

FEDERAL STANDARD

TELECOMMUNICATIONS: ANALOG TO DIGITAL CONVERSION OF RADIO
VOICE BY 4,800 BIT/SECOND CODE EXCITED LINEAR PREDICTION (CELP)

This standard is issued by the General Services Administration pursuant to the Federal Property and Administrative Services Act of 1949, as amended.

1. SCOPE

1.1 **Description.** This standard specifies **interoperability** -related requirements for the conversion of analog voice to a 4,800 bit/s digitized form for digital radio transmission by a method known as Code Excited Linear Prediction (CELP) and the reverse process, the synthesis of analog voice from 4,800 bit/s CELP-digitized voice. In addition to digital radio applications, CELP is also very suitable for encrypted telephone use and other applications wherein voice must be digitized prior to encryption.

1.2 **Purpose.** This standard is to facilitate interoperability between radio telecommunication facilities and systems of the Federal Government.

1.3 **Application.** This standard shall be used by all Federal departments and agencies in the design and procurement of all radio equipment employing 4,800 bit/s digitized voice.

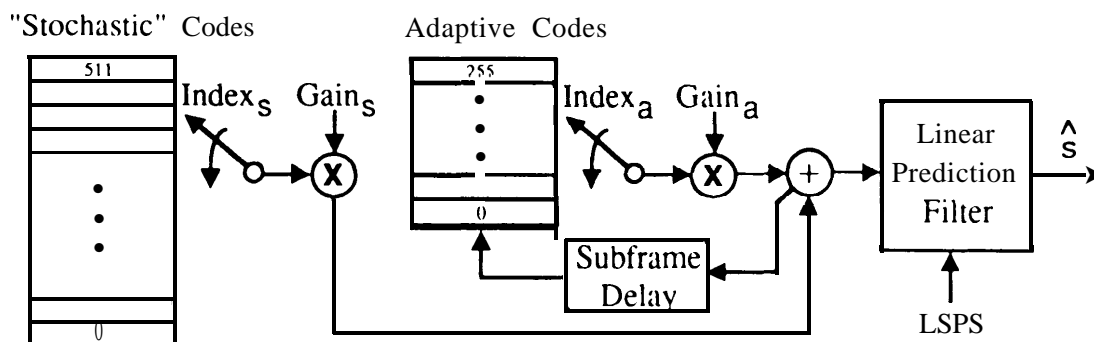
2. CONVENTIONS AND DEFINITIONS

a. **Frame.** A CELP frame is 144 bits in length. A frame interval is 30 ms \pm .01 percent in duration and contains 240 voice samples (8,000 samples/s).

b. **Subframe.** A CELP subframe is 1/4 the length of a frame. Thus, a subframe is 7.5 ms \pm .01 percent in duration and contains 60 voice samples.

3. REQUIREMENTS3.1 Voice Synthesis

3.1.1 **General Description.** Since Code Excited Linear Prediction (CELP) is an analysis-by-synthesis type of technique, voice synthesis is described first. As shown in the diagram below, CELP synthesis involves the excitation of a parameter-adjusted Linear Prediction Filter by the sum of gain-scaled codes selected from a fixed, **stochastically**- derived "stochastic code book" and an adaptive "code book," utilizing parameters transmitted in a 144-bit frame structure.



The fixed, stochastically-derived "stochastic" codes are described in section 3.3. "Stochastic" code gain is described in section 3.4. Adaptive codes are described in section 3.5 and adaptive code gain is described in section 3.6. Section 3.7 describes the Linear Prediction Filter and the Line Spectral Parameters (LSPS) that adjust it. Section 3.8 shows transmission format, including bit assignments for the transmitted information. Single bit error correction on some of the most sensitive bits is described in section 3.9.

3.1.2 Postfiltering. **Postfiltering** may be used to enhance the synthesized voice coming out of the CELP Linear Prediction Filter.

3.1.3 Lowpass Filtering. Lowpass (i.e., reconstruction) filtering shall be employed. A typical lowpass filter has a 3 dB attenuation point at 3,800 Hz, less than 1 dB of passband ripple, and minimum attenuations of 18 dB at 4,000 Hz and 46 dB above 4,400 Hz. In certain applications, mild highpass filtering (e.g., 175 Hz second-order Butterworth) may also be of benefit.

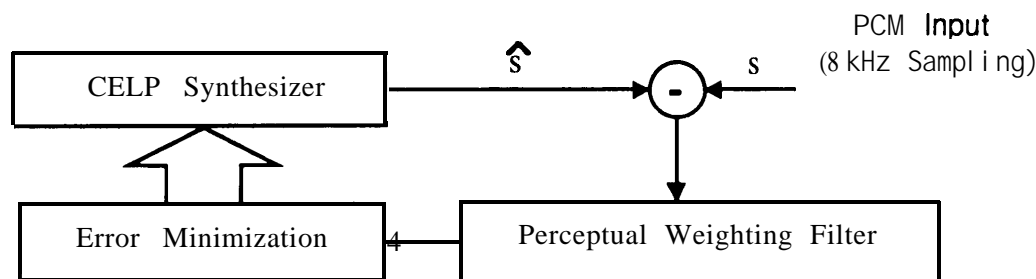
3.2 Voice Analysis

3.2.1 Voice Input Filtering. The analog voice input bandpass should be essentially flat from 200 to 3,400 Hz. A typical input filter has 3 dB attenuation points at 100 and 3,800 Hz; less than 1 dB of inband ripple; and minimum attenuations of 18 dB at 50 Hz, 18 dB at 4,000 Hz, and 46 dB above 4,400 Hz.

3.2.2 Analog-to-Digital Conversion. Analog-to-digital conversion shall use an 8 kHz \pm 0.1 percent sampling frequency and have a dynamic range of at least 12 bits.

3.2.3 Amplitude Scaling. To maintain proper receiver voice levels, analysis shall be based upon use of input digitized voice whose peak values are -32,768 and +32,767.

3.2.4 Analysis. As reflected in the following diagram, CELP is an analysis-by-synthesis type of



technique. The objective of **CELP analysis** is to minimize the perceptual difference (i.e., find the best match) between the actual digitized voice and the synthesized voice resulting from use of the parameters to be transmitted. As shown in the diagram, linear Pulse Code Modulated (PCM) voice sampled at 8 kHz is subtracted from the CELP synthesizer approximation and passed through a perceptual weighting filter (see section 3.2.5). The synthesizer parameters to be transmitted are adjusted for minimum perceptual error with respect to the actual input voice signal.

3.2.5 Perceptual Weighting Filter. It is recommended that a perceptual weighting filter be the cascade of a linear predictive whitening filter and a bandwidth expanded linear predictive synthesis filter. The bandwidth expanded linear predictive synthesis filter's poles are moved radially toward the origin by a weighting factor, typically 0.8.

3.3 "Stochastic" Codes. There are 512 fixed, **stochastically-derived** codes (i.e., vectors). During voice analysis, a code may be selected from a set smaller than 512 in order to reduce computational complexity (at the expense of voice reproduction quality). All 512 fixed, stochastically-derived codes

shall be available for voice synthesis. Each of the fixed, **stochastically-derived** codes contains 60 ternary (i.e., either -1, 0, or 1) elements, representing information used to form the excitation for the Linear Prediction Filter over a **subframe** period (i.e., 8,000 elements/s for 7.5 ins). The fixed, **stochastically-derived** codes, 512 overlapped codes of 60 elements each, are created from a 1,082 element "stochastic code book" as described below.

3.3.1 "Stochastic Code Book". The 1,082 element "stochastic code book" is defined by the following FORTRAN computer program.

```

program codebook
  implicit none
  integer M, L, MAXCODE, WIDTH, j, k
  parameter (M=512, L=60)
  parameter (MAXCODE=2*(M-1)+L)
  parameter (WIDTH= 20)
  real x( 0: MAXCODE+1), THRESH
  parameter (THRESH=1 .2)
  open (unit=10, file= 'codebook.h', status=' new' )
  do 50 k=0, MAXCODE, 2
    call noise(x(k),x(k+1))
    do 20 j=0, 1
      if(abs(x(k+j)).lt. THRESH) then
        x(k+j) = 0.0
      else
        x(k+j) = sign(1.0, x(k+j))
      end if
    20 continue
  50 continue
  do 80 k=1, MAXCODE-WIDTH, WIDTH
    write (10, 90) (int(x(j)), j=k, k+WIDTH-1)
  80 continue
  write (10, 91) (int(x(j)), j=MAXCODE/WIDTH*WIDTH+1, MAXCODE)
  stop 'codebook.h generated'
  90 format(1x, 20(i3, ', '))
  91 format(1x, 20(i3:, ', '))
end
subroutine noise(x1, x2)
  implicit none
  real x1, x2
  integer random, i, j
  real f(2), f1, f2, s
  10 do 30 i=1, 2
    do 20 j=1, 4
      f(i) = (float (random() + 32768 ))/65535.
    20 continue
  30 continue
  f1=2.*f(1)-1.
  f2=2.*f(2)-1.
  s=f1*f1+f2*f2
  if(s .ge. 1.) goto 10
  s=sqrt(-2.*slog(s)/s)
  x1=f1*s
  x2=f2*s
  return
end

```

FED-STD 1016

```

function random ( )
implicit none
integer random
integer MIDTAP, MAXTAP
parameter (MIDTAP=2, MAXTAP=5)
integer y(MAXTAP), j, k, temp
save y, j, k
data y/-21161, -8478, 30892, -10216, 16950/
data j/MIDTAP/, k/MAXTAP/
temp=iand(y(k)+y(j), 65535)
if (temp .gt. 32767) temp=temp-65536
y(k)=temp
random=temp
k=k-1
if (k l e. 0) k=MAXTAP
j=j-1
if (j l e. 0) j=MAXTAP
return
end

```

The first and last 200 elements generated by the above FORTRAN program are shown below. The left-most and highest elements are lowest numbered (e.g., the element in the first row and third column is numbered 2).

```

o, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, -1,
0, 0, -1, 0, -1, 0, 1, 0, 0, 0, 0, 0, 0, 0, -1, 0, -1, -1, 0, 0,
-1, 0, -1, 0, -1, 0, -1, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 1, 0,
0, 0, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, -1, 0,
1, 0, 1, 0, -1, 1, 0, 0, 0, 0, 0, -1, 0, 0, 1, 0, 0, 0, -1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, -1,
0, 1, -1, 0, 0, -1, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
o, 0, 0, -1, 0, -1, -1, 0, 0, -1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
0, -1, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, -1, 0, 1, 0, 0, 0, 0, 0,

```

.....

```

0, 1, 0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 1, 0, 0, 0, 0, -1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 1, 0, 0, 1, 0, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, -1, 0, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, -1, 1, -1, 0, 1, 0, 0, 0,
o, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 0, 0, 0, -1, 0,
0, 0, -1, 0, -1, 0, 0, 0, 0, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0,
0, 0, 0, 1, -1, 0, 0, 0, 0, -1, 0, 1, 0, 0, 1, 0, 0, 0, -1,
0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0, 1,
0, 0,

```

3.3.2 Creation of **Stochastically-derived** Codes. The 512 fixed codes of 60 elements each are assembled from the **1,082 stochastically-derived "stochastic code book" elements** as shown below. The left-most code elements are first-most in time.

Index	Elements In Codes	
511	0, 1, 2,... 58, 59
510	2, 3, 4,... 60, 61
.		
n	$2(511-n)$, $2(511-n)+1$,.. $2(511-n)+59$
.		
.		
1	1,020, 1,021, 1,022,... 1,078, 1,079
0	1,022, 1,023, 1,024,... 1,080, 1,081

3.4 “Stochastic” Code Gain. The relative amplitude (to the nearest table value) to be applied to the **stochastically-derived** code elements of each **subframe** is determined during analysis and coded into 5 bits according to the following table. The decimal index number is then transmitted in binary form. At the receiver, in voice synthesis, the gain index number is used to decode the relative amplitude of the **stochastically** - derived code elements during the **subframe**.

Index	Gain	Index	Gain	Index	Gain	Index	Gain
0	-1,330	8	-178	16	1	24	224
1	-870	9	-136	17	3	25	278
2	-660	10	98	18	13	26	340
3	-520	11	-64	19	35	27	418
4	-418	12	-35	20	64	28	520
5	-340	13	-13	21	98	29	660
6	-278	14	-3	22	136	30	870
7	-224	15	-1	23	178	31	1,330

3.5 Adaptive Codes

3.5.1 Adaptive “Code Book” and Integer-delay Codes. The adaptive “code book” is a 147-element, shifting storage register that is updated at the start of every **subframe** with the previous 60 elements (**subframe** interval) of Linear Prediction Filter excitation. Element ordering is such that the first excitation elements into the Linear Prediction Filter are the first into the adaptive “code book”. Elements already in the storage register are shifted up during the update process and the oldest elements are discarded. The 128 integer-delay overlapped adaptive codes, of 60 elements each, are generated from the information in the adaptive “code book” as follows. Adaptive codes 60 through 147 are composed of elements -60 to -1, -61 to -2, . . . through -147 to -88, respectively (where element -1 was the last element into the adaptive “code book”). Adaptive code “n”, where n ranges from 20 to 59, repeats the adaptive “code book” elements sequentially from -n to -1 to form a 60-element code. As examples, adaptive code 20 repeats adaptive “code book” elements -20 ...-1 three times and adaptive code 59’s elements run -59 . . . -1, -59. Regarding order of utilization, the most negatively numbered (i.e., most delayed) element of a particular adaptive code is used to compute the first element in time of Linear Prediction Filter excitation for a particular subframe period.

3.5.2 Noninteger-delay Codes. An additional 128 implicit, noninteger-delay codes, defined in section 3.5.4, can optionally be made available through interpolation during voice analysis. All 256 adaptive codes shall be available for voice synthesis. Forty point interpolation, or the equivalent, shall be used in both voice analysis and voice synthesis. (However, this does not preclude interpolating with fewer points, perhaps as few as 8 points, during preliminary code search computations). The process of interpolation is described in section 3.5.3.

3.5.3 Interpolation. Anoninteger-delay adaptive code is interpolated from the nearest lower-valued integer-delay code (e.g., noninteger-delay code 49.67 is determined by interpolating from integer-delay code 49). The 60 elements of the integer-delay code are first renumbered as $R_0 \dots R_{59}$ for purposes of description, where R_0 is normally the most negatively numbered (i.e., most delayed) element under the previous numbering system (i.e., -1 to -147), R_1 is the next most negatively numbered element, etc. For example, for integer-delay code 66, R_0 =element -66, R_1 =element -65, and R_{59} =element -7. For integer-delay code 21, because of the repetition within lower numbered integer-delay codes, R_0 =element -21, R_1 =element -20, R_{58} =element -4, and R_{59} =element -3. Likewise, for integer-delay code 58, R_0 =element -58, R_1 =element -57, R_{57} =element -1, **R_{58} =element -58**, and R_{59} =element -57.

The next step in explaining the process of interpolation is to show the computation of the weighting factors. The following FORTRAN program gives interpolation weighting factors for the interpolation of even numbers of points (N) equal to and less than 40. Note: interpolation with less than 8 points is not recommended and interpolation with other than 40 points is only for preliminary computations during voice analysis.

```

program weights
implicit none
integer N, nf, i, j, k
parameter (N=40, nf=5)
real h(-6*N:6*N), w(-N/2:N/2-1, nf), f(nf), pi
data f/0.25, 0.33333333, 0.5, 0.66666667, 0.75/
pi = 4.0*atan(1.0)
open(unit=10, file='weights', status='new')
do 10 k = -6*N, 6*N
    h(k) = 0.54 + 0.46* cos(pi*k/(6*N))
10 continue
do 30 i = 1, nf
    do 20 j = -N/2, N/2-1
        w(j, i) = h(nint(12*(j+f(i)))) * sin(pi*(j+f(i))) / (pi*(j+f(i)))
20 continue
30 continue
write (10,69) f
do 40 j = -N/2, N/2-1
    write (10,70) j, (w(j, k), k = 1, nf)
40 continue
69 format (7x, 5f14.5)
70 format(3x, i4, 5e14.5)
end

```

For N-point interpolation, the above program yields N weighting factors, numbered from $-N/2$ to $N/2-1$. As the final step showing the process of interpolation, these weighting factors are multiplied by the values of the adaptive code elements within a "window" around the element being interpolated (R). The sum of these multiplications is the interpolated value of that element (R'). For example, for 40 point interpolation, to find noninteger-delay code 66.67 (i.e., integer= 66, **fraction= .67**), element -66 becomes R_0 . Weighting factor -20 (i.e., -.001 1766) is multiplied by the value of element -86, weighting factor -19 (i.e., .0014386) is multiplied by the value of element -85, weighting factor 0 (i.e., .41245) is multiplied by the value of element -66, . . . , up to weighting factor 19 (i.e., -.001 1302) being multiplied by the value of element -47. The sum of these becomes the interpolated value of element -66 (i.e., R'_0). The same operation takes place in a "window" around the remaining elements (**$R_1 - R_{59}$**). However, in the example given, the samples run out beyond R_{46} (i.e., for R_{46} weighting factor 19 is multiplied by the value of element -1). In such cases, the previously interpolated values for elements R_0, R_1, R_2, \dots (i.e., R'_0, R'_1, R'_2, \dots) are used to complete interpolation of the remaining elements. Thus, in the example given, when interpolating R_{58} (i.e., element -8), weighting factor -20 is multiplied by the value of element -28 and weighting factor 19

is multiplied by the value of already interpolated element R'11 (i.e., interpolated element -56). The sum of the 40 values found by multiplying the 40 weighting factors by the 40 element values (i.e., element -28, element -27, element -26, . . . element -1, R'0, R*1, . . . R'10, R'11) becomes the value of R'58 (i.e., interpolated element -8).

3.5.4 Coding Within 1st and 3rd Subframes for Transmission. The following table identifies the 8-bit values transmitted to represent each of the 256 (128 integer-delay and 128 noninteger-delay (implicit)) adaptive codes. The notation is such that, for example, hexadecimal (\$) AF is expressed as 10101111 in binary form, where bit 7 is on the left and bit 0 is on the right.

Adaptive Code	Hex	Adaptive Code	Hex	Adaptive Code	Hex	Adaptive Code	Hex	Adaptive Code	Hex
20.00	\$42	34.67	\$C0	51.67	\$98	68.67	\$C5	97.00	\$A1
20.33	\$46	35.00	\$C3	52.00	\$90	69.00	\$C9	98.00	\$97
20.67	\$47	35.33	\$C2	52.33	\$80	69.33	\$C8	99.00	\$87
21.00	\$57	35.67	\$D2	52.67	\$9A	69.67	\$C7	100.0	\$9F
21.33	\$56	36.00	\$D3	53.00	\$8A	70.00	\$CB	101.0	\$8F
21.67	\$59	36.33	\$D1	53.33	\$82	70.33	\$C6	102.0	\$81
22.00	\$58	36.67	\$D0	53.67	\$92	70.67	\$CA	103.0	\$91
22.33	\$AE	37.00	\$30	54.00	\$1A	71.00	\$D6	104.0	\$9B
22.67	\$BE	37.33	\$32	54.33	\$12	71.33	\$DA	105.0	\$8B
23.00	\$BA	37.67	\$3A	54.67	\$00	71.67	\$DB	106.0	\$83
23.33	\$B8	38.00	\$31	55.00	\$08	72.00	\$D7	107.0	\$93
23.67	\$BC	38.33	\$33	55.33	\$06	72.33	\$D9	108.0	\$18
24.00	\$AC	38.67	\$3B	55.67	\$OE	72.67	\$D5	109.0	\$10
24.33	\$A8	39.00	\$3F	56.00	\$OF	73.00	\$D8	110.0	\$04
24.67	\$94	39.33	\$37	56.33	\$07	73.33	\$D4	111.0	\$Oc
25.00	\$84	39.67	\$3E	56.67	\$17	73.67	\$20	112.0	\$16
25.33	\$8C	40.00	\$36	57.00	\$1F	74.00	\$28	113.0	\$1E
25.67	\$9C	40.33	\$34	57.33	\$OD	74.33	\$38	114.0	\$14
26.00	\$9E	40.67	\$4A	57.67	\$05	74.67	\$22	115.0	\$1C
26.25	\$8E	41.00	\$4B	58.00	\$1D	75.00	\$2A	116.0	\$F9
26.50	\$86	41.33	\$4E	58.33	\$15	75.33	\$39	117.0	\$FA
26.75	\$96	41.67	\$4F	58.67	\$FB	75.67	\$29	118.0	\$FD
27.00	\$0A	42.00	\$5F	59.00	\$FF	76.00	\$21	119.0	\$E9
27.25	\$02	42.33	\$5E	59.33	\$EB	76.33	\$23	120.0	\$FE
27.50	\$0B	42.67	\$5C	59.67	\$EF	76.67	\$2B	121.0	\$E8
27.75	\$03	43.00	\$5D	60.00	\$ED	77.00	\$27	122.0	\$FC
28.00	\$1B	43.33	\$54	60.33	\$EA	77.33	\$2F	123.0	\$43
28.25	\$13	43.67	\$55	60.67	\$EE	77.67	\$25	124.0	\$F2
28.50	\$09	44.00	\$50	61.00	\$EC	78.00	\$2D	125.0	\$F6
28.75	\$01	44.33	\$51	61.33	\$E6	78.33	\$3D	126.0	\$F8
29.00	\$19	44.67	\$AA	61.67	\$E2	78.67	\$35	127.0	\$5B
29.25	\$11	45.00	\$A6	62.00	\$E4	79.00	\$3C	128.0	\$5A
29.50	\$F3	45.33	\$A2	62.33	\$E0	79.33	\$2E	129.0	\$63
29.75	\$F7	45.67	\$B6	62.67	\$F4	79.67	\$2C	130.0	\$62
30.00	\$E7	46.00	\$B2	63.00	\$F0	80.00	\$26	131.0	\$77
30.25	\$E3	46.33	\$BB	63.33	\$60	81.00	\$24	132.0	\$76
30.50	\$E5	46.67	\$B0	63.67	\$64	82.00	\$49	133.0	\$52
30.75	\$E1	47.00	\$B9	64.00	\$74	83.00	\$48	134.0	\$53
31.00	\$F1	47.33	\$B4	64.33	\$70	84.00	\$4C	135.0	\$66
31.25	\$F5	47.67	\$BD	64.67	\$73	85.00	\$4D	136.0	\$67
31.50	\$61	48.00	\$A4	65.00	\$72	86.00	\$44	137.0	\$CC
31.75	\$65	48.33	\$A0	65.33	\$6C	87.00	\$45	138.0	\$CD
32.00	\$75	48.67	\$A9	65.67	\$7C	88.00	\$40	139.0	\$AB

(continued)

FED-STD 1016

32.25 \$71	49.00 \$AD	66.00 \$68	89.00 \$41	140.0 \$CF
32.50 \$6D	49.33 \$95	66.33 \$78	90.00 \$A7	141.0 \$CE
32.75 \$7D	49.67 \$85	66.67 \$7A	91.00 \$A3	142.0 \$DE
33.00 \$69	50.00 \$9D	67.00 \$7E	92.00 \$B7	143.0 \$BF
33.25 \$79	50.33 \$8D	67.33 \$6A	93.00 \$B3	144.0 \$DF
33.50 \$7B	50.67 \$89	67.67 \$6E	94.00 \$B1	145.0 \$DD
33.75 \$7F	51.00 \$99	68.00 \$6F	95.00 \$B5	146.0 \$DC
34.00 \$6B	51.33 \$88	68.33 \$C4	96.00 \$A5	147.0 \$AF
34.33 \$C1				

Note: Before an adaptive code is selected for transmission, consideration should be given to the possibility that a multiple of actual "pitch" is being selected. Submultiple of a selected code's pitch equivalent should be reexamined and maybe utilized instead if results are found to be within approximately 0.5 dB of a selected code's squared prediction error.

3.5.5 Coding Within 2nd and 4th **Subframes** for Transmission. The adaptive code selected for transmissions represented in 6 bits, based upon the adaptive code selected in the previous **subframe**. Utilizing the table of adaptive codes in section 3.5.4, if a previous subframe's adaptive code was in the range of 20.00 - 29.25, the adaptive code to be transmitted could run from 20.00 to 38.33. If the previous **subframe's** adaptive code was in the range of 115.0 - 147.0, the adaptive code could run from 84.00 to 147.0. otherwise, the 6 bits will code the range from 31 codes lower to 32 codes higher than the previous **subframe's** adaptive code (considering both integer-delay and noninteger-delay adaptive codes). This is true even if an implementation uses only integer adaptive codes. Binary coding is such that the lowest numbered adaptive code is coded as 000000 and the highest numbered adaptive code is coded as 111111. For example, for previous **subframe** adaptive codes between 29.50 and 114.0, binary 011111 indicates no difference from the adaptive code of the previous **subframe**.

3.6 **Adaptive Code Gain.** The relative amplitude (to the nearest table value) to be applied to the adaptive code elements of each **subframe** is determined during analysis and coded into 5 bits according to the following table. The decimal index number is then transmitted in binary form. At the receiver, in voice synthesis, the adaptive code gain index number is used to decode the relative amplitude of the adaptive code elements during the **subframe**.

Index	Gain	Index	Gain	Index	Gain	Index	Gain
0	-.993	8	0.255	16	0.780	24	1.062
1	-.831	9	0.368	17	0.816	25	1.117
2	-.693	10	0.457	18	0.850	26	1.193
3	-.555	11	0.531	19	0.881	27	1.289
4	-.414	12	0.601	20	0.915	28	1.394
5	-.229	13	0.653	21	0.948	29	1.540
6	0.000	14	0.702	22	0.983	30	1.765
7	0.139	15	0.745	23	1.020	31	1.991

3.7 Linear Prediction

3.7.1 Linear Prediction Filter. A 10th order Linear Prediction Filter is excited by the sum of the gain-adjusted "stochastic" and adaptive codes to synthesize voice.

3.7.2 Line Spectral Parameters. During voice analysis, the parameters of the Linear Prediction Filter are determined and transmitted on a 30 ms frame basis as 10 Line Spectral Parameters (LSPs), on the basis of 10th order linear prediction. It is recommended that LSPs be determined with no preemphasis, 15 Hz bandwidth expansion (i.e., 0.994 weighting factor), and a 30 ms nonoverlapped Hamming window spanning the last two **subframes** of the present frame and first two **subframes** of the next frame. Shown below is the coding to be employed for the 10 LSPs. Note: LSPs are

expressed in terms of frequency (Hz) and must be monotonic (i.e., higher numbered LSPS must not contain lower frequencies than lower numbered LSPS).

Index	LSP1 (Hz)	LSP2 (Hz)	LSP3 (Hz)	LSP4 (Hz)	LSP5 (Hz)	LSP6 (Hz)	LSP7 (Hz)	LSP8 (Hz)	LSP9 (Hz)	LSP10 (Hz)
0	100	210	420	620	1,000	1,470	1,800	2,225	2,760	3,190
1	170	235	460	660	1,050	1,570	1,880	2,400	2,880	3,270
2	225	265	500	720	1,130	1,690	1,960	2,525	3,000	3,350
3	250	295	540	795	1,210	1,830	2,100	2,650	3,100	3,420
4	280	325	585	880	1,285	2,000	2,300	2,800	3,200	3,490
5	340	360	640	970	1,350	2,200	2,480	2,950	3,310	3,590
6	420	400	705	1,080	1,430	2,400	2,700	3,150	3,430	3,710
7	500	440	775	1,170	1,510	2,600	2,900	3,350	3,550	3,830
8		480	850	1,270	1,590					
9		520	950	1,370	1,670					
10		560	1,050	1,470	1,750					
11		610	1,150	1,570	1,850					
12		670	1,250	1,670	1,950					
13		740	1,350	1,770	2,050					
14		810	1,450	1,870	2,150					
15		880	1,550	1,970	2,250					

3.7.3 LSP Weighted Averaging. To determine the LSP values to be used for each subframe, weighted averaging is used between the LSPS transmitted with the previous frame and the LSPS transmitted with the present frame. For each of the **subframes** of a given frame, the following weighting is given to the LSPS transmitted with the previous and present frames before they are utilized.

Subframe	Previous LSPS	Present LSPs
1	7/8	1/8
2	5/8	3/8
3	3/8	5/8
4	1/8	7/8

Thus, for example, if LSP 1 of a frame is 250 Hz and LSP 1 of the previous frame was 340 Hz, the value used for **subframe1** of the present frame is: $7/8(340 \text{ Hz}) + 1/8(250 \text{ Hz}) = 328.75 \text{ Hz}$. For each **subframe**, this method of weighted averaging is employed for all 10 LSPS.

3.8 Transmission Format

3.8.1 Transmission Rate. The transmission rate for CELP shall be 4,800 bits/s ± 0.1 percent.

3.8.2 Bit Assignment. The table below gives the assignment of the 144 bits in each CELP frame (30 ms at 4,800 bits/s). Order of transmission is from bit 1 to bit 144. Abbreviations used in the following table are:

i = bit number, with 0 being the least significant bit

n = **subframe** number

CG(n)-i = Fixed, **Stochastically-derived** Code Gain

CI(n)-i = Fixed, **Stochastically-derived** Code Index

HP-i = Parity

LSP j-i = Line Spectral Parameter (**LSP**), where j = LSP number

PD(n)-i = Adaptive Code Index

PG(n)-i = Adaptive Code Gain

SP = Expansion Bit

SY = Synchronization Bit

Bit	Description	Bit	Description	Bit	Description	Bit	Description
1	PG(4)-4	37	PD(2)-0	73	PD(1)-4	109	CI(1)-2
2	PD(3)-4	38	CI(4)-1	74	CG(3)-2	110	PG(2)-1
3	LSP 1-1	39	LSP 9-0	75	LSP 7-1	111	CI(3)-7
4	CG(2)-4	40	CI(3)-8	76	CI(2)-7	112	LSP 4-0
5	CI(3)-3	41	PG(1)-4	77	CI(3)-0	113	CI(2)-5
6	CI(1)-8	42	CG(2)-2	78	PD(2)-5	114	PD(1)-7
7	PD(4)-0	43	PD(1)-3	79	LSP 4-1	115	PG(1)-0
8	LSP 8-0	44	LSP 6-1	80	CG(1)-0	116	CG(4)-4
9	PG(2)-3	45	CI(3)-4	81	PG(4)-3	117	LSP 5-0
10	CG(3)-0	46	CI(2)-2	82	LSP 9-1	118	PD(4)-2
11	PD(1)-5	47	CG(1)-4	83	PD(3)-6	119	CI(1)-3
12	LSP 3-3	48	PD(2)-3	84	CI(1)-4	120	CI(3)-1
13	CI(2)-3	49	LSP 1-2	85	CG(2)-1	121	LSP 7-2
14	CI(4)-4	50	PG(3)-2	86	LSP 6-2	122	CI(4)-2
15	PD(2)-1	51	HP-1	87	CI(4)-3	123	PD(1)-1
16	LSP 10-0	52	PD(3)-1	88	PG(2)-2	124	PG(2)-4
17	PG(1)-3	53	CG(4)-3	89	PD(4)-3	125	CG(3)-3
18	CG(4)-0	54	LSP 8-1	90	LSP 1-0	126	LSP 3-1
19	LSP 5-2	55	PG(3)-0	91	CG(4)-2	127	CI(1)-7
20	PD(3)-0	56	CI(2)-8	92	LSP 8-2	128	PD(3)-2
21	HP-0	57	PD(4)-1	93	CI(2)-4	129	CI(2)-6
22	CI(1)-1	58	CI(4)-0	94	HP-2	130	LSP 9-2
23	CI(4)-8	59	LSP 3-2	95	PD(2)-2	131	PG(4)-1
24	LSP 2-2	60	PG(2)-0	96	LSP 3-0	132	CG(1)-1
25	PG(3)-1	61	PD(1)-6	97	PG(1)-2	133	PD(2)-4
26	PD(4)-5	62	CG(2)-0	98	CG(3)-4	134	HP-3
27	CG(1)-3	63	CI(3)-6	99	LSP 10-2	135	LSP 6-0
28	CI(3)-5	64	LSP 10-1	100	CI(4)-5	136	PG(3)-3
29	LSP 7-0	65	PG(1)-1	101	CI(2)-0	137	CI(4)-6
30	CI(2)-1	66	CI(4)-7	102	PD(1)-2	138	PD(1)-0
31	PD(3)-7	67	PD(3)-3	103	LSP 5-1	139	LSP 2-3
32	CI(1)-0	68	CG(1)-2	104	SP-0	140	CG(4)-1
33	PG(4)-0	69	LSP 5-3	105	PG(4)-2	141	CI(3)-2
34	LSP 4-3	70	CI(1)-6	106	CG(2)-3	142	LSP 4-2
35	CG(3)-1	71	LSP 2-0	107	LSP 2-1	143	PD(3)-5
36	CI(1)-5	72	PG(3)-4	108	PD(4)-4	144	SY

3.8.3 Synchronization Bit. **The single** synchronization **bit per frame** shall alternate between binary 0 and 1 from frame to frame. It should be coded as binary-0 for the first frame transmitted.

3.8.4 Expansion Bit. When following this standard, the expansion bit shall be set to binary 0 in each frame. The purpose of this bit is to allow for future transition to as yet undefined coding techniques.

3.9 Error Correction

3.9.1 Encoding. Forward error correction of 11 bits with a (15,11) Hamming parity code is provided. The 11 bits protected from a single bit error are: **PD(1)-5, PD(1)-6, PD(1)-7, PG(1)-4, PG(2)-4, PD(3)-5, PD(3)-6, PD(3)-7, PG(3)-4, PG(4)-4, and SP**. (See section 3.8.2 for an explanation of these abbreviations). The 4 parity bits are encoded as follows. **Parity bits HP-0** through **HP-3** each represent an even parity computation on 7 of the 11 protected bits as shown below (i.e., a parity bit is 0 if the sum of the 7 protected bits is even).

HP-0 --PD(1)-5, PD(1)-6, PG(1)-4, PG(2)-4, PD(3)-6, PG(3)-4, and SP
 HP-1 --PD(1)-5, PD(1)-7, PG(1)-4, PD(3)-5, PD(3)-6, PG(4)-4, and Sp
 HP-2 --PD(1)-6, PD(1)-7, PG(1)-4, PD(3)-7, PG(3)-4, PG(4)-4, and SP
 HP-3 --PG(2)-4, PD(3)-5, PD(3)-6, PD(3)-7, PG(3)-4, PG(4)-4, and SP

3.9.2 Decoding. The 4 parity bits (HP-0,1 ,2, and 3) shall be used to correct single errors in the 11 protected bits as follows. Parity bits are calculated on the received 11 protected bits as described in section 3.9.1. Then, these calculated parity bits are Exclusive ORed with the received parity bits. The following table shows how to correct for a single bit error by reversing the binary value of one of the protected bits based upon the resulting binary value of the Exclusive ORed parity bits.

Parity	Result	Invert	Parity	Result	Invert
3	PD(1)-5		11	PD(3)-6	
5	PD(1)-6		12	PD(3)-7	
6	PD(1)-7		13	PG(3)-4	
7	PG(1)-4		14	PG(4)-4	
9	PG(2)-4		15	SP	
10	PD(3)-5				

3.10 General Application Notes

3.10.1 Amplitude Scaling. "Stochastic" elements and gain values contained in this standard are based upon use of input digitized voice that can assume any integer value in the range -32,768 to +32,767 and an impulse of 1.0 to calculate the impulse response of the perceptual weighting filter.

3.10.2 Filter Structure. Various filter structures may be used in the implementation of this standard. All implementations must be interoperable with implementations based upon Direct Form (i.e., block-wise) filtering.

3.10.3 Smoothing. It is desirable to employ smoothing to prevent loud clipped voice (i.e., blasts) and squeaks. Smoothing based upon estimates of the channel error rate (provided by the error correction mechanism described in section 3.9) is recommended.

4. EFFECTIVE DATE. The use of this standard by U.S. Government departments and agencies is mandatory effective 180 days after the date of this standard.

5. CHANGES. When a Government department or agency considers that this standard does not provide for its essential needs, a statement citing specific requirements shall be sent in duplicate to the General Services Administration (K), Washington, DC 20405, in accordance with the provisions of Federal Information Resources Management Regulation 41 CFR 201-20. The General Services Administration will determine the appropriate action to be taken and will notify the agency.

FED-STD 1016

TECHNICAL DEVELOPMENT ACTIVITY:

National Security Agency
Information Security Organization (R2)
Fort George G. Meade, MD 20755

PREPARING ACTIVITY:

National Communications System
Office of Technology and Standards
Washington, DC 20305-2010

MILITARY INTERESTS

Military Coordinating Activity

DCA -- DC

Custodians

Army -- SC

Navy -- EC

Air Force --90

Review Activities

Army -- CR

Navy -- AS,OM

Air Force -- 17

USMC -- MC

NSA -- NS

FEDERAL STANDARD

TELECOMMUNICATIONS: ANALOG TO DIGITAL CONVERSION OF RADIO
VOICE BY 4,800 BIT/SECOND CODE EXCITED LINEAR PREDICTION (CELP)

This standard is issued by the General Services Administration pursuant to the Federal Property and Administrative Services Act of 1949, as amended.

1. SCOPE

1.1 Description. This standard specifies **interoperability** -related requirements for the conversion of analog voice to a 4,800 bit/s digitized form for digital radio transmission by a method known as Code Excited Linear Prediction (CELP) and the reverse process, the synthesis of analog voice from 4,800 bit/s **CELP-digitized** voice. In addition to digital radio applications, CELP is also very suitable for encrypted telephone use and other applications wherein voice must be digitized prior to encryption.

1.2 Purpose. This standard is to facilitate interoperability between radio telecommunication facilities and systems of the Federal Government.

1.3 Application. This standard shall be used by all Federal departments and agencies in the design and procurement of all radio equipment employing 4,800 bit/s digitized voice.

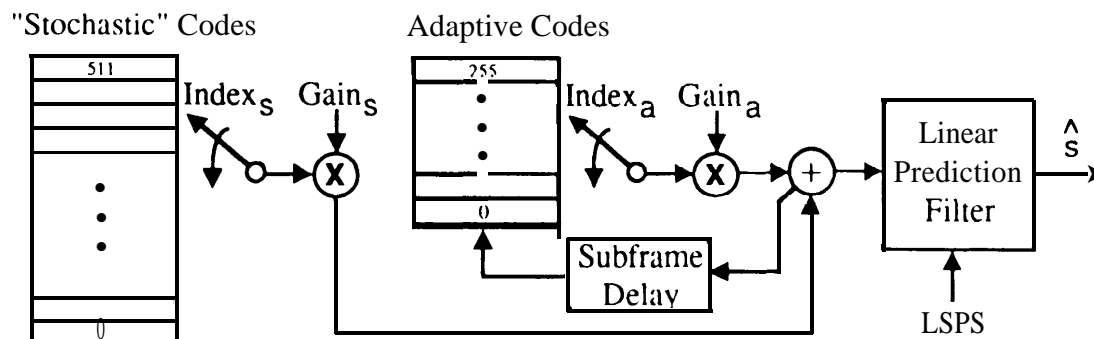
2. CONVENTIONS AND DEFINITIONS

a. **Frame**. A CELP frame is 144 bits in length. A frame interval is 30 ms ± 0.1 percent in duration and contains 240 voice samples (8,000 samples/s).

b. **Subframe**. A CELP subframe is 1/4 the length of a frame. Thus, a subframe is 7.5 ms ± 0.1 percent in duration and contains 60 voice samples.

3. REQUIREMENTS3.1 Voice Synthesis

3.1.1 General Description. Since Code Excited Linear Prediction (CELP) is an analysis-by-synthesis type of technique, voice synthesis is described first. As shown in the diagram below, CELP synthesis involves the excitation of a parameter-adjusted Linear Prediction Filter by the sum of gain-scaled codes selected from a fixed, **stochastically**- derived "stochastic code book" and an adaptive "code book," utilizing parameters transmitted in a 144-bit frame structure.



The fixed, stochastically-derived "stochastic" codes are described in section 3.3. "Stochastic" code gain is described in section 3.4. Adaptive codes are described in section 3.5 and adaptive code gain is described in section 3.6. Section 3.7 describes the Linear Prediction Filter and the Line Spectral Parameters (LSPS) that adjust it. Section 3.8 shows transmission format, including bit assignments for the transmitted information. Single bit error correction on some of the most sensitive bits is described in section 3.9.

3.1.2 Postfiltering. **Postfiltering** may be used to enhance the synthesized voice coming out of the CELP Linear Prediction Filter.

3.1.3 Lowpass Filtering. Lowpass (i.e., reconstruction) filtering shall be employed. A typical lowpass filter has a 3 dB attenuation point at 3,800 Hz, less than 1 dB of passband ripple, and minimum attenuations of 18 dB at 4,000 Hz and 46 dB above 4,400 Hz. In certain applications, mild highpass filtering (e.g., 175 Hz second-order Butterworth) may also be of benefit.

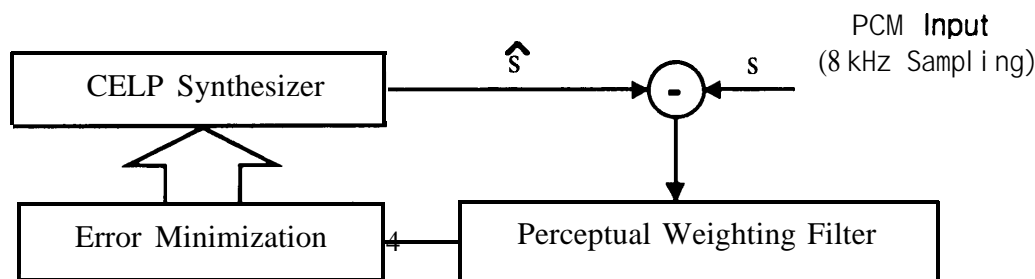
3.2 Voice Analysis

3.2.1 Voice Input Filtering. The analog voice input bandpass should be essentially flat from 200 to 3,400 Hz. A typical input filter has 3 dB attenuation points at 100 and 3,800 Hz; less than 1 dB of inband ripple; and minimum attenuations of 18 dB at 50 Hz, 18 dB at 4,000 Hz, and 46 dB above 4,400 Hz.

3.2.2 Analog-to-Digital Conversion. Analog-to-digital conversion shall use an 8 kHz \pm 0.1 percent sampling frequency and have a dynamic range of at least 12 bits.

3.2.3 Amplitude Scaling. To maintain proper receiver voice levels, analysis shall be based upon use of input digitized voice whose peak values are -32,768 and +32,767.

3.2.4 Analysis. As reflected in the following diagram, CELP is an analysis-by-synthesis type of



technique. The objective of **CELP analysis** is to minimize the perceptual difference (i.e., find the best match) between the actual digitized voice and the synthesized voice resulting from use of the parameters to be transmitted. As shown in the diagram, linear Pulse Code Modulated (PCM) voice sampled at 8 kHz is subtracted from the CELP synthesizer approximation and passed through a perceptual weighting filter (see section 3.2.5). The synthesizer parameters to be transmitted are adjusted for minimum perceptual error with respect to the actual input voice signal.

3.2.5 Perceptual Weighting Filter. It is recommended that a perceptual weighting filter be the cascade of a linear predictive whitening filter and a bandwidth expanded linear predictive synthesis filter. The bandwidth expanded linear predictive synthesis filter's poles are moved radially toward the origin by a weighting factor, typically 0.8.

3.3 "Stochastic" Codes. There are 512 fixed, **stochastically-derived** codes (i.e., vectors). During voice analysis, a code may be selected from a set smaller than 512 in order to reduce computational complexity (at the expense of voice reproduction quality). All 512 fixed, **stochastically-derived** codes

shall be available for voice synthesis. Each of the fixed, **stochastically-derived** codes contains 60 ternary (i.e., either -1, 0, or 1) elements, representing information used to form the excitation for the Linear Prediction Filter over a **subframe** period (i.e., 8,000 elements/s for 7.5 ins). The fixed, **stochastically-derived** codes, 512 overlapped codes of 60 elements each, are created from a 1,082 element "stochastic code book" as described below.

3.3.1 "Stochastic Code Book". The 1,082 element "stochastic code book" is defined by the following FORTRAN computer program.

```

program codebook
implicit none
integer M, L, MAXCODE, WIDTH, j, k
parameter (M=512, L=60)
parameter (MAXCODE=2*(M-1)+L)
parameter (WIDTH= 20)
real x( 0: MAXCODE+1), THRESH
parameter (THRESH=1 .2)
open (unit=10, file= 'codebook.h', status=' new' )
do 50 k=0, MAXCODE, 2
  call noise(x(k),x(k+1))
  do 20 j=0, 1
    if(abs(x(k+j)).lt. THRESH) then
      x(k+j) = 0.0
    else
      x(k+j) = sign(1.0, x(k+j))
    end if
  20 continue
50 continue
  do 80 k=1, MAXCODE-WIDTH, WIDTH
    write (10, 90) (int(x(j)), j=k, k+WIDTH-1)
  80 continue
  write (10, 91) (int(x(j)), j=MAXCODE/WIDTH*WIDTH+1, MAXCODE)
  stop 'codebook.h generated'
90 format(1x, 20(i3, ' '))
91 format(1x, 20(i3:, ' '))
end
subroutine noise(x1, x2)
implicit none
real x1, x2
integer random, i, j
real f(2), f1, f2, s
10 do 30 i=1, 2
  do 20 j=1, 4
    f(i) = (float (random() + 32768 ))/65535.
  20 continue
30 continue
  f1=2.*f(1)-1.
  f2=2.*f(2)-1.
  s=f1*f1+f2*f2
  if(s .ge. 1.) goto 10
  s=sqrt(-2.*slog(s)/s)
  x1=f1*s
  x2=f2*s
  return
end

```


FED-STD 1016

```

function random ( )
implicit none
integer random
integer MIDTAP, MAXTAP
parameter (MIDTAP=2, MAXTAP=5)
integer y(MAXTAP), j, k, temp
save y, j, k
data y/-21161, -8478, 30892, -10216, 16950/
data j/MIDTAP/, k/MAXTAP/
temp=iand(y(k)+y(j), 65535)
if (temp .gt. 32767) temp=temp-65536
y(k)=temp
random=temp
k=k-1
if (k l e. 0) k=MAXTAP
j=j-1
if (j l e. 0) j=MAXTAP
return
end

```

The first and last 200 elements generated by the above FORTRAN program are shown below. The left-most and highest elements are lowest numbered (e.g., the element in the first row and third column is numbered 2).

```

o, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, -1,
0, 0, -1, 0, -1, 0, 1, 0, 0, 0, 0, 0, 0, 0, -1, 0, -1, -1, 0, 0,
-1, 0, -1, 0, -1, 0, -1, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 1, 0,
0, 0, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, -1, 0,
1, 0, 1, 0, -1, 1, 0, 0, 0, 0, 0, -1, 0, 0, 1, 0, 0, 0, -1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, -1,
0, 1, -1, 0, 0, -1, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
o, 0, 0, -1, 0, -1, -1, 0, 0, -1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
0, -1, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, -1, 0, 1, 0, 0, 0, 0, 0,

```

.....

```

0, 1, 0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 1, 0, 0, 0, 0, -1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 1, 0, 0, 1, 0, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, -1, 0, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, -1, 1, -1, 0, 1, 0, 0, 0,
o, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 0, 0, 0, -1, 0,
0, 0, -1, 0, -1, 0, 0, 0, 0, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0,
0, 0, 0, 1, -1, 0, 0, 0, 0, -1, 0, 1, 0, 0, 1, 0, 0, 0, 0, -1,
0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0, 1,
0, 0,

```

3.3.2 Creation of **Stochastically-derived** Codes. The 512 fixed codes of 60 elements each are assembled from the 1,082 stochastically-derived "stochastic code book" elements as shown below. The left-most code elements are first-most in time.

Index	Elements In Codes	
511	0, 1, 2,... 58, 59
510	2, 3, 4,... 60, 61
.		
n	2(511-n), 2(511-n)+1,.. 2(511-n)+59
.		
.		
1	1,020, 1,021, 1,022,... 1,078, 1,079
0	1,022, 1,023, 1,024,... 1,080, 1,081

3.4 “Stochastic” Code Gain. The relative amplitude (to the nearest table value) to be applied to the **stochastically-derived** code elements of each **subframe** is determined during analysis and coded into 5 bits according to the following table. The decimal index number is then transmitted in binary form. At the receiver, in voice synthesis, the gain index number is used to decode the relative amplitude of the **stochastically** - derived code elements during the **subframe**.

Index	Gain	Index	Gain	Index	Gain	Index	Gain
0	-1,330	8	-178	16	1	24	224
1	-870	9	-136	17	3	25	278
2	-660	10	98	18	13	26	340
3	-520	11	-64	19	35	27	418
4	-418	12	-35	20	64	28	520
5	-340	13	-13	21	98	29	660
6	-278	14	-3	22	136	30	870
7	-224	15	-1	23	178	31	1,330

3.5 Adaptive Codes

3.5.1 Adaptive “Code Book” and Integer-delay Codes. The adaptive “code book” is a 147-element, shifting storage register that is updated at the start of every **subframe** with the previous 60 elements (**subframe** interval) of Linear Prediction Filter excitation. Element ordering is such that the first excitation elements into the Linear Prediction Filter are the first into the adaptive “code book”. Elements already in the storage register are shifted up during the update process and the oldest elements are discarded. The 128 integer-delay overlapped adaptive codes, of 60 elements each, are generated from the information in the adaptive “code book” as follows. Adaptive codes 60 through 147 are composed of elements -60 to -1, -61 to -2, . . . through -147 to -88, respectively (where element -1 was the last element into the adaptive “code book”). Adaptive code “n”, where n ranges from 20 to 59, repeats the adaptive “code book” elements sequentially from -n to -1 to form a 60-element code. As examples, adaptive code 20 repeats adaptive “code book” elements -20 ...-1 three times and adaptive code 59’s elements run -59 . . . -1, -59. Regarding order of utilization, the most negatively numbered (i.e., most delayed) element of a particular adaptive code is used to compute the first element in time of Linear Prediction Filter excitation for a particular subframe period.

3.5.2 Noninteger-delay Codes. An additional 128 implicit, noninteger-delay codes, defined in section 3.5.4, can optionally be made available through interpolation during voice analysis. All 256 adaptive codes shall be available for voice synthesis. Forty point interpolation, or the equivalent, shall be used in both voice analysis and voice synthesis. (However, this does not preclude interpolating with fewer points, perhaps as few as 8 points, during preliminary code search computations). The process of interpolation is described in section 3.5.3.

3.5.3 Interpolation. Anoninteger-delay adaptive code is interpolated from the nearest lower-valued integer-delay code (e.g., noninteger-delay code 49.67 is determined by interpolating from integer-delay code 49). The 60 elements of the integer-delay code are first renumbered as **R0 . . . R59** for purposes of description, where **R0** is normally the most negatively numbered (i.e., most delayed) element under the previous numbering system (i.e., -1 to -147), **R1** is the next most negatively numbered element, etc. For example, for integer-delay code 66, **R0=element -66**, **R1=element -65**, and **R59=element -7**. For integer-delay code 21, because of the repetition within lower numbered integer-delay codes, **R0=element -21**, **R1=element -20**, **R58=element -4**, and **R59=element -3**. Likewise, for integer-delay code 58, **R0=element -58**, **R1=element -57**, **R57=element -1**, **R58=element -58**, and **R59=element -57**.

The next step in explaining the process of interpolation is to show the computation of the weighting factors. The following FORTRAN program gives interpolation weighting factors for the interpolation of even numbers of points (N) equal to and less than 40. Note: interpolation with less than 8 points is not recommended and interpolation with other than 40 points is only for preliminary computations during voice analysis.

```

program weights
implicit none
integer N, nf, i, j, k
parameter (N=40, nf=5)
real h(-6*N:6*N), w(-N/2:N/2-1, nf), f(nf), pi
data f/0.25, 0.33333333, 0.5, 0.66666667, 0.75/
pi = 4.0*atan(1.0)
open(unit=10, file='weights', status='new')
do 10 k = -6*N, 6*N
    h(k) = 0.54 + 0.46* cos(pi*k/(6*N))
10 continue
do 30 i = 1, nf
    do 20 j = -N/2, N/2-1
        w(j, i) = h(nint(12*(j+f(i)))) * sin(pi*(j+f(i))) / (pi*(j+f(i)))
20 continue
30 continue
write (10,69) f
do 40 j = -N/2, N/2-1
    write (10,70) j, (w(j, k), k = 1, nf)
40 continue
69 format (7x, 5f14.5)
70 format(3x, i4, 5e14.5)
end

```

For N-point interpolation, the above program yields N weighting factors, numbered from -N/2 to N/2-1. As the final step showing the process of interpolation, these weighting factors are multiplied by the values of the adaptive code elements within a "window" around the element being interpolated (**R**). The sum of these multiplications is the interpolated value of that element (**R'**). For example, for 40 point interpolation, to find noninteger-delay code 66.67 (i.e., integer= 66, **fraction**= .67), element -66 becomes **R0**. Weighting factor -20 (i.e., -.001 1766) is multiplied by the value of element -86, weighting factor -19 (i.e., .0014386) is multiplied by the value of element -85, weighting factor 0 (i.e., .41245) is multiplied by the value of element -66, . . . , up to weighting factor 19 (i.e., -.001 1302) being multiplied by the value of element -47. The sum of these becomes the interpolated value of element -66 (i.e., **R'O**). The same operation takes place in a "window" around the remaining elements (**R1 - R59**). However, in the example given, the samples run out beyond **R46** (i.e., for **R46** weighting factor 19 is multiplied by the value of element -1). In such cases, the previously interpolated values for elements **R0, R1, R2, . . .** (i.e., **R'O, R*1, R'2, . . .**) are used to complete interpolation of the remaining elements. Thus, in the example given, when interpolating **R58** (i.e., element -8), weighting factor -20 is multiplied by the value of element -28 and weighting factor 19

is multiplied by the value of already interpolated element R'11 (i.e., interpolated element -56). The sum of the 40 values found by multiplying the 40 weighting factors by the 40 element values (i.e., element -28, element -27, element -26, . . . element -1, R'0, R*1, . . . R'10, R'11) becomes the value of R'58 (i.e., interpolated element -8).

3.5.4 Coding Within 1st and 3rd Subframes for Transmission. The following table identifies the 8-bit values transmitted to represent each of the 256 (128 integer-delay and 128 noninteger-delay (implicit)) adaptive codes. The notation is such that, for example, hexadecimal (\$) AF is expressed as 10101111 in binary form, where bit 7 is on the left and bit 0 is on the right.

Adaptive Hex Code	Adaptive Hex Code	Adaptive Hex Code	Adaptive Hex Code	Adaptive Hex Code
20.00 \$42	34.67 \$C0	51.67 \$98	68.67 \$C5	97.00 \$A1
20.33 \$46	35.00 \$C3	52.00 \$90	69.00 \$C9	98.00 \$97
20.67 \$47	35.33 \$C2	52.33 \$80	69.33 \$C8	99.00 \$87
21.00 \$57	35.67 \$D2	52.67 \$9A	69.67 \$C7	100.0 \$9F
21.33 \$56	36.00 \$D3	53.00 \$8A	70.00 \$CB	101.0 \$8F
21.67 \$59	36.33 \$D1	53.33 \$82	70.33 \$C6	102.0 \$81
22.00 \$58	36.67 \$D0	53.67 \$92	70.67 \$CA	103.0 \$91
22.33 \$AE	37.00 \$30	54.00 \$1A	71.00 \$D6	104.0 \$9B
22.67 \$BE	37.33 \$32	54.33 \$12	71.33 \$DA	105.0 \$8B
23.00 \$BA	37.67 \$3A	54.67 \$00	71.67 \$DB	106.0 \$83
23.33 \$B8	38.00 \$31	55.00 \$08	72.00 \$D7	107.0 \$93
23.67 \$BC	38.33 \$33	55.33 \$06	72.33 \$D9	108.0 \$18
24.00 \$AC	38.67 \$3B	55.67 \$OE	72.67 \$D5	109.0 \$10
24.33 \$A8	39.00 \$3F	56.00 \$OF	73.00 \$D8	110.0 \$04
24.67 \$94	39.33 \$37	56.33 \$07	73.33 \$D4	111.0 \$Oc
25.00 \$84	39.67 \$3E	56.67 \$17	73.67 \$20	112.0 \$16
25.33 \$8C	40.00 \$36	57.00 \$1F	74.00 \$28	113.0 \$1E
25.67 \$9C	40.33 \$34	57.33 \$OD	74.33 \$38	114.0 \$14
26.00 \$9E	40.67 \$4A	57.67 \$05	74.67 \$22	115.0 \$1C
26.25 \$8E	41.00 \$4B	58.00 \$1D	75.00 \$2A	116.0 \$F9
26.50 \$86	41.33 \$4E	58.33 \$15	75.33 \$39	117.0 \$FA
26.75 \$96	41.67 \$4F	58.67 \$FB	75.67 \$29	118.0 \$FD
27.00 \$0A	42.00 \$5F	59.00 \$FF	76.00 \$21	119.0 \$E9
27.25 \$02	42.33 \$5E	59.33 \$EB	76.33 \$23	120.0 \$FE
27.50 \$0B	42.67 \$5C	59.67 \$EF	76.67 \$2B	121.0 \$E8
27.75 \$03	43.00 \$5D	60.00 \$ED	77.00 \$27	122.0 \$FC
28.00 \$1B	43.33 \$54	60.33 \$EA	77.33 \$2F	123.0 \$43
28.25 \$13	43.67 \$55	60.67 \$EE	77.67 \$25	124.0 \$F2
28.50 \$09	44.00 \$50	61.00 \$EC	78.00 \$2D	125.0 \$F6
28.75 \$01	44.33 \$51	61.33 \$E6	78.33 \$3D	126.0 \$F8
29.00 \$19	44.67 \$AA	61.67 \$E2	78.67 \$35	127.0 \$5B
29.25 \$11	45.00 \$A6	62.00 \$E4	79.00 \$3C	128.0 \$5A
29.50 \$F3	45.33 \$A2	62.33 \$E0	79.33 \$2E	129.0 \$63
29.75 \$F7	45.67 \$B6	62.67 \$F4	79.67 \$2C	130.0 \$62
30.00 \$E7	46.00 \$B2	63.00 \$F0	80.00 \$26	131.0 \$77
30.25 \$E3	46.33 \$BB	63.33 \$60	81.00 \$24	132.0 \$76
30.50 \$E5	46.67 \$B0	63.67 \$64	82.00 \$49	133.0 \$52
30.75 \$E1	47.00 \$B9	64.00 \$74	83.00 \$48	134.0 \$53
31.00 \$F1	47.33 \$B4	64.33 \$70	84.00 \$4C	135.0 \$66
31.25 \$F5	47.67 \$BD	64.67 \$73	85.00 \$4D	136.0 \$67
31.50 \$61	48.00 \$A4	65.00 \$72	86.00 \$44	137.0 \$CC
31.75 \$65	48.33 \$A0	65.33 \$6C	87.00 \$45	138.0 \$CD
32.00 \$75	48.67 \$A9	65.67 \$7C	88.00 \$40	139.0 \$AB

(continued)

32.25 \$71	49.00 \$AD	66.00 \$68	89.00 \$41	140.0 \$CF
32.50 \$6D	49.33 \$95	66.33 \$78	90.00 \$A7	141.0 \$CE
32.75 \$7D	49.67 \$85	66.67 \$7A	91.00 \$A3	142.0 \$DE
33.00 \$69	50.00 \$9D	67.00 \$7E	92.00 \$B7	143.0 \$BF
33.25 \$79	50.33 \$8D	67.33 \$6A	93.00 \$B3	144.0 \$DF
33.50 \$7B	50.67 \$89	67.67 \$6E	94.00 \$B1	145.0 \$DD
33.75 \$7F	51.00 \$99	68.00 \$6F	95.00 \$B5	146.0 \$DC
34.00 \$6B	51.33 \$88	68.33 \$C4	96.00 \$A5	147.0 \$AF
34.33 \$C1				

Note: Before an adaptive code is selected for transmission, consideration should be given to the possibility that a multiple of actual "pitch" is being selected. Submultiple of a selected code's pitch equivalent should be reexamined and maybe utilized instead if results are found to be within approximately 0.5 dB of a selected code's squared prediction error.

3.5.5 Coding Within 2nd and 4th Subframes for Transmission. The adaptive code selected for transmissions represented in 6 bits, based upon the adaptive code selected in the previous subframe. Utilizing the table of adaptive codes in section 3.5.4, if a previous subframe's adaptive code was in the range of 20.00 - 29.25, the adaptive code to be transmitted could run from 20.00 to 38.33. If the previous subframe's adaptive code was in the range of 115.0 - 147.0, the adaptive code could run from 84.00 to 147.0. otherwise, the 6 bits will code the range from 31 codes lower to 32 codes higher than the previous subframe's adaptive code (considering both integer-delay and noninteger-delay adaptive codes). This is true even if an implementation uses only integer adaptive codes. Binary coding is such that the lowest numbered adaptive code is coded as 000000 and the highest numbered adaptive code is coded as 111111. For example, for previous subframe adaptive codes between 29.50 and 114.0, binary 011111 indicates no difference from the adaptive code of the previous subframe.

3.6 Adaptive Code Gain. The relative amplitude (to the nearest table value) to be applied to the adaptive code elements of each subframe is determined during analysis and coded into 5 bits according to the following table. The decimal index number is then transmitted in binary form. At the receiver, in voice synthesis, the adaptive code gain index number is used to decode the relative amplitude of the adaptive code elements during the subframe.

Index	Gain	Index	Gain	Index	Gain	Index	Gain
0	-.993	8	0.255	16	0.780	24	1.062
1	-.831	9	0.368	17	0.816	25	1.117
2	-.693	10	0.457	18	0.850	26	1.193
3	-.555	11	0.531	19	0.881	27	1.289
4	-.414	12	0.601	20	0.915	28	1.394
5	-.229	13	0.653	21	0.948	29	1.540
6	0.000	14	0.702	22	0.983	30	1.765
7	0.139	15	0.745	23	1.020	31	1.991

3.7 Linear Prediction

3.7.1 Linear Prediction Filter. A 10th order Linear Prediction Filter is excited by the sum of the gain-adjusted "stochastic" and adaptive codes to synthesize voice.

3.7.2 Line Spectral Parameters. During voice analysis, the parameters of the Linear Prediction Filter are determined and transmitted on a 30 ms frame basis as 10 Line Spectral Parameters (LSPs), on the basis of 10th order linear prediction. It is recommended that LSPs be determined with no preemphasis, 15 Hz bandwidth expansion (i.e., 0.994 weighting factor), and a 30 ms nonoverlapped Hamming window spanning the last two subframes of the present frame and first two subframes of the next frame. Shown below is the coding to be employed for the 10 LSPs. Note: LSPs are

expressed in terms of frequency (Hz) and must be monotonic (i.e., higher numbered LSPs must not contain lower frequencies than lower numbered **LSPs**).

Index	LSP1 (Hz)	LSP2 (Hz)	LSP3 (Hz)	LSP4 (Hz)	LSP5 (Hz)	LSP6 (Hz)	LSP7 (Hz)	LSP8 (Hz)	LSP9 (Hz)	LSP10 (Hz)
0	100	210	420	620	1,000	1,470	1,800	2,225	2,760	3,190
1	170	235	460	660	1,050	1,570	1,880	2,400	2,880	3,270
2	225	265	500	720	1,130	1,690	1,960	2,525	3,000	3,350
3	250	295	540	795	1,210	1,830	2,100	2,650	3,100	3,420
4	280	325	585	880	1,285	2,000	2,300	2,800	3,200	3,490
5	340	360	640	970	1,350	2,200	2,480	2,950	3,310	3,590
6	420	400	705	1,080	1,430	2,400	2,700	3,150	3,430	3,710
7	500	440	775	1,170	1,510	2,600	2,900	3,350	3,550	3,830
8		480	850	1,270	1,590					
9		520	950	1,370	1,670					
10		560	1,050	1,470	1,750					
11		610	1,150	1,570	1,850					
12		670	1,250	1,670	1,950					
13		740	1,350	1,770	2,050					
14		810	1,450	1,870	2,150					
15		880	1,550	1,970	2,250					

3.7.3 LSP Weighted Averaging. To determine the LSP values to be used for each subframe, weighted **averaging** is used between the LSPs transmitted with the previous frame and the LSPs transmitted with the present frame. For each of the **subframes** of a given frame, the following weighting is given to the LSPs transmitted with the previous and present frames before they are utilized.

Subframe	Previous LSPs	Present LSPs
1	7/8	1/8
2	5/8	3/8
3	3/8	5/8
4	1/8	7/8

Thus, for example, if LSP 1 of a frame is 250 Hz and LSP 1 of the previous frame was 340 Hz, the value used for **subframe 1** of the present frame is: $7/8(340 \text{ Hz}) + 1/8(250 \text{ Hz}) = 328.75 \text{ Hz}$. For each subframe, this method of weighted averaging is employed for all 10 LSPs.

3.8 Transmission Format

3.8.1 Transmission Rate. The transmission rate for CELP shall be 4,800 bits/s ± 0.1 percent.

3.8.2 Bit Assignment. The table below gives the assignment of the 144 bits in each CELP frame (30 ms at 4,800 bits/s). Order of transmission is from bit 1 to bit 144. Abbreviations used in the following table are:

i = bit number, with 0 being the least significant bit

n = **subframe** number

CG(n)-i = Fixed, **Stochastically-derived** Code Gain

CI(n)-i = Fixed, **Stochastically-derived** Code Index

HP-i = Parity

LSP j-i = Line Spectral Parameter (**LSP**), where j = LSP number

PD(n)-i = Adaptive Code Index

PG(n)-i = Adaptive Code Gain

SP = Expansion Bit

SY = Synchronization Bit

Bit	Description	Bit	Description	Bit	Description	Bit	Description
1	PG(4)-4	37	PD(2)-0	73	PD(1)-4	109	CI(1)-2
2	PD(3)-4	38	CI(4)-1	74	CG(3)-2	110	PG(2)-1
3	LSP 1-1	39	LSP 9-0	75	LSP 7-1	111	CI(3)-7
4	CG(2)-4	40	CI(3)-8	76	CI(2)-7	112	LSP 4-0
5	CI(3)-3	41	PG(1)-4	77	CI(3)-0	113	CI(2)-5
6	CI(1)-8	42	CG(2)-2	78	PD(2)-5	114	PD(1)-7
7	PD(4)-0	43	PD(1)-3	79	LSP 4-1	115	PG(1)-0
8	LSP 8-0	44	LSP 6-1	80	CG(1)-0	116	CG(4)-4
9	PG(2)-3	45	CI(3)-4	81	PG(4)-3	117	LSP 5-0
10	CG(3)-0	46	CI(2)-2	82	LSP 9-1	118	PD(4)-2
11	PD(1)-5	47	CG(1)-4	83	PD(3)-6	119	CI(1)-3
12	LSP 3-3	48	PD(2)-3	84	CI(1)-4	120	CI(3)-1
13	CI(2)-3	49	LSP 1-2	85	CG(2)-1	121	LSP 7-2
14	CI(4)-4	50	PG(3)-2	86	LSP 6-2	122	CI(4)-2
15	PD(2)-1	51	HP-1	87	CI(4)-3	123	PD(1)-1
16	LSP 10-0	52	PD(3)-1	88	PG(2)-2	124	PG(2)-4
17	PG(1)-3	53	CG(4)-3	89	PD(4)-3	125	CG(3)-3
18	CG(4)-0	54	LSP 8-1	90	LSP 1-0	126	LSP 3-1
19	LSP 5-2	55	PG(3)-0	91	CG(4)-2	127	CI(1)-7
20	PD(3)-0	56	CI(2)-8	92	LSP 8-2	128	PD(3)-2
21	HP-0	57	PD(4)-1	93	CI(2)-4	129	CI(2)-6
22	CI(1)-1	58	CI(4)-0	94	HP-2	130	LSP 9-2
23	CI(4)-8	59	LSP 3-2	95	PD(2)-2	131	PG(4)-1
24	LSP 2-2	60	PG(2)-0	96	LSP 3-0	132	CG(1)-1
25	PG(3)-1	61	PD(1)-6	97	PG(1)-2	133	PD(2)-4
26	PD(4)-5	62	CG(2)-0	98	CG(3)-4	134	HP-3
27	CG(1)-3	63	CI(3)-6	99	LSP 10-2	135	LSP 6-0
28	CI(3)-5	64	LSP 10-1	100	CI(4)-5	136	PG(3)-3
29	LSP 7-0	65	PG(1)-1	101	CI(2)-0	137	CI(4)-6
30	CI(2)-1	66	CI(4)-7	102	PD(1)-2	138	PD(1)-0
31	PD(3)-7	67	PD(3)-3	103	LSP 5-1	139	LSP 2-3
32	CI(1)-0	68	CG(1)-2	104	SP-0	140	CG(4)-1
33	PG(4)-0	69	LSP 5-3	105	PG(4)-2	141	CI(3)-2
34	LSP 4-3	70	CI(1)-6	106	CG(2)-3	142	LSP 4-2
35	CG(3)-1	71	LSP 2-0	107	LSP 2-1	143	PD(3)-5
36	CI(1)-5	72	PG(3)-4	108	PD(4)-4	144	SY

3.8.3 Synchronization Bit. **The single** synchronization **bit per frame** shall alternate between binary 0 and 1 from frame to frame. It should be coded as binary-0 for the first frame transmitted.

3.8.4 Expansion Bit. When following this standard, the expansion bit shall be set to binary 0 in each frame. The purpose of this bit is to allow for future transition to as yet undefined coding techniques.

3.9 Error Correction

3.9.1 Encoding. Forward error correction of 11 bits with a (15,11) Hamming parity code is provided. The 11 bits protected from a single bit error are: **PD(1)-5, PD(1)-6, PD(1)-7, PG(1)-4, PG(2)-4, PD(3)-5, PD(3)-6, PD(3)-7, PG(3)-4, PG(4)-4, and SP**. (See section 3.8.2 for an explanation of these abbreviations). The 4 parity bits are encoded as follows. **Parity bits HP-0** through **HP-3** each represent an even parity computation on 7 of the 11 protected bits as shown below (i.e., a parity bit is 0 if the sum of the 7 protected bits is even).

HP-0 --PD(1)-5, PD(1)-6, PG(1)-4, PG(2)-4, PD(3)-6, PG(3)-4, and SP
 HP-1 --PD(1)-5, PD(1)-7, PG(1)-4, PD(3)-5, PD(3)-6, PG(4)-4, and Sp
 HP-2 --PD(1)-6, PD(1)-7, PG(1)-4, PD(3)-7, PG(3)-4, PG(4)-4, and SP
 HP-3 --PG(2)-4, PD(3)-5, PD(3)-6, PD(3)-7, PG(3)-4, PG(4)-4, and SP

3.9.2 Decoding. The 4 parity bits (HP-0,1,2, and 3) shall be used to correct single errors in the 11 protected bits as follows. Parity bits are calculated on the received 11 protected bits as described in section 3.9.1. Then, these calculated parity bits are Exclusive ORed with the received parity bits. The following table shows how to correct for a single bit error by reversing the binary value of one of the protected bits based upon the resulting binary value of the Exclusive ORed parity bits.

Parity	Result	Invert	Parity	Result	Invert
3	PD(1)-5		11	PD(3)-6	
5	PD(1)-6		12	PD(3)-7	
6	PD(1)-7		13	PG(3)-4	
7	PG(1)-4		14	PG(4)-4	
9	PG(2)-4		15	SP	
10	PD(3)-5				

3.10 General Application Notes

3.10.1 Amplitude Scaling. "Stochastic" elements and gain values contained in this standard are based upon use of input digitized voice that can assume any integer value in the range -32,768 to +32,767 and an impulse of 1.0 to calculate the impulse response of the perceptual weighting filter.

3.10.2 Filter Structure. Various filter structures may be used in the implementation of this standard. All implementations must be interoperable with implementations based upon Direct Form (i.e., block-wise) filtering.

3.10.3 Smoothing. It is desirable to employ smoothing to prevent loud clipped voice (i.e., blasts) and squeaks. Smoothing based upon estimates of the channel error rate (provided by the error correction mechanism described in section 3.9) is recommended.

4. EFFECTIVE DATE. The use of this standard by U.S. Government departments and agencies is mandatory effective 180 days after the date of this standard.

5. CHANGES. When a Government department or agency considers that this standard does not provide for its essential needs, a statement citing specific requirements shall be sent in duplicate to the General Services Administration (K), Washington, DC 20405, in accordance with the provisions of Federal Information Resources Management Regulation 41 CFR 201-20. The General Services Administration will determine the appropriate action to be taken and will notify the agency.

FED-STD 1016

TECHNICAL DEVELOPMENT ACTIVITY:

National Security Agency
Information Security Organization (R2)
Fort George G. Meade, MD 20755

PREPARING ACTIVITY:

National Communications System
Office of Technology and Standards
Washington, DC 20305-2010

MILITARY INTERESTS

Military Coordinating Activity

DCA -- DC

Custodians

Army -- SC

Navy -- EC

Air Force --90

Review Activities

Army -- CR

Navy -- AS,OM

Air Force -- 17

USMC -- MC

NSA -- NS