IF (AGDEC(D).GE.YGDLG.AND.AGDEC(D).LE.YGDUG) GO TO 1500
IF (AGDEC(D).LT.YGD) GO TO 1480
IF (AGDEC(D).GT.YGDUG) GO TO 1490

1480 FAILG0=1
FAILG0=1
GO TO 1810

1490 FAILG0=0
FAILG0=1
GO TO 1820

1500 RM=RM-0.50
REND=RM+1.
D=-1
GUARD=0
RESF=0.10
GO TO 1170

1510 IF (RDEC(D).GE.-55.00.AND.RDEC(D).LT.-50.00) GO TO 1520
IF (RDEC(D).GE.-50.00.AND.RDEC(D).LT.-45.00) GO TO 1550
IF (RDEC(D).GE.-45.00.AND.RDEC(D).LT.-40.00) GO TO 1580
IF (RDEC(D).GE.-40.00.AND.RDEC(D).LT.-35.00) GO TO 1640
IF (RDEC(D).GE.-35.00.AND.RDEC(D).LE.-30.00) GO TO 1700
IF (RDEC(D).GT.0.AND.RDEC(D).LE.3.00) GO TO 1760

1520 YGD=2.20
IF (AGDEC(D).LE.YGD) GO TO 1530
IF (AGDEC(D).GT.YGD) GO TO 1540

1530 FAILG0=1
FAILG0=1
GO TO 1790

1540 FAILG0=0
FAILG0=1
GO TO 1820

1550 YGD=1.00
IF (AGDEC(D).LE.YGD) GO TO 1560
IF (AGDEC(D).GT.YGD) GO TO 1570

1560 FAILG0=1
FAILG0=1
GO TO 1790

1570 FAILG0=0
FAILG0=1
GO TO 1820

1580 YGD=(RDEC(D)+60.)/10.
YGD=1.00
IF (GDEC(D).GE.YGD) GO TO 1590
IF (GDEC(D).LT.YGD) GO TO 1600

1590 FAILG0=1
GO TO 1610

1600 FAILG0=0

1610 IF (AGDEC(D).LE.YGD) GO TO 1620
IF (AGDEC(D).GT.YGD) GO TO 1630

1620 FAILG0=1
IF (FAILG0.EQ.0) GO TO 1820
GO TO 1790

1630 FAILG0=0
GO TO 1820

1640 YGD=0.60*(MDEC(D)+60.)/18.
YGD=0.50
IF (GDEC(D).GE.YGD) GO TO 1650
IF (GDEC(D).LT.YGD) GO TO 1660

1650 FAILG0=1
GO TO 1670

1660 FAILG0=0

1670 IF (AGDEC(D).LE.YGD) GO TO 1680
IF (AGDEC(D).GT.YGD) GO TO 1690

1680 FAILG0=1
IF (FAILG0.EQ.0) GO TO 1820
1650 GO TO 1700
FAILRD=0
1660 GO TO 1820
YRD=36,
YBD=0.50
IF (GDEC(D),BL,YGU) GO TO 1710
IF (GDEC(D),LT,YBD) GO TO 1720
1710 FAILRD=1
GO TO 1730
1720 FAILRD=0
1730 IF (AGDEC(D),LE,YGU) GO TO 1740
IF (AGDEC(D),LT,YBD) GO TO 1750
1740 FAILRD=1
IF (FAILRD.EQ.0) GO TO 1820
GO TO 1790
1750 FAILRD=0
GO TO 1820
1760 YBD=0.50
IF (AGDEC(D),LE,YBD) GO TO 1770
IF (AGDEC(D),LT,YBD) GO TO 1780
1770 FAILRD=1
FAILRD=1
GO TO 1790
1780 FAILRD=0
FAILRD=1
GO TO 1820
1790 RM=RM+RES
IF (RM.GT.REND) GO TO 1800
GO TO 1170
1800 RM=RM-0.60
GUARD=1
1810 RM=RM+RES
IF (RM.GT.RMAX) GO TO 1820
GO TO 1170
1820 TYPE 1830
1830 FORMAT (20X, 'ENCODER UNDER TEST RESULTS')
IF (FAILGE.EQ.1.AND.FAILGE.EQ.1) GO TO 1840
IF (FAILGE.EQ.0.AND.FAILGE.EQ.0) GO TO 1860
IF (FAILGE.EQ.0) GO TO 1880
IF (FAILGE.EQ.0) GO TO 1900
1840 TYPE 1850
1850 FORMAT (10X, 'ENCODER PASSES TEST')
GO TO 1920
1860 TYPE 1870
1870 FORMAT (10X, 'ENCODER FAILS QD = GAIN TRACKING TEST')
GO TO 1920
1880 TYPE 1890
1890 FORMAT (10X, 'ENCODER FAILS ON QD TEST')
GO TO 1920
1900 TYPE 1910
1910 FORMAT (10X, 'ENCODER FAILS ON GAIN TRACKING TEST')
1920 TYPE 1930
1930 FORMAT (1X, 'INPUT LEVEL (DBM), 5X, 1
QUANTISATION DISTORTION 1 (DB), 5X, GAIN TRACKING (DB)')
DO 1950 N=1,C
1940 FORMAT (5X,F6.2,2X,F6.2,2X,F7.4)
1950 CONTINUE
TYPE 1960
FORMAT (20X, 'DECODER UNDER TEST RESULTS')
IF (FAILQD.EQ.1 .AND. FAILGD.EQ.1) GO TO 1970
IF (FAILQD.EQ.0 .AND. FAILGD.EQ.0) GO TO 1990
IF (FAILD.EQ.0) GO TO 2010
IF (FAILD.EQ.0) GO TO 2030
2020 TYPE 1980
2030 FORMAT (1DX, 'DECODER PASSES TEST')
GO TO 2050
2040 TYPE 2000
2050 FORMAT (1DX, 'DECODER FAILS GD + GAIN TRACKING TEST')
GO TO 2050
2060 TYPE 2020
2070 FORMAT (1DX, 'DECODER FAILS ON GD TEST')
GO TO 2050
2080 TYPE 2040
2090 FORMAT (1DX, 'DECODER FAILS ON GAIN TRACKING TEST')
2050 TYPE 1930
DO 2060 N=1,D
2010 TYPE 1940, RDEC(N), DDEC(N), DDEC(N)
2060 CONTINUE
8000 STOP
END
Subroutine IESTAT
SUBROUTINE IESTAT (IENC)
INTEGER N,S
REAL IENC(6,16)
DO 10 N=1,6
   DO 20 S=1,16
      IF (N.EQ.1) GO TO 30
      IENC(N,S)=2.*((N-2.)*((34.+2.)*(S-1.)))
   GO TO 20
   30   IENC(N,S)=2.*(S-1.)+2.
   20   CONTINUE
   10   CONTINUE
RETURN
END
Subroutine IDSTAT
SUBROUTINE IDSTAT (IDEC)
INTEGER N,S
REAL IDEC(8,16)
DO 10 N=1,8
DO 20 S=1,16
IF (N.EQ.1) GO TO 30
IDEC(N,S)=2.*(N-2.)*(33.+2.*(S-1.))
GO TO 20
30 IDEC(N,S)=2.*(S-1.)+1.
20 CONTINUE
10 CONTINUE
RETURN
END
Subroutine ESTAT1
SUBROUTINE ESTATI (ENC)
INTEGER N, S
REAL ENC(N,16)
      TYPE 5
      FORMAT (10X, +-----------------------------+')
      TYPE 6
      FORMAT (10X, 'HORIZONTAL SHIFT REQUIRED+')
      TYPE 7
      FORMAT (10X, '+-----------------------------+')
      ACCEPT 20, EC
      DO 210 N=1,8
      DO 220 S=1,16
      IF (N.EQ.1) GO TO 230
      ENC(N,S)=2.**(N-2.)*(34.+2.*(S-1.))
      ENC(N,S)=ENC(N,S)+EC*ENC(N,S)
      GO TO 220
      230 ENC(N,S)=2.*(S-1.)+2.
      ENC(N,S)=ENC(N,S)+EC*ENC(N,S)
      220 CONTINUE
      210 CONTINUE
      RETURN
      END
Subroutine ESTAT2
SUBROUTINE ESTAT2 (ENC)
INTEGER N, S
REAL ENC(8, 16)

5 FORMAT (10X, '--------------------------')
6 FORMAT (10X, 'OFFSET VALUE REQUIRED')
7 FORMAT (10X, '--------------------------')

ACCEPT 20, E0

20 FORMAT (F5.2)
DO 30 N = 1, 8
DO 40 S = 1, 16
IF (N.EQ.1) GO TO 50
ENC(N,S) = 2.**(N-2)*((34. + 2.* (S-1.)))
GO TO 40
30 CONTINUE
40 CONTINUE
DO 60 N = 1, 8
DO 70 S = 1, 16
IF (S.EQ.16) GO TO 70
IF (N.EQ.1) GO TO 71
IF (N.EQ.2) GO TO 71
IF (N.EQ.3) GO TO 72
IF (N.EQ.4) GO TO 73
IF (N.EQ.5) GO TO 74
IF (N.EQ.6) GO TO 75
IF (N.EQ.7) GO TO 76
IF (N.EQ.8) GO TO 77
GO TO 100
70 CONTINUE
71 ENC(N,S) = ENC(N,S) + E0*2.
GO TO 70
72 ENC(N,S) = ENC(N,S) + E0*4.
GO TO 70
73 ENC(N,S) = ENC(N,S) + E0*8.
GO TO 70
74 ENC(N,S) = ENC(N,S) + E0*16.
GO TO 70
75 ENC(N,S) = ENC(N,S) + E0*32.
GO TO 70
76 ENC(N,S) = ENC(N,S) + E0*64.
GO TO 70
77 ENC(N,S) = ENC(N,S) + E0*128.
70 CONTINUE
60 CONTINUE
100 RETURN
END
Subroutine ESTAT3
SUBROUTINE ESTAT3 (ENC)
INTEGER N,S
REAL ENC(8,16)
TYPE 5
5 FORMAT (1DX,'---------------------------')
TYPE 6
6 FORMAT (1DX,'+GAIN ERROR REQUIRED+')
TYPE 7
7 FORMAT (1DX,'---------------------------')
ACCEPT 20,EG
20 FORMAT (F4.2)
DO 30 N=1,8
DO 40 S=1,16
IF (N.EQ.1) GO TO 50
ENC(N,S)=2.*N-2.*(34.+2.*S-1.)
GO TO 40
50 ENC(N,S)=2.*S-1.+2.
30 CONTINUE
DO 60 N=1,8
DO 70 S=1,16
IF (S.EQ.16) GO TO 70
ENC(N,S)=EG*ENC(N,S)
70 CONTINUE
60 CONTINUE
RETURN
END
Subroutine DSTAT
SUBROUTINE DSTAT (NIDLD,ED,EDO,EDG)
INTEGER NIDLD
REAL ED,EDO,EDG

10 FORMAT (10X, '-----------------------------------------------')
TYPE 11
11 FORMAT (10X, 'SELECTING DECODER UNDER TEST NON-IDEALITIES')
TYPE 12
12 FORMAT (10X, ' 1=VERTICAL SHIFT')
TYPE 13
13 FORMAT (10X, ' 2=STEP GENERATOR OFFSET')
TYPE 14
14 FORMAT (10X, ' 3=STEP GENERATOR GAIN ERROR')
TYPE 15
15 FORMAT (10X, '-----------------------------------------------')
20 FORMAT (12)
IF (NIDLD.EQ.1) GO TO 21
IF (NIDLD.EQ.2) GO TO 22
IF (NIDLD.EQ.3) GO TO 23
GO TO 100
21 TYPE 30
30 FORMAT (10X, '-----------------------------------------------')
TYPE 31
31 FORMAT (10X, 'OFFSET VALUE REQUIRED')
ACCEPT 40,ED
40 FORMAT (F6.4)
GO TO 100
22 TYPE 50
50 FORMAT (10X, '-----------------------------------------------')
TYPE 51
51 FORMAT (10X, 'OFFSET VALUE REQUIRED')
ACCEPT 60,EDO
60 FORMAT (F5.2)
GO TO 100
23 TYPE 70
70 FORMAT (10X, '-----------------------------------------------')
TYPE 71
71 FORMAT (10X, 'GAIN ERROR REQUIRED')
ACCEPT 80,EDG
80 FORMAT (F4.2)
100 RETURN
END
Subroutine IENCOD
SUBROUTINE IENCOD (IENC, VS, VPE)
INTEGER N,S
REAL IENC(8,16), VS, VPE
X=ABS(VS)
IF (X.LT.2.) GO TO 10
IF (X.GE.4096.) GO TO 20
GO TO 30
10   VPE=0
    GO TO 100
20   VPE=4096.
    GO TO 90
30   DO 40 N=1,6
     DO 50 S=1,16
        IF (S.EQ.16) GO TO 60
        IF (X.GE.IENC(N,S).AND.X.LT.IENC(N,S+1)) GO TO 70
        GO TO 50
60   IF (X.GE.IENC(N,S).AND.X.LT.IENC(N+1,1)) GO TO 70
     GO TO 50
70   VPE=IENC(N,S)
     GO TO 90
50   CONTINUE
40   CONTINUE
90   IF (VS.LT.0) GO TO 95
     GO TO 100
95   VPE=VPE*(-1.)
100   RETURN
END
SUBROUTINE IDECOD (IDEC, V, VPD)
INTEGER N, S
REAL IDEC(B, 16), V, VPD
X = ABS(V)
IF (X.LT.1.) GO TO 10
GO TO 20
10 VPD = 1.
GO TO 90
20 DO 30 N = 1, 8
DO 40 S = 1, 16
IF (S.EQ.16) GO TO 50
IF (X.GE.IDEC(N, S) .AND. X.LT.IDEC(N, S+1)) GO TO 60
GO TO 40
50 IF (X.GE.IDEC(N, S) .AND. X.LT.IDEC(N+1, 1)) GO TO 70
GO TO 40
60 VPD = IDEC(N, S+1)
GO TO 90
70 VPD = IDEC(N+1, 1)
GO TO 90
40 CONTINUE
30 CONTINUE
90 IF (V.LT.0) GO TO 95
GO TO 100
95 VPD = VPD * (-1.)
100 RETURN
END
Subroutine PENCOD
SUBROUTINE PENCOD (ENC, VS, VPE)
INTEGER N, S
REAL ENC(8,16), VS, VPE
X = ABS(VS)
IF (X.LT.2.) GO TO 10
IF (X.GE.4096.) GO TO 20
GO TO 30
10 VPE = 0
GO TO 100
20 VPE = 4096.
GO TO 90
30 DO 40 N = 1, 8
DO 50 S = 1, 16
IF (S.EQ.16) GO TO 60
IF (X.GE.ENC(N,S).AND.X.LT.ENC(N,S+1)) GO TO 70
GO TO 50
50 IF (X.GE.ENC(N,S).AND.X.LT.ENC(N+1,1)) GO TO 70
GO TO 50
60 VPE = ENC(N,S)
GO TO 90
70 CONTINUE
40 CONTINUE
90 IF (VS.LT.0) GO TO 95
GO TO 100
95 VPE = VPE*(-1.)
100 RETURN
END
Subroutine PDECOD
SUBROUTINE PDECOD (IDEC, V, NIDLD, ED, EDO, EDG, VPD)
INTEGER N,S,NIDLD
REAL IDEC(8,16),V,VPD
X=ABS(V)
IF (X.LT.1.) GO TO 10
GO TO 20
10 VPD=1.
GO TO 80
20 DO 30 N=1,8
DO 40 S=1,16
IF (S.EQ.16) GO TO 50
IF (X.GE.IDEC(N,S).AND.X.LT.IDEC(N,S+1)) GO TO 60
GO TO 40
30 CONTINUE
40 CONTINUE
50 IF (NIDLD.EQ.1) GO TO 81
IF (NIDLD.EQ.2) GO TO 82
IF (NIDLD.EQ.3) GO TO 83
GO TO 200
80 VPD=VPD+ED*VPD
GO TO 100
81 VPD=VPD+EDO*2.
GO TO 100
82 VPD=VPD+EDO*4.
GO TO 100
83 VPD=VPD+EDO*8.
GO TO 100
84 VPD=VPD+EDO*16.
GO TO 100
85 VPD=VPD+EDO*32.
GO TO 100
86 VPD=VPD+EDO*64.
GO TO 100
87 VPD=VPD+EDO*128.
GO TO 100
88 IF (S.EQ.16) GO TO 100
VPD=EDG*VPD
100 IF (V.LT.0) GO TO 150
GO TO 200
150 VPD=VPD*(-1.)
200 RETURN
END
Author  Debbo G A
Name of thesis  A proposed test system and strategy for the production testing of pulse code modulation codecs  1984

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.