/ NCFACK	
000	NAMELIST/NE48/ NCFACK	
000	NAMELIST/NE49/ NCFACK	
000	NAMELIST/NE50/ NCFACK	
000	NAMELIST/NE51/ NCFACK	
000	NAMELIST/NE52/ NCFACK	
000	NAMELIST/NE53/ NCFACK	
000	NAMELIST/NE54/ NCFACK	
000	NAMELIST/NE55/ NCFACK	
000	NAMELIST/NE56/ NCFACK	
000	NAMELIST/NE57/ NCFACK	
000	NAMELIST/NE58/ NCFACK	
000	NAMELIST/NE59/ NCFACK	
000	NAMELIST/NE60/ NCFACK	
000	NAMELIST/NE61/ NCFACK	
000	NAMELIST/NE62/ NCFACK	
000	NAMELIST/NE63/ NCFACK	
000	NAMELIST/NE64/ NCFACK	
000	NAMELIST/NE65/ NCFACK	
000	NAMELIST/NE66/ NCFACK	
000	NAMELIST/NE67/ NCFACK	
000	NAMELIST/NE68/ NCFACK	
000	NAMELIST/NE69/ NCFACK	
000	NAMELIST/NE70/ NCFACK	
000	NAMELIST/NE71/ NCFACK	
000	NAMELIST/NE72/ NCFACK	
000	NAMELIST/NE73/ NCFACK	
000	NAMELIST/NE74/ NCFACK	
000	NAMELIST/NE75/ NCFACK	
000	NAMELIST/NE76/ NCFACK	
000	NAMELIST/NE77/ NCFACK	
000	NAMELIST/NE78/ NCFACK	
000	NAMELIST/NE79/ NCFACK	
000	NAMELIST/NE80/ NCFACK	
000	NAMELIST/NE81/ NCFACK	
000	NAMELIST/NE82/ NCFACK	
000	NAMELIST/NE83/ NCFACK	
000	NAMELIST/NE84/ NCFACK	
000	NAMELIST/NE85/ NCFACK	
000	NAMELIST/NE86/ NCFACK	
000	NAMELIST/NE87/ NCFACK	
000	NAMELIST/NE88/ NCFACK	
000	NAMELIST/NE89/ NCFACK	
000	NAMELIST/NE90/ NCFACK	
000	NAMELIST/NE91/ NCFACK	
000	NAMELIST/NE92/ NCFACK	
000	NAMELIST/NE93/ NCFACK	
000	NAMELIST/NE94/ NCFACK	
000	NAMELIST/NE95/ NCFACK	
000	NAMELIST/NE96/ NCFACK	
000	NAMELIST/NE97/ NCFACK	
000	NAMELIST/NE98/ NCFACK	
000	NAMELIST/NE99/ NCFACK	
000	NAMELIST/NE00/ NCFACK	

2-13

203 NAMELIST/NL204/ NS,SIMBW,DETFIN,NIME,ILDDUM,ISIMFO

204 NAMELIST/NL205/ NTYPEK

205 NAMELIST/NL206/ NTYPEK

206 NAMELIST/NL207/ NTYPEK

207 NAMELIST/NL301/ SIMBW,N2,FS,KFFO,ISDUMP,ICINV,ICFOR,SIMFO,WSCAN 176

208 * INKPLI,NGHMT,TIME,ANTAZO,ANTELO,AZEST,EL,ST,INPTAZ,NPTEL, 177

209 * ANTAZ,ANTEC,KNGCEL,TI,FI,LAMBDA,NPIS,PKI,IPKPI 178

210 NAMELIST/NL302/ NFCS,NAWPH,KWPH,WNUVEL,VCLANG,KNEXT,KNO,AZEXT, 179

211 * AZO,KM,ELEXT,ILLO,NN,RNGCEL,ILDDUM

212 * ,KK,NELKS,MODL,DELAZ,DELEL,ICFLG,DDPHK,KCELL

213 NAMELIST/NL303/ ST,RNG,LF,VOL,NSKF,NAUTL,TH,TL,IFCUL

214 NAMELIST/NL304/ ST,RNG,LF,VOL,NSKF,NAUTL,TH,TL,IFCUL

215 NAMELIST/NL305/ ST,RNG,LF,VOL,NSKF,NAUTL,TH,TL,IFCUL

216 NAMELIST/NL306/ ST,RNG,LF,VOL,NSKF,NAUTL,TH,TL,IFCUL

217 NAMELIST/NL307/ ST,RNG,LF,VIL,VOL,GEHL,NAUTL,TH,TL,IFCUL,NSKP

218 NAMELIST/NL308/ ST,RNG,LF,VIL,VOL,GEHL,NAUTL,TH,TL,IFCUL,NSKP

219 NAMELIST/NL309/ ST,RNG,LF,VIL,VOL,GEHL,NAUTL,TH,TL,IFCUL,NSKP

220 NAMELIST/NL310/ ST,RNG,LF,VIL,VOL,GEHL,NAUTL,TH,TL,IFCUL,NSKP

221 NAMELIST/NL401/ TFX,NTFN

222 NAMELIST/NL402/ TFX,NTFN

223 NAMELIST/NL403/ FFR,FFI,FFR,FBI,IFFRN,IFFRQ,IFFRI,IFFID,

224 * IFFRN,IFFRQ,IFFIN,IFBID,NBITDF,MUCLDF

225 NAMELIST/NL404/ FFR,FFI,FFR,FBI,IFFRN,IFFRQ,IFFRI,IFFID,

226 * IFFRN,IFFRQ,IFFIN,IFBID,NBITDF,MUCLDF

227 NAMELIST/NL405/ FFO,FFI,FBI,FB2,IFFON,IFFOL,IFFIN,IFFID,IFFIN, 196

228 * IFFID,IFFON,IFFOL,IFFIN,IFFID,IFFIN,IFFID,IFFIN,

229 NAMELIST/NL406/ FFO,FFI,FBI,FB2,IFFON,IFFOL,IFFIN,IFFID,IFFIN, 198

230 * IFFID,IFFON,IFFOL,IFFIN,IFFID,IFFIN,IFFID,IFFIN,

231 NAMELIST/NL407/ NZ,NP,SE,F2END,IPUL

07/11/77

PROJECT LISTING

AUTOFLOW CHART SET - TWO/SCL RAUSIM

CONTENTS

PAGE

CONTENTS

2-14....

100	NAMCELLIST/NL400/ IN,INF,SE,FZER,FPULL	
100	NAMCELLIST/NL400/ FECK	
100	NAMCELLIST/NL410/ FECK	
100	NAMCELLIST/NL410/ NTSAR,DX,RFEC,THE TAS,NKWS,NPSCG,IRPSCG,NPSCR,	203
100	* IRPSR,INCRM,IDDUMP,NPCLS,TSKLS	
100	NAMCELLIST/NL420/ PW, FMEW,NPWTX,SPW,NSUEP,DWTIM,KISTIM,	205
100	* FALLIN,ISTART,FCLDE,WTX,IDDUMP,SIMFO,VPEAK,CHIRP,FSTRT	
100	NAMCELLIST/NL410/ PW, FMEW,NPWTX,SPW,NSUEP,DWTIM,KISTIM,	207
100	* FALLIN,ISTART,PCLDE,WTX,IDDUMP,SIMFO,VPEAK,CHIRP,FSTRT	
100	NAMCELLIST/NL422/ NIAPS,SEC,FODEC,ITAF	
100	NAMCELLIST/NL430/ NIAPS,SEC,FODEC,ITAF	
100	NAMCELLIST/NL400/ DX,THE TAS,NKWS,NPSCG,IRPSCG,	
100	* NPCLS,ICOPCL,ICGNM,NPWTX,FMEW,WTX,KISTIM,FALLIN,	
100	* IDDUMP,TIMES	
100	NAMCELLIST/NL410/ NTSAR,DX,THE TAS,NKWS,NPSCG,IRPSCG,TCLINE,	
100	* ANGLE,TSKLS,IDDUMP,FCLDE,THE TAS,MOUTSR,FECK,TOLMN	
100	NAMCELLIST/NL420/ NTSAR,DX,THE TAS,NKWS,NPSCR,IRPSR,TCLINE,	
100	* ANGLE,IDDUMP,KCCLL,THE TAS,MOUTSR,FECK,TOLMN	
100	NAMCELLIST/NL400/ FFO,FF1,FB1,FB2,IFFON,IFFOD,IFFIN,IFFID,IFBIN,	217
100	* IFFID,IFFON,IFFOD,NOITDF,MODELF	
100	NAMCELLIST/NL400/ FFO,FF1,FB1,FB2,IFFON,IFFOD,IFFIN,IFFID,IFBIN,	219
100	* IFFID,IFFON,IFFOD,NOITDF,MODELF	
100	NAMCELLIST/NL400/ FFO,FF1,FB1,FB2,IFFON,IFFOD,IFFIN,IFFID,IFBIN,	221
100	* IFFID,IFFON,IFFOD,NOITDF,MODELF	
100	NAMCELLIST/NL400/ FFO,FF1,FB1,FB2,IFFON,IFFOD,IFFIN,IFFID,IFBIN,	223
100	* IFFID,IFFON,IFFOD,NOITDF,MODELF	
100	NAMCELLIST/NL400/ FECK	
100	NAMCELLIST/NL400/ FECK	
100	NAMCELLIST/NL400/ FECK	

```

261 NAMELIST/NL437/ FBCK
262 NAMELIST/NL440/ NCELL
263 NAMELIST/NL441/ NCELL
264 NAMELIST/NL451/ FBCK,FECDL,RECIFT,THEIAS,DX,NROWS
265 NAMELIST/NL452/ FBCK,FECDL,RECIFT,THEIAS,DX,NROWS
266 NAMELIST/NL453/ DX,THEIAS,NROWS,NIPSCG,IRPSCG,
267 * NPULS,NGOPLS,TCGNLM,NFWTX,FMBW,WTX,KISTIM,FALTIM,TIMESB
268 NAMELIST/NL454/ SIMFO,SPW,NSUBP,BPRI
269 NAMELIST/NL455/ DX,THEIAS,NROWS,TCGNLM,SPW
270 NAMELIST/NL456/ GAIN
271 NAMELIST/NL457/ GAIN
272 NAMELIST/NL458/ GAIN
273 NAMELIST/NL459/ TAVG
274 NAMELIST/NL460/ TAVG
275 NAMELIST/NL461/ NSEC,SE,FECDL,FECDL
276 NAMELIST/NL462/ NSEC,SE,FECDL,FECDL
277 NAMELIST/NL463/ RADIUS,NSAM,NHFZ,FSAM
278 NAMELIST/NL501/ IDUMP,HTGT,FTGT,ANGTGT,TORINT,KSIM,NSEAT,
279 * TSCAT,TGTVEL
280 NAMELIST/NL502/ IDUMP,HTGT,FTGT,ANGTGT,TORINT,KSIM,NSEAT,
281 * TSCAT,TGTVEL
282 NAMELIST/NL504/ NTSAR,DX,KFFO,THEIAS,NROWS,NIPSCG,IRPSCG,NBPSE, 235
283 * XIPSK,XNCFM,IDUMP,NPULS,TSKUS
284 NAMELIST/NL505/ IDUMP,NTSAR,DX,THEIAS,NROWS,NIPSK,XIPSK,TOLINE, 237
285 * AUTNT,AUTOP,ALETT,TSKUS,FECDL,THEIAS,MONTSE,RECIFT,FBCK,
286 * TELMEN
287 NAMELIST/NL506/ NSR,MODEPH,IPY,ICDLE,CHIRP,ESTRT,SPW,NSUBP,SWTIM,
288 * TLTART,SIMFO,VPEAK,FISTIM,FALTIM
289 NAMELIST/NL507/ NSR,MODEPH,IPY,ICDLE,CHIRP,ESTRT,SPW,NSUBP,SWTIM,

```


2-77

```

319      1614  ITLMP(1) = 0
320          JSTEP=0
321          ISIM=1
322          IFLAGG = 1
323          IFCFLG=0
324          VPLAK=1.0
325      C
326      2000 CONTINUE
327          IF( IFLAGG .EQ. 0 ) GO TO 2003
328          IFLAGG = 0
329          WRITE(6,2001) ISIM,ICFG
330      2001 FORMAT(1H1,'SIMULATION PASS',I4,'DATA LOAD FOR CONFIGURATION NUMBE273
331          *R',I4,' IS BEGINNING')
332      2003 CONTINUE
333          FSHIFT=0.0
334          IFC=0
335          IGDUMP=0
336          INPTF=0
337          READ(5,6) ICARD,III,MODULE
338      6  FORMAT( A6 , 3X, I3, 3X, I3 )
339          WRITE(6,16) ICARD,III,MODULE
340      16 FORMAT(////,1H ,A6 ,10X,I3,10X,I3)
341      C
342          IF( ICARD .EQ. NCLDEX ) GO TO 1600
343          IF( ICARD .EQ. EXEC ) GO TO 1605
344          IF( ICARD .EQ. LDEXEC ) GO TO 1610
345          IF( ICARD .EQ. ENDPAS ) GO TO 1500
346          IF( ICARD .EQ. ENDJOB ) GO TO 1800
347          IF( ICARD .EQ. ENDCFG ) GO TO 1700

```

000120

OBJECT LISTING

AUTOFLOW CHART SET - FWC/SCL RAUSIM

2-18

000130

CONTENTS

000140 IF(ICARD .EQ. MODIFY) GO TO 1710

000150 IF(ICARD .EQ. SKIP) GO TO 1710

000160 GO TO 1601F01F01

000170 IF(ICARD .EQ. STORX)

000180 IF(ICARD .EQ. STORX) GO TO 1610

000190 IF(ICARD .EQ. STORX) GO TO 1620

000200 IF(ICARD .EQ. STORX) GO TO 1625

000210 IF(ICARD .EQ. STORX) GO TO 1630

000220 IF(ICARD .EQ. STORX) GO TO 1635

000230 IF(ICARD .EQ. STORX) GO TO 1640

000240 IF(ICARD .EQ. STORX) GO TO 1645

000250 IF(ICARD .EQ. STORX) GO TO 1650

000260 IF(ICARD .EQ. STORX) GO TO 1655

000270 IF(ICARD .EQ. STORX) GO TO 1660

000280 IF(ICARD .EQ. STORX) GO TO 1665

000290 IF(ICARD .EQ. STORX) GO TO 1670

000300 IF(ICARD .EQ. STORX) GO TO 1675

000310 IF(ICARD .EQ. STORX) GO TO 1680

000320 IF(ICARD .EQ. STORX) GO TO 1685

000330 IF(ICARD .EQ. STORX) GO TO 1690

000340 IF(ICARD .EQ. STORX) GO TO 1695

000350 IF(ICARD .EQ. STORX) GO TO 1700

000360 IF(ICARD .EQ. STORX) GO TO 1705

000370 IF(ICARD .EQ. STORX) GO TO 1710

000380 IF(ICARD .EQ. STORX) GO TO 1715

000390 GO TO 1601F01F01

000400 IF(ICARD .EQ. STORX)

000410 IF(ICARD .EQ. STORX) GO TO 1710

316

000420 IF(ICARD .EQ. STORX)

```

377      C
378      31 CONTINUE
379          IF(INDEX.EQ.415.OR.INDEX.EQ.426.OR.INDEX.EQ.504) GO TO 32
380          IF(INDEX.EQ.505) GO TO 32
381          GO TO 34
382      32 DO 33 J=1,100
383          33 TSKLCS(J)=1.0
384      34 CONTINUE
385          IF(INDEX.LT.303.OR.INDEX.GT.320) GO TO 35
386          DO 36 J=1,16
387              ITEMPI(200+J)=BLANK
388          ITEMPI(250+J)=BLANK
389          ITEMPI(300+J)=BLANK
390      36 CONTINUE
391      35 CONTINUE
392      C
393          IF(INDEX.LE.100.OR.INDEX.GE.600) GO TO 2040
394          IF(INDEX.GT.100.AND.INDEX.LT.200) GO TO 100
395          IF(INDEX.GT.200.AND.INDEX.LT.300) GO TO 200
396          IF(INDEX.GT.300.AND.INDEX.LT.400) GO TO 300
397          IF(INDEX.GT.400.AND.INDEX.LT.500) GO TO 400
398          IF(INDEX.GT.500.AND.INDEX.LT.600) GO TO 500
399      C
400      C
401          100 INDEX=INDEX-100
402          GO TO(101,102,103,104,105,106,107,108,109,110,111,112,113,114,
403              * 115,116,117,118,119,120),INDEX
404          200 INDEX=INDEX-200
405          GO TO(201,202,203,204,205,206,207,208,209,210,211,212,213,214,

```

330

333

0001170

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NO

CONTENTS

2-10

```

401      * 215,216,217,218,219,220,221,222,223,224,225,226,227,228,229, 334
402      * 230,231,232,233,234,235,236,237,238,239,240,241,242,243),INDEX 335
403      300 INDEX=INDEX-300
404      GO TO (301,302,303,304,305,306,307,308,309,310,311,312,313),INDEX 337
405      400 INDEX=INDEX-400
406      GO TO (401,402,403,404,405,406,407,408,409,410,411,412,413,414, 339
407      * 415,416,417,418,419,420,421,422,423,424,425,426,427,428,429, 340
408      * 430,431,432,433,434,435,436,437,438,439,440,441,442,443,444, 341
409      * 445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,
410      * 460,461,462,463,464,465,466,467,468,469,470),INDEX
411      500 INDEX=INDEX-500
412      GO TO (501,502,503,504,505,506,507,508,509,510,511,512,513,
413      * 514,515),INDEX
414      L
415      101 READ(5,NL101)
416      IF (IADD1.EQ.224.AND.IADD1.GE.1) GO TO 1101
417      IADD1=1
418      1101 CONTINUE
419      CALL SDEKA(N00001,N00013,N00129,RRAND,XRNDM)
420      1102 CONTINUE
421      SIGMA=-2.0*SIGMA*SIGMA
422      ULEMEAN=0.5*UEAT
423      XUEAT=UEAT/3.63597304E10
424      IF (JKN0.EQ.0) JEND=1
425      CALL SDEKA(N00001,N00001,N00012,ITEMP,XRNDM)
426      WRITE(5,NL101)
427      GO TO 1000
428      L
429      1001 READ(5,NL101)

```

- 1) $\boxed{a = b} \quad \overline{Fa}$ produces $\boxed{a = b} \quad \boxed{\overline{Fa}} \quad \overline{Fb}$
- 2) $\boxed{\overline{Fa}} (a = b)$ produces $\boxed{\overline{Fa}} \quad \boxed{a = b} \quad Fb$

Though reduction is defined as a post processing operation, it is, in reality, more of a resolution operation and, therefore, should be executed within the resolution procedure or prior to other post processing operations.

(6) Ordering

Literals to the right of the right-most framed literal are free to be reordered. Reordering may not occur across framed literals without special care.

2.4 FACT DETERMINATION

OL-deduction, as has been stated, is a refutation procedure. A questionable assertion, that is, a query, is presented in negated form to the OL-deduction mechanism. Deductions are then generated and are sought to be refuted. A refutable deduction implies the existence of a collection of rules and facts (possibly a null collection) that refutes the negated query, and consequently, satisfies the assertion of the query. It is the collection of facts that is the answer to the positive query and that is provided by the inference system.

A resolved deduction is a clause consisting of only resolution literals; that is, it is a deduction clause that has been fully resolved.

Each resolution (framed) literal in a resolved deduction clause is a literal about which the information system can supply facts or is an acceptable inferred literal. An inferred literal will often be a simple binary proposition, signifying a yes or no; true or false; 0 or 1 value. In the more general case, where the framed literal is a complex predicate, the literal will tend to be a fact literal for which the information system will supply all facts. For example, given $\boxed{L(X,5)}$ where $L(X,5)$ means "X" is a worker in factory "5", then the information system will search out all workers in factory number 5 and supply these as facts satisfying the literal. We call this procedure "fact determination" When fact determination is applied to all of the fact literals of a completely resolved deduction, we then determine a largest subset of these facts that can satisfy the deduction as an entirety.

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FNU/SCL RADSIM

2-22

CARD NO	****	CONTENTS
404		208 READ(5,NL208)
405		WRITE(6,NL208)
406		GO TO 1206
407		209 READ(5,NL209)
408		WRITE(6,NL209)
409		1206 IF(NHIST .LE. 8000) GO TO 1000
470		WRITE(6,1210)
471		1210 FORMAT(' THE SPECIFIED NUMBER OF POINTS IN HISTOGRAM EXCEEDS
472		*8000 NHIST SET TO 8000 *)
473		NHIST = 8000
474		GO TO 1000
475	C	
476	C	
477		210 CONTINUE
478		READ(5,NL210)
479		WRITE(6,NL210)
480		GO TO 1000
481		211 READ(5,NL211)
482		WRITE(6,NL211)
483		GO TO 1000
484		212 CONTINUE
485		READ(5,NL212)
486		WRITE(6,NL212)
487		GO TO 1000
488		213 READ(5,NL213)
489		WRITE(6,NL213)
490		GO TO 1000
491	C	
492		214 READ(5,NL214)

```

976      WRITE(6,NL219)
977      GO TO 1000
978      215 READ(5,NL215)
979      WRITE(6,NL215)
980      GO TO 1000
981      216 READ(5,NL216)
982      WRITE(6,NL216)
983      GO TO 1217
984      217 READ(5,NL217)
985      WRITE(6,NL217)
986      217 IF (ADCF5.EQ.0.0) GO TO 5217
987      N=FIX(FS/ADCF5+0.49)
988      CRFS=ABS((ADCF5-(FS/FLGAT(N))))/FS)
989      IF (CRFS.EQ.1.E-04) GO TO 1000
990      ADCF5=FS/FLGAT(N)
991      WRITE(6,NL217) ADCF5
992      217 FORMAT(' THE SPECIFIED VALUE OF ( FS/ADCF5 ) NOT',
993      * ' AN INTEGER ...ADCF5 SET EQUAL TO', C15.0)
994      GO TO 1000
995      217 ADCF5=FS
996      GO TO 1000
997      218 READ(5,NL218)
998      WRITE(6,NL218)
999      GO TO 1000
1000     219 READ(5,NL219)
1001     WRITE(6,NL219)
1002     GO TO 1000
1003     C
1004     220 READ(5,NL220)

```

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RADSIM

2-24

CARD NO

CONTENTS

```

220          WRITE(6,NL220)
221          GO TO 1000
222          READ(5,NL221)
223          WRITE(6,NL221)
224          GO TO 1000
225          READ(5,NL222)
226          WRITE(6,NL222)
227          GO TO 1000
228          READ(5,NL223)
229          WRITE(6,NL223)
230          GO TO 1000
231          READ(5,NL224)
232          WRITE(6,NL224)
233          GO TO 1000
234          READ(5,NL225)
235          WRITE(6,NL225)
236          GO TO 1000
237          READ(5,NL226)
238          WRITE(6,NL226)
239          GO TO 1000
240          READ(5,NL227)
241          WRITE(6,NL227)
242          GO TO 1000
243          READ(5,NL228)
244          WRITE(6,NL228)
245          GO TO 1000
246          READ(5,NL229)
247          WRITE(6,NL229)
248          GO TO 1000

```

```

551      231 READ(5,NL231)
552          WRITE(6,NL231)
553          GO TO 1000
554      232 READ(5,NL232)
555          WRITE(6,NL232)
556          GO TO 1000
557      233 READ(5,NL233)
558          WRITE(6,NL233)
559          GO TO 1000
560      234 READ(5,NL234)
561          WRITE(6,NL234)
562          GO TO 1000
563      235 READ(5,NL235)
564          WRITE(6,NL235)
565          GO TO 1000
566      236 READ(5,NL236)
567          WRITE(6,NL236)
568          GO TO 1000
569      237 READ(5,NL237)
570          WRITE(6,NL237)
571          GO TO 1000
572      C
573      201 CONTINUE
574          ICOR=1
575          ICINV=1
576          READ(5,NL201)
577          IF (S.NL.0.0) TI=1.0/FS
578          IF (TI.NL.0.0.AND. FS.EQ.0.0) FS=1.0/TI
579          IF (KFFU.NL.0.0) LAMBDA= 0.2997928/FFU

```

2-25

08/22/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RAD/SIM

2-26

CARD NO.

CONTENTS

```

580      IF (F1.EQ.0.0) F1=FS/ELLAT(2**N2)
581      IF (SIMW.EQ.0.0.AND.SIMW.GT.FS) SIMW=FS
582      IF (RNOCELL.EQ.0.0.AND.SIMW.NE.0.0) RNOCELL=1.0/SIMW
583      IF (RNOCELL.EQ.1.1) RNOCELL=1
584      AZANG=AN(AZU+WSCAN*TIME
585      CLANG=ANTCLC
586      IF (F1.EQ.0.0) GO TO 2701
587      NF=FIX(S.MBW/F1)
588      IF (NF.LE.192) GO TO 2301
589      WRITE(6,3501)
590      3501 FORMAT(' THE REQUIRED NUMBER OF FREQ. DOMAIN SAMPLES EXCEEDS ')
591      *,' 192.....THIS JOB WILL BE TERMINATED.')
592      GO TO 1797
593      2301 WRITE(6,NL302)
594      GO TO 1000
595      3501 WRITE(6,11) NPREPT
596      IFC=FLD(6,6,1,XEC(1))
597      READ(11,IFC) ITEMPT
598      2301 IREPT=IREPT+1
599      IF (IREPT.GT.NPREPT) GO TO 1401
600      ICLUCK=ICLUCK+1
601      IFC=ICLUCK
602      IPRPT=IPRPT+1
603      IF (IPRPT.GT.NPRIS) IPRPT=1
604      TIME=TIME+PR1(IPRPT)
605      AZANG=ANTA2U+WSCAN*TIME
606      WRITE(11,IFC) ITEMPT
607      FLD(6,6,1,XEC(1))=IFC
608      GO TO 1500

```

```

009      11      FORMAT(' THIS CONFIGURATION WILL BE REPEATED',I3,
010              * 'TIMES')
011      302 READ(S,NL302)
012              IF (KNOCCL.LT.11) KNOCCL=11
013              CALL CLINT($I302)
014              WRITE(6,NL302)
015              GO TO 1000
016      303 READ(S,NL303)
017              WRITE(6,NL303)
018              GO TO 1000
019      304 READ(S,NL304)
020              WRITE(6,NL304)
021              GO TO 1000
022      305 READ(S,NL305)
023              WRITE(6,NL305)
024              GO TO 1000
025      306 READ(S,NL306)
026              WRITE(6,NL306)
027              GO TO 1000
028      307 READ(S,NL307)
029              WRITE(6,NL307)
030              GO TO 1000
031      308 READ(S,NL308)
032              WRITE(6,NL308)
033              GO TO 1000
034      309 READ(S,NL309)
035              WRITE(6,NL309)
036              GO TO 1000
037      310 READ(S,NL310)

```

2-27

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RADSTM

2-28

CONTENTS

CARD NO

036	WRITE(6,NL310)
037	GO TO 1000
040	C
041	401 READ(5,NL401)
042	WRITE(6,NL401)
043	GO TO 1000
044	402 READ(5,NL402)
045	WRITE(6,NL402)
046	GO TO 1000
047	C
048	403 READ(5,NL403)
049	WRITE(6,NL403)
050	GO TO 1000
051	404 READ(5,NL404)
052	WRITE(6,NL404)
053	GO TO 1000
054	405 READ(5,NL405)
055	WRITE(6,NL405)
056	GO TO 1000
057	406 READ(5,NL406)
058	WRITE(6,NL406)
059	GO TO 1000
060	C
061	407 READ(5,NL407)
062	WRITE(6,NL407)
063	GO TO 1000
064	408 READ(5,NL408)
065	WRITE(6,NL408)
066	GO TO 1000

```

007      409 READ(5,NL409)
008          WRITE(6,NL409)
009          GO TO 1000
010      410 READ(5,NL410)
011          WRITE(6,NL410)
012          GO TO 1000
013      413 READ(5,NL413)
014          WRITE(6,NL413)
015          GO TO 1000
016      C
017      420 READ(5,NL420)
018          WRITE(6,NL420)
019      1420 IF(RISTIM.LT.TI) RISTIM=TI
020          IF(FALTIM.LT.TI) FALTIM=TI
021          IF(SWTIM.LT.TI) SWTIM=TI
022          GO TO 1000
023      421 READ(5,NL421)
024          WRITE(6,NL421)
025          GO TO 1420
026      C
027      422 READ(5,NL422)
028          WRITE(6,NL422)
029          GO TO 1000
030      423 READ(5,NL423)
031          WRITE(6,NL423)
032          GO TO 1000
033      C
034      425 READ(5,NL425)
035          WRITE(6,NL425)

```



```

125      GO TO 1000
126      434 READ(5,NL434)
127      WRITE(6,NL434)
128      GO TO 1000
129      435 READ(5,NL435)
130      WRITE(6,NL435)
131      GO TO 1000
132      436 READ(5,NL436)
133      WRITE(6,NL436)
134      GO TO 1000
135      437 READ(5,NL437)
136      WRITE(6,NL437)
137      GO TO 1000
138      440 READ(5,NL440)
139      WRITE(6,NL440)
140      GO TO 1000
141      441 READ(5,NL441)
142      WRITE(6,NL441)
143      GO TO 1000
144      442 READ(5,NL442)
145      WRITE(6,NL442)
146      GO TO 1000
147      443 READ(5,NL443)
148      WRITE(6,NL443)
149      GO TO 1000
150      450 READ(5,NL450)
151      WRITE(6,NL450)
152      GO TO 1000
153      451 READ(5,NL451)
154      WRITE(6,NL451)
155      GO TO 1000
156      452 READ(5,NL452)
157      WRITE(6,NL452)
158      GO TO 1000
159      453 READ(5,NL453)
160      WRITE(6,NL453)
161      GO TO 1000
162      454 READ(5,NL454)

```

02/11/76

Input: 1121160

AUTOFLOW CHART SET - FWL/SCL KADSIM

2-32

CONTENTS

CARD NO

0001	WELL (8, NL454)
0002	CCLC 1000
0003	WELL (9, NL455)
0004	WELL (10, NL456)
0005	CCLC 1000
0006	WELL (11, NL457)
0007	WELL (12, NL458)
0008	CCLC 1000
0009	WELL (13, NL459)
0010	WELL (14, NL460)
0011	CCLC 1000
0012	WELL (15, NL461)
0013	WELL (16, NL462)
0014	CCLC 1000
0015	WELL (17, NL463)
0016	WELL (18, NL464)
0017	CCLC 1000
0018	WELL (19, NL465)
0019	WELL (20, NL466)
0020	CCLC 1000
0021	WELL (21, NL467)
0022	WELL (22, NL468)
0023	CCLC 1000
0024	WELL (23, NL469)
0025	WELL (24, NL470)
0026	CCLC 1000
0027	WELL (25, NL471)
0028	WELL (26, NL472)
0029	CCLC 1000
0030	WELL (27, NL473)
0031	WELL (28, NL474)
0032	CCLC 1000
0033	WELL (29, NL475)
0034	WELL (30, NL476)
0035	CCLC 1000
0036	WELL (31, NL477)
0037	WELL (32, NL478)
0038	CCLC 1000
0039	WELL (33, NL479)
0040	WELL (34, NL480)
0041	CCLC 1000
0042	WELL (35, NL481)
0043	WELL (36, NL482)
0044	CCLC 1000
0045	WELL (37, NL483)
0046	WELL (38, NL484)
0047	CCLC 1000
0048	WELL (39, NL485)
0049	WELL (40, NL486)
0050	CCLC 1000
0051	WELL (41, NL487)
0052	WELL (42, NL488)
0053	CCLC 1000
0054	WELL (43, NL489)
0055	WELL (44, NL490)
0056	CCLC 1000
0057	WELL (45, NL491)
0058	WELL (46, NL492)
0059	CCLC 1000
0060	WELL (47, NL493)
0061	WELL (48, NL494)
0062	CCLC 1000
0063	WELL (49, NL495)
0064	WELL (50, NL496)
0065	CCLC 1000
0066	WELL (51, NL497)
0067	WELL (52, NL498)
0068	CCLC 1000
0069	WELL (53, NL499)
0070	WELL (54, NL500)
0071	CCLC 1000
0072	WELL (55, NL501)
0073	WELL (56, NL502)
0074	CCLC 1000
0075	WELL (57, NL503)
0076	WELL (58, NL504)
0077	CCLC 1000
0078	WELL (59, NL505)
0079	WELL (60, NL506)
0080	CCLC 1000
0081	WELL (61, NL507)
0082	WELL (62, NL508)
0083	CCLC 1000
0084	WELL (63, NL509)
0085	WELL (64, NL510)
0086	CCLC 1000
0087	WELL (65, NL511)
0088	WELL (66, NL512)
0089	CCLC 1000
0090	WELL (67, NL513)
0091	WELL (68, NL514)
0092	CCLC 1000
0093	WELL (69, NL515)
0094	WELL (70, NL516)
0095	CCLC 1000
0096	WELL (71, NL517)
0097	WELL (72, NL518)
0098	CCLC 1000
0099	WELL (73, NL519)
0100	WELL (74, NL520)
0101	CCLC 1000
0102	WELL (75, NL521)
0103	WELL (76, NL522)
0104	CCLC 1000
0105	WELL (77, NL523)
0106	WELL (78, NL524)
0107	CCLC 1000
0108	WELL (79, NL525)
0109	WELL (80, NL526)
0110	CCLC 1000
0111	WELL (81, NL527)
0112	WELL (82, NL528)
0113	CCLC 1000
0114	WELL (83, NL529)
0115	WELL (84, NL530)
0116	CCLC 1000
0117	WELL (85, NL531)
0118	WELL (86, NL532)
0119	CCLC 1000
0120	WELL (87, NL533)
0121	WELL (88, NL534)
0122	CCLC 1000
0123	WELL (89, NL535)
0124	WELL (90, NL536)
0125	CCLC 1000
0126	WELL (91, NL537)
0127	WELL (92, NL538)
0128	CCLC 1000
0129	WELL (93, NL539)
0130	WELL (94, NL540)
0131	CCLC 1000
0132	WELL (95, NL541)
0133	WELL (96, NL542)
0134	CCLC 1000
0135	WELL (97, NL543)
0136	WELL (98, NL544)
0137	CCLC 1000
0138	WELL (99, NL545)
0139	WELL (100, NL546)
0140	CCLC 1000
0141	WELL (101, NL547)
0142	WELL (102, NL548)
0143	CCLC 1000
0144	WELL (103, NL549)
0145	WELL (104, NL550)
0146	CCLC 1000
0147	WELL (105, NL551)
0148	WELL (106, NL552)
0149	CCLC 1000
0150	WELL (107, NL553)
0151	WELL (108, NL554)
0152	CCLC 1000
0153	WELL (109, NL555)
0154	WELL (110, NL556)
0155	CCLC 1000
0156	WELL (111, NL557)
0157	WELL (112, NL558)
0158	CCLC 1000
0159	WELL (113, NL559)
0160	WELL (114, NL560)
0161	CCLC 1000
0162	WELL (115, NL561)
0163	WELL (116, NL562)
0164	CCLC 1000
0165	WELL (117, NL563)
0166	WELL (118, NL564)
0167	CCLC 1000
0168	WELL (119, NL565)
0169	WELL (120, NL566)
0170	CCLC 1000
0171	WELL (121, NL567)
0172	WELL (122, NL568)
0173	CCLC 1000
0174	WELL (123, NL569)
0175	WELL (124, NL570)
0176	CCLC 1000
0177	WELL (125, NL571)
0178	WELL (126, NL572)
0179	CCLC 1000
0180	WELL (127, NL573)
0181	WELL (128, NL574)
0182	CCLC 1000
0183	WELL (129, NL575)
0184	WELL (130, NL576)
0185	CCLC 1000
0186	WELL (131, NL577)
0187	WELL (132, NL578)
0188	CCLC 1000
0189	WELL (133, NL579)
0190	WELL (134, NL580)
0191	CCLC 1000
0192	WELL (135, NL581)
0193	WELL (136, NL582)
0194	CCLC 1000
0195	WELL (137, NL583)
0196	WELL (138, NL584)
0197	CCLC 1000
0198	WELL (139, NL585)
0199	WELL (140, NL586)
0200	CCLC 1000

```

177          WRITE(6,NL485)
178          GOTO 1000
179      C
180      501 READ(5,NL501)
181          WRITE(6,NL501)
182          GOTO 1000
183      502 READ(5,NL502)
184          WRITE(6,NL502)
185          GOTO 1000
186      504 CONTINUE
187          READ(5,NL504)
188          WRITE(6,NL504)
189          GOTO 1000
190      505 READ(5,NL505)
191          WRITE(6,NL505)
192          IF (NF.GT.4000) GOTO 2426
193          GOTO 1000
194      C
195      506 READ(5,NL506)
196          WRITE(6,NL506)
197          CALL PHENC(31506)
198          MODULE=420
199          GOTO 1420
200      507 READ(5,NL507)
201          WRITE(6,NL507)
202          CALL PHENC(31506)
203          MODULE=421
204          GOTO 1420
205      508 FODUC=SIMFO

```


041	204 CONTINUE
042	205 CONTINUE
043	206 CONTINUE
044	207 CONTINUE
045	220 CONTINUE
046	230 CONTINUE
047	C
048	312 CONTINUE
049	313 CONTINUE
050	C
051	414 CONTINUE
052	415 CONTINUE
053	416 CONTINUE
054	417 CONTINUE
055	418 CONTINUE
056	419 CONTINUE
057	431 CONTINUE
058	439 CONTINUE
059	442 CONTINUE
060	443 CONTINUE
061	445 CONTINUE
062	446 CONTINUE
063	447 CONTINUE
064	448 CONTINUE
065	449 CONTINUE
066	451 CONTINUE
067	C
068	502 CONTINUE
069	C

2-37

```

999      513  CONTINUE
900      514  CONTINUE
901      515  CONTINUE
902      2040 WRITE(6,51) JSTEP,MODULE
903          51 FORMAT( ' THE MODULE NUMBER FOR THE ',I3,'TH STEP IS INVALID...THE 699
904          * MODULE NUMBER IS ',I4)
905          JSTEP=JSTEP-1
906          IBLOCK=IBLOCK-1
907          GO TO 2000
908      C      ** LOAD DATA BLOCK AND EXEC WORD **
909      1000 WRITE (11,IFC) ITEMP
910      1001 ITEXEC= MODULE
911          ITEXEC= IPACK(12,ICW,ITEXEC)
912          ITEXEC= IPACK(24,IFC,ITEXEC)
913          ITEXEC(JSTEP)= IPACK(30,ICFG,ITEXEC)
914          GO TO 2000
915      C      ** ENDPAS **
916      1500 IBLOCK=IBLOCK+1
917          ITEXEC(500)=IBLOCK
918          WRITE (11,IBEXEC) ITEXEC
919          WRITE (6,61) IBEXEC,IFBLOCK
920      61  FORMAT(' IBEXEC = ',I10,' IFBLOCK = ',I10)
921          ITEXEC=IBLOCK
922          ISIM=ISIM+1
923          IF(INKEPT.EQ.0) GOTO 1501
924          IF(IKEPT.EQ.0) GOTO 5301
925          GOTO 6301
926      1501 IFLAGG=1
927          GO TO 2000

```

7042
7043
730

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWO/SCL RADSIM

2-38****

CARD NO	****	CONTENTS	
910		1530 WRITE(6,1530)	
911		1530 FORMAT(* MORE THAN 250 JOB STEPS PER SIMULATION PASS ATTEMPTED...734	
912		* THE SIMULATION ACTIVITY IS DELETED*)	
913		GO TO 1010	
914	C		
915		1005 INDEX = MODULE	
916		ICW = 1	
917		IFC=15000	
918		IF(1FCFLG.EQ.0) IFC=15000-1	
919		JSTEP = JSTEP+1	
920		IF(JSTEP.GT. 250) GO TO 1530	
921		GO TO 1050	
922	C		
923		1010 ICW=1	
924		GO TO 1011	
925		1010 ICW=2	
926		IFCFL=1	
927		IFCFLK = IFCFLK+1	
928		IF(1FCFLK.GT.40) GO TO 1799	
929		1011 INDEX = MODULE	
930		JSTEP=JSTEP+1	
931		IF(JSTEP.GT.250) GO TO 1530	
932		IFC=15000	
933		WRITE(6,1012) INDEX	752
934		1012 FORMAT(40X, * THE NEXT BLOCK OF DATA TO BE LOADED IS FOR MODULE NUM754	
935		*LEN*, IN //)	755
936		GO TO 31	756
937	C		
938		1020 CONTINUE	758

457	ICW=4
458	GO TO 1001
459	1020 CONTINUE
460	ICW=5
461	GO TO 1001
462	1025 CONTINUE
463	ICW=6
464	GO TO 1001
465	1030 CONTINUE
466	ICW=7
467	GO TO 1001
468	1035 CONTINUE
469	ICW=8
470	GO TO 1001
471	1040 CONTINUE
472	ICW=9
473	GO TO 1001
474	1045 CONTINUE
475	ICW=10
476	GO TO 1001
477	1050 CONTINUE
478	ICW=11
479	GO TO 1001
480	1055 CONTINUE
481	ICW=12
482	GO TO 1001
483	1060 CONTINUE
484	ICW=13
485	GO TO 1001

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RAUSIM

CARD NO

CONTENTS

2-40 ****

900	1005 CONTINUE
907	ICW=13
908	GL TO 1001
909	1070 CONTINUE
990	ICW=14
991	GL TO 1001
992	1075 CONTINUE
993	ICW=15
994	GL TO 1001
995	1080 CONTINUE
996	ICW=16
997	GL TO 1001
998	1085 CONTINUE
999	ICW=17
1000	GL TO 1001
1001	1090 CONTINUE
1002	ICW=18
1003	GL TO 1001
1004	1095 CONTINUE
1005	ICW=19
1006	GL TO 1001
1007	1730 CONTINUE
1008	ICW=20
1009	GL TO 1001
1010	1735 CONTINUE
1011	ICW=21
1012	GL TO 1001
1013	1740 CONTINUE
1014	ICW=22

2-41

```

1013          GO TO 1001
1014          1745 CONTINUE
1015          ICW=23
1016          GO TO 1001
1017          C          ** ENDCFG **
1018          1700 CONTINUE                                822
1019          ISIM=ISIM-1                                825
1020          WRITE(6,1701) ICFG,ISIM                    826
1021          1701 FORMAT(1H0,'THE DATA LOAD FOR CONFIGURATION NUMBER',I3,' HAS BEEN
1022          *COMPLETED.....',I3,' SIMULATION PASSES WERE LOADED') 828
1023          ICFG=ICFG+1
1024          GO TO 5                                      830
1025          C          ** MODIFY **
1026          1710 CONTINUE                                832
1027          JSTEP = 111                                833
1028          INDEX=MODULE                                834
1029          ICW=FLR(12,24)/JEXEC(JSTEP)
1030          IFC=FLD(16,2,IFXEC(JSTEP))
1031          IF(ICARD=EQ.SKIP) WRITE(6,1751) JSTEP
1032          1751 FORMAT(' STEP NUMBER',I3,' WILL BE BYPASSED FOR REMAINDER',
1033          * ' OF THIS CONFIGURATION')
1034          IF(ICARD=EQ.MODIFY) WRITE(6,1712) ISIM,IFC ,JSTEP,MODULE
1035          1712 FORMAT(1H ,*MOD FOR PASS NUMBER',I3,' DATA BLOCK',I3,' FOR J
1036          *CB STEP',I4,' CALLING MODULE',I4,' IS TO BE MODIFIED' ) 838
1037          C
1038          1740          IF( IFC .NE.0 ) GO TO 1720      839
1039          1741          WRITE(6,1714)                  840
1040          1714 FORMAT( ' 1BLOCK = 0 MODIFICATION NOT PERFORMED.' ) 841
1041          GO TO 2000

```

02/11/75

INDEX LISTING

AUTOFLOW CHART SET - FWC/SCL KADSIM

2-42 ****

CARD NO	****	CONTENTS	
1044	1720	READ (11*IFC) ITEMP	
1045		IBLOCK=IBLOCK+1	
1046		IFC=IBLOCK	
1047		IF (ICARD.EQ.MODIFY) GOTO 35	
1048		MODULE=0	
1049		GOTO 1001	
1050	C		851
1051	1502	WRITE(6,2502) IFCG	852
1052	1502	FORMAT(' NON-STANDARD RETURN FROM CLINT.....THE CONFIGURATIONS PRE	853
1053		*CEEDING*,15,* WILL BE EXECUTED*)	854
1054		GO TO 1797	855
1055	1506	WRITE(6,2506)	856
1056	2506	FORMAT(' NON-STANDARD RETURN FROM PHENC....PRECEDING	8520
1057		* CONFIGURATIONS WILL BE EXECUTED*)	8521
1058		GO TO 1797	
1059	1799	WRITE(6,1799) IFCG	859
1060	1791	FORMAT(' THE NUMBER OF DATA BLOCKS TO BE LOADED EXCEEDS STORAGE	860
1061		*AVAILABLE.....THE CONFIGURATIONS PRECEEDING*,15,* WILL BE EXECUTED*	861
1062		*)	862
1063	1797	CONTINUE	
1064	1800	CONTINUE	
1065		IFC=IFC-1	
1066		WRITE(6,1801) IFCG	866
1067	1801	FORMAT('THE DATA LOAD FOR*,15,* CONFIGURATIONS HAS BEEN COMPLE	867
1068		*TED.....SIMULATION ACTIVITY FOLLOWS*)	868
1069	1810	IEXEC(1)=0	
1070		WRITE (11*IBEXEC) IEXEC	
1071	C		873
1072		CALL EXIT	874
1073	C		875
1074		STOP	876
1075		END	

SECTION 3

SIMULATION CONTROLLER

(MAIN - 2)

The simulation controller serves as the switchboard which connects the simulation modules together in the manner prescribed by the control cards read by the simulation data loader. Figure 3-1 is a block diagram of the Simulation Load Module. The control word blocks and System/Module Parameter Tables are stored in DSRN #11 which was initialized during the simulation data loader activity. The clutter scatterer parameters are stored in DSRN #2 which was initialized during the simulation data loader activity. In addition, provisions are available for reading and writing data on user defined disc and magnetic tape data sets.

There are two storage arrays allocated for storing simulated signal data. These arrays are designated as XT and YT. In addition, two auxiliary arrays are allocated for storing either signal data, processed data such as probability distributions, or other miscellaneous data. These arrays are designated as XA and XB. All four arrays are composed of 8192 elements.

The following is a description of the sequence of events that occur in performing a step of the simulation.

1. The operation to be performed in the Jth step is defined by the control word stored in location J of the control word block. This word is compared with the control word dictionary to determine (1) the operation to be performed, (2) the reference number of the module to be executed, and (3) the Data Set Reference Number.
2. If the operation to be performed is the execution of a module with no input data requirements, then control is transferred to the designated module. If the operation to be performed is execution of a module requiring input data, then the System/Module Parameter table is loaded from DSRN #11 and control is transferred to the designated module. For operations involving data transfers the contents of the designated array are either loaded from or transferred to the DSRN which was specified.

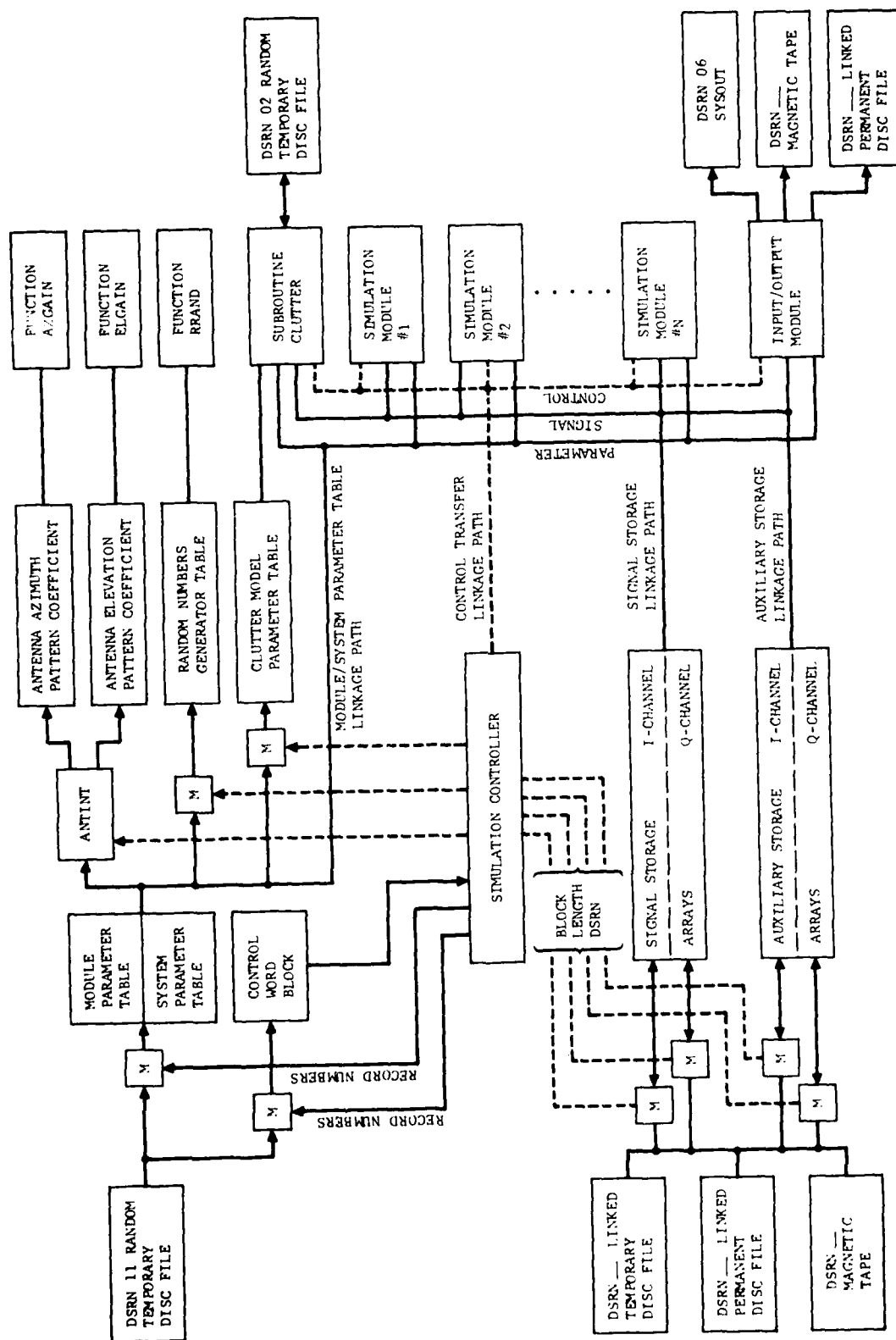


Figure 3-1 SIMULATION LOAD MODULE FUNCTIONAL BLOCK DIAGRAM

3. Once the operation for the Jth step is completed the job step counter is incremented by 1 and the process is repeated.

When a blank (all zeros) control word is encountered, the simulation pass is ended. The job step counter is reset to 1 and a new pass through the simulation is begun with a different set of parameters. A block of data containing all zeros is used to designate the end of a simulation configuration. When this is encountered the simulation controller will move the next block of data into the control word buffer. This new block of control words defines a new simulation configuration. When a new control word block containing all zeros is encountered, the simulation job is terminated.

Flow Chart: Page 9-34

Cross Reference Table: Page 9-210

1076 C * MAIN 2 * 3-4

1077 COMMON IT,TORIG,DELT,TDUM,XI(8193),ITY,TORIGY,DELTG,TDUMY,

1078 * YI(8193),IXA,XACRIG,XADEL,XADUM,XA(8193),IEXEC(500)

1079 COMMON/BLK1/IDATA(500)

1080 COMMON/BLK330/ICLCUN(30)

1081 COMMON/AZPAT/ CLEFAZ(304)

1082 COMMON/ELPAT/ CLEFEL(304)

1083 COMMON/EC0002/ IXB,XBCRIG,XBDEL,XBDUM,XB(8193)

1084 COMMON/BLKEND/ KNEUDAT(141) 80

1085 COMMON/SYS/ MODULE,ICW,IFC

1086 C

1087 DIMENSION XIT(516),YIT(516),XAT(516),XBT(516) 100

1088 C

1089 DIMENSION XD1(4010),XD2(4010),XB1(4010),XB2(4010)

1090 DIMENSION IXI(8192),IYT(8192)

1091 DIMENSION ANTAZ(150),ANTEL(150)

1092 DIMENSION XTM(4010),YTM(4010)

1093 C

1094 DATA AXI/' XI '/,AYI/' YI '/,AXA/' XA '/,AXB/' XB '/ 180

1095 DATA N193,N194,N195,N196/-1,-2,-1,0/

1096 C

1097 CHARACTER LABEL*16

1098 EQUIVALENCE (IEXEC, IEXEC(500))

1099 EQUIVALENCE (IDATA(316),LABEL)

1100 EQUIVALENCE (XI(1),IXI(1)),(YI(1),IYT(1))

1101 EQUIVALENCE (XIT(1), IT),(YIT(1), IY)

02/21/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

3-5
0000

CARD NO

CONTENTS

```

1102      *          ( XAT(1), IXA      ), ( XB1(1), IXB      )
1103      EQUIVALENCE (XD1(1),XA(1)),(XD2(1),XA(4011))
1104      EQUIVALENCE (XB1(1),XB(1)),(XB2(1),XB(4011))
1105      EQUIVALENCE (XTM(1),XT(4011)),(YTM(1),YT(4011))
1106      EQUIVALENCE (IDATA(150), AZBST ), (IDATA(151),NP1AZ),
1107      *          (IDATA(152), ELEST ), (IDATA(153),NP1EL),
1108      *          (IDATA(201),ANTAZ(1)), (IDATA(351),ANTEL(1))
1109      EQUIVALENCE (IDATA(160), MODEDF )
1110      EQUIVALENCE (IDATA( 1), RZ      ),(IDATA( 2), FS      ),
1111      *          (IDATA( 3), RFFG      ),(IDATA( 4), SIMBW      ),
1112      *          (IDATA( 5), ISDUMP      ),(IDATA( 6), ICINV      ),
1113      *          (IDATA( 7), ICFLR      ),(IDATA( 8), SIMFO      ),
1114      *          (IDATA( 9), NORMET      ),(IDATA(10), WSCAN      ),
1115      *          (IDATA(11), FI      ),(IDATA(12), TI      ),
1116      *          (IDATA(13), LAMBDA      ),(IDATA(14), KNGLEL      ),
1117      *          (IDATA(15), PRF      ),(IDATA(16), TIME      ),
1118      *          (IDATA(17), AZANG      ),(IDATA(18), ANTAZO      ),
1119      *          (IDATA(19), ELANG      ),(IDATA(20), ANTELO      )
1120      C
1121      DATA N00000,N00001,N00012,N00021,N00013,N00124,N00701
1122      * 70,1,12,21,13,124,201/
1123      C
1124      CALL KANC12(11,500)
1125      ITEXEC=1
1126      ILLCON(20)=0
1127      CALL FACF(100,1,1,0)
1128      ISDUMPE=1
1129      CNO  READ(11,'IDEXEC') ITEXEC
1130      WRITE(6,CNO) ITEXEC

```

3-6

```

1131      195 FORMAT(' IEXEC= ',I3)
1132      WRITE(6,95) (J, IEXEC(J), J=1,250)
1133      195 FORMAT(10,50X, 'IEXEC(',I3, ')= ',I3)
1134      194 IPASS=1
1135      ICFG=IFLG(0,0,IEXEC(1))
1136      IF(ICFG.EC.0.AND.IEXEC.EC.1) GO TO 2000
1137      IF(ICFG.EC.1) GO TO 2050
1138      GO TO 999
1139      900 CONTINUE
1140      WRITE(6,71) IPASS,ICFG
1141      71 FORMAT(10,' SIMULATION PASS NUMBER',I3,' OF CONFIGURATION NUMBER'
1142      ' ,I3,' HAS BEEN COMPLETED')
1143      IPASS=IPASS+1
1144      READ(11,1) IEXEC
1145      WRITE(6,96) IEXEC
1146      IF(ISSUMP.EQ.1.OR.ISSUMP.EQ.2) WRITE(6,95) (J,IEXEC(J),J=1,250)
1147      IF(ICFG.EC.EQ.1) GO TO 994
1148      994 CONTINUE
1149      WRITE(6,70) IPASS,ICFG
1150      70 FORMAT(10,' SIMULATION PASS NUMBER',I3,' OF CONFIGURATION NUMBER'
1151      ' ,I3,' IS BEGINNING')
1152      JSTEP=0
1153      1000 J=JSTEP+1
1154      1001 READ(1,4,12) IEXEC(JSTEP)
1155      1002 READ(1,4,14) IEXEC(JSTEP)
1156      1003 READ(1,4,16) IEXEC(JSTEP)
1157      IF(1000.EC.1) GO TO 900
1158      IF(MPM.EC.1)

```

550

730

740

00/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

3-7

CARD NO

CONTENTS

```

1160      CC TL (951,952,953,954,955,956,957,958,959,960,961,962,963,964,
1161      * 965,966,967,968,969,970,971,972,973),ICW
1162      951  CONTINUE
1163      952  TIMEUL=TIME
1164      AZANGU=AZANG
1165      ELANGU=ELANG
1166      READ(11,1FC) IDATA
1167      IF (MODULE=00,301) GOTO 986
1168      TIME=TIMEUL
1169      AZANG=AZANGU
1170      ELANG=ELANGU
1171      986  CONTINUE
1172      WRITE(6,2041)
1173      2041 FORMAT (1H1)
1174      WRITE(6,2042) MODULE,ICW,IFC,ICFG
1175      2042 FORMAT (' MODULE=',I10,' ICW=',I10,' IFC=',I10,' ICFG=',I10)
1176      IF (ISDUMP=0,2) WRITE(6,2040) (J,IDATA(J), J=1,500)
1177      2040 FORMAT(1H ,5(2X, 'IDATA(',I3, ')= ',G12))
1178      WRITE(6,20) ITEMP
1179      10 FORMAT(' THE NEXT SCHEDULED OPERATION HAS AN EXEC NUMBER OF ',I4)
1180      C
1181      IF (ITEMP.LT.303.GR.ITEMP.GT.310) GOTO 998
1182      ENCODE(LABEL,37) IPASS,ICFG
1183      37  FORMAT(' PASS=',I2,' ICFG=',I2)
1184      998  CONTINUE
1185      IF (ITEMP .LE. 100 .OR. ITEMP .GE. 200 ) GO TO 2100
1186      IF (ITEMP.GT.100.AND.ITEMP.LT.200) GO TO 100
1187      IF (ITEMP.GT.200.AND.ITEMP.LT.300) GO TO 200
1188      IF (ITEMP.GT.300.AND.ITEMP.LT.400) GO TO 300

```

900

3-9

```

1189      IF (ITEMP.GT.400.AND.ITEMP.LT.500) GO TO 400
1190      IF (ITEMP.GT.500.AND.ITEMP.LT.600) GO TO 500
1191      C
1192      C
1193      100 ITEM=ITEM-100
1194      GO TO(101,102,103,104,105,106,107,108,109,110,111,112,113,114,
1195      * 115,116,117,118,119,120),ITEM
1196      200 ITEM=ITEM-200
1197      GO TO(201,202,203,204,205,206,207,208,209,210,211,212,213,214,
1198      * 215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,
1199      * 230,231,232,233,234,235,236,237,238,239,240,241,242,243),ITEM
1200      300 ITEM=ITEM-300
1201      GO TO(301,302,303,304,305,306,307,308,309,310,311,312,313),ITEM
1202      400 ITEM=ITEM-400
1203      GO TO (401,402,403,404,405,406,407,408,409,410,411,412,413,414,
1204      * 415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,
1205      * 430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,
1206      * 445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,
1207      * 460,461,462,463,464,465,466,467,468,469,470),ITEM
1208      500 ITEM=ITEM-500
1209      GO TO(501,502,503,504,505,506,507,508,509,510,511,512),ITEM
1210      C
1211      C
1212      101 CONTINUE
1213      CALL SUBLKX(N00201,N00013,N00129,1,DATA,RNDDAT) 1240
1214      102 CALL SUBLKX(N00021,N00001,N00012,1,DATA,RNDDAT) 1250
1215      WRITE(6,1101) (J,RNDDAT(J), J=1,141)
1216      1101 FORMAT(1H , 4( 2X, 'RNDDAT(',I3, ') = ', U12)) 1270
1217      GO TO 1000

```

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RADSIM

CARD NO

CONTENTS

3-9

1210	103 CALL RTUPDB(XT,YT,XA,XB)
1219	GO TO 1000
1220	104 CALL XYTCDE(XT,YT,XA)
1221	GO TO 1000
1222	105 CALL XYTUM(XT,YT,XA)
1223	GO TO 1000
1224	106 CALL XYTUM2(XT,YT,XA)
1225	GO TO 1000
1226	108 CALL XYTODB(XT,YT,XT)
1227	GO TO 1000
1228	110 CALL RTUPM(XT,YT,XA,XB)
1229	GO TO 1000
1230	111 CALL RTCPM2(XT,YT,XA,XF)
1231	GO TO 1000
1232	113 CALL PLTFMT(XT,YT,XA,\$1070)
1233	GO TO 1000
1234	C
1235	114 CALL DBLXX(N193,8197,XT,XA)
1236	CALL DBLXX(N193,8197,YT,XB)
1237	GO TO 1000
1238	C
1239	115 CALL DBLXX(N193,8197,XA,XT)
1240	CALL DBLXX(N193,8197,XB,YT)
1241	GO TO 1000
1242	C
1243	116 CALL EFGYKL(XT)
1244	GO TO 1000
1245	117 CALL EFGYKL(YT)
1246	GO TO 1000

1450


```
1247      118  CALL ERGYCP(XT,YT)
1248          GOTO 1000
1249      L
1250      201  CALL DFT(XT,YT)
1251          GO TO 1000
1252      202  CONTINUE
1253          CALL ZFFT(XT,YT)
1254          GO TO 1000
1255      203  CONTINUE
1256          CALL ZIFFT(XT,YT)
1257          GO TO 1000
1258      204  CALL CONV(XT,YT,XA,XB,$1130)
1259          GO TO 1000
1260      205  CALL CONVP(XT,YT,XA,XB,$1130)
1261          GO TO 1000
1262      206  CALL DIVA(XT,YT,XA,XB,$1130)
1263          GO TO 1000
1264      207  CALL ADDA(XT,XA,$1130)
1265          GO TO 1000
1266      208  CALL CUMDIS(XT,XA)
1267          GO TO 1000
1268      209  CALL CUMDIS(YT,XA)
1269          GO TO 1000
1270      210  CALL LUTCUM(XB,XA)
1271          GO TO 1000
1272      211  CALL OUTCUM(XA,XA)
1273          GO TO 1000
1274      212  CALL PDF(XB,XA)
1275          GO TO 1000
```

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NO

CONTENTS

3-11

1276	213 CALL PDF(XA,XA)
1277	GO TO 1000
1278	214 CALL RNDARY(XT)
1279	GO TO 1000
1280	215 CALL RNDARY(YT)
1281	GO TO 1000
1282	216 CALL ATOD(XT,IXT)
1283	GO TO 1000
1284	217 CALL AILD(YT,IYT)
1285	GO TO 1000
1286	218 CALL DTDA(IXT,XT)
1287	GO TO 1000
1288	219 CALL DTDA(IYT,YT)
1289	GO TO 1000
1290	220 CALL DTDA(IYT,XA)
1291	GO TO 1000
1292	221 CALL WEITFE(XI,YT,\$1010)
1293	GO TO 1000
1294	222 CALL WEITCF(XT,YT,\$1010)
1295	GO TO 1000
1296	223 CALL WEITMP(XT,YT,\$1010)
1297	GO TO 1000
1298	224 CALL SHIFTXT,YT,XT,YT)
1299	GO TO 1000
1300	225 CALL SHIFTXT,YT,XA,XB)
1301	GO TO 1000
1302	226 CALL SHIFTS(XT,YT,XA,XF)
1303	GO TO 1000
1304	227 CONTINUE

1305	GO TO 972
1306	1227 CONTINUE
1307	GO TO 973
1308	2227 CALL SHIFTS(XT,YT,XA,XB)
1309	GO TO 1000
1310	228 CALL DTDA(IXT,XA)
1311	GO TO 1000
1312	229 CALL RSHIFT(XT,YT,XT,YT)
1313	GO TO 1000
1314	230 CALL RSHIFT(XT,YT,XA,XB)
1315	GO TO 1000
1316	231 CALL RSHFIS(XT,YT,XA,XB)
1317	GO TO 1000
1318	232 CONTINUE
1319	GO TO 972
1320	1232 GO TO 973
1321	2232 CALL RSHFIS(XT,YT,XA,XB)
1322	GO TO 1000
1323	233 CALL DFTRF(XT,YT)
1324	GO TO 1000
1325	234 CALL DFTRG(XT,YT)
1326	GO TO 1000
1327	235 CALL ADDRND(XT)
1328	GO TO 1000
1329	236 CALL ADDRND(YT)
1330	GO TO 1000
1331	237 CALL ADDRNDG(XT,YT)
1332	GO TO 1000
1333	238 CALL CLNV(XT,YT,XL1,XL2,\$1130)

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FMC/SCL RADSIM

CARD NO

CONTENTS

3-13

1334		GO TO 1000
1335	239	CALL ADDA(YT,XB,\$1130)
1336		GO TO 1000
1337	C	
1338	301	CONTINUE
1339		IF(ICFG.GT.1.AND.NPTAZ.EQ.0) GO TO 1302
1340		CALL ANTINT(NPTAZ,AZEST,ANTAZ,CDEF42)
1341	1302	IF(ICFG.GT.1.AND.NPTEL.EQ.0) GO TO 1000
1342		CALL ANT.NT(NPTEL,ELBST,ANTEL,CDEFEL)
1343		GO TO 1000
1344	302	CALL SDBLKX(46,1,13,10DATA,ICLCN)
1345		CALL SDBLKX(120,14,9,10DATA,ICLCN)
1346		GO TO 1000
1347	303	CALL PTLIST(XT)
1348		GO TO 1000
1349	304	CALL PTLIST(YT)
1350		GO TO 1000
1351	305	CALL PTLIST(XA)
1352		GO TO 1000
1353	306	CALL PTLIST(XB)
1354		GO TO 1000
1355	307	CALL PLUTTK(XT)
1356		GO TO 1000
1357	308	CALL PLUTTK(YT)
1358		GO TO 1000
1359	309	CALL PLUTTK(XA)
1360		GO TO 1000
1361	310	CALL PLUTTK(XB)
1362		GO TO 1000

```

1362      312 CALL SPCAVG(XA,31050)
1364          GO TO 1000
1366      313 CALL SCANNR(XA)
1368          GO TO 1000
1370      0
1372      0
1374      401 CALL NONLIN(XT,31050)
1376          GO TO 1000
1378      402 CALL NONLIN(YT,31050)
1380          GO TO 1000
1382      403 CALL CLIGFL(XT,YT)
1384          GO TO 1000
1386      404 CALL CDFNCL(XT,YT)
1388          GO TO 1000
1390      405 CALL KLIGFL(XT,YT)
1392          GO TO 1000
1394      406 CALL KDFNCL(XT,YT)
1396          GO TO 1000
1398      407 CALL FILT(XT,YT)
1400          GO TO 1000
1402      408 CALL FILT(XA,XE)
1404          GO TO 1000
1406      409 CALL INUTR(XT,XT)
1408          GO TO 1000
1410      410 CALL INUTR(YT,YT)
1412          GO TO 1000
1414      411 CALL ANTABY(XT,YT,31000)
1416          GO TO 1000
1418      412 CALL PWDET(XT,AT)

```

25003-14

2520

2700

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NO

CONTENTS

3-15

1392	GO TO 1000
1393	415 CALL FWDCT(YT,YT)
1394	GO TO 1000
1395	416 CALL FWDCT(XT,XT)
1396	GO TO 1000
1397	417 CALL FWDCT(YT,YT)
1398	GO TO 1000
1399	418 CALL SCLCT(XT,XT)
1400	GO TO 1000
1401	419 CALL SCLCT(YT,YT)
1402	GO TO 1000
1403	420 CALL FGENXY(XT,YT,\$1030)
1404	GO TO 1000
1405	421 CALL FGENMPLAT,YT,\$1030)
1406	GO TO 1000
1407	422 CALL DIGTFL(IXT,IXT)
1408	GO TO 1000
1409	423 CALL DIGTFL(IYT,IYT)
1410	GO TO 1000
1411	425 CALL CGCN(XT,YT,XD1,XD2,\$1040)
1412	GO TO 1000
1413	426 CALL TSARY(XT,YT,XD1,XD2,\$1020)
1414	GO TO 1000
1415	427 CALL TSARY1(XT,YT,XD1,XD2,\$1020)
1416	GO TO 1000
1417	430 GO TO(1430,2430),MODULOF
1418	1430 CALL MTJFLT(XT,XT)
1419	GO TO 1000
1420	1430 CALL MTJFLT(IXT,IXT)

28e0

1421 GO TO 1000
1422 431 GO TO(1431,2431),MODEOF
1423 1431 CALL MTIFLT(YT,YT)
1424 GO TO 1000
1425 2431 CALL MTIFLT(1YT,1YT)
1426 GO TO 1000
1427 432 GO TO(1432,2432),MODEOF
1428 1432 CALL MTINCL(XT,XT)
1429 GO TO 1000
1430 2432 CALL MTINCL(1XT,1XT)
1431 GO TO 1000
1432 433 GO TO(1433,2433),MODEOF
1433 1433 CALL MTINCL(YT,YT)
1434 GO TO 1000
1435 2433 CALL MTINCL(1YT,1YT)
1436 GO TO 1000
1437 434 CALL SWPINT(XT,XT)
1438 GO TO 1000
1439 435 CALL SWPINT(YT,YT)
1440 GO TO 1000
1441 436 CALL NCSWPI(XT,XT)
1442 GO TO 1000
1443 437 CALL NCSWPI(YT,YT)
1444 GO TO 1000
1445 438 CALL HELM(XT,XT)
1446 GO TO 1000
1447 438 CALL HELM(YT,YT)
1448 GO TO 1000
1449 439 CALL DEFAH(1XT,1XA)

02/11/75

INPUT LISTING

AUTOFLOW CHART SET - FHL/SCL KAUSIM

CARD NO.

CONTENTS

3-17

1450	GO TO 1000
1451	441 CALL UCPR(IYT,XB)
1452	GO TO 1000
1453	442 CALL IHLIM(IXT,IXT)
1454	GO TO 1000
1455	443 CALL IHLIM(IYT,IYT)
1456	GO TO 1000
1457	445 CALL IHMDT(IXT,IXT)
1458	GO TO 1000
1459	446 CALL IHMDT(IYT,IYT)
1460	GO TO 1000
1461	447 CALL IFMDT(IXT,IXT)
1462	GO TO 1000
1463	448 CALL IFMDT(IYT,IYT)
1464	GO TO 1000
1465	449 CALL ISMDT(IXT,IXT)
1466	GO TO 1000
1467	450 CALL ISMDT(IYT,IYT)
1468	GO TO 1000
1469	451 CALL KCLP(XT,YT,XA,XB)
1470	GO TO 1000
1471	452 CALL KCLTP(XT,YT)
1472	GO TO 1000
1473	453 CALL CGLNSP(XT,YT)
1474	GO TO 1000
1475	454 CALL FAPRM(XT,YT)
1476	GO TO 1000
1477	455 CALL CGLNUN(XT,YT)
1478	GO TO 1000


```

1474      456  CALL LAMPKE(XT,XT)
1480      GOTO 1000
1481      457  CALL LAMPKE(YT,YT)
1482      GOTO 1000
1483      458  CALL LAMPKE(XT,YT,XT,YT)
1484      GOTO 1000
1485      459  CALL LFAK(XT,XA)
1486      GOTO 1000
1487      460  CALL LFAK(YT,XB)
1488      GOTO 1000
1489      461  CALL DIOFIL(XT,YT)
1490      GOTO 1000
1491      462  CALL DIOFNL(XT,YT)
1492      GOTO 1000
1493      463  CALL DIOFSF(XT,YT)
1494      GOTO 1000
1495      C
1496      501  CALL TARGET(XT,YT)
1497      GO TO 1000
1498      502  CALL TGTNCL(XT,YT)
1499      GO TO 1000
1500      503  IF (ICLCN(20)-EQ.0) GO TO 1121
1501      CALL CLUTIA (XT,YT,XR1,XR2,$1120)
1502      GO TO 1000
1503      504  CALL ANIPAT(XT,YT,$1000)
1504      GO TO 1000
1505      505  CALL TSKPAT(XT,YT,XL1,XD2,XTM,YTM,$1162)
1506      GO TO 1000
1507      50  CALL PHDEC(XT,YT)

```

08/11/75

INPUT LISTING

AUTOFLUX CHART SET - FWD/SEL RADSIM

3-19

CARD NO

CONTENTS

1508		GOTO 1000
1509	509	CALL PHDEC(XA,XB)
1510		GOTO 1000
1511	510	CALL WVGUJ(EXT,YT)
1512		GOTO 1000
1513	511	CALL IONWS(AT,YT)
1514		GOTO 1000
1515	512	CALL ECM(EXT,YT)
1516		GOTO 1000
1517	C	
1518		107 CONTINUE
1519		109 CONTINUE
1520		112 CONTINUE
1521		114 CONTINUE
1522		120 CONTINUE
1523	C	
1524		240 CONTINUE
1525		241 CONTINUE
1526		242 CONTINUE
1527		243 CONTINUE
1528	C	
1529		311 CONTINUE
1530	C	
1531		411 CONTINUE
1532		412 CONTINUE
1533		424 CONTINUE
1534		426 CONTINUE
1535		429 CONTINUE
1536		444 CONTINUE

3-70

```

1537      C
1538      500 CONTINUE
1539      507 CONTINUE
1540      509 CONTINUE
1541      505 CONTINUE
1542      508 CONTINUE
1543      507 CONTINUE
1544      508 CONTINUE
1545      509 CONTINUE
1546      570 CONTINUE
1547      C
1548      WRITE(6,501) JSTEP, MODULE
1549      501 FORMAT ( ' NOTHING PERFORMED FOR ', I3, 'TH INPUT DATA ITEM. DATA I
1550      TEM = ', I4 )
1551      GO TO 1000
1552      C
1553      953 CONTINUE
1554      REWIND IFC
1555      WRITE(6,1953) IFC
1556      1953 FORMAT ( ' TEMPORARY FILE NUMBER', I5, ' HAS BEEN REWOUND' )
1557      GO TO 1000
1558      C
1559      954 CONTINUE
1560      WRITE(IFC) XTT
1561      IFC(17,612) WRITE(IFC) (XT(J),J=513,1T )
1562      WRITE(6,1954) XTT,IFC
1563      1954 FORMAT ( ' THE ARRAY', A6, 'HAS BEEN STORED IN DATA SET NO.', I5)
1564      GO TO 1000
1565      C

```

4140

4160

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FMU/SCL RADSIM

CARD NO

CONTENTS

3-21

1566	955 CONTINUE	
1567	WRITE(IFC) YIT	
1568	IF(ITY.GT.512) WRITE(IFC) (YT(J),J=513,ITY)	4210
1569	WRITE(6,1954) AYT,IFC	
1570	GO TO 1006	
1571	C	
1572	956 CONTINUE	
1573	WRITE(IFC) XAT	
1574	IF(IXA.GT.512) WRITE(IFC) (XA(J),J=513,IXA)	4270
1575	WRITE(6,1954) AXA,IFC	
1576	GO TO 1000	
1577	C	
1578	957 CONTINUE	
1579	WRITE(IFC) XBT	
1580	IF(IXB.GT.512) WRITE(IFC) (XB(J),J=513,IXB)	4330
1581	WRITE(6,1954) AXB,IFC	
1582	GO TO 1000	
1583	C	
1584	958 CONTINUE	
1585	READ(IFC) XTT	
1586	IF(ITY.GT.512) READ(IFC) (XT(J),J=513,ITY)	4390
1587	WRITE(6,1954) AXT,IFC	
1588	1950 FORMAT(' THE ARRAY',A6, 'HAS BEEN LOADED FROM DATA SET NO.',I5)	4410
1589	GO TO 1000	
1590	C	
1591	959 CONTINUE	
1592	READ(IFC) YIT	
1593	IF(ITY.GT.512) READ(IFC) (YT(J),J=513,ITY)	4460
1594	WRITE(6,1954) AYT,IFC	

3-22

1595	GO TO 1000	
1596	C	
1597	960 CONTINUE	
1598	READ(IFC) XAT	
1599	IF(IXA.GT.512) READ(IFC) (XA(J),J=513,IXA)	4520
1600	WRITE(6,1458) AXA,IFC	
1601	GO TO 1000	
1602	C	
1603	961 CONTINUE	
1604	READ(IFC) XBI	
1605	IF(IAB.GT.512) READ(IFC) (XB(J),J=513,IAB)	4580
1606	WRITE(6,1458) AXB,IFC	
1607	GO TO 1000	
1608	962 CONTINUE	
1609	GO TO 1000	
1610	963 CONTINUE	
1611	GO TO 1000	
1612	964 CONTINUE	
1613	GO TO 1000	
1614	965 CONTINUE	
1615	GO TO 1000	
1616	966 CONTINUE	
1617	GO TO 1000	
1618	967 CONTINUE	
1619	GO TO 1000	
1620	968 CONTINUE	
1621	GO TO 1000	
1622	969 CONTINUE	
1623	GO TO 1000	

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NL

CONTENTS

3-23

```

1624      C
1625      970 CONTINUE
1626      DO 1970 K=1,8197
1627      1970 XTT(K)=0.0
1628      11=2**N2
1629      DELT=11
1630      WRITE(6,2970) AXI
1631      2970 FORMAT(' THE ARRAY ',A6,' HAS BEEN SET TO 0.0 ')
1632      GO TO 1000
1633      971 CONTINUE
1634      DO 1971 K=1,8197
1635      1971 YTT(K)=0.0
1636      11Y=2**N2
1637      DELTY=11
1638      WRITE(6,2970) AYT
1639      GO TO 1000
1640      972 CONTINUE
1641      DO 1972 K=1,8197
1642      1972 XAT(K)=0.0
1643      WRITE(6,2970) AXA
1644      IF(MODULO.EQ.227) GO TO 1227
1645      IF(MODULO.EQ.232) GO TO 1237
1646      1XA=2**N2
1647      XADEL=11
1648      GO TO 1000
1649      973 CONTINUE
1650      DO 1973 K=1,8197
1651      1973 XET(K)=0.0
1652      WRITE(6,2970) AXE

```

4850

```

1003      IF (MODULO.EQ.227) GO TO 2227
1004      IF (MODULO.EQ.232) GO TO 2232
1005      IXB=Z**N2
1006      XBDCL=11
1007      GO TO 1000
1008      C
1009      1010 WRITE(6,1015)
1010      1011 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE WE11' )
1011      GO TO 2020
1012      C
1013      1020 WRITE(6,1025)
1014      1025 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE ISARY' )
1015      GO TO 2020
1016      C
1017      1030 WRITE(6,1035)
1018      1035 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE EGENXY' )
1019      GO TO 2020
1020      1040 WRITE(6,1045)
1021      1045 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE CGEN' )
1022      GO TO 2020
1023      C
1024      1050 WRITE(6,1055)
1025      1055 FORMAT( 1H , ' NONLINEAR TRANSFORM IMPROPERLY DEFINED ' )
1026      GO TO 2020
1027      C
1028      1060 WRITE(6,1065)
1029      1065 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE ANTARY' )
1030      GO TO 2020
1031      C

```

3-24

5111

5112

5113

5114

5141

06/11/75

INPUT LISTING

ACTIFLOW CHART SET - FWD/SCL RADSIM

3-25

LINE NO

CONTENTS

```

1002      1070 WRITE(6,1070)
1003      1075 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE PLTEMP' )
1004      GO TO 2020
1005      C
1006      1090 WRITE(6,1090)
1007      1095 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE SPCLAVG ' )
1008      GO TO 1000
1009      C
1010      1120 WRITE(6,1120)
1011      1125 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE CLUTTR' )
1012      GO TO 1000
1013      C
1014      1121 WRITE(6,1121)
1015      1122 FORMAT( 'THE CLUTTER MODEL HAS NOT BEEN PROPERLY INITIALIZED BY '
1016      *CLINT** *CLUTTR** WILL NOT BE EXECUTED.** )
1017      GO TO 2020
1018      C
1019      1130 WRITE(6,1130)
1020      1135 FORMAT( ' NON-STANDARD RETURN FROM SUBROUTINE CONV' )
1021      GO TO 2020
1022      C
1023      1132 WRITE(6,1132)
1024      1133 FORMAT( ' SUBROUTINE TSKPAT NOT EXECUTED ' )
1025      GO TO 2020
1026      C
1027      2100 WRITE(6,2100) JSTEP, MODULE, ITEMP
1028      2105 FORMAT( ' STEP ', I3, ' WITH MODULE = ', I4, ' HAS SCHEDULED AN OP
1029      *ERATION WITH EXEC NUMBER ', I3, ' WHICH IS OUT OF RANGE.' )
1030      GO TO 1000

```


3-26

```

1711      C
1712      2000 WRITE(6,2000)
1713      2005 FORMAT ('QA CONDITION EXISTS WHICH WILL CAUSE THE REMAINDER OF THE
1714              *S CONFIGURATION TO BE BYPASSED.*')
1715      GO TO 1790
1716      C
1717      2000 WRITE(6,2005)
1718      2005 FORMAT (' THIS JOB COMPLETE ' )
1719      CALL EXIT
1720      2000 WRITE(6,2005)
1721      2005 FORMAT(' THIS JOB ABORTED BECAUSE OF IMPROPER INPUT DATA',
1722              * '...SEE PRINT OUT FROM ACTIVITY 1')
1723      CALL EXIT
1724      STOP
1725      END

```

CALL DTOA (IX,X)

Where: IX contains the Input Waveform
X contains the Output Waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. If the computed digital output is greater than the largest number which can be specified by NBITS, it will be set to the maximum number.
- b. Flow Chart: Page 9-98
- c. Cross Reference Table: 9-220.

6. THEORY OF OPERATION

The analog-to-digital conversion is performed by dividing the floating point input signal by the value of the least significant digital bit. If round off is to be performed, 0.5 is then added to the value. The value thus obtained is converted to an integer number and its absolute value is tested against the maximum allowable number, $2^{*(NBITS-1)}$. If the magnitude of the signal is greater than the maximum allowable, it is set equal to the maximum. The basic mechanization equation for the ATOD module is:

$$IX(J) = IFIX(X(J)/XLSB + 0.5 * IROFF)$$

The digital-to-analog conversion is made by changing the input to a floating point number and multiplying it times the value of the least significant bit. The basic mechanization equation for the DTOA module is:

$$X(J) = FLOAT(IX(J)) * ZLSB$$

where: $ZLSB = 2^{-(N196)}$

4-4

```

2599      SUBROUTINE ATOD(X,IX)
2600      C
2601      C**** THIS SUBROUTINE PERFORMS AN ANALOG TO DIGITAL (REAL TO INTEGER)
2602      C          CONVERSION WITH SPECIFIED DIGITAL PRECISION AND SATURATION
2603      C          LEVEL, OR A DIGITAL TO ANALOG (INTEGER TO REAL) CONVERSION
2604      C          ON THE INPUT DATA ARRAY. *****
2605      C
2606      COMMON/BLK1/IK1(500)
2607      DIMENSION X(1),IX(1)
2608      EQUIVALENCE (IK1(103), XLSB      ),
2609      *           (IK1(104), NBITS    1, (IK1(105), IKOFF      )

```

08/11/75

INFCO LISTING

AUTOFLOW CHART 111 - FWC/SCL RADSIM

4-4a

CARD NO

CONTENTS

```

2010      *,          (BN1(144), ADCFS          )
2011      DATA N193,N194,N195,N196/-3,-2,-1,0/
2012      WRITE (6, 120)
2013      TIMADC=1.0/ABS(ADCFS)
2014      I1=X(N195)
2015      TIMADC=X(N194)
2016      TIME=TIMADC+I1*0.5
2017      N=IBOUL(X(N193))
2018      K=1
2019      IF (NE115 .LT. 31) GO TO 20
2020      NE115 = 31
2021      WRITE (6, 110) NE115
2022      GO CONTINUE
2023      MAX = 2**(NE115-1)
2024      RCF = 0.5*FLCAT(1/RCF)
2025      DXLSB = 1.0 / XLSB
2026      C
2027      C*** ANALOG TO DIGITAL CONVERSION *****
2028      C
2029      DO 40 J = 1, N
2030      IF (TIME.LT.TIMADC) GO TO 30
2031      IX(K)=IFIX(X(J)*DXLSB+RCF)
2032      C
2033      C*** TEST FOR CALCULATION *****
2034      C
2035      IF (IX(J).GT.MAX) IX(K)=123456789
2036      TIME=TIMADC+TIME
2037      K=K+1
2038      C

```

```

2039      DO      IF (ADCF5) 31,32,32
2040      31      IX(K)=IX(K-1)
2041              K=K+1
2042      32      TIME=TIME+1
2043      40      CONTINUE
2044      C
2045              IX(N193)=K-1
2046              IX(N194) = IBCDL( X(N194) )
2047              IX(N195)=IBCDL(TIADC)
2048              IF (ADCF5.LT.0.0) IX(N195)=BCDL(TI)
2049              IX(N196) = IBCDL( XLSB )
2050      RETURN
2051      ENTRY DIDA(IX,X)
2052      WHILE (0, 150)
2053              X(N193) = BCDL(IX(N193))
2054              X(N194) = BCDL(IX(N194))
2055              X(N195) = BCDL(IX(N195))
2056              N = IX(N193)
2057              ZLSB=BCDL(IX(N196))
2058      C
2059      ***** DIGITAL TO ANALOG CONVERSION *****
2060      C
2061              DO 60 J= 1, N
2062              X(J) = FLOCAT(IX(J)) * ZLSB
2063      60      CONTINUE
2064      RETURN
2065      IIO FORMAT ('# NBITS IS EXCESSIVE. THE VALUE OF NBITS HAS BEEN SET TO
2066      *%,1 ')
2067      IAD FORMAT(/// 49X,'* * * * A TO D CONVERTER * * * *' / / )
2068      IDA FORMAT(/// 49X,'* * * * D TO A CONVERTER * * * *' / / )
2069      END

```

SUBROUTINE CDIGFL

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
CDIGFL	LTI-4	403
CDFNCL	LTI-4	404

2. PURPOSE:

This subroutine is used to simulate a cross-coupled digital filter and synthesize a filter transfer function having one complex pole and/or one complex zero.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>I</u>	<u>Description</u>
FFR	0	F	Feed - Forward coefficient - R
FFI	0	F	Feed - Forward coefficient - I
FBR	0	F	Feedback coefficient - R
FB1	0	F	Feedback coefficient - I

4. CALLING SEQUENCES:

CALL CDIGFL (X,Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

The storage registers (TX1 and TY1) are cleared before execution begins.

CALL CDFNCL (X,Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

The storage registers, TX1 and TY1, are not cleared before execution begins.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. Flow Chart: Page 9-96

b. Cross Reference Table: 9-220

6. THEORY OF OPERATION

The block diagram of the complex digital filter simulated by this module is shown in Figure CDIGFL-1. The Z-plane transfer function is given by the following expression:

$$T(Z) = \frac{Z - (FFR + jFFI)}{Z - (FBR + jFBI)}$$

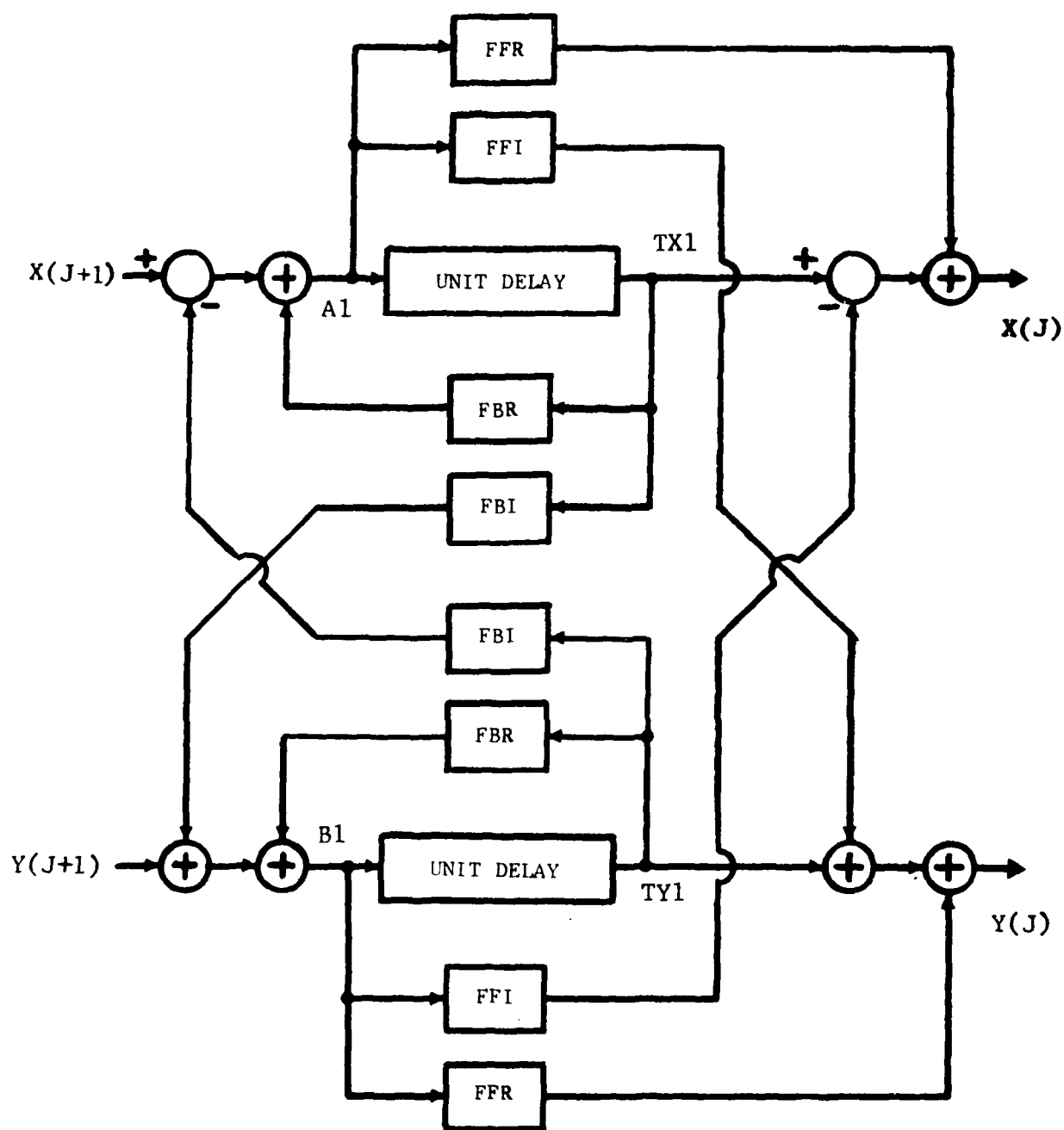


Figure CDIGFL-1 BLOCK DIAGRAM OF CDIGFL/CDFNCL

2501	SUBROUTINE CDFNCL(X,Y)	4-9
2502	COMMON/CLK1/CLK1(500)	
2503	EQUIVALENCE (BK1(65), FFX), (BK1(65), FFXY),	470
2504	* (BK1(70), FFX), (BK1(71), FFXY)	480
2505	DATA N153/50/	490
2506	DIMENSION X(1),Y(1)	
2507	TX1=0.0	510
2508	TY1=0.0	520
2509	ENTRY CDFNCL(X,Y)	
2510	N=BLUE(X(N153))	
2511	DO 10 J=1,N	570
2512	A1=X(J)+TX1*FFX-TY1*FFXY	
2513	B1=Y(J)+TY1*FFX+TX1*FFXY	
2514	X(J)=A1*FFX-B1*FFXY+TX1	
2515	Y(J)=B1*FFX+A1*FFXY+TY1	
2516	TX1=A1	620
2517	TY1=B1	621
2518	10 CONTINUE	622
2519	RETURN	623
2520	END	624

SUBROUTINE CFAR

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
CFAR	NL-2	459,460

2. PURPOSE:

This subroutine is used to simulate a constant false alarm rate (CFAR) video processor.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
TAVG	R	F	Width of averaging window used in determining video gain ($TAVG \geq 2*TI$)

4. CALLING SEQUENCES:

CALL CFAR(VIN,VOUT)

Where: VIN contains the Input Waveform
VOUT contains the Output Waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- The input and output arrays cannot be equivalenced.
- Flow Chart: Page 9-66
- Cross Reference Table: Page 9-216

6. THEORY OF OPERATION

The block diagram of the CFAR processor simulated by this module is shown in Figure CFAR-1. Each sample of the input waveform is passed through a variable gain amplifier which is controlled by the average value

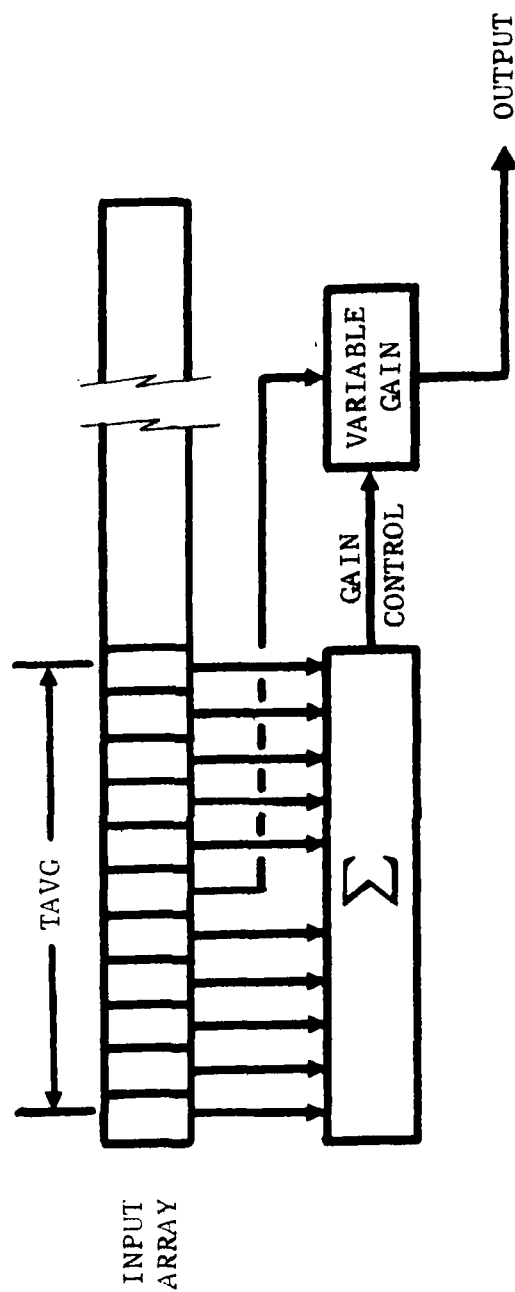


Figure CFAR-1 BLOCK DIAGRAM OF CFAR

of this waveform surrounding the sample being processed. The gain applied to the Jth sample is given by the following expression:

$$G(J) = \frac{2 * NCELL2}{\sum_{K = J - NCELL2}^{K = J + NCELL2} (|VIN(K)| - |VIN(J)|)}$$

where: $NCELL2 < J \leq NPTS - NCELL2$

$NCELL2 = \text{IFIX}(TAVG * 0.5 / TI)$

TI = independent variable increment between samples

NPTS = number of samples in the input array

```

2000      SUBROUTINE CFAR(VIN,VOUT)
2001      COMMON/BLK1/ BK1(200)
2002      DIMENSION VIN(1),VOUT(1)
2003      EQUIVALENCE (BK1(196), TAVG )
2004      DATA N143,N144,N145/-3,-2,-1/
2005      NCELL2=IFIX(TAVG*0.5/VIN(N145))
2006      XCELL=2*NCELL2
2007      NPTS=BLD(L(VIN(N143)))
2008      NSTOP=NPTS-NCELL2
2009      NCELL1=NCELL2+1
2010      AV=0.0
2011      DO 100 J=1,NCELL2
2012      AV=AV+ABS(VIN(J))
2013      100  CONTINUE
2014      GF=1.0
2015      DO 200 J=1,NPTS
2016      IF(J.LE.NSTOP) AV=AV+ABS(VIN(J+NCELL2))
2017      IF(J.GT.NCELL1) AV=AV-ABS(VIN(J-NCELL1))
2018      DVSR=AV-ABS(VIN(J))
2019      IF(DVSR.GT.1.0E-20) GF=XCELL/DVSR
2020      VOUT(J)=VIN(J)*GF
2021      200  CONTINUE
2022      VOUT(N143)=VIN(N143)
2023      VOUT(N144)=VIN(N144)
2024      VOUT(N145)=VIN(N145)
2025      RETURN
2026      END

```

SUBROUTINE CLUTTR

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
CLUTTR	LTI-3 or LTV-3	503

2. PURPOSE:

This subroutine generates the transfer function of a clutter volume. A pictorial diagram of the clutter volume is shown in Figure CLUTTR-1.

3. INPUT PARAMETERS:

All required input data is entered v'ia the clutter model initialization subroutine (CLINT).

4. CALLING SEQUENCES:

CALL CLUTTR(XT,YT,GAZ,GEL)

WHERE: XT contains the Output Impulse response - R

YT contains the Output Impulse response - I

GAZ-Temporary storage for antenna azimuth gain values (MM=number of storage cells required)

GEL-Temporary storage for antenna elevation gain values (NN - number of storage cells required)

5. RESTRICTIONS, REQUIREMENTS, AND MISCELLANEOUS DATA

- The subroutines CLINT must have been successfully executed prior to calling CLUTTR. See p 6-4 for restrictions on CLINT.
- The functions AZGAIN and ELGAIN are used to compute the antenna gain associated with each scatterer.

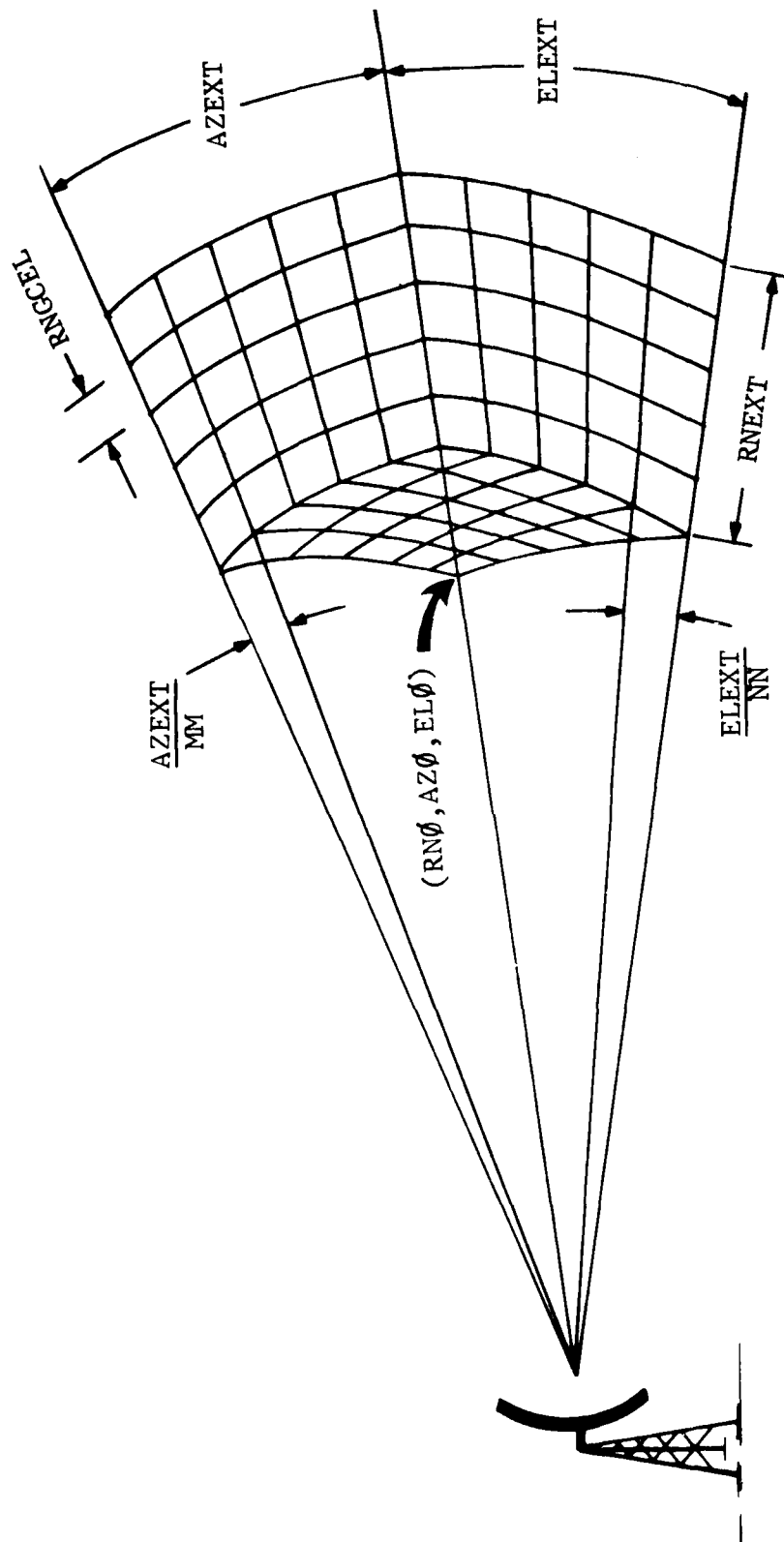


Figure CLUTTR-1 CLUTTER MODEL GEOMETRY

c. Flow Chart: Page 9-112

d. Cross Reference Table: Page 9-222

6. THEORY OF OPERATION

The clutter model is based on the premise that a volume of clutter can be represented by an ensemble of discrete scatterers. The RCS of these scatterers is derived from a theoretical probability density function, but can be easily modified to allow the use of a density function derived from measurements data. The phase behavior of the scatterer is determined from motion of the scatterer due to wind and from any inherent frequency spread derived from measurements data. The basic mechanization equation for the clutter model is given by the following expressions:

a. MODE = 1 (stationary clutter):

$$XT(K) + jYT(K) = \sum_{N=1}^{NN} GEL(N) \sum_{M=1}^{MM} GAZ(M) [CLUX(IPNT) + jCLUY(IPNT)]$$

where: GEL(N) - Normalized antenna gain corresponding to an elevation angle of $EL000 + DELEL * (N-1)$

GAZ(M) - Antenna gain corresponding to an azimuth angle of $AZ000 + DELAZ * (M-1)$

IPNT - Scatterer element pointer which is determined as follows:
 $IPNT = (K*(N-1)(M-1) + (N-1)*MM + M) \text{ MODULO } 250$

CLUX() - Array containing real part of clutter scatterer specification

CLUY() - Array containing imaginary part of clutter scatterer specification

b. MODE = 2 (Time varying clutter):

$$XT(K) + jYT(K) = \sum_{N=1}^{NN} GEL(N) \sum_{M=1}^{MM} GAZ(M) [CLUX(PDNT) + jCLUY(IDNT)] * e^{PHDEL}$$

"New" $CLUX(IDNT) + jCLUY(PDNT)$

Where: PHDEL = is the phase term to be applied to each scatter

$$= 57.29578 * \text{DELTIM} * \text{DOPFRQ} * \cos \left(\left| \text{XVLANG} - \text{DELAZ} * (M-1) \right| / 57.29567 \right) + \text{PHRW}$$

DELTIM = Radar interpulse period

PHRW = The random walk phase. This variable is generated by a random number generator from a uniform distribution.

$$-\frac{\text{RWPH}}{2.0} < \text{PHRW} < \frac{\text{RWPH}}{2.0}$$

The pulse phase shift due to doppler is computed via a recursive loop which is more efficient than a direct evaluation of the cosine function.

$$\begin{aligned} C(N+1) &= C(N) * \text{DELCS} - S(N) * \text{DELSN} \\ S(N+1) &= C(N) * \text{DELSN} + S(N) * \text{DELCS} \end{aligned}$$

where C = Phase shift due to doppler

$$\begin{aligned} C(1) &= \text{DOPFRQ} * \text{DELTIM} * 2\pi * \cos(\text{XVLANG}) \\ S(1) &= \text{DOPFRQ} * \text{DELTIM} * 2\pi * \sin(\text{XVLANG}) \end{aligned}$$

The clutter scatterer parameters (CLUX and CLUY) are updated by PHDEL for each execution of the clutter model and the new values are stored in the random disc file.

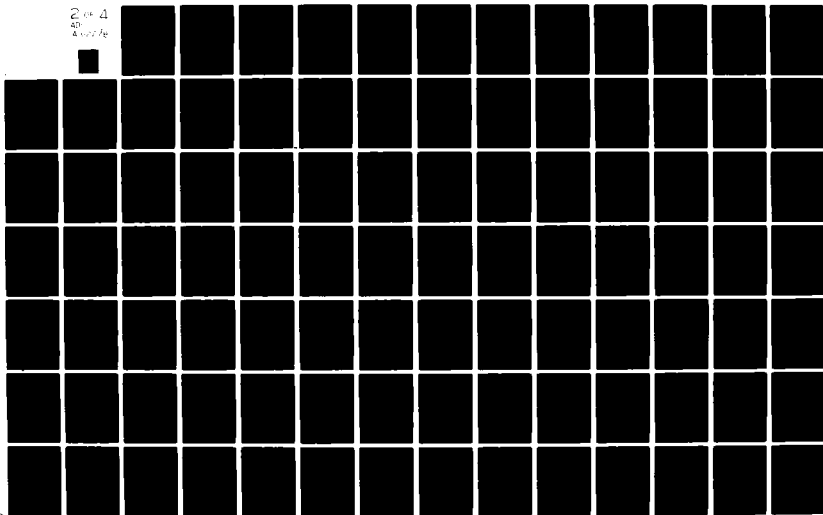
AD-A102 278

GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV F/G 17/9
ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING, VOLUME II. PAR--ETC(U)
JUN 76 R J HANCOCK, F H CLEVELAND F30602-73-C-0380

UNCLASSIFIED

RADC-TR-76-186-VOL-2-PT-1 NL

2 of 4
40
4/10/76



```

2955 SUBROUTINE CLUTTR(XT,YT,GAZ,GEL,*)
2956 COMMON/BLK1/ BK1(200)
2957 COMMON/BLK330/BK4(30)
2958 DIMENSION GAZ(1),GEL(1),CLUX(250),CLUY(250)
2959 COMMON/BLKRRND/ IIRI(12),NRAND(129)
2960 DIMENSION XT(1),YT(1),CSCAT(500),IRAND(128)
2961 EQUIVALENCE (CLUX(1),CSCAT(1)) , (IRAND(1),NRAND(12))
2962 EQUIVALENCE (CLUY(1),CSCAT(251))
2963 EQUIVALENCE (BK1( 14), TCELL      ),(BK1( 21), IOMP      ),
2964 *          (BK1( 17), TI          ),(BK1( 24), FS          ),
2965 *          (BK4( 3), RWPH         ),
2966 *          (BK4( 6), RNEXT        ),
2967 *          (BK4( 7), RNUDD        ),(BK4( 9), AZ000        ),
2968 *          (BK4( 10), MM          ),(BK4( 12), EL000        ),
2969 *          (BK4( 13), NN          ),(BK4( 14), KK          ),
2970 *          (BK4( 16), MLDE        ),(BK4( 17), DELAZ        ),
2971 *          (BK4( 18), DELEL        ),(BK4( 19), XLANG        ),
2972 *          (BK4( 20), ICFLG        ),(BK4( 21), DUFFAQ        ),
2973 *          (BK4( 22), KCELL        )
2974 EQUIVALENCE (BK1( 16), TIME      )
2975 DATA N195,N194,N195,N196/-3,-2,-1,0/
2976 DATA UTIME,IMULT,F12/0.0,1220703125,6.2831853/
2977 CALL RAND12(12,500)
2978
2979
2980 WRITE(6,10) (BK4(J),J=1,30)
2981
2982 DO 10 I=1,10
2983   I1=I*8X,C12=8X,C12=8X,C12=8X,C12=8X,C12=8X,C12=8X,C12=8X,C12=8X,C12=8X
2984   RWPHC=RWPH*F12/( 260.0*(2.0**35))
2985   DELAZK=DELAZ/57.24578
2986   DELTIME=TIME-UTIME
2987   UTIME=TIME
2988   U= DUFFAQ*DELTIME*F12
2989   CUF=CUF+(XLANG-DELAZK)

```

000
4-18a

```

2987      ZL=0.514*(ZVLAND-DELTAZK)
2988      DELC=0.514*DELCAZK)
2989      DELD=0.514*DELDZK)
2990      ANGLE=AL000
2991      DO 100 M=1,MM
2992      CAZ(M)=AZCAIN(ANGLE)
2993      ANGLE=ANGLE+DELTAZ
2994      100 CONTINUE
2995      IF (100*DELCAZ) WRITE(6,1001) (CAZ(J),J=1,MM)
2996      ANGLE=AL000
2997      DO 100 M=1,MM
2998      CEL(M)=ELCAIN(ANGLE)
2999      ANGLE=ANGLE+DELEL
3000      100 CONTINUE
3001      IF (100*DELEL) WRITE(6,1001) (CEL(J),J=1,MM)
3002      DO 200 J=1,NCULL
3003      XI(J)=0.0
3004      YI(J)=0.0
3005      200 CONTINUE
3006      C
3007      RTIME=0.5*TI
3008      IREC=1
3009      ENBW=0.0
3010      IFNI=1
3011      FLAC(2*IREC)=USCAT
3012      DO 500 K=1,KK
3013      A=0.0
3014      B=0.0
3015      DO 600 N=1,NN

```

01/21/77

IRREG LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

4-19

CARD NO

CONTENTS

3016 C=0
 3017 S=0
 3018 GO TO C=1,MM
 3019 IF (IRNT.LT.750) GO TO 750
 3020 GO TO (725,730),MODE
 3021 750 WRITE (2,IRNT) CSCAT
 3022 725 IRFC=IRFC+1
 3023 IRAT (2,IRFC) CSCAT
 3024 IRNT=1
 3025 GO C=0,2 (M)*CELL (N)
 3026 GO TO (760,770),MODE
 3027 760 A=A+CELL (IRNT)*C
 3028 E=E+CELL (IRNT)*C
 3029 GO TO 600
 3030 770 CONTINUE
 3031 IF (RWPH.LT.0.0) GO TO 780
 3032 IRND=IRND*(MAU)
 3033 IRN=IRND*JEND*IMULT
 3034 IRN=FLU (1,35,IRN)
 3035 JRND=IRND
 3036 IRAND (MAU)=IRND
 3037 MAU=FLU (15,7,IRND)
 3038 RHEW=FLUAT (IRND)*RWPHC
 3039 780 CONTINUE
 3040 IF (L=0.0) GO TO 790
 3041 TIME=L
 3042 C=C*DELS-S*DELSN
 3043 S=TIME*DELSN+S*DELS
 3044 790 CONTINUE

4-19a

```

3042      FPHULL=FPHN*U
3043      VCCS=(COS(FPHULL)
3044      VSIN=SIN(FPHULL)
3045      TIME=CLUX(IPNT)
3046      CLUX(IPNT)=TIME*VCCS-CLUY(IPNT)*VSIN
3047      CLUY(IPNT)=TIME*VSIN+CLUY(IPNT)*VCCS
3048      A=A+G*CLUX(IPNT)
3049      U=U+G*CLUY(IPNT)
3050      DOO CONTINUE
3051      IPNT=IPNT+1
3052      DOO CONTINUE
3053      DOO CONTINUE
3054      CLC=FIX(NTIME*ES)
3055      XT(CL)=A
3056      YT(CL)=U
3057      TIME=TIME+CLCL
3058      DOO CONTINUE
3059      C
3060      IF(MOD(LC,2) WRITE(2,1REC) C$CAT
3061      XT(N192)=MOD(FCELL)
3062      XT(N194)=E.N000
3063      XT(N192)=T1
3064      YT(N192)=XT(N192)
3065      YT(N194)=XT(N194)
3066      YT(N195)=XT(N195)
3067      1001 FORMAT(1H ,6I20.5)
3068      IF(LDME.LC.1) WRITE(6,1001) (XT(J), YT(J), J=1,KCELL)
3069      RETURN
3070      END

```

SUBROUTINE DCFAR

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
DCFAR	NL-2	440,441

2. PURPOSE:

This subroutine is used to simulate a digital constant false alarm rate (CFAR) video processor.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NCELL	R	I	Number of cells to be averaged in computing the video gain.

4. CALLING SEQUENCES:

CALL DCFAR (IN, IOUT)

Where: IN contains the Input Waveform
IOUT contains the Output Waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- The input and output arrays cannot be equivalenced.
- All calculations are in integer format.
- Flow Chart: Page 9-93
- Cross Reference Table: Page 9-220

6. THEORY OF OPERATION

The block diagram of the digital CFAR processor simulated by this module is shown in Figure DCFAR-1. Each sample of the input waveform is passed through a variable gain amplifier which is controlled by the average value of the waveform surrounding the sample being processed. The gain applied to the Jth sample is given by the following expression:

$$IG = \frac{NCELL}{\sum_{K=J-NCELL2}^{K=J+NCELL2} |IN(K)| - |IN(J)|}$$

where: $NCELL\ 2 < J \leq NPTS - NCELL2$

$NCELL\ 2 = NCELL/2$

$NPTS$ = Number of input samples in the input array

The value of the Jth output sample is computed as follows:

$$IOUT(J) = IN(J) * IG$$

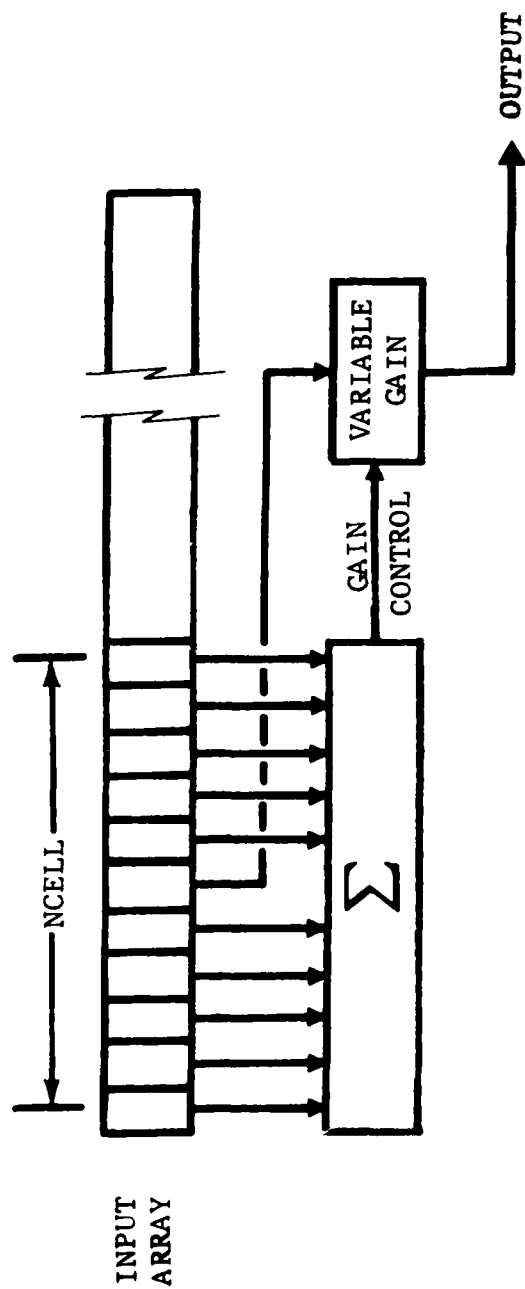


Figure DCFAR-1 BLOCK DIAGRAM OF DCFAR

```

2515      SUBROUTINE GCFAB(IN,IOUT)
2516      COMMON/CLK1/ CLK1(200)
2517      DIMENSION IN(1),IOUT(1)
2518      CLC1=VALLOC - CLK1(170), NCELL
2519      DATA N193,N194,N195,N196/-3,-2,-1,0/
2520      NCELL2=NCELL/2
2521      NCELL=NCELL*2
2522      NPTS=40, N193
2523      NCTP=NPTS-NCELL
2524      IAV=1
2525      DO 100 J=1,NCELL2
2526      IAV=IAV+IN(J)
2527      100 CONTINUE
2528      NCELL1=NCELL2+1
2529      DO 200 J=1,NPTS
2530      IF (J.LE.NCTP) IAV=IAV+IABS(IN(J+NCELL2))
2531      IF (J.GT.NCELL1) IAV=IAV-IABS(IN(J-NCELL1))
2532      ICF=IAV-IABS(IN(J))
2533      IF (ICF.LE.0) ICF=1
2534      IOUT(J)=(IN(J)*NCELL)/ICF
2535      200 CONTINUE
2536      IOUT(N193)=IN(N193)
2537      IOUT(N194)=IN(N194)
2538      IOUT(N195)=IN(N195)
2539      IOUT(N196)=IN(N196)
2540      RETURN
2541      END

```

SUBROUTINE DFT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
DFT	LTI-2	201
DFTFØ	LTI-2	234
DFTRF	LTI-2	233
DFTNCL	LTI-2S	None

2. PURPOSE:

This subroutine computes the finite discrete Fourier transform of a sequence of δ -functions.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
SIMFØ	0	F	Center frequency of discrete output spectrum (DFTFØ only)
FI	R	F	Frequency increment between samples of the output spectrum
FEXT	R	F	Width of output spectrum
NIMP	R	I	Number of δ -functions to be transformed
IDMP	R	I	Diagnostic data dump parameter = 0; no diagnostic data = 1; dump internal parameters: PS, DELPS, NPTS, NIMP, DIN = 2; dump output spectrum: PS, DELPS, NPTS, NIMP, DIN
FØ	0	F	Center frequency of discrete output spectrum (DFTRF only)
INORM	R	I	Normalization Flag = 0; $f_N = T_I$ = 1; $f_N = 1.0/NIMP$ = 2; $f_N = T_I$ = 3; $f_N = 1.0/NIMP$ = 4; $f_N = 1.0$

<u>Name</u>	<u>O/R</u>	<u>T</u>	Description
TI	O	F	Time increment between simulated waveform samples. Used in this subroutine for normalization only (required only if INORM = 0 or 2)
DIN	R	F	Array containing the parameters of the δ -functions. The specification for the Jth δ -function is the following: DIN(1,J) = phase angle DIN(2,J) = time of occurrence DIN(3,J) = amplitude

4. CALLING SEQUENCES:

Video spectrum

CALL DFT (X,Y)

Where: X contains the Output Waveform - R
Y contains the Output Waveform - I

The range of the independent variable, f , used in computing the spectrum is:

$$- \frac{\text{FEXT}}{2} \leq f < \frac{\text{FEXT}}{2}$$

IF spectrum

CALL DFTFØ (X,Y)

Where: X contains the Output Waveform - R
Y contains the Output Waveform - I

The range of the independent variable, f , used in computing the spectrum is:

$$\text{SIMFØ} - \frac{\text{FEXT}}{2} \leq f < \text{SIMFØ} + \frac{\text{FEXT}}{2}$$

RF spectrum

CALL DFTRF (X,Y)

where: X contains the Output Waveform - R

Y contains the Output Waveform - I

The range of the independent variable, f , used in computing the spectrum is:

$$F0 - \frac{FEXT}{2} \leq f < F0 + \frac{FEXT}{2}$$

Transform additional sample

CALL DFTNCL (X,Y)

Where: X contains the Input Waveform - R

Y contains the Input Waveform - I

X contains the Output Waveform - R

Y contains the Output Waveform - I

The computed spectrum is added to the spectrum contained in arrays X and Y. The range of the independent variable is determined by the previous execution of DFT, DFTF0, or DFTRF.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. This subroutine is quite time consuming and therefore should be used only when necessary. For most applications the AFFT subroutine which uses the Fast Fourier Transform algorithm should be used. The execution time for this subroutine can be estimated using the following expression

$$\frac{FEXT}{FI} \times NIMP \times 50 \mu S$$

- b. This subroutine is used normally in conjunction with other subroutines such as PHDEC and CGEN.
- c. The output spectrum has a center frequency of zero, i.e. it is the low pass equivalent.
- d. Flow Chart: Page 9-142
- e. Cross Reference Table: Page 9-226
- f. DFT lets user generate 100 delays to be considered.

6. THEORY OF OPERATION

The Fourier transform of a sequence of samples represented by S-functions is given by the following expression:

$$S(f) = F_N \int_{-\infty}^{\infty} \left[\sum_{m=1}^{NIMP} a(m) e^{j\phi(m)} \delta(t-t(m)) \right] e^{-j2\pi ft} dt$$

where: F_N is the normalization factor determined by the flag INORM.

Interchanging the order of integration and summation the following result is obtained:

$$S(f) = F_N \sum_{m=1}^{NIMP} a(m) e^{j\phi(m)} e^{-j2\pi ft(m)}$$

If only K equally spaced samples of $S(f)$ are computed, the following expression results:

$$S_k = S_*(f) \Big|_{f=kFI+f_0} = F_N \sum_{m=1}^{NIMP} a(m) e^{j\phi(m)} e^{-j2\pi kFI t(m)}$$

where: $K = \frac{FEXT}{FI}$ and f_0 is defined in the following manner for each module name:

$$\text{DFT: } f_0 = -\frac{FEXT}{2}$$

$$\text{DFTF}\phi: f_0 = \text{SIMF}\phi - \frac{FEXT}{2}$$

$$\text{DFTRF: } f_0 = F\phi - \frac{FEXT}{2}$$

```

3097      SUBROUTINE DEF(X,Y)
3098      COMMON/BLK1/ BK1(200),DIN(3,100)
3099      EQUIVALENCE (SIMFG,BK1(5)), (F1,BK1(11)), (FEX1,BK1(4)),
3100      *           (NIMP,BK1(200)), (IDMP,BK1(21)), (FO,BK1(5))
3101      *           , (INCRM ,BK1(9)), ( T1 ,BK1(12))
3102      DIMENSION      X(1) , Y(1)
3103      DATA N193,N194,N195,PI2,D1/-3,-2,-1,6.2831853,2.777777UL-03/
3104      FCENT=0.0
3105      GO TO 10
3106      C
3107      ENTRY DEFEG(X,Y)
3108      FCENT=SIMFG
3109      GO TO 10
3110      C
3111      ENTRY DETRI(X,Y)

```

4-28

1650

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SEL KAUSIM

4-28a

CARD NO

CONTENTS

```

3712          FCENT=F0
3713          C
3714          10 CONTINUE
3715          NPTS = IFIX(FEXT / F1 )
3716          NPTS2= NPTS/2
3717          NPTS= 2*NPTS2
3718          DO 50 J=1,NPTS
3719          X(J)=0.0
3720          Y(J)=0.0
3721          50 CONTINUE
3722          X(N193)=BDCL(NPTS)
3723          X(N194)=-F1*FLUAT(NPTS2)
3724          X(N195)=F1
3725          Y(N193)=X(N193)
3726          Y(N194)=X(N194)
3727          Y(N195)=X(N195)
3728          C
3729          ENTRY DFNCL(X,Y)
3730          IF (10MP) 60,60,55
3731          55 CONTINUE
3732          WRITE(6,1007) ((DIN(K,J),K=1,3),J=1,NIMP)
3733          1007 FORMAT(1H ,6E15.5)
3734          60 CONTINUE
3735          XN=F1
3736          IF (INORM.EQ.1.OR.INORM.EQ.3) XN=1.0/FLUAT(NIMP)
3737          IF (INORM.EQ.4) XN=1.0
3738          DO 200 J=1,NIMP
3739          A=DIN(3,J)*XN
3740          T=DIN(2,J)

```



```

2141      T1=-T(1,1)*COS(1,1)*D1
2142      DELPH=-T1*1
2143      IPASS=1
2144      N=INT(T1*4)
2145      K=DELPH
2146      C
2147      1) CONTINUE
2148      PS=(PH-AINT(PH))*PIZ
2149      CLEPS=(DELPH-AINT(DELPH))*PIZ
2150      CS=COS(PS)*A
2151      SN=SIN(PS)*A
2152      ACDEL=COS(CLEPS)
2153      ASDEL=SIN(CLEPS)
2154      IF (TEMP.EQ.0) GO TO 70
2155      WRITE(6,100)PS,DELPH,NFIS,NIRF
2156      CALL FORMAT(' T= ',1PE15.0, ' DELPH= ',1PE15.0, ' NIS= ',110, ' NIRF= ',110)
2157      70 CONTINUE
2158      GO TO 1-156157
2159      X(R)=X(R)+CS
2160      Y(R)=Y(R)+SN
2161      TEMP=CS
2162      C1=CS*ACDEL-1)*ASDEL
2163      C2=SN*ACDEL+TEMP*ASDEL
2164      C=C1*DEL
2165      GO TO 156157
2166      156157
2167      IF (TEMP.EQ.0) GO TO 700
2168      IF (N=0)

```

00/11/75

INPUT LISTING

AUTIFLOW CHART SET - FWC/SOL KAUSIM

4-29a

CARD NO.

CONTENTS

```

0000      K=NPISL
0001      KDEL=-1
0002      PH=PH-DELEH
0003      DELEH=DELEH
0004      GO TO 71
0005      200 CONTINUE
0006      C
0007      IF((LM-2) CO,TS,CO
0008      75 CONTINUE
0009      WRITE(6,1000) (X(M),Y(M), M=1,NPTS)
0010      1000 FORMAT(100, 0020.5)
0011      10 CONTINUE
0012      C
0013      RETURN
0014      END
  
```

2430

SUBROUTINE DIGFIL

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
DIGFIL	LTI-4	461
DIGFNC	LTI-4	462
ECMFL	LTI-4	None

2. PURPOSE:

This subroutine is used to simulate a multiple section digital filter.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NP	R	I	Number of 2-delay sections
SF	R	F	Scale factor
FB	0	F	Array containing the feedback coefficients. The coefficients for the Kth section are the following: FB(1,K) = 1-delay feedback coef. FB(2,K) = 2-delay feedback coef.
FF	0	F	Array containing the feed-forward coefficients. The coefficients for the Kth section are the following: FF(1,K) = 0-delay feed-forward coef. FF(2,K) = 1-delay feed-forward coef.

4. CALLING SEQUENCES:

CALL DIGFIL (X,Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

The storage register arrays (XM and YM)
 are cleared before execution begins.

CALL DIGFNC (X,Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

The storage register arrays (XM and YM)
 are not cleared before execution begins.

CALL ECMFL (X,Y)

Where: X contains the Input Waveform sample - R
 Y contains the Input Waveform sample - I
 X contains the Output Waveform sample - R
 Y contains the Output Waveform sample - I

One complex sample processed for each
 execution of the module.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. Flow Chart: Page 9-101
- b. Cross Reference Table: Page 9-221

6. THEORY OF OPERATION

The block diagram of one section of the digital filter simulated by this module is shown in Figure DIGFIL-1. The Z-plane transfer function for this section is given by the following expression:

$$T_K(Z) = FF(1,K) \frac{Z^2 + \frac{FF(2,K)}{FF(1,K)} Z + \frac{1}{FF(1,K)}}{Z^2 - FB(1,K) Z - FB(2,K)}$$

The Z-plane transfer function for the complete filter is given by the following expression:

$$T(Z) = SF \prod_{K=1}^{K=NP} T_K(Z)$$

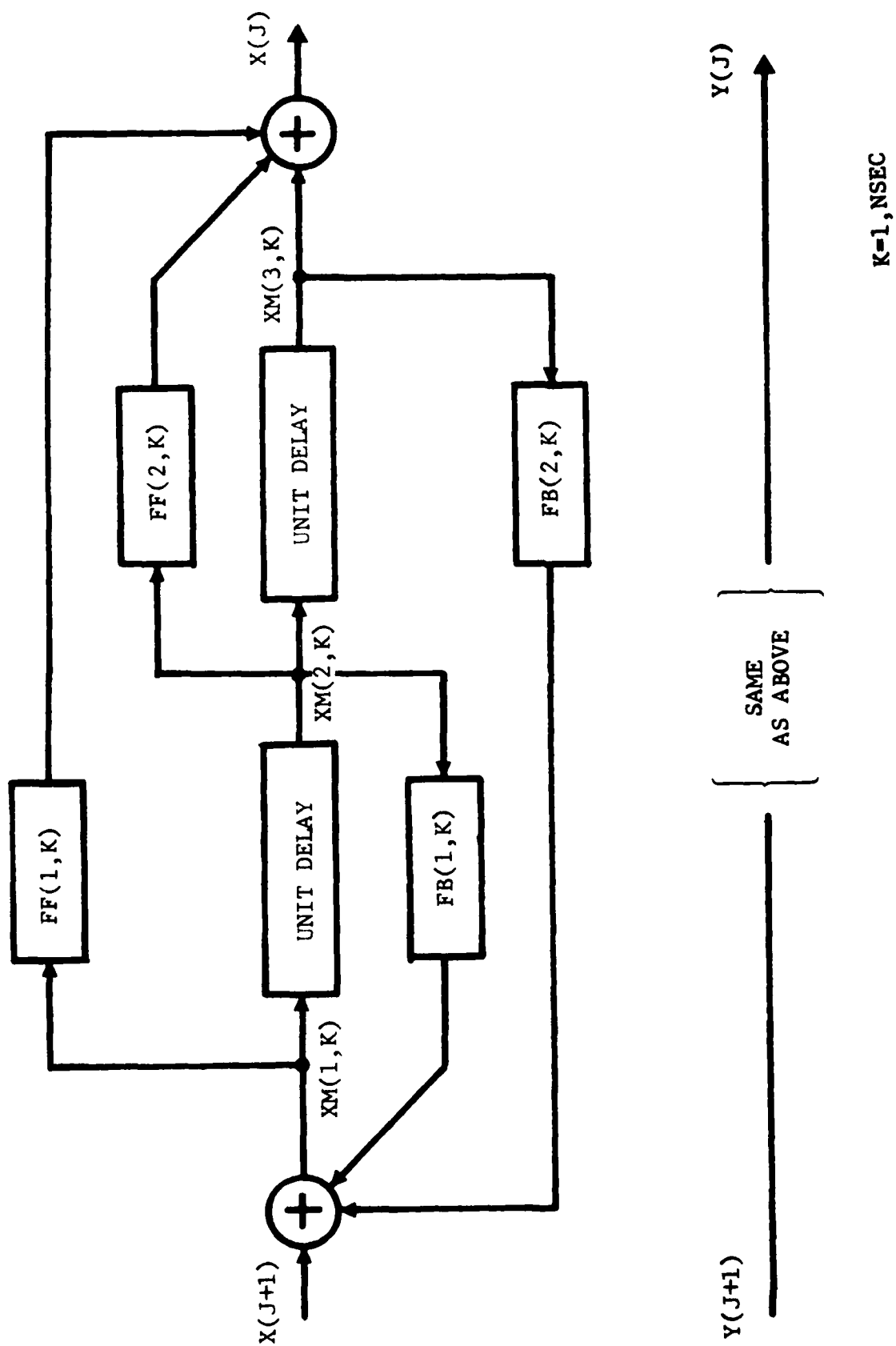


Figure DIGFIL-1 BLOCK DIAGRAM OF DIGFIL/DIGFNC

```

2044      SUBROUTINE DIGFIL(X,Y)
2045      COMMON/BLK1/ FR(200),FR(2,25),FF(2,25)
2046      EQUIVALENCE (FR(199), NP ), (FR( 74), SF )
2047      DATA N193/-3/
2048      DIMENSION X(1),Y(1),XM(3,25),YM(3,25)
2049      DO 100 J=1,25
2700      DO 100 K=1,3
2701      XM(K,J)=0.0
2702      YM(K,J)=0.0
2703      100 CONTINUE
2704      ENTRY DIGFIL(X,Y)
2705      N=ABS(X(N193))
2706      GO TO 150
2707      ENTRY ELMFIL(X,Y)
2708      N=1
2709      150 DO 200 J=1,N
2710      XX=X(J)
2711      YY=Y(J)
2712      DO 300 K=1,NP
2713      XM(1,K)=XM(2,K)*FR(1,K)+XM(3,K)*FR(2,K)+XX
2714      YM(1,K)=YM(2,K)*FR(1,K)+YM(3,K)*FR(2,K)+YY
2715      XX=XM(1,K)*FR(1,K)+XM(2,K)*FR(2,K)+XM(3,K)
2716      YY=YM(1,K)*FR(1,K)+YM(2,K)*FR(2,K)+YM(3,K)
2717      XM(2,K)=XM(1,K)
2718      XM(3,K)=XM(2,K)
2719      YM(3,K)=YM(2,K)
2720      YM(2,K)=YM(1,K)
2721      300 CONTINUE
2722      X(J)=XX*SF
2723      Y(J)=YY*SF
2724      200 CONTINUE
2725      RETURN
2726      END

```

SUBROUTINE DIGFSF

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
DIGFSF	LTI-4	463
ECMFSU	LTI-4	None
ECMFSF	LTI-4	None

2. PURPOSE:

This subroutine is used to simulate a digital filter based on the frequency sampling design concept.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
RADIUS	R	F	Radius of circle in Z-plane which is locus of pole/zero locations
NSAM	R	I	Number of frequency samples used to specify the transfer function
NDFZ	R	I	Number of zeros synthesized
FSAM	R	F	Array containing the filter transfer function specification. The coefficients for the Kth sample are the following: $FSAM(1,K) = \text{real coefficient}$ $FSAM(2,K) = \text{imaginary coefficient}$ The frequency associated with the Kth sample is given by the following expression. $f_K = \frac{f_s}{NDFZ} (K - NDC) \quad K = 1, NSAM$ where: $NDC = IFIX(NSAM/2) + 1$ $f_s = \text{Sampling rate of data to be processed}$

4. CALLING SEQUENCES:

Process a waveform

CALL DIGFSF (X,Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

The storage register arrays (S and PF)
are cleared before execution begins.

Set up for use by ECM module

CALL ECMFSU (X,Y)

Where: X contains the Input Waveform sample - R
 Y contains the Input Waveform sample - I
 X contains the Output Waveform sample - R
 Y contains the Output Waveform sample - I

The storage register arrays (S and PF)
are cleared and coefficients are
initialized.

Process 1 sample (called by ECM module)

CALL ECMFSF (X,Y)

Where: X contains the Input Waveform sample - R
 Y contains the Input Waveform sample - I
 X contains the Output Waveform sample - R
 Y contains the Output Waveform sample - I

The storage register arrays (S and PF)
are not cleared before execution begins.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. This subroutine is structured to simulate a low pass digital filter only.

b. References:

C. M. Rader and B. Gold: "Digital Filter Design Techniques in the Frequency Domain", Proc. IEEE, Vol. 55, pp 149-171, Feb. 1967.

L. R. Rabiner, B. Gold, and C. A. McGonegal: "An Approach to the Approximation Problem for Nonrecursive Digital Filters", IEEE Trans. Audio Electroacoust., Vol. AU-18, pp 83-106, June 1970.

L. R. Rabiner and B. Gold: Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975, pp 105-123.

c. Flow Chart: Page 9-107

d. Cross Reference Table: Page 9-222.

6. THEORY OF OPERATION

The basic concept of this filter design technique is as follows. A comb filter is used to synthesize N zeros around a circle of radius R in the Z -plane. The banks of parallel resonators are placed in cascade with the comb filter. The resonators are chosen such that once the desired passband region their poles exactly cancel the zeros. The weight applied to the resonators are samples of the desired frequency domain transfer function.

The block diagram of the digital filter simulated by the subroutine is shown in Figure DIGFSF-1. Since the phases at resonance of the consecutive resonators differ by π , a phase adjustment is inserted after each resonator. The location of the zeros synthesized by the comb filter are shown in Figure DIGFSF-2. The block diagram of a typical complex resonator is shown in Figure DIGFSF-3.

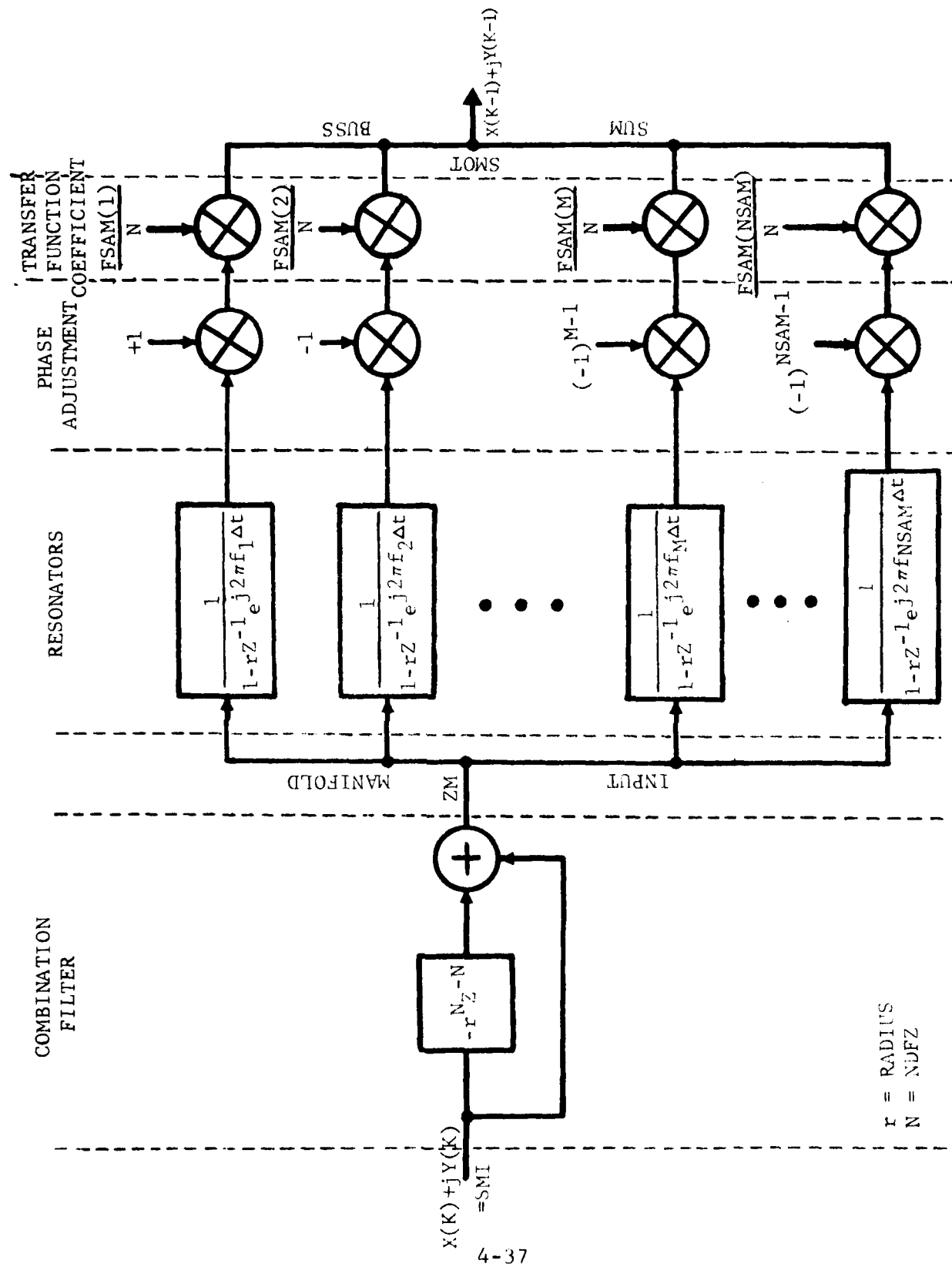


Figure DIGFSF-1 BLOCK DIAGRAM OF DIGFSF

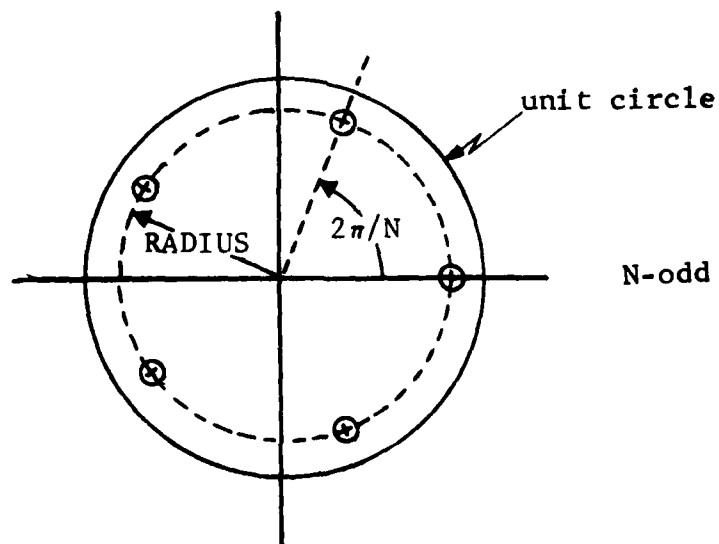
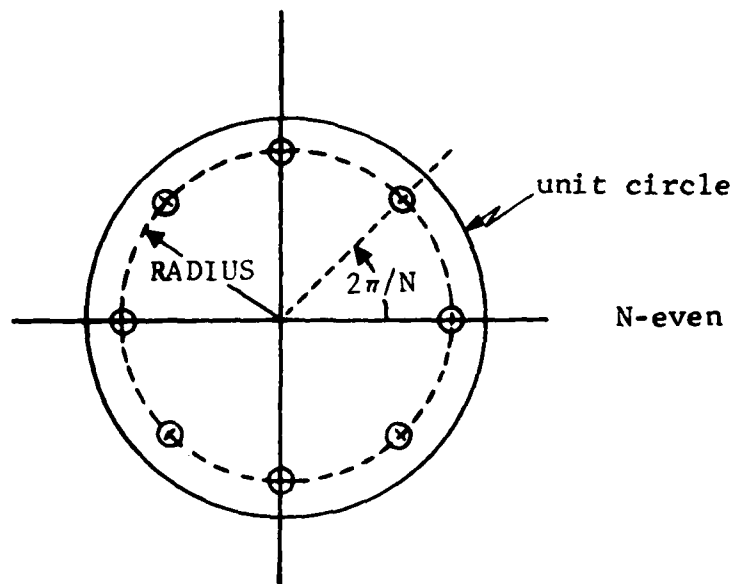
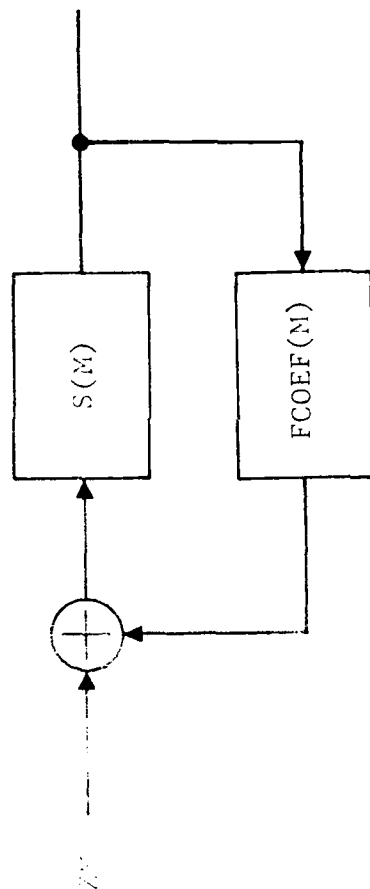


Figure DIGFSF-2 COMBINATION FILTER ZERO LOCATIONS
IN THE Z-PLANE



where: $\text{FCOEF}(N) = e^{j2\pi f_M \Delta t}$

$$f_M = f_S (M\text{-NDC}) / \text{NDFZ}$$

Figure DIGFSF-3 BLOCK DIAGRAM OF A COMPLEX RESONATOR

4-40

[illegible]

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

4-40a

CARD NO

CONTENTS

```

2842      FCOEF(K)=CMPLX(COS(PH),SIN(PH))*RADIUS
2843      FSAM(K)=FSAM(K)*DELSCL
2844      DELSCL=-DELSCL
2845      S(K)=CZERO
2846      100 CONTINUE
2847      NDC=NDFZ+1
2848      DO 105 K=1,NDC
2849      FF(K)=CZERO
2850      105 CONTINUE
2851      DELSCL=RADIUS**NDFZ
2852      IRP=1
2853      ISP=1
2854      ENTRY ECMFDF(X,Y)
2855      DO 210 K=1,NPTS
2856      IRP=IRP+1
2857      SMI=CMPLX(X(K),Y(K))
2858      FF(ISP)=SMI*DELSCL
2859      ZM=SMI-FF(IRP)
2860      ISP=IRP
2861      IF (IRP.EQ.NDC) IRP=0
2862      SMUT=CZERO
2863      DO 300 M=1,NSAM
2864      SMUT=SMUT+S(M)*FSAM(M)
2865      S(M)=S(M)*FCOEF(M)+ZM
2866      300 CONTINUE
2867      X(K)=REAL(SMUT)*XNI
2868      Y(K)=AIMAG(SMUT)*XNI
2869      400 CONTINUE
2870      RETURN

```

SUBROUTINE DIGTFL

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
DIGTFL	NL-2	422,423

2. PURPOSE:

This subroutine is used to simulate a digital transversal filter.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NTAPS	R	I	Number of taps
ITAP	R	I	Array containing the tap specifications. The coefficient for the Kth tap are the following: ITAP(1,K) = numerator of tap gain ITAP(2,K) = denominator of tap gain ITAP(3,K) = delay in sampling increments

4. CALLING SEQUENCES:

CALL DIGTFL(IN, IOUT)

Where: IN contains the Input Waveform

 IOUT contains the Output Waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. This device is typically used as a digital phase decoder.
- b. Flow Chart: Page 9-95
- c. Cross Reference Table: Page 9-220

6. THEORY OF OPERATION

The block diagram of the digital transversal filter simulated by this module is shown in Figure DIGTFL-1. All calculations performed by this module are in integer format.

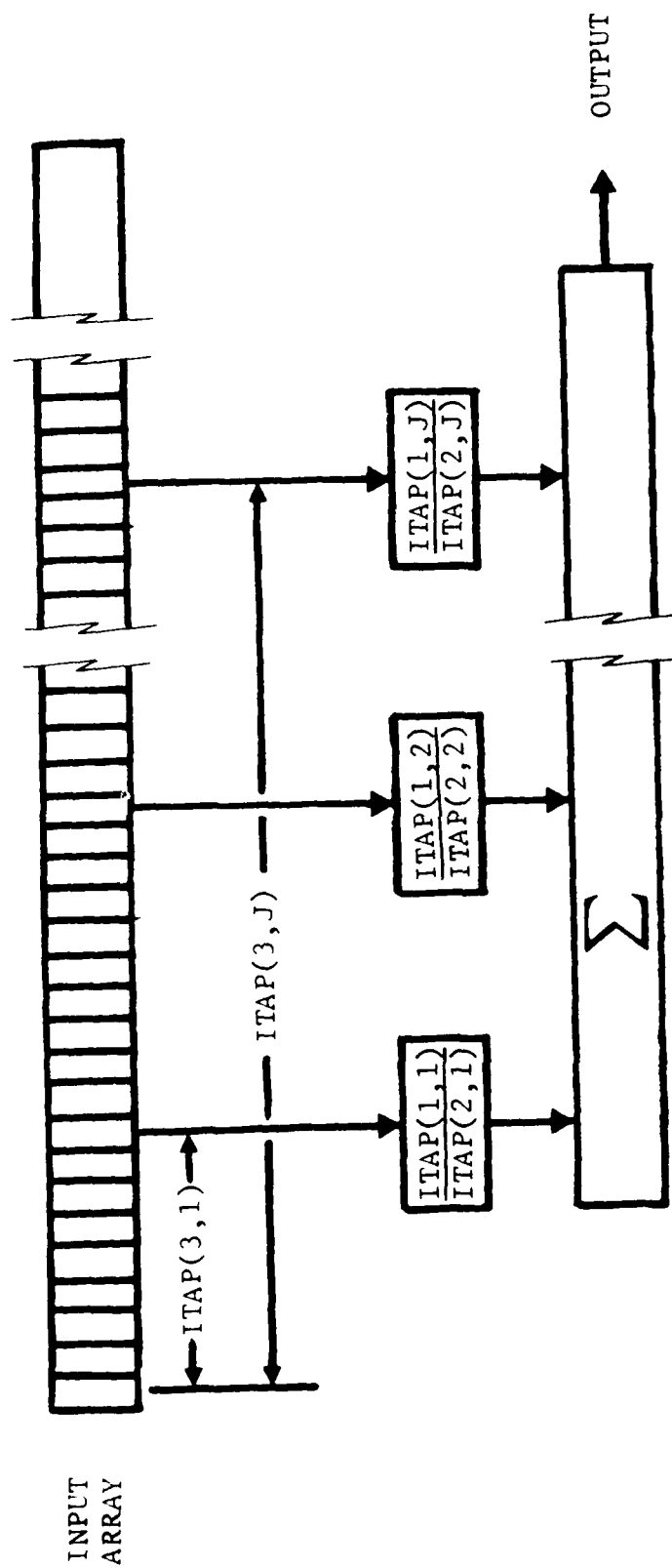


Figure DIGTFL-1 BLOCK DIAGRAM OF DIGTFL

```

2542      SUBROUTINE DIG1FL(IN,ICUT)
2543      COMMON/BLK1/BLK1(200),ITAP(3,100)
2544      DIMENSION IN(1),ICUT(1)
2545      EQUIVALENCE (BLK1(101), NTAPS )
2546      DATA N193,N194,N195,N196/-3,-2,-1,0/
2547      NSTOP=IN(N193)
2548      NSTOP=NSTOP+ITAP(3,NTAPS)
2549      DO 100 J=1,NSTOP
2550      IA=0
2551      DO 200 K=1,NTAPS

```

4-44

00/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NO.

CONTENTS

```

2552      IA=IA+ (ITAP(1,K)* IN(J+ITAP(3,K)) ) / ITAP(2,K)
2553      200 CONTINUE
2554      ICUT(J)=IA
2555      100 CONTINUE
2556      ICUT(N193)=NSTOP
2557      ICUT(N194)=IN(N194)
2558      ICUT(N195)=IN(N195)
2559      RETURN
2560      END

```

SUBROUTINE ECM

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
ECM	SOU-1S	512

2. PURPOSE:

This subroutine simulates a noise modulation jammer.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
TIME	R	F	Elapsed time since beginning of simulation
JRNGØ	0	F	True jammer range corresponding to TIME = 0.0
JRSIM	R	F	Jammer range to be used in computing the starting time of the jammer.
JMAZ	R	F	Jammer azimuth angle
JHGT	0	F	Jammer height above ground
JERP	R	F	Jammer effective radiated power
JFMBW	0	F	Jammer swept bandwidth
JPW	0	F	Jammer on time for pulsed jammers
JFØ	0	F	Jammer center frequency offset from radar center frequency
JVEL	0	F	Jammer radial velocity with respect to the radar site
JPERØD	0	F	Jammer pulse repetition interval for pulsed jammer

<u>Name</u>	<u>O/R</u>	<u>I</u>	<u>Description</u>
N2	R	I	2**N2 = total number of signal storage array elements available
NDFZ	R	I	Number of zeros to be synthesized by the frequency sampling digital filter (DIGFSF).

In addition to the above parameters the characteristics of the jammer spectrum must be specified via the parameters of DIGFSF. These parameters are also members of NLS12.

4. CALLING SEQUENCES:

CALL ECM (X, Y)

Where: X contains the Input Waveform - R
Y contains the Input Waveform - I
X contains the Output Waveform - R
Y contains the Output Waveform - I

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The digital filter routine DIGFSF is used to specify the spectral characteristics of the jammer.
- b. Flow Chart: Page 9-104
- c. Cross Reference Table: Page 9-221

6. THEORY OF OPERATION

This module has four modes of operation:

- (1) Spot noise/Barrage noise

JPW = 0.0

- (2) Swept noise (single sweep)

JFMBW ≠ 0.0

JPW ≠ 0.0

JPEROD = 0.0

(3) Swept noise (repeating sweep)

JFMBW $\neq$ 0.0
JPW $\neq$ 0.0
JPEROD = JPW

(4) Pulsed noise

JPW $\neq$ 0.0
JPEROD > JPW

Dependeing on the range and azimuth positioning of the jammer with respect to the target either a self screening or a standoff jammer can be simulated.

A block diagram of this module is shown in figure ECM-1. The output of the ECM simulation is summed with the data in arrays X and Y.

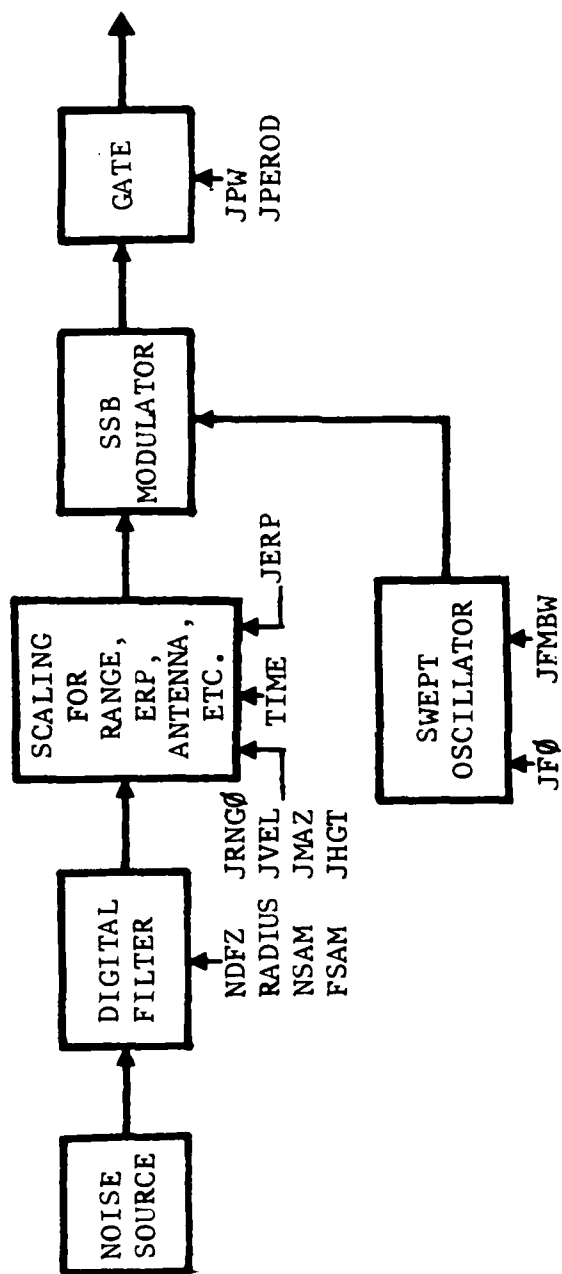


Figure ECM-1 BLOCK DIAGRAM OF ECM

[illegible]

This model is incomplete since the TWT does not produce saturated noise and the output of this model is not properly scaled for range effects. This will be accomplished in the near future.

02/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RADSIM

4-50

CARD NO

CONTENTS

```

2784      200  JBG=IFIX(JST/DELTA)
2785          IF(JBG.LT.1) JBG=1
2786          JST=IFIX(JPW/DELTA)+JBG
2787          IF(JST.GT.NITL) JST=NITL
2788          FSTR1=JPD-JFM6W*0.5
2789          CHIRP2=JFM6W/JPW*0.5
2790          IF(JPERED.EQ.0.0) GOTO 300
2791          NRPT=1
2792          IF(JPERED.GT.JPW) GOTO 296
2793          JCTOFF=1
2794          GOTO 300
2795      290  JCTOFF=IFIX((JPERED-JPW)/DELTA)+1
2796      300  IF(NRPT.GE.NITL) GOTO 310
2797          NRPT=NRPT+1
2798          DO 301 JER,NITL
2799              X(J)=0.0
2800              Y(J)=0.0
2801      301  CONTINUE
2802              X(NRPT)=DELT(NITL)
2803              Y(NRPT)=X(NRPT)
2804      310  T=0.0
2805              DO 300 J=JCT,JST
2806                  AF=FF*DELTA
2807                  CALL LUMDEF(XF,YF)
2808                  PRF=(PRF1+CHIRP2*T)*T
2809                  PRF=AMOD(PRF+1.0)*PI2
2810                  C=CELU(PR)
2811                  S=1.0/PRF
2812                  X(J)=X(J)+C*PRF+S*YF
2813                  Y(J)=Y(J)+C*YF+S*XF
2814                  T=T+DELTA
2815      300  CONTINUE
2816              IF(NITL.EQ.0) RETURN
2817              JCG=JCT+JCTOFF
2818              IF(JCG.GE.NITL) RETURN
2819              JST=JCT+AFIX(JPW/DELTA)
2820              IF(JST.GE.NITL) JST=NITL
2821              GOTO 310
2822      END

```


SUBROUTINE FGENXY

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
FGENXY	SOU-1 or LTI-3	420
FGENMP	SOU-1 or LTI-3	421

2. PURPOSE:

This subroutine is used to simulate the radar waveform generator. In general, the waveform generator subsystem in a radar drives the power amplifier.

3. INPUT PARAMETERS

a. MODE #1; single pulse, internal modulation

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FS	R	F	Simulation sampling rate
FØ	R	F	Center frequency of output waveform
INORM	R	I	Normalization flag
TI	R	F	Time increment between output waveform samples
NPWTX	R	I	Set = 0 for this mode
FMBW	0	F	Linear FM bandwidth
NSUBP	R	I	Set = 0 for this mode
RISTIM	0	F	Rise Time
FALTIM	0	F	Fall Time
TSTART	0	F	Start time of the output waveform
PW	R	F	Pulsewidth

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>

VPEAK	R	F	Peak output voltage

N2	R	I	Simulation parameter used to specify maximum array length

Figure FGENXY-1(a) shows the relationship between some of the input parameters and the output waveform.

b. MODE #2; single pulse, User modulation function

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>

FS	R	F	Simulation sampling rate

F0	0	F	Center frequency of output waveform

INORM	R	I	Normalization flag

TI	R	F	Time increment between output waveform samples

NPWTX	R	I	Number of points in user specified modulation function

WT	R	F	Array containing the user specified modulation function. The specification for the Jth sample of the weighting function is the following:

WT(1,J) = Gain

WT(2,J) = Time

WT(3,J) = Phase angle

NSUBP	R	I	Set = 0 for this mode

FSTART	0	F	Starting frequency at time = TSTART

CHIRP	0	F	Linear FM sweep rate

TSTART	0	F	Start time of the output waveform

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
VPEAK	R	F	Peak output voltage
N2	R	I	Simulation parameter used to specify maximum array length

NOTE: Either FO or FSTART & FMBW must be specified.

Figure FGENXY-1(b) shows the relationship between some of the input parameters and the output waveform.

c. MODE #3; phase coded waveform

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FS	R	F	Simulation sampling rate
FØ	0	F	Center frequency of output waveform
INORM	R	I	Normalization flag
TI	R	F	Time increment between output waveform samples
NPWTX	R	I	Set = 0 for this mode
NSUBP	R	I	Number of subpulses
SPW			Subpulse width
PCODE	R	F	Array containing the phase code
SWTIM	0	F	Switching time between subpulses
RISTIM	0	F	Rise time
FALTIM	0	F	Fall time
FSTART	0	F	Starting frequency at time = TSTART
CHIRP	0	F	Linear FM sweep rate
TSTART	0	F	Start time of the output waveform

<u>Time</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
VPEAK	R	F	Peak output waveform
N2	R	I	Simulation parameter used to specify maximum array length

NOTE: Either FO or FSTART & FMBW must be specified.

Figure FGENXY-1(c) shows the relationship between some of the input parameters and the output waveform.

4. CALLING SEQUENCES:

CALL FGENXY (X,Y)

Where: X contains the Output Waveform - R
Y contains the Output Waveform - I

CALL FGENMP (X,Y)

Where: X contains the Output Waveform - M
Y contains the Output Waveform - P

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The number of waveform samples to be generated must be less than $2*N2$ where $N2 \leq 13$.
- b. The normalization flag, INORM, is used to indicate whether the output is to be interpreted as a matched filter impulse response or as a frequency generator output waveform.

If INORM = 0, 2 or 4 the output represents a frequency generator output.

If INORM = 1 or 3 the output represents the impulse response of an ideal matched filter and therefore TSTART is set equal to zero and no trailing zeros are added to the output. This procedure is required because the normalization procedure used in the Fourier transform routine is different for waveforms and impulse responses.

- c. For Mode #3 the sign of NSUBP indicates whether a polyphase or binary phase code is to be generated.

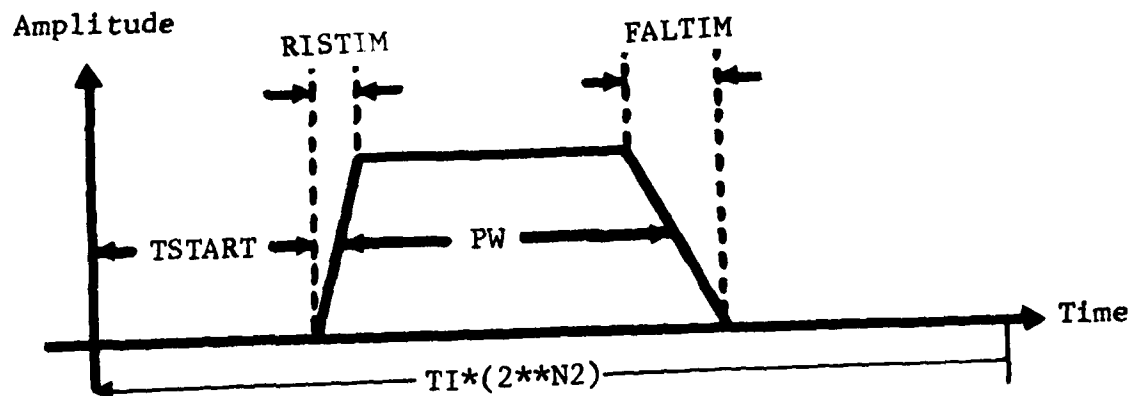


Figure FGENXY-1a OUTPUT WAVEFORM ENVELOPE - MODE 1

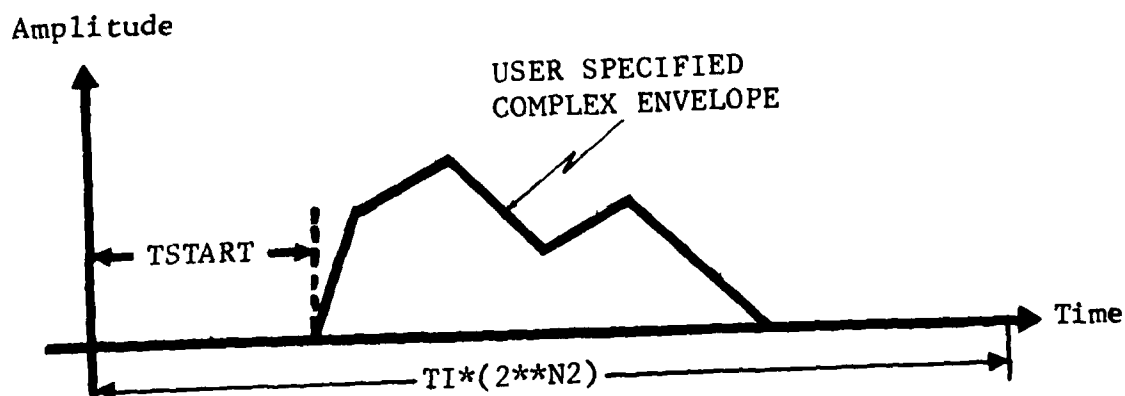


Figure FGENXY-1b OUTPUT WAVEFORM ENVELOPE - MODE 2

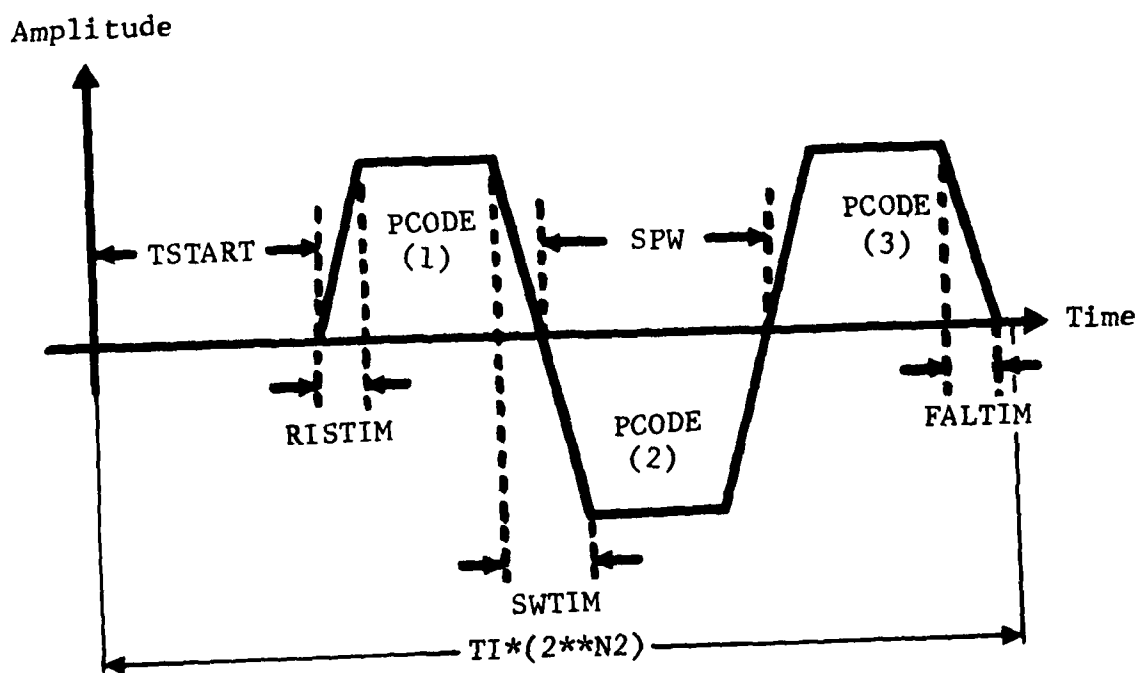


Figure FGENXY-1c OUTPUT WAVEFORM ENVELOPE - MODE 3

If NSUBP is positive the phase is either 0° or 180° with respect to the carrier. If NSUBP is negative the phase between subpulses is determined by linear interpolation.

- d. Operation of this module in Mode #3 causes the phase code array (PCODE) to be copied into the storage array XMITPC for later use by the phase decoder routine, PHDEC. In this way the phase decoder is automatically slaved to the phase code of the waveform generator.
- e. Abort error codes
 - 1, User attempted to specify both a phase code and a complex envelope
 - 2, User attempted to specify a complex envelope with less than 4 points
 - 3, The time span of the waveform to be generated is greater than the simulation time span, i.e. $T_1 \cdot (2 \cdot N^2)$.
- 104, An error occurred during execution of the weighting function routine, WEITCP or WEITMP.
- f. Flow Chart: Page 9-72
- g. Cross Reference Table: Page 9-217.

6. THEORY OF OPERATION

This simulation is used to simulate the master frequency source of a radar system. The basic mechanization equation is given by the following expression:

$$S(t) = a(t) \exp j(2\pi f_0 t + \theta(t))$$

The envelope $a(t)$ and the phase modulation term $\theta(t)$ are determined either directly by the user (MODE #2) or indirectly through parameters (MODE #1 or #2).

This module can also handle a staggered PRF by specifying NPRIS and PRI(J) in the system module (301) namelist. NPRIS is the number of pulse repetitions and must be ≤ 11 . PRI(J) are the intervals in seconds of the pulses. For example, under \$NL301 might be included "... , NPRIS = 4, PRI = 1.0, 0.9, 1.1, 1.0\$."

```

2074      SUBROUTINE FGENXY(X,Y)
2075      COMMON/BLK1/ VAR(200), WT(3,100)
2076      COMMON/PHCLDE/ XMITPC(302)
2077      DIMENSION PCODE(300),X(1),Y(1)
2078      EQUIVALENCE (WT(1,1),PCODE(1))
2079      EQUIVALENCE (VARI 2) ,FS 1 , (VARI 3) , FC 1 ,
2080      *      (VARI 4) , INCRM 1 ,
2081      *      (VARI 12) ,TI 1 , (VARI 37) , NPWT 1 ,
2082      *      (VARI 38) , ORIG 1 , (VARI 42) , CHIEF 1 ,
2083      *      (VARI 43) , FMRW 1 , (VARI 44) , NEWTX 1 ,
2084      *      (VARI 45) , SPW 1 , (VARI 46) , NSUBP 1 ,
2085      *      (VARI 47) , SWTIM 1 , (VARI 48) , RISTIM1 ,
2086      *      (VARI 49) , FALTIM 1 , (VARI 100) , TSTART1 ,
2087      *      (VARI 41) , FSTART1 , (VARI 50) , PW 1 ,

```

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RAUSIM

4-57a

CARD NO

CONTENTS

```

2008      *      (VAR(129) , VPEAK) , (VAR( 1) , N2      )
2009      DATA N143,N144,N145,DTR/-3,-2,-1,1.7453292E-02/
2010      DATA XL1X0/2.7777E-03/,PIZ/6.2E31E53/
2011      ITYPE=1
2012      GOTO 100
2013      ENTRY PGENMP(X,Y)
2014      ITYPE=2
2015      100  IFPM=0
2016      IF(NSUBP.EQ.0) GOTO 110
2017      NSUBP=-NSUBP
2018      IFPM=1
2019      110  IF(LMINP.EQ.0) FSTART=FC
2020      NTL=2*IN2
2021      IF(NSUBP.EQ.0) GOTO 200
2022      IF(NPWTX.NE.0) CALL ABCRT(1)
2023      ITEMP=PCLL(1)
2024      XM1TCL(1)=BCL(NSUBP)
2025      XM1TCL(302)=SPW
2026      DO 300 J=1,NSUBP
2027      XM1TCL(J)=PCLL(J)
2028      300  CONTINUE
2029      GOTO 600
2030      20  CONTINUE
2031      IF(NPWTX.EQ.0) GOTO 300
2032      IF(NPWTX.LT.4) CALL ABORT(2)
2033      SPW=NT(2,NPWTX)
2034      F1TIM=71
2035      FALTIM=11
2036      GOTO 400

```



```

2118      SPW=PW*(RISTIM+ALTIM)*0.5
2119      IF(CHIRP.EQ.0.0) CHIRP=FMBW/PW
2120      FSTART=FL-CHIRP*(PW+RISTIM)*0.5
2121      400  CONTINUE
2122      SUBPH=0.0
2123      NSUBP=1
2124      SWTIM=RISTIM+ALTIM
2125      600  NSTRT=0
2126      IF(INCRM.EQ.1.0.OR.INCRM.EQ.3) GOTO 651
2127      NSTRT=IFIX(FSTART*FS)
2128      FSTART=TI+FLGAT(NSTRT)
2129      NSTRT=NSTRT+1
2130      DO 650 J=1,NSTRT
2131      X(J)=0.0
2132      Y(J)=0.0
2133      650  CONTINUE
2134      601  NS=IFIX(FLGAT(NSUBP)*SPW*FS)+NSTRT
2135      IF(NS.GT.NTIL) CALL ABORT(3)
2136      TIME=0.0
2137      K=1
2138      J=NSTRT
2139      *
2140      ***** LEADING EDGE OF XMITTED WAVEFORM *****
2141      *
2142      IREQ=0
2143      ASLOPE=VPEAK/RISTIM
2144      IF(IPHM.EQ.0) ASLOPE=ASLOPE*CLS(SUBPH*DTK)
2145      PSLOPE=0.0

```

4-58

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RAUSIM

H-58a ****~

CARD NO

CONTENTS

```

2146      TCRIG=0.0
2147      PCRIG=SULPH
2148      ALRIG=0.0
2149      BREAK=K1STIM
2150      700  J=J+1
2151          TIME=TIME+TI
2152      800  IF (TIME.GE.BREAK) GOTO 900
2153          DTIME=TIME-TCRIG
2154          PH=(PSTART+CHIRP*0.5*TIME) * TIME
2155          IF (IPHM.EQ.1) PH=PH+(PCRIG+PSLOPE*DTIME)*XCIFC
2156          AMPL=ALRIG + ASLOPE*DTIME
2157      850  IF (ITY.EE.1) GOTO 870
2158          X(J)=AMPL
2159          Y(J)=PH*300.0
2160          GOTO 700
2161      870  PHASE=(PH-FLCAT(IFIX(PH)))*PI/2
2162          X(J)=AMPL*COS(PHASE)
2163          Y(J)=AMPL*SIN(PHASE)
2164          GOTO 700
2165      900  IF EG=IFEG+1
2166          TCRIG=BREAK
2167          IF (IFEG.EE.1) GOTO 910
2168          IF (IFEG.EE.4) GOTO 1000
2169          IF (IFEG.EE.3) GOTO 920
2170          IFEG=IFEG-4
2171      *
2172      ***** TOP OF SUBPULSE *****
2173      *
2174      910  BREAK=BREAK+SPW-SWTIM

```

```

2175      FLLCFE=0.0
2176      FLEIG=SUBPH
2177      ASLOPE=0.0
2178      ALKIC=VPEAK
2179      IF (IPHM.EQ.0) ALFIC=VPEAK*CCS(SUBPH*DTK)
2180      GOTO 100
2181      *
2182      ***** SWITCH REGION *****
2183      *
2184      900   K=K+1
2185      IF (K.GT.NSURF) GOTO 950
2186      EFLAK=EFLAK+SWTIM
2187      SUBPH=CODE(K)
2188      FSLCFE=(SOUTH - POFIG)/SWTIM
2189      IF (IPHM.EQ.0) ASLOPE=(VPEAK*CCS(SUBPH*DTK)-AUFIG)/SWTIM
2190      GOTO 100
2191      *
2192      ***** TRAILING EDGE OF XMITTED WAVEFORM *****
2193      *
2194      950   OFEAK=EFLAK*FALTIM
2195      AULCFE=-AUFIG/FALTIM
2196      IF EG=3
2197      GOTO 100
2198      1000  IF (INGRM.EQ.1.OR.INGRM.EQ.3) GOTO 1501
2199      DO 1500 K=J,NITL
2200      X(K)=0.0
2201      Y(K)=0.0
2202      1500  CONTINUE
2203      GOTO 1502

```

02/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

4-59*

CARD NO.

CONTENTS

```

2204      1501  NTL=J-1
2205      1502  A(N195)=S00L(NTL)
2206              A(N194)=0.0
2207              X(N195)=11
2208              Y(N195)=X(N195)
2209              Y(N194)=X(N194)
2210              Y(N195)=X(N195)
2211              IF (NWTX,0.0) RETURN
2212              CFAC=12.681
2213              NWT=HWTX
2214              IF (TYPE,N1,1) GOTO 1599
2215              GO 2000 J=1,NWTX
2216              XM=WT(1,J)
2217              WT(1,J)=XM*CU2(WT(3,J)*CTR)
2218              WT(2,J)=AM*CU2(WT(3,J)*CTR)
2219      2000  CONTINUE
2220              CALL WRTTFC(X,Y,11.00)
2221              RETURN
2222      1599  CALL WRTTFC(X,Y,12.00)
2223              RETURN
2224      1600  CALL XCRB(104)
2225              RETURN
2226      END

```

SUBROUTINE FILT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
FILT	LTI-1	407,408

2. PURPOSE:

This subroutine simulates a continuous filter which is defined by an S-domain polynomial transfer function.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NZ	R	I	Number of zeros in the transfer function.
NP	R	I	Number of poles in the transfer function.
SF	R	F	Scale factor.
FZERO	O	F	Array containing the zero specifications. The s-plane location for the Kth zero is given by the following coefficients: FZERO(1,K) = real component FZERO(2,K) = imaginary component
FPOLE	O	F	Array containing the pole specifications. The s-plane location for the Kth pole is given by the following coefficients: FPOLE(1,K) = real component FPOLE(2,K) = imaginary component

4. CALLING SEQUENCES:

CALL FILT(X,Y)

Where:

X contains the Input Waveform - R

Y contains the Input Waveform - I

X contains the Output Waveform - R

Y contains the Output Waveform - I

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. A purely imaginary pole may not be specified within the frequency extent of the input waveform. This would cause either zero or an extremely small number to occur in the denominator of the transfer function.

b. Flow Chart: Page 9-55

c. Cross Reference Table: Page 9-215

6. THEORY OF OPERATION

The S-domain transfer function for a general filter can be represented by the ratio of two polynomials as shown in the following expression

$$H(S) = \frac{(S-Z_1)(S-Z_2)(S-Z_3) \dots (S-Z_{NZ})}{(S-P_1)(S-P_2)(S-P_3) \dots (S-P_{NP})} \cdot SF$$

In general, $H(S)$ is a complex function of the frequency variable s . The filter output signal $S_o(f)$ can be determined using the following expression

$$S_o(f) = H(f) S_i(f)$$

where $S_i(f)$ is the input signal representation in the frequency domain. The discrete representation of these equations used in the module are the following:

$$H(J) = SF \cdot \frac{\prod_{K=1}^{K=NZ} \{j \text{FREQ}(J) - [FZERO(1,K) + j FZERO(2,K)]\}}{\prod_{K=1}^{K=NP} \{j \text{FREQ}(J) - [FP0LE(1,K) + j FP0LE(2,K)]\}}$$

$$X(J) + j Y(J) = H(J) * [X(J) + j Y(J)]$$

where: FREQ(J) is the frequency associated with the Jth sample

1720 SUBROUTINE FILT(X, Y)
1721 COMMON/FLK1/ BK1(200),FZERO(2,50),FFOLE(2,50)
1728 EQUIVALENCE (NZ ,BK1(72)) , (NP ,BK1(73)) , (SF ,BK1(74))
1729 DIMENSION X(1), Y(1)
1730 DATA N193,N194,N195,N196/-3,-2,-1,0/
1731 FREQ=X(N194)
1732 N = IBCCL(X(N193))
1733 DOLF = X(N195)
1734 IF(SF .EQ. 0.) SF = 1.0
1735 DO 100 J=1,N
1736 IF(NZ.EC.0) GO TO 150
1737 DO 200 K=1,NZ
1738 WD = FREQ - FZERL(2,K)
1739 A = X(J)

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RAUSIM

4-63a

CARD NO.

CONTENTS

```
1740      X(J) = -A * FZERU(1,K) - Y(J) * WU
1741      Y(J) =  A * WU - Y(J) * FZERU(1,K)
1742      200 CONTINUE
1743      150 IF(NP,EL,0) GO TO 300
1744      DO 250 K=1,NP
1745      WU = FREQ * FPOLE(2,K)
1746      A = X(J)
1747      AA = FPOLE(1,K) * FPOLE(1,K) + WU * WU
1748      X(J) = (-A * FPOLE(1,K) + Y(J) * WU) / AA
1749      Y(J) = (-A * WU - Y(J) * FPOLE(1,K)) / AA
1750      250 CONTINUE
1751      300 X(J) = X(J) * SF
1752      Y(J) = Y(J) * SF
1753      FREQ = FREQ * DELF
1754      100 CONTINUE
1755      RETURN
1756      END
```

SUBROUTINE HET

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
HET	LTV-4	-

2. PURPOSE:

This subroutine simulates a single sideband modulator which is used to heterodyne waveforms.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FSHIFT	R	F	Frequency shift to be applied to the input waveform.

4. CALLING SEQUENCE:

CALL HET(X,Y)

where:

- X contains the Input Waveform - R
- Y contains the Input Waveform - I
- X contains the Output Waveform - R
- Y contains the Output Waveform - I

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA:

- a. Flow Chart: Page 9-103
- b. Cross Reference Table: Page 9-221

6. THEORY OF OPERATION:

The basic mechanization equation for this module is the following:

$$s'(t) = s(t) \times e^{j \text{FSHIFT} \cdot t}$$

or

$$X'(t) + jY'(t) = [X(t) + jY(t)] \cdot [\cos(\text{FSHIFT} \cdot t) + j \sin(\text{FSHIFT} \cdot t)]$$

The block diagram of a SSB modulator is shown in Figure HET-1.

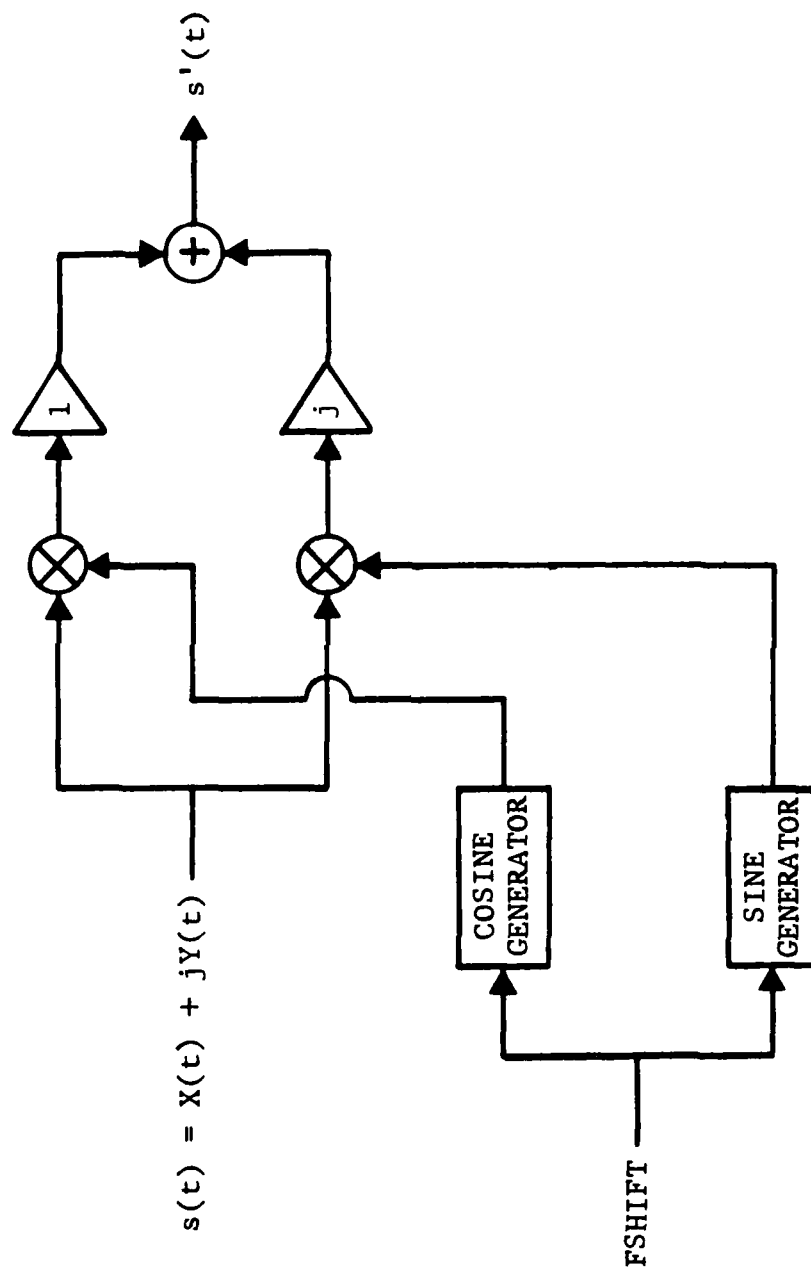


Figure HET-1 BLOCK DIAGRAM OF SSB MODULATOR

4-646

```

2707 SUBROUTINE RUT(X,Y)
2708 DIMENSION X(1),Y(1)
2709 COMMON ZPERIZ VAR(500)
2710 EQUIVALENCE (VAR(15), FSHIFT)
2711 DATA N193,N194,N195,N196 /-3,-2,-1, 0/,PI/70.,.05,0.5/
2712 NFIS=CODE(X(N193))
2713 OR=X(N194)
2714 ULL=X(N195)
2715 DOUTGG=1,NFIS
2716 THETA=OR*FSHIFT
2717 THETA= (THETA- FLOOR(FIX(THETA)))*PI
2718 C= COS(THETA)
2719 S= SIN(THETA)
2720 TMP=X(J)
2721 X(J)= X(J)*C - Y(J)*S
2722 Y(J)= TMP*S + Y(J)*C
2723 OR= OR*ULL
2724 100 CONTINUE
2725 RETURN
2726 END

```

SUBROUTINE HWDET

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
HWDET	NL-1	414,415
FWDET	NL-1	416,417
SQDET	NL-1	418,419
HLIM	NL-1	438,439
IHWDET	NL-1	445,446
IFWDET	NL-1	447,448
ISQDET	NL-1	449,450
IHLIM	NL-1	442,443

2. PURPOSE:

This subroutine simulates a signal detector. Depending on the entry point, a half-wave detector, a full-wave detector, a square-law detector, or a hard limiter is simulated.

3. INPUT PARAMETERS:

NONE

4. CALLING SEQUENCES AND THEORY OF OPERATION

a. Half-Wave Detector

CALL HWDET(XIN,XOUT)

Where: XIN contains the Input Waveform

XOUT contains the Output Waveform

$XOUT(J) = XIN(J) ; \text{ if } XIN(J) > 0.0$
 $= 0.0 ; \text{ otherwise}$

b. Full-Wave Detector

CALL FWDET(XIN,XOUT)

Where: XIN contains the Input Waveform
XOUT contains the Output Waveform
$$XOUT(J) = |XIN(J)|$$

c. Square-Law Detector

CALL SQDET(XIN,XOUT)

Where: XIN contains the Input Waveform
XOUT contains the Output Waveform
$$XOUT(J) = XIN(J)*XIN(J)$$

d. Hard Limiter

CALL HLIM(XIN,XOUT)

Where: XIN contains the Input Waveform
XOUT contains the Output Waveform
$$XOUT(J) = 1.0 \text{ ; if } XIN(J) \geq 0.0$$
$$= -1.0; \text{ if } XIN(J) < 0.0$$

e. Digital Half-Wave Detector

CALL IHWDET(IN,IOUT)

Where: IN contains the Input Waveform
IOUT contains the Output Waveform
$$IOUT(J) = IOUT(J) \text{ ; if } IOUT(J) > 0$$
$$= 0 \text{ ; otherwise}$$

f. Digital Full-Wave Detector

CALL IFWDET(IN, IOUT)

Where: IN contains the Input Waveform

IOUT contains the Output Waveform

IOUT(J) = LABS(IN(J))

g. Digital Square-Law Detector

CALL ISQDET(IN, IOUT)

Where: IN contains the Input Waveform

IOUT contains the Output Waveform

```

IOUT(J) = IN(J)*IN(J)

```

h. Digital Hard Limiter

CALL IHLIM(IN, IOUT)

Where: IN contains the Input Waveform

IOUT contains the Output Waveform

```
IOUT(J) = 1 ; IN(J) > 0
```

$$= 0 ; \text{IN}(J) = 0$$

$$= 1 ; \text{IN}(J) < 0$$

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. Flow Chart: Page 9-137

b. Cross Reference Table: Page 9-225

```

3560      SUBROUTINE FWDDET(XIN, XOUT)
3561      DIMENSION XIN(1),XOUT(1),IN(1),IOUT(1)
3562      DATA N193,N194,N195,N196/-3,-2,-1,0/
3563      NS = IBOUND(XIN(N193))
3564      DO 20 I=1,NS
3565          XOUT(I) = XIN(I)
3566      GO IF( XOUT(1) .LT. 0.0 ) XOUT(1) = 0.0
3567      GO TO 100
3568      C
3569      ENTRY FWDET(XIN, XOUT)
3570      NS = IBOUND(XIN(N193))
3571      DO 40 I=1,NS
3572          XOUT(I) = ABS(XIN(I))
3573      GO TO 100
3574      C
3575      ENTRY SQDET(XIN, XOUT)
3576      NS = IBOUND(XIN(N193))
3577      DO 60 I=1,NS
3578          XOUT(I) = ATN(1)*XIN(I)
3579      GO TO 100
3580      C
3581      ENTRY BLIM(XIN,XOUT)
3582      NS=IBOUND(XIN(N193))
3583      DO 80 I=1,NS
3584          IF(XIN(I)) 01,02,02
3585          XOUT(I)=-1.0
3586      GO TO 100
3587      01 XOUT(I)=1.0
3588      GO CONTINUE
3589      C
3590      XOUT(N193) = XIN(N193)
3591      XOUT(N194) = XIN(N194)
3592      XOUT(N195) = XIN(N195)
3593      RETURN
3594      C
3595      ENTRY IDWDET(IN,IOUT)

```


06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FMC/SCC - FADSIM

4-69

CARD NO

CONTENTS

```

3596      NS= IN(IN193)
3597      DO 1=1,NS
3598      IF(IN(1)) 31,31,32
3599      31 ICUT(1)=0
3600      GO TO 30
3601      32 ICUT(1)=IN(1)
3602      30 CONTINUE
3603      GO TO 200
3604      C
3605      ENTRY IFWGET(IN,ICUT)
3606      NS= IN(IN194)
3607      DO 1=1,NS
3608      34 ICUT(1)=1465(IN(1))
3609      GO TO 200
3610      C
3611      ENTRY ICUT(1)=IN(1)
3612      NS= IN(IN195)
3613      DO 1=1,NS
3614      36 ICUT(1)=IN(1)*IN(1)
3615      GO TO 200
3616      C
3617      ENTRY IN(1)=IN(1)*ICUT(1)
3618      NS=IN(IN196)
3619      DO 1=1,NS
3620      IF(IN(1)) 34,34,35
3621      34 ICUT(1)=1
3622      GO TO 30
3623      35 ICUT(1)=1
3624      GO TO 30
3625      36 ICUT(1)=1
3626      GO TO 30
3627      37 ICUT(1)=1
3628      GO TO 30
3629      38 ICUT(1)=1
3630      GO TO 30
3631      39 ICUT(1)=1
3632      GO TO 30
3633      40 ICUT(1)=1
3634      GO TO 30
3635      41 ICUT(1)=1
3636      GO TO 30
3637      42 ICUT(1)=1
3638      GO TO 30
3639      43 ICUT(1)=1
3640      GO TO 30
3641      44 ICUT(1)=1
3642      GO TO 30
3643      45 ICUT(1)=1
3644      GO TO 30
3645      46 ICUT(1)=1
3646      GO TO 30
3647      47 ICUT(1)=1
3648      GO TO 30
3649      48 ICUT(1)=1
3650      GO TO 30
3651      49 ICUT(1)=1
3652      GO TO 30
3653      50 ICUT(1)=1
3654      GO TO 30
3655      51 ICUT(1)=1
3656      GO TO 30
3657      52 ICUT(1)=1
3658      GO TO 30
3659      53 ICUT(1)=1
3660      GO TO 30
3661      54 ICUT(1)=1
3662      GO TO 30
3663      55 ICUT(1)=1
3664      GO TO 30
3665      56 ICUT(1)=1
3666      GO TO 30
3667      57 ICUT(1)=1
3668      GO TO 30
3669      58 ICUT(1)=1
3670      GO TO 30
3671      59 ICUT(1)=1
3672      GO TO 30
3673      60 ICUT(1)=1
3674      GO TO 30
3675      61 ICUT(1)=1
3676      GO TO 30
3677      62 ICUT(1)=1
3678      GO TO 30
3679      63 ICUT(1)=1
3680      GO TO 30
3681      64 ICUT(1)=1
3682      GO TO 30
3683      65 ICUT(1)=1
3684      GO TO 30
3685      66 ICUT(1)=1
3686      GO TO 30
3687      67 ICUT(1)=1
3688      GO TO 30
3689      68 ICUT(1)=1
3690      GO TO 30
3691      69 ICUT(1)=1
3692      GO TO 30
3693      70 ICUT(1)=1
3694      GO TO 30
3695      71 ICUT(1)=1
3696      GO TO 30
3697      72 ICUT(1)=1
3698      GO TO 30
3699      73 ICUT(1)=1
3700      GO TO 30
3701      74 ICUT(1)=1
3702      GO TO 30
3703      75 ICUT(1)=1
3704      GO TO 30
3705      76 ICUT(1)=1
3706      GO TO 30
3707      77 ICUT(1)=1
3708      GO TO 30
3709      78 ICUT(1)=1
3710      GO TO 30
3711      79 ICUT(1)=1
3712      GO TO 30
3713      80 ICUT(1)=1
3714      GO TO 30
3715      81 ICUT(1)=1
3716      GO TO 30
3717      82 ICUT(1)=1
3718      GO TO 30
3719      83 ICUT(1)=1
3720      GO TO 30
3721      84 ICUT(1)=1
3722      GO TO 30
3723      85 ICUT(1)=1
3724      GO TO 30
3725      86 ICUT(1)=1
3726      GO TO 30
3727      87 ICUT(1)=1
3728      GO TO 30
3729      88 ICUT(1)=1
3730      GO TO 30
3731      89 ICUT(1)=1
3732      GO TO 30
3733      90 ICUT(1)=1
3734      GO TO 30
3735      91 ICUT(1)=1
3736      GO TO 30
3737      92 ICUT(1)=1
3738      GO TO 30
3739      93 ICUT(1)=1
3740      GO TO 30
3741      94 ICUT(1)=1
3742      GO TO 30
3743      95 ICUT(1)=1
3744      GO TO 30
3745      96 ICUT(1)=1
3746      GO TO 30
3747      97 ICUT(1)=1
3748      GO TO 30
3749      98 ICUT(1)=1
3750      GO TO 30
3751      99 ICUT(1)=1
3752      GO TO 30
3753      100 ICUT(1)=1
3754      GO TO 30
3755      101 ICUT(1)=1
3756      GO TO 30
3757      102 ICUT(1)=1
3758      GO TO 30
3759      103 ICUT(1)=1
3760      GO TO 30
3761      104 ICUT(1)=1
3762      GO TO 30
3763      105 ICUT(1)=1
3764      GO TO 30
3765      106 ICUT(1)=1
3766      GO TO 30
3767      107 ICUT(1)=1
3768      GO TO 30
3769      108 ICUT(1)=1
3770      GO TO 30
3771      109 ICUT(1)=1
3772      GO TO 30
3773      110 ICUT(1)=1
3774      GO TO 30
3775      111 ICUT(1)=1
3776      GO TO 30
3777      112 ICUT(1)=1
3778      GO TO 30
3779      113 ICUT(1)=1
3780      GO TO 30
3781      114 ICUT(1)=1
3782      GO TO 30
3783      115 ICUT(1)=1
3784      GO TO 30
3785      116 ICUT(1)=1
3786      GO TO 30
3787      117 ICUT(1)=1
3788      GO TO 30
3789      118 ICUT(1)=1
3790      GO TO 30
3791      119 ICUT(1)=1
3792      GO TO 30
3793      120 ICUT(1)=1
3794      GO TO 30
3795      121 ICUT(1)=1
3796      GO TO 30
3797      122 ICUT(1)=1
3798      GO TO 30
3799      123 ICUT(1)=1
3800      GO TO 30
3801      124 ICUT(1)=1
3802      GO TO 30
3803      125 ICUT(1)=1
3804      GO TO 30
3805      126 ICUT(1)=1
3806      GO TO 30
3807      127 ICUT(1)=1
3808      GO TO 30
3809      128 ICUT(1)=1
3810      GO TO 30
3811      129 ICUT(1)=1
3812      GO TO 30
3813      130 ICUT(1)=1
3814      GO TO 30
3815      131 ICUT(1)=1
3816      GO TO 30
3817      132 ICUT(1)=1
3818      GO TO 30
3819      133 ICUT(1)=1
3820      GO TO 30
3821      134 ICUT(1)=1
3822      GO TO 30
3823      135 ICUT(1)=1
3824      GO TO 30
3825      136 ICUT(1)=1
3826      GO TO 30
3827      137 ICUT(1)=1
3828      GO TO 30
3829      138 ICUT(1)=1
3830      GO TO 30
3831      139 ICUT(1)=1
3832      GO TO 30
3833      140 ICUT(1)=1
3834      GO TO 30
3835      141 ICUT(1)=1
3836      GO TO 30
3837      142 ICUT(1)=1
3838      GO TO 30
3839      143 ICUT(1)=1
3840      GO TO 30
3841      144 ICUT(1)=1
3842      GO TO 30
3843      145 ICUT(1)=1
3844      GO TO 30
3845      146 ICUT(1)=1
3846      GO TO 30
3847      147 ICUT(1)=1
3848      GO TO 30
3849      148 ICUT(1)=1
3850      GO TO 30
3851      149 ICUT(1)=1
3852      GO TO 30
3853      150 ICUT(1)=1
3854      GO TO 30
3855      151 ICUT(1)=1
3856      GO TO 30
3857      152 ICUT(1)=1
3858      GO TO 30
3859      153 ICUT(1)=1
3860      GO TO 30
3861      154 ICUT(1)=1
3862      GO TO 30
3863      155 ICUT(1)=1
3864      GO TO 30
3865      156 ICUT(1)=1
3866      GO TO 30
3867      157 ICUT(1)=1
3868      GO TO 30
3869      158 ICUT(1)=1
3870      GO TO 30
3871      159 ICUT(1)=1
3872      GO TO 30
3873      160 ICUT(1)=1
3874      GO TO 30
3875      161 ICUT(1)=1
3876      GO TO 30
3877      162 ICUT(1)=1
3878      GO TO 30
3879      163 ICUT(1)=1
3880      GO TO 30
3881      164 ICUT(1)=1
3882      GO TO 30
3883      165 ICUT(1)=1
3884      GO TO 30
3885      166 ICUT(1)=1
3886      GO TO 30
3887      167 ICUT(1)=1
3888      GO TO 30
3889      168 ICUT(1)=1
3890      GO TO 30
3891      169 ICUT(1)=1
3892      GO TO 30
3893      170 ICUT(1)=1
3894      GO TO 30
3895      171 ICUT(1)=1
3896      GO TO 30
3897      172 ICUT(1)=1
3898      GO TO 30
3899      173 ICUT(1)=1
3900      GO TO 30
3901      174 ICUT(1)=1
3902      GO TO 30
3903      175 ICUT(1)=1
3904      GO TO 30
3905      176 ICUT(1)=1
3906      GO TO 30
3907      177 ICUT(1)=1
3908      GO TO 30
3909      178 ICUT(1)=1
3910      GO TO 30
3911      179 ICUT(1)=1
3912      GO TO 30
3913      180 ICUT(1)=1
3914      GO TO 30
3915      181 ICUT(1)=1
3916      GO TO 30
3917      182 ICUT(1)=1
3918      GO TO 30
3919      183 ICUT(1)=1
3920      GO TO 30
3921      184 ICUT(1)=1
3922      GO TO 30
3923      185 ICUT(1)=1
3924      GO TO 30
3925      186 ICUT(1)=1
3926      GO TO 30
3927      187 ICUT(1)=1
3928      GO TO 30
3929      188 ICUT(1)=1
3930      GO TO 30
3931      189 ICUT(1)=1
3932      GO TO 30
3933      190 ICUT(1)=1
3934      GO TO 30
3935      191 ICUT(1)=1
3936      GO TO 30
3937      192 ICUT(1)=1
3938      GO TO 30
3939      193 ICUT(1)=1
3940      GO TO 30
3941      194 ICUT(1)=1
3942      GO TO 30
3943      195 ICUT(1)=1
3944      GO TO 30
3945      196 ICUT(1)=1
3946      GO TO 30
3947      197 ICUT(1)=1
3948      GO TO 30
3949      198 ICUT(1)=1
3950      GO TO 30
3951      199 ICUT(1)=1
3952      GO TO 30
3953      200 ICUT(1)=1
3954      GO TO 30
3955      201 ICUT(1)=1
3956      GO TO 30
3957      202 ICUT(1)=1
3958      GO TO 30
3959      203 ICUT(1)=1
3960      GO TO 30
3961      204 ICUT(1)=1
3962      GO TO 30
3963      205 ICUT(1)=1
3964      GO TO 30
3965      206 ICUT(1)=1
3966      GO TO 30
3967      207 ICUT(1)=1
3968      GO TO 30
3969      208 ICUT(1)=1
3970      GO TO 30
3971      209 ICUT(1)=1
3972      GO TO 30
3973      210 ICUT(1)=1
3974      GO TO 30
3975      211 ICUT(1)=1
3976      GO TO 30
3977      212 ICUT(1)=1
3978      GO TO 30
3979      213 ICUT(1)=1
3980      GO TO 30
3981      214 ICUT(1)=1
3982      GO TO 30
3983      215 ICUT(1)=1
3984      GO TO 30
3985      216 ICUT(1)=1
3986      GO TO 30
3987      217 ICUT(1)=1
3988      GO TO 30
3989      218 ICUT(1)=1
3990      GO TO 30
3991      219 ICUT(1)=1
3992      GO TO 30
3993      220 ICUT(1)=1
3994      GO TO 30
3995      221 ICUT(1)=1
3996      GO TO 30
3997      222 ICUT(1)=1
3998      GO TO 30
3999      223 ICUT(1)=1
3999      GO TO 30

```

SUBROUTINE INGTOR

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
INGTOR	LTI-4	409, 410
INGNCL	LTI-4	None

2. PURPOSE:

This subroutine is used to simulate a digital integrator.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FBCK	R	F	Feedback coefficient

4. CALLING SEQUENCES:

CALL INGTOR (DIN, DOUT)

Where: DIN contains the Input Waveform
DOUT contains the Output Waveform

The storage register (C1) is cleared before execution begins.

CALL INGNCL (DIN, DOUT)

Where: DIN contains the Input Waveform
DOUT contains the Output Waveform

The storage register (C1) is not cleared before execution begins.

5. RESTRICTIONS, REQUIREMENT, MISCELLANEOUS DATA

a. Flow Chart: Page 9-97

b. Cross Reference Table: Page 9-220

6. THEORY OF OPERATION

The block diagram of the digital integrator simulated by this module is shown in Figure INGTOR-1. The Z-plane transfer function is given by the following expression:

$$T(Z) = \frac{Z}{Z-FBCK}$$

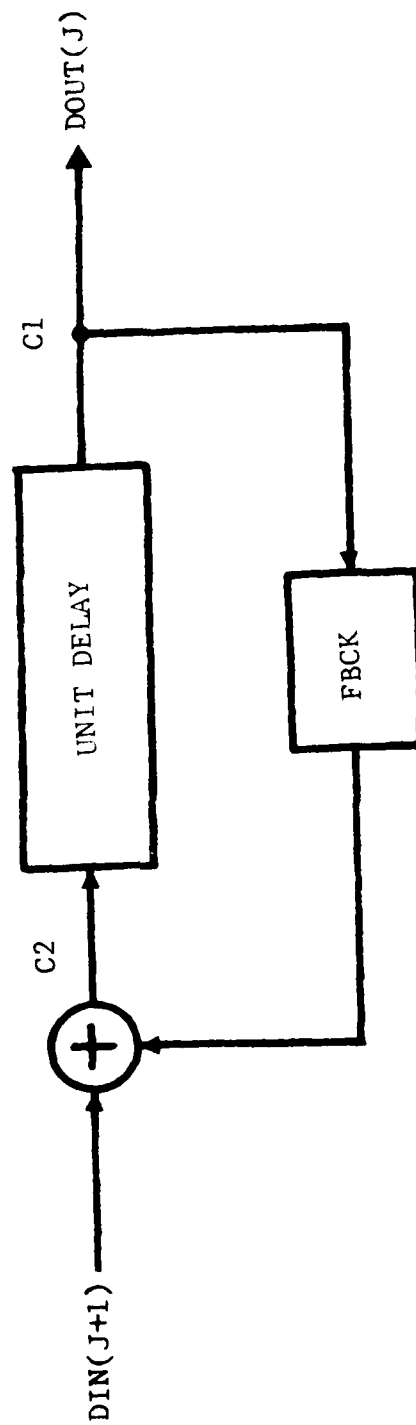


Figure INGTOR-1 BLOCK DIAGRAM OF INGTOR

```

2581      SUBROUTINE INCTER(UIN,UOUT)
2582      COMMON/IK1/IK1(500)
2583      DIMENSION UIN(1),UOUT(1)
2584      EQUIVALENCE (IK1( 21), IDMY      ), (IK1( 75), FECK      )
2585      DATA N193,N194,N195,N196/-3,-2,-1,0/
2586      U1 = 0.0
2587      UIN = INGRU(UIN,UOUT)
2588      N = 1+UIN(1)/(N193)
2589      UOUT(N193) = UIN(N193)
2590      UOUT(N194) = UIN(N194)
2591      UOUT(N195) = UIN(N195)
2592      DO 10 J=1,6
2593      U2 = UIN(J)+U1*FECK
2594      UOUT(J) = U1
2595      U1 = U2
2596      10 CONTINUE
2597      RETURN
2598      END

```

070
4-73

SUBROUTINE IONOS

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
IONOS	LTI-1	511

2. PURPOSE:

This subroutine is used to simulate the effect of the ionosphere on signals propagating through it.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
RFFØ	R	F	Center frequency of the electromagnetic wave traversing the ionosphere
SEDENS	R	F	The integrated electron density along the propagation path

4. CALLING SEQUENCES:

CALL IONOS (X,Y)

Where:

X contains the Input Waveform - R
Y contains the Input Waveform - I
X contains the Output Waveform - R
Y contains the Output Waveform - I

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- The effect of the earth's magnetic field is not included in this model.

b. The effect of electron collisions with ions and neutral particles is not included in this model. The major effect resulting from electron collisions is a damping term which results in absorption of energy from the electromagnetic wave transversing the ionized medium. For frequencies above 100 MHz electron collisions have negligible effect on the phase behavior of the ionosphere as a function of frequency.

c. Reference:

Berkowitz, R. S. ed. Modern Radar Analysis, Evaluation, and System Design, John Wiley & Sons, Inc., New York, 1965, pp 349-354, 364-370.

d. Flow Chart: Page 9-117

e. Cross Reference Table: Page 9-223

6. THEORY OF OPERATION

The index of refraction of an ionized medium is given by the following expression:

$$n = \sqrt{1 - \frac{N_e e^2}{m\pi f^2}} \quad (1)$$

where: N_e is the electron density (electrons/cm³)

e is the electron charge (4.8×10^{-10} esu)

f is the frequency of this incident electromagnetic energy

Collisions between electrons and ions or neutral particles are neglected.

The phase velocity of the propagating wave, V_p , is given by the following expression:

$$V_p = c/n \quad (2)$$

where: c is the speed of light in free space

Therefore, combining equations 1 and 2 the following expression is obtained:

$$V_p = \frac{C}{\sqrt{1 - N_e e^2 / m \pi f^2}} \quad (3)$$

The phase velocity will approach infinity as the denominator of equation 3 decreases to zero. When this condition arises, further wave propagation is impossible. The frequency for which this occurs is called the critical frequency and is given by the following expression:

$$f_c^2 = \frac{N_e e^2}{m \pi} = 8.0 \times 10^7 N_e \text{ (Hz)}^2 \quad (4)$$

Equation 3 now becomes the following:

$$V_p = \frac{C}{\sqrt{1 - f_c^2 / f^2}} \quad (5)$$

The differential phase shift $d\phi$ that a wave will encounter in traversing an element of path length ds is given by the following expression:

$$d\phi = \frac{2\pi}{\lambda} ds \quad (6)$$

where: λ is the wavelength of the radiated electromagnetic energy

Since $\lambda = V_p / f$ the following expression is obtained by substitution of equation 5 into equation 6:

$$d\phi = \frac{2\pi f}{C} \sqrt{1 - f_c^2 / f^2} ds \quad (7)$$

This expression can be further simplified by adding the term $\frac{f_c^4}{4f^2}$ to the terms inside the square root

radical. For most radar applications $f_c \ll f$ so this will introduce a very small error. Equation 7 now becomes:

$$\begin{aligned} d\phi &= \frac{2\pi f}{C} \sqrt{\left(1 - \frac{f_c^2}{2f^2}\right)^2} ds \\ &= \frac{2\pi f}{C} \left(1 - \frac{f_c^2}{2f^2}\right) ds \end{aligned} \quad (8)$$

In order to determine the phase shift over a propagation path from s_1 to s_2 the following expression will be used

$$\begin{aligned} \phi &= \frac{2\pi f}{C} \int_{s_1}^{s_2} \left(1 - \frac{f_c^2}{2f^2}\right) ds \\ &= \frac{2\pi f}{C} \int_{s_1}^{s_2} ds - \frac{2\pi}{2Cf} \int_{s_1}^{s_2} f_c^2 ds \end{aligned}$$

substituting for f_c^2 the following is obtained:

$$\phi = \frac{2\pi f}{C} \int_{s_1}^{s_2} ds - \frac{\pi}{Cf} 8.0 \times 10^7 \int_{s_1}^{s_2} N_e ds \quad (9)$$

The first term of equation 9 is the linear phase shift due to propagation over a path of length $s_2 - s_1$ and will be omitted since it represents a time delay only

Therefore, the equation used to calculate the dispersive effect of the ionosphere is given by the following expression:

$$\phi(f) = -2\pi \frac{4.0 \times 10^7}{3.0 \times 10^{10}} \frac{1}{f} \quad \text{SEDENS}$$

where: $\text{SEDENS} = \int_{s_1}^{s_2} N_e ds = \text{integrated electron density}$

$$f = RFF\phi + k \Delta F, \quad -\frac{FEXT}{2 \Delta F} < k < \frac{FEXT}{2 \Delta F}$$

FEXT = frequency extent of input waveform representations

F = independent variable spacing of input waveform representations.

4-79

```

3104      SUBROUTINE LENS(X,Y)
3105      COMMON/LENK1/ VAR(500)
3106      DIMENSION X(1),Y(1)
3107      EQUIVALENCE (VAR(145),SEDENS 1,(VAR( 5),RFFO 1)
3108      * VARIABLE SEDENS IS INTEGRATED ELECTRON DENSITY ALONG
3109      * PROPAGATION PATH (ELECTRONS/CM*CM)
3110      DATA N193,N194,N195/-3,-2,-1/ CONS/ 1.3353E-03 /,PI2/6.2831853/
3111      NPIS=BOUL(X(N193))
3112      FREQ= (X(N194)+RFFO) *1.0E+09
3113      DELF=X(N195)*1.0E+09
3114      DO 200 J=1,NPIS
3115      THETA= CONS*SEDENS/FREQ
3116      THETA=(THETA-FLUAT(IFIX(THETA)))*PI2
3117      TMP=X(J)
3118      C=COS(THETA)
3119      S=SIN(THETA)
3120      X(J)=X(J)*C-Y(J)*S
3121      Y(J)=TMP*S+Y(J)*C
3122      FREQ=FREQ+DELF
3123      200 CONTINUE
3124      RETURN
3125      END

```

SUBROUTINE LAMPCP

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
LAMPCP	LTI-1 or LTI-4	458
LAMPRE	LTI-1 or LTI-4	456, 457

2. PURPOSE:

This subroutine is used to simulate a linear amplifier.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
GAIN	R	F	Amplifier gain

4. CALLING SEQUENCES:

CALL LAMPCP (XIN, YIN, XOUT, YOUT)

Where:

- XIN contains the Input Waveform - R
- YIN contains the Input Waveform - I
- XOUT contains the Output Waveform - R
- YOUT contains the Output Waveform - I

CALL LAMPRE (XIN, XOUT)

Where:

- XIN contains the Input Waveform
- XOUT contains the Output Waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. Flow Chart: Page 9-68
- b. Cross Reference Table: Page 9-217

6. THEORY OF OPERATION

The relationship between the input and output is given by the following expressions:

$$\begin{aligned} XOUT(J) &= GAIN * XIN(J) && ; \text{ LAMPCP \& LAMPRE} \\ YOUT(J) &= GAIN * YIN(J) && ; \text{ LAMPCP} \end{aligned}$$

2027 SUBROUTINE LAMPKP(XIN,YIN,XOUT,YOUT)
 2028 DIMENSION XIN(1),YIN(1),XOUT(1),YOUT(1)
 2029 COMMON/BLK1/ B(200)

482

05/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL FAUSIM

CARD NO	****	CONTENTS	****
2020		DATA N193,N194,N195/-3,-2,-1/	
2031		EQUIVALENCE (E(145) , GAIN)	
2032		MODE=1	
2033		GO TO 100	
2034		ENTRY LAMPKE(XIN,XOUT)	
2035		MODE=(	
2036	100	NPTS=EOUL(XIN(N193))	
2037		DO 200 J=1,NPTS	
2038		XOUT(J)=XIN(J)*GAIN	
2039	200	CONTINUE	
2040		IF (MODE.EQ.0) GO TO 500	
2041		DO 300 J=1,NPTS	
2042		YOUT(J)=YIN(J)*GAIN	
2043	300	CONTINUE	
2044		YOUT(N195)=YIN(N193)	
2045		YOUT(N194)=YIN(N194)	
2046		YOUT(N195)=YIN(N195)	
2047	500	XOUT(N193)=XIN(N193)	
2048		XOUT(N194)=XIN(N194)	
2049		XOUT(N195)=XIN(N195)	
2050		RETURN	
2051		END	

SUBROUTINE MTIFLT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
MTIFLT (MODEDF=1)	LTI-4	430,431
MTINCL (MODEDF=1)	LTI-4	432,433
MTIIFT (MODEDF=2)	NL-2	430,431
MTIINC (MODEDF=2)	NL-2	432,433

2. PURPOSE:

This subroutine is used to simulate a double delay MTI processor. Either floating point or integer arithmetic is selectable.

3. INPUT PARAMETERS

a. MODEDF = 1

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FF0	0	F	Feed - forward coefficient - 0 delay (no delay)
FF1	0	F	Feed - forward coefficient - 1 delay (single delay)
FB1	0	F	Feedback coefficient - 1 delay
FB2	0	F	Feedback coefficient - 2 delay (double delay)

b. MODEDF=2

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
IFF0N	0	I	Numerator of Feed - forward coef - 0 delay
IFF0D	R	I	Denominator of feed - forward coef - 0 delay
IFF1N	0	I	Numerator of feed - forward coef - 1 delay

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
IFF1D	R	I	Denominator of feed - forward coef - 1 delay
IFB1N	O	I	Numerator of feedback coef - 1 delay
IFB1D	R	I	Denominator of feedback coef - 1 delay
IFB2N	O	I	Numerator of feedback coef - 2 delay
IFB2D	R	I	Denominator of feedback coef. - 2 delay
NBITS	R	I	Number of bits to be used in storing one range sample in the arrays (ISR1 and ISR2) which represents the digital delay lines (includes sign bit).

4. CALLING SEQUENCES:

Floating point arithmetic

CALL MTIFLT (DIN, DOUT)

Where: DIN contains the Input Waveform

DOUT contains the Output Waveform

The range bit storage arrays (SR1 and SR2) are cleared prior to execution.

The entry point is used only if MODEDF=1.

Floating point arithmetic

CALL MTINCL (DIN, DOUT)

Where: DIN contains the Input Waveform

DOUT contains the Output Waveform

The range bin storage arrays (SR1 and SR2) are not cleared prior to execution except for the first execution of this subroutine. This entry point is used only if MODEDF=1.

Integer arithmetic

CALL MTIIFT (IN, IOUT)

Where: IN contains the Input Waveform

IOUT contains the Output Waveform

The range bin arrays (ISR1 and ISR2) are cleared prior to execution.

This entry point is used only if MODEDF=2.

Integer Arithmetic

CALL MTIINC (IN, OUT)

Where: IN contains the Input Waveform

IOUT contains the Output Waveform

The range bin storage arrays (ISR1 and ISR2) are not cleared prior to execution except for the first execution of this subroutine.

This entry point is used only if MODEDF=2.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The maximum number of range bins simulated is 2048. If more than 2048 range samples are contained in the input array, only the first 2048 are processed.
- b. The parameter MODEDF is used by the simulation controller to determine whether the floating point or the fixed point module entry points are to be used.
- c. The entry points which use integer arithmetic are considered nonlinear because of truncation in arithmetic and saturation in the delay line.
- d. Flow Chart: Page 9-201
- e. Cross Reference Table: 9-234

6. THEORY OF OPERATION

The block diagram of the two-delay MTI processor digital filters simulated by this module are shown in Figure MTIFLT-1 and MTIFLT-2. The Z-plane transfer function for each range bin (floating point arithmetic) is given by the following expression:

$$T(z) = \text{FF}\emptyset \frac{z^2 + \frac{\text{FF1}}{\text{FF}\emptyset} z + \frac{1}{\text{FF}\emptyset}}{z^2 - \text{FB1} z - \text{FB2}}$$

The Z-plane transfer function for each range bin (integer arithmetic) is approximated by the following expression:

$$T(z) = \frac{\text{IFF}\emptyset\text{N}}{\text{IFF}\emptyset\text{D}} \frac{z^2 + \frac{\text{IFF1N} \cdot \text{IFF}\emptyset\text{D}}{\text{IFF}\emptyset\text{N} \cdot \text{IFF1D}} z + \frac{\text{IFF}\emptyset\text{D}}{\text{IFF}\emptyset\text{N}}}{z^2 - \frac{\text{IFB1N}}{\text{IFB1D}} z - \frac{\text{IFB2N}}{\text{IFB2D}}}$$

The delay represented by the Z operator is determined by the radar pulse repetition interval.

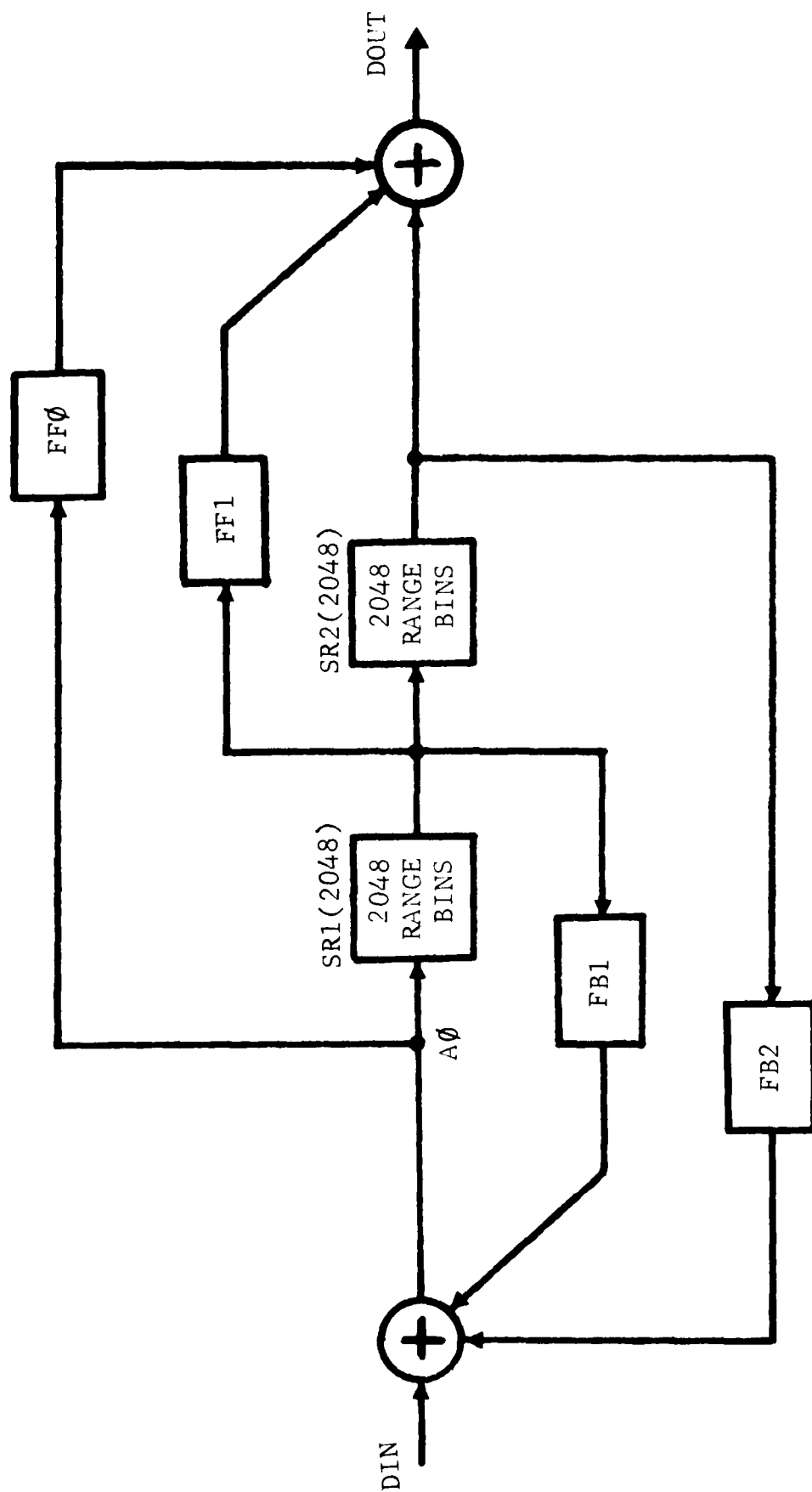


Figure MTIFLT-1 BLOCK DIAGRAM OF MTIFLT/MTINCL
(Floating-point Arithmetic)

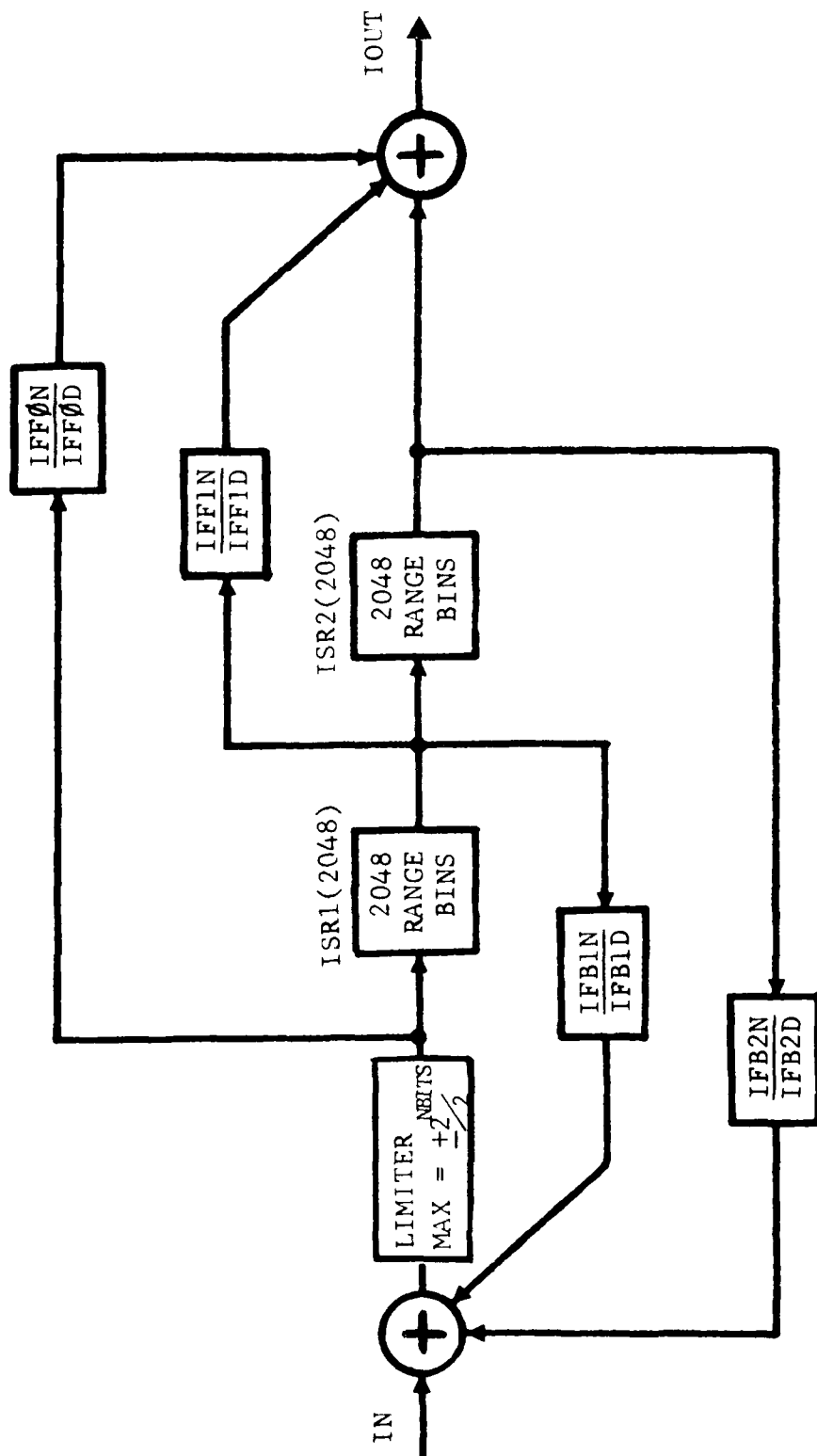


Figure MTIFLT-2 BLOCK DIAGRAM OF MTIFLT/MTINCL (Integer Arithmetic)

5199	SUBROUTINE MTINCL(DIN,DOUT)	UC7TS001 489
5200	COMMON/BLK1/BK1(500)	UC7TS002
5201	DIMENSION DIN(1),DOUT(1),IN(1),IDOUT(1),SK1(2048),SK2(2048),	UC7TS003
5202	* ISK1(2048),ISK2(2048)	UC7TS004
5203	EQUIVALENCE (SK1(1),ISK1(1)), (SK2(1),ISK2(1))	UC7TS005
5204	DATA N193,N194,N195,N196/-3,-2,-1,0/	UC7TS006
5205	DATA IFLG/0/	320
5206	EQUIVALENCE (BK1(21), IDMY), (BK1(66), FFO),	UC7TS008
5207	1 (BK1(69), FF1), (BK1(70), FB1),	UC7TS009
5208	2 (BK1(71), FB2)	UC7TS010
5209	EQUIVALENCE (BK1(160), IFFGN),(BK1(161), IFFUD),	UC7TS011
5210	* (BK1(162), IFFIN),(BK1(163), IFFID),	UC7TS012
5211	* (BK1(164), IFBIN),(BK1(165), IFBID),	UC7TS013
5212	* (BK1(166), IFBZN),(BK1(167), IFBZD),	UC7TS014
5213	* (BK1(169), NBITS)	UC7TS015
5214	ICON=1	401
5215	GO TO 40	402
5216	ENTRY MTINCL(DIN,DOUT)	403
5217	ICON=2	404
5218	40 IF(IFLG.EQ.1.AND.ICON.EQ.2) GO TO 50	405
5219	DO 20 J=1,2048	UC7TS016

08/18/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL KAUSIM

H-90

CARD NO	****	CONTENTS	****
5220		SR1(J)=0.0	UC7TS017
5221		SR2(J)=0.0	UC7TS018
5222	20	CONTINUE	UC7TS019
5223		IFLG=1	450
5224	50	CONTINUE	451
5225		N = 1800L(DIN(N193))	UC7TS021
5226		IF(N.LE.2048) GO TO 15	UC7TS022
5227		N=2048	UC7TS023
5228		WRITE(6,25)	UC7TS024
5229	25	FORMAT(' TOO MANY POINTS IN INPUT ARRAY....FIRST 2048 PROCESSED'	UC7TS025
5230		*)	UC7TS026
5231	15	DOOT(N193) = BLCL(N)	UC7TS027
5232		DOOT(N194) = DIN(N194)	UC7TS028
5233		DOOT(N195) = DIN(N195)	UC7TS029
5234		DO 10 J=1,N	UC7TS030
5235		A0= SR1(J)*FB1+SR2(J)*FB2+DIN(J)	UC7TS031
5236		DOOT(J)= A0*FF0+SR1(J)*FF1+ SR2(J)	UC7TS032
5237		SR2(J)=SR1(J)	UC7TS033
5238		SR1(J)=A0	UC7TS034
5239	10	CONTINUE	UC7TS035
5240		RETURN	UC7TS036
5241		ENTRY MT11F1(IN,DOOT)	UC7TS037
5242		ICCN=1	621
5243		GO TO 60	622
5244		ENTRY MT11NC(IN,ICUT)	623
5245		ICCN=2	624
5246	60	IF(IFLG.EQ.1.AND.ICCN.EQ.2) GO TO 70	625
5247		DO 100 J=1,2048	UC7TS038
5248		SR1(J)=0	UC7TS039

5249	ISK2(J)=0	4-90A UC7TS040
5250	100 CONTINUE	UC7TS041
5251	IFLG=1	670
5252	70 CONTINUE	671
5253	IF(NBITS.GT.0.AND.NBITS.LE.31) GO TO 95	UC7TS043
5254	NBITS=31	UC7TS044
5255	95 CONTINUE	UC7TS045
5256	MAX=2*(NBITS-1)	UC7TS046
5257	N=IN(N193)	UC7TS047
5258	IF(N.LE.2048) GO TO 105	UC7TS048
5259	N=2048	UC7TS049
5260	WRITE(6,25)	UC7TS050
5261	105 IOUT(N193)= N	UC7TS051
5262	IOUT(N194)=IN(N194)	UC7TS052
5263	IOUT(N195)=IN(N195)	UC7TS053
5264	DO 110 J=1,N	UC7TS054
5265	IO=(ISK1(J)*IFB1N)/IFB1D +(ISK2(J)*IFB2N)/IFB2D + IN(J)	UC7TS055
5266	IF(IAES(10).GT.MAX) IO=ISIGN(MAX,IO)	UC7TS056
5267	IOUT(J)=(IO*IFFON)/IFFOD +(ISK1(J)*IFF1N)/IFF1D + ISK2(J)	UC7TS057
5268	ISK2(J)=ISK1(J)	UC7TS058
5269	ISK1(J)=IO	UC7TS059
5270	110 CONTINUE	UC7TS060
5271	RETURN	UC7TS061
5272	END	

SUBROUTINE NONLIN

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
NONLIN	NL-1	401, 402

2. PURPOSE:

This subroutine is used to simulate a zero memory nonlinear device.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NPTS	R	I	Number of points used to specify the transfer function
TFN	R	F	Array containing the specification of the transfer function. The coefficients for the Jth point are the following:

TFN(1,J) = Input voltage

TFN(2,J) = Output voltage

4. CALLING SEQUENCES:

CALL NONLIN (A,\$mmmm)

Where:

A contains the Input Waveform

A contains the Output Waveform

\$mmmm is the statement number in the calling program to which control is returned if a discrepancy is detected in the input data.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The independent variable in the transfer function specification array must be monotonically increasing.
- b. The input data must be within the limits of the specified transfer function.
- c. Flow Chart: Page 9-135
- d. Cross Reference Table: Page 9-225

6. THEORY OF OPERATION

The input vs output characteristics of the device to be simulated are specified via the array TFN. A table lookup and linear interpolation scheme are used to compute the output value for each sample of the input waveform. An abnormal termination will occur if an input sample is outside the range of input values specified in the array TFN.

An example of the input data for this module is the nonlinear transfer function shown in Figure NONLIN-1. The associated input data statement is as follows:

\$NL401 TFN = -2.0, -2.0, -1.0, -2.0, 1.0, 2.0, 2.0,
2.0, NTFN = 4\$

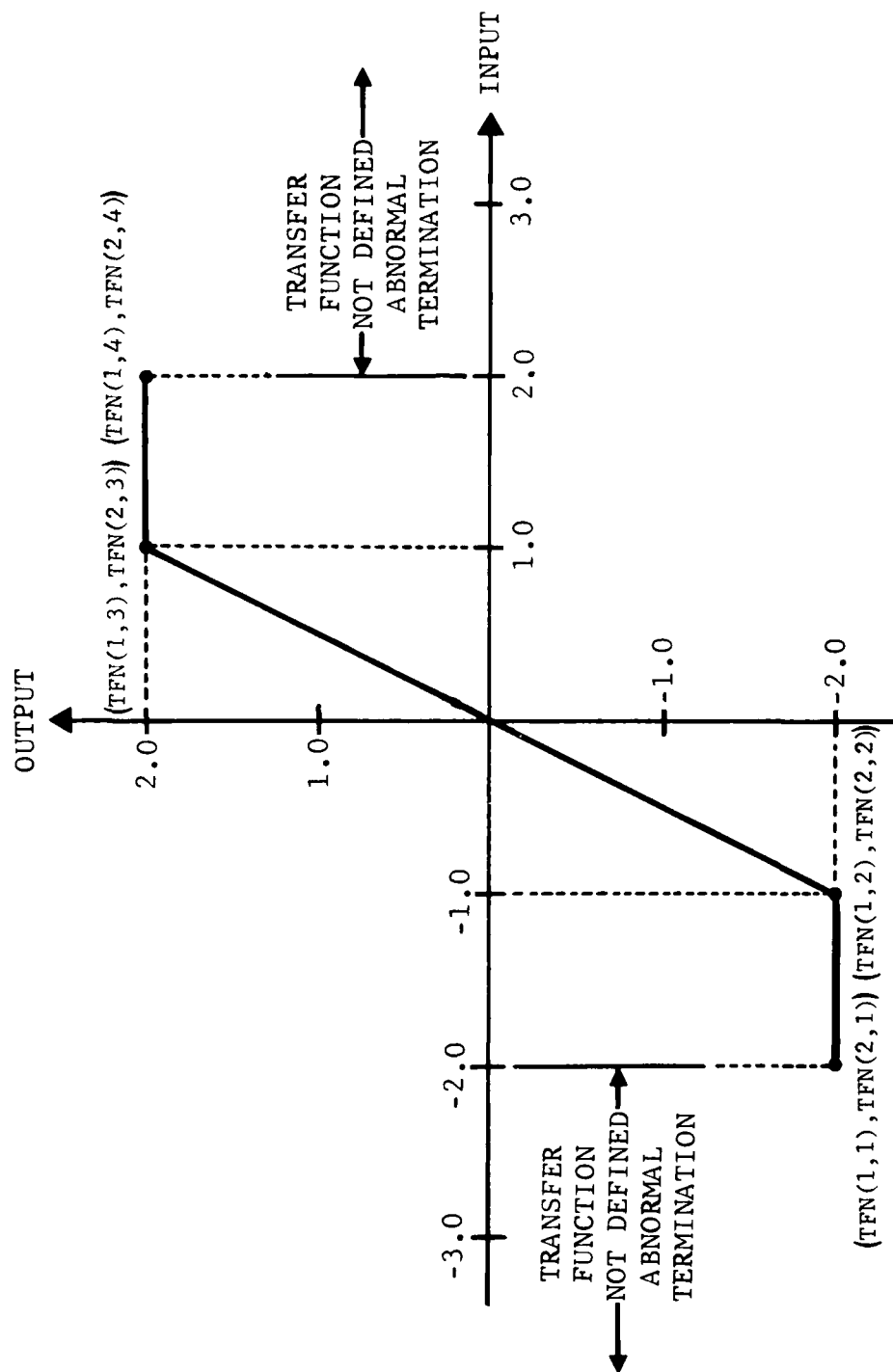


Figure NONLIN-1 EXAMPLE OF NON-LINEAR TRANSFER FUNCTION

4-93

```

3536      SUBROUTINE NONLIN(A,*)
3537      COMMON/BLK1/BLK1(500)
3538      DIMENSION A(1),TFN(2,50)
3539      EQUIVALENCE (BLK1( 21), IDMY          ), (BLK1( 67), NPIS          ),
3540      *          (BLK1(201), TFN(1,1)      )
3541      DATA N193,N194,N195,N196/-3,-2,-1,0/
3542      NIN = 1+ABS(A(N193))
3543      K=1
3544      DO 100 J=1,NIN
3545          GO TO 105
3546      103 K=K+1
3547          GO TO 105
3548      104 K=K-1
3549      105 IF (K.GT.NPIS) GO TO 500
3550          IF (K.LT.1) GO TO 500
3551          IF (A(J).LT.TFN(1,K))GO TO 104
3552          IF (A(J).GT.TFN(1,K+1))GO TO 103
3553          PCT = (A(J)-TFN(1,K)) / (TFN(1,K+1)-TFN(1,K))
3554          A(J) = TFN(2,K) + PCT*(TFN(2,K+1)-TFN(2,K))
3555      100 CONTINUE
3556          RETURN
3557      500 CONTINUE
3558          RETURN 1
3559          END

```

SUBROUTINE PHDEC

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
PHDEC	LTI-2	508, 509

2. PURPOSE:

This subroutine is used to simulate an analog phase decoder.

3. INPUT PARAMETERS

a. Automatic mode; INPTF = 0

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
SIMF0	R	F	Center frequency of waveform previously generated by FGENXY/FGENMP. This parameter is used to specify the center frequency of the tapped delay line.
INPTF	R	I	Set = 0 for this mode
SPW	R	F	Subpulsewidth of waveform previously generated by FGENXY/FGENMP. This parameter is used to specify the delay line tap spacing (TAPSPC).
XMITPC	R	F	Array containing the phase code of the waveform generated by FGENXY/FGENMP.
NSUBP	R	I	Number of subpulses in the waveform generated by FGENXY/FGENMP. This parameter specifies the number of delay line taps.

NOTE: No user supplied inputs required for this mode. All required parameters are initialized by FGENXY/FGENMP.

b. User Supplied data mode; INPTF = 1

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FØDEC	R	F	Center frequency of the tapped delay line.
TAPSPC	O	F	Spacing between delay line taps. This parameter is set equal to SPW if it is not specified by the user.
INPTF	R	I	Set = 1 for this mode
XMITPC	R	F	Array containing the phase code of the waveform generated by FGENXY/FGENMP. (Initialized by FGCENXY/FGENMP)
NSUBP	R	I	Number of subpulses in the waveform generated by FGENXY/FGENMP. This parameter specifies the number of delay line taps. (Initialized by FGENXY/FGENM)

4. CALLING SEQUENCES:

CALL PHDEC (X,Y)

Where: X contains the Output Waveform - R
Y contains the Output Waveform - I

5. RESTRICTIONS, RECOMMENDATIONS, MISCELLANEOUS DATA

a. If INPTF=1 and TAPSPC is specified, then TAPSPC should be an integral multiple of the period of the waveform center frequency, i.e. $TAPSPC = N * 1.0 / SIMFØ$ where N is an integer. Otherwise, the taps will not have the desired phase shifters as specified in the array XMITPC.

b. Flow Chart: Page 9-78

c. Cross Reference Table: Page 9-218

6. THEORY OF OPERATION

This module generates the transfer function of a tapped analog delay line. The main use of this module is to simulate a surface acoustic wave device. A block diagram of this device represented by this module is shown in Figure PHDEC-1. This module is structured to represent only the taps (energy pickoffs) of the delay line. The wave launches must be specified separately in terms of its transfer function. This can be done via either the FILT or the WEIT modules. A pictorial diagram of a surface acoustic wave device and its representation are shown in Figures PHENC-2(a) and -2(b). The impulse response of the device represented by this module is given by the following expression:

$$h(t) = \sum_{K=1}^{K=NSUBP} g_K \delta(t - Kt_d) e^{j\theta_K}$$

where: g_K = tap gain
 t_d = tap spacing
 θ_K = tap phase shift

The transfer function computed by this module is given by the following expression:

$$H(f) = \mathcal{F}[h(t)] = \int_{-\infty}^{\infty} \sum_{k=1}^{NSUBP} g_k \delta(t - kt_d) e^{j\theta_k} e^{-j2\pi ft} dt$$

Interchanging the order of integration and summation

$$H(f) = \sum_1^{NSUBP} g_k e^{j\theta_k} \int_{-\infty}^{\infty} \delta(t - kt_d) e^{-j2\pi ft} dt$$

$$H(f) = \sum_1^{NSUBP} g_k e^{j\theta_k} e^{-j2\pi fkt_d}$$

This version of the phase decoder module sets all $g_k = 1.0$. If a phase decoder with gain weighting or errors is desired then the DFT routine should be used to compute the transfer function.

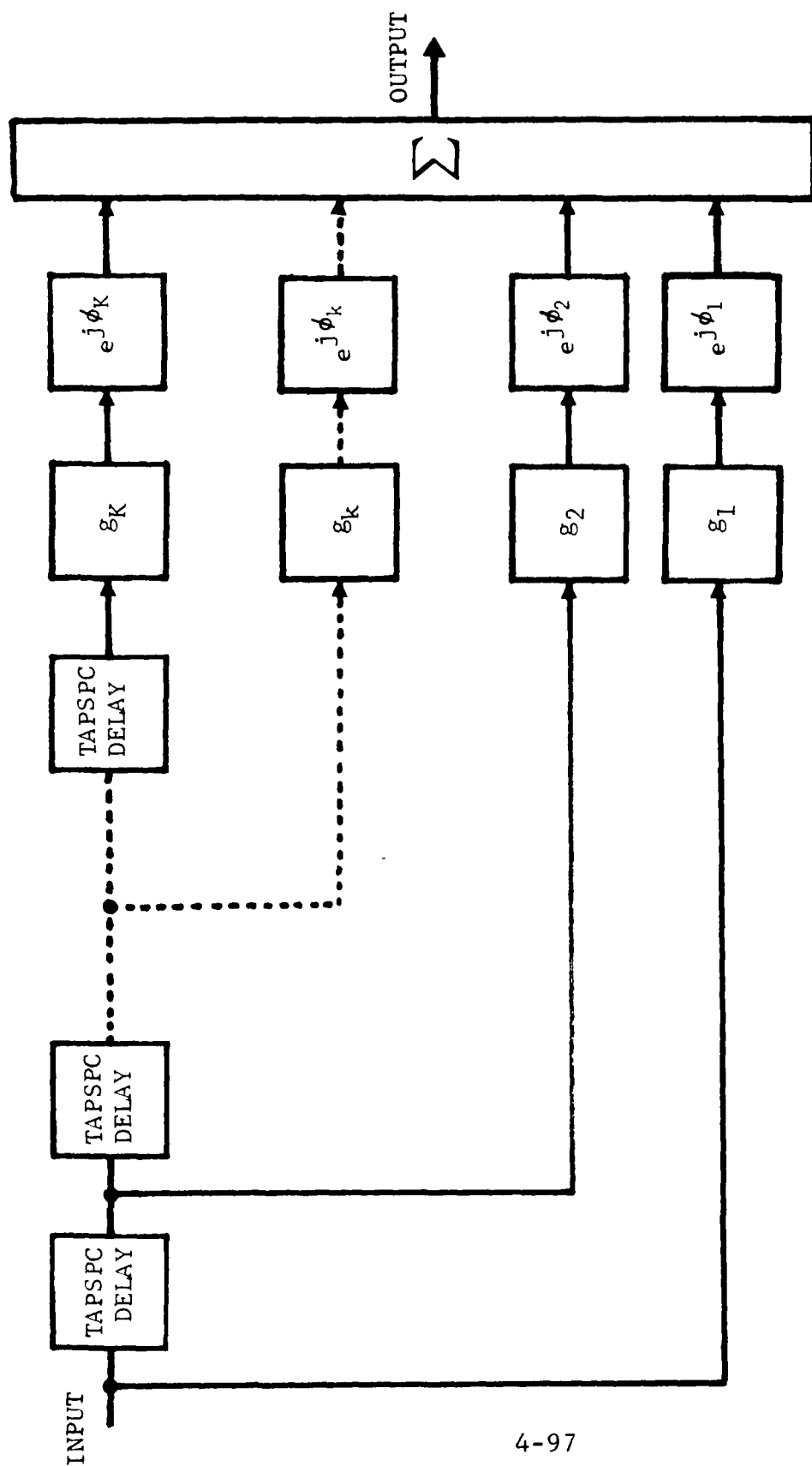


Figure PHDEC-1 BLOCK DIAGRAM OF A TAPPED DELAY LINE PHASE DECODER

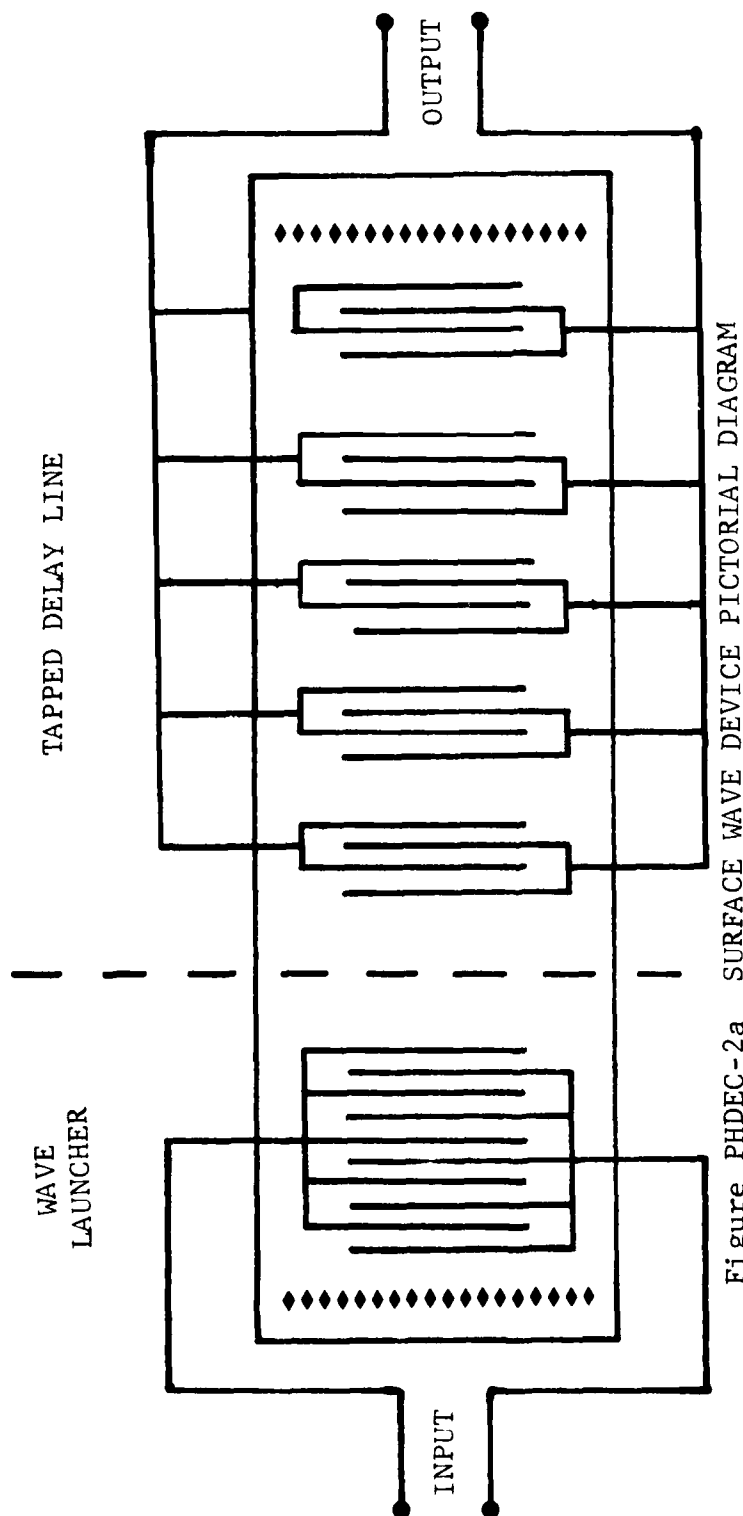


Figure PHDEC-2a SURFACE WAVE DEVICE PICTORIAL DIAGRAM

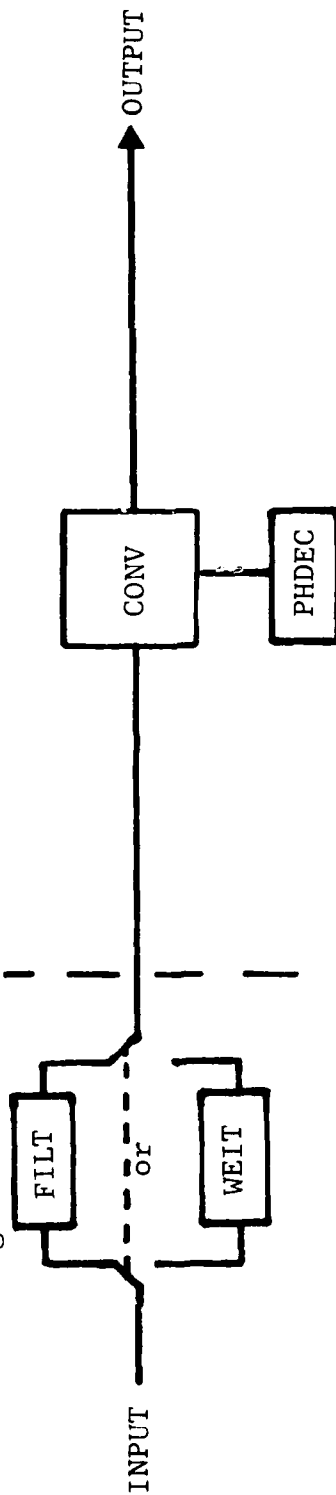


Figure PHDEC-2b SIMULATION MODEL BLOCK DIAGRAM


```

2221      SUBROUTINE DETFUC(X,Y)
2222      DIMENSION X(1),Y(1)
2223      COMMON/THCULZ/ ANJTPC(500),NSUPP,SPW
2224      COMMON/SERIZ/ ITEMF(199),NIME,UN(5,100)
2225      EQUIVALENCE (ITEMF(102), FODEC), (ITEMF(154), TAFSPC),
2226      *              (ITEMF(1), SIMFC), (ITEMF(155), INPLE)
2227      IF (INPLE.EQ.1) SIMFC=FODEC
2228      IF (INPLE.EQ.0) TAFSPC=SPW
2229      IF (INPLE.EQ.1.AND.TAFSPC.EQ.0.0) TAFSPC=SPW
2230      IF ASS=NSUPP
2231      TIME=0.0
2232      100 CONTINUE
2233      NIME=ITEMF(5)
2234      IF (IFASS.EQ.100) NIME=100
2235      CC=200-J=1/NIME
2236      UN(1,J)=ANJTPC(IFASS+1-J)
2237      UN(2,J)=TIME
2238      UN(3,J)=X*Y
2239      UN(4,J)=TIME*TAFSPC
2240      200 CONTINUE
2241      IF (IFASS.EQ.0) CALL DETFUC(X,Y)
2242      IF (IFASS.NE.0) CALL DETNCL(X,Y)
2243      IF (NIME.NE.200) RETURN
2244      IF ASS=IFASS=100
2245      IF (IFASS.EQ.0) RETURN
2246      CC=100
2247      END

```

SUBROUTINE PHENC

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
PHENC/FCGNXY	SOU-1 or LTI-3	506
PHENC/FGENMP	SOU-1 or LTI-3	507

2. PURPOSE:

This subroutine generates a phase code which is subsequently used by FGENXY/FGENMP to generate a binary phase coded waveform. Either the module FGENXY or FGENMP is automatically scheduled for execution.

3. INPUT PARAMETERS

a. MODEPH = 1; Barker Phase Code

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
MODEPH	R	I	Set = 1 for this mode
NSUBP	R	I	Number of subpulses to be generated. The allowable values of this parameter and the corresponding Barker codes are as follows:
			2 + -
			3 + + -
			4 + + - +
			5 + + + - +
			7 + + + - - + -
			11 + + + - - - + - - + -
			13 + + + + + - - + + - + - +

b. MODEPH = 2; Pseudo Random Phase Code

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
MODEPH	R	I	Set = 2 for this mode
NSUBP	R	I	Number of subpulses. Maximum value = 300 for this mode.

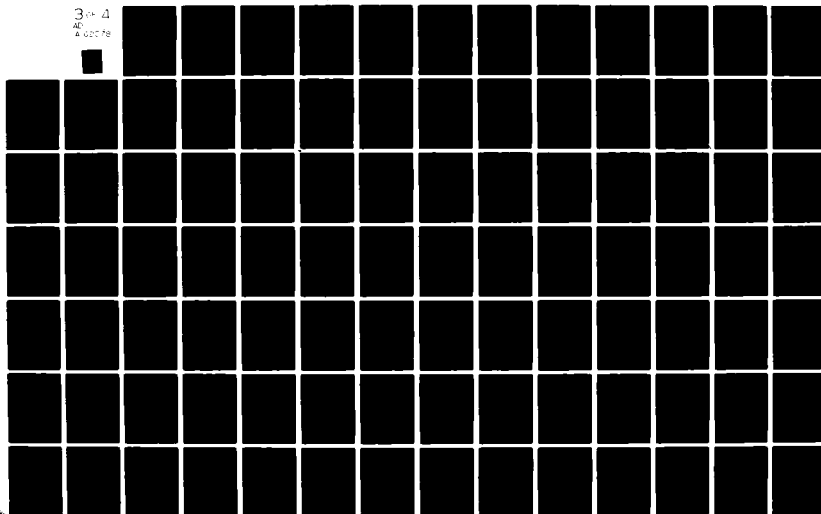
AD-A102 278

GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV F/G 17/9
ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING, VOLUME II, PAR--ETC(U)
JUN 76 R J HANCOCK, F H CLEVELAND F30602-73-C-0380

UNCLASSIFIED

RADC-TR-76-186-VOL-2-PT-1 NL

3 of 4
40
4 copies



<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
IPY	R	I	Array representing the shift register used to generate the pseudo random sequence. The storage cells of this shift register contains either 1 or 0. The user can specify the initial condition of this register if desired.
NSR	R	I	Number of stages in the shift register.

c. MODEPH = 3; User Specified Phase Code

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
MODEPH	R	I	Set = 3 for this mode
NSUBP	R	I	Number of subpulses. For this mode this parameter is limited to 35.
CODE	R	B	Word containing the phase code. Each bit of this 36 bit word represents the phase of one subpulse. The phase for the first subpulse is determined by the LSB; i.e. right most bit.

4. CALLING SEQUENCES:

CALL PHENC (\$mmmm)

Where: mmmm is the statement number to which control will be transferred if an error is detected in the input data.

5. RESTRICTIONS, RECOMMENDATIONS, MISCELLANEOUS DATA

a. Reference for Barker and Pseudo random phase codes:

Skolnik, M. I.: RADAR HANDBOOK, McGraw-Hill Book Company, New York, 1970, pp 20-19 and 20-20.

b. Flow Chart: Page 9-167

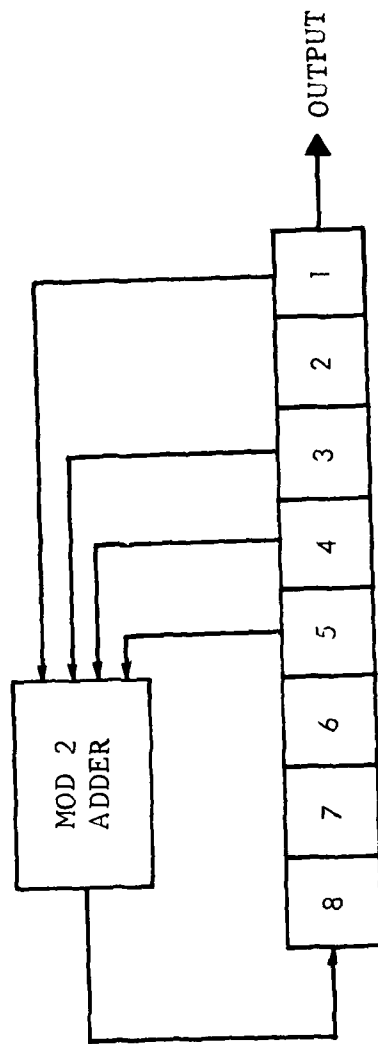
c. Cross Reference Table: Page 9-229

6. THEORY OF OPERATION

This module generates a phase code determined by the input parameters and loads the array PCODE with the phase code. The operations performed for $MODEPH = 1$ or 3 consist of connecting bits of the word CODE into phase angles. A binary "1" is converted into 180 degrees and a binary "0" is converted into 0 degrees.

Figure PHDEC-1 is a block diagram of the shift registers used to generate the pseudo random phase codes. The parameter NSR determines the selection of shift register taps (Reference Skolnik, p 20-20, table 6). The output from the delay line is converted to a phase code according to the procedure outlined in the preceeding paragraph.

8 STAGE SEQUENCE



2,3,4,5,6 or 7 STAGE SEQUENCE

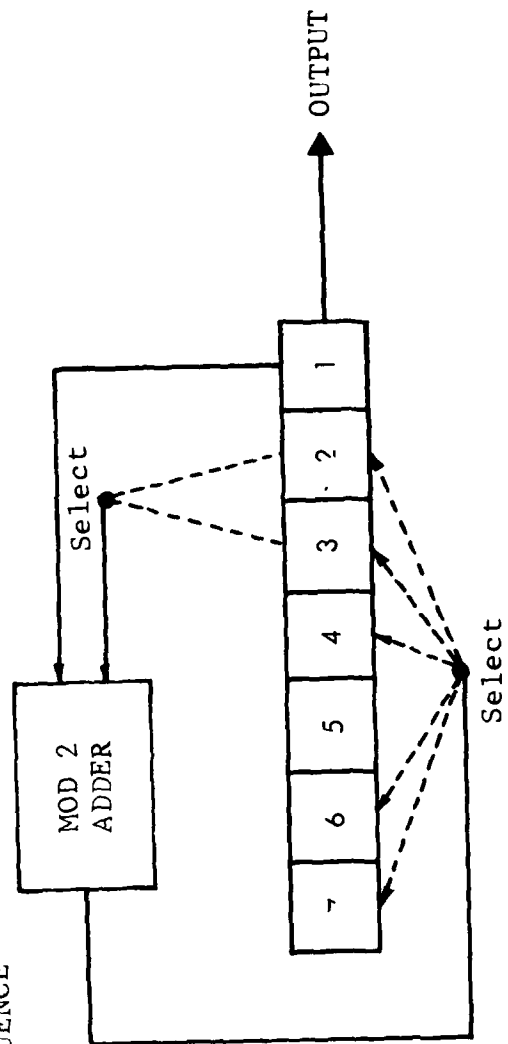


Figure PHENC-1 BLOCK DIAGRAM OF PSEUDO-RANDOM PHASE CODE GENERATORS

4-104

```

4014 SUBROUTINE PHENC(*)
4015 COMMON/BERZ/ B(200),PCODE(300)
4016 EQUIVALENCE (B( 96), NSUMF), (B(163), MODEPH),
4017 *          (B(184), NSF  ), (B(126), IPY(1) ),
4018 *          (B(194), CODE )
4019 DIMENSION IPY(5),BCOMP(15)
4020 DATA /CODE/ 1,0,01,01,02,02,00,015,00,00,00,000,00,001,7,N12,1/
4021 IF (MODEPH.EQ.1) CODE=BCOMP(NSUMF)
4022 IF (MODEPH.EQ.1.OR.MODEPH.EQ.3) GO TO 100
4023 IT=2
4024 IF (NSR.EQ.5.OR.NSR.EQ.8) IT=5
4025 DO 300 J=1,NSDEP
4026 ISUM=IPY(1)+IPY(IT)
4027 IF (NSR.NE.0) GO TO 150
4028 ISUM=ISUM+IPY(4)+IPY(5)
4029 IF (ISUM.EQ.4) ISUM=0
4030 IF (ISUM.EQ.5) ISUM=1
4031 IF (ISUM.EQ.6) ISUM=0
4032 IPY(NSR+1)=ISUM
4033 PCODE(J)=100.0*IPY(1)
4034 DO 100 K=1,NSR
4035 IPY(K)=IPY(K+1)
4036 GO TO 100
4037 CONTINUE
4038 GO TO 100
4039 CONTINUE
4040 RETURN
4041 END
4042 IF (CODE.EQ.1) RETURN, 1
4043 K=0+MODEP
4044 DO 300 J=1,NSDEP
4045 IF (FEI(K+J,1,001))
4046 PCODE(J)=FEI*100.0
4047 CONTINUE
4048 RETURN
4049 END

```

SUBROUTINE RDIGFL

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
RDIGFL	LTI-4	405
RDFNCL	LTI-4	406

2. PURPOSE:

This subroutine is used to simulate a double delay digital filter and is capable of synthesizing a filter transfer function having two poles and two zeros.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FF0	0	F	Feed - forward coefficient - 0 delay
FF1	0	F	Feed - forward coefficient - 1 delay
FB1	0	F	Feedback coefficient - 1 delay
FB2	0	F	Feedback coefficient - 2 delay

4. CALLING SEQUENCES:

CALL RDIGFL (X,Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

The storage registers (RR1, RR2, RI1, and RI2) are cleared before execution begins.

CALL RDFNCL (X,Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the output Waveform - I

The storage registers are not cleared before execution begins.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. Flow Chart: Page 9-100
- b. Cross Reference Table: Page 9-221

6. THEORY OF OPERATION

The block diagram of the two delay digital filter simulated by this module is shown in Figure RDIGFL-1. The Z-plane transfer function is given by the following expression:

$$T(Z) = FF\emptyset \frac{Z^2 + \frac{FF1}{FF\emptyset} Z + 1/FF\emptyset}{Z^2 - FB1 Z - FB2}$$

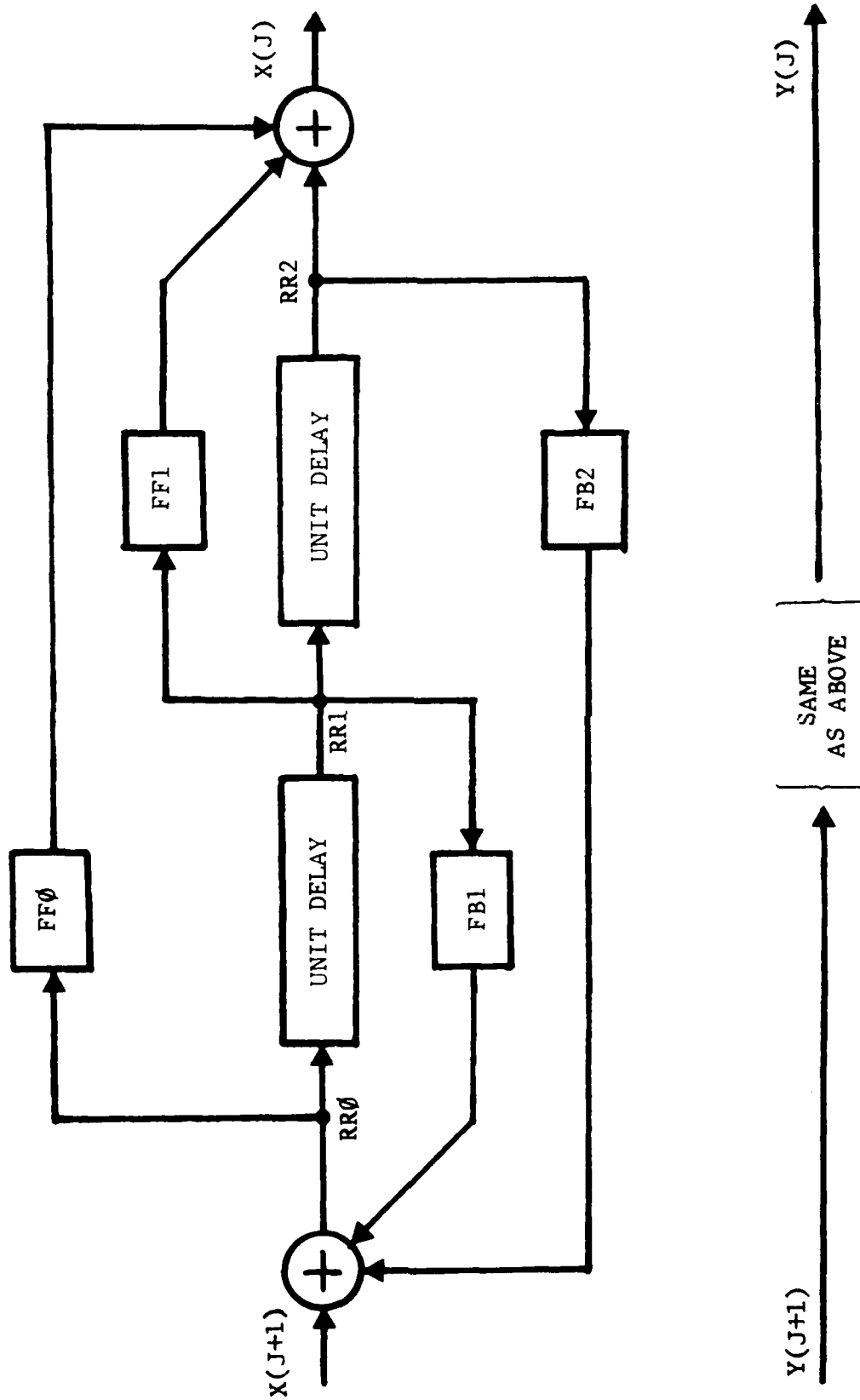


Figure RDIGFL-1 BLOCK DIAGRAM OF RDIGFL/RDFNCL

2670 SUBROUTINE RDIGFL (X,Y)
 2671 COMMON/BLK1/BLK1(500)
 2672 EQUIVALENCE (BK1(68), FF0), (BK1(69), FF1), (BK1(70), FB1),
 2673 * (BK1(71), FH2)
 2674 DATA N143/-3/
 2675 DIMENSION X(1),Y(1)
 2676 RK1=0.0
 2677 RK2=0.0
 2678 RI1=0.0
 2679 RI2=0.0
 2680 ENTRY RUFNCL (X,Y)
 2681 N=BLDCL(X(N143))
 2682 DO 10 J=1,N
 2683 RKU=RK1*FB1+RK2*FF2+X(J)
 2684 RIU=RI1*FB1+RI2*FF2+Y(J)
 2685 X(J)=RKU*FF0+RK1*FF1+RR2
 2686 Y(J)=RIU*FF0+RI1*FF1+RI2
 2687 RR2=RK1
 2688 RK1=RKU
 2689 RI2=RI1
 2690 RI1=RIU
 2691 10 CONTINUE
 2692 RETURN
 2693 END

4-108

SUBROUTINE RNDARY

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
RNDARY	SOU-1	214, 215
ADDRND	SOU-1S	235, 236
ADRND	SOU-1S	237

2. PURPOSE:

This subroutine loads an array with random numbers generated by the random number generator function subroutine RRAND.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NPTS	R	I	Number of random distribution samples to be loaded into the array (RNDARY only).
NTYPE	R	I	Code indicating the type of random distribution to be used (RNDARY and ADDRND only).
TI	R	F	Simulation sampling increment (RNDARY only).

4. CALLING SEQUENCES:

CALL RNDARY(RND)

Where: RND contains the Output Waveform

CALL ADDRND(RND)

Where: RND contains the Input Waveform

RND contains the Output Waveform

CALL ADRNDC(RND,RNDY)

Where: RND contains the Input Waveform - R
 RNDY contains the Input Waveform - I
 RND contains the Output Waveform - R
 RNDY contains the Output Waveform - I

 This module is used only to add
 Gaussian samples to the input waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. User supplied function subprogram RRAND is called by the subprogram and must be initialized via NL101 or NL102 before this subroutine is executed.
- b. The random distribution code (NTYPE) is as follows:
 - (1) Floating point uniform distribution with specified mean and extent.
 - (2) Floating point Rayleigh distribution with specified standard deviation.
 - (3) Floating point Gaussian distribution with specified mean and standard deviation.
 - (4) Positive integer uniform distribution from 0 to $2^{*}36$.
 - (5) Not used by this subprogram.
 - (6) Swerling Target Models #1 & #2 with specified mean cross section.
 - (7) Swerling Target Models #3 & #4 with specified mean cross section.
 - (8) Sine distribution.
- c. Flow Chart: Page 9-150
- d. Cross Reference Table: Page 9-227

6. THEORY OF OPERATION

The mechanization equations for each entry point are the following:

RNDARY: $RND(J) = RRAND(NTYPE) ; KJ \leq NPTS$

ADDRND: $RND(J) = RND(J) + RRAND(NTYPE)$; the range of J is determined by the input array.

ADRND: $RND(J) = RND(J) + RRAND(3)$
 $RNDY(J) = RNDY(J) + DUM$

where: $RRAND(3)$ contains the cosine gaussian distribution sample

DUM contains the sine gaussian distribution sample

4-112

```

3887      SUBROUTINE KNDARY(KND)
3888      COMMON/ERR1/ERR1(500)
3889      COMMON/ERRKND/ KND0AT(141)
3890      DIMENSION KND(1),KNDY(1)
3891      EQUIVALENCE (ERR1( 44), NPTS  ),(ERR1( 45), NTYPE)
3892      EQUIVALENCE (ERR1( 12), T1)
3893      DATA N193,N194,N195,N196/-3,-2,-1,0/
3894      C
3895      DO 10 I=1,NPTS
3896      KND(I)=KRAND(NTYPE)
3897      10  CONTINUE
3898      KND(N193)=B00L(NPTS)
3899      KND(N194)=0.0
3900      KND(N195)=11
3901      RETURN
3902      ENTRY ADDKND(KND)
3903      NPTS=B00L(KND(N193))
3904      DO 20 I=1,NPTS
3905      KND(I)=KND(I)+KRAND(NTYPE)
3906      20  CONTINUE
3907      C
3908      RETURN
3909      ENTRY ADKND(KND,KNDY)
3910      NPTS=B00L(KND(N193))
3911      DO 40 I=1,NPTS
3912      KND(I)=KND(I)+KRAND(NTYPE)
3913      KNDY(I)=KNDY(I)+KND0AT(4)
3914      40  CONTINUE
3915      RETURN
3916      END

```

SUBROUTINE SHIFT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
SHIFT	LTI-1	224,225
SHIFTS	LTI-1S	226,227
RSHIFT	LTI-1	229,230
RSHFTS	LTI-1S	231,232

2. PURPOSE:

This subroutine is used to delay a waveform in time and/or introduce a constant phase shift.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
TØ	R	F	Time delay
THT	R	F	Phase shift
TJIT	O	F	Time delay jitter
SIMFØ	R	F	Center frequency to be used in computing time delay effect.

4. CALLING SEQUENCES:

CALL SHIFT (X,Y,A,B)

Where: X contains the Input Waveform - R
Y contains the Input Waveform - I
A contains the Output Waveform - R
B contains the Output Waveform - I

The basic mechanization equation for this entry point is the following:

$$A(J)+jB(J) = [X(J)+jY(J)] * e^{j[THT-2\pi(FSTRT+FI*J)T\emptyset]}$$

Where: FI = frequency increment
 FSTRT = frequency corresponding to the first element of the input array

CALL SHIFTS (X,Y,A,B)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 A contains the Output Waveform - R
 B contains the Output Waveform - I

The basic mechanization equation for this entry point is the following:

$$A(J)+jB(J) = A(J)+jB(J) + [X(J)+jY(J)] * e^{j[THT-2\pi(FSTRT +FI*J)T\emptyset]}$$

CALL RSHIFT (X,Y,A,B)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 A contains the Output Waveform - R
 B contains the Output Waveform - I

The basic mechanization equation for this entry point is the following:

$$A(J) + jB(J) = [X(J)+jY(J)] * e^{j[THT-2\pi(SIMF\emptyset+FI*J)(T\emptyset+p\cdot TJIT)]}$$

where p is a sample of a random process having a uniform distribution from -1/2 to 1/2.

CALL RSHFTS(X,Y,A,B)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 A contains the Output Waveform - R
 B contains the Output Waveform - I

The basic mechanization equation for this entry point is the following:

$$A(J)+jB(J)=A(J)+jB(J)+[X(J)+jY(J)]*e^{j[THT-2\pi(SIMF\theta+FI*J)\cdot(T\theta+p\cdot TJIT)]}$$

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. Flow Chart: Page 9-63

b. Cross Reference Table: Page 9-216

6. THEORY OF OPERATION

If the Fourier transform pair is given for the signal $s(t)$:

$$s(t) \longleftrightarrow S(f)$$

$$\text{then } s(t-t_0) \longleftrightarrow S(f) e^{-j2\pi f t_0}$$

This expression was derived in reference 1.

In addition the introduction of a constant phase shift in the time waveform results in the same constant phase shift in the frequency spectrum, e.g.

$$s(t)e^{j\theta} = S(f) e^{j\theta}$$

Note: Since these subroutines have classification code LTI-1 or 1S the input and output waveforms are in the frequency domain.

1901 SUBROUTINE SHIFT(X,Y,A,B)
 1902 COMMON/BLK1/ BK1(200)
 1903 EQUIVALENCE (BK1(180),T0),(BK1(181),THT),(BK1(182),TJ1)
 1904 * , (BK1(8),SIMF0)
 1905 DIMENSION X(1),Y(1),A(1),B(1)
 1906 DATA N193,N194,N195,N196/-3,-2,-1.0/
 1907 DATA P12,EX/0.2831853,2.9103830E-11/
 1908 C
 1909 ICCN=0
 1910 TUK=TU
 1911 GO TO 10
 1912 C
 1913 ENTRY KSHIFT(X,Y,A,B)

4-116

1780

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

4-116a

CONTENTS

CARD NO

1914	ICUN=0
1915	GO TO 27
1916	C
1917	ENTRY SHIFTS(X,Y,A,B)
1918	ICUN=1
1919	TER=10
1920	GO TO 10
1921	C
1922	ENTRY ASHIFTS(X,Y,A,B)
1923	ICUN=1
1924	GO TO 27
1925	C
1926	10 NPTS=1000(XCN193)
1927	IF(TER.EC.0.0)GO TO 20
1928	NPTS2= NPTS/2
1929	TRASS=1
1930	K=HPTS+1
1931	KUL=1
1932	TH=TH/20000
1933	PH=TIMPH*TER+TH
1934	PLPH=X(NPTS)*TER
1935	15 PH= (PH-AINT(PH))*PI2
1936	PLPH= (PLPH-AINT(PLPH))*PI2
1937	ACCEL= COS(PLPH)
1938	ASDEL= SIN(PLPH)
1939	CS= COS(PH)
1940	SN= SIN(PH)
1941	GO TO 25
1942	C

4-117

```

1940      TIT=TIME*FACZ/1000.0
1941      ACCEL=1.0
1942      ASCEL=0.0
1943      CS= COS( TH)
1944      SN= SIN( TH)
1945      REFID=REFID
1946      IPASS=2
1947      N=1
1948      KLOC=2
1949      C
1950      DO 100 I=1,1000000
1951      C
1952      IF (TIME-REFID)/10.0 .LT. 1.0 .AND. (TIME-REFID)/10.0 .GT. 0.0
1953      THEN
1954      TIME=REFID+10.0
1955      IF (TIME-REFID)/10.0 .LT. 1.0 .AND. (TIME-REFID)/10.0 .GT. 0.0
1956      THEN
1957      TIME=REFID+10.0
1958      C
1959      DO 100 I=1,1000000
1960      C
1961      DO 100 I=1,1000000
1962      C
1963      DO 100 I=1,1000000
1964      C
1965      DO 100 I=1,1000000
1966      C
1967      DO 100 I=1,1000000
1968      C
1969      DO 100 I=1,1000000
1970      C
1971      DO 100 I=1,1000000
1972      C
1973      DO 100 I=1,1000000
1974      C
1975      DO 100 I=1,1000000
1976      C
1977      DO 100 I=1,1000000
1978      C
1979      DO 100 I=1,1000000
1980      C
1981      DO 100 I=1,1000000
1982      C
1983      DO 100 I=1,1000000
1984      C
1985      DO 100 I=1,1000000
1986      C
1987      DO 100 I=1,1000000
1988      C
1989      DO 100 I=1,1000000
1990      C
1991      DO 100 I=1,1000000
1992      C
1993      DO 100 I=1,1000000
1994      C
1995      DO 100 I=1,1000000
1996      C
1997      DO 100 I=1,1000000
1998      C
1999      DO 100 I=1,1000000
2000      C

```

08/11/77

INPUT LISTING

AUTOFLOW CHART SET - FLOW/SOL - FADSIM

4-117a....

LINE NO

CONTENTS

```

1972      DE 400 L=1,NFIDZ
1973      TIME=X(K)
1974      A(K)=A(K)+TIME*CS-Y(K)*SN
1975      L(K)=L(K)+Y(K)*CS+TIME*SN
1976      TEMP=L
1977      CS=CS*ACCEL-CH*ACCEL
1978      SN=SN*ACCEL+TEMP*ACCEL
1979      K=K*NDL
1980      GOTO CONTING
1981      C
1982      GOTO DEFAULACCEL OF TO GO
1983      IF ACCEL
1984      K=NFIDZ
1985      KDEL=1
1986      FPF=1-CCF1
1987      CCFPF=1-CCF1
1988      GOTO 10
1989      GOTO CONTING
1990      C
1991      X(1981)=X(1981)
1992      X(1982)=X(1982)
1993      X(1983)=X(1983)
1994      X(1984)=X(1984)
1995      X(1985)=X(1985)
1996      X(1986)=X(1986)
1997      X(1987)=X(1987)
1998      X(1988)=X(1988)
1999      X(1989)=X(1989)
2000      C
2001      J=1
2002      J=

```

SUBROUTINE SWPINT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
SWPINT	LTI-4	434,435
NCSWPI	LTI-4	436,437

2. PURPOSE:

This subroutine is used to simulate a video post detection sweep integrator.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
FBCK	R	F	Feedback coefficient

4. CALLING SEQUENCES:

CALL SWPINT (DIN, DOUT)

Where: DIN contains the Input Waveform

DOUT contains the Output Waveform

The range bin storage array (SRI) is cleared before execution begins.

CALL NCSWPI (DIN, DOUT)

Where: DIN contains the Input Waveform

DOUT contains the Output Waveform

The range bin storage array (SRI) is not cleared prior to execution except for the first execution of this subroutine.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The maximum number of range bins simulated is 2048.
If more than 2048 range samples are contained in

the input array, only the first 2048 are processed.

b. Flow Chart: Page 9-204

c. Cross Reference Table: Page 9-234

6. THEORY OF OPERATION

The block diagram of the video sweep integrator simulated by this module is shown in Figure SWPINT-1. The Z-plane transfer function for each range bin is given by the following expression:

$$T(Z) = \frac{Z}{Z - FBCK}$$

The delay represented by the Z operator is determined by the radar pulse repetition interval.

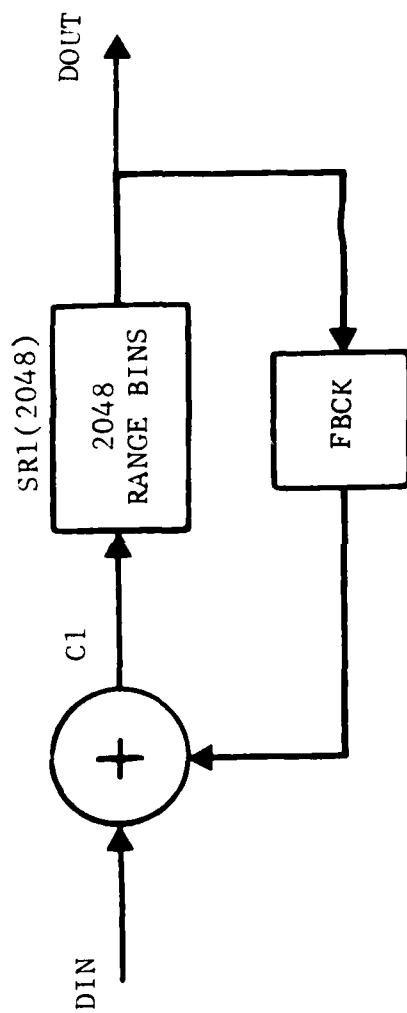


Figure SWPINT-1 BLOCK DIAGRAM OF SWPINT/NCSWPI

4-121

5273	SUBROUTINE SWPINT(DIN,DCUT)	UC7PL002
5274	COMMON/DEN1/DK1(500)	UC7PL001
5275	DIMENSION DIN(1),DCUT(1),SK1(2048)	UC7PL002
5276	EQUIVALENCE (DK1(21), IDMY), (DK1(75), FLEK)	UC7PL003
5277	DATA N193,N194,N195,N196/-3,-2,-1,0/	UC7PL004
5278	DATA IFLG/0/	UC7PL005
5279	ICCN=1	60
5280	GO TO 40	61
5281	ENTRY MCSWPINT(DIN,DCUT)	62
5282	ICCN=2	63
5283	40 IF(IFLG.EQ.1.AND.ICCN.EQ.2) GO TO 50	64
5284	GO TO 30 IF I=1,2,4,8	65
5285	SK1(I)=0.0	UC7PL007
5286	20 CONTINUE	UC7PL008
5287	IFLG=1	UC7PL009
5288	50 CONTINUE	100
5289	N = 1900+(1.0-N195)	101
5290	IF(N.EQ.0) GO TO 25	UC7PL011
5291	N=1951	UC7PL012
5292	WRITE(6,55)	UC7PL013
5293	55 FORMAT(' TOO MANY POINTS IN INPUT ARRAY....FIRST 2048 PROCESSED')	UC7PL014
5294	25 DCUT(N193)= DCUT(N)	UC7PL015
5295	DCUT(N194) = DIN(N194)	UC7PL016
5296	DCUT(N195) = DIN(N195)	UC7PL017
5297	GO TO 30 IF I=1,2	UC7PL018
5298	DI=DI*(I)+SK1(I)*FLEK	UC7PL019
5299	DCUT(I)=SK1(I)	UC7PL020
5300	DI=DI*(I)+F1	UC7PL021
5301	10 CONTINUE	UC7PL022
5302	RETURN	UC7PL023
5303	END	UC7PL024

SUBROUTINE TARGET

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
TARGET	LTI-2	501
TGTNCL	LTI-2S	502

2. PURPOSE:

This subroutine simulates a target represented by a set of discrete scattering centers.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
F $\emptyset$	R	F	RF Center Frequency
FEXT	R	F	Simulation bandwidth
FI	R	F	Frequency increment of the output transfer function
HTGT	O	F	Height of Target
RTGT $\emptyset$	O	F	Range to target (R in Figure 1) (Used to compute target return variation as a function of range and altitude)
ANGTGT	O	F	Angle from the radar reference line to the target (θ in Figure 1)
TOR ₁ NT	O	F	Angle from Reference line #1 to the target coordinate system reference line ($\emptyset$ in Figure 1). Reference line #1 is parallel to the radar coordinate system reference line and passes through the center of target rotation.

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NSCAT	R	I	Number of scatterers
TSCAT	R	F	Array containing the location and radar cross section of each scatterer. TSCAT (1,k) is the radar cross section of the kth scatterer (σ_k in Figure 1). TSCAT(2,k) is the range coordinate of the kth scatterer in the target coordinate system (r_k in Figure 1). TSCAT(3,k) is the angle coordinate of the kth scatterer in the target coordinate system (β_k in Figure 1).
R000	R	F	Range of target in the simulation (used to compute time delay only)
TGTVEL	0	F	Target radial velocity
TIME	0	F	Elapsed time since beginning of simulation

4. CALLING SEQUENCES:

CALL TARGET (X,Y)

Where: X contains the Output Waveform - R
Y contains the Output Waveform - I

CALL TGTNCL (X,Y)

Where: X contains the Input Waveform - R
Y contains the Input Waveform - I
X contains the Output Waveform - R
Y contains the Output Waveform - I

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The discrete scattering center representation used herein is obtained by imaging coherent short pulse measurements made on the target or a model of the target. The imaging procedure is described in Appendix H of this report.

b. The target representation implemented herein includes the pulse to pulse phase shift due to target motion with respect to the radar system. However, the intrapulse phase shift is not included in this model.

c. Flow Chart: Page 9-109

d. Cross Reference Table: Page 9-222

6. THEORY OF OPERATION

The discrete scatterer target scintillation model is based on the premise that a target can be represented as an ensemble of scattering centers. These scattering centers are assumed to be sufficiently small that they can be represented by impulses. Therefore, this model can be described by the following expression:

$$g(t) = \sum_{n=1}^N \sqrt{\sigma_n} \delta(t - t_n)$$

where σ_n is the radar cross section of the nth scatterer
 t_n is the time delay of the nth scatterer
 $\delta(\)$ is the Dirac delta function.

The Fourier transform of the above expression is given by the following equation:

$$G(f) = \int_{-\infty}^{\infty} \sum_{n=1}^N \sqrt{\sigma_n} \delta(t - t_n) e^{-j2\pi ft} dt$$

$$G(f) = \sum_{n=1}^N \sqrt{\sigma_n} e^{-j2\pi ft_n}$$

In the target model the following equations were used in evaluating the target response in the frequency domain.

$$X(K) = G_{\epsilon} (ELANG) * G_{\psi}(ANGTGT) * \sum_{J=1}^{NSCAT} \sqrt{\sigma_J} \cos 2\pi K(FI+FSTRT)t_n$$

$$Y(K) = G_{\epsilon} (ELANG) * G_{\psi}(ANGTGT) * \sum_{J=1}^{NSCAT} \sqrt{\sigma_J} \sin 2\pi K(FI+FSTRT)t_n$$

where:

K is the frequency index
 FI is the frequency increment
 FSTRT is the starting frequency
 FSTRT = FO - FEXT/2

G_{ψ} is the antenna azimuth gain function
 G_{ϵ} is the antenna elevation gain function

$$ELANG = \sin^{-1} (HTGT/RTGT)$$

t_n is the time displacement of the scatterer

$$t_n = \frac{1}{0.149896} RTGT + TSCAT(2,J) * \cos(\theta - \phi - TSCAT(3,J))$$

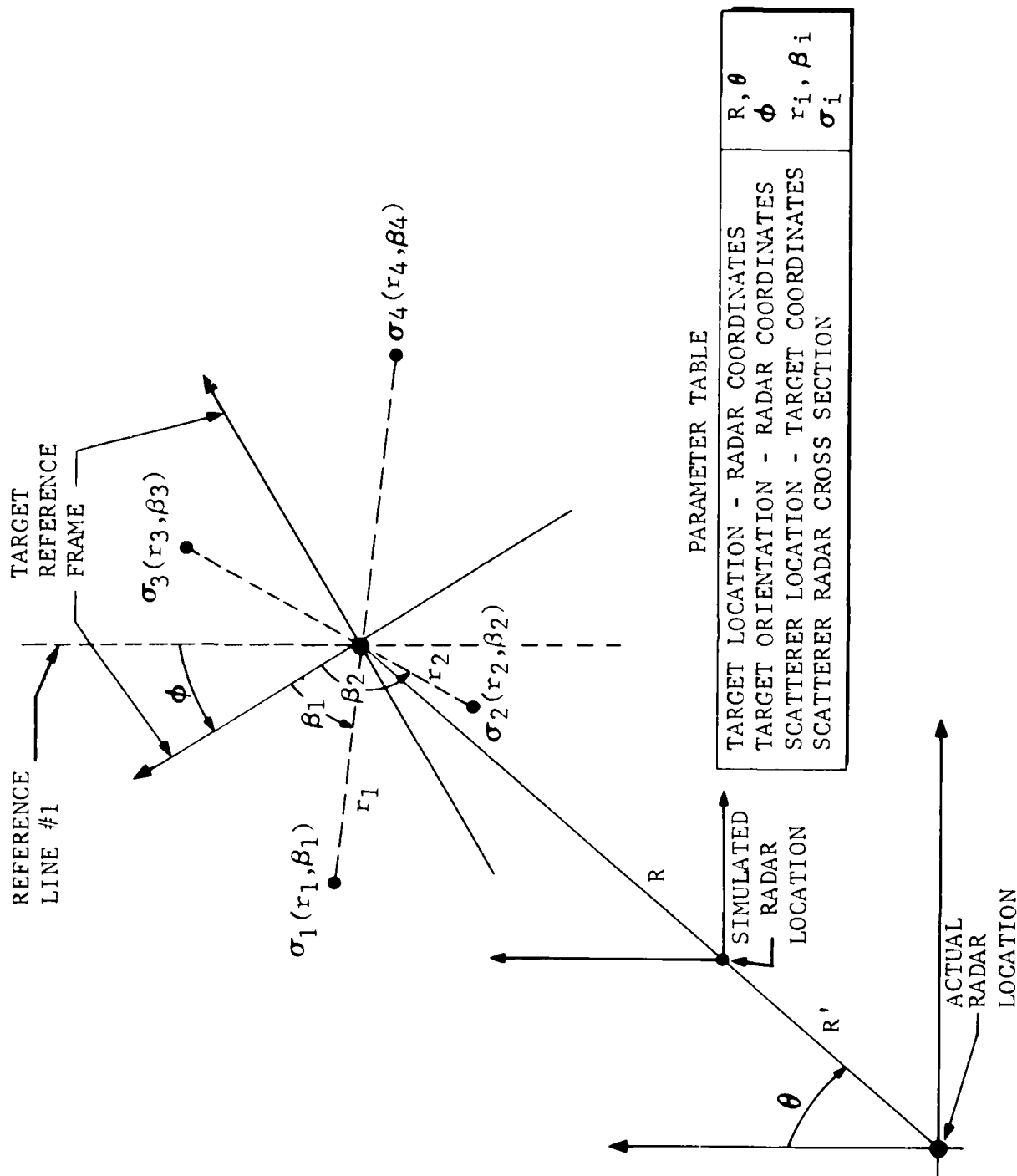


Figure TARGET-1 Target Model Geometry

4-12

```

2072 SUBROUTINE TARGET(X,Y)
2073 COMMON/DEFIZ BK1(200),TSCAT(5,100)
2074 EQUIVALENCE ( BK1( 3), F0      ),( BK1( 4), FEXT      ),
2075 *           ( BK1(11), F1      ),
2076 *           ( BK1(21), IDMY      ),( BK1(100), RIGT      ),
2077 *           ( BK1(107), RTGTO      ),( BK1(108), ANGTGT      ),
2078 *           ( BK1(109), TORINT      ),( BK1(110), SCOC      ),
2079 *           ( BK1(111), RSCAT      ),( BK1(112), TOLVEL      )
2080 * ,           ( BK1( 10), TIME      )
2081 DIMENSION X(1) , Y(1)
2082 DATA N190,N195,N196,N197,-1,-1,-1,0/
2083 DATA P12/0.25,0.1055/
2084 C
2085 IOUN=1
2086 GO TO 10
2087 C
2088 ENTRY TORLOC(X,Y)
2089 IOUN=0
2090 C
2091 IC CONTINUE
2092 GRTO = TRIA(FEXT / F1 )
2093 GRIS = GRTO/2
2094 KRIS = GRIS*2
2095 IF (X(1)>0.0) GO TO 40
2096 C
2097 GO TO 30,GRTO
2098 X(0)=0.
2099 Y(0)=0.

```

4-127

007170

AFCF LISTING

AIRFLOW CHART SET - FWD/SOL - FADSIM

007170

CONTINUED

4-127a

```

2900      GO CONTINUE
2901      C
2902      GO CONTINUE
2903      ACF=ANG10*(1-FRNT)
2904      DE=1.0
2905      FVE=FCO0*TOVEL*TIME
2906      FTO=FTO0*TOVEL*TIME
2907      ACF=TOVEL*(20.0)*C*FEGAIN(ARSD*(HTG/ZEIGT)*FZL*G*VZ)
2908      DE=ZOWIN(ARW*VZ)*DE
2909      C
2910      C=COS(PI*PI/180.0)
2911      F1=FEF*(1+ISQAT(0.5))*COS((A1-ISQAT(0.5))*NO.01/50.0)
2912      F1=ZOWIN*F1
2913      F1=FEF*ISQAT(1.0)*C
2914      F1=FEF*F1
2915      F1=FEF*F1
2916      F1=FEF*F1
2917      F1=FEF*F1
2918      F1=FEF*F1
2919      GO CONTINUE
2920      F1=FEF*F1*F1*F1*F1*F1
2921      F1=FEF*F1*F1*F1*F1*F1
2922      F1=FEF*F1*F1*F1*F1*F1
2923      F1=FEF*F1*F1*F1*F1*F1
2924      F1=FEF*F1*F1*F1*F1*F1
2925      F1=FEF*F1*F1*F1*F1*F1
2926      F1=FEF*F1*F1*F1*F1*F1
2927      F1=FEF*F1*F1*F1*F1*F1
2928      F1=FEF*F1*F1*F1*F1*F1
2929      F1=FEF*F1*F1*F1*F1*F1
2930      F1=FEF*F1*F1*F1*F1*F1
2931      F1=FEF*F1*F1*F1*F1*F1
2932      F1=FEF*F1*F1*F1*F1*F1
2933      F1=FEF*F1*F1*F1*F1*F1
2934      F1=FEF*F1*F1*F1*F1*F1
2935      F1=FEF*F1*F1*F1*F1*F1
2936      F1=FEF*F1*F1*F1*F1*F1
2937      F1=FEF*F1*F1*F1*F1*F1
2938      F1=FEF*F1*F1*F1*F1*F1
2939      F1=FEF*F1*F1*F1*F1*F1
2940      F1=FEF*F1*F1*F1*F1*F1
2941      F1=FEF*F1*F1*F1*F1*F1
2942      F1=FEF*F1*F1*F1*F1*F1
2943      F1=FEF*F1*F1*F1*F1*F1
2944      F1=FEF*F1*F1*F1*F1*F1
2945      F1=FEF*F1*F1*F1*F1*F1
2946      F1=FEF*F1*F1*F1*F1*F1
2947      F1=FEF*F1*F1*F1*F1*F1
2948      F1=FEF*F1*F1*F1*F1*F1
2949      F1=FEF*F1*F1*F1*F1*F1
2950      F1=FEF*F1*F1*F1*F1*F1
2951      F1=FEF*F1*F1*F1*F1*F1
2952      F1=FEF*F1*F1*F1*F1*F1
2953      F1=FEF*F1*F1*F1*F1*F1
2954      F1=FEF*F1*F1*F1*F1*F1
2955      F1=FEF*F1*F1*F1*F1*F1
2956      F1=FEF*F1*F1*F1*F1*F1
2957      F1=FEF*F1*F1*F1*F1*F1
2958      F1=FEF*F1*F1*F1*F1*F1
2959      F1=FEF*F1*F1*F1*F1*F1
2960      F1=FEF*F1*F1*F1*F1*F1
2961      F1=FEF*F1*F1*F1*F1*F1
2962      F1=FEF*F1*F1*F1*F1*F1
2963      F1=FEF*F1*F1*F1*F1*F1
2964      F1=FEF*F1*F1*F1*F1*F1
2965      F1=FEF*F1*F1*F1*F1*F1
2966      F1=FEF*F1*F1*F1*F1*F1
2967      F1=FEF*F1*F1*F1*F1*F1
2968      F1=FEF*F1*F1*F1*F1*F1
2969      F1=FEF*F1*F1*F1*F1*F1
2970      F1=FEF*F1*F1*F1*F1*F1
2971      F1=FEF*F1*F1*F1*F1*F1
2972      F1=FEF*F1*F1*F1*F1*F1
2973      F1=FEF*F1*F1*F1*F1*F1
2974      F1=FEF*F1*F1*F1*F1*F1
2975      F1=FEF*F1*F1*F1*F1*F1
2976      F1=FEF*F1*F1*F1*F1*F1
2977      F1=FEF*F1*F1*F1*F1*F1
2978      F1=FEF*F1*F1*F1*F1*F1
2979      F1=FEF*F1*F1*F1*F1*F1
2980      F1=FEF*F1*F1*F1*F1*F1
2981      F1=FEF*F1*F1*F1*F1*F1
2982      F1=FEF*F1*F1*F1*F1*F1
2983      F1=FEF*F1*F1*F1*F1*F1
2984      F1=FEF*F1*F1*F1*F1*F1
2985      F1=FEF*F1*F1*F1*F1*F1
2986      F1=FEF*F1*F1*F1*F1*F1
2987      F1=FEF*F1*F1*F1*F1*F1
2988      F1=FEF*F1*F1*F1*F1*F1
2989      F1=FEF*F1*F1*F1*F1*F1
2990      F1=FEF*F1*F1*F1*F1*F1
2991      F1=FEF*F1*F1*F1*F1*F1
2992      F1=FEF*F1*F1*F1*F1*F1
2993      F1=FEF*F1*F1*F1*F1*F1
2994      F1=FEF*F1*F1*F1*F1*F1
2995      F1=FEF*F1*F1*F1*F1*F1
2996      F1=FEF*F1*F1*F1*F1*F1
2997      F1=FEF*F1*F1*F1*F1*F1
2998      F1=FEF*F1*F1*F1*F1*F1
2999      F1=FEF*F1*F1*F1*F1*F1
3000      F1=FEF*F1*F1*F1*F1*F1

```

4-128

```

2429      YIN)=YIN)+SH
2430      TEMPUS
2431      CUSUS*ACCEL-IN*ASDEL
2432      DIFSN*ACCEL+TEMP*ASDEL
2433      KEN*NEEL
2434
2435      GO CONTINUE
2436
2437      C
2438      IF (IPAGE-ONE) GO TO 200
2439      IF (USE)
2440      KEN*FIS
2441      KULL=-1
2442      DIFAC=DIFFAS+ULLPH
2443      ULLPH=-ULLPH
2444      GO TO 71
2445
2446      GO CONTINUE
2447
2448      C
2449      X(195) = ULLPH*FIS
2450      X(194) = -FIS*FLOAT(FIS/2)
2451      X(195) = 1
2452      Y(195) = X(195)
2453      Y(194) = X(194)
2454      Y(193) = X(193)
2455
2456      C
2457      C
2458      C
2459

```

SUBROUTINE WEITRE

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
WEITRE	LTI-1 or LTV-4	221
WEITCP	LTI-1 or LTV-4	222
WEITMP	LTI-1 or LTV-4	223

2. PURPOSE:

This subroutine is used to represent (1) a linear time variant device in terms of its measured transfer functions or (2) a linear time varying device in terms of its measured gain (complex) as a function of time.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NPWT	R	I	Number of points used to specify the weighting function
WT	R	F	Array containing the specified weighting function. The specification for the Jth sample is the following:

WEITRE or WEITCP

WT (1,J) = Real component

WT (2,J) = Time

WT (3,J) = Imaginary component
(WEITCP only)

WEITMP

WT (1,J) = Gain

WT (2,J) = Time

WT (3,J) = Phase angle

ORIG O F Amount the independent variables
of the weighting function is to
be shifted.

4. CALLING SEQUENCES:

Real Weighting Function

CALL WEITRE (X, Y, \$mmmm)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

Complex Rectangular Weighting Function

CALL WEITCP (X, Y, \$mmmm)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

Complex Polar Weighting Function

CALL WEITMP (X, Y, \$mmmm)

Where: X contains the Input Waveform - M
 Y contains the Input Waveform - P
 X contains the Output Waveform - M
 Y contains the Output Waveform - P

\$mmmm is the number of the statement to which
control is transferred if an error is detected.
in the input data.

5. RESTRICTIONS, REQUIREMENTS, AND MISCELLANEOUS DATA

- a. The input weighting function independent variable must be monotonically increasing, except that two adjacent points may be specified with the same value. In this case, if the input value to be weighted falls exactly on that point, the first dependent value specified will be used.
- b. If the extent of the weighting function does not include any part of the input data, the subroutine will perform a nonstandard exit.

- c. If the extent of the weighting function contains only a portion of the input data, all other input points are set to zero.
- d. The subroutine performs a linear interpolation between specified points.
- e. Flow Chart: Page 9-57
- f. Cross Reference Table: Page 9-215

6. THEORY OF OPERATION

The basic mechanization equations for each entry point are as follows:

WEITCP:

$$\begin{aligned} X(J) &= X(J) * XWT - Y(J) * YWT \\ Y(J) &= X(J) * YWT + Y(J) * XWT \end{aligned}$$

A table lookup/linear interpolation scheme is used to determine the complex weight, $XWT + j YWT$. The value of the independent variable used in computing the weight for the Jth sample is given by the following expression:

$$VIV = XORIG + J \cdot DEL - ORIG$$

where: XORIG = Independent variable for first element of the input arrays.

DEL = Independent variable increment between samples of the input array.

WEITRE:

$$X(J) = X(J) * XWT$$

$$Y(J) = Y(J) * XWT$$

The procedure for determination of the weight, XWT, is the same as that described above except YWT is not computed.

WEITMP:

$$X(J) = X(J) * AMPL$$

$$Y(J) = Y(J) + PHAS$$

A table lookup/interpolation scheme is used to determine the complex weight $AMPL * EXP (PHAS)$. The procedure for determining value of the independent variable used in computing the weight is described under WEITCP.

4-133

CART. NO.

CONTENTS

```

1190      220 K=K+1
1195      DT = WT(2,K+1) - WT(2,K)
1200      IF (DT .EQ. 0.0) GO TO 220
1205      SCLPE1 = (WT(1,K+1)-WT(1,K)) /DT
1210      SCLPE3 = (WT(3,K+1)-WT(3,K)) /DT
1215      250 IF (K .GE. NPWT) GO TO 275
1220      TIMX = F - WT(1,K)
1225      IF (TIMX ) 275, 270, 280
1230      270 XT= WT(1,K)
1235      IF (ICCN .EQ. 1 .OR. ICUN .EQ. 3) YT = WT(3,K)
1240      GO TO 275
1245      280 IF(1.0-CL*WT(2,K+1)) GO TO 220
1250      XT = TIMX * SCLPE1 + WT(1,K)
1255      IF(ICUN.EQ.3.(K-ICUN.EQ.3)) YT=TIMX*SCLPE3+WT(3,K)
1260      275 CONTINUE
1265      IF (ICCN .EQ. 1) GO TO 285
1270      A = X(J)
1275      X(J) = A * XT + Y(J) * YT
1280      Y(J) = A * YT + Y(J) * XT
1285      GO TO 250
1290      285 X(J) = X(J)*XT
1295      Y(J)=Y(J)+YT
1300      290 T = T +T1
1305      295 CONTINUE
1310      RETURN
1315      300 WRITE (C,1000)

```

WH

WH

```

1320      1000 FORMAT(' THE WEIGHTING ARRAY, WT, IS NOT DEFINED OVER THE COMPLETE
1330      1000 1-DIMENSIONAL RANGE OF THE OBJECT TO BE USED IN THE ANALYSIS AND
1340      1000  THEREFORE THE RESULTS WILL NOT BE PERFECT. **Z Z **')
1350      2000 FORMAT(' THE WEIGHTING ARRAY, WT, IS NOT PROBABLY PERFECT. THE
1360      2000  ANALYSIS RESULTS WILL NOT BE PERFECT. **Z Z **')

```


SUBROUTINE WVGUID

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
WVGUID	LTI-1	510

2. PURPOSE:

This routine is used to simulate a length of waveguide for the purpose of predicting the effect of nonlinear phase behavior on waveforms.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
RFF	R	F	Center frequency of the electromagnetic wave propagating in the waveguide
CFREQ	R	F	Cutoff frequency of the waveguide
XWLENG	R	F	Length of the waveguide

4. CALLING SEQUENCES:

CALL WVGUID (X,Y)

Where:

X	contains the Input Waveform - R
Y	contains the Input Waveform - I
X	contains the Output Waveform - R
Y	contains the Output Waveform - I

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. References:

Atwater, H. A.; Introduction to Microwave Theory, McGraw-Hill, New York, 162, Chapter 3.

b. Flow Chart: Page 9-116

c. Cross Reference Table: Page 9-223

6. THEORY OF OPERATION

A hollow rectangular perfectly conducting waveguide excited in only one of its modes has a propagation function which is described by the following expression:

$$\beta_g(f) = \sqrt{k^2 - k_c^2}$$

where: $k = 2\pi f \sqrt{\mu_0 \epsilon_0}$

$$k_c = 2\pi f_c \sqrt{\mu_0 \epsilon_0}$$

$$f_c = \text{cutoff frequency}$$

Substituting $\frac{1}{c^2} = \mu_0 \epsilon_0$ the following expression is obtained:

$$\beta_g(f) = \frac{2\pi}{c} \sqrt{f^2 - f_c^2}$$

Therefore the transfer function for a length of waveguide considering only the phase term is given by the following expressions:

$$\begin{aligned} G(f) &= e^{-j \frac{2\pi}{c} z \sqrt{f^2 - f_c^2}} ; f_c < f \\ &= e^{\frac{2\pi}{c} z \sqrt{f_c^2 - f^2}} ; f \leq f_c \end{aligned}$$

where z is the length of the waveguide.

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NO

CONTENTS

4-137

```

3074      SCOROTIME=WCOTL(X,Y)
3075      COMMON/BK1/ VAR(500)
3076      DIMENSION X(1),Y(1)
3077      EQUIVALENCE (VAR(140), (FFED), (VAR(147), XWLENG))
3078      EQUIVALENCE (VAR( 3), FFEU)
3079      DATA N193,N194,N195,-1,-2,-17,CONST/3.52737,PI/6.283185/
3080      * (UNITS: (FFED(MHZ), XWLENG(METERS), CONST(NANOS/PER))
3081      CF2=CFREQ*CFEQU
3082      I1=CONST*XWLENG
3083      I2= I1*PI
3084      NPIS=DOUBLE(X(N193))
3085      FFEU=X(N194)+FFED
3086      CCEU=X(N195)
3087      DO 100 J=1,NPIS
3088      F2=FFEU*FREQW
3089      IF (F2-FF2) CCEU=100
3090      THEIA=I1*SQRT(F2-CF2)
3091      THEIA= (THEIA+FLOAT(FFIX(THEIA)))* PI2
3092      CCEUS(THEIA)
3093      SE=SIGN(THEIA)
3094      CCEU=100
3095      C=CCEU*(-I2*SQRT(CF2-F2))
3096      SE=C/C
3097      TIME=X(J)
3098      X(J)=X(J)+C-Y(J)*S
3099      Y(J)=TIME+S*Y(J)*C
3100      FREQ=FREQ+FOLEF
3101      100 CONTINUE
3102      RETURN

```

SECTION 5

CONNECTION MODULES

This section includes all those modules that bind the stimulus/transfer function modules together in order to form the core of the simulation. The functions performed by these modules include array addition or array multiplication (CONV) and conversion between time and frequency domain representations (ZFFT).

SUBROUTINE CONV

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
CONV	Connection	204
CONVMP	Connection	205
DIVA	Connection	206
ADDA	Connection	207,239

2. PURPOSE:

This subroutine serves to connect simulation modules together by performing array addition, multiplication and division.

3. INPUT PARAMETERS:

None

4. CALLING SEQUENCES:

Complex Rectangular Array Multiplication

CALL CONV (XI, XQ, FI, FQ, \$mmmm)

Where: XI contains the Input Waveform - R Complex
 XQ contains the Input Waveform - I Array #1
 FI contains the Input Waveform - R Complex
 FQ contains the Input Waveform - I Array #2
 XI contains the Output Waveform - R
 XQ contains the Output Waveform - I

mmmm is the return address for a non-standard
return

$$\begin{aligned} \text{XI}(\text{J}) &= \text{XI}(\text{J}) * \text{FI}(\text{J}) - \text{XQ}(\text{J}) * \text{FQ}(\text{J}) \\ \text{XQ}(\text{J}) &= \text{XI}(\text{J}) * \text{FQ}(\text{J}) + \text{XQ}(\text{J}) * \text{FI}(\text{J}) \end{aligned}$$

Complex Polar Array Multiplication

CALL CONVMP (XI, XQ, FI, FQ, \$mmmm)

Where: XI contains the Input Waveform - M Complex
 XQ contains the Input Waveform - P Array #1
 FI contains the Input Waveform - M Complex
 FQ contains the Input Waveform - P Array #2
 XI contains the Output Waveform - M
 XQ contains the Output Waveform - P

mmmm is the return address for a non-standard return.

$$\begin{aligned} \text{XI}(\text{J}) &= \text{XI}(\text{J}) * \text{FI}(\text{J}) \\ \text{XQ}(\text{J}) &= \text{XQ}(\text{J}) + \text{FQ}(\text{J}) \end{aligned}$$

Complex Polar Array Division

CALL DIVA (XI, XQ, FI, FQ, \$mmmm)

Where: XI contains the Input Waveform - M Complex
 XQ contains the Input Waveform - P Array #1
 FI contains the Input Waveform - M Complex
 FQ contains the Input Waveform - P Array #2
 XI contains the Output Waveform - M
 XQ contains the Output Waveform - P

mmmm is the return address for a non-standard return

$$\begin{aligned} \text{XI}(\text{J}) &= \text{XI}(\text{J}) / \text{FI}(\text{J}) \\ \text{XQ}(\text{J}) &= \text{XQ}(\text{J}) - \text{FQ}(\text{J}) \end{aligned}$$

Array Addition

CALL ADDA (XI, FI, \$mmmm)

Where: XI contains the Input Waveform
 FI contains the Input Waveform
 XI contains the Output Waveform

mmmm is the return address for a non-standard return

$$\text{XI}(\text{J}) = \text{XI}(\text{J}) + \text{FI}(\text{J})$$

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The difference in the independent variable increment of the input arrays must be less than or equal to 10^{-4} .
- b. The output is set to zero for any cases where the independent variables of the two arrays do not overlap.
- c. When polar operations are performed the phase angle is measured in degrees.
- d. If the arrays do not overlap, or if they must be shifted more than 4097 increments, the problem will abort and the non-standard return will be used.
- e. Flow Chart: Page 9-60
- f. Cross Reference Table: Page 9-216.

6. THEORY OF OPERATION

Depending upon the entry point through which the subroutine is called, a control flag is set to an integer number 1 to 4. This number indicates to the subroutine the type of operation which is to be performed, either rectangular multiplication, polar multiplication, polar division, or addition. After the flag is set, the two arrays are examined (beginning at statement number 1000) to verify that the values of the increments of the two arrays are within $1/10000$ of each other. The arrays are inspected to verify that they overlap, at least in part, and one array is shifted, if required, to align the two origins. If the increments are not within $1/10000$ of each other, or if the arrays do not overlap, or if the origins of the two arrays are not within 4097 increments of each other, the problem is aborted. Otherwise, the proper arithmetic manipulation is performed, according to the value of the control flag and the results are placed in the X array(s). The F array(s) remain unmodified after exit from the subroutine. For the divide operation, the X complex array is the dividend and the F complex array is the divisor.

02/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWO/SCL RADSIM

S-5a

CARD NO.	****	CONTENTS
1056		IXI=1+FIX((CRIGN-XI(N194))/ XI(N195) + 0.40)
1057		IFI=1+FIX((CRIGN-FI(N194))/ XI(N195) + 0.40)
1058		IF(IXI+NPTS.GT.NXI+1) NPTS=NPTS-1
1059		IF(IFI+NPTS.GT.NFI+1) NPTS=NPTS-1
1060		WRITE(6,50) NPTS,IXI,IFI,NXI,NFI,CRIGN,ENDATA,XI(N194),FI(N194)
1061	50	FORMAT(1H ,511Z,4E14.6)
1062		XI(N195)=BODL(NPTS)
1063		XI(N194)=CRIGN
1064		IF(IGCN.EQ.4) GO TO 700
1065	701	XI(N195) = XI(N193)
1066		XI(N194) = XI(N194)
1067	701	CONTINUE
1068	C	*****
1069		DO 20 J=1,NPTS
1070		GO TO (101,102,103,104),IGCN
1071	C	
1072	101	CONTINUE
1073		A=A1(IXI)
1074		XI(J)= A*FI(IFI) + XC(IXI)*FC(IFI)
1075		XC(J)= A*FC(IFI) + XC(IXI)*FI(IFI)
1076		GO TO 25
1077	C	
1078	102	CONTINUE
1079		XI(J)= XI(IXI)*FI(IFI)
1080		XC(J)= XC(IXI)+FC(IFI)
1081		GO TO 15
1082	C	
1083	103	CONTINUE
1084		XI(J)= XI(IXI)/FI(IFI)

5-6

```

1000      X2(IJ)= X2(IXI)-F2(IF1)
1001      GO TO 25
1002      C
1003      104 CONTINUE
1004      X1(IJ)= X1(IXI)+F1(IF1)
1005      C
1006      20 IXI=IXI+1
1007      IF1=IF1+4
1008      20 CONTINUE
1009      WRITE(6,50) IXI,IF1
1010      C
1011      RETURN
1012      200 WRITE(6,201)
1013      201 FORMAT(' INDEPENDENT VARIABLE INCREMENTS DO NOT MATCH ')
1014      202 RETURN 1
1015      END

```

SUBROUTINE ZFFT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
ZFFT	Connection	202
ZIFFT	Connection	203
IFT	Connection	Not User Referenced

2. PURPOSE:

This subroutine performs the finite discrete Fourier Transform (ZFFT) and inverse Fourier Transform (ZIFFT or IFT) of a sequence of input data samples. The mechanization is based on the Fast Fourier Transform (FFT) algorithm developed by Langdon and Sande from the approach of J. W. Tukey and J. Cooley.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
N2	R	I	Power of 2 which determines the total number of prints (NTHPOW) transformed by the FFT.
FSHIFT	R	F	Frequency shift to be applied to frequency domain representation of the waveform.
ICPLXI	O	I	Control flag which indicates the nature of the inverse transform (ZIFFT) input data = 1 complex waveform $Z = X + jY$ = 0 real waveform $Z = X$
ICPLXF	O	I	Control flag which indicates the nature of the forward transform (ZFFT) input data = 1 complex waveform $Z = X + jY$ = 0 real waveform $Z = X$
SIMBW	R	F	Width of output spectrum when forward transform (ZFFT) is performed

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
INORM	R	I	Normalization flag
			$= 0; \quad \frac{\text{ZFFT}}{f_N} = \text{TI} \quad \quad \frac{\text{ZIFFT}}{f_N} = \text{FI}$
			$= 1; \quad f_N = 1.0/\text{NP} \quad \quad f_N = \text{FI}$
			$= 2; \quad f_N = \text{TI} \quad \quad f_N = 1.0/(2**\text{N2})$
			$= 3; \quad f_N = 1.0/\text{NP} \quad \quad f_N = 1.0/(2**\text{N2})$
			$= 4; \quad f_N = 1.0 \quad \quad f_N = 1.0$

4. CALLING SEQUENCES:

Fourier Transform

CALL ZFFT (X,Y)

Where:

X contains the Input Waveform - R

Y contains the Input Waveform - I

X contains the Output Waveform - R

Y contains the Output Waveform - I

Increase Fourier Transform

CALL ZIFFT (X, Y)

Where:

X contains the Input Waveform - R

Y contains the Input Waveform - I

X contains the Output Waveform - R

Y contains the Output Waveform - I

Inverse FFT only

CALL IFT (X, Y)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 X contains the Output Waveform - R
 Y contains the Output Waveform - I

This entry point is used in computing antenna patterns and is called by ANTPAT module only.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The maximum value of N2 is 13 which gives 8192 transformed samples.
- b. The number of points transformed by the FFT algorithm is 2 raised to an integer power. If the number of input samples is not equal to a power of 2, then zeros are used to fill in the array elements not defined by the input data; i.e., zeros are placed in the input array locations from NP + 1 to 2**N2.
- c. When the Forward Transform is executed the output spectrum is shifted by FSHIFT and has a width of SIMBW: i.e., the output spectrum extends from FSHIFT - SIMBW/2 to FSHIFT + SIMBW/2.
- d. When the Forward Transform is executed the origin of the independent variable X (N194) is set to the discrete frequency nearest FSHIFT - SIMBW/2 since the FFT output consists of a discrete set of frequency samples. When the Inverse Transform is executed the origin of the independent variable X(N194) is set to zero.
- e. The independent variable increment is set to 1.0/(X(N195)*(2**N2POW)) for both the forward and inverse transform.
- f. When the inverse transform is executed the input spectrum is shifted by FSHIFT before the FFT is executed. The value of FSHIFT when the inverse transform is performed can be different from the value used when the forward transform was performed.

The following relationship must be satisfied

$$-FS/2 \leq FSHIFT \pm SIMBW/2 < FS/2$$

g. Flow Chart: Page 9-125

h. Cross Reference Table: 9-224

6. THEORY OF OPERATION

- a. ZFFT - This entry point connects the time domain representation of a waveform into the corresponding frequency domain representation. The Fourier transform relationship is given by the following expression:

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt$$

The basic mechanization equation used in computing the discrete Fourier transform is the following:

$$S_k = f_N \sum_{i=0}^{NP-1} s_i W^{-k_i}$$

where: k is the frequency index $0 \leq k \leq NTHPOW-1$
 i is the time index $0 \leq i \leq NP-1$
 f_N is the normalization factor
 $W = \exp(j2\pi/NTHPOW)$

In general s_i and S_k are complex arrays.

- b. ZIFFT and IFT - These entry points connect the frequency domain representation of a waveform into the corresponding time domain representation. The inverse Fourier transform relationship is given by the following expression:

$$s(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi ft} df$$

The basic mechanization equation used in computing the inverse discrete Fourier transform is the following:

$$s_i = f_N \sum_{k=0}^{NTHPOW-1} S_k w^{ki} \quad 0 \leq i \leq NTHPOW-1$$

The only difference between ZIFFT and IFT is that IFT performs the inverse FFT only and no normalization is performed. The FFT mechanization utilized in this module evaluates the above expression directly. In order to compute the forward transform using the same inverse FFT the following expression is used.

$$S_k = f_N \left[\sum_{i=0}^{NP-1} s_i^* w^{ki} \right]^*$$

CORRECTING LEFT(X,Y)

5-12

COMMON/BER1/ BK1(500)

DOUBLE PRECISION CIEL,SCCL,CS,SN,CT,PI,SCALE

31

EQUIVALENCE (NZ, BK1(1)), (FENHFT, BK1(25)), (CECXL, BK1(6)),

*, (CEPLAF, BK1(7)), (SIMW, BK1(4)), (IRUN, BK1(9))

DATA 112/0.22310050667007

51

DATA N193,N194,N195,N196/-3,-1,-1,07

EDGE X(1),Y(1),1,11,12,13,14

INTERP PASS, SECECC, L(14)

EQUIVALENCE (J,J1), (N4PCW,J2), (FAS,J3),

*, (X1LTH,J4), (L1NGTH,J5), (SECECC,J6), (ICALL,J7),

*, (IASC,J8), (A1,J12), (C1,L1), (C2,L1), (C3,L1),

*, (C4,L1), (C5,L1), (K1,L6), (K2,L7), (C1,LC1), (C2,LC2),

*, (C3,L1)

EQUIVALENCE (L18,L(1)), (L13,L(2)), (L17,L(3)), (C11,L(4)),

*, (L10,L(5)), (C9,L(6)), (C1,L(7)), (L7,L(8)), (C1,L(9)), (L5,L(10)),

*, (L6,L(11)), (L3,L(12)), (L1,L(13)), (L1,L(14))

IFZC = 1

GO TO 5

CALL LEFT(X,Y)

IFZC = 1

GO TO 5

CALL LEFT(X,Y)

IFZC = 1


```

3277          9 CONTINUE
3278          NTHPOW = 2** N2
3279          NTTL2=NTHPOW/2
3280          NP= BOGL(X(N193))
3281          GO TO(502,501,411),IFLAG
3282          502 R1=X(N195)
3283              R2=1.0/R1
3284              IF(SIMBW.EQ.R2) SIMBW=R2
3285              IF(INCRM.EQ.1.OR.INCRM.EQ.3) R1=1.0/FLUAT(NP)
3286              IF(INCRM.EQ.4) R1=1.0
3287              IF(ICPLAF.EQ.0) GO TO 700
3288              DO 11 J=1,NP
3289                  X(J)= X(J)*R1
3290                  Y(J)=-Y(J)*R1
3291              11 CONTINUE
3292              GO TO 701
3293          700 DO 702 J=1,NP
3294              X(J)= X(J)*R1
3295              Y(J) =0.0
3296          702 CONTINUE
3297          701 CONTINUE
3298              IF(NTHPOW-NP) 32,31,30
3299          30    L1=NP+1
3300              DO 10 J=L1,NTHPOW
3301                  X(J) = 0.0
3302                  Y(J) = 0.0
3303          10 CONTINUE
3304              GO TO 500
3305          32    WRITE(6,33) NP,NTHPOW

```

5-13

260

06/17/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

5-14

CARD NO

CONTENTS

```

3306      33  FORMAT(' NUMBER OF INPUT SAMPLES',I4,' EXCEEDS SPECIFIED TIME',
3307      * ' SPAN=2**N2=',I4)
3308      34  CONTINUE
3309      35  GO TO 360
3310      36  CONTINUE
3311      37  IF (N2.EQ.0.NTHPCW.CN2.EQ.13) GO TO 399
3312      38  N2=N2+1
3313      39  WRITE (6,310) N2
3314      40  FORMAT (' THE SIZE OF THE TRANSFER ARRAY HAS BEEN EXPANDED',
3315      * ' TO 2**',I2)
3316      41  GO TO 39
3317      42  R1=X(N194)*FSHIFT
3318      43  R2=FLCAT(NITL2)*X(N195)
3319      44  NSHIFT=IFIX( (R1-R2) / X(N195) + 0.40 )
3320      45  R1=R1-X(N195)/2.0
3321      46  R2=R2+X(N195)/2.0
3322      47  R3=R2+X(N195)*FLCAT(NSHIFT)
3323      48  WRITE (6,300) NSHIFT,NP,P1,R2,R3,B1,T1,X(N194)
3324      49  FORMAT(1H,2I10,6E15.6)
3325      50  IF (N2.GE.64.AND.R3.LE.T1) GO TO 60
3326      51  IF (R3.LE.0.1) NSHIFT=NSHIFT+1
3327      52  IF (R3.GT.1.1) NSHIFT=NSHIFT-1
3328      53  GO TO 49
3329      54  IF (NSHIFT.LE.0) GO TO 404
3330      55  R2=R2+NSHIFT
3331      56  R3=R3
3332      57  IF (N2.EQ.NTHPCW) GO TO 370
3333      58  IF (N2.EQ.13) GO TO 360
3334      59  N2=N2+1

```

```

3335      WRITE(6,310) N2
3336      GO TO 4
3337 380  WRITE(6,385) L2,NTHPOW
3338      385 FORMAT(' THE NUMBER OF ARRAY ELEMENTS REQUIRED AFTER HETERODYNING',
3339      *15, ' EXCEEDED AVAILABLE STORAGE...ARRAY REDUCED TO',15,
3340      * ' BY DELETING HIGH FREQ TERMS')
3341      L3=NTHPOW-NSHIFT
3342      L4=NTHPOW
3343 370  IF(NSHIFT.LT.0) GOTO 430
3344      DO 400 L1=1,NF
3345      X(L2)= X(L3)
3346      Y(L2)= Y(L3)
3347      L2=L2-1
3348      L3=L3-1
3349      400 CONTINUE
3350      DO 401 L1=1,NSHIFT
3351      X(L1)=0.0
3352      Y(L1)=0.0
3353      401 CONTINUE
3354 420  L2=NF*(NSHIFT+1)
3355      IF(L2.GT.NTHPOW) GOTO 403
3356      DO 402 L1=L2,NTHPOW
3357      X(L1)=0.0
3358      Y(L1)=0.0
3359      402 CONTINUE
3360      GO TO 405
3361 404  IF(N2.LT.15) GOTO 407
3362      N2=N2+1
3363      WRITE(6,310) N2

```

0044879

INPL: LISTING

AUTOFLOW CHART SET - FWL/SCL RADSIM

5-10

6470 100

◆ ◆ ◆ ◆

CONTENTS

◆◆◆

[illegible]

5-17

```

3393      IF (INCRML*EC*4) RS=1.0
3394      LI=NTL2
3395      DO 409 J=1,NTL2
3396      LI=LI+1
3397      R1=X(LI)*RS
3398      R2=Y(LI)*RS
3399      X(LI)=X(J)*RS
3400      Y(LI)=Y(J)*RS
3401      X(J)=R1
3402      Y(J)=R2
3403      409 CONTINUE
3404      411 CONTINUE
3405      X(N143)=EC*CL(NTHPLW)
3406      X(N144)=0.0
3407      500 CONTINUE
3408      X(N145)=1.0/(FLOAT(NTHPLW)*X(N145))
3409      N4FOW = N2 / 2
3410      IF (N4FOW*EW=0) GO TO 3
3411      DO 2 PASS=1,N4FOW
3412      NXLTH=2*(( N2 -2*PASS)
3413      LENGTH=4*NXLTH
3414      SCALE=PI2/DBLE(FLOAT(LENGTH))
3415      COSL=COS(SCALE)
3416      SIDL=SIN(SCALE)
3417      CS=1.000
3418      SN=0.000
3419      DO 2 J=1,NXLTH
3420      CI=CS
3421      SI=SN

```

1050

1120

1160

1151

1152

1153

1154

1180

1190

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RAUSIM

5-18

CARD NO.	****	CONTENTS	****
0422		$U = C_3 * C_5 - SN * SN$	1206
0423		$U = 4 * U * C_3 * SN$	1210
0424		$U = C_1 * C_7 - S_1 * S_2$	1220
0425		$S_2 = C_2 * S_1 + S_2 * C_1$	1230
0426		$U = C_5$	1240
0427		$C_5 = C_5 * C_1 + S_1 * S_2$	1250
0428		$SN = C_1 * S_2 + SN * C_2$	1260
0429		$DO = SELECC = LENGTH, NTHROW, LENGTH$	1416
0430		$J_1 = SELECC = LENGTH + J$	
0431		$J_2 = J_1 * NALTH$	
0432		$J_3 = J_2 * NALTH$	
0433		$J_4 = J_3 * NALTH$	
0434		$F_1 = X(J_1) * X(J_2)$	
0435		$F_2 = X(J_1) * X(J_3)$	
0436		$F_3 = X(J_2) * X(J_4)$	
0437		$F_4 = X(J_3) * X(J_4)$	
0438		$I_1 = Y(J_1) * Y(J_2)$	
0439		$I_2 = Y(J_1) * Y(J_3)$	
0440		$I_3 = Y(J_2) * Y(J_4)$	
0441		$I_4 = Y(J_3) * Y(J_4)$	
0442		$X(J_1) = 11 * F_1$	
0443		$Y(J_1) = 11 * I_1$	
0444		$X(J_2) = (1 * (F_2 - F_4) - 21 * (I_2 + F_3))$	
0445		$Y(J_2) = (1 * (F_2 - F_4) + (1 * (I_2 + F_3))$	
0446		$X(J_3) = (1 * (F_1 - F_3) - 51 * (I_1 - I_2))$	
0447		$Y(J_3) = (1 * (F_1 - F_3) + 51 * (I_1 - I_2))$	
0448		$X(J_4) = (1 * (F_2 + F_4) - 53 * (I_2 - F_3))$	
0449		$Y(J_4) = (1 * (F_2 + F_4) + 53 * (I_2 - F_3))$	
0450		END	

```

3451      5 IF( N2 .EQ.2*N4POW) GO TO 5
3452      DO 4 J=1,NTHPCW,2
3453      K=X(J)+X(J+1)
3454      X(J+1)=X(J)-X(J+1)
3455      X(J) = K
3456      I=Y(J)+Y(J+1)
3457      Y(J+1)=Y(J)-Y(J+1)
3458      Y(J)=I
3459      DO 6 J=1,14
3460      L(J)=1
3461      6 IF(J.LE. N2 ) L(J)=2** ( N2 +1-J)
3462      IJ=1
3463      DO 7 J1=1,L1
3464      DO 7 J2=J1,L2,L1
3465      DO 7 J3=J2,L3,L2
3466      DO 7 J4=J3,L4,L3
3467      DO 7 J5=J4,L5,L4
3468      DO 7 J6=J5,L6,L5
3469      DO 7 J7=J6,L7,L6
3470      DO 7 J8=J7,L8,L7
3471      DO 7 J9=J8,L9,L8
3472      DO 7 J10=J9,L10,L9
3473      DO 7 J11=J10,L11,L10
3474      DO 7 J12=J11,L12,L11
3475      DO 7 J13=J12,L13,L12
3476      DO 7 J14=J13,L14,L13
3477      IF(IJ.GE.51) GO TO 7
3478      K=X(IJ)
3479      X(IJ)=X(J1)

```

03/11/75

INPUT LISTING

AUTOFLOW LPART SET - FWD/SCL KADSIM

5-20

CARD NO

CONTENTS

```

3480      X(J1)=R
3481      I=Y(IJ)
3482      Y(IJ)=Y(J1)
3483      Y(J1)=I
3484      7 IJ=IJ+1
3485      IF (IFLAG.EQ.2) GO TO 15
3486      LI=NTL2
3487      DO 14 J=1,NTL2
3488      L1=L1+1
3489      R1=X(L1)
3490      R2=Y(L1)
3491      X(L1)=X(J)
3492      Y(L1)=Y(J)
3493      X(J)=R1
3494      Y(J)=R2
3495      14 CONTINUE
3496      R2=-X(N195)*FLCAT(NTL2)
3497      X(N194)=R2
3498      IF (IFLAG.EQ.3) GO TO 15
3499      NP=FIX(SIMBW/X(N195))/2
3500      F1=FSHIFT-X(N195)*NP
3501      NSHIFT=FIX(FSHIFT/X(N195))-NP+NTL2
3502      NP=NP*R
3503      B1=R1-X(N195)/2.0
3504      T1=F1+X(N195)/2.0
3505      46 R3=R2+X(N195)*NSHIFT
3506      WRITE(6,300) NSHIFT,NP,R1,R2,R3,B1,T1
3507      IF (R3.GE.B1.AND.R3.LE.T1) GO TO 47
3508      IF (R3.LT.B1) NSHIFT=NSHIFT+1

```



```

3509      IF(NS.GT.11) NSHIFT=NSHIFT-1
3510      GOTO 48
3511 47      IF(NSHIFT.GE.0) GOTO 49
3512      WRITE(6,55)
3513 55      FORMAT(' THE VALUE OF NSHIFT IS NEGATIVE...NSHIFT SET TO 0')
3514      GOTO 52
3515 49      LI=NSHIFT*NP
3516      IF(LI.LE.NTHPW) GOTO 50
3517      WRITE(6,51) NSHIFT,NP
3518 51      FORMAT(' THE SUM OF NSHIFT*,110,* AND 11*,110,* IS GT NTHPW....',
3519      *NSHIFT SET TO 0')
3520 52      NSHIFT=0
3521 50      CONTINUE
3522      X(N1NS)=GUESS(NP)
3523      X(N1NS)=-X(N1NS)*(NP/2)
3524      WRITE(6,500) NSHIFT,NP,EL,END,X(N1NS)
3525 500      FORMAT(' 0-1,NP',
3526      *NSHIFT=NSHIFT+1
3527      *X(I)=X(NSHIFT)
3528      *Y(I)=-Y(NSHIFT)
3529 415      CONTINUE
3530 15      GOTO 16
3531      Y(1193)=X(1193)
3532      Y(1194)=X(1194)
3533      Y(1195)=X(1195)
3534      GOTO 6
3535      STOP

```

SECTION 6

PERIPHERAL MODULES

This section includes all those modules that do not participate in the actual simulation process, but serve as the interface between the simulation and the user. In some cases this is as simple as outputting data to a disc file. In other cases the data is processed to extract some parameter, e.g. energy in a waveform, or to map data into a different representation space, e.g. the generation of a probability histogram. The following modules are included in this group:

CLINT
CUMDIS
ERGYCP
PLOTTR
PLTFMT
PTLIST
RTOPDB
SCANNR

Also included in this group is the module SPCAVG, which is located in Volume III.

The following peripheral modules are related to the bistatic target model and target imaging and are included in Volume I, Part 3:

ERRPRC
(BISTATIC ANTENNA INITIALIZATION)

SUBROUTINE CLINT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
CLINT	PERIP.	302

2. PURPOSE:

This subroutine initializes the clutter model scatterers and parameters.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
LAMBDA	O	F	Wavelength of the transmitted waveform. This parameter is used to compute the doppler shift of weather or chaff clutter.
TCELL	R	F	Range dimension of clutter element.
IDMP	O	I	Output data print flag. IDMP ≠ 0 (a) Scatterer arrays are printed (CLINT) (b) Antenna Azimuth and Elevation arrays are printed (cluttr) IDMP = 1 Clutter impluse response is printed by (cluttr).
NRCS	R	I	Scatterer RCS probability distribution flag.
RWPH	O	F	Scatterer random walk maximum phase shift. Represents the random motion of cluttr from one transmit cycle to the next.
WNDVEL	O	F	Speed of the clutter volume due to wind, meters/second.

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
TI	R	F	Time increment between samples (used to compute number of storage cells required by clutter impulse response).
VELANG	O	F	Relative angle of clutter velocity vector with respect to the radar site.
RNEXT	O	F	Range dimension of the clutter volume in monoseconds.
RN000	O	F	Clutter volume starting range.
AZEXT	O	F	Azimuth dimension of the clutter volume in degrees
AZ000	O	F	Clutter volume starting azimuth angle.
MM	O	I	Number of clutter volume azimuth increments.
ELEXT	O	F	Elevation dimension of the clut- ter volume in degrees.
EL000	O	F	Clutter volume starting elevation angle.
NN	O	I	Number of clutter volume eleva- tion increments.

4. CALLING SEQUENCES:

CALL CLINT (\$NNNN)

WHERE:

NNNN Statement number for abnormal
return

5. RESTRICTION, REQUIREMENTS, MISCELLANEOUS DATA:

- a. The clutter model was split into two subroutines in order that the initialization process would be separate and easily changed to allow modeling of more complex clutter environments.
- b. The total number of scatterers (NSCAT) is limited only by the amount of available disc space and is approximately equal to 160 times the number of blocks of available disc space. The number of scatterers is equal to NN times MM times KK, where $KK = RNEXT/TCELL$. Other than their product; NN, MM and KK have no limitations.
- c. The random number generator function (RRAND) is called from this subroutine.
- d. The clutter scatterer element arrays are stored in a random disc file, FC = 02, the record size is 500 words.
- e. Flow Chart: Page 9-198
- f. Cross Reference Table: Page 9-234

6. THEORY OF OPERATION

This program can be divided into two functional sections and each will be discussed in the following paragraph.

- a. User parameter testing and clutter model parameter calculation. Refer to Figures CLINT - 1(a) and 1(b).

XVLANG The angle between the wind velocity vector and a radial line from the first clutter scatterer element (0,0,0) to the radar.

$$XVLANG = (AZ000 - VELANG)/57.29578$$

DOPFRQ The maximum doppler frequency shift due to the wind.

$DOPGRQ = -2.0 * WNDVEL/LAMBDA$
If LAMBDA is zero this calculation is not performed and DOPFRQ is set to zero.

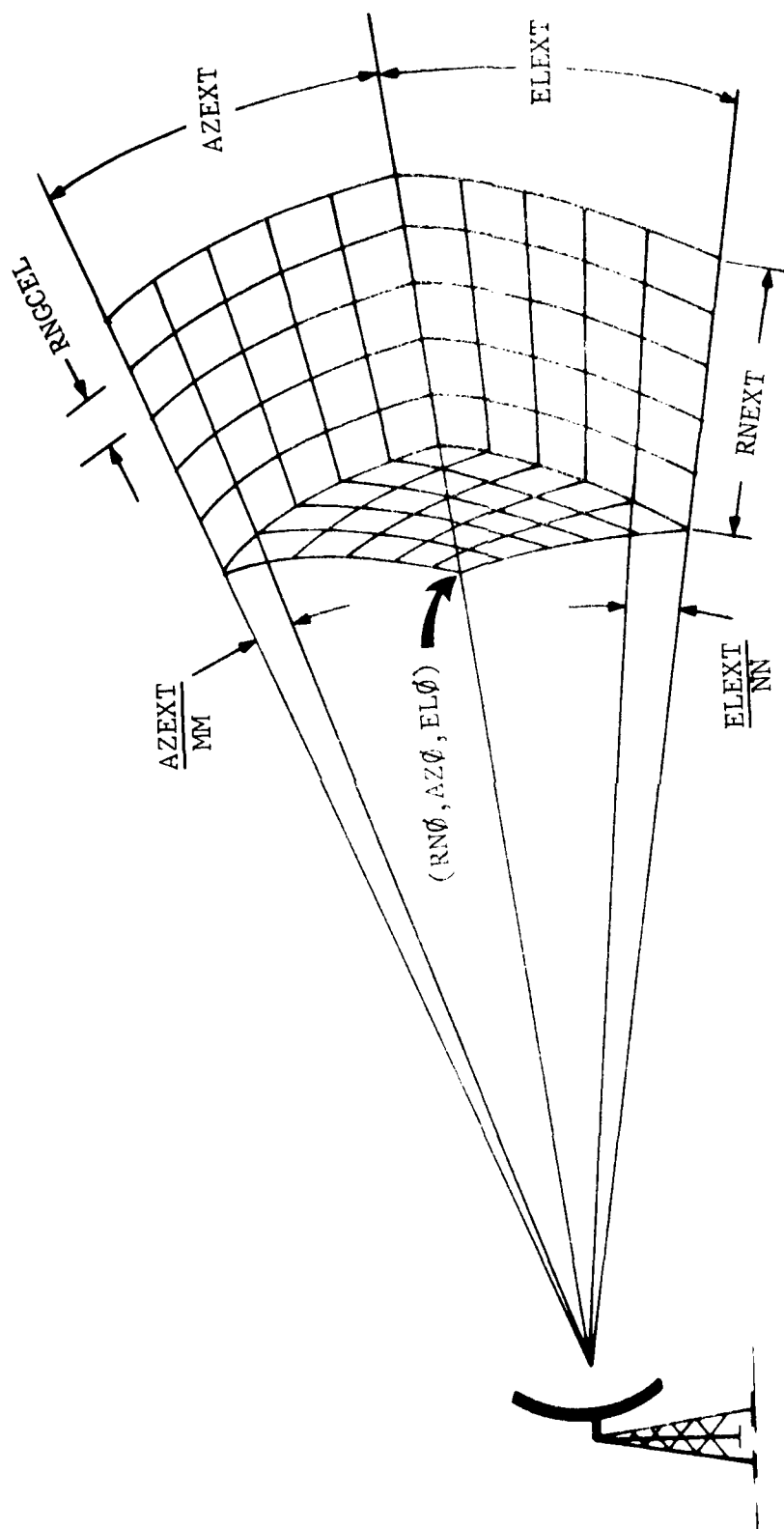


Figure CLINT-1a CLUTTER MODEL GEOMETRY

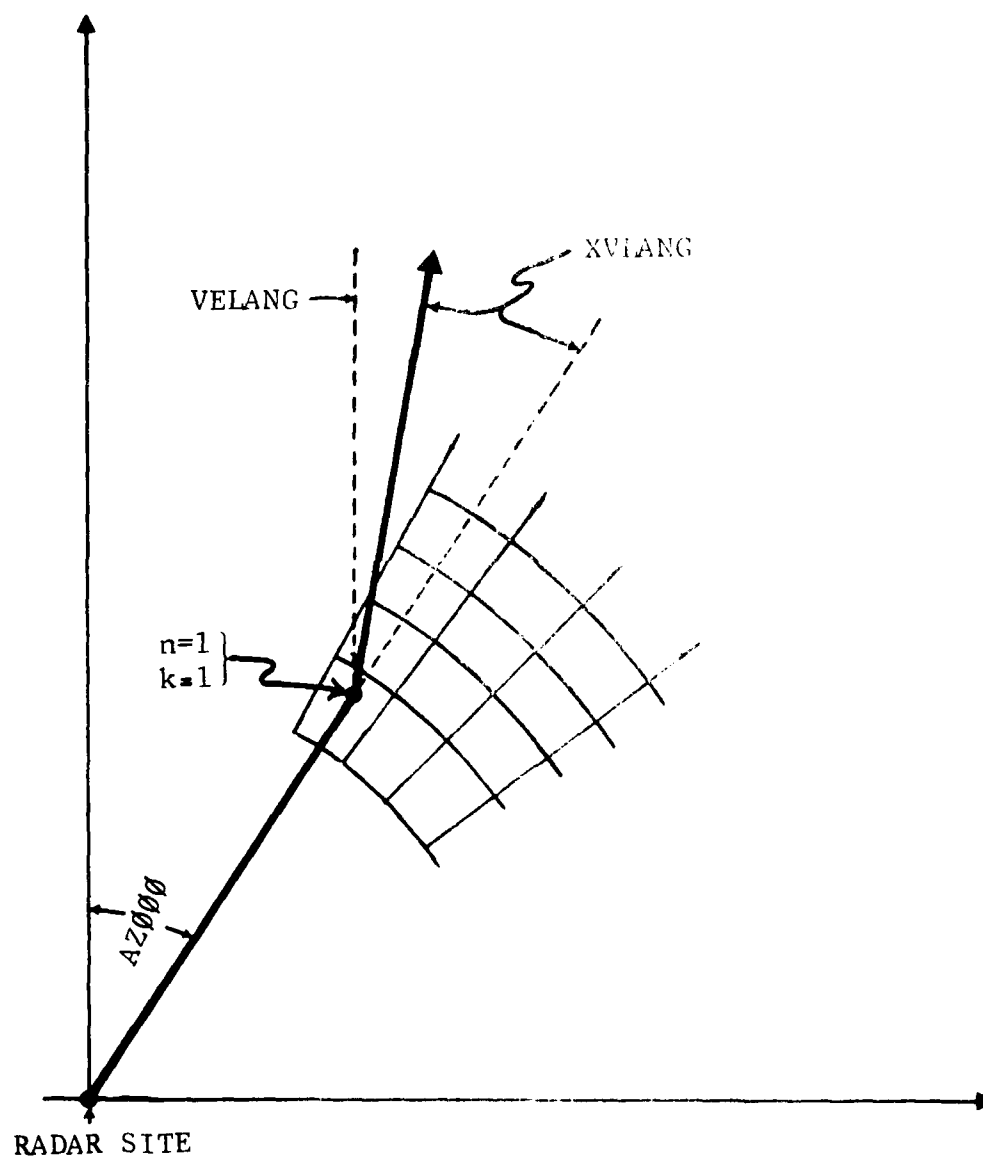


Figure CLINT-1b CLUTTER MODEL GEOMETRY
(PLANE VIEW)

MM If this parameter was not specified it is set to 1.

NN If this parameter was not specified it is set to 1.

KCELL The total number of storage cells required to represent the clutter impulse response. If this parameter exceeds 4096 (half the available storage) then RNEXT is automatically reduced to make $KCELL < 4096$.

$KCELL = RNEXT/TI$

KK Number of clutter volume range increments.

$KK = RNEXT/TCELL$

NBLKS The number of random disc records required to store the clutter scatterer. This parameter is arbitrarily limited to 200 and can be increased providing the control cards for DSRN 02 are changed accordingly.

DELAZ Azimuth increment between clutter scatterers.

$DELAZ = AZEXT/MM$

DELEL Elevation increment between clutter scatterers.

$DELEL = ELEXT/NN$

ICFLG This parameter is set to 1 if the clutter model has been successfully initialized. This is tested before CLUTTR can be executed.

MODE This parameter is used to indicate if stationary (MODES) or time varying (MODE=2) clutter is to be modeled. When stationary clutter is modeled certain simplifications can be made in the processing performed in the clutter model routine CLUTTR.

- b. Each clutter scatter element is represented by a radar cross section value and a phase angle. These two parameters are converted to rectangular representation and stored in the arrays CLUX and CLUY.

CLUX(J) = A*RRAND(8)

CLUY(J) = A*DUM

where $A = \sqrt{RCS(J)} = \sqrt{RRAND(NRCS)}$

RRAND(8) = COS(θ)

DUM = SIN(θ)

θ is a sample of a random uniform distribution $300.0 < \theta \leq 0.0$

5122 SUBROUTINE CLINT(*)
 5123 COMMON/BLK1/ CLUX(250),CLUY(250)
 5124 COMMON/BLK2/ BK2(500)
 5125 COMMON/BLK3/ IDMF(8),DUM
 5126 DIMENSION CSCAT(500)
 5127 EQUIVALENCE (CLUX(1),CSCAT(1))
 5128 EQUIVALENCE (BK2(13), LAMBDA) , (BK2(14), TCELL) ,
 5129 * (BK2(21), IDMP) , (BK2(46), NRCS) ,
 5130 * (BK2(47), NRWPH) , (BK2(48), RWPH) ,
 5131 * (BK2(49), WNOVEL) , (BK2(50), VELANG) ,
 5132 * (BK2(51), RNEXT) , (BK2(52), RNOOO) ,

6-9

C

```

5133      *          (BK2( 53), AZEXT          ),(BK2( 54), AZU00          ),
5134      *          (BK2( 55), MM              ),(BK2( 56), ELEXT          ),
5135      *          (BK2( 57), EL000           ),(BK2( 58), NN              ),
5136      *          (BK2(120), KK              ),(BK2(121), NELKS          ),
5137      *          (BK2(122), MODE            ),(BK2(123), ULLAZ          ),
5138      *          (BK2(124), DELFL          ),(BK2(125), XVLANG          ),
5139      *          (BK2(126), ICFLG          ),(BK2(127), DCPFRG          ),
5140      *          (BK2(128), KCELL           ),(BK2( 12), T1              ),
5141      REAL LAMEDA
5142      CALL KANSIZ(107,500)
5143      XVLANG= (A2000-VELANG)/57.29576
5144      IF(MM.GT.0) GO TO 100
5145      MM=1
5146      WRITE(6,95)
5147      95 FORMAT(1H0,' THE VARIABLE MM HAS BEEN SET TO 1 ' )
5148      100 IF(NN.GT.0) GO TO 200
5149      NN=1
5150      WRITE(6,195)
5151      195 FORMAT(1H0,' THE VARIABLE NN HAS BEEN SET TO 1 ' )
5152      200 CONTINUE
5153      KCELL=IFIX(KNLXT/T1)
5154      IF(KCELL.LT.8192) GOTO 50
5155      KCELL=2192
5156      KNEXT=11*8192.0
5157      50  KK=IFIX(KNLXT/KCELL)
5158      IF(KK.GT.0) GO TO 211
5159      KK=1
5160      211 WRITE(6,212) KK
5161      212 FORMAT(1H0,' THE VARIABLE KK HAS BEEN SET TO',15)

```

6-9a

08/16/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

6-10

CARD NO

CONTENTS

```

5162      NSCAT=MM*NN*KK
5163      NBLKS= NSCAT/250
5164      IF(NBLKS*250.LT.NSCAT) NBLKS=NBLKS+1
5165      IF(NBLKS.GT.200) GO TO 888
5166      DELAZ= AZLXI/FLUAT(MM)
5167      DELEL= ELEXT/FLUAT(NN)
5168      MODE=2
5169      IF(LAMBDA.NE.0.0) DOPFRQ=-2.0*WNOVL/LAMBDA
5170      IF(DOPFRQ.EQ.0.0.AND.RWPH.EQ.0.0) MODE=1
5171      IPNT=1
5172      IREC=1
5173      DO 1000 N=1,NB
5174      DO 1010 M=1,MM
5175      DO 1020 L=1,LL
5176      IF(IPNT.LE.250) GO TO 950
5177      IF(LIMP.EQ.0) GO TO 940
5178      WRITE(6,930) (CLUX(J),CLUY(J), J=1,IPNT)
5179      930 FORMAT(4H ,G12.0)
5180      940 WRITE(2*IREC) (SCAT
5181      IREC=IREC+1
5182      IPNT=1
5183      950      A=KRAND(INES)
5184      CLUX(IPNT)= A*KRAND(2)
5185      CLUY(IPNT)= A*0.0M
5186      1020      IPNT=IPNT+1
5187      1010      CONTINUE
5188      1000      CONTINUE
5189      IF(IPNT.EQ.1) GO TO 887
5190      WRITE(2*IREC) (SCAT

```

C

C

C

```

5141      IF (TEMP.NE.0) WRITE(6,930) (CLOX(J),CLOY(J), J=1,IPNT)
5142
5143      887 ICFLG=1
5144
5145      RETURN
5146
5147      USE WRITE(6,889) KK,NN,MM,NSCAT
5148
5149      889 FORMAT(1H,' THE PRODUCT OF KK=',14,' NN=',14,' MM=',14,' IS 100
5150
5151      *LARGE,',110,' THIS JOB WILL TERMINATE' )
5152
5153      RETURN 1
5154
5155      END

```

SUBROUTINE CUMDIS

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
CUMDIS	Peripheral	208,209
CUM2	Peripheral	Not User Referenced
OUTCUM	Peripheral	210,211
PDF	Peripheral	212,213

2. PURPOSE:

This subroutine processes a sequence of input samples in order to generate the probability density histogram and cumulative probability distribution.

3. INPUT PARAMETERS (CUMDIS):

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
TLIM	R	F	The upper limit of the histogram independent variable
BLIM	R	F	The lower limit of the histogram independent variable
NIXF	R	I	The number of elements in the histogram

INPUT PARAMETERS (OUTCUM):

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NCPACK	R	I	The number of histogram elements combined to form one element of the output cumulative distribution

INPUT PARAMETERS (PDF):

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NDPACK	R	I	The number of histogram elements combined to form one element of the output probability density.

4. CALLING SEQUENCES:

Initial histogram computation

CALL CUMDIS (DATAIN, XF)

Where: DATAIN contains the Input Waveform
XF contains the Output Histogram

Add additional data to histogram

CALL CUM2 (DATAIN, XF)

Where: DATAIN contains the Input Waveform
XF contains the Output Histogram

Cumulative distribution

CALL OUTCUM (DATOUT, XF)

Where: XF contains the Input Histogram
DATOUT contains the cumulative
distribution

Probability density

CALL PDF (DATOUT, XF)

Where: XF contains the Input Histogram
DATOUT contains the probability density

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The number of points NIXF in the histogram XF cannot exceed 8000.
- b. Any value of the input array which lies below the bound of BLIM will be placed in the lowest element of the histogram. Any value of the input array which lies above the bound of TLIM will be placed in the highest element of the histogram.

- c. The variables NCPACK and NDPACK must be greater than or equal to 1.
- d. Flow Chart: Page 9-152
- e. Cross Reference Table: Page 9-228

6. THEORY OF OPERATION

- a. The width of the histogram elements is determined using the following expression

$$DINC = (TLIM - BLIM) / NIXF$$

- b. The value of each sample in the input array DATAIN is divided by DINC to determine the corresponding histogram element. The number of samples which lie within the bounds of that histogram element is incremented by one.
- c. After the histogram is generated an entry through OUTCUM causes the cumulative distribution to be calculated and placed in array DATOUT. The number of histogram elements combined to generate one output sample is determined by the variable NCPACK.
- d. If the density function is desired, entry is made through the entry point PDF. The number of histogram elements combined to generate one output sample is determined by the variable NDPACK.

6-14

```

3917      SUBROUTINE COMD15(DATAIN,XF)
3918      C
3919      COMMON/BLK1/BLK1(500)
3920      C
3921      DIMENSION DATAIN(1),DATOUT(1),XF(1)
3922      DECEIVANCE (BK1( 39), TLIM
3923      1          (BK1( 40), FLIM
3924      2          (BK1( 42), NCPACK
3925      DATA N193,N194,N195,N196/-3,-2,-1,0/
3926      C
3927      C      CALCULATE HISTOGRAM ELEMENT WIDTH
3928      DINC = (TLIM-FLIM)/FLOAT(N1XF)
3929      LADD = 1- IFIX(BLIM/DINC)
3930      XF(N193)=DDEUL(N1XF)
3931      XF(N194)=BLIM
3932      XF(N195)=DINC
3933      XF(N196)=DDEUL(LADD)
3934      C
3935      DO 20 J=1,N1XF
3936      20 XF(J)=0.0
3937      CL TO 1000
3938      ENTRY COM2(DATAIN,XF)
3939      N1XF=IDDEUL(XF(N193))
3940      DINC=XF(N195)
3941      LADD=IDDEUL(XF(N196))
3942      C
3943      C      ADD POINTS TO PREL. DENSITY HISTOGRAM

```

06/11/75

INFLU LISTING

AUTOFLOW CRAFT SET - FWC/SCL RAUSIM

CARD NO

CONTENTS

674a

3944	C		C
3945		1000 CONTINUE	C
3946		NDATA = IECLL(DATAIN(N193))	C
3947		DINC=1.0/DINC	C
3948		DO DO J=1,NDATA	C
3949		IADD=IFIX(DATAIN(J)*DINC) + IADD	C
3950		IF(IADD.GT.NIXF) IADD=NIXF	C
3951		IF(IADD.LT.1) IADD=1	C
3952		DO XF(IADD)=XF(IADD)+1.0	C
3953	C		C
3954		RETURN	C
3955	C		C
3956		ENTRY (CIRCM(DATOUT,XF))	C
3957	C		C
3958	C	CALCULATE CUMULATIVE PROB. DISTRIBUTION	C
3959	C		C
3960		IF(NCPACK.LE.0) NCPACK=1	C
3961		NIXF=IECLL(XF(N193))	C
3962		DINC=XF(N195)	C
3963		KEND = NIXF/NCPACK	C
3964		DATOUT(N195) = DINC*FLOAT(NCPACK)	C
3965		DO I = -NCPACK	C
3966		CUM=0.0	C
3967	C		C
3968		DO DO J=1,KEND	C
3969		DO I = JJ + NCPACK	C
3970		DO K=1,NCPACK	C
3971		70 CUM=CUM+XF(I+JJ)	C
3972		30 DATOUT(J)=CUM	C

3970	C		C
3975		ICCN=0	C 6-15
3977		CC 10 200	C
3978	C		C
3979		ENTRY FOR(DATOUT,XF)	C
3979	C		C
3979		ICCN=1	
3980		IF (NDPACK.EQ.0) NDPACK=1	
3981		NIXF=BOOL(XF(N195))	
3982		DINC=XF(N195)	C
3983		KEND = NIXF/NDPACK	C
3984		DATOUT(N195)=DINC*FLOAT(NDPACK)	C
3985		JJ= -NDPACK	C
3985	C		C
3987		CUM=0.0	C
3988		CC 100 JJ=1,KEND	C
3989		LEN=0.0	C
3990		JJ=JJ + NDPACK	C
3991		CC 90 K=1,NDPACK	C
3992		DEN=LEN+XF(K+JJ)	C
3993		90 CONTINUE	C
3994		CUM=CUM+LEN	C
3995		DATOUT(JJ)=DEN	C
3996		100 CONTINUE	C
3997	C		C
3998		200 CONTINUE	C
3999		CUM=1.0/CUM	
4000		IF (ICCN.EQ.1) CUM=CUM/DATOUT(N195)	
4001		CC 300 K=1,KEND	C

00711775

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NO	****	CONTENTS	****
4002		DATOUT(K)=DATOUT(K)*CUM	C
4003		300 CONTINUE	C
4004		DATOUT(N193) = BOOL(KEND)	C
4005		DATOUT(N194)=XF(N194)	C
4006		RETURN	C
4007		END	C

SUBROUTINE ERGYCP

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
ERGYCP	Peripheral	118
ERGYRE	Peripheral	116,117

2. PURPOSE:

This subroutine computes the energy contained in a waveform.

3. INPUT PARAMETERS:

None

4. CALLING SEQUENCES:

Complex waveform

CALL ERGYCP (X,Y)

Where: X contains the Input Waveform - R

Y contains the Input Waveform - I

$$\text{Energy} = \Delta \sum_{J=1}^{\text{NPTS}} X(J) ** (J) + Y(J) * Y(J)$$

NPTS = Number of waveform samples

Δ = Independent variable increment between samples

Real Waveform

CALL ERGYRE (X)

Where: X contains the Input Waveform

$$\text{Energy} = \Delta \sum_{J=1}^{\text{NPTS}} X(J) * X(J)$$

a. The computed energy is printed on the output listing.
The measurement unit is watt-nanoseconds, normally.

b. Flow Chart: Page 9-70

c. Cross Reference Table: Page 9-217.

```

2012      SUBROUTINE EGYCF(X,Y)
2020      DIMENSION X(1),Y(1)
2034      DATA N195,N199,N195/-3,-2,-1/
2035      MDEL=1
2036      CDEL=100
2037      EGY=EYEL(X)
2038      MDEL=0
2039      100  N15=2002(X(N195))
2040      DELT=X(N195)
2041      EGY=0.0
2042      DO 200 J=1,NPTS
2043      EGY=EGY+X(J)*X(J)
2044      200  CONTINUE
2045      IF(MDEL.EC.EC) GOTO 500
2046      DO 300 J=1,NPTS
2047      EGY=EGY+Y(J)*Y(J)
2048      300  CONTINUE
2049      EGY=EGY*DELT
2050      WRITE(6,1000) EGY
2051      1000  FORMAT(' ENERGY IN THE WAVEFORM =',E15.5,' WATT-NAVSECONDS ')
2052      RETURN
2053      END

```

SUBROUTINE PLOTTR

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
PLOTTR	PERIP	307,308,309,310

2. PURPOSE:

This subroutine converts the data contained in the input array into a form suitable for plotting by a HP9820 desk calculator. In addition, the output can also be listed on a TTY via TSS or punched on cards.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
SIVST	R	F	Value of the independent variable at the start of the plot
XIVRNG	R	F	Range of the independent variable which is to be plotted
LOGFLG	R	I	Control index which indicates the nature of the input data
NSKP	O	I	This variable controls the number of input samples shipped between each plotted point. For example, if NSKP=0 or 1, each point is plotted; if NSKP=2, every other point is plotted.
NAUTO	O	I	Control flag for automatic data scanner which determines the maximum and minimum values of the dependent variable. If NAUTO=0, the scanner determines the value of TH and TL. If NAUTO=1, the use provided input parameters TH and TL are used.
TH	O	F	Maximum dependent variable value to be plotted

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
IC	R	I	Logical unit designator for output file
VIL	0	C	Independent variable label
VDL	0	C	Dependent variable label
GLBL	0	C	Plot title and miscellaneous data

4. CALLING SEQUENCES:

CALL PLOTTR (DV)

Where: DV contains the Input Waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. If an output file code (IC) is not specified then no data will be transferred and control will immediately be transferred back to the calling program.
- b. If the user specified parameters, XIVST and SIVRNG, specify a plot that is outside the range of the input data array then the complete input array is processed and transferred to the output file.
- c. If the file code is negative, specifying HP paper tape format, then all plot labels and miscellaneous data are suppressed. Only TH, TL, XIVOR, DELTA, and NPTS and the plot data are transferred.
- d. All floating point output values are in E13.5 format.
- e. Flow Chart: Page 9-90
- f. Cross Reference Table: Page 9-219.

6. THEORY OF OPERATION

- a. Upon entering the subroutine the parameter IC is tested to determine that IC is not zero and what operating mode is to be used. If IC = 0 then the HP paper tape mode (ICON = 2) is selected. If IC = 1 then the card image mode (ICON=1) is selected.
- b. If (ICON=1) the following lines (records) are transmitted to the output file.
 1. Header line "** Punch Card Data Output"
 2. XIVST, XIVRNG, XIVFR (not used), LOGFLG, VIL (first 12 characters)
 3. VIL (last 60 characters)
 4. VDL (first 60 characters)
 5. VDL (last 12 characters), GLBL (first 48 characters)
- c. The input parameters, XVST and XIVRNG, are tested to verify that the range of the desired plot is within the limit of the input array, DV. If the desired plot extends past either end of DV, then the values of XIVST and XIVRNG are changed to be compatible with the available data. If the desired plot is totally outside the available data, then the complete input array is processed.
- d. If NAUTO = 0 the input data is scanned to determine the maximum, TH, and the minimum, TL, values of the dependent variable to be plotted.
- e. IF(ICON=1) the following lines (records) are transmitted to the output file.
 1. GLBL (last 10 characters)
 2. NSKP, NAUTO, IC
- f. The following lines (records) are transmitted to the output file.
 1. TH, TL, XIVOR, DELTA, NPTS
 2. 5 dependent variable values
 3. 5 dependent variable values
 - .
 - .
 - .
 - N. Last line of plot data

6-21

```

2440      C FLUTR
2441          SUBROUTINE FLUTR(DV)
2442          COMMON/CLK1/ BK1(200),VIL(50),VDE(50),GLFL(50)
2443          EQUIVALENCE ( BK1( 59), XIVST      ),( BK1( 60), XIVRNG      ),
2444          *              ( BK1( 61), XIVER      ),( BK1( 62), LOGFLG      ),
2445          *              ( BK1( 63), NSKP      ),( BK1( 64), NAUTO      ),
2446          *              ( BK1( 65), IH      ),( BK1( 66), IL      ),
2447          *              ( BK1(113), IC      )
2448          DATA N193,N194,N195,N196/-3,-2,-1,0/
2449          DIMENSION CV(1)
2450          IF(ABS(IC).GT.4) GO TO 200
2451          IF(IC) 500,100,10
2452          IC=5
2453          GO TO 9
2454          IC=-10
2455          GO TO 2
2456          GO TO 12
2457          * WRITE(10,5)
2458          * FORMAT(11X,11H FUNCH CARD DATA CULFCT *)
2459          C
2460          * WRITE(10,7) XIVST, XIVRNG, XIVER, LOGFLG, ( VIL(J), J=1,2 )
2461          * FORMAT(1H,5D10.5,11D,4D)
2462          C
2463          * WRITE(10,9) ( VDE(J), J=1,10 )
2464          * FORMAT(1H,10D10.5)

```

6-210

```

2465      C
2466      WRITE(10,6) ( VDL(J), J=1,10 )
2467      C
2468      WRITE(10,6) ( VDL(J), J=11,12 ), ( GLBL(J), J=1,8 )
2469      C
2470      12  CONTINUE
2471      NTTL = 1800L(DV(N193))
2472      XIVOR = DV(N194)
2473      DELTA = DV(N195)
2474      ENDDAT = XIVOR + DELTA*(NTTL-1)
2475      NST=FIX((XIVST-XIVOR)/DELTA)+1
2476      IF(NST.LT.1) NST=1
2477      NSTOP=FIX((XIVST-XIVOR+XIVRNG)/DELTA)+1
2478      IF(NSTOP.GT.NTTL) NSTOP=NTTL
2479      IF(NSTOP.GT.NST) GOTO 11
2480      NSTOP=NTTL
2481      NST=1
2482      11  CONTINUE
2483      C
2484      XIVOR=XIVOR+(NST-1)*DELTA
2485      NPTS=(NSTOP-NST)/NSKP+1
2486      DELTA=DELTA*FLOAT(NSKP)
2487      IF(NAUTL.EQ.1) GO TO 100
2488      TH=DV(NST)
2489      TL=TH
2490      DO 50 J=NST,NSTOP
2491      TH=AMAX1(DV(J),TH)
2492      TL=AMIN1(DV(J),TL)
2493      50  CONTINUE

```

12/11/01

10.007 425 111.0

AUTOFLOW CHART SET - FWD/SCL KAUSIM

641. 100

• • • •

CONTENTS

6-22

```

2494         CONTINUE
2495         DO 1000 J=1,N
2496             WRITE(10,66) (CLP(J),J=15,19,16)
2497         66    FORMAT(10F4.6)
2498             WRITE(10,18) NSKF,NACTU,10
2499         18    FORMAT(10F4.15)
2500             WRITE(10,22) 10,10,XIVCF,DELTA,NP12
2501         22    FORMAT(10F4.15,11F5)
2502             NUPF=NSKF
2503             NSCF=NSF,NSCF,1,NUPF
2504             N=J+NSKF
2505             WRITE(10,24) (UM(N),N=J,K,NSKF)
2506         24    FORMAT(10E15.4)
2507             GO TO 1000
2508     C
2509         RETURN
2510     66    WRITE(10,66)
2511     66    FORMAT(' FILE DATA FILE IMPROPERLY DESIGNED...NO DATA',
2512             * ' TRANSFERRED')
2513         RETURN
2514     END

```

SUBROUTINE PLTFMT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
PLTFMT	PERIP	113

2. PURPOSE:

This subroutine is used to transform input data which is provided at equal increments of $\sin \theta$ into an output array which is equally incremented in θ . The new points are determined using a parabolic interpolation. In addition, the input array is tested to determine if all input points are in visible space (independent variable between $+1$), and superfluous data is eliminated.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
DELTHE	R	F	Output array independent variable increment
STARPT	0	F	Lower limit of the output array
STOPT	0	F	Upper limit of the output array

4. CALLING SEQUENCES:

CALL PLTFMT (GXIN, GYIN, GOUT, \$nnnn)

Where: GXIN contains the Input E-field - Real Component
 GYIN contains the Input E-field - Imaginary Component
 GOUT contains the Output gain

nnnn is the statement number to which control is transferred when a non-recoverable error is detected.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The lower and upper limits of the output array must be specified in ascending order in order to

be effective. If the upper limit is less than or equal to the lower limit or no limits are specified, all valid input data will be interpolated.

- b. A double parabolic interpolation is performed to determine each output sample except in the vicinity of the end points.
- c. Flow Chart: Page 9-175
- d. Cross Reference Table: Page 9-231.

6. THEORY OF OPERATION

PLTFMT is intended primarily for converting antenna pattern data from equal increments of $\sin \theta$, generated by the Fast Fourier Transform, to equal increments of θ . This conversion is important in that it allows the pattern array to be stored as an array of gain points, equally spaced in angle. This eliminates the necessity of storing a gain-angle data pair, since the location of any point in angle can be determined from the ordinal number of the point, the origin of the array, and the angle increment. It is especially important in converting antenna pattern data for plots, since it is desirable that the points be equispaced in the independent variable direction. An additional feature of PLTFMT is that it will automatically eliminate the extraneous output from the Fast Fourier Transform, which is contained in the region commonly known as "invisible space", i.e., that region where the dependent variable lies outside $-1 \leq \sin \theta \leq +1$. Furthermore, by specifying angle limits for the region of interest, superfluous data is eliminated, thereby reducing processing time and storage requirements.

- a. In the first section of the program the input array input parameters are initialized. The end points specified for the input data are examined to determine if they are realistic and within the boundaries of the input array. If not, one of three things happens. (1) If the specified end points exceed the array limits or ± 90 degrees, they are reduced to a useable value. (2) If the limits are equal or reversed, i.e., start point equals stop point or start point is greater than stop point, the limits are set to ± 90 degrees or to be array limits, whichever is less. (3) If the specified

end points are in the proper order, but do not encompass any of the array data, the problem is terminated. If the adjusted end points do not encompass all of the input array, the portion of the input array to simplify processing and the array definition parameters are adjusted accordingly.

- b. The output array elements are computed by interpolating between the input array points. A double parabolic interpolation is used except in the end segments where a single parabolic interpolation is used.

The double parabolic interpolation is performed by generating two quadratic equations, using three adjacent points to generate each equation. Assuming the points are labeled X_1 , X_2 , X_3 , and X_4 , and the required location is between X_2 and X_3 , then the equations are generated using X_1 , X_2 , X_3 , and X_2 , X_3 , X_4 . Once the quadratic coefficients have been determined, the value of each quadratic at the point of interest is determined, and the two values are averaged to find the value of the dependent variable.

Within the end segments, adjacent points are not available to obtain two quadratic equations. Therefore, a single quadratic is generated and the necessary points are interpolated from this single equation. In all cases, if the point of interest falls exactly on a point in the input array, the actual value of that point is used.

[illegible]

6-26a

4553	ORIGIN=OXIN(N194)	2110
4554	LELIN=OXIN(N195)	2120
4555	IF(STARPT.GE.-90.0) GO TO 302	2130
4556	STARPT=-90.0	2140
4557	WRITE(6,100)	2150
4558	100 FORMAT('START POINT WAS DEFINED BELOW HORIZON. START POINT HAS B	2160
4559	*EEN REDEFINED TO -90 DEG.*')	2170
4560	302 IF(STOPT.LE.90.0) GO TO 303	2180
4561	STOP=90.0	2190
4562	WRITE(6,111)	2200
4563	111 FORMAT('STOP POINT WAS DEFINED BELOW HORIZON. STOP POINT HAS BEE	2210
4564	*N REDEFINED TO 90 DEG.*')	2220
4565	303 IF(STARPT.L1.STOPT) GO TO 305	2230
4566	STARPT=-90.0	2240
4567	STOP=90.0	2250
4568	WRITE(6,109)	2260
4569	109 FORMAT('STARTING POINT SPECIFIED AS GREATER THAN OR EQUAL TO END	2270
4570	* POINT. ALL AVAILABLE VISIBLE SPACE WILL BE INTERPOLATED.*')	2280
4571	305 SINSTR=SIN(STARPT/RAD)	2290
4572	SINST=SIN(STOPT/RAD)	2300
4573	DELKAD=DELTHE /RAD	2310
4574	4 NSTRT=FIX((SINSTR-ORIGIN)/LELIN*0.5)	2320
4575	IF(NSTRT.GE.2) GO TO 10	2330
4576	STARPT=STARPT+DELTHE	2340
4577	SINSTR=SIN(STARPT/RAD)	2350
4578	GO TO 9	2360
4579	10 NSTOP=FIX((SINST-ORIGIN)/LELIN*0.5)	2370
4580	IF(NSTOP+1.LE.NPIS) GO TO 15	2380
4581	STOP=STOP+DELTHE	2390

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

6-27

CARD NO	****	CONTENTS	****
4582		SINSTP=SIN(STUPT/RAD)	2400
4583		GO TO 10	2410
4584	15	STPRAD=STUPT/KAD	2420
4585		ANGLE=STAKPT/RAD	2430
4586		VIV=ORIGIN+DELIN*FLOAT(NSTRT-2)	2400
4587		K=1	2441
4588		WRITE(6,1011) STUPT,STARPT,ORIGIN,DELIN,NPTS,NSTRT,NSTOP	2411
4589	1011	FORMAT(1H,4E15.7,3I10)	2412
4590		DO 300 J=NSTRT,NSTOP	2450
4591		X1=ARSIN(VIV)	2460
4592		X2=AFSIN(VIV+DELIN)	2470
4593		VIV=VIV+DELIN	2480
4594		X3=ARSIN(VIV+DELIN)	2490
4595		BREAK=(X3+X2)/2.0	2500
4596		IF (BREAK.GT.STPRAD) BREAK=STPRAD	2510
4597		Y1=GXIN(J-1)	2520
4598		Y2=GXIN(J)	2530
4599		Y3=GXIN(J+1)	2540
4600		MODE=1	2550
4601		X12=X1-X2	2560
4602	350	YX12=(Y1-Y2)/X12	2570
4603		C1=(YX12-((Y1-Y3)/(X1-X3)))/(X2-X3)	2580
4604		C2=YX12-C1*(X1+X2)	2590
4605		C3=Y1-(C1*X1+C2)*X1	2600
4606		GO TO (401,402), MODE	2610
4607	401	CX1=C1	2620
4608		CX2=C2	2630
4609		CX3=C3	2640
4610		ML1=C1	2650

4011	Y1=CYIN(J-1)	6-27-a 2660
4012	Y2=CYIN(J)	2670
4013	Y3=CYIN(J+1)	2680
4014	GO TO 300	2690
4015	401 CY1=C1	2700
4016	CY2=C2	2710
4017	CY3=C3	2720
4018	401 IF (ANGLE.GT.PI/2) GO TO 299	2730
4019	C1=(CX1*ANGLE+CX2)*ANGLE+CX3	2740
4020	C2=(CY1*ANGLE+CY2)*ANGLE+CY3	2750
4021	GOUT(K)=10.0*ALG10((R*GR+G)*G)	2750
4022	ANGLE=ANGLE+DEGRAD	2770
4023	K=K+1	2780
4024	GO TO 400	2790
4025	299 IF (ANGLE.GT.SIPRAD) GO TO 301	2800
4026	300 CONTINUE	2810
4027	301 GOUT(N193)=GOUL(K-1)	2820
4028	GOUL(N194)=STAKPT	2830
4029	GOUL(N195)=ULLTHE	2840
4030	K=TURN	2850
4031	END	2860

SUBROUTINE PTLIST

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
PTLIST	PERIP	303,304,305,306

2. PURPOSE:

This subroutine converts the data contained in the input array into a compact form for transmission to a remote processing station which has a CRT plotter.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
ST	R	F	Value of the independent variable at the start of the plot
RNG	R	F	Range of the independent variable which is to be plotted
LF	R	I	Control index which indicates the nature of the input data
NSKP	O	I	This variable controls the number of input samples skipped between each plotted point. For example, if NSKP = 0 or 1, each point is plotted; if NSKP = 2, every other point is plotted.
NAUTO	O	I	Control flag for automatic data scanner which determines the maximum and minimum values of the dependent variable. If NAUTO = 0 the scanner determines the value of TH and TL. If NAUTO = 1, the user provided input parameters TH and TL are used
TH	O	F	Maximum dependent variable value to be plotted

AD-A102 278

GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV F/G 17/9
ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING, VOLUME II. PAR--ETC(U)
JUN 76 R J HANCOCK, F H CLEVELAND F30602-73-C-0380

UNCLASSIFIED

RADC-TR-76-186-VOL-2-PT-1 NL

4 of 4

AD-A102 278



END

DATE

FILED

9 81

DTIC

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
TL	O	F	Minimum dependent variable value to be plotted
IFCODE	R	I	Logical unit designator for output file

4. CALLING SEQUENCES:

CALL PTLIST (DV)

Where: DV contains the Input Waveform

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. A number of tests are performed on the input data to ensure that a plot can be successfully completed. If all the necessary conditions are not satisfied, a condition code index is set and the plotter is bypassed with control returned to the calling program. The following is a list of the condition code values and the meaning of each:

<u>CONDITION CODE INDEX</u>	<u>Error Message</u>
1	The starting point, ST, is outside the input data array. (Too large)
2	The starting point, ST, is outside the input data array. (Too small)
3	The input data variables, ST, RNG, are improperly defined and result in a condition which places the stopping point before the starting point.
4	The user supplied value of TH is less than TL when NAUTO = 1.

- b. The subroutine PACK is used to convert the fixed point output data into a form suitable for transmission to remote terminal via TSS.

c. If an output file code (IFCODE) is not specified then no data will be transferred and control will immediately be transferred back to the calling program.

d. Flow Chart: Page 9-82

e. Cross Reference Table: Page 9-218

6. THEORY OF OPERATION

a. Upon entering the subroutine the input parameters are first tested to verify that the range of the desired plot is within the limits of the input data array, DV.

b. If NAUTO = 0, the input data is scanned to determine the maximum, TH, and minimum, TL, values to be plotted. From these two variables the upper and lower limits of the plot are determined.

c. For magnitude data, TH and TL are tested to determine if the input data is bipolar or unipolar.

d. From TH and TL the dependent variable range of the plot is computed and the LSB and BIAS to be used in the conversion from floating point to fixed point is computed.

e. The following plot parameters are processed as one record:

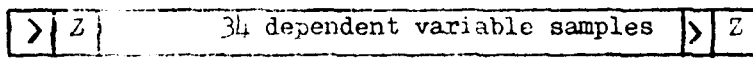
1. TH - Maximum value of input array
2. TL - Minimum value of input array
3. IVST - First value of input array which was processed for output
4. NSTOP - Last value of input array which was processed for output
5. OR - Independent variable value for first output point
6. DEL - Independent variable spacing
7. RNG - Independent variable range of the output data

f. The following plot parameters are processed for output as one record:

1. BIAS - Number subtracted from input data before conversion to fixed point
 2. XLSB - LSB of output data
 3. - Not used
 4. ITH - Largest dependent variable output value
 5. ITL - Smallest dependent variable output value
 6. NOUT - Number of output samples
- g. The dependent variable values are processed for output. The mechanization equation for this conversion is the following:

$$IDAT = (DV(J) - BIAS) / XLSB \quad IVST \leq J \leq NSTOP$$

The integer IDAT is then processed by the routine PACK which converts it into 2 characters and packs it into the array line which represents our output record and has the following structure:



header character pair

trailer character pair

```

2260      SUBROUTINE PILEST( DV)
2261      COMMON/BER17/ BK1(200),MIL(50),VOL(50),CLDE(50)
2262      EQUIVALENCE ( BK1(50), ST          ),( BK1(20), KNO          ),
2263      *              ( BK1(62), LF          ),( BK1(65), NSKP          ),
2264      *              ( BK1(60), NAUTG          ),( BK1(65), TH          ),
2265      *              ( BK1(66), TL          ),( BK1(115), IFCODE          )
2266      DIMENSION DV(11),LINE(10)
2267      DATA N193-N194,N195,N196/-3.42,-1.07,XM1800/1.00,-1.07
2268      DATA L1(10)/0.7517,0.000000/
2269      IF IFCODE(1000) GOTO 800
2270      IF (XKP(1),1) KKP=1
2271      NTL = TRUNC(DV(19.33))
2272      LF = LV(N194)
2273      DEL = LV(N195)
2274      L1=L1*880
2275      L1=NSKP*DEL*NTL-13
2276      IF (L1(10) GOTO 100
2277      KNO=KNO-LK+51
2278      ST=0
2279      GO CONTINUE
2280      IF (L1(10) GOTO 100
2281      KKP=1
2282      Z0 = L1(10)
2283      L1(10) = TRUNC(L1(10)/Z0)
2284      GO TO 100

```


6-32a

```

2291      IF (IVST.GE.NITL) NCTRL =1
2292      NSTOP = IFIX((ST-CR+RNGI)/DEL) + 1
2293      CR=CR+(IVST-1)*DEL
2294      IF (NSTOP.GI.NITL) NSTOP = NITL
2295      IF (NSTOP.LE.IVST) NCTRL = 4
2296      IF (NCTRL.NE. 0) GO TO 1100
2297      IF (LP.LI.2) (1 TO 40
2298      DO 45 J=IVST,NSTOP
2299      X=AMAX1(AES(DV(J)),XMINDB)
2300      IF (LP.LE.2) GO TO 41
2301      DV(J)=10.0*ALOG10(X)
2302      GO TO 42
2303      41 DV(J)=ALOG10(X)
2304      42 CONTINUE
2305      43 CONTINUE
2306      40 CONTINUE
2307      TH = DV(IVST)
2308      TL = TH
2309      IF (NCTRL.NE.0) GO TO 1100
2310      DO 100 J=IVST,NSTOP
2311      TH = AMAX1(DV(J),TH)
2312      TL = AMIN1(DV(J),TL)
2313      100 CONTINUE
2314      IF (TH.GT.TL) GO TO 2001
2315      NCTRL=5
2316      GO TO 1100
2317      2001 CONTINUE
2318      DEL=DEL*FLCAT(NSKP)
2319      NGUT=(NSTOP-IVST)/NSKP+1

```

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

6-33

CARD NO

CONTENTS

```

2320      LINE(1)=BOLD(TH)
2321      LINE(2)=BOLD(TL)
2322      LINE(3)=IVSI
2323      LINE(4)=NSTOP
2324      LINE(5)=BOLD(LK)
2325      LINE(6)=BOLD(DEL)
2326      LINE(7)=BOLD(RAG)
2327      WRITE(1,FOLDER) LINE
2328      IF(LF.16.1.0F.LF.16.3) GOTO 700
2329      IF(TH.0.0.0) ICON=1
2330      IF(TL.0.0.0) ICON=2
2331      IF(TH.0.0.0.AND.TL.0.0.0) ICON=3
2332      FTH= AMAX1(ABS(TH),ABS(TL))
2333      PTH= ALOG10(FTH)
2334      ITEST = IFIX(PTH)
2335      IF(PTH.1.0.0) GO TO 150
2336      X=FTH/(10.00**ITEST)
2337      GO TO 160
2338      150 X=FTH*(10.00**(IABS(ITEST)+1))
2339      160 CONTINUE
2340      N=0
2341      IF(X.0.1.0.AND.X.1.1.0) N=2
2342      IF(X.0.1.0.AND.X.1.3.5) N=4
2343      IF(X.0.3.5.AND.X.1.4.5) N=5
2344      IF(X.0.4.5.AND.X.1.7.0) N=8
2345      IF(X.0.7.0.AND.X.1.9.0) N=10
2346      IF(PTH.1.0.0) GO TO 370
2347      PTH=FLOOR(PTH)*(10.00**ITEST)
2348      GO TO 300

```

```

2349      TH = FLGAT(N) / (10.0** (IABS(ITEST)+1))
2350      GO CONTINUE
2351      ALSO=TH/4000.0
2352      BIAS=0.0
2353      GO TO(349,401,402),ICON
2354      349 AMX = 0.0
2355      AMN = -TH
2356      AMD = -TH/2.0
2357      DINC= TH/100.0
2358      GO TO 404
2359      401 AMX = TH
2360      AMN = 0.0
2361      AMD = TH/2.0
2362      DINC= TH/100.0
2363      GO TO 404
2364      402 AMX = TH
2365      AMN = -TH
2366      AMD = 0
2367      DINC= TH/50.0
2368      GO TO 404
2369      100 AMD=ABS(TH-TL)
2370      N = (IFIX (AMD/10.0))*10
2371      IF (AMD.GT.100.0) N=100
2372      AMX = (FLGAT(IFIX(TH/10.0)))*10.0
2373      IF (AMX.LT.TH) AMX=AMX+10.0
2374      AMN= AMX-FLGAT(N)
2375      IF (AMD.LT.10. AND N.LT.100) N=N+10
2376      IF (N.EQ.50) N=100
2377      IF (N.EQ.100) N=40

```

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RADSIM

634 ****

CARD NO

CONTENTS

```

2378      IF(N.EQ.60.OR.N.EQ.70) N=60
2379      BIAS= AMX-FLOAT(N)
2380      XLSB=0.05
2381      IF(N.LE.40) XLSB=0.01
2382      IF(N.GT.40.AND.N.LE.60) XLSB=0.02
2383      404 IF(NOTRLE.NE.0)GO TO 1100
2384      ITL=(TL-BIAS)/XLSB
2385      ITH=(TH-BIAS)/XLSB
2386      LINE(1)=BDDL(BIAS)
2387      LINE(2)=BDDL(XLSB)
2388      LINE(3)=N
2389      LINE(4)=ITH
2390      LINE(5)=ITL
2391      LINE(6)=ACCU
2392      WRITE(1F000) LINE
2393      LINE(1)=ELSTCH
2394      LINE(10)=ELSTCH
2395      501 IWD=1
2396      IDAT=18
2397      CALL PACK(INJCT,IWD,IBIT,LINE,1100)
2398      DO 400 J=1VST,NSTCP,NSKP
2399      IDAT=(OV(J)-BIAS)/XLSB
2400      IF(IDAT.GT.4095) IDAT=4095
2401      IF(IDAT.LT.-4095) IDAT=-4095
2402      CALL PACK(IDAT,IWD,IBIT,LINE,1500)
2403      GO TO 400
2404      500 WRITE(1F000) LINE
2405      IWD=1
2406      IDAT=18

```

2900 CONFIDENTIAL
 2901 IN THE EVENT OF A DISASTROUS FAILURE
 2902 WITHIN THE FRAME
 2903 RETURN
 2904 WITH THE FOLLOWING NOTE
 2905 RETURN
 2906 THE FOLLOWING IS NOT ATTACHED TO THE CONDITION CODE - 1, 14
 2907 WITH (O-V-G-I)
 2908 THE FOLLOWING IS NOT ATTACHED TO THE CONDITION CODE - 1, 14
 2909 WITH (O-V-G-I)
 2910 THE FOLLOWING IS NOT ATTACHED TO THE CONDITION CODE - 1, 14
 2911 WITH (O-V-G-I)
 2912 THE FOLLOWING IS NOT ATTACHED TO THE CONDITION CODE - 1, 14
 2913 WITH (O-V-G-I)

b-34a

SUBROUTINE RTOPDB

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
RTOPDB	Peripheral	103
RTOPM	Peripheral	110
RTOPM2	Peripheral	111
XYTOM	Peripheral	105
XYTOM2	Peripheral	106
XYTODB	Peripheral	104,108

2. PURPOSE:

This subroutine performs a rectangular to polar conversion on the input arrays. Depending on the entry point used, the output is in two arrays in the form of linear magnitude and phase (RTOPM), magnitude squared and phase (RTOPM2) or magnitude in decibels and phase (RTOPDB); or in one array of linear magnitude (XYTOM), linear magnitude squared (XYTOM2) or magnitude in decibels (XYTODB).

3. INPUT PARAMETERS: None

4. CALLING SEQUENCES:

Phase and decibel magnitude

CALL RTOPDB (X,Y,M,P)

Where:

X contains the Input Waveform - R

Y contains the Input Waveform - I

M contains the Output Waveform

P contains the Output Waveform - P

$$M(J) = 10.0 * ALOG10 \left[X(J) * X(J) + Y(J) * Y(J) \right]$$

$$P(J) = \frac{180.0}{\pi} * ATAN2 (Y(J), X(J))$$

Phase and linear magnitude

CALL RTCPM (X,Y,M,P)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 M contains the Output Waveform - M
 P contains the Output Waveform - P

$$M(J) = \sqrt{X(J) * X(J) + Y(J) * Y(J)}$$

$$P(J) = \frac{180.0}{\pi} * \text{ATAN2} (Y(J), X(J))$$

Phase and magnitude squared

CALL RTOPM2 (X,Y,M,P)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 M contains the Output Waveform - M
 P contains the Output Waveform - P

$$M(J) = X(J) * X(J) + Y(J) * Y(J)$$

$$P(J) = \frac{180.0}{\pi} * \text{ATAN2} (Y(J), X(J))$$

Linear magnitude and no phase

CALL XYTOM (X, Y, M)

Where: X contains the Input Waveform - R
 Y contains the Input Waveform - I
 M contains the Output Waveform

$$M(J) = \sqrt{X(J) * X(J) + Y(J) * Y(J)}$$

Magnitude squared and no phase

CALL XYTOM2 (X,Y,M)

Where:

X contains the Input Waveform - R

Y contains the Input Waveform - I

M contains the Output Waveform

$$M(J) = X(J) * X(J) + Y(J) * Y(J)$$

Magnitude in dB and no phase

CALL XYTODB (X,Y,M)

Where:

X contains the Input Waveform - R

Y contains the Input Waveform - I

M contains the Output Waveform

$$M(J) = 10.0 * \text{ALOGIO} \left[X(J) * X(J) + Y(J) * Y(J) \right]$$

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. For entry points RTOPDB and XYTODB the minimum output value is -200 dB.
- b. The range of the output phase angle is from -190° to $+170^{\circ}$. The break point where phase "rolls over" was chosen to be 170° instead of 180° because many waveforms have a large number of samples around 180° and each time the waveform's phase changes from $180 - E$ to $180 + E$ the output phase would change from $180 - E$ to $-180 + E$. This results in a phase plot with many undesirable and unnecessary discontinuities.
- c. Flow Chart: Page 9-156
- d. Cross Reference Table: Page 9-228.


```

4007 SUBROUTINE KTOPDB(X,Y,M,P)
4008 COMMON/BLK1/ BK1(500)
4009 EQUIVALENCE (BK1( 30), NAVG)
4010 DIMENSION X(1),Y(1),M(1),P(1)
4011 REAL M
4012 DATA N193,N196,N4/-3.0,4/
4013 DATA A1/57.7957E/
4014 IOUN = 1
4015 NPHAS=1
4016 GO TO 10
4017 ENTRY KTOPM(X,Y,M,P)
4018 IOUN = 2
4019 NPHAS=1
4020 GO TO 10
4021 ENTRY KTOPM2(X,Y,M,P)
4022 IOUN = 0
4023 NPHAS=1
4024 GO TO 10
4025 ENTRY XYTOPM(X,Y,M)
4026 IOUN=1
4027 NPHAS=0
4028 GO TO 10
4029 ENTRY XYTOPM2(X,Y,M)

```

R
6-38

K

R

R

R

K

R

K

R

K

R

K

R

K

R

K

R

K

R

K

R

```

4031      ICON=0
4032      NPHAS=0
4033      GO TO 10
4034      ENTRY XYTAB(X,Y,M)
4035      ICON=2
4036      NPHAS=0
4037      C
4038      10 CONTINUE
4039      N=1000(X(N195),N4,X,M)
4040      CALL DBLKX(N195,N4,X,M)
4041      IF(NPHAS.EQ.1) CALL DBLKX(N195,N4,X,P)
4042      IF(NPHAS.EQ.0.AND.NAVC.GT.1) GO TO 200
4043      IF(NPHAS.EQ.1) P(N196)=0.0
4044      C+++++*****
4045      DO 100 J=1,N
4046      IF(NPHAS.EQ.0) GO TO 50
4047      IF(X(J).EQ.0.0.AND.Y(J).EQ.0.0) GO TO 40
4048      P(J) = A1 * ATAN2(Y(J),X(J))
4049      IF(P(J).GT.170.0) P(J)=P(J)-360.0
4050      GO TO 50
4051      40 P(J) = P(J-1)
4052      50 CONTINUE
4053      M(J)= X(J)*X(J)+Y(J)*Y(J)
4054      IF(ICON.EQ.1) M(J)=SQRT(M(J))
4055      IF(ICON.NE.2) GO TO 100
4056      IF(M(J).LT.1.0E-20) M(J)=1.0E-20
4057      M(J)= 10.0*ALOG10(M(J))
4058      100 CONTINUE
4059      RETURN

```

00/11271

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RAUSIM

6-39

CARD NO

CONTENTS

```

4060      100  GO 210 J=1,N
4061          M(J)=X(J)*X(J)+Y(J)*Y(J)
4062      110  CONTINUE
4063          AV=0.0
4064          DO 220 J=1,NAVG
4065              AV=AV+M(J)
4066      220  CONTINUE
4067          N=N-NAVG
4068          XAVG=1.0/FLGAT(NAVG)
4069          DO 230 J=1,N
4070              DUM=M(J)
4071              M(J)=AV*XAVG
4072              IF (ICUN.EQ.1) M(J)=SQRT(M(J))
4073              IF (ICUN.NE.2) GO TO 225
4074              IF (M(J).LT.1.0E-20) M(J)=1.0E-20
4075              M(J)= 10.0*ALOG10(M(J))
4076      225  AV=AV-DUM+M(J+NAVG)
4077      230  CONTINUE
4078          M(N195)=BDUL(N)
4079          PLTURN
4080      END

```

R

SUBROUTINE SCANNR

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
SCANNR	PERIP	313

2. PURPOSE:

This subroutine scans an antenna pattern in the form of an input array and determines the location and gain of (1) the main lobe, (2) the highest ten sidelobes, and (3) the first twenty sidelobes on either side of the main lobe. In addition, the first minimum on either side of the main lobe is determined.

3. INPUT PARAMETERS:

None

4. CALLING SEQUENCES:

CALL SCANNR (DV)

Where: DV contains the Input Waveform in
 dB gain

5. RESTRICTION, REQUIREMENTS, MISCELLANEOUS DATA

- a. All output data from this subroutine is in the form of a printed listing which is formatted for ease of interpretation.
- b. If a vestigial lobe exists, the minimum between it and the main lobe will be detected as the main lobe null.
- c. Flow Chart: Page 9-161
- d. Cross Reference Table: Page 9-229

6. THEORY OF OPERATION

- a. The antenna pattern data is scanned from left to right (increasing θ) and the gain maxima are stored in a 22 element array. Within this array, the maxima are ordered according to gain with the highest in element 1 and the lowest in element 22.
- b. The pattern data to the right of the main lobe is scanned to locate the main lobe null and the first twenty sidelobes. After this data is processed and printed, a third scan is made to the left of the main lobe for the first null and the first twenty sidelobes.
- c. The logical equation used to determine the peaks of the lobes is as follows:

Given three adjacent points $DV(NTHPT-1)$, $DV(NTHPT)$, and $DV(NTHPT+1)$ the following difference parameters are computed.

$$\begin{aligned} DELDV1 &= DV(NTHPT) - DV(NTHPT-1) \\ DELDV2 &= DV(NTHPT+1) - DV(NTHPT) \end{aligned}$$

then if $DELDV1 \geq 0$ and $DELDV2 < 0$, the point $DV(NTHPT)$ is considered to be a maxima, i.e., the peak of a lobe.

- d. In order to find the nulls of the main lobe, the difference parameters defined in 6(c) are tested for the following condition:

$$DELDV1 < 0 \text{ and } DELDV2 \geq 0$$

If this condition is satisfied then $DV(NTHPT)$ is considered to be a null. After the first null is found, a flag is set which causes the null test to be bypassed for the remainder of the scan on that side of the main lobe. It should be noted that the null logic will consider the minimum between the main lobe and a vestigial sidelobe (if one exists) to be the main lobe null.

4129	SUBROUTINE SCANNER (IV)	6.42 UC7SCN01
4130	C	UC7SCN02
4131	***** THIS SUBROUTINE SCANS THE UNINTERPOLATED ANTENNA PATTERN DATA AND	UC7SCN03
4132	C PRINTS OUT MAIN LOBE NULL POINTS AND SIGNIFICANT SIDE-LOBES. *****	UC7SCN04
4133	C	UC7SCN05
4134	DIMENSION LCOEFF(22), ALCOEFF(22), DV(1)	UC7SCN06
4135	DATA N193,N194,N195,N196/-3,-2,-1,0/	UC7SCN07
4136	EQUIVALENC (RPTS,XPTS)	UC7SCN08
4137	C	UC7SCN09
4138	WRITE(6,1)	UC7SCN10
4139	1 FORMAT (100/47X,'* * * * * ANTENNA LOBE SCANNER * * * * * / / /')	UC7SCN11
4140	RPTS = 10000(DV(N193))	UC7SCN22
4141	XCPU = 10000(DV(N194))	UC7SCN13
4142	PTSPC = 10000(DV(N195))	UC7SCN14
4143	***** INITIALIZE LOBE STORAGE ARRAY *****	UC7SCN15
4144	DO 10 N=1,22	UC7SCN16
4145	LCOEFF(N) = 0	UC7SCN17
4146	10 ALCOEFF(N) = -2000.0	UC7SCN18
4147	C	UC7SCN19
4148	***** FIND MAXIMA *****	UC7SCN37
4149	DO 20 N1=NPT-1,NPTS	
4150	LCOEFF(N1) = DV(N1000) - DV(N1001)	UC7SCN39

6-43

```

      DELDV1 = (V(INTRPT+1) - V(INTRFT)) UC7SCN40
      IF (DELDV1.GT. 0.0 .OR. DELDV1.LE. 0.0) GO TO 70 UC7SCN41
      C UC7SCN42
      LDEPT(22) = INTRFT UC7SCN43
      ALDGN(22) = (V(INTRFT)) UC7SCN44
      C UC7SCN45
      C**** CURET TWENTY-ONE HIGHEST MAXIMA ***** UC7S
      C UC5U IN=122 UC7SCN47
      IF (ALDGN(22-HK) .LE. ALDGN(22-HF)) GO TO 70 UC7SCN48
      ANTR12 = ALDGN(22-HF) UC7SCN49
      ALDGN(22-HF) = ALDGN(22-HK) UC7SCN50
      ALDGN(22-HK) = ANTR12 UC7SCN52
      ALDEFT = LDEPT(22-HF) UC7SCN52
      LDEPT(22-HK) = LDEPT(22-HF) UC7SCN53
      C LDEPT(22-HK) = ALDEFT UC7SCN54
      C UC7SCN55
      C UC7SCN56
      C UC7SCN57
      C**** COMPUTE MAIN LURE ANGLE AND CUTOFF ***** UC7SCN58
      ANGMN = XCR0 + (LDEPT(1)-1)*PI/PL UC7SCN59
      WRITE (C410) ALDGN(1)
      DO FORMATT* THE MAIN LURE GAIN IS *,PI/PL)
      C UC7SCN61
      WRITE (C422) ANGMN UC7SCN62
      C UC7SCN63
      C UC7SCN64
      C UC7SCN65
      C UC7SCN66
      C UC7SCN67
      C**** COMPUTE SIDE-LOBE RELATIVE GAIN AND ANGLE, AND CUTOFF ***** UC7SCN67

```

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWC/SCL RADSIM

6-44

CARD NO

CONTENTS

4176	DL 75 N=2,21	UC7SCN68
4177	GAIN = ALDSGN(N) - ALDSGN(1)	UC7SCN69
4178	NK = N-1	UC7SCN70
4179	ANGLE = XORG + (LCPPT(N)-1)*FTSPL	UC7SCN71
4180	WRITE(6,74) NK, GAIN, ANGLE	UC7SCN73
4181	74 FORMAT (36X, 12, 18X, F13.7, 14X, F13.7)	UC7SCN
4182	75 CONTINUE	UC7SCN75
4183	C	UC7SCN76
4184	WRITE(6,76)	UC7SCN77
4185	76 FORMAT(1H1,21X,'MAIN LOBE NULL AND FIRST 20 SIDE-LOBES TO RIGHT OF MAIN LOBE')	UC7SCN78
4186	* (ADDSV1 MAIN LOBE)	UC7SCN79
4187	* /36X,'FAIR',13X,'REL. GAIN',13X,'TRUE ANGLE',13X,'REL. ANGLE'	
4188	* / 57X,'(OR)',16X,'(DEG)',16X,'(DEG)' /)	
4189	C	UC7SCN82
4190	C**** INITIALIZE RIGHTHAND SCANNER *****	UC7SCN83
4191	NDLFLG = 0	UC7SCN84
4192	NLS = 1	UC7SCN85
4193	NWSTRT = LCPPT(1)+1	UC7SCN86
4194	C**** FIND SIDE-LOBES TO RIGHT OF MAIN LOBE *****	UC7SCN87
4195	DL 65 NTHPT=NWSTRT,NPTS	UC7SCN88
4196	DELVD1 = DV(NTHPT) - DV(NTHPT-1)	UC7SCN89
4197	DELVD2 = DV(NTHPT+1) - DV(NTHPT)	UC7SCN90
4198	IF (DELVD1 .LT. 0.0 .AND. DELVD2 .LT. 0.0) GO TO 88	UC7SCN91
4199	C**** FIND RIGHTHAND MAIN LOBE NULL *****	UC7SCN92
4200	IF (NDLFLG .EQ. 0 .AND. DELVD1 .LT. 0.0 .AND. DELVD2 .LT. 0.0) GO TO 85	UC7SCN93
4201	NDLFLG = 1	UC7SCN94
4202	C**** COMPUTE ANGLE OF RIGHTHAND MAIN LOBE NULL *****	UC7SCN95
4203	ANGLRT = XORG + (NTHPT-1)*FTSPL	UC7SCN9
4204	WRITE (6, 84) ANGLRT	UC7SCN98

6-45

4205	84 FORMAT(30X,'MAIN LOBE',4X,'** NULL **',13X,F13.7)	UC7SCN00
4206	GO TO 85	
4207	C	UC7SCN01
4208	C*** COMPUTE RELATIVE GAIN AND RELATIVE ANGLE OF RIGHTHAND SIDE-LOBES	UC7SCN02
4209	82 CONTINUE	UC7SCN03
4210	GAIN = DVINTHPT - ALDPGN(1)	UC7SCN04
4211	ANGLB = XCRG + (NTHPT-1)*PTSPC	UC7SCN05
4212	RELANG=ANGLB - ANGMN	
4213	C	UC7SCN07
4214	WRITE(6,80) NLB, GAIN, ANGLB, RELANG	
4215	80 FORMAT (39X,13,3(13X,F13.7))	
4216	NLB = NLB+1	UC7SCN10
4217	C*** EXIT AFTER TWENTY SIDE-LOBES OR DROP THROUGH AT END OF DATA *****	UC7SCN11
4218	IF(NLB .GT. 20) GO TO 87	UC7SCN12
4219	C	UC7SCN13
4220	85 CONTINUE	UC7SCN14
4221	WRITE(6,86)	UC7SCN15
4222	86 FORMAT(45X, 'NO FURTHER SIDE LOBES ON THIS SIDE')	UC7SCN16
4223	C	UC7SCN17
4224	87 WRITE(6,88)	UC7SCN18
4225	88 FORMAT(1H0/22X,'MAIN LOBE NULL AND FIRST 20 SIDE-LOBES TO LEFT OF	UC7SCN19
4226	*(LELW) MAIN LOBE'	UC7SCN20
4227	*/30X,'RANK',13X,'REL. GAIN',13X,'TRUE ANGLE',13X,'REL. ANGLE'	
4228	*/57X,'(DB)',16X,'(DEG)',16X,'(DEG)' /)	
4229	C	UC7SCN23
4230	C*** INITIALIZE LEFTHAND SCANNER *****	UC7SCN24
4231	NWSTRT = LOBPT(1)	UC7SCN25
4232	NWSTP = NWSTRT-2	
4233	NLB = 1	UC7SCN27

08/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

6-46

CARD NO.	****	CONTENTS	****
4234		NULLFLG = 0	UC7SCN28
4235	C		UC7SCN29
4236	C****	FIND SIDE-LOBES TO LEFT OF MAIN LOBE *****	UC7SCN30
4237		DO 43 NIHTP=1,NWSTP	UC7SCN31
4238		NIHTP = NWSTP - NIHTP	UC7SCN32
4239		DELDV1 = DV(NXPT) - DV(NXPT-1)	UC7SCN33
4240		DELDV2 = DV(NXPT+1) - DV(NXPT)	UC7SCN34
4241		IF (DELDV1.LT. 0.0 .AND. DELDV2.LT. 0.0) GO TO 40	UC7SCN35
4242	C****	FIND LEFTHAND MAIN LOBE NULL *****	UC7SCN36
4243		IF (NULLFLG.EQ.1 .OR. DELDV1.GE.0.0 .OR. DELDV2.LT.0.0) GO TO 43	UC7SCN37
4244	C****	COMPUTE ANGLE OF LEFTHAND MAIN LOBE NULL *****	UC7SCN38
4245		AMNLEFT = XORG + (NXPT-1)*FTSPL	UC7SCN39
4246		NULLFLG = 1	UC7SCN41
4247		WRITE (6, 84) AMNLEFT	UC7SCN42
4248		GO TO 43	UC7SCN43
4249	C		UC7SCN44
4250	C****	COMPUTE RELATIVE GAIN AND RELATIVE ANGLE OF LEFTHAND SIDE-LOBES *	UC7SCN45
4251		GO CONTINUE	UC7SCN46
4252		GAIN = DV(NXPT) - ALDGN(1)	UC7SCN47
4253		ANGLE = XORG + (NXPT-1)*FTSPL	UC7SCN48
4254		RELANG=ANGLE - ANCMN	
4255	C		UC7SCN50
4256		WRITE (6, 84) NL, GAIN, ANGLE, RELANG	
4257		GO FORMAT(24X, 13, 2(13X, F13.7))	
4258	C		UC7SCN53
4259		IEL = IEL+1	UC7SCN54
4260	C****	EXIT AFTER TWENTY SIDE-LOBES OR DROP THROUGH AT END OF DATA ****	UC7SCN55
4261		IF IEL.GT. 20 GO TO 43	UC7SCN56
4262		GO CONTINUE	UC7SCN57
4263	C		UC7SCN58
4264		WRITE (6, 86)	UC7SCN59
4265	C		UC7SCN60
4266		GO RETURN	UC7SCN61
4267		END	UC7SCN62

SECTION 7
SUPERVISORY MODULES

This section includes those modules that supervise existing stimulus/transfer function modules to simulate the frequency scanned and time scanned array radar systems. This group includes ANTARY and TSRPAT, which are located in Volume III.

SECTION 8

SUBORDINATE MODULES

This section includes all those modules which are subordinate to the larger stimulus/transfer function modules, peripheral modules or supervisory modules. They include both subroutines and functions. Included in this group are the following modules:

ABORT	IBOOL
ANTINT	IFLD
AZGAIN/ELGAIN	IPACK
BLOCK DATA	PACK
DBLXX	RRAND

The following subordinate modules are related to the bistatic target model and target imaging and are located in Volume I, Part 3:

BISTGT
GAM
BESS
EXPI

EXPI is also located in Volume IV, Part 2.

SUBROUTINE ABORT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification</u>	<u>Reference Number</u>
ABORT	Subordinate	Not user referenced
ERRMSG	Subordinate	Not user referenced

2. PURPOSE:

This subroutine is used to print coded error messages to the user.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NCODE	R	I	Error message code number. 0 < NCODE ≤ 999

4. CALLING SEQUENCES:

CALL ABORT (NCODE)

A message is printed which states a fatal error has occurred and execution will terminate.

CALL ERRMSG (NCODE)

A message is printed which states a non-fatal error has occurred, a fix-up procedure done, and execution continued.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. Flow Chart: Page 9-81

b. Cross Reference Table: Page 9-218

8-3

```

2254      SUBROUTINE ABORT(NCODE)
2255      COMMON/SYS/ MODULE
2256      WRITE(6,1000) NCODE,MODULE
2257      1000  FORMAT(' ERROR : ',I3,' OCCURRED DURING EXECUTION OF MODULE : ',
2258      * 12,'.....FATAL ERROR: JOB WILL TERMINATE' )
2259      CALL EXIT
2260      ENTRY ERRMSG(NCODE)
2261      WRITE(6,1001) NCODE,MODULE

```

02/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CARD NO

CONTENTS

```

2262      1001  FORMAT(' ERROR : ',I3,' OCCURRED DURING EXECUTION OF MODULE : ',
2263      * 13,'.....FIX-UP DONE, REFER TO C.P.D. ; JOB WILL CONTINUE')
2264      RETURN
2265      END

```

SUBROUTINE ANTINT

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
ANTINT	Subordinate	(301)

2. PURPOSE:

This subroutine generates an interpolation table from the user provided sampled antenna pattern data. The interpolation table is subsequently used by the functions AZGAIN and ELGAIN to compute antenna gain.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NPT	R	I	Number of user specified antenna gain values
BSIT	O	F	Angle between zero angle of user specified pattern and zero angle in radar coordinates (usually = 0.0)
ANTP(2,75)	R	F	User specified gain pattern array. ANTP(1,J) is the gain of the Jth sample. ANTP(2,J) is the angle of the Jth sample.

4. CALLING SEQUENCES:

CALL ANTINT (NPT, BSIT, ANTP, CDEF)

Where: ANTP contains the Input Antenna Pattern
COEF contains the Output Interpolation coefficients table.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. If no antenna pattern is specified then the coefficients table is set up to represent an omnidirectional antenna of unity gain.

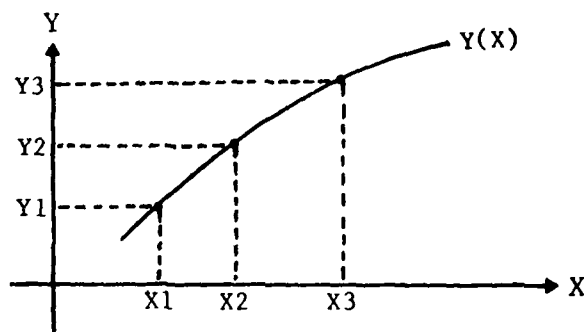
b. Flow Chart: Page 9-140

c. Cross Reference Table: Page 9-226.

6. THEORY OF OPERATION

This module implements a simplified antenna representation which is based on computing the antenna gain as the product of samples from an azimuth and elevation pattern cuts through the peak of the main lobe. The azimuth pattern is true antenna gain in power units. The elevation pattern is normalized to unity gain at the peak of the main lobe.

The parabolic interpolation scheme mechanized herein is described in the following paragraph. Given three points a parabolic curve can be determined which passes through them.



$$Y(X) = X^2 A + XB + C$$

where A, B, and C are the coefficients to be determined

$$A = \frac{XY_{12} - XY_{13}}{X_2 - X_3}$$

$$B = XY_{12} - A * (X_1 + X_2)$$

$$C = Y_1 - (X_1 A + B) * X_1$$

$$XY_{12} = \frac{Y_1 - Y_2}{X_1 - X_2} \quad \text{and} \quad XY_{13} = \frac{Y_1 - Y_3}{X_1 - X_3}$$

In determining the coefficient table elements a double parabolic coefficient is used where possible. This is done by determining 2 sets of coefficients for 2 curves through 4 points and averaging the coefficients. Typically a double parabolic curve fit is smoother than a single parabolic curve fit.


```

3633 SUBROUTINE ANTINT(NPT,ESIT,ANTP,COEF)
3634 C
3635 DIMENSION ANTP(2,75),COEF(4,75)
3636 IF(NPT.GT.2) GO TO 50
3637 COEF(1,1)=0.0
3638 COEF(2,1)=0.0
3639 COEF(3,1)=0.0
3640 COEF(4,1)=1.0
3641 COEF(1,2)=500.0
3642 COEF(2,2)=0.0
3643 COEF(3,2)=0.0
3644 COEF(4,2)=0.0
3645 IF(NPT.NE.2) GO TO 40
3646 COEF(1,2)=ANTP(2,1)+ESIT
3647 COEF(4,1)=ANTP(1,1)
3648 COEF(1,2)=ANTP(2,2)+ESIT
3649 40 COEF(1,75)=BOUL(2)
3650 NPT=2
3651 GO TO 50
3652 50 CONTINUE
3653 COEF(1,75)=BOUL(NPT)

```

8-6

06/11/75

INPUT LISTING

AUTOFLOW CHART SET - FWD/SCL FAUSIM

8-7
888

CARD NO

CONTENTS

```

3654      NPT2=NPT-2
3655      A1=0.0
3656      B1=0.0
3657      C1=0.0
3658      FLG =1.0
3659      DO 100 K=1,NPT2
3660      L=K+1
3661      M=K+2
3662      XY12=(ANTP(1,K)-ANTP(1,L))/(ANTP(2,K)-ANTP(2,L))
3663      XY13=(ANTP(1,K)-ANTP(1,M))/(ANTP(2,K)-ANTP(2,M))
3664      A2=(XY12-XY13)/(ANTP(2,L)-ANTP(2,M))
3665      B2=XY12-A2*(ANTP(2,K)+ANTP(2,L)+2.0*BS1T)
3666      TEMP=ANTP(2,K)+BS1T
3667      C2=ANTP(1,K)-TEMP*(A2*TEMP+E2)
3668      C1=ANTP(1,K)-AGTP(2,K)*(A2*ANTP(2,K)+S2)
3669      C
3670      CCLF(1,K)=ANTP(2,K)*PS1T
3671      CCLF(2,K)=FLG*(A1+A2)
3672      CCLF(3,K)=FLG*(C1+B2)
3673      CCLF(4,K)=FLG*(C1+C2)
3674      A1=A2
3675      B1=B2
3676      C1=C2
3677      FLG =0.0
3678      2=0 CONTINUE
3679      L
3680      CCLF(1,L)=ANTP(2,L)*PS1T
3681      CCLF(2,L)=A1
3682      CCLF(3,L)=B1

```

8-7a

```

3683      COEF(4,L)=01
3684      COEF(1,M)=ANTP(2,M)+RSIT
3685      COEF(2,M)=0.0
3686      COEF(3,M)=0.0
3687      COEF(4,M)=0.0
3688      C
3689      GO CONTINUE
3690      WRITE(6,4)INPT,DSIT
3691      45 FORMAT(1H1,'SUBROUTINE ANTIM   NPT =',I3, '   COE =',I4(11,7))
3692      WRITE(6,105) (( ANTP(J,K),J=1,7),( COEF(L,K),L=1,4),K=1,NPT)
3693      105 FORMAT (1H, 'C(1F420.7)')
3694      C
3695      RETURN
3696      END

```

SUBROUTINE AZGAIN/ELGAIN

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
AZGAIN	Subordinate	Not user referenced
ELGAIN	Subordinate	Not user referenced

2. PURPOSE:

This function is to compute the antenna azimuth or elevation gain for any angle within the range of the user specified antenna data.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
ANTANG	R	F	The angle of the antenna main lobe in radar coordinates

4. CALLING SEQUENCES:

CALL G = AZGAIN (AZANGL) * ELGAIN (ELANGL)

Where: GA is the antenna gain corresponding to an azimuth angle of AZANGL and elevation angle of ELANGL.

5. RESTRICTIONS, REQUIREMENTS AND MISCELLANEOUS DATA

- The coefficients table must have been successfully initialized by the subroutine ANTINT before the function can be executed.
- Flow Charts: Pages 9-159/160
- Cross Reference Tables: Page 9-229.

6. THEORY OF OPERATION

The input angle ANGL is tested to determine if it is within the range of the input data. If it is outside the range, the gain value returned is zero. Otherwise, the gain is computed from the gain coefficients (A,B, and C)

that are closest in angle to ANGL.

$$AZGAIN = (ANGR ** 2) * A + ANGR * B + C$$

Note: AZANGL is the angle measured in degrees CCW from any desired reference direction. ELANGL is the angle measured in degrees up from the horizon.

```

4001      FUNCTION AZGAIN(ANGL)                                0
4002      C                                                    0
4003      COMMON/AZPAT/ CUEF(4,75),NGUEF                     1
4004      COMMON /BK1/ BK1(500)                               0
4005      EQUIVALENCE (ANTANG,BK1(17))                        0
4006      DATA KULU/1/                                       0
4007      C                                                    1
4008      ANGR=ANGL-ANTANG                                       0
4009      IF (CUEF(1,1).LE.ANGR.AND.ANGR.LE.CUEF(1,NGUEF))GO TO 10 0
4010      AZGAIN=0.0                                           0
4011      RETURN                                                0
4012      C                                                    0
4013      IF (ANGR-CUEF(1,KULU))20,50,30                       0
4014      KULU=KULU-1                                           0
4015      GO TO 10                                              0
4016      IF (ANGR.LE.CUEF(1,KOLD+1))GO TO 50                 0
4017      KOLD=KOLD+1                                           0
4018      GO TO 10                                              0
4019      C                                                    0
4020      AZGAIN=ABS( (ANGR*CUEF(2,KOLD)+CUEF(3,KOLD))*AMGR+CUEF(4,KOLD) ) 0
4021      C                                                    0
4022      C                                                    0
4023      RETURN                                                0
4024      END                                                    0
4025      FUNCTION ELGAIN(ANGL)                                  1
4026      C                                                    1
4027      COMMON/ELPAT/ CUEF(4,75),NGUEF                       1
4028      COMMON /BK1/ BK1(500)                               1
4029      EQUIVALENCE (ANTANG,BK1(17))                          1
4030      DATA KULU/1/                                       1
4031      C                                                    1
4032      ANGR=ANGL-ANTANG                                       1
4033      IF (CUEF(1,1).LE.ANGR.AND.ANGR.LE.CUEF(1,NGUEF))GO TO 10 1
4034      ELGAIN=0.0                                           1
4035      RETURN                                                1
4036      C                                                    1
4037      IF (ANGR-CUEF(1,KOLD))20,50,30                       1
4038      KULU=KULU-1                                           1
4039      GO TO 10                                              1
4040      IF (ANGR.LE.CUEF(1,KOLD+1))GO TO 30                 1
4041      KOLD=KOLD+1                                           1
4042      GO TO 10                                              1
4043      C                                                    1
4044      ELGAIN=(ANGR*CUEF(2,KOLD)+CUEF(3,KOLD))*AMGR+CUEF(4,KOLD) 1
4045      C                                                    1
4046      C                                                    1
4047      RETURN                                                1
4048      END                                                    1

```

SUBROUTINE BLOCK DATA

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
BLOCK DATA	Subordinate	None

2. PURPOSE:

This non-executable module initializes the labled common BLKRND which contains the random number array used by the random number generator.

3. INPUT PARAMETERS:

None

4. CALLING SEQUENCES:

None

This module is used only at load time to initialize the labeled area.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. The numbers used to initialize the random number array were derived from tables of random numbers published by the RAND Corporation: A Million Random Digits with 100,000 Normal Deviates, the Free Press of Glencoe, New York, 1955.
- b. Flow Chart: Page 9-124
- c. Cross Reference Table: Page 9-224

3216	BLOCK DATA	890 8-11
3217	COMMON/CLKEND/ IUMY1,IRND,IAUD1,JKND,IUUM(1),NKND1(64),NKND2(64)	900
3218	DATA IUMY1,IAUD1/1,1/	910
3219	DATA NKND1/15134181997,27509864464,30323512272,50066227369,	920
3220	* 14051007193,16402190290,26306990212,1126617090,	930
3221	* 1601629773,11849273156,19404991345,66577717830,	940
3222	* 62583434137,33025570091,11012391622,1545136861,	950
3223	* 31267410086,13482139250,26483855902,2419774290,	960
3224	* 11557620695,30512809719,12630506319,17722700814,	970
3225	* 69722597022,16966280091,16243824041,16288044806,	980
3226	* 26116896402,13576004754,1118309520,29154237811,	990
3227	* 13164942096,29908963758,63564966630,24513428519,	1000
3228	* 25267307992,16416151777,32749370939,21116173576,	1010
3229	* 19291173043,20743061171,21719359574,19674491967,	1020
3230	* 1929435324,68846120356,27142309993,15621175936,	1030
3231	* 16510917113,2346150781,28275636692,10060745449,	1040
3232	* 61761066309,17600458273,1664182774,28473284721,	1050
3233	* 17160281937,79260744156,3853554466,3669753700,	1060
3234	* 16666001290,5882873659,14024711880,1101451740,	1070
3235	* 8100413807	1080
3236	DATA NKND2/1205150044,46247664059,15416720600,19301245178,	1090
3237	* 15071201490,13988647055,01739757401,07219355507,	1100
3238	* 26126765609,32320902560,19471392797,07603799917,	1110
3239	* 44786071344,29317492972,7114643693,16136711823,	1120
3240	* 29170304246,26856574010,2033500001,14001557540,	1130
3241	* 25072582240,21374670070,15676667894,30962152182,	1140
3242	* 26171064000,16114261142,19574710484,10215018400,	1150
3243	* 16110257399,18938138207,0106076697,15062718195,	1160
3244	* 02577965400,16742167695,11616643767,11174119000,	1170
3245	* 50152427161,30197467160,2130949700,33366425769,	1180
3246	* 1723741632,25322444250,2607050175,11911381553,	1190
3247	* 30224145581,07655423387,3026402591,13101028674,	1200
3248	* 30533512969,07218771539,00229536670,19191604401,	1210
3249	* 2617205420,29107116503,1639907915,009730110,	1220
3250	* 39814607225504776974,30256345160,10622789361,	1230
3251	* 1177114000,10556707007,16190207090,97790171197	1240
3252	DATA	1250

SUBROUTINE DBLKX

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
DBLKX	Subordinate	Not user referenced
SDBLKX	Subordinate	Not user referenced

2. PURPOSE:

This subroutine is used to transfer a defined block of data from one array to another array.

3. INPUT PARAMETERS:

a. DBLKX

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NST	R	I	First element of the input array to be transferred. Also, the first element of the output array to be used.
NWORD	R	I	Number of elements to be transferred.

b. SDBLKX

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NST	R	I	First element of the input array to be transferred.
NWORD	R	I	Number of elements to be transferred.
LOC	R	I	First element of the output array to be used.

4. CALLING SEQUENCES:

CALL DBLKX (NST, NWORD, X, Y)

Where X contains the Input Waveform
Y contains the Output Waveform

$Y(NST+J) = X(NST+J) \quad J=1, NWORD$

CALL SDBLKX (LOC, NST, NWORD, X,Y)

Where: X contains the Input Waveform
 Y contains the Output Waveform

$Y(LOC+J) = X(NST+J) \quad J=1, NWORD$

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- a. All input data to this subroutine is provided through the calling argument list. This subroutine is not directly available to the program user, but is used by other subprograms for data transfer.
- b. Flow Chart: Page 9-122
- c. Cross Reference Table: Page 9-223.

3190	SUBROUTINE SDBLKX(NST,NWORD,X,Y)	UC7DBL01
3195	DIMENSION X(1),Y(1)	UC7DBL02
3200	N=NST	UC7DBL03
3201	DO 10 J=1	UC7DBL04
3202	ENTRY SDBLKX(LOC,NST,NWORD,X,Y)	UC7DBL05
3203	N=LOC	UC7DBL06
3204	20 CONTINUE	UC7DBL07
3205	NSTOP=NST+NWORD-1	UC7DBL08
3206	DO 10 J=NST,NSTOP	UC7DBL09
3207	Y(J)=X(N)	UC7DBL10
3208	N=N+1	UC7DBL11
3209	10 CONTINUE	UC7DBL12
3210	RETURN	UC7DBL13
3211	END	

FUNCTION IBOOL

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Name</u>
IBOOL	N/A	-

2. PURPOSE:

This function was used when checking out RADSIM on IBM computers. It has been replaced by the standard Honeywell intrinsic function BOOL.

3. INPUT PARAMETERS: N/A

4. CALLING SEQUENCES: N/A

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. Flow Chart: Page 9-119

b. Cross Reference Table: Page 9-223

3198	SUBROUTINE DBLKX(NST,NWORD,X,Y)	UC7DBL01
3199	DIMENSION X(1),Y(1)	UC7DBL02
3200	K=NST	UC7DBL03
3201	GO TO 20	UC7DBL04
3202	ENTRY SDBLKX(LDC,NST,NWORD,X,Y)	UC7DBL05
3203	K=LDC	UC7DBL06
3204	20 CONTINUE	UC7DBL07
3205	NSTOP=NST+NWORD-1	UC7DBL08
3206	DO 10 J=NST,NSTOP	UC7DBL09
3207	Y(J)=X(K)	UC7DBL10
3208	K=K+1	UC7DBL11
3209	10 CONTINUE	UC7DBL12
3210	RETURN	UC7DBL13
3211	END	

FUNCTION IFLD

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
IFLD	N/A	-

2. PURPOSE:

This function was used when checking out RADSIM on IBM computers. It has been replaced by the standard Honeywell intrinsic function FLD.

3. INPUT PARAMETERS: N/A

4. CALLING SEQUENCES: N/A

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

a. Flow Chart: Page 9-123

b. Cross Reference Table: Page 9-224

```

3212      FUNCTION IFLD(IST,NBITS,IWORD)
3213      IFLD=FLD(IST,NBITS,IWORD)
3214      RETURN
3215      END

```

UC708L14

UC71FD01

UC71FD02

UC71FD03

UC71FD04

FUNCTION IPACK

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
IPACK	Subordinate	Not User Referenced

2. PURPOSE:

This function subprogram is used to pack two or more groups of bits into a standard computer word (36 bits).

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
ISTR	R	I	The lowest bit into which data transfer is to occur. The bit number is counted starting with the LSB. ISTR = 0 corresponds to the LSB and ISTR = 35 corresponds to the MSB.
IDATA	R	I	The data which is to be packed into the output word.
IWORD	R	I	The word into which the data is to be packed.

4. CALLING SEQUENCES:

CALL IOUT = IPACK (ISTR, IDATA, IWORD)

IOUT = IWORD $\oplus$ (IDATA*(2**ISTR)) $\oplus$ = inclusive or

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA

- The field (bit locations) of the output word which is to receive the input data must contain zeros before execution of the function.
- If the data to be placed in the output word is larger than the allocated field then either the data in a higher field will be destroyed or if it is the highest field then some of the most

significant bits of IDATA will extend past the standard word length and be lost.

c. Flow Chart: Page 9-118

d. Cross Reference Table: Page 9-223

6. THEORY OF OPERATION

The input data, IDATA, is multiplied by the power of two equivalent to the number of bit positions that it must be shifted in order to line up with its field in the output word. Once the data has been shifted to the proper position, then it is merged into the output word, IWORD, by using an inclusive OR operation.

This function is not directly available to the simulation user, but is used by subprogram within the simulation.

0100	FUNCTION IPACK(ISTP, IDATA, IWORD)	UC7IPK01
0101	ITEMP=IDATA*(2**ISTP)	UC7IPK02
0110	IPACK = IORLOR (OR(ITEMP, IWORD))	UC7IPK03
0111	RETURN	UC7IPK05
0100	END	UC7IPK06

SUBROUTINE PACK

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
PACK	SUBORDINATE	Not user referenced

2. PURPOSE:

This subroutine takes a word in integer format and converts the lower 13 bits of the word into two 8 bit ASCII characters. The ASCII characters are subsequently transmitted to a remote plotting terminal via TSS. Remote plotter accuracy assumed to be 12 bits maximum.

3. INPUT PARAMETERS:

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
IWD	R	I	Output array word location pointer.
IBIT	R	I	Bit displacement to be used in placing the characters in the output word

4. CALLING SEQUENCES:

CALL PACK (IDAT, IWD, IBIT, IARY, \$IIII)

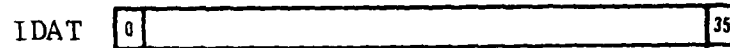
Where: IDAT is the Input word
IARY is the Output array
IIII is the statement number to which control is transferred if the output array is full (68 characters)

5. RESTRICTIONS, RECOMMENDATIONS, MISCELLANEOUS DATA

- The maximum array length is restricted to 68 characters to be compatible with TSS.
- The maximum allowable accuracy of the word is 13 bits, i.e. bits 0 through 24 must be zero.
- Flow Chart: Page 9-88
- Cross Reference Table: Page 9-219

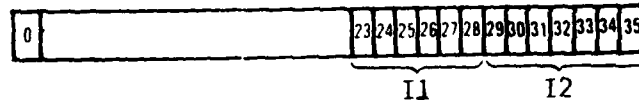
6. THEORY OF OPERATION

The input word IDAT is shown pictorially as follows:



Bit 0 is the MSB and Bit 35 is the LSB.

This word is divided into two characters, I1 and I2 as illustrated by the following:



Next I2 is tested to determine if any TSS command characters have occurred. If so, the value of I2 is incremented by +1 or -1 as follows:

<u>TSS Character</u>	<u>Actual No.</u>	<u>Increment</u>
DC1 (XON)	21 ₈	+1
CAN	30 ₈	+1
EOT	4 ₈	+1
RUBOUT	177 ₈	-1
SOH	1	-1
ETX	3	-1

Since I1 is 6 bits in length a rubout (177₈) cannot occur. Therefore, the remaining TSS characters to protect against have values which are less than 32₁₀. Accordingly, I1 is tested and if its value is less than 32₁₀ then bit 7 of I1 is set "on" to preclude occurrence of the undesired character.

Finally, I1 and I2 are packed into the output word pointed to by IWD. The parameter, IBIT, specifies the displacement of I1 from bit 0 of the output word, IARY (IWD).

8-20

```

2819      SUBROUTINE PACK(IDAT,IWD,IBIT,IARY,*)
2820      DATA IRCT/0177/,IRIT7/0100/,IXON/0217/,ICANC/0307/
2821      DATA IEUT/47/,IZ/01727/,ICONA/17/,ICUNC/37/
2822      DIMENSION IARY(1)
2823      I=IARY(IWD)
2824      J1=FLO(23,6,1EAT)
2825      J2=FLO(29,7,1DAT)
2826      IF (J1.EQ.02.AND.J1.GT.0) J1=J1+IRIT7
2827      IF (J2.EQ.1EUT.OR.J2.EQ.ICONA.OR.J2.EQ.1CUNC) J2=J2+1
2828      IF (J2.EQ.1XON.OR.J2.EQ.1CANC.OR.J2.EQ.1EUT) J2=J2+1
2829      IF (J2.EQ.1Z) J2=J2+1
2830      FLO(16,7,9,IARY(IWD))=J1
2831      IBIT=IBIT+9
2832      FLO(10,11,9,IARY(IWD))=J2
2833      IBIT=IBIT+9
2834      IF (IBIT.EQ.05) RETURN
2835      IBIT=0

```

00211700

IDENT LISTING

AUTOFLOW CHART SET - FWD/SCL RADSIM

CREF: RC

CONTENTS

```

2836      IWD=IWD+9
2837      IF (IWD.EQ.17) RETURN 1
2838      RETURN
2839      END

```


FUNCTION RRAND

1. MODULE IDENTIFICATION:

<u>Name</u>	<u>Classification Code</u>	<u>Reference Number</u>
RRAND	Subordinate	101, 102 (Parameter initialization only)

2. PURPOSE:

This function generates random numbers for use in various subprograms of the simulation model. Samples from the uniform, Gaussian and Rayleigh distributions can be generated. In addition, two statistical target models are incorporated which are based on the Chi-square distribution with 2 and 4 degrees of freedom.

3. INPUT PARAMETERS: (User supplied data through namelist NL101)

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
MAD1	O	I	The starting address for selecting random numbers from the random number array ($1 \leq \text{MAD} \leq 128$)
UMEAN	O	F	Mean value of the uniform distribution
UUEXT	O	F	Width of the uniform distribution
XMEAN	O	F	Mean value of the Gaussian distribution
SIGMA	O	F	Standard deviation of the Gaussian and Rayleigh distributions, and average cross section for target models
NTYPE	R	I	Control integer which specifies the type of distribution to be generated

<u>Name</u>	<u>O/R</u>	<u>T</u>	<u>Description</u>
NTYPE	R	I	NTYPE = 1 Uniform distribution (floating point output) = 2 Rayleigh = 3 Gaussian = 4 Uniform distribution (integer 0 to 2^{35}) = 5 Not used = 6 Swerling Target Models #1 and #2 = 7 Swerling Target Models #3 and #4 = 8 Sine
NRAND	R	I	Array containing random numbers which are used to generate the output variates

Variables initialized during execution of the initializer load module based on user supplied data. These intermediate values are generated in order to minimize execution time of the function, i.e., to avoid repeating the same calculations for each entry to RRAND.

$SIG2SQ = 2.0 * SIGMA * SIGMA$
 $UL = UMEAN - 0.5 * UUEXT$
 $UEXT = UUEXT/2 ** 35$

4. CALLING SEQUENCE:

VAR = RRAND (NTYPE)

VAR will contain the random sample generated by the function from the NTYPE probability distribution.

5. RESTRICTIONS, REQUIREMENTS, MISCELLANEOUS DATA:

- a. Before any call can be made to the function RRAND the input data must be loaded using namelist 101. Any subsequent changes in random variate distributions are also made through namelist 101.
- b. For convenience and to minimize program steps the array IRAND was equivalenced to the array NRAND but displaced by one location. This structure allows an address of zero to be used, i.e., an

address of zero will access IRAND(0) which overlays N RAND(1\). If this were not done, a test would have to be performed on MAD1 to ensure that an address of zero did not occur.

- c. A list of random numbers suitable for initializing the array N RAND are included at the end of this section.
- d. Flow Chart Page 9-120
- e. Cross Reference Table Page 9-223

6. THEORY OF OPERATION:

For each call to the function R RAND a number IRND is selected from the random number table IRAND. The address of the number selected from the table is MAD1 which is also a random number. The number IRND is added to the random number JRND which was generated by the previous execution of the function. The sign bit is set to zero to ensure a positive number. By adding the two random numbers and truncating the overflow, a new random number is generated which is also called IRND. IRND is placed in the random number table location previously occupied by the original IRND. In this manner the random number table is updated by generating new random numbers and inserting them in the table. From this random number IRND, 7 bits are selected to determine the new address MAD1 to be used in the next call to the function. The 7-bit address field allows the addresses to range from 0 to 127. The random number IRND just generated is an integer having an uniform distribution from 0 to $2^{35} - 1$. Once the random number is generated JRND is set equal to IRND for use in the subsequent executions of the function. In order to convert this number to a floating point number r having a uniform distribution from 0 to 1.0, the following conversion is used.

$$r = \text{FLOAT}(\text{IRND})/2^{35}$$

From this uniform distribution other probability distributions can be generated by using transformations which map a uniform distribution into the desired distribution. The following is a list of the transformations used in the function.

a. Uniform distribution $a \leq x \leq b$

$$x_n = (b - a)r_n + a$$

b. Rayleigh distribution $P(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$

$$x_n = \sqrt{-2\sigma^2 \ln r_n}$$

c. Gaussian distribution $P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$

$$x_n = \sqrt{2\sigma^2 \ln r_n} * \cos 2\pi r_{n+1}$$

d. Swerling Target Models #1 & #2 $P(x) = \frac{1}{\sigma} e^{-\left(\frac{x}{\sigma}\right)^2}$

$$x_n = -\bar{\sigma} \ln r_n \quad \bar{\sigma} = \text{average cross section}$$

This distribution characterizes a target with a large number of independent scatterers of approximately equal cross section (Swerling cases 1 and 2).

e. Swerling Target Models #3 & #4 $P(x) = \frac{4\sigma}{\bar{\sigma}^2} e^{-\left(\frac{2\sigma}{\bar{\sigma}}\right)^2}$

$$x_{2n} = \frac{-\bar{\sigma}}{2} (\ln r_n + \ln r_{n+1}) = \text{average cross section}$$

This distribution characterizes a target with one dominant scatterer plus smaller scatterers or one large reflector subject to small changes in orientation (Swerling Cases 3 and 4).

3137	FUNCTION KRANLINTYPE)	8-25 UC7RNY01
3138	COMMON/LEKKNL/IDMY1, IRND, MAD1, JKND, UMEAN, UOEXT, XMEAN,	UC7RNY03
3139	*SIGMA, UOM, SIG250, UL, UOEXT, NAAND(129)	UC7RNY04
3140	DIMENSION IRAND(128)	UC7RNY05
3141	DATA IREPT/07, IMULT/1220703125/	
3142	DATA IMAX/42944672967, NZF10/05530/	
3143	DATA C1/2.4103E30E-22/	UC7RNY07
3144	EQUIVALENCE (IFAND(1), NEAND(2))	UC7RNY08
3145	DO KEND=IRAND(MAD1)	UC7RNY09
3146	IRNDEKEND=JKND*IMULT	210
3147	IRND = FL0(1, 35, IRND)	UC7RNY11
3148	IRAND(MAD1)=IRND	UC7RNY12
3149	MAD1 = FL0(12, 7, IRND)	UC7RNY13
3150	IF (IDMY1.EQ.1) WRITE (6,1000) IFND, IRND, MAD1, MAD1, KEND, KEND,	
3151	* JKND, JKND	
3152	1000 FORMAT(1H * 4(115,015))	
3153	DO 10 1200, 300, 400, 500, 600, 700, 800, 900, N1=PI	250
3154	200 KEND = FL04(1RND)*UOEXT + UL	UC7RNY15
3155	JKND = IRND	UC7RNY16
3156	RETURN	UC7RNY17
3157	300 KEND = SORTESIG250*ALEC(FL04(1RND)*UL1)	UC7RNY18
3158	JKND = IRND	UC7RNY19
3159	RETURN	UC7RNY20
3160	900 CONTINUE	320

8-25a

3161	400 I1=FLU(11,17,KRND)-N7P16	
3162	I2=FLU(18,17,KRND)-N7P16	
3163	I3=I1*I1+I2*I2	350
3164	IF (I3.LT.1MAX) GO TO 20	
3165	JRND=IRND	
3166	GO TO 10	
3167	20 S=1.0/FLUAT(15)	
3168	VCCS=S*FLUAT(11*I1-I2*I2)	380
3169	VSINE=S*2.0*FLUAT(11*I2)	390
3170	IF (TYPECCS.5) GO TO 405	400
3171	KRANI=VCCS	410
3172	DUM=VSINE	420
3173	JRND=IRND	430
3174	RETURN	440
3175	405 DUM=SKT(SIG256*ALUG(FLOAT(JRND)*C1))	480
3176	KRANI=DUM*VCCS + XMEAN	UC7RNY38
3177	DUM=DUM*VSINE + XMEAN	UC7RNY39
3178	JRND=IRND	UC7RNY40
3179	RETURN	UC7RNY41
3180	500 KRANI=60UL(IRND)	UC7RNY42
3181	JRND=IRND	UC7RNY43
3182	RETURN	UC7RNY44
3183	600 KRANI=DUM	UC7RNY45
3184	RETURN	UC7RNY46
3185	700 KRANI=-SIGMA*ALUG(FLOAT(IRND)*C1)	UC7RNY47
3186	JRND=IRND	UC7RNY48
3187	RETURN	UC7RNY49
3188	800 IF (IREPT.EQ.1) GO TO 801	UC7RNY50
3189	SUM=0.0	UC7RNY51
3190	801 SUM=SUM+ALUG(FLOAT(IRND)*C1)	UC7RNY52
3191	IREPT=IREPT+1	UC7RNY53
3192	IF (IREPT.NE.2) GO TO 10	UC7RNY54
3193	IREPT=0	UC7RNY55
3194	KRANI=-SIGMA*SUM/2.0	UC7RNY56
3195	JRND=IRND	UC7RNY57
3196	RETURN	UC7RNY58
3197	END	UC7RNY59

LIST OF RANDOM NUMBERS

The following list of random numbers is suitable for initializing the random number array NRAND used by the function subprogram RRAND. The numbers were derived from tables of random numbers published by RAND Corporation: A Million Random Digits with 100,000 Normal Deviates, the Free Press of Glencoe, New York, 1955.

*NL101 NRAND= 12068158044,06847664659,15416782760,19382343178,
33122308420,25052201840,13588647055,01734737408,07289355507,
16534467415,24386072834,29317493572,07114843643,16232718423,
08160418880,16410917813,23416520791,28825638452,10800745449,
29107616508,23120785669,32320902560,15471392797,07683759917,
03665736170,29170504246,26866574818,20335880812,14861357546,
03491463822,25072568248,31374670078,13676667551,30463132192,
05804776974,20172084006,16184261842,14974210467,10283018420,
30256545185,18310257399,18938188207,01286074657,19662214195,
10832795361,01577045480,16742867655,11686848767,18174114680,
18174114680,30892487160,30892487160,28360549700,33368415709,
10556707007,17235921632,25322444850,30007056175,13488881553,
10140208896,30224148581,07655423387,32626402551,13101024674,
09779017119,30533512969,07218771539,00229536870,29198604401,
01702686304,15134181997,27509664464,30323512272,30068227398,
17006458873,14051007893,16402190290,26306590212,11260717646,
16841482774,16801629773,11349273156,19404991345,06977712830,
26473264721,02883434137,33025570091,11012391622,13431365861,
17160292937,31267410036,13462139250,26463885902,24215774296,
29620744156,11557820695,30512809719,12630506319,17722780814,
08883554436,04722557022,16500280091,16242824041,16388044606,
03669953728,26212698408,13570094754,11188309528,29134237821,
16068801392,13164942056,29908968258,03564586686,24513426529,
05883873859,25262307992,16416251777,32749370939,21116178576,
14824731880,19395173043,20743061171,21319359579,19074491967,
18081451748,19244390324,08846123356,27142309954,15825176938,

S E C T I O N 9

R A D S I M C O M P U T E R P R O G R A M

F L O W C H A R T S

PUBLISHED IN PART 2 OF THIS VOLUME.

SECTION 10

INDEX

(See also Section 6 in Volume I, Part 1
and Appendix B in Volume IV, Part 1)

<u>PROGRAM/ SUBROUTINE/ FUNCTION</u>	<u>MODULE NUMBERS</u>	<u>DESCRIPTION & LISTING PAGE</u>	<u>FLOW CHART PAGE</u>	<u>CROSS REF. PAGE</u>
ABORT	-	8-2	9-81	9-218
ADDA	207/239	See CONV		
ADDRND	235/236	See RNDARY		
ADRND	237	See RNDARY		
ANTARY	413	Vol III	9-170	9-230
ANTINT	-	8-4	9-140	9-226
ANTPAT	504	See ANTARY		
ATOD	216/217	4-2	9-98	9-220
AZGAIN	-	8-8	9-159	9-229
BLOCK DATA	-	8-10	9-124	9-224
CDFNCL	404	See CDIGFL		
CDIGFL	403	4-6	9-96	9-220
CFAR	459,460	4-10	9-66	9-216
CGEN	425	Vol III	9-182	9-231
CGENCW	455	See CGENSF		
CGENSF	453	Vol III	9-149	9-227
CLINT	302	6-2	9-198	9-234
CLUTTR	503	4-14	9-112	9-222
CONV	204	5-2	9-60	9-216
CONVMP	205	See CONV		
CUMDIS	208/209	6-11	9-152	9-228
CUM 2	-	See CUMDIS		
DBLKX	114/115	8-12	9-122	9-223
DCFAR	440/441	4-20	9-93	9-220
DFT	201	4-24	9-142	9-226
DFTFØ	234	See DFT		
DFTRF	233	See DFT		
DIGFIL	461	4-30	9-101	9-221
DIGFNC	462	See DIGFIL		
DIGFSF	463	4-34	9-107	9-222
DIGTFL	422/423	4-41	9-95	9-220
DIVA	206	See CONV		
DTOA	219/ 220/228	See ATOD		

<u>PROGRAM/ SUBROUTINE/ FUNCTION</u>	<u>MODULE NUMBERS</u>	<u>DESCRIPTION & LISTING PAGE</u>	<u>FLOW CHART PAGE</u>	<u>CROSS REF. PAGE</u>
ECM	512	4-45	9-104	9-221
ECMFL	-	See DIGFIL		
ECMFSU	-	See DIGFSF		
ELGAIN	-	8-8	9-160	9-229
ERGYCP	118	6-16	9-70	9-217
ERGYRE	116/117	See ERGYCP		
ERRMSG	-	See ABORT		
FGENMP	421	See FGENXY		
FGENXY	420	4-51	9-72	9-217
FILT	407/408	4-60	9-55	9-215
FWDET	416/417	See HWDET		
HET	-	4-64	9-103	9-221
HLIM	438/439	See HWDET		
HWDET	414/415	4-65	9-137	9-225
IBOOL	-	8-14	9-119	9-223
IFLD	-	8-15	9-123	9-224
IFT	-	See ZFFT		
IFWDET	447/448	See HWDET		
IHLIM	442/443	See HWDET		
IHWDET	445/446	See HWDET		
INGTOR	409/410	4-70	9-97	9-220
IONOS	511	4-74	9-117	9-223
IPACK	-	8-16	9-118	9-223
ISQDET	449/450	See HWDET		
LAMPCP	458	4-80	9-68	9-217
LAMPRE	456/457	See LAMPCP		
MAIN 1	-	2-1	9-2	9-206
MAIN 2	-	2-2	9-34	9-210
MTIFLT	430/431	4-83	9-201	9-234
MTIIFT	-	See MTIFLT		
MTIINC	-	See MTIFLT		
MTINCL	432/433	See MTIFLT		
NCSWPI	436/437	See SWPINT		
NONLIN	401/402	4-91	9-135	9-225
OUTCUM	210/211	See CUMDIS		
PACK	-	8-18	9-88	9-219
PDF	212/213	See CUMDIS		

<u>PROGRAM/ SUBROUTINE/ FUNCTION</u>	<u>MODULE NUMBERS</u>	<u>DESCRIPTION & LISTING PAGE</u>	<u>FLOW CHART PAGE</u>	<u>CROSS REF. PAGE</u>
PHDEC	508/509	4-94	9-78	9-218
PHENC	506/507	4-100	9-167	9-229
PLOTTR	307-310	6-18	9-90	9-219
PLTFMT	113	6-23	9-175	9-231
PTLIST	303-306	6-28	9-82	9-218
PXFRM	454	Vol III	9-147	9-227
RDIGFL	405	4-105	9-100	9-221
RECF	451	Vol III	9-145	9-227
RECFTF	452	See RECF		
RNDARY	214/215	4-109	9-150	9-227
RRAND	101/102	8-21	9-120	9-223
RSHFTS	231/232	See SHIFT		
RSHIFT	229/230	See SHIFT		
RTOPDB	103	6-35	9-156	9-228
RTOPM	110	See RTOPDB		
RTOPM2	111	See RTOPDB		
SCANNR	313	6-40	9-161	9-229
SDBLKX	-	See DBLKX		
SHIFT	224/225	4-113	9-63	9-216
SHIFTS	226/227	See SHIFT		
SPCAVG	312	Vol III	9-165	9-229
SQDET	418/419	See HWDET		
SWPINT	434/435	4-118	9-204	9-234
TARGET	501	4-122	9-109	9-222
TGTNCL	502	See TARGET		
TSARM	-	See TSARY		
TSARY	426	Vol III	9-186	9-232
TSARY 1	427	See TSARY		
TSARY 2	-	See TSARY		
TSRPAT	505	Vol III	9-178	9-231
WEITCP	222	See WEITRE		
WEITMP	223	See WEITRE		
WEITRE	221	4-129	9-57	9-215
WVGUID	510	4-135	9-116	9-223
XYTODB	104/108	See RTOPDB		
XYTOM	105	See RTOPDB		
XYTOM2	106	See RTOPDB		

<u>PROGRAM/ SUBROUTINE FUNCTION</u>	<u>MODULE NUMBERS</u>	<u>DESCRIPTION & LISTING PAGE</u>	<u>FLOW CHART PAGE</u>	<u>CROSS REF. PAGE</u>
ZFFT	202	5-7	9-125	9-224
ZIFFT	203	See ZFFT		

SECTION 11
RECURSIVE COMPUTATION
OF SINE / COSINE PAIRS

Many computer programs require the evaluation of sine and cosine functions for angles that are uniformly spaced over the interval from 0° to 360° , for example, DFT subroutine. In many cases the CPU time required to execute these computer programs can be significantly reduced by the use of the following digital oscillator to generate sine/cosine pairs. The Z-plane representation of the digital oscillator is shown in Figure 11-1.

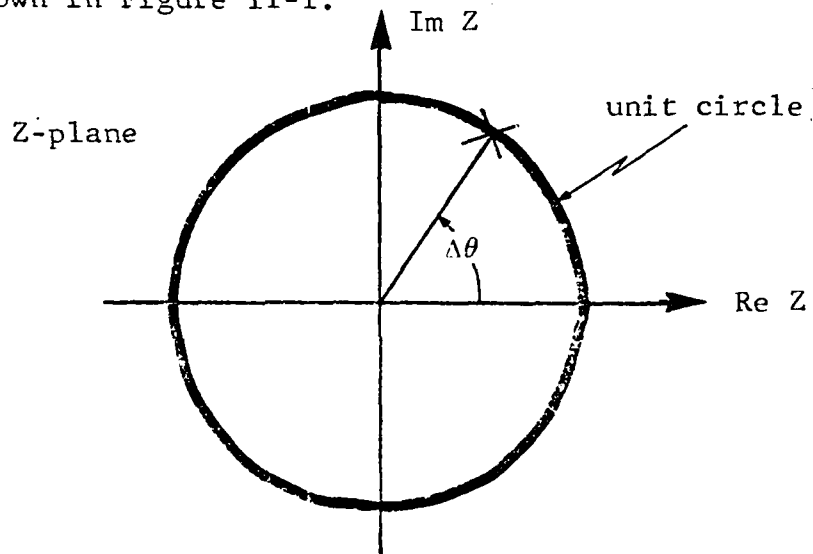


Figure 11-1 Z-PLANE REPRESENTATION OF THE DIGITAL OSCILLATOR

The variable $\Delta\theta$ is the angular increment for which the sine/cosine pairs are to be calculated. The digital oscillator block diagram is shown in Figure 11-2.

The stimulus $[\cos \theta_0 + j \sin \theta_0]$ is applied only when $n=0$ and is zero otherwise. Before execution of the digital oscillator routine, the storage registers (A & B) used to generate the unit delays must be cleared.

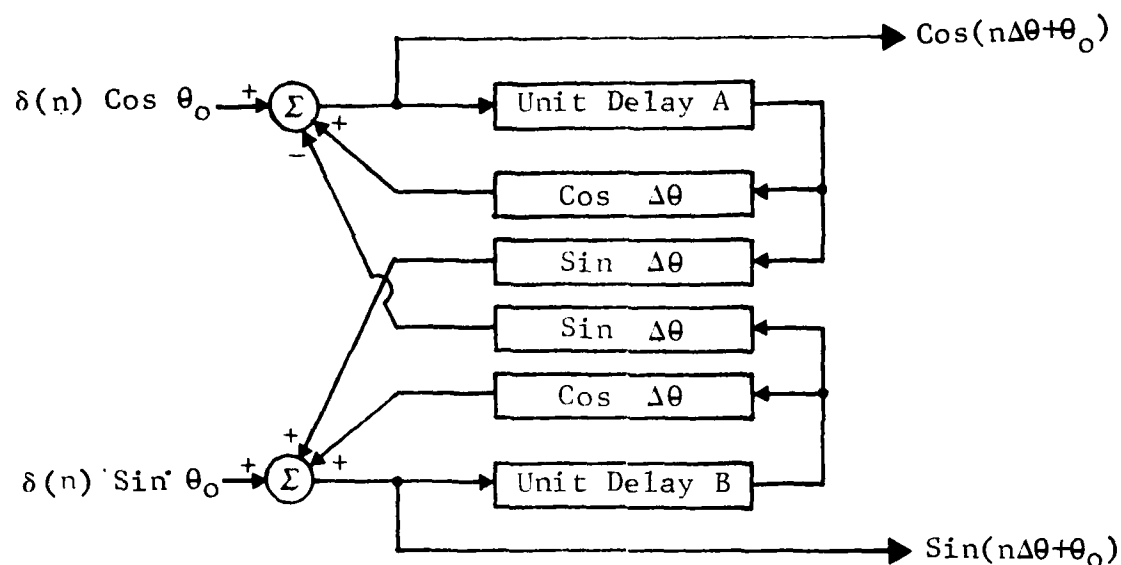


Figure 11-2 DIGITAL OSCILLATOR BLOCK DIAGRAM

n = number of executions of the digital oscillator routine.

The Fortran statements for implementing the digital oscillator are the following:

```

      A = Cos (θ₀)
      B = Sin (θ₀)
      DELC = Cos (Δθ)
      DELS = Sin (Δθ)

      Loop
      n=1, N
      { A and B contain the
        sine/cosine of [θ₀+(n-1)Δθ] }
      TEMP = A
      A = A * DELC - B * DELS
      B = TEMP * DELS + B * DELC
  
```

where $\Delta\theta$ = angle increment,
 θ_0 = starting angle.

*MISSION
of
Rome Air Development Center*

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C³) activities, and in the C³ areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

