

100
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-513

*Program Listing for the Reliability Block
Diagram Computation Program of
JPL Technical Report 32-1543*

Paul O. Chelson

R. Eric Eckstein

(NASA-CR-125541) PROGRAM LISTING FOR THE
RELIABILITY BLOCK DIAGRAM COMPUTATION
PROGRAM OF JPL TECHNICAL REPORT 32-1543
P.O. Chelson, et al (Jet Propulsion Lab.)
1 Dec. 1971 41 p

FACI (NASA CR OR TMX OR AD NUMBER)

(CATEGORY)

CSCL 14D G3/15

N72-16332

Unclas

14224

RECEIVED
NASA STI FACILITY
INPUT BRANCH

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

December 1, 1971

40p.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-513

*Program Listing for the Reliability Block
Diagram Computation Program of
JPL Technical Report 32-1543*

Paul O. Chelson

R. Eric Eckstein

**JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA**

December 1, 1971

Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

PRECEDING PAGE BLANK NOT FILMED

PREFACE

The work described in this report was performed by the Quality Assurance and Reliability Division of the Jet Propulsion Laboratory.

CONTENTS

I.	Introduction	1
II.	Main Program MAIN	2
	A. Subroutine SFLEFT	16
	B. Subroutine DUPOUT	17
	C. Subroutine PATHUP	18
	D. Subroutine PATHOT	19
	E. Subroutine NUMOUT	20
	F. Subroutine NEGOUT	21
	G. Subroutine TRINTR	22
	H. Subroutine PATHIN	25
	I. Subroutine TRPRNT	26
	J. Subroutine TREE	27
	K. Subroutine SYSP	31
	L. Subroutine STBYPR	33
	M. Subroutine DUPTRE	34
	N. Subroutine STDBY2	35

ABSTRACT

This technical memorandum presents the computer program listing for the reliability block diagram computation program described in Technical Report 32-1543, Reliability Computation From Reliability Block Diagrams, Jet Propulsion Laboratory, Pasadena, Calif., Dec. 1, 1971. The program is written in FORTRAN V and is currently running on a UNIVAC 1108. Each subroutine contains a description of its function.

I. INTRODUCTION

This technical memorandum presents the computer program listing for the reliability block diagram computation program described in Technical Report 32-1543, Reliability Computation From Reliability Block Diagrams, Jet Propulsion Laboratory, Pasadena, Calif., Dec. 1, 1971. The program is written in FORTRAN V and is currently running on a UNIVAC 1108. Each subroutine contains a description of its function.

TREA*FTREE.MAIN

```

1      C                      SYSTEM RELIABILITY PROGRAM                      MAIN 10
2      C                                                                MAIN 20
3      C                                                                MAIN 30
4      C                                                                MAIN 40
5      C                                                                MAIN 50
6      C    UNIVAC 1102, FORTRAN V VERSION, 65K CORE STORAGE REQUIRED.        MAIN 60
7      C    WRITTEN BY PAUL CHELSON AND RON ECKSTEIN 1/1/70                  MAIN 70
8      C    LATEST MODIFICATION 7/12/71                                     MAIN 75
9      C                                                                MAIN 80
10     C    *****MAIN 90
11     C                                                                MAIN 100
12     C    DATA DECK SETUP.                                               MAIN 110
13     C                                                                MAIN 120
14     C    ALL INTEGER FORMATS RIGHT JUSTIFIED. COLUMN NUMBERS APPEAR INSIDE MAIN 130
15     C    PARENTHESIS ( ). ANY SERIES OF BLOCK NUMBERS STARTS AT THE LEFT MAIN 140
16     C    MOST COLUMN AN FILLS IN SUCCESSIVELY TO THE RIGHT IN I2 FORMAT.  MAIN 150
17     C    ONE DATA CARD PER BLOCK NUMBER(U.O.S.).                       MAIN 160
18     C                                                                MAIN 170
19     C    BLOCK NUMBER(1-2), INPUT TO THAT BLOCK NUMBER(3-32), OUTPUT FROM MAIN 180
20     C    THAT BLOCK NUMBER(33-63).                                       MAIN 190
21     C    SUCCESS BLOCK NUMBER WITH ZERO OUTPUT HAS A 9 IN (80) AND IS LAST MAIN 200
22     C    CARD IN THIS SERIES.                                           MAIN 210
23     C                                                                MAIN 220
24     C    NUMBER OF SENSE BLOCKS(1-2).                                    MAIN 230
25     C    IF THERE ARE SENSE BLOCKS, SENSE BLOCK NUMBER(1-2), STANDBY BLOCKS MAIN 240
26     C    CONTROLLED BY THAT SENSE BLOCK(3-60).                          MAIN 250
27     C                                                                MAIN 260
28     C    NUMBER OF EQUIVALENT BLOCK SETS(1-2).                          MAIN 270
29     C    EQUIVALENT BLOCKS MAY NOT BE CONTROLLED BY A SENSE BLOCK(SWITCH). MAIN 280
30     C    IF THERE ARE EQUIVALENT BLOCKS, EQUIVALENT BLOCK NUMBERS(1-40) FCR MAIN 290
31     C    EACH SET.                                                       MAIN 300
32     C                                                                MAIN 310
33     C    DISTRIBUTION TYPE(1-2), MISSION TIME(3-14) E FORMAT. -         MAIN 320
34     C    1 = EXPONENTIAL, 2 = NOT AVAILAELE.                             MAIN 330
35     C                                                                MAIN 340
36     C    ACTIVE PARAMETERS FOR EACH BLOCK OTHER THAN SENSE BLOCKS(SWITCHES) MAIN 350
37     C    RC IS SET EQUAL TO 1.0 IF(15-25) LEFT BLANK.                    MAIN 360
38     C    BLOCCK NUMBER(1-2), LAMBDA(3-14) E FORMAT, RC(15-25) F FORMAT.  MAIN 370
39     C    LAST CARD IN THIS SERIES HAS LAST(80) SET EQUAL TO 5.          MAIN 380
40     C                                                                MAIN 390
41     C    IF NO DORMANCY INVOLVED, A BLANK CARD, OTHERWISE -             MAIN 400
42     C    BLANK(1-2), DORMANCY FACTOR(3-14) E FORMAT. TO BE MULTIPLIED BY MAIN 410
43     C    ACTIVE LAMBDA'S OR 99(1-2), BLANK(3-80) AND READ INDIVIDUAL DORMANT MAIN 420
44     C    LAMBDA'S - BLOCCK NUMBER(1-2), LAMBDA DORMANT(3-14) E FORMAT.  MAIN 430
45     C    LAST CARD IN THIS SERIES HAS LAST(80) SET EQUAL TO 6.          MAIN 440
46     C                                                                MAIN 450
47     C                                                                MAIN 460
48     C    SWITCHING OPTIONS FOR EACH SENSE BLOCK.                        MAIN 470
49     C                                                                MAIN 480
50     C    0 = NO SWITCHING( PROBABILITY OF SWITCH WORKING = 1.0).        MAIN 490
51     C    SENSE BLOCK(1-2).                                              MAIN 500
52     C                                                                MAIN 510
53     C    1 = CONSTANT PROBABILITY THAT SWITCH WORKS.                    MAIN 520
54     C    SENSE BLOCK(1-2),BLANK(3-14),PROBABILITY(15-25) F FORMAT, 1(80). MAIN 530
55     C                                                                MAIN 540
56     C    2 = DORMANT FAILURE RATE FOR SWITCH.                           MAIN 550
57     C    SENSE BLOCK(1-2),LAMBDA DORMANT(3-14) E FORMAT, 2(80).        MAIN 560
58     C                                                                MAIN 570
59     C    3 = DORMANT AND ACTIVE FAILURE RATE FOR SWITCH( 2 CARDS/SWITCH). MAIN 580
60     C    SENSE BLOCK(1-2),LAMBDA DORMANT(3-14) E FORMAT, 3(80).        MAIN 590
61     C    SENSE BLOCK(1-2),LAMBDA ACTIVE(3-14).                         MAIN 600
62     C                                                                MAIN 610
63     C    4 = NOT AVAILABLE.                                             MAIN 620
64     C    5 = NOT AVAILABLE.                                             MAIN 630
65     C                                                                MAIN 640
66     C    LAST CARD IN DATA DECK HAS LAST(80) SET EQUAL TO 7, 8, OR 9 -  MAIN 650
67     C    7 = RECALCULATE WITH NEW PARAMETERS, 8 = NEW DIAGRAM, 9 = END. MAIN 660
68     C                                                                MAIN 670
69     C    *****MAIN 680
70     C                                                                MAIN 690
71     C    COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6              MAIN 700
72     C    COMMON/LGCUB/IB(15,2,50),IIS(50),IRB(50,30),ITEMP(41,20),N1,   MAIN 710
73     C    1 ISAVE(50,200),ISUC,JSUC,JS1,JSAVE(50,200),IJS1,TOT,PRC3(65),  MAIN 720
74     C    2 FRCB1(65),IFRINT,NSTDBY(15),NSBYMX,NOPRNT                    MAIN 730
75     C    COMMON/SMSUB/NPUP(200),NPUPMX,NBOUT(50),NBOTMX,NBIN(50),NBINMX, MAIN 740
```



```

76      1 NBNUM(50),NENMAX                                MAIN 750
77      DIMENSION RO(50),NP(50),RT(65,101), PSNS(101),IS(50),TL(50)    MAIN 760
78      DIMENSION IN(15),IGUT(15),JRB(30),NRB(30),KRB(30),KSAVE(50,200)  MAIN 770
79      DIMENSION LSAVE(50,200), TLD(50),NHOLD(15)                MAIN 780
80      DIMENSION SWPROB(50),TLDS(50),TLS(50)                    MAIN 790
81      DOUBLE PRECISION DPREC,DPREC1,DPREC2,DPREC3,DPREC4,DPREC5    MAIN 800
82      DOUBLE PRECISION PINT,YAVE,PSNS,TOT,PROB,PROB1            MAIN 810
83      C                                                         MAIN 820
84      C                                                         MAIN 830
85      C                                                         MAIN 840
86      M1 = 50 @MAXIMUM NUMBER OF BLOCKS IN DIAGRAM.           MAIN 850
87      M2 = 200 @MAXIMUM NUMBER OF SUCCESS PATHS.              MAIN 860
88      M3 = 14 @MAXIMUM NUMBER OF INPUTS/OUTPUTS TO/FROM ONE BLOCK.  MAIN 870
89      M4 = 15 @MAXIMUM NUMBER OF SENSE BLOCKS CONTROLLING STANDBY.  MAIN 880
90      M5 = 29 @MAXIMUM NUMBER OF STANDBY BLOCKS CONTROLLED BY 1 SENSOR. MAIN 890
91      M6 = 20 @MAXIMUM NUMBER OF EQUIVALENT BLOCKS IN A SINGLE SET.  MAIN 900
92      M7 = 20 @MAXIMUM NUMBER OF EQUIVALENT BLOCK SETS.        MAIN 910
93      L1=M1+1                                                  MAIN 920
94      L3=M3+1                                                  MAIN 930
95      L4=M4+M1                                                 MAIN 940
96      L5=M5+1                                                  MAIN 950
97      L6=(2*M6)+1                                             MAIN 960
98      L16=M6+1                                                MAIN 970
99      C                                                         MAIN 980
100     C                                                         MAIN 990
101     C   THE FOLLOWING SHOWS THE RELATIONSHIP OF THE ABOVE TO STORAGE.  MAIN1000
102     C                                                         MAIN1010
103     C   COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6     MAIN1020
104     C   COMMON/LOSSUB/IB(L3,2,M1),IIS(M1),IRE(M1,L5),ITEMP(L6,M7),N1,  MAIN1030
105     C   1 ISAVE(M1,M2),ISUC,JSUC,JS1,JSAVE(M1,M2),IJS1,TOT,PROB(L4),  MAIN1040
106     C   2 PROE1(L4),IFPRINT,NSTDBY(M4),NSBYMX                MAIN1050
107     C   COMMON/SMSUB/NPUP(M2),NPUPMX,NBOUT(M1),NBOTHX,NBIN(M1),NBINMX,  MAIN1060
108     C   1 NBNUM(M1),NENMAX                                    MAIN1070
109     C                                                         MAIN1080
110     C   NST = NSTEP + 1                                       MAIN1090
111     C   DIMENSION RO(M1),NP(M1),RT(L4,NST), PSNS(NST),IS(M1),TL(M1)  MAIN1100
112     C   DIMENSION IN(M1),IGUT(M1),JRB(L5),NRB(L5),KRB(L5),KSAVE(M1,M2)  MAIN1110
113     C   DIMENSION LSAVE(M1,M2),IDR(M1),TLD(M1),NHOLD(L3)    MAIN1120
114     C   DIMENSION SWPROB(M1),TLDS(M1),TLS(M1)                MAIN1130
115     C                                                         MAIN1131
116     C                                                         MAIN1132
117     C   THE VARIABLE NSIG SPECIFIES THE NUMBER OF *SIGNIFICANT* FIGURES  MAIN1133
118     C   TO BE PRINTED FOR THE RELIABILITIES. NOTE THAT *SIGNIFICANT*  MAIN1134
119     C   FIGURES IS DEFINED AS THE NUMBER OF NON-NINES IN THE RELIABILITY  MAIN1135
120     C   NUMBER. THUS, .99966, .985, AND .56 ALL HAVE TWO *SIGNIFICANT*  MAIN1136
121     C   DIGITS.                                               MAIN1137
122     C                                                         MAIN1138
123     C   NSIG=3                                                MAIN1139
124     C                                                         MAIN1140
125     C   THE ARRAY V HOLDS THE FORMAT FOR THE FINAL RELIABILITY PRINTING.  MAIN1141
126     C   THE ARRAY VK HOLDS THE ALPHA NUMBERS NEEDED TO CHANGE THE  MAIN1142
127     C   VARIABLE FORMAT V.                                    MAIN1143
128     C                                                         MAIN1144
129     C   DIMENSION V(13),VK(10)                                MAIN1145
130     C                                                         MAIN1146
131     C   DATA V/'(36HCR','ELIABI','LITY O','F THE ','SYSTEM',' THRU ','  MAIN1147
132     C   1'TIME','F10.C','9H HCU','RS = ','F','10','.','8','')'/  MAIN1148
133     C   DATA VK/'1','2','3','4','5','6','7','8','9','10'/'  MAIN1149
134     C                                                         MAIN1150
135     C   CLEAR VARIABLES TO ZERO AND SET CONSTANTS.           MAIN1160
136     C                                                         MAIN1170
137     C   IPRINT=0                                              MAIN1175
138     C   999 LAST=0                                             MAIN1180
139     C   NOPRNT=1                                              MAIN1190
140     C   DO 15 I=1,L3                                          MAIN1200
141     C   DO 19 J=1,2                                           MAIN1210
142     C   DO 19 K=1,M1                                          MAIN1220
143     C   19 IB(I,J,K)=0                                         MAIN1230
144     C   DO 6 I=1,M1                                           MAIN1240
145     C   IS(I)=0                                               MAIN1250
146     C   IIS(I)=0                                              MAIN1260
147     C   DO 6 K=1,L5                                           MAIN1270
148     C   6 IRB(I,K)=0                                          MAIN1280
149     C                                                         MAIN1290
150     C   997 REWIND 10                                         MAIN1300
151     C   NSBYMX=0                                             MAIN1320

```

```

152          KT=0                                MAIN133C
153          NBHOLD=0                             MAIN1340
154          NSTEP=100                            MAIN1350
155          JJJ=0                                MAIN1360
156          KRB1=1                               MAIN1370
157          NRBI=0                               MAIN1380
158          DO 10 I=1,L4                         MAIN1390
159          PROB(I)=0.00+0                       MAIN1400
160          PROBI(I)=C.0L+0                      MAIN1410
161          DO 10 L=1,L01                        MAIN1420
162          RT(I,L)=0.                           MAIN143C
163          10 PSNS(L)=C.0D+0                    MAIN1440
164          DO 15 I=1,M4                         MAIN1450
165          15 NSTDEY(I)=0                       MAIN1460
166          DO 16 I=1,L3                        MAIN1470
167          IN(I)=0                              MAIN1480
168          16 IOUT(I)=0                         MAIN1490
169          DO 20 I=1,M1                        MAIN1500
170          SWPRCB(I)=1.0                       MAIN1510
171          TLDS(I)=0.                          MAIN1520
172          TLS(I)=0.                           MAIN1530
173          TL(I)=0.                            MAIN1540
174          TLD(I)=0.                          MAIN1550
175          RD(I)=0.                            MAIN1560
176          NP(I)=0                             MAIN1570
177          DO 17 K=1,L5                        MAIN1580
178          NRB(K)=0                            MAIN1590
179          17 KRB(K)=0                         MAIN1600
180          DO 20 K=1,M2                        MAIN1610
181          JSAVE(I,K)=0                       MAIN1620
182          20 ISAVE(I,K)=0                    MAIN1630
183          C                                     MAIN1640
184          IF(LAST.EQ.7) GO TO 3000           MAIN1650
185          C                                     MAIN1660
186          C   READ INPUTS AND OUTPUTS TO BLOCK NUMBERS. MAIN1670
187          C                                     MAIN1680
188          1099 READ (5,1200,ERR=300,END=300) NB,IN,IOUT,LAST MAIN1690
189          1200 FORMAT(31I2,17X,I1)           MAIN1700
190          C                                     MAIN1710
191          C   PRINT DATA CARDS.              MAIN1720
192          C                                     MAIN1730
193          IF(IPRINT.NE.4) GO TO 1210          MAIN1740
194          PRINT 1202, NB,IN,IOUT,LAST        MAIN1750
195          1202 FORMAT(' DATA CARD'31I3,' LAST='I2) MAIN1760
196          C                                     MAIN1770
197          C   NBHOLD = TEMPORARY HOLD FOR THE HIGHEST NUMBER BLOCK(NBMAX). MAIN1780
198          C                                     MAIN1790
199          1210 IF(NBHOLD-NB)1020,1021,1022    MAIN1800
200          1020 NBHOLD=NB                     MAIN1810
201          GO TO 1022                          MAIN1820
202          1021 PRINT 1023, NB                MAIN1830
203          1023 FORMAT('!ERROR IN MAIN PROGRAM AT 1023.'/'OMORE THAN ONE BLOCK NUMMAIN1840
204          1024 1BER ='I4)                   MAIN1850
205          GO TO 1056                          MAIN1860
206          C                                     MAIN1861
207          C   REREAD CARD IF IPRINT IS BEING SPECIFIED MAIN1862
208          C                                     MAIN1863
209          300 READ(0,301,ERR=302,END=302) I,J MAIN1864
210          301 FORMAT(7X,I1,6X,I1)           MAIN1865
211          IF((I.GT.4).OR.(J.GT.3)) GO TO 302 MAIN1866
212          IPRINT=I                           MAIN1867
213          IF(J.EQ.0) J=3
214          NSIG=J
215          PRINT 303,IPRINT,NSIG
216          303 FORMAT('0IPRINT OPTIGN SET AT'I4,' NSIG OPTIGN SET AT'I4/1H1) MAIN1868
217          GO TO 1099                          MAIN1870
218          302 PRINT 304                       MAIN1871
219          304 FORMAT('0GERROR** ATTEMPT TO SPECIFY IPRINT OR NSIG, BUT SPECIFIED MAIN1872
220          1024 OF RANGE.'/' CARD IGNORED')
221          GO TO 1099                          MAIN1873
222          C                                     MAIN1874
223          C   GENERATE ARRAY IB WITH IB(1,1,NB) = NUMBER OF INPUTS TO NB, MAIN1875
224          C   IB(1,2,NB) = NUMBER OF OUTPUTS FROM NB, MAIN1880
225          C   THE REST OF THE IB ARRAY CONSISTING OF INPUTS AND OUTPUTS TO NB. MAIN1890
226          C                                     MAIN1900
227          1022 DO 1010 I=1,L3                MAIN1910

```

```

228      IF(IN(I).EQ.0) GO TO 1011          MAIN1930
229      1010 CONTINUE                     MAIN1940
230      C                                 MAIN1950
231      1011 IN1=I-1                       MAIN1960
232      DO 1012 I=1,L3                   MAIN1970
233      IF(ICUT(I).EQ.0) GO TO 1013      MAIN1980
234      1012 CONTINUE                     MAIN1990
235      1013 ICUT1=I-1                    MAIN2000
236      IB(1,1,NB)=IN1                    MAIN2010
237      IB(1,2,NB)=ICUT1                  MAIN2020
238      DO 1014 I=1,IN1                    MAIN2030
239      1014 IB(I+1,1,NB)=IN(I)            MAIN2040
240      DC 1015 I=1,ICUT1                  MAIN2050
241      1015 IB(I+1,2,NB)=ICUT(I)         MAIN2060
242      IF(LAST.NE.9) GO TO 1099          MAIN2070
243      C                                 MAIN2080
244      C CHECK SUCCESS BLOCK OUTPUT = ZERO. MAIN2090
245      C                                 MAIN2100
246      DO 1025 I=1,L3                       MAIN2110
247      IF(ICUT(I).EQ.0) GO TO 1025      MAIN2120
248      PRINT 1024, ICUT(I)                MAIN2130
249      1024 FORMAT('1ERRGR IN MAIN PROGRAM AT 1024.'/'0SUCCESS BLOCK HAS OUTPUT MAIN2140
250      IT ='I4')                            MAIN2150
251      GO TO 1066                           MAIN2160
252      1025 CONTINUE                         MAIN2170
253      C                                 MAIN2180
254      C ISUC = SUCCESS BLOCK NUMBER(BASE OF PROBABILITY TREE TO BE GENER- MAIN2190
255      C ATED BY CALLING SUBROUTINE TREE). MAIN2200
256      C                                 MAIN2210
257      1017 ISUC=NB                           MAIN2220
258      C JSUC AND IPRINT ARE USED IN SUBROUTINE TREE TO CONTROL PRINTING. MAIN2230
259      C                                 MAIN2240
260      JSUC=NB                               MAIN2250
261      C                                 MAIN2260
262      C NEMAX = THE HIGHEST BLOCK NUMBER OF BLOCKS 1-50/1-M1. MAIN2270
263      C                                 MAIN2280
264      NEMAX=NBHOLD                           MAIN2290
265      C                                 MAIN2300
266      C PRINT RELIABILITY BLOCK DIAGRAM. MAIN2310
267      C                                 MAIN2320
268      PRINT 632                             MAIN2330
269      632 FORMAT('1RELIABILITY BLOCK DIAGRAM') MAIN2340
270      DO 18 I=1,M1                           MAIN2350
271      IF(IB(1,1,I))21,22,21                 MAIN2360
272      22 IF(IB(1,2,I))21,18,21             MAIN2370
273      21 K=IB(1,1,I)+1                       MAIN2380
274      IF(K.LE.1) GO TO 40                    MAIN2381
275      PRINT 202,I,(IB(J,1,I),J=2,K)         MAIN2382
276      202 FORMAT(1H05HBLOCKI3,3X,5HINPUT14(I3,1X)) MAIN2383
277      GO TO 41                               MAIN2384
278      40 PRINT 205,I                          MAIN2385
279      205 FORMAT(1H05HELOCKI3,3X,5HINPUT)   MAIN2386
280      41 K=IB(1,2,1)+1                       MAIN2387
281      IF(K.LE.1) GO TO 42                    MAIN2388
282      PRINT 203,(IB(J,2,I),J=2,K)           MAIN2389
283      203 FORMAT(11X,6HOUTPUT,14(I3,1X))    MAIN2390
284      GO TO 18                               MAIN2391
285      42 PRINT 206                            MAIN2392
286      206 FORMAT(11X,6HOUTPUT)              MAIN2393
287      18 CONTINUE                           MAIN2400
288      C                                 MAIN2410
289      C READ SENSE BLOCK AND STANDBY BLOCKS OF THAT SENSE BLOCK. MAIN2420
290      C                                 MAIN2430
291      C IIS ARRAY HOLDS SENSE BLOCKS OF ORIGINAL TREE. MAIN2440
292      C IS ARRAY HOLDS SENSE BLOCKS OF ORIGINAL TREE AND VARIES. MAIN2450
293      C NNSR = NUMBER OF SENSE BLOCKS IN ORIGINAL TREE. MAIN2460
294      C NSR = NUMBER OF SENSE BLOCKS IN ORIGINAL TREE AND VARIES. MAIN2470
295      C IRB ARRAY HOLDS STANDBY BLOCKS CONTROLLED BY THEIR SENSE BLOCKS. MAIN2480
296      C IRB(NUMBER OF SENSE BLOCK, 1) = MAXIMUM NUMBER OF STANDBY BLOCKS MAIN2490
297      C OF THAT SENSE BLOCK. MAIN2500
298      C IRB(NUMBER OF SENSE BLOCK, NB) = STANDBY BLOCK NUMBER CONTROLLED MAIN2510
299      C BY THAT SENSE BLOCK. MAIN2520
300      C                                 MAIN2530
301      READ 1200, NSR                          MAIN2540
302      NNSR=NSR                               MAIN2550
303      IF(NSR.EQ.0) GO TO 840                MAIN2560

```

```

304          DO 2001 I=1,NSR                                MAIN2570
305          READ 1200 NS,JRB                                MAIN2580
306          IS(I)=NS                                        MAIN2590
307          IS(I)=NS                                        MAIN2600
308          DO 2002 J=1,M5                                  MAIN2610
309          IRB(NS,J+1)=JRB(J)                             MAIN2620
310          IF (JRB(J).EQ.0) GO TO 2003                   MAIN2630
311          2002 CONTINUE                                   MAIN2640
312          2003 IRB(NS,1)=J-1                             MAIN2650
313          2001 CONTINUE                                   MAIN2660
314          DO 2020 I=1,NSR                                MAIN2670
315          INS=IS(I)                                       MAIN2680
316          C                                              MAIN2690
317          C  STANDBY BLOCKS(NHOLD) CONTROLLED BY A SENSE BLOCK(INS) MUST FOLLOW MAIN2700
318          C  ALL OTHER INPUT BLOCK NUMBERS IN IB(NB,1,INS). MAIN2710
319          C                                              MAIN2720
320          DO 2200 J=1,L3                                  MAIN2730
321          NHOLD(J)=0                                       MAIN2740
322          2200 CONTINUE                                   MAIN2750
323          NUM=C                                           MAIN2760
324          KEND=IRB(INS,1)+1                               MAIN2770
325          C                                              MAIN2780
326          DO 2250 L=2,L3                                  MAIN2790
327          IF(IB(L,1,INS).EQ.0) GO TO 2260                MAIN2800
328          DO 2240 K=2,KEND                                 MAIN2810
329          IF(IB(L,1,INS).NE.IRB(INS,K)) GO TO 2240      MAIN2820
330          NUM=NUM+1                                       MAIN2830
331          NHOLD(NUM)=IRB(INS,K)                           MAIN2840
332          IB(L,1,INS)=0                                    MAIN2850
333          GO TO 2250                                       MAIN2860
334          2240 CONTINUE                                   MAIN2870
335          2250 CONTINUE                                   MAIN2880
336          2260 IF(NUM.NE.0) GO TO 2300                    MAIN2890
337          PRINT 2270, INS,NUM,INS                          MAIN2900
338          2270 FORMAT('1ERROR IN MAIN PROGRAM AT 2270.'/'0ERROR DURING SENSE BLOCK MAIN2910
339          1K'I3,' INPUT REARRANGE TO STORE ACTIVE BLOCKS BEFORE STANDBY BLOCK MAIN2920
340          2S IN IB('I3,'.1,'I3,')')
341          GO TO 1066                                       MAIN2940
342          C                                              MAIN2950
343          C  SHIFT IB(NB,1,INS) TO LEFT ELIMINATING ZEROES. MAIN2960
344          C                                              MAIN2970
345          2300 ILEFT=0                                       MAIN2980
346          DO 2350 J=2,L3                                  MAIN2990
347          IF(IB(J,1,INS).NE.0) GO TO 2330                MAIN3000
348          ILEFT=ILEFT+1                                    MAIN3010
349          GO TO 2350                                       MAIN3020
350          2330 IF(ILEFT.EQ.0) GO TO 2350                  MAIN3030
351          C                                              MAIN3040
352          C  MOVE NB TO LEFT.                                MAIN3050
353          C                                              MAIN3060
354          M=J-ILEFT                                       MAIN3070
355          IB(M,1,INS)=IB(J,1,INS)                          MAIN3080
356          2350 CONTINUE                                   MAIN3090
357          C                                              MAIN3100
358          C  ADD NS IN NHOLD AT END.                        MAIN3110
359          C                                              MAIN3120
360          KBEG=L3-ILEFT+1                                  MAIN3130
361          M=0                                              MAIN3140
362          DO 2370 K=KBEG,L3                                MAIN3150
363          M=M+1                                           MAIN3160
364          IB(K,1,INS)=NHOLD(M)                             MAIN3170
365          2370 CONTINUE                                   MAIN3180
366          C                                              MAIN3190
367          C  PRINT SENSE BLOCK AND STANDBY BLOCKS OF THAT SENSE BLOCK. MAIN3200
368          C                                              MAIN3210
369          DO 2025 K=2,L5                                  MAIN3211
370          IF(IRB(INS,K).EQ.0) GO TO 2026                  MAIN3212
371          2025 CONTINUE                                   MAIN3213
372          2026 K=K-1                                       MAIN3214
373          PRINT 2021, INS, (IRB(INS,J), J=2,K)           MAIN3220
374          2021 FORMAT('0SENSE SWITCH'I3,' CONTROLS THE FOLLOWING BLOCKS',2(1H , MAIN3230
375          13CI4))
376          2020 CONTINUE                                   MAIN3239
377          C                                              MAIN3240
378          C  READ AND GENERATE ITEMP ARRAY OF EQUIVALENT BLOCKS. MAIN3250
379          C  N1 = MAXIMUM NUMBER OF EQUIVALENT BLOCK SETS. MAIN3260

```

380	C		MAIN3270
381		840 READ 1200, N1	MAIN3280
382		IF(N1.EQ.0) GO TO 2040	MAIN3290
383		141 DO 125 I=1,N1	MAIN3300
384		125 READ 124,(ITEMP(J,I),J=2,LL6)	MAIN3310
385		124 FORMAT(40I2)	MAIN3320
386	C		MAIN3330
387	C	PRINT EQUIVALENT BLOCKS.	MAIN3340
388	C		MAIN3350
389		140 PRINT 905	MAIN3360
390		905 FORMAT(1HC17EQUIVALENT BLOCKS)	MAIN3370
391		DO 123 I=1,N1	MAIN3380
392		PRINT 906,(ITEMP(J,I),J=2,LL6)	MAIN3390
393		906 FORMAT('C'40I3)	MAIN3400
394		DO 123 K=2,LL6	MAIN3410
395		123 ITEMP(K+M6,I)=-ITEMP(K,I)	MAIN3420
396		2040 CONTINUE	MAIN3430
397	C		MAIN3440
398	C	RELIABILITY BLOCK DIAGRAM INPUT CHECKED.	MAIN3450
399	C		MAIN3460
400	C		MAIN3470
401	C	GENERATE ORIGINAL TREE.	MAIN3480
402	C	ISAVE HOLDS THE PROBABILITY TREE SUCCESS PATHS	MAIN3490
403	C	JS1 = NUMBER OF SUCCESS PATHS.	MAIN3500
404	C		MAIN3510
405		3000 CALL TREE(\$998)	MAIN3520
406	C		MAIN3530
407	C	DUPLICATE ORIGINAL TREE INTO JSAVE.	MAIN3540
408	C		MAIN3550
409		CALL DUPTRE(JSAVE,IJS1,ISAVE,JS1)	MAIN3560
410	C		MAIN3570
411	C	IDIST DETERMINES WHAT FAILURE DISTRIBUTION IS USED(1=EXPONENTIAL)	MAIN3580
412	C	TTOT = MISSION TIME. IPRINT 1 = PRINT ALL TREES,	MAIN3590
413	C	0 = SUPPRESS PRINTING OF ALL TREES EXCEPT THE ORIGINAL TREE.	MAIN3600
414	C		MAIN3610
415		READ 4000, IDIST, TTOT	MAIN3620
416		4000 FORMAT(I2,E12.7)	MAIN3630
417	C		MAIN3640
418	C	READ PARAMETERS OF FAILURE DISTRIBUTION.	MAIN3650
419	C	RC IS SET = 1.0 IF READ IN AS BLANK OR ZERO.	MAIN3660
420	C		MAIN3670
421		GO TO (4100,4200,4300,4400,4500),IDIST	MAIN3680
422	C		MAIN3690
423	C	READ ACTIVE PARAMETERS.	MAIN3700
424	C		MAIN3710
425		4100 READ 4101, NB, TLAMB0, RC1, LAST	MAIN3720
426		4101 FORMAT(I2,E12.7,F10.7,55X,I1)	MAIN3730
427		IF(RC1)4109,4110,4109	MAIN3740
428		4109 TL(NB)=TLAMB0	MAIN3750
429		RC(NB)=RC1	MAIN3760
430		GO TO 4120	MAIN3770
431		4110 RC(NB)=1.0	MAIN3780
432		TL(NB)=TLAMB0	MAIN3790
433		4120 IF(LAST.LT.5) GO TO 4100	MAIN3800
434	C		MAIN3810
435	C	READ DORMANCY PARAMETERS.	MAIN3820
436	C	DFACT = DORMANCY FACTOR TO BE MULTIPLIED BY STANDBY ACTIVE LAMBDA	MAIN3830
437	C	IF IDUMMY NON ZERO AND DFACT = 0 - READ LAMBDA DORMANT = TLD(NB).	MAIN3840
438	C		MAIN3850
439		IF(NSR.EQ.0) GO TO 4710	MAIN3860
440		READ 4101, IDUMMY, DFACT	MAIN3870
441		IF(IDUMMY.EQ.0) GO TO 4160	MAIN3880
442		IF(DFACT.GT.0.) GO TO 4160	MAIN3890
443		4150 READ 4101, NB, TLAMB0, DUMMY, LAST	MAIN3900
444		TLD(NB)=TLAMB0	MAIN3910
445		IF(LAST.LT.6) GO TO 4150	MAIN3920
446		GO TO 4190	MAIN3930
447		4160 DO 4180 J=1,NSR	MAIN3940
448		NS=IS(J)	MAIN3950
449		KEND=IRB(NS,1)+1	MAIN3960
450		DO 4170 K=2,KEND	MAIN3970
451		NL=IRB(NS,K)	MAIN3980
452		TLD(NS)=DFACT*TL(NB)	MAIN3990
453		4170 CONTINUE	MAIN4000
454		4180 CONTINUE	MAIN4010
455	C		MAIN4020

```

456      C                                          MAIN4030
457      C      READ SWITCHING OPTIONS AND DATA FOR EACH SENSE BLOCK.      MAIN4040
458      C                                          MAIN4050
459      C      0 = PERFECT SWITCHING( PROBABILITY SWITCH WORKS EQUALS 1.0)    MAIN4060
460      C      1 = CONSTANT PROBABILITY THAT SWITCH WORKS.                  MAIN4070
461      C      2 = DORMANT FAILURE RATE FOR SWITCH.                        MAIN4080
462      C      3 = DORMANT AND ACTIVE FAILURE RATE FOR SWITCH.             MAIN4090
463      C      4 = NOT AVAILABLE                                           MAIN4100
464      C      5 = NOT AVAILABLE                                           MAIN4110
465      C                                          MAIN4120
466      C      4190 DO 4700 J=1,NSR                                         MAIN4130
467      C          READ 4101, NB,TLAMB0,SPROB,ISOPT                          MAIN4140
468      C          IF(ISOPT.EQ.0) GO TO 4600                                MAIN4150
469      C          GO TO (4610,4620,4630,4640,4650), ISOPT                  MAIN4160
470      C      4600 SWPROB(NB)=1.0                                          MAIN4170
471      C          TLDS(NB)=0.0                                             MAIN4180
472      C          TLS(NB)=0.0                                              MAIN4190
473      C          GO TO 4680                                               MAIN4200
474      C      4610 SWPROB(NB)=SPROB                                        MAIN4210
475      C          TLDS(NB)=0.0                                             MAIN4220
476      C          TLS(NB)=0.0                                              MAIN4230
477      C          GO TO 4680                                               MAIN4240
478      C      4620 SWPROB(NB)=1.0                                          MAIN4250
479      C          TLDS(NB)=TLAMB0                                         MAIN4260
480      C          TLS(NB)=0.0                                              MAIN4270
481      C          GO TO 4680                                               MAIN4280
482      C      4630 SWPROB(NB)=1.0                                          MAIN4290
483      C          TLDS(NB)=TLAMB0                                         MAIN4300
484      C          READ 4101, NB,TLAMB0                                     MAIN4310
485      C          TLS(NB)=TLAMB0                                         MAIN4320
486      C          GO TO 4680                                               MAIN4330
487      C      4640 CONTINUE                                               MAIN4340
488      C          GO TO 4680                                               MAIN4350
489      C      4650 CONTINUE                                               MAIN4360
490      C      4680 TLD(NB)=0.0                                             MAIN4370
491      C          TL(NB)=0.0                                               MAIN4380
492      C          RC(NB)=1.0                                               MAIN4390
493      C      4700 CONTINUE                                               MAIN4400
494      C                                          MAIN4410
495      C      READ LAST(80), LAST = 7, 8, CR 9.                            MAIN4420
496      C      7 = RECALCULATE WITH NEW PARAMETERS, 8 = NEW DIAGRAM, 9 = END. MAIN4430
497      C                                          MAIN4440
498      C      4710 READ 2707, LAST                                         MAIN4450
499      C          GO TO 4800                                               MAIN4460
500      C                                          MAIN4470
501      C      4200 CONTINUE                                               MAIN4480
502      C      4300 CONTINUE                                               MAIN4490
503      C      4400 CONTINUE                                               MAIN4500
504      C      4500 CONTINUE                                               MAIN4510
505      C                                          MAIN4520
506      C      4200,4300,4400,4500 AVAILABLE IF YOU WANT TO USE OTHER THAN   MAIN4530
507      C      EXPONENTIAL DISTRIBUTION.  ADD APPROPRIATE STATMENTS FOR NEW  MAIN4540
508      C      DISTRIBUTION.                                               MAIN4550
509      C                                          MAIN4560
510      C      4800 IF(NSR.EQ.0) GO TO 8000                                  MAIN4570
511      C                                          MAIN4580
512      C      FIND A SENSE TREE WITH NO OTHER SENSE BLOCKS IN IT.         MAIN4590
513      C                                          MAIN4600
514      C      7000 CONTINUE                                               MAIN4610
515      C          DO 7010 JJ=1,M1                                          MAIN4620
516      C      7010 NP(JJ)=0                                               MAIN4630
517      C          INDEF=0                                                 MAIN4640
518      C          M=1                                                     MAIN4650
519      C          J=1                                                     MAIN4660
520      C          I=1                                                     MAIN4670
521      C      7011 IF(JSAVE(I,J).NE.0) GO TO 7001                          MAIN4680
522      C          IF(I-1) 7002,7003,7002                                  MAIN4690
523      C      7001 K=1                                                     MAIN4700
524      C      7004 L=1                                                     MAIN4710
525      C      7005 CONTINUE                                               MAIN4720
526      C          IF(NP(L).EQ.IS(K)) GO TO 7006                            MAIN4730
527      C          IF(L.EQ.M) GO TO 7007                                    MAIN4740
528      C          L=L+1                                                    MAIN4750
529      C          GO TO 7005                                               MAIN4760
530      C      7007 IF(ABS(JSAVE(I,J)).EQ.IS(K)) GO TO 7008                MAIN4770

```

```

531      7006 IF(K.EQ.NSR) GO TO 7009                MAIN4780
532          K=K+1                                MAIN4790
533          GO TO 7004                            MAIN4800
534      7008 IF(IS(K).EQ.INDEP) GO TO 7009        MAIN4810
535          M=M+1                                MAIN4820
536          NP(M-1)=INDEP                        MAIN4830
537          INDEF=IS(K)                          MAIN4840
538          IF(IPRINT.NE.4) GO TO 7009          MAIN4850
539          PRINT 20001, INDEP                   MAIN4860
540      7009 IF(I.EQ.M1) GO TO 7002              MAIN4870
541          I=I+1                                MAIN4880
542          GO TO 7011                            MAIN4890
543      7002 IF(J.EQ. M2) GO TO 7003            MAIN4900
544          I=1                                  MAIN4910
545          J=J+1                                MAIN4920
546          GO TO 7011                            MAIN4930
547      7003 ISUC=INDEF                          MAIN4940
548          IF(IPRINT.NE.4) GO TO 7030          MAIN4950
549          PRINT 20001, ISUC                   MAIN4960
550      20001 FORMAT (I3)                        MAIN4970
551          PRINT 20002, (IS(I),I=1,M1)         MAIN4980
552          PRINT 20002, (NP(I),I=1,M1)         MAIN4990
553      20002 FORMAT(1HC25(I3,1X)/1H 25(I3,1X)) MAIN5000
554      C                                         MAIN5010
555      C                                         MAIN5020
556      C      DETERMINE WHAT TYPE OF STANDBY IS INVOLVED. MAIN5030
557      C                                         MAIN5040
558      C                                         MAIN5050
559      C      IF SENSE BLOCK CONTROLS ONE STANDBY BLOCK, SET ISTDBY = 1, SEE IF MAIN5060
560      C      THAT STANDBY BLOCK HAS AN INPUT. IF NO INPUT GO TO 7700. IF IT MAIN5070
561      C      HAS AN INPUT GO TO 7500. IF SENSE BLOCK CONTROLS MORE THAN ONE MAIN5080
562      C      STANDBY BLOCK, SET ISTDBY=2 AND CALL SUBROUTINE STDBY2. MAIN5090
563      C                                         MAIN5100
564      7030 IF(IRB(INDEP,1).GT.1) GO TO 7050    MAIN5110
565          ISTDBY=1                             MAIN5120
566          IRBB=IRB(INDEP,2)                   MAIN5130
567          ITEST=IB(1,1,IRBB)                  MAIN5140
568          IF(ITEST.EQ.0) GO TO 7700           MAIN5150
569          GO TO 7500                           MAIN5160
570      7050 ISTDBY=2                           MAIN5170
571      C                                         MAIN5180
572          CALL STDBY2($998,INDEP)             MAIN5190
573      C                                         MAIN5200
574      C      DUPLICATE STANDBY TREE RETURNED IN ISAVE INTO LSAVE. MAIN5210
575      C                                         MAIN5220
576          CALL DUPTREI(LSAVE,LJS1,ISAVE, JS1) MAIN5230
577          IF(IPRINT.LT.3) GO TO 7200         MAIN5240
578          CALL TRPRNT(2,JSAVE,IJS1, 1 )      MAIN5250
579          CALL TRPRNT(4,LSAVE,LJS1, 2 )      MAIN5260
580      C                                         MAIN5270
581      C      GENERATE FULL STANDBY TREE.       MAIN5280
582      C                                         MAIN5290
583      C      SUBROUTINE TRENUM STORES IN NBNUM ALL THE DIFFERENT ABSOLUTE VALUE MAIN5300
584      C      BLOCK NUMBERS IN THE STANDBY TREE. NUMOUT ZEROES OUT THESE MAIN5310
585      C      BLOCK NUMBERS.                   MAIN5320
586      C                                         MAIN5330
587      7200 CALL TRENUM(LSAVE,LJS1)           MAIN5340
588      C                                         MAIN5350
589          ISUC=IRB(INDEP,2)                   MAIN5360
590      C                                         MAIN5370
591      C      IF STANDBY BLOCK HAS NO INPUT, GO TO 7500. MAIN5380
592      C                                         MAIN5390
593          ITEST=IB(1,1,ISUC)                  MAIN5400
594          IF(ITEST.EQ.0) GO TO 7500           MAIN5410
595      C                                         MAIN5420
596          CALL TREE($998)                     MAIN5430
597      C                                         MAIN5440
598      C      ZERO OUT STANDBY BLOCKS IN STANDBY TREE. MAIN5450
599      C                                         MAIN5460
600          CALL NUMOUT($998,ISAVE, JS1)      MAIN5470
601          CALL SFLEFT($998,ISAVE, JS1)       MAIN5480
602          CALL DUPOUT($998,ISAVE, JS1)       MAIN5490
603          IF(IPRINT.LT.3) GO TO 7400         MAIN5500
604          CALL TRPRNT(1,ISAVE, JS1, 3 )      MAIN5510
605      7400 CALL TRENUM(ISAVE,JS1)           MAIN5520

```

```

606      C                                MAIN553C
607      C                                MAIN554C
608      C      CALCULATE THE PROBABILITY(PI TREE) THAT YOU NEED THE STANDBY TREE. MAIN555C
609      C                                MAIN556C
610      C                                MAIN557C
611      C      7500 ISUC=INDEP                MAIN558C
612      C      CALL TREE($998)                MAIN559C
613      C                                MAIN560C
614      C      DUPLICATE SENSE TREE INTO KSAVE. MAIN561C
615      C                                MAIN562C
616      C      CALL DUPTRE(KSAVE,KJS1,ISAVE, JS1) MAIN563C
617      C                                MAIN564C
618      C      SUBTRACT FULL STANDBY TREE FROM SENSE TREE GIVING PI TREE. MAIN565C
619      C                                MAIN566C
620      C      GO TO (7513,7518),ISTDBY        MAIN567C
621      C                                MAIN568C
622      C      GENERATE FULL STANDBY TREE.     MAIN569C
623      C                                MAIN570C
624      C      7515 ISUC=IR3(INDEP,2)          MAIN571C
625      C      CALL TREE($998)                MAIN572C
626      C                                MAIN573C
627      C      CALL TRENUM(ISAVE, JS1)         MAIN574C
628      C                                MAIN575C
629      C      INITIALIZE                      MAIN576C
630      C                                MAIN577C
631      C      7518 DO 7520 I=1,M1            MAIN578C
632      C      NBOUT(I)=C                      MAIN579C
633      C      7520 CONTINUE                   MAIN580C
634      C                                MAIN581C
635      C      ELIMINATE PATHS IN SENSE TREE IF THEY CONTAIN THE STANDBY BLOCKS. MAIN582C
636      C                                MAIN583C
637      C      NBOTMX=IRB(INDEP,1)             MAIN584C
638      C      DO 7530 J=1,NBOTMX              MAIN585C
639      C      NBOUT(J)=IRB(INDEP,J+1)         MAIN586C
640      C      7530 CONTINUE                   MAIN587C
641      C      CALL PATHGT($998,KSAVE,KJS1)     MAIN588C
642      C      IF(IPRINT.LT.3) GO TO 7535      MAIN589C
643      C      CALL TRPRNT(3,KSAVE,KJS1, 4 )   MAIN590C
644      C                                MAIN591C
645      C      7535 NENMAX=NBENMAX+1           MAIN592C
646      C      NENUM(NENMAX)=INDEP            MAIN593C
647      C      CALL NUMOUT($998,KSAVE,KJS1)     MAIN594C
648      C      CALL SFLEFT($998,KSAVE,KJS1)    MAIN595C
649      C      CALL DUPOUT($998,KSAVE,KJS1)    MAIN596C
650      C      IF(IPRINT.LT.3) GO TO 7538      MAIN597C
651      C      CALL TRPRNT(3,KSAVE,KJS1, 5 )   MAIN598C
652      C                                MAIN599C
653      C      CHECK PI TREE FOR STANDBY TREES REPLACED BY STANDBY BLOCK(51-65)/ MAIN600C
654      C      (L1-L4).                        MAIN601C
655      C                                MAIN602C
656      C      7538 IF (NSBYMX.LE.1) GO TO 7545 MAIN603C
657      C      REWIND 10                       MAIN604C
658      C      DO 7540 N=1,NSBYMX              MAIN605C
659      C      READ(10) NSS,JS1                MAIN606C
660      C      DO 7537 I=1,M2                  MAIN607C
661      C      READ(10) (ISAVE(J,I),J=1,M1)    MAIN608C
662      C      7537 CONTINUE                   MAIN609C
663      C      CALL TRINTR($998,KSAVE,KJS1,ISAVE,JS1,NSTDBY,NSBYMX,NSS) MAIN610C
664      C      7540 CONTINUE                   MAIN611C
665      C                                MAIN612C
666      C      DUPLICATE KSAVE INTO ISAVE.     MAIN613C
667      C                                MAIN614C
668      C      7545 CALL DUPTRE(ISAVE, JS1,KSAVE,KJS1) MAIN615C
669      C                                MAIN616C
670      C      PI TREE NOW IN ISAVE.           MAIN617C
671      C                                MAIN618C
672      C      IF(IPRINT.LT.3) GO TO 5000      MAIN619C
673      C      CALL TRPRNT(1,ISAVE, JS1, 6 )   MAIN620C
674      C      GO TO 5000                      MAIN621C
675      C                                MAIN622C
676      C      GENERATE SENSE TREE.            MAIN623C
677      C                                MAIN624C
678      C      7700 ISUC=INDEP                MAIN625C
679      C      CALL TREE($998)                MAIN626C
680      C                                MAIN627C
681      C      INITIALIZE                      MAIN628C

```



```

682      C
683      DO 7710 I=1,M1
684      NBOUT(I)=0
685      NBNUM(I)=0
686      7710 CONTINUE
687      C
688      C      ELIMINATE PATHS IN SENSE TREE IF THEY CONTAIN THE STANDBY BLOCK.
689      C
690      NBOUT(1)=1RB(INDEP,2)
691      NEOTMX=1
692      CALL PATHGT($998,ISAVE, JS1)
693      IF(IPRINT.LT.3) GO TO 7720
694      CALL TRPRNT(1,ISAVE, JS1, 7 )
695      C
696      C      ZERO OUT SENSE BLOCK NUMBER IN SENSE TREE.
697      C
698      7720 NBNUM(1)=INDEP
699      NBNMAX=1
700      CALL NUMOUT($998,ISAVE, JS1)
701      CALL SFLEFT($998,ISAVE, JS1)
702      CALL DUPOUT($998,ISAVE, JS1)
703      IF(IPRINT.LT.3) GO TO 5000
704      CALL TRPRNT(1,ISAVE, JS1, 8 )
705      C
706      C      CALCULATE PI, THE PROBABILITY YOU NEED SENSE TREE, STORE IN PSNS.
707      C
708      5000 T=0.
709      NCW=0
710      K=0
711      9000 CONTINUE
712      GO TO (5100,5200,5300,5400,5500),IDIST
713      C
714      5100 DO 5101 I=1,NBMAX
715      DPREC=-T*TL(I)
716      PROB1(I)=RD(I)*DEXP(DPREC)
717      5101 CONTINUE
718      C
719      C      KRB ARRAY CONTAINS STANDBY BLOCKS THAT HAVE HAD THEIR PROBABILITY
720      C      PREVIOUSLY CALCULATED.
721      C
722      DO 5110 J=1,KRB1
723      IF(KRB(J).EQ.0) GO TO 5110
724      NB=KRB(J)
725      PROB1(NB)=RT(NB,K+1)
726      5110 CONTINUE
727      C
728      C      SUBROUTINE SYSP CALCULATES THE SYSTEM PROBABILITY - RETURNS TOT.
729      C
730      CALL SYSP
731      C
732      IF(NOW.EQ.1) GO TO 9001
733      K=K+1
734      T=(TTOT*K)/NSTEP
735      PSNS(K)=1.0D+0 - TOT
736      C
737      C      PREVENT DIVISION BY ZERO WHEN 18 SIGNIFICANT FIGURES LOST.
738      C
739      IF(PSNS(K).GT.0.0D+0) GO TO 5160
740      IF(K.EQ.1) GO TO 5160
741      C
742      C      NEXT STATEMENT ELIMINATES DIVISION BY ZERO LATER WHEN CALCULATING
743      C      RT. THE NUMBER USED RESULTS FROM SUBTRACTING .9999999999999999
744      C      FROM 1.0D+0 JUST BEFORE LOSING ALL SIGNIFICANT FIGURES. THIS WILL
745      C      CAUSE AN INTEGRATION ERROR IN CALCULATING RT, THE RELIABILITY OF
746      C      A SENSE BLOCK OR SENSE TREE. THE SYSTEM RELIABILITY MAY NOT BE
747      C      EFFECTED IF IT APPROACHES 1.0.
748      C
749      PSNS(K) = +.173472347597680709D-017
750      5160 IF(K.LE.NSTEP) GO TO 5100
751      ISTEP=NSTEP+1
752      IF(IPRINT.NE.4) GO TO 6000
753      PRINT 5180, (PSNS(K),K=1,ISTEP)
754      5180 FORMAT(5026.18)
755      GO TO 6000
756      5200 CONTINUE

```

```

MAIN6290
MAIN6300
MAIN6310
MAIN6320
MAIN6330
MAIN6340
MAIN6350
MAIN6360
MAIN6370
MAIN6380
MAIN6390
MAIN6400
MAIN6410
MAIN6420
MAIN6430
MAIN6440
MAIN6450
MAIN6460
MAIN6470
MAIN6480
MAIN6490
MAIN6500
MAIN6510
MAIN6520
MAIN6530
MAIN6540
MAIN6550
MAIN6560
MAIN6570
MAIN6580
MAIN6590
MAIN6600
MAIN6610
MAIN6620
MAIN6630
MAIN6640
MAIN6650
MAIN6660
MAIN6670
MAIN6680
MAIN6690
MAIN6700
MAIN6710
MAIN6720
MAIN6730
MAIN6740
MAIN6750
MAIN6760
MAIN6770
MAIN6780
MAIN6790
MAIN6800
MAIN6810
MAIN6820
MAIN6830
MAIN6840
MAIN6850
MAIN6860
MAIN6870
MAIN6880
MAIN6890
MAIN6900
MAIN6910
MAIN6920
MAIN6930
MAIN6940
MAIN6950
MAIN6960
MAIN6970
MAIN6980
MAIN6990
MAIN7000
MAIN7010
MAIN7020
MAIN7030

```

```

757      5300 CONTINUE                                MAIN704C
758      5400 CONTINUE                                MAIN7050
759      5500 CONTINUE                                MAIN7060
760      C SEE COMMENT STATEMENT AFTER 4500.         MAIN7070
761      GO TO 6000                                    MAIN7080
762      C                                             MAIN7090
763      C      NUMERICALLY INTEGRATE THE RELIABILITY OF THE STANDBY BLOCKS. MAIN7100
764      C      TIME VARIABLE = JT = 0-MISSION TIME(TTOT). KT = 1-101 INCREMENTS. MAIN7110
765      C                                             MAIN7120
766      C                                             MAIN7130
767      6000 GO TO (6100,6400),ISTDBY                MAIN7140
768      C                                             MAIN7150
769      C      METHOD FOR ONE STANDBY BLOCK(ISTDBY=1). MAIN7160
770      C                                             MAIN7170
771      6100 NB=IRB(INDEP,2)                          MAIN7180
772      RT(NB,1)=RC(NB)                              MAIN7190
773      GO TO (6102,6202,6302,6402,6502),IDIST      MAIN7200
774      C                                             MAIN7210
775      6102 ISTEP=NSTEP+1                            MAIN7220
776      DO 6170 KT=2,ISTEP                            MAIN7230
777      DPREC = (-TTOT*(KT-1)/NSTEP)*TL(NB)          MAIN7240
778      DPREC1= (-TTOT*(KT-1)/NSTEP)*TLS(INDEP)     MAIN7250
779      PINT= RC(NB)*PSNS(1)*DEXP(DPREC)*SWPROB(INDEP)*DEXP(DPREC1) MAIN7260
780      JTEND=KT-1                                     MAIN7270
781      DO 6150 JT=1,JTEND                             MAIN7280
782      DPREC2=((-TTOT*(KT-1)/NSTEP)*TL(NB))+((-TTOT*(JT-1)/NSTEP)* MAIN7290
783      1 (TLD(NB)-TL(NB)))                            MAIN7300
784      DPREC3=((-TTOT*(KT-1)/NSTEP)*TL(NB))+((-TTOT*(JT )/NSTEP)* MAIN7310
785      1 (TLD(NB)-TL(NB)))                            MAIN7320
786      DPREC4=((-TTOT*(KT-1)/NSTEP)*TLS(INDEP))+((-TTOT*(JT-1)/NSTEP)* MAIN7330
787      1 (TLD(INDEP)-TLS(INDEP)))                    MAIN7340
788      DPREC5=((-TTOT*(KT-1)/NSTEP)*TLS(INDEP))+((-TTOT*(JT )/NSTEP)* MAIN7350
789      1 (TLD(INDEP)-TLS(INDEP)))                    MAIN7360
790      YAVE=((DEXP(DPREC2)*DEXP(DPREC4))+DEXP(DPREC3)*DEXP(DPREC5))/2.00 MAIN7370
791      PINT = PINT + RC(NB)*(PSNS(JT+1)-PSNS(JT)) * YAVE * SWPROB(INDEP) MAIN7380
792      6150 CONTINUE                                    MAIN7390
793      RT(NB,KT)=PINT/PSNS(KT)                        MAIN7400
794      6170 CONTINUE                                    MAIN7410
795      IF(IPRINT.NE.4) GO TO 6010                     MAIN7420
796      PRINT 5888, (RT(NB,KT), KT=1,ISTEP)           MAIN7430
797      5888 FORMAT(8E15.9)                            MAIN7440
798      GO TO 6010                                     MAIN7450
799      6202 CONTINUE                                    MAIN7460
800      6302 CONTINUE                                    MAIN7470
801      6402 CONTINUE                                    MAIN7480
802      6502 CONTINUE                                    MAIN7490
803      C SEE COMMENT STATEMENT AFTER 4500.         MAIN7500
804      C                                             MAIN7510
805      C                                             MAIN7520
806      C      METHOD FOR MORE THAN ONE STANDBY BLOCK(ISTDBY=2) MAIN7530
807      C      RC FOR ALL STANDBY BLOCKS(51-65) = 1  MAIN7540
808      C                                             MAIN7550
809      6400 NB=NSTDBY(NSBYMX)                         MAIN7560
810      RT(NB,1)=1.0                                  MAIN7570
811      GO TO (6103,6203,6303,6403,6503),IDIST      MAIN7580
812      C                                             MAIN7590
813      6103 ISTEP=NSTEP+1                            MAIN7600
814      DO 6500 KT=2,ISTEP                            MAIN7610
815      DPREC = STBYPR(LSAVE,LJS1,RC,TL,RT,KT,0,TTOT,NSTEP,TLD) MAIN7620
816      DPREC1= (-TTOT*(KT-1)/NSTEP)*TLS(INDEP)     MAIN7630
817      PINT=PSNS(1)*DPREC*SWPROB(INDEP)*DEXP(DPREC1) MAIN7640
818      JTEND=KT-1                                     MAIN7650
819      DO 6450 JT=1,JTEND                             MAIN7660
820      DPREC4=((-TTOT*(KT-1)/NSTEP)*TLS(INDEP))+((-TTOT*(JT-1)/NSTEP)* MAIN7670
821      1 (TLD(INDEP)-TLS(INDEP)))                    MAIN7680
822      DPREC5=((-TTOT*(KT-1)/NSTEP)*TLS(INDEP))+((-TTOT*(JT )/NSTEP)* MAIN7690
823      1 (TLD(INDEP)-TLS(INDEP)))                    MAIN7700
824      JJT=JT-1                                       MAIN7710
825      YAVE=(STBYPR(LSAVE,LJS1,RC,TL,RT,KT,JT,TTOT,NSTEP,TLD)*DEXP(DPREC5) MAIN7720
826      1 ) + STBYPR(LSAVE,LJS1,RC,TL,RT,KT,JJT,TTOT,NSTEP,TLD)*DEXP(DPREC4) MAIN7730
827      24))/2.00+0                                     MAIN7740
828      PINT = PINT + (PSNS(JT+1)-PSNS(JT)) * YAVE * SWPROB(INDEP) MAIN7750
829      6450 CONTINUE                                    MAIN7760
830      RT(NB,KT)=PINT/PSNS(KT)                        MAIN7770
831      6500 CONTINUE                                    MAIN7780
832      IF(IPRINT.NE.4) GO TO 6010                     MAIN7790

```

833		PRINT 5889, (RT(NB,KT), KT=1,ISTEP)	MAIN7800
834		GO TO 6010	MAIN7810
835	6203	CONTINUE	MAIN7820
836	6303	CONTINUE	MAIN7830
837	6403	CONTINUE	MAIN7840
838	6503	CONTINUE	MAIN7850
839		GO TO 6010	MAIN7860
840	C		MAIN7870
841	C		MAIN7880
842	C	KRB ARRAY CONTAINS STANDBY BLOCKS WHOSE RELIABILITY HAS BEEN	MAIN7890
843	C	CALCULATED.	MAIN7900
844	C		MAIN7910
845	6010	DO 6004 N=1,L5	MAIN7920
846		IF(KRB(N).EQ.C) GO TO 6005	MAIN7930
847	6004	CONTINUE	MAIN7940
848	6005	KRB(N)=NB	MAIN7950
849		KRB1=N	MAIN7960
850	C		MAIN7970
851	C	CHECK IF ALL SENSE BLOCKS ACCOUNTED FOR, IF SO GO TO 8000.	MAIN7980
852	C		MAIN7990
853		IQ=0	MAIN8000
854		DO 6006 I=1,M1	MAIN8010
855		IF(IQ.EQ.1) GO TO 6007	MAIN8020
856		IF(IS(I).EQ.INDEP) IQ=1	MAIN8030
857		GO TO 6016	MAIN8040
858	6007	IS(I-1)=IS(I)	MAIN8050
859	6016	CONTINUE	MAIN8060
860		IF(IS(I).EQ.C) GO TO 6008	MAIN8070
861	6008	CONTINUE	MAIN8080
862	6006	NSR=NSR-1	MAIN8090
863		IF(I.EQ.2) GO TO 8000	MAIN8100
864		JJJ=JJJ+1	MAIN8110
865		IF(JJJ.GE.M4) GO TO 8000	MAIN8120
866		GO TO 7000	MAIN8130
867	C		MAIN8140
868	C	FINALIZE VARIABLES AND CALCULATE RELIABILITY OF TOTAL SYSTEM.	MAIN8150
869	C	SET NOW = 1.	MAIN8160
870	C		MAIN8170
871	8000	T=TTOT	MAIN8180
872		K=NSTEP	MAIN8190
873		NOW=1	MAIN8200
874	C		MAIN8210
875	C	DUPLICATE JSAVE INTO ISAVE.	MAIN8220
876	C		MAIN8230
877		CALL DUPTRE(ISAVE, JS1,JSAVE,IJS1)	MAIN8240
878		GO TO 9000	MAIN8250
879	C		MAIN8260
880	C	PRINT RESULTS IF RC = 1, SUPPRESS PRINTING OF RC.	MAIN8270
881	C		MAIN8280
882	9001	PRINT 806	MAIN8290
883	806	FORMAT(1H1)	MAIN8300
884	C		MAIN8310
885		IF(NSBYMX.LE.C) GO TO 700	MAIN8320
886	C		MAIN8330
887	C	PRINT STANDBY BLOCKS(51-65) AND THEIR TREES.	MAIN8340
888	C	STORE ALL DIFFERENT STANDBY BLOCKS IN THESE TREES IN NRB ARRAY.	MAIN8350
889	C		MAIN8360
890		END FILE 10	MAIN8370
891		REWIND 10	MAIN8380
892		DC 530 N=1,NSBYMX	MAIN8390
893		READ(10) NS, LJS1	MAIN8400
894		DC 500 I=1,M2	MAIN8410
895		READ(10) (LSAVE(J,I),J=1,M1)	MAIN8420
896	500	CONTINUE	MAIN8430
897		IF(IPRINT.LE.1) GO TO 525	MAIN8440
898		PRINT 510, NS	MAIN8450
899	510	FORMAT('0STANDBY TREE REPLACED BY BLOCK'I3)	MAIN8460
900		CALL TRPRNT(5,LSAVE,LJS1, 0)	MAIN8470
901	525	CALL TRENUM(LSAVE,LJS1)	MAIN8480
902		DO 520 K=1,NBMAX	MAIN8490
903		NRB1=NRB1+1	MAIN8500
904		NRB(NRB1)=NBNUM(K)	MAIN8510
905	520	CONTINUE	MAIN8520
906	530	CONTINUE	MAIN8530
907	C		MAIN8540
908	C	PRINT THE ORIGINAL TREE WITH STANDBY BLOCKS(51-65)/(L1-L4).	MAIN8550

```

909      C      IF(IPRINT.LE.1) GO TO 700                                MAIN8565
910      PRINT 560, L1, L4                                              MAIN8570
911      560 FORMAT('ORIGINAL PROBABILITY TREE WITH STANDBY BLOCKS (*I3, MAIN8580
912      1 * THRU*I3,* )')                                             MAIN8590
913      CALL TRPRNT(5,ISAVE, JS1, C )                                  MAIN8600
914      PRINT 805                                                       MAIN8610
915      C      MAIN8620
916      700 PRINT 701                                                  MAIN8630
917      701 FORMAT('          ACTIVE F/R      DORMANT F/R      R=INITIAL MAIN8640
918      1 RELIABILITY')                                               MAIN8650
919      DO 704 I=1,L4                                                  MAIN8660
920      IF(I.LE.M1) GO TO 705                                          MAIN8670
921      IF(PROB1(I).LE.0) GO TO 704                                    MAIN8680
922      PRINT 703, I, PROB1(I)                                         MAIN8690
923      703 FORMAT(' BLOCK*I3,44X,D17.7)                               MAIN8700
924      GO TO 704                                                       MAIN8710
925      C      MAIN8720
926      C      IF IN NRB ARRAY, HAS BEEN REPLACED, THEREFORE NO RELIABILITY. MAIN8730
927      C      MAIN8740
928      706 DO 702 J=1,NRB1                                           MAIN8750
929      IF(I.NE.NRB(J)) GO TO 702                                       MAIN8760
930      IF(RC(I).GE.1.0) GO TO 711                                       MAIN8770
931      PRINT 707, I, TL(I), TLD(I), RC(I)                                MAIN8780
932      GO TO 704                                                       MAIN8790
933      711 PRINT 709, I, TL(I),TLD(I)                                  MAIN8800
934      GO TO 704                                                       MAIN8810
935      702 CONTINUE                                                  MAIN8820
936      C      MAIN8830
937      C      DO NOT PRINT SENSE BLOCKS SINCE TL(I)=0, TLD(I)=0, AND RC(I)=1 ARE MAIN8840
938      C      NOT RELATED TO THE SENSE SWITCH OPTION PARAMETERS AND ARE ONLY MAIN8850
939      C      USED TO SET SENSE BLOCK PROBABILITY = 1.0 IN TREE PATHS.  MAIN8860
940      C      MAIN8870
941      DO 715 J=1,NNSR                                               MAIN8880
942      IF(IIS(J).EQ.1) GO TO 704                                       MAIN8890
943      715 CONTINUE                                                  MAIN8900
944      C      MAIN8910
945      IF(PROB1(I).LE.0.) GO TO 704                                       MAIN8920
946      IF(RC(I)-1.)705,708,705                                       MAIN8930
947      705 PRINT 707, I, TL(I), TLD(I), RC(I), PROB1(I)              MAIN8940
948      707 FORMAT(' BLOCK*I3, 2E16.7 ,F12.7,D17.7)                   MAIN8950
949      GO TO 704                                                       MAIN8960
950      708 PRINT 709, I, TL(I), TLD(I), PROB1(I)                      MAIN8970
951      709 FORMAT(' BLOCK*I3, 2E16.7 ,12X,D17.7)                     MAIN8980
952      704 CONTINUE                                                  MAIN8990
953      C      MAIN9000
954      C      PRINT SENSE SWITCH OPTIONS.                               MAIN9010
955      C      MAIN9020
956      IF(NNSR.EQ.0) GO TO 740                                       MAIN9030
957      PRINT 720                                                       MAIN9040
958      720 FORMAT('SENSE SWITCH ACTIVE F/R      DORMANT F/R      PROBABILITY') MAIN9050
959      DO 730 J=1,NNSR                                               MAIN9060
960      NS=IIS(J)                                                       MAIN9070
961      PRINT 725, NS,TL(NS),TLD(NS),SWPRGB(NS)                          MAIN9080
962      725 FORMAT(' BLOCK*I3, 2E16.7 ,F12.7,E16.7)                   MAIN9090
963      730 CONTINUE                                                  MAIN9100
964      740 R=TOT                                                       MAIN9110
965      K=-LOG10(1.-R)                                                  MAIN9120
966      K=K+NSIG                                                       MAIN9130
967      IF(K.GT.8) K=8                                                  MAIN9131
968      V(12)=VK(K+2)                                                  MAIN9132
969      V(14)=VK(K)                                                    MAIN9135
970      WRITE(6,V) TTOT,R                                              MAIN9137
971      PRINT 806                                                       MAIN9140
972      C      MAIN9150
973      C      MODIFY PROBABILITY DATA IF LAST = 7, GO TO 980.        MAIN9160
974      C      MAIN9170
975      IF(LAST.LT.7) GO TO 998                                          MAIN9180
976      IF(LAST.EQ.7) GO TO 980                                          MAIN9190
977      IF(LAST.EQ.8) GO TO 999                                          MAIN9200
978      IF(LAST.EQ.9) GO TO 1066                                         MAIN9210
979      GO TO 996                                                       MAIN9220
980      C      MAIN9230
981      C      INITIALIZE VARIABLES FOR RECALCULATION WITH NEW PARAMETERS. MAIN9240
982      C      MAIN9250
983      C      MAIN9260
984      980 DO 990 I=1,M1

```

985	990 IS(I)=IIS(I)	MAIN9270
986	NSR=NNSR	MAIN9280
987	ISUC=JSUC	MAIN9290
988	GO TO 997	MAIN9310
989	C	MAIN9320
990	C ERROR RETURN FROM SUBROUTINES. READ TO NEW DIAGRAM DATA DECK.	MAIN9330
991	C	MAIN9340
992	998 READ=2707, LAST	MAIN9350
993	2707 FORMAT (79X,I1)	MAIN9360
994	IF(LAST-8) 998,999,1066	MAIN9370
995	1066 END	MAIN9380

TREE*FREE.SFLEFT

1	C	SUBROUTINE SFLEFT SHIFTS PATHS IN ISAVE TO THE LEFT UNTIL NONZERO	SLFT 10
2	C	NB AT LEFT-MOST. FILL IN ZEROS AFTER SHIFTING. IF 50 ZEROS PJS	SLFT 20
3	C	PATH NUMBER IN NPUP AND ELIMINATES IT BY CALLING PATHUP.	SLFT 30
4	C	WRITTEN BY RON ECKSTEIN 1/1/70	SLFT 40
5	C		SLFT 50
6		SUBROUTINE SFLEFT(S,ISAVE,JS1)	SLFT 60
7	C		SLFT 70
8		COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6	SLFT 80
9		COMMON/SMSUB/NPUP(200),NPUPMX,NBOUT(50),NBOTMX,NBIN(50),NBINMX,	SLFT 90
10		1 NBNUM(50),NBMAX	SLFT 100
11		DIMENSION ISAVE(M1, M2)	SLFT 110
12	C		SLFT 120
13	C	INITIALIZE	SLFT 130
14		DO 10 I=1,M2	SLFT 140
15		NPUP(I)=0	SLFT 150
16		10 CONTINUE	SLFT 160
17		L=0	SLFT 170
18	C		SLFT 180
19		DO 100 I=1,JS1	SLFT 190
20		ILEFT=0	SLFT 200
21		DO 90 J=1,M1	SLFT 210
22		IF(ISAVE(J,I).NE.0) GO TO 70	SLFT 220
23		ILEFT=ILEFT+1	SLFT 230
24		GO TO 90	SLFT 240
25		70 IF(ILEFT.EQ.0) GO TO 90	SLFT 250
26	C		SLFT 260
27	C	MOVE NB TO LEFT	SLFT 270
28	C		SLFT 280
29		M=J-ILEFT	SLFT 290
30		ISAVE(M,I)=ISAVE(J,I)	SLFT 300
31		90 CONTINUE	SLFT 310
32		IF(ILEFT.NE.M1) GO TO 120	SLFT 320
33		L=L+1	SLFT 330
34		NPUP(L)=I	SLFT 340
35		GO TO 100	SLFT 350
36	C		SLFT 360
37	C	ZERO OUT REDUNDANT BLOCK NUMBERS.	SLFT 370
38	C		SLFT 380
39		120 IBEG=M1-ILEFT+1	SLFT 390
40		DO 150 K=IBEG,M1	SLFT 400
41		ISAVE(K,I)=0	SLFT 410
42		150 CONTINUE	SLFT 420
43		100 CONTINUE	SLFT 430
44		NPUPMX=L	SLFT 440
45		IF(NPUPMX.EQ.0) GO TO 1000	SLFT 450
46		CALL PATHUP(1100,ISAVE,JS1)	SLFT 460
47		1000 RETURN	SLFT 470
48		1100 PRINT 1101	SLFT 480
49		1101 FORMAT(' ERROR IN PATHUP CALLED FROM SFLEFT.')	SLFT 490
50		RETURN 1	SLFT 500
51		END	SLFT 510

```

TREA*FTREE.DUPOUT
 1 C SUBROUTINE DUPOUT STORES PATH NUMBERS THAT ARE DUPLICATES IN NPUP DPOT 10
 2 C AND ELIMINATES THEM BY CALLING PATHUP. DPOT 20
 3 C WRITTEN BY RON ECKSTEIN 1/1/70 DPOT 30
 4 C DPOT 40
 5 SUBROUTINE DUPOUT($,ISAVE,JS1) DPOT 50
 6 C DPOT 60
 7 COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 DPOT 70
 8 COMMON/SMSUB/NPUP(200),NPUPMX,NBOUT(50),NBOTMX,NBIN(50),NBINMX, DPOT 80
 9 1 NENUM(50),NBNMAX DPOT 90
10 DIMENSION ISAVE(M1, M2) DPOT 100
11 C DPOT 110
12 C INITIALIZE DPOT 120
13 C DPOT 130
14 DO 10 I=1,M2 DPOT 140
15 NPUP(I)=0 DPOT 150
16 10 CONTINUE DPOT 160
17 L=0 DPOT 170
18 C DPOT 180
19 DO 100 I=1,JS1 DPOT 190
20 C DPOT 200
21 C CHECK IF PATH ALREADY A DUPLICATE OF PREVIOUS PATH. DPOT 210
22 C DPOT 220
23 IF(L.EQ.0) GO TO 90 DPOT 230
24 DO 85 J=1,L DPOT 240
25 IF(NPUP(J).EQ.I) GO TO 100 DPOT 250
26 85 CONTINUE DPOT 260
27 C DPOT 270
28 C CHECK PATH AGAINST ALL HIGHER PATH NUMBERS. DPOT 280
29 C DPOT 290
30 90 II=I+1 DPOT 300
31 95 IF(II.GT.JS1) GO TO 100 DPOT 310
32 C DPOT 320
33 C CHECK IF PATH ALREADY A DUPLICATE OF PREVIOUS PATH. DPOT 330
34 C DPOT 340
35 IF(L.EQ.0) GO TO 75 DPOT 350
36 DO 80 J=1,L DPOT 360
37 IF(NPUP(J).EQ.II) GO TO 60 DPOT 370
38 80 CONTINUE DPOT 380
39 75 DO 70 K=1,M1 DPOT 390
40 IF(ISAVE(K,I).NE.ISAVE(K,II)) GO TO 60 DPOT 400
41 70 CONTINUE DPOT 410
42 C DPOT 420
43 C STORE DUPLICATE PATH NUMBER IN NPUP. DPOT 430
44 C DPOT 440
45 L=L+1 DPOT 450
46 NPUP(L)=II DPOT 460
47 60 II=II+1 DPOT 470
48 GO TO 95 DPOT 480
49 100 CONTINUE DPOT 490
50 NPUPMX=L DPOT 500
51 IF(NPUPMX.EQ.0) GO TO 1000 DPOT 510
52 CALL PATHUP($1100,ISAVE,JS1) DPOT 520
53 1000 RETURN DPOT 530
54 1100 PRINT 1101 DPOT 540
55 1101 FORMAT(' ERROR IN PATHUP CALLED FROM DUPOUT. ') DPOT 550
56 RETURN 1 DPOT 560
57 END DPOT 570

```

TREE*FTREE.PATHUP

1	C	SUBROUTINE PATHUP TAKES PATH NUMBERS TO BE ELIMINATED FROM NPUP	PAUP	10
2	C	ARRAY, ELIMINATES THESE PATHS FROM ISAVE, MOVING UP REMAINING PATH	PAUP	20
3	C	AS IT DOES, AND ZEROS OUT REDUNDANT PATHS AT THE END OF THE TREE.	PAUP	30
4	C	WRITTEN BY RON ECKSTEIN 1/1/70	PAUP	40
5	C		PAUP	50
6	C	SUBROUTINE PATHUP(\$,ISAVE,JS1)	PAUP	60
7	C		PAUP	70
8	C	COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6	PAUP	80
9	C	COMMON/SMSUB/NPUP(200),NPUPMX,NBOUT(50),NBOTMX,NBIN(50),NBINMX,	PAUP	90
10	C	1 NENUM(50),NBNMAX	PAUP	100
11	C	DIMENSION ISAVE(M1, M2)	PAUP	110
12	C		PAUP	120
13	C	CHECK IF THERE ARE PATHS TO BE ELIMINATED.	PAUP	130
14	C		PAUP	140
15	C	IF(NPUPMX.EQ.0) GO TO 200	PAUP	150
16	C		PAUP	160
17	C	IUP=0	PAUP	170
18	C	DO 100 I=1,JS1	PAUP	180
19	C	DO 90 J=1,NPUPMX	PAUP	190
20	C	IF(I.NE.NFUP(J)) GO TO 90	PAUP	200
21	C	IUP=IUP+1	PAUP	210
22	C	IF(IUP.GT.NPUPMX) GO TO 110	PAUP	220
23	C	GO TO 100	PAUP	230
24	C	90 CONTINUE	PAUP	240
25	C	IF(IUP.EQ.0) GO TO 100	PAUP	250
26	C		PAUP	260
27	C	MOVE PATH NUMBER M UP A NUMBER = IUP.	PAUP	270
28	C		PAUP	280
29	C	50 M=I-IUP	PAUP	290
30	C	DO 40 K=1,M1	PAUP	300
31	C	ISAVE(K,M)=ISAVE(K,I)	PAUP	310
32	C	40 CONTINUE	PAUP	320
33	C	100 CONTINUE	PAUP	330
34	C	IF(IUP.EQ.NPUPMX) GO TO 150	PAUP	340
35	C	110 PRINT 120, IUP,NPUPMX	PAUP	350
36	C	120 FORMAT('1ERROR IN PATHUP AT 110. IUP='I4, ' NPUPMX='I4)	PAUP	360
37	C	RETURN 1	PAUP	370
38	C	150 IEND=JS1	PAUP	380
39	C		PAUP	390
40	C	RESET JS1	PAUP	400
41	C		PAUP	410
42	C	JS1=JS1-IUP	PAUP	420
43	C		PAUP	430
44	C	ZERO OUT REDUNDANT PATHS.	PAUP	440
45	C		PAUP	450
46	C	ISTART=JS1+1	PAUP	460
47	C	DO 160 I=ISTART,IEND	PAUP	470
48	C	DO 160 J=1,M1	PAUP	480
49	C	ISAVE(J,I)=0	PAUP	490
50	C	160 CONTINUE	PAUP	500
51	C	200 RETURN	PAUP	510
52	C	END	PAUP	520


```

TREA*FTREE.PATHOT
 1 C SUBROUTINE PATHOT COMPARES NUMBERS IN NBOUOT ARRAY WITH EACH NUMBER PAOT 10
 2 C IN EACH SUCCESS PATH IN ISAVE AND GENERATES NPUP ARRAY WITH PATH PAOT 20
 3 C NUMBERS TO BE ELIMINATED IF ANY ABSOLUTE VALUES ARE EQUAL. PAOT 30
 4 C WRITTEN BY RON ECKSTEIN 1/1/70 PAOT 40
 5 C PAOT 50
 6 SUBROUTINE PATHOT(% ,ISAVE ,JS1) PAOT 60
 7 C PAOT 70
 8 COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 PAOT 80
 9 COMMON/SMSUB/NPUF(200),NPUPMX,NECUT(50),NEOTMX,NBIN(50),NBINMX, PAOT 90
10 1 NBNUM(50),NBNMAX PAOT 100
11 DIMENSION ISAVE(M1, M2) PAOT 110
12 C PAOT 120
13 C INITIALIZE PAOT 130
14 C PAOT 140
15 DO 10 I=1,M2 PAOT 150
16 NPUP(I)=0 PAOT 160
17 10 CONTINUE PAOT 170
18 L=0 PAOT 180
19 C PAOT 190
20 DO 100 I=1,JS1 PAOT 200
21 DO 90 J=1,M1 PAOT 210
22 DO 80 K=1,NBOTMX PAOT 220
23 IF(NBOCUT(K).NE.ABS(ISAVE(J,I)))GO TO 80 PAOT 230
24 L=L+1 PAOT 240
25 NPUP(L)=I PAOT 250
26 GO TO 100 PAOT 260
27 80 CONTINUE PAOT 270
28 90 CONTINUE PAOT 280
29 100 CONTINUE PAOT 290
30 NPUPMX=L PAOT 300
31 IF(NPUPMX.EQ.0) GO TO 1000 PAOT 310
32 CALL PATHUP(%1100,ISAVE,JS1) PAOT 320
33 1000 RETURN PAOT 330
34 1100 PRINT 1101 PAOT 340
35 1101 FORMAT(' ERROR IN PATHUP CALLED FROM PATHOT. ') PAOT 350
36 RETURN 1 PAOT 360
37 END PAOT 370

```

TREA*F TREE.NUMOUT			
1	C	SUBROUTINE NUMOUT ZEROES OUT ALL NUMBER BLOCKS IN ISAVE THAT ARE	NMOT 10
2	C	EQUAL IN ABSOLUTE VALUE TO ANY OF THE NBNUM ARRAY NUMBER BLOCKS.	NMOT 20
3	C	IF 50 ZEROES, STORE PATH NUMBER IN NPUP AND ELIMINATE(PATHUP).	NMOT 30
4	C	WRITTEN BY RON ECKSTEIN 1/1/70	NMOT 40
5	C		NMOT 50
6	C	SUBROUTINE NUMOUT(\$,ISAVE,JS1)	NMOT 60
7	C		NMOT 70
8		COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6	NMOT 80
9		COMMON/SMSUB/NPUP(200),NPUPMX,NBOUT(50),NBOTMX,NBIN(50),NBNMX,	NMOT 90
10		1 NBNUM(50),NBNMAX	NMOT 100
11		DIMENSION ISAVE(M1, M2)	NMOT 110
12	C		NMOT 120
13	C	INITIALIZE	NMOT 130
14	C		NMOT 140
15		DO 10 I=1,M2	NMOT 150
16		NPUP(I)=0	NMOT 160
17	10	CONTINUE	NMOT 170
18		NPUPMX=0	NMOT 180
19		L=0	NMOT 190
20	C		NMOT 200
21		DO 100 I=1,JS1	NMOT 210
22		ICOUNT=0	NMOT 220
23		DO 90 J=1,M1	NMOT 230
24		DO 80 K=1,NBNMAX	NMOT 240
25		IF(ISAVE(J,I).EQ.0) GO TO 75	NMOT 250
26		IF(NBNUM(K).EQ.ABS(ISAVE(J,I)))GO TO 70	NMOT 260
27	80	CONTINUE	NMOT 270
28		GO TO 90	NMOT 280
29		70 ISAVE(J,I)=0	NMOT 290
30		75 ICOUNT=ICOUNT+1	NMOT 300
31		90 CONTINUE	NMOT 310
32	C		NMOT 320
33	C	IF 50 ZEROES STORE PATH NUMBER IN NPUP.	NMOT 330
34	C		NMOT 340
35		IF(ICOUNT.LT.M1) GO TO 100	NMOT 350
36		L=L+1	NMOT 360
37		NPUP(L)=I	NMOT 370
38	100	CONTINUE	NMOT 380
39		NPUPMX=L	NMOT 390
40		IF(NPUPMX.EQ.0) GO TO 1000	NMOT 400
41		CALL PATHUP(\$1100,ISAVE,JS1)	NMOT 410
42	1000	RETURN	NMOT 420
43	1100	PRINT 1101	NMOT 430
44	1101	FORMAT(' ERROR IN PATHUP CALLED FROM NUMOUT.')	NMOT 440
45		RETURN 1	NMOT 450
46		END	NMOT 460

TREA*FTREE.NEGOUT

1	C	SUBROUTINE NEGOUT STORES FAILURE PATH NUMBERS ENDING WITH A	NGOT 10
2	C	NEGATIVE SIGN IN NPUP ARRAY, THEN CALLS PATHUP TO ELIMINATE.	NGOT 20
3	C	WRITTEN BY RON ECKSTEIN 1/1/70	NGOT 30
4	C		NGOT 40
5	C	SUBROUTINE NEGOUT(\$,ISAVE,JS1)	NGOT 50
6	C		NGOT 60
7	C	COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6	NGOT 70
8	C	COMMON/SMSUB/NPUP(200),NPUPMX,NBOUT(50),NBOTMX,NBIN(50),NBINMX,	NGOT 80
9	C	1 NBNUM(50),NBNMAX	NGOT 90
10	C	DIMENSION ISAVE(M1, M2)	NGOT 100
11	C		NGOT 110
12	C	INITIALIZE	NGOT 120
13	C		NGOT 130
14	C	DO 10 I=1,M2	NGOT 140
15	C	NPUP(I)=0	NGOT 150
16	C	10 CONTINUE	NGOT 160
17	C	L=0	NGOT 170
18	C		NGOT 180
19	C	DO 100 I=1,JS1	NGOT 190
20	C	DO 80 J=1,M1	NGOT 200
21	C	IF (ISAVE(J,I).EQ.G) GO TO 90	NGOT 210
22	C	80 CONTINUE	NGOT 220
23	C	J=L1	NGOT 230
24	C		NGOT 240
25	C	CHECK LAST NONZERO BLOCK NUMBER IN PATH FOR SIGN.	NGOT 250
26	C		NGOT 260
27	C	90 IF (J.LE.1) GO TO 92	NGOT 270
28	C	IF (ISAVE(J-1,I)) 91,92,100	NGOT 280
29	C	91 L=L+1	NGOT 290
30	C	NPUP(L)=I	NGOT 300
31	C	GO TO 100	NGOT 310
32	C	92 PRINT 95	NGOT 320
33	C	95 FORMAT('ERROR IN SUBROUTINE NEGOUT AT 92.')	NGOT 330
34	C	IF (J.GT.1) RETURN 1	NGOT 340
35	C	PRINT 94	NGOT 350
36	C	94 FORMAT(' ISAVE(1,1)=0')	NGOT 360
37	C	GO TO 1000	NGOT 370
38	C	100 CONTINUE	NGOT 380
39	C	NPUPMX=L	NGOT 390
40	C	IF (NPUPMX.EQ.G) GO TO 1000	NGOT 400
41	C	CALL PATHUP(\$1100,ISAVE,JS1)	NGOT 410
42	C	1000 RETURN	NGOT 420
43	C	1100 PRINT 1101	NGOT 430
44	C	1101 FORMAT(' ERROR IN PATHUP CALLED FROM NEGOUT.')	NGOT 440
45	C	RETURN 1	NGOT 450
46	C	END	NGOT 460

```

TREA*FTREE*TRINTR
 1 C SUBROUTINE TRINTR PUTS THE STANDBY BLOCK(51-65) IN PLACE OF THE TRTR 10
 2 C FIRST PATH OF THE STANDBY TREE IMBEDDED IN THE PATH OF THE TRTR 20
 3 C ORIGINAL TREE IN JSAVE, THEN ELIMINATES ORIGINAL TREE PATHS THAT TRTR 30
 4 C CONTAIN 2ND-JS1 STANDBY PATHS. THEN PUTS THE NEGATIVE STANDBY TRTR 40
 5 C BLOCK(51-65)/(L1-L4) IN AND ELIMINATES CORRESPONDING PATHS. TRTR 50
 6 C WRITTEN BY RGN ECKSTEIN 1/1/70 TRTR 60
 7 C TRTR 70
 8 SUBRGUTINE TRINTR($,JSAVE,IJS1,ISAVE,JS1,NSTDBY,NSBYMX,NSS) TRTR 80
 9 C TRTR 90
10 COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 TRTR 100
11 COMMON/SMSUB/NPUP(200),NPUPMX,NBOUT(50),NBOTMX,NBIN(50),NBINMX, TRTR 110
12 1 NBNUM(50),NBNMAX TRTR 120
13 DIMENSION JSAVE(M1, M2), ISAVE(M1, M2),NSTDBY(M4) TRTR 130
14 C TRTR 140
15 C INITIALIZE TRTR 150
16 C TRTR 160
17 JJ=0 TRTR 170
18 C TRTR 180
19 5 DO 10 J=1,M2 TRTR 190
20 NPUP(J)=0 TRTR 200
21 10 CONTINUE TRTR 210
22 L=1 TRTR 220
23 C TRTR 230
24 DO 1000 I=1,IJS1 TRTR 240
25 IF(L.GT.JS1) GO TO 1010 TRTR 250
26 J=1 TRTR 260
27 K=1 TRTR 270
28 ISTART=0 TRTR 280
29 30 IF(J.GT.M1) GO TO 1000 TRTR 290
30 IF(JSAVE(J,I).EQ.0) GO TO 1000 TRTR 300
31 IF(JSAVE(J,I).NE.ISAVE(K,L)) GO TO 80 TRTR 310
32 IF(ISTART.NE.0) GO TO 50 TRTR 320
33 ISTART=J TRTR 330
34 NPATH=I TRTR 340
35 50 K=K+1 TRTR 350
36 IF(K.GT.M1) GO TO 100 TRTR 360
37 IF(ISAVE(K,L).EQ.0) GO TO 100 TRTR 370
38 80 J=J+1 TRTR 380
39 GO TO 30 TRTR 390
40 100 IF(L.GT.1) GO TO 130 TRTR 400
41 ISTOP=ISTART+K-2 TRTR 410
42 GO TO 300 TRTR 420
43 C TRTR 430
44 C STORE ORIGINAL TREE PATHS THAT CONTAIN 2ND-JS1 STANDBY PATHS. TRTR 440
45 C TRTR 450
46 130 DO 150 J=1,M2 TRTR 460
47 IF(NPUP(J).NE.0) GO TO 150 TRTR 470
48 NPUP(J)=I TRTR 480
49 NPUPMX=J TRTR 490
50 GO TO 900 TRTR 500
51 150 CONTINUE TRTR 510
52 PRINT 160 TRTR 520
53 160 FORMAT(*1ERROR IN TRI-TR AT 160. -PUP IS FULL.*) TRTR 530
54 RETURN 1 TRTR 540
55 300 IF(L.EQ.1) GO TO 304 TRTR 550
56 PRINT 301, L TRTR 560
57 301 FORMAT(*1ERROR IN TRINTR AT 300. L=*I4) TRTR 570
58 RETURN 1 TRTR 580
59 C TRTR 590
60 C IF NSS EQUAL ZERO, TRINTR IS BEING CALLED FROM STDBY2. OTHERWISE, TRTR 600
61 C TRINTR IS BEING CALLED FROM MAIN PROGRAM TO REPLACE STANDBY TREES TRTR 610
62 C PREVIOUSLY ASSIGNED A STANDBY BLOCK (51-65)/(L1-L4). TRTR 620
63 C TRTR 630
64 304 IF(NSS.EQ.0) GO TO 305 TRTR 640
65 JSAVE(ISTART,NPATH)=NSS TRTR 650
66 GO TO 325 TRTR 660
67 C TRTR 670
68 C ASSIGN STANDBY BLOCK NUMBER(51-65) ONLY ONCE TRTR 680
69 C TRTR 690
70 305 IF(JJ.NE.C) GO TO 313 TRTR 700
71 DO 310 J=1,M4 TRTR 710
72 IF(NSTDBY(J).EQ.0) GO TO 312 TRTR 720
73 310 CONTINUE TRTR 730
74 PRINT 315 TRTR 740

```

75	315	FORMAT('!ERROR IN TRINTR AT 315. NSTDBY IS FULL.')	TRTR 750
76		RETURN 1	TRTR 760
77	312	JJ=J	TRTR 770
78		NSBYMX=JJ	TRTR 780
79	C		TRTR 790
80	C	PUT NEXT STANDBY BLOCK(51-65) IN PLACE OF THE FIRST PATH OF THE	TRTR 800
81	C	STANDBY TREE IMBEDDED IN THE PATH OF THE ORIGINAL TREE.	TRTR 810
82	C		TRTR 820
83	313	NSTDBY(JJ)=M1+JJ	TRTR 830
84		JSAVE(ISTART,NPATH)=NSTDBY(JJ)	TRTR 840
85	325	ISTART=ISTART+1	TRTR 850
86	C		TRTR 860
87	C	ZERO OUT 2ND-NTH NUMBER BLOCKS IN FIRST PATH IN ISAVE IMBEDDED IN	TRTR 870
88	C	JSAVE.	TRTR 880
89	C		TRTR 890
90		IF(ISTART.GT.ISTOP) GO TO 900	TRTR 900
91	320	DO 330 M=ISTART,ISTOP	TRTR 910
92		JSAVE(M,NPATH)=0	TRTR 920
93	330	CONTINUE	TRTR 930
94		CALL SFLEFT(\$1100,JSAVE,IJS1)	TRTR 940
95	900	L=L+1	TRTR 950
96	1000	CONTINUE	TRTR 960
97	C		TRTR 970
98	C	CHECK IF NO MORE STANDBY TREES IN ORIGINAL TREE.	TRTR 980
99	C		TRTR 990
100		IF(ISTART.EQ.0) GO TO 1111	TRTR1000
101	1010	CALL PATHUP(\$1100,JSAVE,IJS1)	TRTR1010
102		GO TO 5	TRTR1020
103	C		TRTR1030
104	C		TRTR1040
105	C	PUT NEGATIVE STANDBY BLOCK(51-65)/(L1-L4) IN PLACE OF FIRST NB	TRTR1050
106	C	THAT IS EQUAL IN ABSOLUTE VALUE TO ANY OF THE NBNUM ARRAY OF THE	TRTR1060
107	C	TREE THAT THE STANDBY BLOCK(51-65)/(L1-L4) REPLACES. ZERO OUT THE	TRTR1070
108	C	FOLLOWING NB IF THEY ARE EQUAL IN ABSOLUTE VALUE TO ANY OF THE	TRTR1080
109	C	NBNUM ARRAY. CALL SFLEFT TO ELIMINATE AND RESTORE PATHS.	TRTR1090
110	C		TRTR1100
111	C		TRTR1110
112	1111	CALL TRENUM(ISAVE,JS1)	TRTR1120
113	C		TRTR1130
114	C	ASSIGN STANDBY BLOCK(51-65)/(L1-L4) ONLY ONCE.	TRTR1140
115	C		TRTR1150
116		IF(NSS.EQ.C) GO TO 450	TRTR1160
117		NS=NSS	TRTR1170
118		GO TO 460	TRTR1180
119	450	NS=NSTDBY(JJ)	TRTR1190
120	C		TRTR1200
121	460	DO 2000 I=1,IJS1	TRTR1210
122		DO 600 J=1,M1	TRTR1220
123		DO 500 K=1,NBNMAX	TRTR1230
124		IF(JSAVE(J,I).EQ.C) GO TO 2000	TRTR1240
125		IF(NBNUM(K).EQ.ABS(JSAVE(J,I))) GO TO 510	TRTR1250
126	500	CONTINUE	TRTR1260
127		GO TO 600	TRTR1270
128	C		TRTR1280
129	C	REPLACE WITH NEGATIVE STANDBY BLOCK(51-65)/(L1-L4).	TRTR1290
130	C		TRTR1300
131	510	JSAVE(J,I)=-NS	TRTR1310
132	C		TRTR1320
133	C	ZERO OUT ANY NB FOLLOWING IF EQUAL TO NBNUM ARRAY.	TRTR1330
134	C		TRTR1340
135		LSTART=J+1	TRTR1350
136		DO 530 M=LSTART,M1	TRTR1360
137		DO 520 N=1,NBNMAX	TRTR1370
138		IF(JSAVE(M,I).EQ.C) GO TO 2000	TRTR1380
139		IF(NBNUM(N).EQ.ABS(JSAVE(M,I))) GO TO 525	TRTR1390
140	520	CONTINUE	TRTR1400
141		GO TO 2000	TRTR1410
142	525	JSAVE(M,I)=0	TRTR1420
143	530	CONTINUE	TRTR1430
144		GO TO 2000	TRTR1440
145	600	CONTINUE	TRTR1450
146	2000	CONTINUE	TRTR1460
147		CALL SFLEFT(\$1100,JSAVE,IJS1)	TRTR1470
148	C		TRTR1480
149	C	ELIMINATE DUPLICATE PATHS.	TRTR1490

150	C		TRTR1500
151		CALL DUPOUT(*1100,JSAVE,IJS1)	TRTR1510
152		RETURN	TRTR1520
153	1100	PRINT 1101, NSS	TRTR1530
154	1101	FORMAT(' SUBROUTINE CALLED FROM TRINTR. NSS='I4)	TRTR1540
155		RETURN 1	TRTR1550
156		END	TRTR1560

```

TREA*FTREE.PATHIN
 1 C SUBRCUTINE PATHIN ZEROS OUT ALL NUMBER BLOCKS IN ISAVE THAT ARE PAIN 10
 2 C NOT EQUAL TO ANY OF THE NBIN ARRAY NUMBER BLOCKS. IF PATH HAS 50 PAIN 20
 3 C ZEROS, STORE PATH NUMBER IN NPUP AND CALL PATHUP TO ELIMINATE. PAIN 30
 4 C WRITTEN BY RON ECKSTEIN 1/1/70 PAIN 40
 5 C PAIN 50
 6 C SUBRCUTINE PATHIN(% ,ISAVE,JS1) PAIN 60
 7 C PAIN 70
 8 C COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 PAIN 80
 9 C COMMON/SMSUB/NPUP(200),NPUPMX,NBCUT(50),NBOTMX,NBIN(50),NBINMX, PAIN 90
10 1 NBNUM(50),NBNMAX PAIN 100
11 DIMENSION ISAVE(M1, M2) PAIN 110
12 C PAIN 120
13 C INITIALIZE PAIN 130
14 C PAIN 140
15 DO 10 I=1,M2 PAIN 150
16 NPUP(I)=0 PAIN 160
17 10 CONTINUE PAIN 170
18 L=0 PAIN 180
19 C PAIN 190
20 DO 100 I=1,JS1 PAIN 200
21 ICOUNT=0 PAIN 210
22 DO 90 J=1,M1 PAIN 220
23 DO 80 K=1,NBINMX PAIN 230
24 IF (ISAVE(J,I).EQ.0) GO TO 85 PAIN 240
25 IF (NBIN(K).EQ.ABS(ISAVE(J,I)))GO TO 90 PAIN 250
26 80 CONTINUE PAIN 260
27 ISAVE(J,I)=0 PAIN 270
28 85 ICOUNT=ICOUNT+1 PAIN 280
29 90 CONTINUE PAIN 290
30 C PAIN 300
31 C IF 50 ZERGES STORE PATH NUMBER IN NPUP. PAIN 310
32 C PAIN 320
33 IF (ICOUNT.LT.M1) GO TO 100 PAIN 330
34 L=L+1 PAIN 340
35 NPUP(L)=I PAIN 350
36 100 CONTINUE PAIN 360
37 NPUPMX=L PAIN 370
38 IF (NPUPMX.EQ.0) GO TO 1000 PAIN 380
39 CALL PATHUP(%1100,ISAVE,JS1) PAIN 390
40 1000 RETURN PAIN 400
41 1100 PRINT 1101 PAIN 410
42 1101 FORMAT(' ERROR IN PATHUP CALLED FROM PATHIN. ') PAIN 420
43 RETURN 1 PAIN 430
44 END PAIN 440

```

```

TREA*FTREE.TRPRNT
 1      C      SUBROUTINE TRPRNT PRINTS TREE PATHS, ARRAY, AND LOCATION OF CALL. TRPT 10
 2      C      WRITTEN BY RON ECKSTEIN 1/1/70 TRPT 20
 3      C      TRPT 30
 4      C      SUBROUTINE TRPRNT(I,ISAVE,JS1,NSEQ) TRPT 40
 5      C      TRPT 50
 6      C      COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 TRPT 60
 7      C      DIMENSION ISAVE(M1, M2) TRPT 70
 8      C      TRPT 80
 9      C      GO TO (1,2,3,4,20),I TRPT 90
10      1 PRINT 5, NSEQ TRPT 100
11      5 FORMAT(*OISAVE AT *I5) TRPT 110
12      GO TO 20 TRPT 120
13      2 PRINT 6, NSEQ TRPT 130
14      6 FORMAT(*CJSAVE AT *I5) TRPT 140
15      GO TO 20 TRPT 150
16      3 PRINT 7, NSEQ TRPT 160
17      7 FORMAT(*DKSAVE AT *I5) TRPT 170
18      GO TO 20 TRPT 180
19      4 PRINT 8, NSEQ TRPT 190
20      8 FORMAT(*DLSAVE AT *I5) TRPT 200
21      20 DO 100 J=1,JS1 TRPT 210
22      DO 120 N=1,M1 TRPT 220
23      IF(ISAVE(N,J).EQ.0) GO TO 130 TRPT 221
24      120 CONTINUE TRPT 222
25      N=M1+1 TRPT 223
26      130 IEND=N-1 TRPT 224
27      PRINT 110, (ISAVE(N,J),N=1,IEND) TRPT 230
28      110 FORMAT(1H025(I3,1X),3(/1H 25(I3,1X))) TRPT 240
29      100 CONTINUE TRPT 250
30      RETURN TRPT 260
31      END TRPT 270

```



```

TREA*FTREE.TREE
 1 C SUBROUTINE TREE GENERATES THE PROBABILITY TREE SUCCESS PATHS. TREE 10
 2 C WRITTEN BY PAUL CHELSON AND REVISED BY RON ECKSTEIN 1/1/70 TREE 20
 3 C TREE 30
 4 C SUBROUTINE TREE INPUTS IN COMMON. TREE 40
 5 C TREE 50
 6 C IB(15,2,50) = IB(1,1,BLOCK NUMBER)=MAXIMUM NUMBER OF INPUTS OF NB. TREE 60
 7 C IB(1,2,BLOCK NUMBER)=MAXIMUM NUMBER OF OUTPUTS OF NB. TREE 70
 8 C THE REST OF THE ARRAY = INPUTS AND OUTPUTS OF BLOCK NUMBERS(NB) TREE 80
 9 C ISUC = SUCCESS BLOCK NUMBER(BASE OF PROBABILITY TREE). TREE 90
10 C JSUC = ISUC TO PRINT ORIGINAL PROB TREE, THEN ISUC VARIES. TREE 100
11 C IPRINT = PRINT ALL TREES IF = 1, PRINT ORIGINAL TREE ONLY IF = 0. TREE 110
12 C NOPRNT = 1/0 = DO/DO NOT PRINT TREES WHEN MODIFYING PROB DATA. TREE 120
13 C IDR(50) = HOLDS BLOCKS MADE INACTIVE BY A FAILED BLOCK(ID). TREE 130
14 C N1 = MAXIMUM NUMBER OF EQUIVALENT BLOCK SETS FROM MAIN PROGRAM. TREE 140
15 C ITEMP = ARRAY OF EQUIVALENT BLOCKS FROM MAIN PROGRAM. TREE 150
16 C TREE 160
17 C SUBROUTINE TREE OUTPUTS IN COMMON. TREE 170
18 C TREE 180
19 C ISAVE(50,200) = SUCCESS PATHS ( + - BLOCK NUMBER, PATH NUMBER) TREE 190
20 C JS1 = MAXIMUM NUMBER OF SUCCESS PATHS. TREE 200
21 C TREE 210
22 C SUBROUTINE TREE VARIABLES. TREE 220
23 C TREE 230
24 C IS = BLOCK NUMBER BEING WORKED ON. TREE 240
25 C JS = PATH NUMBER BEING WORKED ON. TREE 250
26 C ID = A FAILED BLOCK. TREE 260
27 C IE1 = NUMBER OF INPUTS FOR EACH OUTPUT BLOCK. TREE 270
28 C TREE 280
29 C SUBROUTINE TREE($) TREE 290
30 C TREE 300
31 C COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 TREE 310
32 C COMMON/LGSUB/IB(15,2,50),IIS(50),IRB(50,30),ITEMP(41,20),N1, TREE 320
33 C 1 ISAVE(50,200),ISUC,JSUC,JS1,JSAVE(50,200),IJS1,TOT,PROB(65), TREE 330
34 C 2 PROB1(65),IPRINT,NSTDBY(15),NSBYMX,NOPRNT TREE 340
35 C DIMENSION IDR(50) TREE 350
36 C DOUBLE PRECISION TOT,PROB,PROB1 TREE 360
37 C TREE 370
38 C INITIALIZE TREE 380
39 C TREE 390
40 C JD=0 TREE 400
41 C DO 20 I=1,M1 TREE 410
42 C IDR(I)=0 TREE 420
43 C DO 20 K=1,M2 TREE 430
44 C 20 ISAVE(I,K)=0 TREE 440
45 C ITRY=ISUC TREE 450
46 C JS=1 TREE 460
47 C IS=1 TREE 470
48 C TREE 480
49 C 700 ISAVE(IS,JS)=ITRY TREE 490
50 C 701 K=ISAVE(IS,JS) TREE 500
51 C IF(K)703,702,703 TREE 510
52 C 703 IS=IS+1 TREE 520
53 C TREE 530
54 C CHECK TO SEE IF IN DEAD ARRAY TREE 540
55 C TREE 550
56 C J=1 TREE 560
57 C 3 J=J+1 TREE 570
58 C DO 1 I=1,M1 TREE 580
59 C IF(IB(J,1,K)-IDR(I))1,2,1 TREE 590
60 C 2 IF(IB(J,1,K))3,4,3 TREE 600
61 C 1 CONTINUE TREE 610
62 C 4 ISAVE(IS,JS)=IB(J,1,K) TREE 620
63 C GO TO 701 TREE 630
64 C 702 IS1=IS-2 TREE 640
65 C TREE 650
66 C CHECK TO SEE IF MAXIMUM NUMBER OF PATHS REACHED. TREE 660
67 C TREE 670
68 C IF(JS-M2)902,901,902 TREE 680
69 C 901 IDR1=902 TREE 690
70 C PRINT 900, IDR1 TREE 700
71 C 900 FORMAT('!ERROR IN SUBROUTINE TREE AT'I4) TREE 710
72 C PRINT 910, JS TREE 720
73 C 910 FORMAT('0JS, THE NUMBER OF SUCCESS PATHS GENERATED, EQUALS M2, THE TREE 730
74 C 1 MAXIMUM NUMBER OF SUCCESS PATHS.'/'* THE DIAGRAM SHOULD BE BROKEN TREE 740
75 C 2DOWN INTO SEVERAL SUB-DIAGRAMS CONNECTED IN SERIES OR PARALLEL AND TREE 750

```

```

76      3 RERUN INDIVIDUALLY.' /* THEN ENTER EACH SUB-DIAGRAM AS ONE BLOCK TREE 760
77      4ON A FINAL RUN.'/* DO NOT BREAK UP EQUIVALENT BLOCK SETS OR STANDSTREE 770
78      5Y REDUNDANT SYSTEMS.'/* IF YOUR DIAGRAM CANNOT BE BRCKEN DOWN SUFFTREE 780
79      6ICIENTLY, CONTACT PAUL CHELSON OR RON ECKSTEIN AT 213 354-5691.'/*TREE 790
80      7 JS='I4) TREE 800
81      RETURN 1 TREE 810
82      C TREE 820
83      C CHECK PATH TO SEE IF ELIMINATED BY EQUIVALENT BLOCKS IN IT. TREE 830
84      C (IF THE SIGN OF ANY TWG EQUIVALENT BLOCKS IN A SET ALTERNATES) TREE 840
85      C TREE 850
86      902 IF(N1.EQ.C) GO TO 1712 TREE 860
87      DO 1130 L=1,M7 TREE 870
88      ITEMP(1,L)=0 TREE 880
89      1130 CONTINUE TREE 890
90      DO 1711 J=1,M1 TREE 900
91      K=ISAVE(J,JS) TREE 910
92      IF(K)1108,1711,1108 TREE 920
93      1108 IF(N1)1711,1711,1107 TREE 930
94      1107 DO 1100 L=1,N1 TREE 940
95      DO 1100 M=2,L6 TREE 950
96      IF(K-ITEMP(M,L))1100,1101,1100 TREE 960
97      1100 CONTINUE TREE 970
98      GO TO 1711 TREE 980
99      1101 IF(ITEMP(1,L))1112,1103,1104 TREE 990
100     1103 ITEMP(1,L)=K TREE1000
101     GO TO 1711 TREE1010
102     1104 IF(K)1102,1711,1711 TREE1020
103     1112 IF(K)1711,1102,1102 TREE1030
104     1102 ISAVE(IS-1,JS)=-ISAVE(IS-1,JS) TREE1040
105     JS=JS-1 TREE1050
106     GO TO 510 TREE1060
107     1711 CONTINUE TREE1070
108     C TREE1080
109     1712 DO 500 N=1,IS1 TREE1090
110     500 ISAVE(N,JS+1)=ISAVE(N,JS) TREE1100
111     ISAVE(IS-1,JS+1)=-ISAVE(IS-1,JS) TREE1110
112     C TREE1120
113     C ID=A FAILED BLOCK, IDR HOLDS FAILED BLOCKS AND BLOCKS MADE TREE1130
114     C INACTIVE BY A FAILED BLOCK. AN INACTIVE BLOCK IS A BLOCK FOR WHICH TREE1131
115     C ITS INPUTS ARE ALL FAILED, THUS IT IS OF NO USE AS A POSSIBLE TREE1132
116     C BLOCK IN A SUCCESS PATH TREE1133
117     C TREE1140
118     510 ID=-ISAVE(IS-1,JS+1) TREE1150
119     JS=JS+1 TREE1160
120     JD3=0 TREE1170
121     C TREE1180
122     C BEGIN DEAD ROUTINE WITH ID TREE1190
123     C CHECK ALL BLGCKS FOR WHICH ID IS AN INPUT BLOCK. WHENEVER ALL TREE1191
124     C INPUTS TO A BLOCK ARE IN IDR(THUS FAILED OR INACTIVE), THEN THAT TREE1192
125     C BLOCK IS ALSO INACTIVE,SO IS PUT IN IDR. TREE1193
126     C TREE1200
127     397 IDR(JD+1)=ID TREE1210
128     JD1=1+JD3 TREE1220
129     380 JD=JD+1 TREE1230
130     JD1=JD1-1 TREE1240
131     IF(IDR(JD))379,378,379 TREE1250
132     379 IDR1=IDR(JD) TREE1260
133     K=IB(1,2,IDR1) TREE1270
134     C TREE1280
135     C K IS NO. OF CUTPUTS OF IDR(JD) TREE1290
136     C TREE1300
137     DO 350 I=1,K TREE1310
138     IDR2=IB(I+1,2,IDR1) TREE1320
139     IB1=IB(1,1,IDR2) TREE1330
140     C TREE1340
141     C IB1 IS NO. OF INPUTS FOR EACH OUTPUT BLOCK TREE1350
142     C TREE1360
143     IF(IB1-2)352,353,353 TREE1370
144     353 DO 354 N=1,IB1 TREE1380
145     JN=1 TREE1390
146     N2=N+1 TREE1400
147     DO 355 M=1,M1 TREE1410
148     IF(IB(N2,1,IDR2)-IDR(M))355,356,355 TREE1420
149     356 JN=2 TREE1430
150     355 CONTINUE TREE1440
151     GO TO (350,354),JN TREE1450

```

```

152      354 CONTINUE                                TREE1460
153      C                                          TREE1461
154      C      FINISHING THE 354 LOOP MEANS THAT ALL INPUTS TO IDR2 ARE IN IDR. TREE1462
155      C      THUS IDR2 SHOULD ALSO BE INACTIVE AND IN IDR TREE1463
156      C                                          TREE1464
157      C      IS IT ALREADY IN IDR TREE1466
158      C                                          TREE1467
159      352 DO 390 N=1,M1                            TREE1470
160      IF(IDR2-IDR(N))390,350,390                TREE1480
161      390 CONTINUE                                TREE1490
162      C                                          TREE1491
163      C      IF NOT, PUT IDR2 IN IDR TREE1492
164      C                                          TREE1493
165      C      JD1=JD1+1                            TREE1500
166      C      JD2=JD+JD1                          TREE1510
167      C      IDR(JD2)=IDR2                       TREE1520
168      C      IF(IDR(JD2)-ISUC)350,381,350        TREE1530
169      350 CONTINUE                                TREE1540
170      GO TO 380                                    TREE1550
171      C                                          TREE1560
172      C      381 ROUTE INDICATES TOTAL FAILURE TREE1570
173      C                                          TREE1580
174      381 K=0                                       TREE1590
175      395 K=K+1                                     TREE1600
176      C      IK1=L1-K                             TREE1610
177      C      IF(ISAVE(IK1,JS))395,395,396        TREE1620
178      396 ISAVE(IK1,JS)=-ISAVE(IK1,JS)          TREE1630
179      C      ID=-ISAVE(IK1,JS)                   TREE1640
180      C      IK2=IK1+1                            TREE1650
181      C      IS=IK2                                TREE1660
182      C      DO 377 N=IK2,M1                      TREE1670
183      377 ISAVE(N,JS)=0                            TREE1680
184      C      IF(IK1-1)398,900,398                TREE1690
185      398 DO 361 I=1,M1                            TREE1700
186      361 IDR(I)=0                                  TREE1710
187      C      JD=0                                  TREE1720
188      C      IK3=IK1-1                            TREE1730
189      C      JD3=1                                  TREE1740
190      C      DO 611 I=1,IK3                       TREE1750
191      C      IF(ISAVE(I,JS))612,613,611          TREE1760
192      C                                          TREE1770
193      C      ERROR 613, A SUCCESS PATH HAS A ZERO INSIDE IT. TREE1780
194      C                                          TREE1790
195      613 ISUC=613                                  TREE1800
196      C      PRINT 900, ISUC                       TREE1810
197      C      PRINT 950                              TREE1820
198      950 FORMAT('O CHECK THE DIAGRAM ON THE FIRST PAGE OF THE PRINT OUT TO STREE1830
199      1EE IF ALL BLOCK NUMBERS ARE LISTED AND THAT THE INPUT/OUTPUT LISTSTREE1840
200      2*/' CORRECTLY DESCRIBE YOUR DIAGRAM. IF SO, MAKE SURE YOUR DIAGRATREE1850
201      3M SATISFIES THE DEFINITION OF A RELIABILITY BLOCK DIAGRAM. ') TREE1860
202      C      PRINT 920, JS                          TREE1870
203      920 FORMAT('O A SUCCESS PATH HAS A ZERO INSIDE IT. JS=*I4) TREE1880
204      C      RETURN 1                               TREE1890
205      C                                          TREE1900
206      612 JD3=JD3+1                                TREE1910
207      C      IDR(JD3)=-ISAVE(I,JS)                TREE1920
208      611 CONTINUE                                TREE1930
209      C      JD3=JD3-1                              TREE1940
210      C      GO TO 337                              TREE1950
211      378 JD=JD-1                                  TREE1960
212      C      DO 385 I=1,M1                          TREE1970
213      C      I1=L1-I                                TREE1980
214      C      ISA=ISAVE(I1,JS)                      TREE1990
215      C      IF(ISA)385,385,610                   TREE2000
216      610 DO 386 J=1,M1                            TREE2010
217      C      IF(ISA-IDR(J))386,385,386            TREE2020
218      386 CONTINUE                                TREE2030
219      C      GO TO 1000                              TREE2040
220      385 CONTINUE                                TREE2050
221      C                                          TREE2060
222      C      ERROR 385, THE 385 DO LOOP HAS BEEN COMPLETED. TREE2070
223      C                                          TREE2080
224      C      ISUC=385                               TREE2090
225      C      PRINT 900, ISUC                       TREE2100
226      C      PRINT 950                              TREE2110
227      C      PRINT 930, JS                          TREE2120

```

228	930	FORMAT('0THE 385 DO LOOP HAS BEEN COMPLETED. JS='I4)	TREE2130
229		RETURN 1	TREE2140
230	C		TREE2150
231	1000	JN=I8(1,1,ISA)	TREE2160
232		DO 387 J=1,JN	TREE2170
233		N2=I8(J+1,1,ISA)	TREE2180
234		DO 388 I=1,M1	TREE2190
235		IF(N2-IDR(I))388,387,388	TREE2200
236	388	CONTINUE	TREE2210
237		GO TO 630	TREE2220
238	387	CONTINUE	TREE2230
239	C		TREE2240
240	C	ERROR 387, THE 387 DO LOOP HAS BEEN COMPLETED.	TREE2250
241	C		TREE2260
242		ISUC=387	TREE2270
243		PRINT 900, ISUC	TREE2280
244		PRINT 950	TREE2290
245		PRINT 940, ISA	TREE2300
246	940	FORMAT('0ISA, BLOCK NUMBER NOW BEING PUT INTO PATH, IS NOT AN ELEMENT	TREE2310
247		MENT OF IDR ARRAY, YET ALL ITS INPUTS ARE ELEMENTS. ISA='I4)	TREE2320
248		RETURN 1	TREE2330
249	630	ITRY=N2	TREE2340
250	C		TREE2350
251	C	END DEAD ROUTINE. RETURN ITRY	TREE2360
252	C		TREE2370
253		GO TO 700	TREE2380
254	C		TREE2390
255	C	800 ROUTE MEANS END OF TREE	TREE2400
256	C	ZERO OUT FIRST BLOCK NUMBER IN FAILURE PATH.	TREE2410
257	C		TREE2420
258	800	ISAVE(1,JS)=0	TREE2430
259		JS1=JS-1	TREE2440
260	C		TREE2450
261	C	IF IPRINT .NE. 4 - DO NOT PRINT TREES AGAIN WHEN MODIFYING PROB	TREE2460
262	C	IF IPRINT = 2 - PRINT ORIGINAL TREE ONLY.	TREE2480
263	C	IF IPRINT = 3 - PRINT ALL TREES.	TREE2490
264	C		TREE2500
265		IF(IPRINT.LE.1) GO TO 10000	TREE2510
266		IF(ISUC.NE.JSUC) GO TO 2000	TREE2520
267		GO TO 2001	TREE2530
268	2000	IF(IPRINT.LT.3) GO TO 10000	TREE2540
269	2001	PRINT 633, ISUC	TREE2550
270	633	FORMAT('1PROBABILITY TREE'I4)	TREE2560
271		CALL TRPRNT(5,ISAVE,JS1,0)	TREE2570
272	10000	RETURN	TREE2580
273		END	TREE2590

TREA*FTREE.SYSP

```

1 C SUBROUTINE SYSP CALCULATES THE SYSTEM PROBABILITY - RETURNS TOT. SYSP 10
2 C WRITTEN BY PAUL CHELSON AND REVISED BY RON ECKSTEIN 1/1/70 SYSP 20
3 C SYSP 30
4 C SUBROUTINE SYSP INPUTS IN COMMON. SYSP 40
5 C SYSP 50
6 C JS1 = MAXIMUM NUMBER OF SUCCESS PATHS FROM TREE SUBROUTINE. SYSP 60
7 C ISAVE(50,200) = (BLOCK NUMBERS, SUCCESS PATH NUMBER) FROM TREE. SYSP 70
8 C N1 = MAXIMUM NUMBER OF EQUIVALENT BLOCK SETS FROM MAIN PROGRAM. SYSP 80
9 C ITEMP = ARRAY OF EQUIVALENT BLOCKS FROM MAIN PROGRAM. SYSP 90
10 C PROBI = PROBABILITY OF THE BLOCKS FROM MAIN PROGRAM. SYSP 100
11 C SYSP 110
12 C SUBROUTINE SYSP VARIABLES. SYSP 120
13 C SYSP 130
14 C PROB = PROBI SYSP 140
15 C K = ISAVE(J,I) = BLOCK NUMBER BEING WORKED ON. SYSP 150
16 C PTOT = PROBABILITY OF ONE SUCCESS PATH(PRODUCT OF PROB(K) IN PATH) SYSP 160
17 C TOT = THE SUM OF ALL PTOT ( SYSTEM PROBABILITY ). SYSP 170
18 C SYSP 180
19 C SUBROUTINE SYSP SYSP 190
20 C SYSP 200
21 C COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 SYSP 210
22 C COMMON/LGSUB/IB(15,2,50),IIS(50),IRB(50,30),ITEMP(41,20),N1, SYSP 220
23 C 1 ISAVE(50,200),ISUC,JSUC,JS1,JSAVE(50,200),IJS1,TOT,PROB(65), SYSP 230
24 C 2 PROBI(65),IPRINT,NSTDBY(15),NSBYMX,NOPRNT SYSP 240
25 C DOUBLE PRECISION PTOT,TOT,PROB,PROBI SYSP 250
26 C SYSP 260
27 C INITIALIZE. SYSP 270
28 C SYSP 280
29 C DO 122 I=1,L4 SYSP 290
30 C PROB(I)=PROBI(I) SYSP 300
31 C 122 CONTINUE SYSP 310
32 C TOT=0.00+0 SYSP 320
33 C SYSP 330
34 C ADD PROBABILITIES OF PATHS. SYSP 340
35 C SYSP 350
36 C DO 710 I=1,JS1 SYSP 360
37 C PTOT=1. SYSP 370
38 C DO 130 L=1,M7 SYSP 380
39 C ITEMP(1,L)=0 SYSP 390
40 C 130 CONTINUE SYSP 400
41 C SYSP 410
42 C MULTIPLY PROBABILITIES OF BLOCK NUMBERS IN A PATH. SYSP 420
43 C SYSP 430
44 C DO 711 J=1,M1 SYSP 440
45 C K=ISAVE(J,I) SYSP 450
46 C IF(K)108,710,108 SYSP 460
47 C SYSP 470
48 C CHECK PATH TO SEE IF ELIMINATED BY EQUIVALENT BLOCKS IN IT. SYSP 480
49 C (IF THE SIGN OF ANY TWO EQUIVALENT BLOCKS IN A SET ALTERNATES) SYSP 490
50 C SYSP 500
51 C 108 IF(N1)106,106,107 SYSP 510
52 C 107 DO 100 L=1,N1 SYSP 520
53 C DO 100 M=2,LC SYSP 530
54 C IF(K-ITEMP(M,L))100,101,100 SYSP 540
55 C 100 CONTINUE SYSP 550
56 C GO TO 106 SYSP 560
57 C 101 IF(ITEMP(1,L))102,103,104 SYSP 570
58 C 103 ITEMP(1,L)=K SYSP 580
59 C GO TO 106 SYSP 590
60 C 104 IF(K)105,109,109 SYSP 600
61 C 109 PROB(K)=1.0 SYSP 610
62 C GO TO 712 SYSP 620
63 C 105 K=-K SYSP 630
64 C PROB(K)=1.0 SYSP 640
65 C GO TO 110 SYSP 650
66 C 102 IF(K)111,112,112 SYSP 660
67 C 112 PROB(K)=0.0 SYSP 670
68 C GO TO 712 SYSP 680
69 C 111 K=-K SYSP 690
70 C PROB(K)=0.0 SYSP 700
71 C GO TO 110 SYSP 710
72 C 106 IF(K)714,711,712 SYSP 720
73 C 712 PTOT=PTOT*PROB(K) SYSP 730
74 C GO TO 730 SYSP 740
75 C 714 K=-K SYSP 750

```

```
76      110 PTOT=PTOT*(1.0-PROB(K))
77      730 PROB(K)=PROB1(K)
78      711 CONTINUE
79      710 TOT=TOT+PTOT
80      RETURN
81      END
```

```
SYSP 760
SYSP 770
SYSP 780
SYSP 790
SYSP 800
SYSP 810
```

```

TREA*FTREE.STBYPR
 1      C      FUNCTION STBYPR CALCULATES THE STANDBY TREE PROBABILITY FOR TIME SBPR 10
 2      C      T1 THRU TTOT. SBPR 20
 3      C      WRITTEN BY RGN ECKSTEIN 1/1/70 SBPR 30
 4      C      SBPR 40
 5      C      FUNCTION STBYPR(ISAVE,JS1,RO,TL,RT,KT,JT,TTOT,NSTEP,TLD) SBPR 50
 6      C      SBPR 60
 7      COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 SBPR 70
 8      DIMENSION ISAVE(M1,M2),RO(M1),TL(M1),RT(L4,LG1),TLD(M1) SBPR 80
 9      DOUBLE PRECISION TOT,PTOT,DPREC,STEP1,STEP2 SBPR 9C
10     C      SBPR 100
11     C      INITIALIZE SBPR 110
12     C      SBPR 120
13     STEP1=-TTOT*(KT-1)/NSTEP SBPR 130
14     STEP2=-TTOT*JT/NSTEP SBPR 140
15     TOT=0. SBPR 150
16     C      SBPR 160
17     DO 100 I=1,JS1 SBPR 170
18     PTOT=1.0 SBPR 180
19     DO 90 J=1,M1 SBPR 190
20     K=ISAVE(J,I) SBPR 200
21     IF(K)70,90,80 SBPR 210
22     70 K=-K SBPR 220
23     IF(K.GT.M1) GO TO 86 SBPR 230
24     DPREC=(STEP1*TL(K))+(STEP2*(TLD(K)-TL(K))) SBPR 240
25     PTOT=PTOT*(1.0-RO(K)*DEXP(DPREC)) SBPR 250
26     GO TO 90 SBPR 260
27     C      SBPR 270
28     C      CHECK FOR STANDBY BLOCK(51-65)/(L1-L4). SBPR 280
29     C      SBPR 290
30     80 IF(K.GT.M1) GO TO 85 SBPR 300
31     DPREC=(STEP1*TL(K))+(STEP2*(TLD(K)-TL(K))) SBPR 310
32     PTOT=PTOT*RO(K)*DEXP(DPREC) SBPR 320
33     GO TO 90 SBPR 330
34     85 PTOT=PTOT*RT(K,JT) SBPR 340
35     GO TO 90 SBPR 350
36     86 PTOT=PTOT*(1.0-RT(K,JT)) SBPR 360
37     90 CONTINUE SBPR 370
38     100 TOT=TOT+PTOT SBPR 380
39     STBYPR=TOT SBPR 390
40     RETURN SBPR 400
41     END SBPR 410

```

```

TREA*FTREE.DUPTRE
 1      C      SUBROUTINE DUPTRE DUPLICATES ISAVE INTO JSAVE AND SETS IJS1=JS1. DPTR 1C
 2      C      WRITTEN BY RGN ECKSTEIN 1/1/70 DPTR 20
 3      C      DPTR 3C
 4      C      SUBROUTINE DUPTRE(JSAVE,IJS1,ISAVE,JS1) DPTR 40
 5      C      DPTR 50
 6      C      COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 DPTR 60
 7      C      DIMENSION ISAVE(M1,M2),JSAVE(M1,M2) DPTR 7C
 8      C      DPTR 80
 9      C      DUPLICATE ISAVE INTO JSAVE. SET IJS1 = JS1. DPTR 9C
10     C      DPTR 100
11     C      DO 10 I=1,M2 DPTR 110
12     C      DO 10 J=1,M1 DPTR 120
13     C      JSAVE(J,I)=ISAVE(J,I) DPTR 130
14     C      10 CONTINUE DPTR 140
15     C      IJS1=JS1 DPTR 15C
16     C      RETURN DPTR 160
17     C      END DPTR 170

```



```

TREA*FTREE.STDBY2
 1.  C   SUBROUTINE STDBY2 IS FOR 2 OR MORE STANDBY BLOCKS CONTROLLED BY A STBY 10
 2  C   SENSE BLOCK. IT GENERATES THE STANDBY TREE, CALLS SUBROUTINE STBY 20
 3  C   TRINTR WHICH PUTS THE STANDBY BLOCK(51-65) IN PLACE OF THE FIRST STBY 30
 4  C   PATH OF THE STANDBY TREE IMBEDDED IN THE PATH OF THE ORIGINAL TREE STBY 40
 5  C   THEN ELIMINATES ORIGINAL TREE PATHS THAT CONTAIN 2ND-JS1 STANDBY STBY 50
 6  C   PATHS. THEN PUTS THE NEGATIVE STANDBY BLOCK(51-65)/(L1-L4) IN STBY 60
 7  C   AND ELIMINATES CORRESPONDING PATHS. WRITE STANDBY TREE ON TAPE. STBY 70
 8  C   WRITTEN BY RGN ECKSTEIN 1/1/70 STBY 80
 9  C   STBY 90
10  C   SUBROUTINE STDBY2(%NS) STBY 100
11  C   STBY 110
12  C   COMMON/ALLSUB/M1,M2,M3,M4,M5,M6,M7,L1,L3,L4,L5,L6 STBY 120
13  C   COMMON/LGSUB/IB(15,2,50),IIS(50),IRB(50,30),ITEMP(41,20),N1, STBY 130
14  C   1 ISAVE(50,200),ISUC,JSUC,JS1,JSAVE(50,200),IJS1,TGT,PROB(65), STBY 140
15  C   2 PROB1(65),IPRINT,NSTDBY(15),NSBYMX,NOPRNT STBY 150
16  C   COMMON/SMSUB/NPUF(200),NFUPMX,NBOUT(50),NBOTMX,NBIN(50),NBINMX, STBY 160
17  C   1 NBNUM(50),NBMAX STBY 170
18  C   DOUBLE PRECISION TGT,PROB,PRCB1 STBY 190
19  C   STBY 200
20  C   INITIALIZE STBY 210
21  C   STBY 220
22  C   DO 10 I=1,M1 STBY 230
23  C   NBIN(I)=0 STBY 240
24  C   10 CONTINUE STBY 250
25  C   NSS=0 STBY 260
26  C   STBY 270
27  C   GENERATE SENSE TREE STBY 280
28  C   STBY 290
29  C   ISUC=NS STBY 300
30  C   CALL TREE(%1100) STBY 310
31  C   STBY 320
32  C   STBY 330
33  C   GENERATE STANDBY TREE BY ZERCING OUT ALL BLOCK NUMBERS IN ISAVE STBY 340
34  C   EXCEPT THOSE STANDBY BLOCK NUMBERS CONTROLLED BY THE SENSE BLOCK. STBY 350
35  C   STBY 360
36  C   STBY 370
37  C   IEND=IRB(NS,1)+1 STBY 380
38  C   L=0 STBY 390
39  C   DO 60 I=2,IEND STBY 400
40  C   L=L+1 STBY 410
41  C   NBIN(L)=IRB(NS,I) STBY 420
42  C   60 CONTINUE STBY 430
43  C   NBINMX=L STBY 440
44  C   STBY 450
45  C   SUBROUTINE PATHIN ZEROS OUT ALL NUMBER BLOCKS IN ISAVE THAT ARE STBY 460
46  C   NOT EQUAL TO ANY OF THE NBIN ARRAY NUMBER BLOCKS. STBY 470
47  C   STBY 480
48  C   CALL PATHIN(%1100,ISAVE,JS1) STBY 490
49  C   STBY 500
50  C   SUBROUTINE SFLEFT SHIFTS PATHS IN ISAVE TO THE LEFT UNTIL NONZERO STBY 510
51  C   NB AT LEFT-MOST. FILL IN ZEROS AFTER SHIFTING. IF 50 ZEROS PUTS STBY 520
52  C   PATH NUMBER IN NPUP ARRAY AND CALLS PATHUP TO ELIMINATE. STBY 530
53  C   STBY 540
54  C   CALL SFLEFT(%1100,ISAVE,JS1) STBY 550
55  C   STBY 560
56  C   SUBROUTINE DUPOUT STORES PATH NUMBERS THAT ARE DUPLICATES IN NPUP STBY 570
57  C   AND CALLS PATHUP TO ELIMINATE DUPLICATE PATHS. STBY 580
58  C   STBY 590
59  C   CALL DUPOUT(%1100,ISAVE,JS1) STBY 600
60  C   STBY 610
61  C   SUBROUTINE NEGOUT STORES FAILURE PATH NUMBERS ENDING WITH A STBY 620
62  C   NEGATIVE SIGN IN NPUP ARRAY, THEN CALLS PATHUP TO ELIMINATE. STBY 630
63  C   STBY 640
64  C   CALL NEGOUT(%1100,ISAVE,JS1) STBY 650
65  C   STBY 660
66  C   NOW WE HAVE STANDBY TREE IN ISAVE. STBY 670
67  C   STBY 680
68  C   SUBROUTINE TRINTR PUTS THE STANDBY BLOCK(51-65) IN PLACE OF THE STBY 690
69  C   FIRST PATH OF THE STANDBY TREE IMBEDDED IN THE PATH OF THE STBY 700
70  C   ORIGINAL TREE IN JSAVE, THEN ELIMINATES ORIGINAL TREE PATHS THAT STBY 710
71  C   CONTAIN 2ND-JS1 STANDBY PATHS. THEN PUTS THE NEGATIVE STANDBY STBY 720
72  C   BLOCK(51-65)/(L1-L4) IN AND ELIMINATES CORRESPONDING PATHS. STBY 730
73  C   STBY 740
74  C   CALL TRINTR(%1100,JSAVE,IJS1,ISAVE,JS1,NSTDBY,NSBYMX,NSS) STBY 750
75  C   STBY 760

```

76	C	SAVE STANDBY TREE AND ITS STANDBY BLOCK(51-65)/(L1-L4).	STBY 770
77	C		STBY 780
78		WRITE(10) NSTDBY(NSBYMX), JS1	STBY 790
79		DO 900 I=1,M2	STBY 800
80		WRITE(10) (ISAVE(J,I),J=1,M1)	STBY 810
81	900	CONTINUE	STBY 820
82		RETURN	STBY 830
83	1100	PRINT 1101	STBY 840
84	1101	FORMAT(' SUBROUTINE CALLED FROM STDBY2.')	STBY 850
85		RETURN 1	STBY 860
86		END	STBY 870