INTERNATIONAL TELECOMMUNICATION UNION

CCITT G.726

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

GENERAL ASPECTS OF DIGITAL TRANSMISSION SYSTEMS; TERMINAL EQUIPMENTS

40, 32, 24, 16 kbit/s ADAPTIVE DIFFERENTIAL PULSE CODE MODULATION (ADPCM)

Recommendation G.726

Geneva, 1990

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation G.726 was prepared by Study Group XV and was approved under the Resolution No. 2 procedure on the 14 of December 1990.

CCITT NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication Administration and a recognized private operating agency.

© ITU 1990

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

Recommendation G.726

40, 32, 24, 16 kbit/s ADAPTIVE DIFFERENTIAL PULSE CODE MODULATION (ADPCM)[1)]

1 General

The characteristics below are recommended for the conversion of a 64 kbit/s A-law or μ-law pulse code modulation (PCM) channel to and from a 40, 32, 24 or 16 kbit/s channel. The conversion is applied to the PCM bit stream using an ADPCM transcoding technique. The relationship between the voice frequency signals and the PCM encoding/decoding laws is fully specified in Recommendation G.711.

The principal application of 24 and 16 kbit/s channels is for overload channels carrying voice in Digital Circuit Multiplication Equipment (DCME).

The principal application of 40 kbit/s channels is to carry data modem signals in DCME, especially for modems operating at greater than 4800 kbit/s.

Sections 1.1 and 1.2 of this Recommendation provide an outline description of the ADPCM transcoding algorithm, §§ 2 and 3 provide the principles and functional descriptions of the ADPCM encoding and decoding algorithms respectively, whilst § 4 is the precise specification for the algorithm computations. Networking aspects and digital test sequences are addressed in Appendices I and II, respectively, to this Recommendation.

Simplified block diagrams of both the ADPCM encoder and decoder are shown in Figure 1/G.726.

In § 4, each sub-block in the encoder and decoder is precisely defined using one particular logical sequence. If other methods of computation are used, extreme care should be taken to ensure that they yield *exactly* the same value for the output processing variables. Any further departures from the processes detailed in § 4 will incur performance penalties which may be severe.

Note 1 – Prior to the definition of this Recommendation, other ADPCM algorithms of performance similar to the 40 kbit/s algorithm specified here have been incorporated in DCME designs and used in telecommunications networks. These algorithms may be considered by bilateral agreement for limited DCME applications under certain circumstances. Technical descriptions providing information on two such algorithm approaches can be found in COM XVIII No. 101 and COM XVIII No. 102 of the 1984-1988 Study Period.

Note 2 – The assignment of 16, 24, 32 and 40 kbit/s DCME channels and the associated selection of coding rates are beyond the scope of this Recommendation; see, for example, Recommendation G.763 (revised, 1990).

Note 3 – Signalling and multiplexing considerations are beyond the scope of this Recommendation; see, for example, Recommendations G.761 and G.763 (revised, 1990).

---

1) This Recommendation completely replaces the text of Recommendations G.721 and G.723 published in Volume III.4 of the Blue Book. It should be noted that systems designed in accordance with the present Recommendation will be compatible with systems designed in accordance with the Blue Book version.

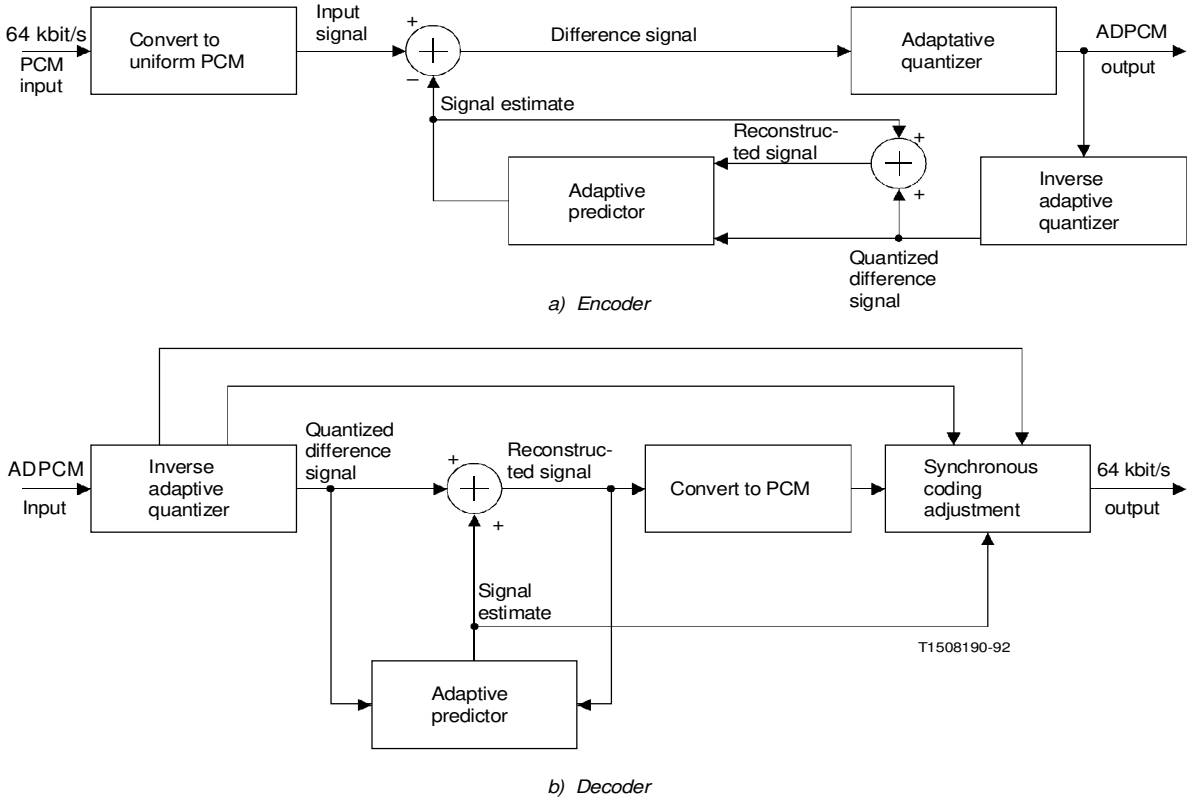

FIGURE 1/G.726

Simplified block diagrams

1.1 *ADPCM encoder*

Subsequent to the conversion of the A-law or μ-law PCM input signal to uniform PCM, a difference signal is obtained, by subtracting an estimate of the input signal from the input signal itself. An adaptive 31-, 15-, 7-, or 4-level quantizer is used to assign five, four, three or two binary digits, respectively, to the value of the difference signal for transmission to the decoder. An inverse quantizer produces a quantized difference signal from these same five, four, three or two binary digits, respectively. The signal estimate is added to this quantized difference signal to produce the reconstructed version of the input signal. Both the reconstructed signal and the quantized difference signal are operated upon by an adaptive predictor which produces the estimate of the input signal, thereby completing the feedback loop.

1.2 *ADPCM decoder*

The decoder includes a structure identical to the feedback portion of the encoder, together with a uniform PCM to A-law or μ-law conversion and a synchronous coding adjustment.

The synchronous coding adjustment prevents cumulative distortion occurring on synchronous tandem codings (ADPCM-PCM-ADPCM, etc., digital connections) under certain conditions (see § 3.7). The synchronous coding adjustment is achieved by adjusting the PCM output codes in a manner which attempts to eliminate quantizing distortion in the next ADPCM encoding stage.

2 ADPCM encoder principles

Figure 2/G.726 is a block schematic of the encoder. For each variable to be described, k is the sampling index and samples are taken at 125 µ s intervals. A fundamental description of each block is given below in §§ 2.1 to 2.8.

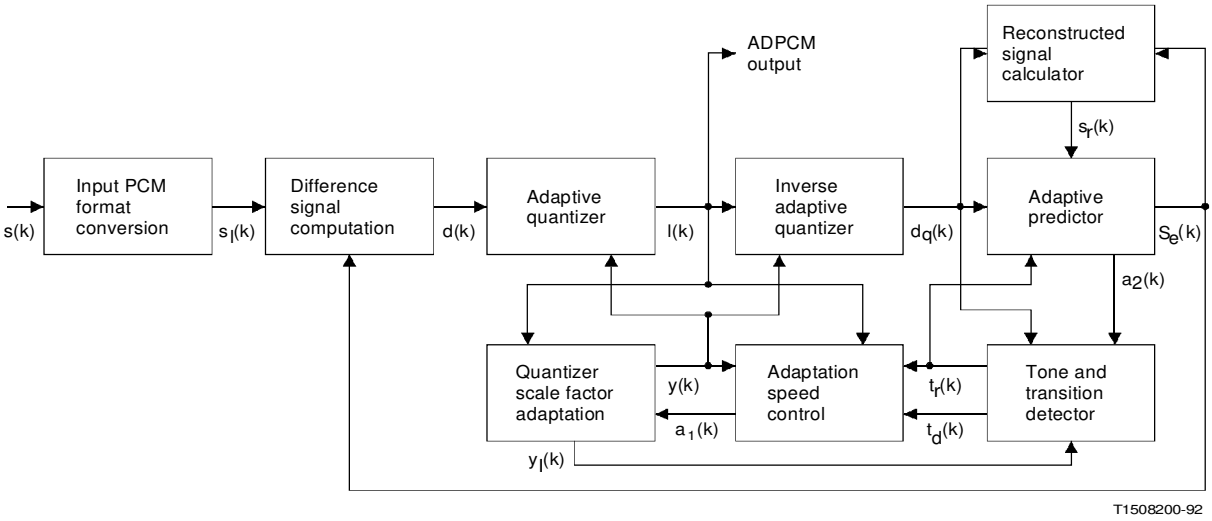

FIGURE 2/G.726

Encoder block schematic

2.1 *Input PCM format conversion*

This block converts the input signal $s(k)$ from A-law or µ-law PCM to a uniform PCM signal $s_l(k)$.

2.2 *Difference signal computation*

This block calculates the difference signal $d(k)$ from the uniform PCM signal $s_l(k)$ and the signal estimate $s_e(k)$:

$$d(k) = s_l(k) - s_e(k) \tag{2-1}$$

2.3 *Adaptive quantizer*

A 31-, 15, 7- or 4-level non-uniform adaptive quantizer is used to quantize the difference signal $d(k)$ for operating at 40, 32, 24 or 16 kbit/s, respectively. Prior to quantization, $d(k)$ is converted to a base 2 logarithmic representation and scaled by $y(k)$ which is computed by the scale factor adaptation block. The normalized input/output characteristic (infinite precision values) of the quantizer is given in Tables 1/G.726 through 4/G.726.

2.3.1 *Operation at 40 kbit/s*

Five binary digits are used to specify the quantized level representing $d(k)$ (four for the magnitude and one for the sign). The 5-bit quantizer output $I(k)$ forms the 40 kbit/s output signal; $I(k)$ takes on one of 31 non-zero values, $I(k)$ is also fed to the inverse adaptive quantizer, the adaptation speed control and the quantizer scale factor adaptation blocks that operate on a 5-bit $I(k)$ having one of 32 possible values. $I(k) = 00000$ is a legitimate input to these blocks when used in the decoder, due to transmission errors.

TABLE 1/G.726

Quantizer normalized input/output characteristic for 40 kbit/s operation

Normalized quantizer input range $\log_2 \mid d(k) \mid - y(k)$	$\mid I(k) \mid$	Normalized quantizer output $\log_2 \mid d_q(k) \mid - y(k)$
[4.31, + ∞)	15	4.42
[4.12, 4.31)	14	4.21
[3.91, 4.12)	13	4.02
[3.70, 3.91)	12	3.81
[3.47, 3.70)	11	3.59
[3.22, 3.47)	10	3.35
[2.95, 3.22)	9	3.09
[2.64, 2.95)	8	2.80
[2.32, 2.64)	7	2.48
[1.95, 2.32)	6	2.14
[1.54, 1.95)	5	1.75
[1.08, 1.54)	4	1.32
[0.52, 1.08)	3	0.81
[–0.13, 0.52)	2	0.22
[–0.96, –0.13)	1	–0.52
(– ∞, –0.96)	0	–∞

Note – In Tables 1/G.726 through 4/G.726, "[" indicates that the endpoint value is included in the range, and "(" or ")" indicates that the endpoint value is excluded from the range.

2.3.2 *Operation at 32 kbit/s*

Four binary digits are used to specify the quantized level representing $d(k)$ (three for the magnitude and one for the sign). The 4-bit quantizer output $I(k)$ forms the 32 kbit/s output signal; it is also fed to the inverse adaptive quantizer, the adaptation speed control and the quantizer scale factor adaptation blocks. $I(k) = 0000$ is a legitimate input to these blocks when used in the decoder, due to transmission errors.

TABLE 2/G.726

Quantizer normalized input/output characteristic for 32 kbit/s operation

Normalized quantizer input range $\log_2 \mid d(k) \mid - y(k)$	$\mid I(k) \mid$	Normalized quantizer output $\log_2 \mid d_q(k) \mid - y(k)$
[3.12, + ∞)	7	3.32
[2.72, 3.12)	6	2.91
[2.34, 2.72)	5	2.52
[1.91, 2.34)	4	2.13
[1.38, 1.91)	3	1.66
[0.62, 1.38)	2	1.05
[–0.98, 0.62)	1	0.031
(– ∞,–0.98)	0	– ∞

2.3.3 *Operation at 24 kbit/s*

Three binary digits are used to specify the quantized level representing $d(k)$ (two for the magnitude and one for the sign). The 3-bit quantizer output $I(k)$ forms the 24 kbit/s output signal, where $I(k)$ takes on one of sevel non-zero values. $I(k)$ is also fed to the inverse adaptive quantizer, the adaptation speed control and the quantizer scale factor adaptation blocks, each of which is modified to operate on a 3-bit $I(k)$ having any of the eight possible values. $I(k) = 000$ is a legitimate input to these blocks when used in the decoder, due to transmission errors.

TABLE 3/G.726

Quantizer normalized input/output characteristic for 24 kbit/s operation

Normalized quantizer input range $\log_2 \mid d(k) \mid - y(k)$	$\mid I(k) \mid$	Normalized quantizer output $\log_2 \mid d_q(k) \mid - y(k)$
[2.58, + ∞)	3	2.91
[1.70, 2.58)	2	2.13
[0.06, 1.70)	1	1.05
(– ∞,–0.06)	0	– ∞

2.3.4 *Operation at 16 kbit/s*

Two binary digits are used to specify the quantized level representing $d(k)$ (one for the magnitude and one for the sign). The 2-bit quantizer output $I(k)$ forms the 16 kbit/s output signal; it is also fed to the inverse adaptive quantizer, the adaptation speed control and the quantizer scale factor adaptation blocks.

TABLE 4/G.726

Quantizer normalized input/output characteristic for 16 kbit/s operation

Normalized quantizer input range $\log_2 \mid d(k) \mid - y(k)$	$\mid I(k) \mid$	Normalized quantizer output $\log_2 \mid d_q(k) \mid - y(k)$
[2.04, + ∞)	1	2.85
(−∞, −2.04)	0	0.91

Unlike the quantizers described in § 2.3.1 for operation at 40 kbit/s, in § 2.3.2 for operation at 32 kbit/s and in § 2.3.3 for operation at 24 kbit/s, the quantizer for operation at 16 kbit/s is an even-level (4-level) quantizer. The even-level quantizer for the 16 kbit/s ADPCM has been selected because of its superior performance over a corresponding odd-level (3-level) quantizer.

2.4 *Inverse adaptive quantizer*

A quantized version $d_q(k)$ of the difference signal is produced by scaling, using $y(k)$, specific values selected from the normalized quantizing characteristic given in Tables 1/G.726 through 4/G.726 and then transforming the result from the logarithmic domain.

2.5 *Quantizer scale factor adaptation*

This block computes $y(k)$, the scaling factor for the quantizer and the inverse quantizer. The inputs are the 5-bit, 4-bit, 3-bit, 2-bit quantizer output $I(k)$ and the adaptation speed control parameter $a_l(k)$.

The basic principle used in scaling the quantizer is bimodal adaptation:

– fast for signals (e.g. speech) that produce difference signals with large fluctuations;

– slow for signals (e.g. voiceband data, tones) that produce difference signals with small fluctuations.

The speed of adaptation is controlled by a combination of fast and slow scale factors.

The fast (unlocked) scale factor $y_u(k)$ is recursively computed in the base 2 logarithmic domain from the resultant logarithmic scale factor $y(k)$:

$$y_u(k) = (1 - 2^{-5})\, y(k) + 2^{-5}\, W[I(k)], \tag{2-2}$$

where $y_u(k)$ is limited by $1.06 \le y_u(k) \le 10.00$.

For 40 kbit/s ADPCM, the discrete function $W(I)$ is defined as follows (infinite precision values):

| I(k) |	15	14	13	12	11	10	9	8
W[I(k)]	43.50	33.06	27.50	22.38	17.50	13.69	11.19	8.81

| I(k) |	7	6	5	4	3	2	1	0
W[I(k)]	6.29	3.63	2.56	2.50	2.44	1.50	0.88	0.88

For 32 kbit/s ADPCM, the discrete function *W(I)* is defined as follows (infinite precision values):

| I(k) |	7	6	5	4	3	2	1	0
W[I(k)]	70.13	22.19	12.38	7.00	4.00	2.56	1.13	–0.75

For 24 kbit/s ADPCM, the discrete function *W(I)* is defined as follows (infinite precision values):

| I(k) |	3	2	1	0
W[I(k)]	36.38	8.56	1.88	–0.25

For16 kbit/s APDCM, the discrete function *W(I)* is defined as follows (infinite precision values):

| I(k) |	1	0
W[I(k)]	27.44	–1.38

The factor $(1 - 2^{-5})$ introduces finite memory into the adaptive process so that the states of the encoder and decoder converge following transmission errors.

The slow (locked) scale factor $y_l(k)$ is derived from $y_u(k)$ with a low pass-filter operation:

$$y_l(k) = (1 - 2^{-6})\, y_l(k-1) + 2^{-6}\, y_u(k) \tag{2-3}$$

The fast and slow scale factors are then combined to form the resultant scale factor:

$$y(k) = a_l(k)\, y_u(k-1) + [1 - a_l(k)]\, y_l(k-1) \tag{2-4}$$

where $0 \le a_l(k) \le 1$ (see § 2.6).

2.6 *Adaptation speed control*

The controlling parameter $a_l(k)$ can assume values in the range [0, 1]. It tends towards unity for speech signals and towards zero for voiceband data signals. It is derived from a measure of the rate-of-change of the difference signal values.

Two measures of the average magnitude of $I(k)$ are computed:

$$d_{ms}(k) = (1 - 2^{-5})\, d_{ms}(k-1) + 2^{-5}\, F[I(k)] \tag{2-5}$$

and

$$d_{ml}(k) = (1 - 2^{-7})\, d_{ml}(k-1) + 2^{-7}\, F[I(k)] \tag{2-6}$$

For 40 kbit/s ADPCM, $F[I(k)]$ is defined by:

| I(k) |	15	14	13	12	11	10	9	8
F[I(k)]	6	6	5	4	3	2	1	1

| I(k) |	7	6	5	4	3	2	1	0
F[I(k)]	1	1	1	0	0	0	0	0

For 32 kbit/s ADPCM, $F[I(k)]$ is defined by:

| I(k) |	7	6	5	4	3	2	1	0
F[I(k)]	7	3	1	1	1	0	0	0

For 24 kbit/s ADPCM, $F[I(k)]$ is defined by:

| I(k) |	3	2	1	0
F[I(k)]	7	2	1	0

For 16 kbit/s ADPCM, $F[I(k)]$ is defined by:

| I(k) |	1	0
F[I(k)]	7	0

Thus $d_{ms}(k)$ is a relatively short term average of $F[I(k)]$ and $d_{ml}(k)$ is a relatively long term average of $F[I(k)]$.

Using these two averages, the variable $a_p(k)$ is defined:

$$a_p(k) = \begin{cases} (1 - 2^{-4})a_p(k-1) + 2^{-3}, \text{ if } | d_{ms}(k) - d_{ml}(k) | \geq 2^{-3}\, d_{ml}(k) \\ (1 - 2^{-4})a_p(k-1) + 2^{-3}, \text{ if } y(k) < 3 \\ (1 - 2^{-4})a_p(k-1) + 2^{-3}, \text{ if } t_d(k) = 1 \\ 1, \text{ if } t_r(k) = 1 \\ (1 - 2^{-4})a_p(k-1), \text{ otherwise} \end{cases} \tag{2-7}$$

Thus, $a_p(k)$ tends towards the value 2 if the difference between $d_{ms}(k)$ and $d_{ml}(k)$ is large (average magnitude of $I(k)$ changing) and $a_p(k)$ tends towards the value 0 if the difference is small (average magnitude of $I(k)$ relatively constant). $a_p(k)$ also tends towards 2 for idle channel (indicated by $y(k) < 3$) or partial band signals (indicated by $t_d(k) = 1$ as described in § 2.8). Note that $a_p(k)$ is set to 1 upon detection of a partial band signal transition (indicated by $t_r(k) = 1$, see § 2.8).

$a_p(k-1)$ is then limited to yield $a_l(k)$ used in Equation (2-4) above:

$$a_l(k) = \begin{cases} 1, & a_p(k-1) > 1 \\ a_p(k-1), & a_p(k-1) \leq 1. \end{cases} \tag{2-8}$$

This asymmetrical limiting has the effect of delaying the start of a fast to slow state transition until the absolute value of $I(k)$ remains constant for some time. This tends to eliminate premature transitions for pulsed input signals such as switched carrier voiceband data.

2.7 *Adaptive predictor and reconstructed signal calculator*

The primary function of the adaptive predictor is to compute the signal estimate $s_e(k)$ from the quantized difference signal $d_q(k)$. Two adaptive predictor structures are used, a sixth order section that models zeros and a second order section that models poles in the input signal. This dual structure effectively caters for the variety of input signals which might be encountered.

The signal estimate is computed by:

$$s_e(k) = \sum_{i=1}^{2} a_i(k-1)\, s_r(k-i) + s_{ez}(k), \tag{2-9}$$

where

$$s_{ez}(k) = \sum_{i=1}^{6} b_i(k-1)\, d_q(k-i),$$

and the reconstructed signal is defined as

$$s_r(k-i) = s_e(k-i) + d_q(k-i).$$

Both sets of predictor coefficients are updated using a simplified gradient algorithm:

for the second order predictor:

$$a_1(k) = (1 - 2^{-8})a_1(k-1) + (3 \cdot 2^{-8})\ \mathrm{sgn}\,[p(k)]\ \mathrm{sgn}\,[p(k-1)], \tag{2-10}$$

$$\begin{aligned} a_2(k) = (1\ - 2^{-7})a_2(k-1) + 2^{-7}\ \big\{&\mathrm{sgn}\,[p(k)]\ \mathrm{sgn}\,[p(k-2)] \\ &- f\,[a_1(k-1)]\ \mathrm{sgn}\,[p(k)]\ \mathrm{sgn}\,[p(k-1)]\big\}, \end{aligned} \tag{2-11}$$

where

$$p(k) = d_q(k) + s_{ez}(k),$$

$$f(a_1) = \begin{cases} 4a_1, & |\ a_1\ | \le 2^{-1} \\ 2\ \mathrm{sgn}\ (a_1), & |\ a_1\ | > 2^{-1}, \end{cases}$$

and sgn [0] = 1, except sgn [$p(k-i$] is defined to be 0 only if $p(k-i) = 0$ and $i = 0$;

with the stability constraints:

$$|\ a_2(k)\ | \le 0.75 \text{ and } |\ a_1(k)\ | \le 1 - 2^{-4} - a_2(k).$$

If $t_r(k) = 1$ (see § 2.8), then $a_1(k) = a_2(k) = 0$.

For the sixth order predictor:

$$b_i(k) = (1 - 2^{-8})b_i(k-1) + 2^{-7}\ \mathrm{sgn}\,[d_q(k)]\ \mathrm{sgn}\,[d_q(k-i)], \tag{2-12A}$$

for $i = 1, 2, \ldots, 6$.

For 40 kbit/s coding, the adaptive predictor is changed to decrease the leak factor used for zeros coefficient operation. In this case, Equation 2.12A becomes:

$$b_i(k) = (1 - 2^{-9})b_i(k-1) + 2^{-7}\,\mathrm{sgn}\,[d_q(k)]\,\mathrm{sgn}\,[d_q(k-i)]. \tag{2-12B}$$

If $t_r(k) = 1$ (see § 2.8), then $b_1(k) = b_2(k) = \ldots = b_6(k) = 0$.

As above, sgn [0] = 1, except sgn $[d_q(k-i)]$ is defined to be 0 only if $d_q(k-i) = 0$ and $i = 0$. Note that $b_i(k)$ is implicitly limited to ± 2.

2.8 *Tone and transition detector*

In order to improve performance for signals originating from frequency shift keying (FSK) modems operating in the character mode, a two-step detection process is defined. First, partial band signal (e.g. tone) detection is invoked so that the quantizer can be driven into the fast mode of adaptation:

$$t_d(k) = \begin{cases} 1, & a_2(k) < -0.71875 \\ 0, & \text{otherwise} \end{cases} \tag{2-13}$$

In addition, a transition from a partial band signal is defined so that the predictor coefficients can be set to zero and the quantizer can be forced into the fast mode of adaptation:

$$t_r(k) = \begin{cases} 1, & a_2(k) < -0.71875 \text{ and } |\,d_q(k)\,| > 24 \cdot 2^{y_l(k)} \\ 0, & \text{otherwise} \end{cases} \tag{2-14}$$

3 ADPCM decoder principles

Figure 3/G.726 is a block schematic of the decoder. A functional description of each block is given in §§ 3.1 to 3.7 below.

3.1 *Inverse adaptive quantizer*

The function of this block is described in § 2.4.

3.2 *Quantizer scale factor adaptation*

The function of this block is described in § 2.5.

3.3 *Adaptation speed control*

The function of this block is described in § 2.6.

3.4 *Adaptive predictor and reconstructed signal calculator*

The functions of this block are described in § 2.7.

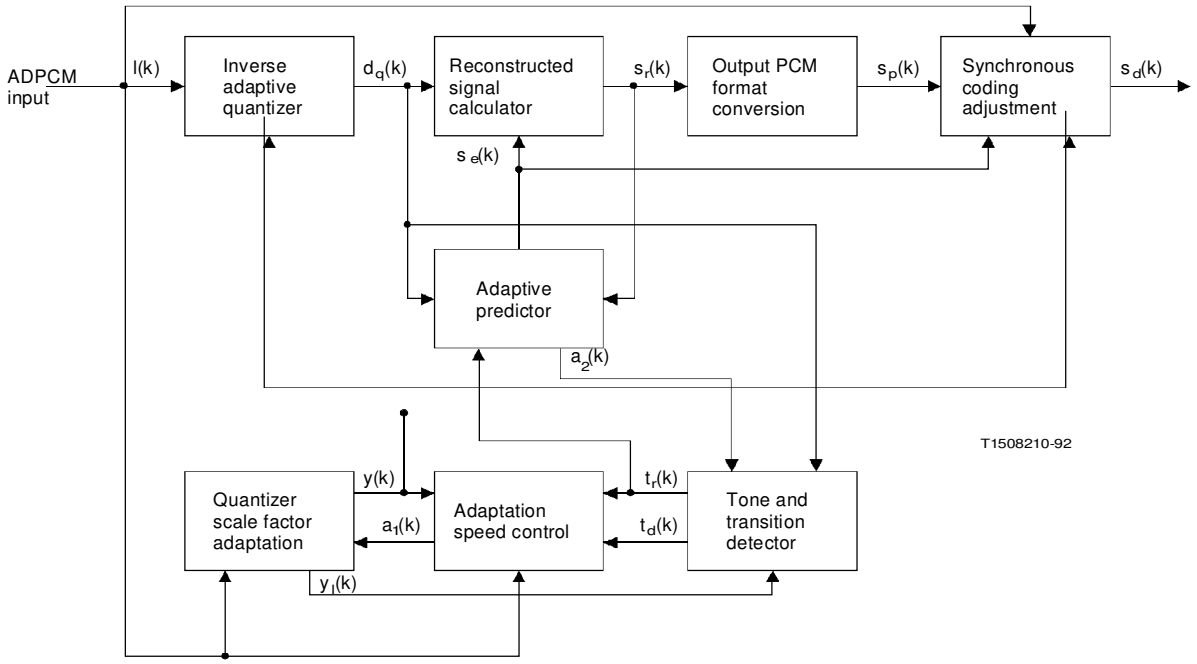

FIGURE 3/G.726

Decoder block schematic

3.5 *Tone and transition detector*

The function of this block is described in § 2.8.

3.6 *Output PCM format conversion*

This block converts the reconstructed uniform PCM signal $s_r(k)$ into an A-law or μ-law PCM signal $s_p(k)$ as required.

3.7 *Synchronous coding adjustment*

The synchronous coding adjustment prevents cumulative distortion occurring on synchronous tandem codings (ADPCM-PCM-ADPCM, etc. digital connections), when:

i) the transmission of the ADPCM and the intermediate 64 kbit/s PCM signals is error free, and,

ii) the ADPCM and intermediate 64 kbit/s PCM bit streams are not disturbed by digital signal processing devices.

If the coder and decoder have different initial conditions, as may occur after switching for example, then the synchronous tandeming may take time to establish. Furthermore, if this property is disturbed or not acquired initially then it may be recovered for those signals of sufficient level with spectra that occupy the majority of the 200 to 3400 Hz band (e.g. speech, 4800 bit/s voiceband data).

When a decoder is synchronously connected to an encoder, the synchronous coding adjustment block estimates quantization in the encoder. If all state variables in both the decoder and the encoder have identical values and there are no transmission errors, the forced equivalence of both 4-bit quantizer output sequences for all values of k guarantees the property of non-accumulation of distortion.

This is accomplished by first converting the A-law or μ-law signal $s_p(k)$ to a uniform PCM signal $s_{lx}(k)$ and then computing a difference signal $d_x(k)$:

$$d_x(k) = s_{lx}(k) - s_e(k). \tag{3-1}$$

The difference signal $d_x(k)$ is then compared to the ADPCM quantizer decision interval determined by $I(k)$ and $y(k)$. The signal $s_d(k)$ is then defined as follows:

$$s_d(k) = \begin{cases} s_p^+(k), & d_x(k) < \text{lower interval boundary} \\ s_p^-(k), & d_x(k) \geq \text{upper interval boundary} \\ s_p(k), & \text{otherwise} \end{cases} \tag{3-2}$$

where

$s_d(k)$ is the output PCM code word of the decoder,

$s^+_p(k)$ is the PCM code word that represents the next more positive PCM output level (when $s_p(k)$ represents the most positive output level, then $s^+_p(k)$: is constrained to be the value $s_p(k)$),

$s^-_p(k)$ is the PCM code word that represents the next more negative PCM output level (when $s_p(k)$ represents the most negative output level, then $s^-_p(k)$: is constrained to be the value $s_p(k)$).

4 Computational details

Sections 4.1 and 4.2 provide the computational details for each of the encoder and decoder elements.

Proper timing for the encoder and decoder is obtained by executing all of the delay blocks simultaneously and proceeding to calculate the signals which can be derived using this information. For example, SE signal of Figure 9/G.726 is calculated using delay values and then SE signal is used as shown in Figure 4/G.726.

Implementations of the algorithm may be confirmed with a reasonable level of confidence by using the digital test sequences described in Appendix II to this Recommendation. The sequences are given in terms of encoder PCM input words, ADPCM words and decoder PCM output words.

4.1 *Input and output signals*

Table 2/G.726 defines the input and output signals for the encoder and decoder.

An optional signal R represents a reset function that sets all internal memory elements to a specified condition so that an encoder or decoder can be forced into a known state, for applications which require an immediate reset function (e.g. digital circuit multiplication equipment, in which case the reset is mandatory, not optional).

TABLE 5/G.726

Input and output signals

ENCODER			
	Name	Number of bits	Description
Input	S	8	PCM input word
Input	LAW	1	PCM law select, 0 = μ-law, 1 = A-law
Input	R (*optional*)	1	Reset
Output	I	5	40 kbit/s ADPCM word
Output	I	4	32 kbit/s ADPCM word
Output	I	3	24 kbit/s ADPCM word
Output	I	2	16 kbit/s ADPCM word
DECODER			
	Name	Number of bits	Description
Input	I	5	40 kbit/s ADPCM word
Input	I	4	32 kbit/s ADPCM word
Input	I	3	24 kbit/s ADPCM word
Input	I	2	16 kbit/s ADPCM word
Input	LAW	1	PCM law select, 0 = μ-law, 1 = A-law
Input	R (*optional*)	1	Reset
Output	SD	8	Decoder PCM output word

4.2 *Description of variables and detailed specification of sub-blocks*

This section contains a detailed expansion of all blocks in Figures 2/G.726 and 3/G.726 described in §§ 2 and 3. The expansions are illustrated in Figures 4/G.726 to 11/G.726 with the internal processing variables as defined in Table 6/G.726. A brief functional description and full specification is given for each sub-block.

The notations used in the sub-block descriptions are as follows:

<<*n* denotes an *n*-bit shift left operation (zero fill),

>>*n* denotes an *n*-bit shift right operation (in the direction of the least significant bit and zero fill),

& denotes the logical “and” operation,

+ denotes arithmetic addition,

– denotes arithmetic subtraction,

* denotes arithmetic multiplication,

** denotes the logical “exclusive or” operation,

| delineates comments to equations.

TABLE 6/G.726

Internal processing variables

Name	Bits	Binary representation	Optional reset values	Description
A1[a], A2[a]	16 TC	S, 0,.., –14	0	Delayed predictor second order coefficients
A1P, A2P	16 TC	S, 0,.., –14		Second order predictor coefficients
A1R, A2R	16 TC	S, 0,.., –14		Triggered second order predictor coefficients
A1T	16 TC	S, 0,.., –14		Unlimited a_1 coefficient
A2T	16 TC	S, 0,.., –14		Unlimited a_2 coefficient
AL	7 SM	0,.., –6		Limited speed control parameter
AP[a]	10 SM	1,.., –8	0	Delayed unlimited speed control parameter
APP	10 SM	1,.., –8		Unlimited speed control parameter
APR	10 SM	1,.., –8		Triggered unlimited speed control parameter
AX	1 SM	1		Speed control parameter update
B1[a],..., B6[a]	16 TC	S, 0,.., –14	0	Delayed sixth order predictor coefficients
B1P,..., B6P	16 TC	S, 0,.., –14		Sixth order predictor coefficients
B1R,..., B6R	16TC	S, 0,.., –14		Triggered sixth order predictor coefficients
D	16 TC	S, 14,.., 0		Difference signal, only in encoder
DL	11 SM	3,.., –7		Log_2 (difference signal), only in encoder
DLN	12 TC	S, 3,.., –7		Log_2 (normalized difference), only in encoder
DLNX	12 TC	S, 3,.., –7		Log_2 (normalized difference), only in decoder
DLX	11 SM	3,.., –7		Log_2 (difference signal), only in decoder
DML[a]	14SM	2,.., –11	0	Delayed long term average of F(I) sequence
DMLP	14 SM	2,.., –11		Long term average of F(I) sequence
DMS[a]	12 SM	2,.., –9	0	Delayed short term average of F(I) sequence
DMSP	12 SM	2,.., –9		Short term average of F(I) sequence
DQ[b]	15 SM	S, 13,.., 0		Quantized difference signal (16, 24 or 32 kbit/s operation)
DQ[b]	16 SM	S, 14,.., 0		Quantized difference signal (16, 24, 32 or 40 kbit/s operation)
DQ0	11 FL	S, 4e, 6m		Quantized difference signal with delay 0
DQ1[a],..., DQ6[a]	11 FL	S, 4e, 6m	32	Quantized difference signal with delays 1 to 6
DQL	12 TC	S, 3,.., –7		Log_2 (quantized difference signal)

a) Indicates variables that are set to specific values by the optional reset. When reset is invoked, the output of the DELAY sub-block (see § 4.2.4) is given in column 4.

b) For 40 kbit/s ADPCM, DQ must be implemented as a 16 bit signed magnitude. For 16, 24 and 32 kbit/s, DQ may be implemented as a 15 or 16 bit signed magnitude.

TC denotes two's complement
SM denotes signed magnitude
FL denotes floating point
e denotes exponent bits
m denotes mantissa bits
S denotes sign bit

TABLE 6/G.726 (*continued*)

Name	Bits	Binary representation	Optional reset values	Description
DQLN	12 TC	S, 3,.., –7		Log_2 (normalized quantized difference)
DQS	1 TC	S		Sign bit of quantized difference signal
DS	1TC	S		Sign bit of difference signal, only in encoder
DSX	1TC	S		Sign bit of difference signal, only in decoder
DX	16 TC	S, 14,.., 0		Difference signal, only in decoder
FI	3 SM	2,.., 0		Output of F(I)
PK0	1 TC	S		Sign of DQ + SEZ with delay 0
PK1[a], PK2[a]	1 TC	S	0	Sign of DQ + SEZ with delays 1 and 2
SE	15 TC	S, 13,.., 0		Signal estimate
SEZ	15 TC	S, 13,.., 0		Sixth order predictor partial signal estimate
SIGPK	1 TC	0		Sgn[p(k)] flag
SL	14 TC	S, 12,.., 0		Linear input signal, only in encoder
SLX	14 TC	S, 12,.., 0		Quantized reconstructed, signal, only in decoder
SP	8			PCM reconstructed signal, only in decoder
SR	16 TC	S, 14,.., 0		Reconstructed signal
SR0	11 FL	S, 4e, 6m		Reconstructed signal with delay 0
SR1[a], SR2[a]	11 FL	S, 4e, 6m	32	Reconstructed signal with delays 1 and 2
TD[a]	1 TC	0	0	Delayed tone detect
TDP	1 TC	0		Tone detect
TDR	1 TC	0		Triggered tone detect
TR	1 TC	0		Transition detect
U1,...,U6	1 TC	S		Sixth order predictor coefficient update sign bit
WA1,WA2	16 TC	S, 13,.., –1		Partial product of signal estimate
WB1,...,WB6	16 TC	S, 13,.., –1		Partial product of signal estimate
WI	12 TC	S, 6,.., –4		Quantizer multiplier
Y	13 SM	3,.., –9		Quantizer scale factor
YL[a]	19 SM	3,.., –15	34816	Delayed slow quantizer scale factor
YLP	19 SM	3,.., –15		Slow quantizer scale factor
YU[a]	13 SM	3,.., –9	544	Delayed fast quantizer scale factor
YUP	13 SM	3,.., –9		Fast quantizer scale factor
YUT	13 SM	3,.., –9		Unlimited fast quantizer scale factor

[a] Indicates variables that are set to specific values by the optional reset. When reset is invoked, the output of the DELAY sub-block (see § 4.2.4) is given in column 4.

TC denotes two's complement
SM denotes signed magnitude
FL denotes floating point
e denotes exponent bits
m denotes mantissa bits
S denotes sign bit

4.2.1 *Input PCM format conversion and difference signal computation*

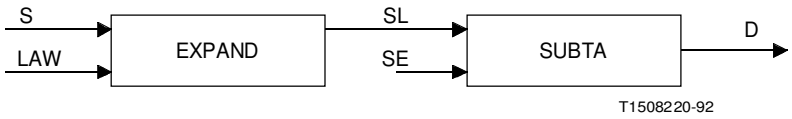

FIGURE 4/G.726

Input PCM format conversion and difference signal computation

EXPAND

Input: S (SP in decoder), LAW
Output: SL (SLX in decoder)
Function: Convert either A-law or μ-law PCM to uniform PCM.

Decode PCM code word, S, according to Recommendation G.711 using character signals (column 6, before inversion of even bits for A-law) and values at decoder output (see column 7). The values at decoder output, SS, must be represented in 13-bit signed magnitude form for A-law PCM and 14-bit signed magnitude form for μ-law PCM (the sign bit is equal to one for negative values).

Note – For A-law S (and SP) includes even bit inversion (see Note 2 below Table 1/G.711).

when $LAW = 0$, $SSS = SS >> 13$ | μ-law
$SSQ = SS \,\&\, 8191$ |

when $LAW = 1$, $SSS = SS >> 12$ |
$SSM = SS \,\&\, 4095$ | A–law
$SSQ = SSM << 1$ |

then

$$SL = \begin{cases} SSQ, & SSS = 0 \\ (16384 - SSQ) \,\&\, 16383, & SSS = 1 \end{cases}$$

| Convert signed magnitude to two's complement

SUBTA

Inputs: SL (SLX in decoder), SE
Output: D (DX in decoder)
Function: Compute difference signal by subtracting signal estimate from input signal (or quantized reconstructed signal in decoder).

$SLS = SL >> 13$

$$SLI = \begin{cases} SL, & SLS = 0 \\ 49152 + SL, & SLS = 1 \end{cases}$$ | Sign extension

$SES = SE >> 14$

$$SEI = \begin{cases} SE, & SES = 0 \\ 32768 + SE, & SES = 1 \end{cases}$$ | Sign extension

$D = (SLI + 65536 - SEI)\ \&\ 65535$

4.2.2 *Adaptive quantizer*

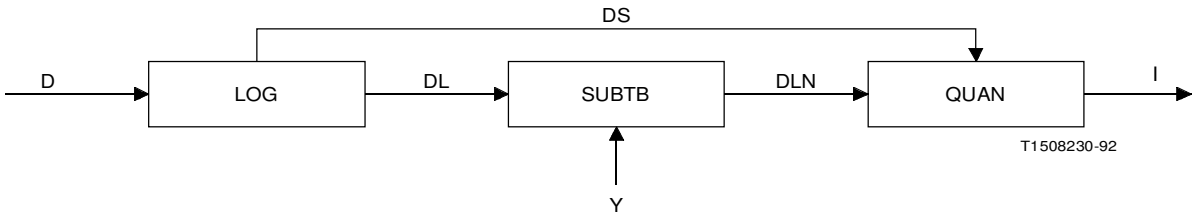

FIGURE 5/G.726
Adaptive quantizer

LOG

Input: D (DX in decoder)
Outputs: DL (DLX in decoder), DS (DSX in decoder)
Function: Convert difference signal from the linear to the logarithmic domain.

$DS = D >> 15$

$$DQM = \begin{cases} D, & DS = 0 \\ (65536 - D)\ \&\ 32767, & DS = 1 \end{cases}$$

| Convert D from two's
| complement to signed
| magnitude

$$EXP = \begin{cases} 14, & 16384 \le DQM \\ 13, & 8192 \le DQM \le 16383 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1, & 2 \le DQM \le 3 \\ 0, & 0 \le DQM \le 1 \end{cases}$$

| Compute exponent

$MANT = ((DQM << 7) >> EXP)\ \&\ 127$

| Compute approximation
| $\log_2 (1 + x) = x$

$DL = (EXP << 7) + MANT$

| Combine 7 mantissa bits and
| 4 exponent bits into one
| 11-bit word

QUAN (encoder only)

Inputs: DLN, DS
Output: I
Function: Quantize difference signal in logarithmic domain.

TABLE 7/G.726

Quantizer decision levels and 5-bit outputs for 40 kbit/s ADPCM

DS	DLN	I	
		12345	
0	553-2047	01111	
0	528- 552	01110	
0	502- 527	01101	
0	475- 501	01100	
0	445- 474	01011	
0	413- 444	01010	
0	378- 412	01001	
0	339- 377	01000	
0	298- 338	00111	
0	250- 297	00110	
0	198- 249	00101	
0	139- 197	00100	
0	68- 138	00011	
0	0- 67	00010	– – | Positive portion of interval
0	4080-4095	00010	– – | Negative portion of interval
0	3974-4079	00001	
0	2048-3973	11111	
1	2048-3973	11111	
1	3974-4079	11110	
1	4080-4095	11101	– – | Negative portion of interval
1	0- 67	11101	– – | Positive portion of interval
1	68- 138	11100	
1	139- 197	11011	
1	198- 249	11010	
1	250- 297	11001	
1	298- 338	11000	
1	339- 377	10111	
1	378- 412	10110	
1	413- 444	10101	
1	445- 474	10100	
1	475- 501	10011	
1	502- 527	10010	
1	528- 552	10001	
1	553-2047	10000	

Note – The I values are transmitted starting with bit 1.

TABLE 8/G.726

Quantizer decision levels and 4-bit outputs for 32 kbit/s ADPCM

DS	DLN	I	
		1234	
0	400-2047	0111	
0	349- 399	0110	
0	300- 348	0101	
0	246- 299	0100	
0	178- 245	0011	
0	80- 177	0010	
0	0- 79	0001	– – | Positive portion of interval
0	3972-4095	0001	– – | Negative portion of interval
0	2048-3971	1111	
1	2048-3971	1111	
1	3972-4095	1110	– – | Negative portion of interval
1	0- 79	1110	– – | Positive portion of interval
1	80- 177	1101	
1	178- 245	1100	
1	246- 299	1011	
1	300- 348	1010	
1	349- 399	1001	
1	400-2047	1000	

Note – The I values are transmitted starting with bit 1.

TABLE 9/G.726

Quantizer decision levels and 3-bit outputs for 24 kbit/s ADPCM

DS	DLN	I	
		123	
0	331-2047	011	
0	218- 330	010	
0	8- 217	001	
0	0- 7	111	– – | Positive portion of interval
0	2048-4095	111	– – | Negative portion of interval
1	2048-4095	111	– – | Negative portion of interval

TABLE 10/G.726

Quantizer decision levels and 2-bit outputs for 16 kbit/s ADPCM

DS	DLN	I	
		12	
0	261-2047	01	
0	0- 260	00	– – | Positive portion of interval
0	2048-4095	00	– – | Negative portion of interval
1	2048-4095	11	– – | Negative portion of interval
1	0- 260	11	– – | Positive portion of interval
1	261-2047	10	

SUBTB

Inputs: DL (DLX in decoder), Y
Output: DLN (DLNX in decoder)
Function: Scale logarithmic version of difference signal by subtracting scale factor.

$DLN = (DL + 4096 - (Y >> 2)) \,\&\, 4095$

4.2.3 *Inverse adaptive quantizer*

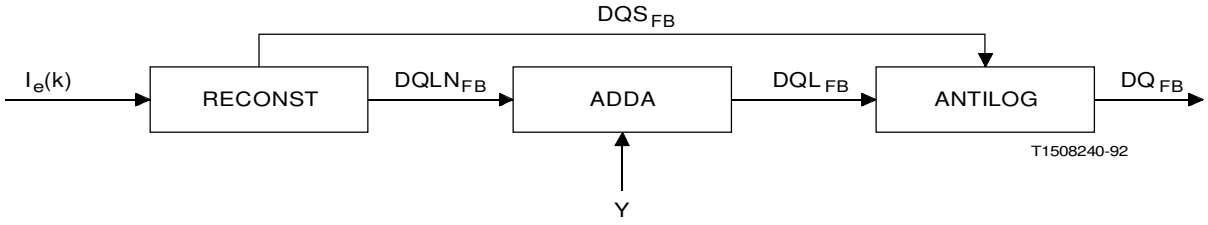

FIGURE 6/G.726

Inverse adaptive quantizer

ADDA

Inputs: DQLN, Y
Output: DQL
Function: Addition of scale factor to logarithmic version of quantized difference signal.

$DQL = (DQLN + (Y >> 2)) \,\&\, 4095$

ANTILOG

Inputs: DQL, DQS
Output: DQ
Function: Convert quantized difference signal from the logarithmic to the linear domain.

$DS = DQL >> 11$ | Extract 4-bit exponent
$DEX = (DQL >> 7) \& 15$

$DMN = DQL \& 127$ | Extract 7-bit mantissa

$DQT = (1 << 7) + DMN$ | Convert mantissa to linear using
$DQMAG = \begin{cases} (DQT << 7) >> (14 - DEX), & DS = 0 \\ 0, & DS = 1 \end{cases}$ | approximation $2^x = 1 + x$
|

$DQ = \begin{cases} (DQS << 14) + DQMAG: & for\ 15\ SM\ DQ \\ (DQS << 15) + DQMAG: & for\ 16\ SM\ DQ \end{cases}$ | Attach sign bit to signed
| magnitude word

RECONST

Input: I
Outputs: DQLN, DQS
Function: Reconstruction of quantized difference signal in the logarithmic domain.

For 40 kbit/s ADPCM:

$DQS = I >> 4$

TABLE 11/G.726

Quantizer output levels for 40 kbit/s ADPCM

I	DQS	DQLN
12345		
01111	0	566
01110	0	539
01101	0	514
01100	0	488
01011	0	459
01010	0	429
01001	0	395
01000	0	358
00111	0	318
00110	0	274
00101	0	224
00100	0	169
00011	0	104
00010	0	28
00001	0	4030
00000	0	2048
11111	1	2048
11110	1	4030
11101	1	28
11100	1	104
11011	1	169
11010	1	224
11001	1	274
11000	1	318
10111	1	358
10110	1	395
10101	1	429
10100	1	459
10011	1	488
10010	1	514
10001	1	539
10000	1	566

Note 1 – The I values are received starting with bit 1.

Note 2 – It is possible for the decoder to receive the code word 00000 because of transmission disturbances (e.g., line bit errors).

For 32 kbit/s ADPCM:

$DQS = I >> 3$

TABLE 12/G.726

Quantizer output levels for 32 kbit/s ADPCM

I	DQS	DQLN
1234		
0111	0	425
0110	0	373
0101	0	323
0100	0	273
0011	0	213
0010	0	135
0001	0	4
0000	0	2048
1111	1	2048
1110	1	4
1101	1	135
1100	1	213
1011	1	273
1010	1	323
1001	1	373
1000	1	425

Note 1 – The I values are received starting with bit 1.

Note 2 – It is possible for the decoder to receive the code word 0000 because of transmission disturbances (e.g., line bit errors).

For 24 kbit/s ADPCM:

$DQS = I >> 2$

TABLE 13/G.726

Quantizer output levels for 24 kbit/s ADPCM

I	DQS	DQLN
123		
011	0	373
010	0	273
001	0	135
000	0	2048
111	1	2048
110	1	135
101	1	273
100	1	373

Note 1 – The I values are received starting with bit 1.

Note 2 – It is possible for the decoder to receive the code word 000 because of transmission disturbance (e.g. line bit errors).

For 16 kbit/s ADPCM:

$DQS = I >> 1$

TABLE 14/G.726

Quantizer output levels for 16 kbit/s

I	DQS	DQLN
12		
01	0	365
00	0	116
11	1	116
10	1	365§

Note 1 – The I values are received starting with bit 1.

4.2.4 *Quantizer scale factor adaptation*

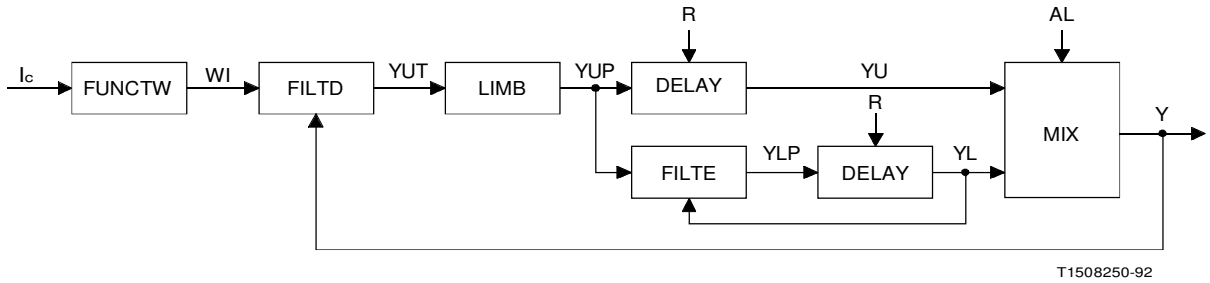

FIGURE 7/G.726
Quantizer scale factor adaptation

DELAY

Inputs: x, R (optional)
Output: y
Function: Memory block. For the input x, the output is given by:

$$y(k) = \begin{cases} x(k-1), & R = 0 \\ \text{optional reset value given in column 4 of Table 6/G.726}, & R = 1 \end{cases}$$ | optional reset

---

FILTD

Inputs: WI, Y
Output: YUT
Function: Update of fast quantizer scale factor.

$DIF = ((WI << 5) + 131072 - Y)\ \&\ 131071, DIFS = DIF >> 16$ | Compute difference
|

$$DIFSX = \begin{cases} DIF >> 5, & DIFS = 0 \\ (DIF >> 5) + 4096, & DIFS = 1 \end{cases}$$ | Time constant is 1/32,
| Sign extension

$YUT = (Y + DIFSX)\ \&\ 8191$

FILTE

Inputs: YUP, YL
Output: YLP
Function: Update of slow quantizer scale factor.

$DIF = (YUP + ((1048576 - YL) >> 6)) \,\&\, 16383$ | Compute difference

$DIFS = DIF >> 13$ | Time constant is 1/64

$$DIFSX = \begin{cases} DIF, & DIFS = 0 \\ DIF + 507904, & DIFS = 1 \end{cases}$$ | Sign extension

$YLP = (YL + DIFSX) \,\&\, 524287$

FUNCTW

Input: I
Output: WI
Function: Map quantizer output into logarithmic version of scale factor multiplier.

For 40 kbit/s ADPCM:

$IS = I >> 4$

$$IM = \begin{cases} I \,\&\, 15, & IS = 0 \\ (31 - I) \,\&\, 15, & IS = 1 \end{cases}$$

$$WI = \begin{cases} 696, & IM = 15 \\ 529, & IM = 14 \\ 440, & IM = 13 \\ 358, & IM = 12 \\ 280, & IM = 11 \\ 219, & IM = 10 \\ 179, & IM = 9 \\ 141, & IM = 8 \\ 100, & IM = 7 \\ 58, & IM = 6 \\ 41, & IM = 5 \\ 40, & IM = 4 \\ 39, & IM = 3 \\ 24, & IM = 2 \\ 14, & IM = 1 \\ 14, & IM = 0 \end{cases}$$ | Scale factor multipliers

For 32 kbit/s ADPCM:

$IS = I >> 3$

$$IM = \begin{cases} I \,\&\, 7, & IS = 0 \\ (15 - I) \,\&\, 7, & IS = 1 \end{cases}$$

$$WI = \begin{cases} 1122, & IM = 7 \\ 355, & IM = 6 \\ 198, & IM = 5 \\ 112, & IM = 4 \\ 64, & IM = 3 \\ 41, & IM = 2 \\ 18, & IM = 1 \\ 4084, & IM = 0 \end{cases} \quad \left| \begin{array}{l} \\ \\ \\ \\ \text{Scale factor multipliers} \\ \\ \\ \\ \end{array} \right.$$

For 24 kbit/s ADPCM:

$IS = I >> 2$

$$IM = \begin{cases} I \,\&\, 3, & IS = 0 \\ (7 - I) \,\&\, 3, & IS = 1 \end{cases}$$

$$WI = \begin{cases} 582, & IM = 3 \\ 137, & IM = 2 \\ 30, & IM = 1 \\ 4092, & IM = 0 \end{cases} \quad \left| \begin{array}{l} \\ \\ \text{Scale factor multipliers} \\ \\ \\ \end{array} \right.$$

For 16 kbit/s ADPCM:

$IS = I >> 1$

$$IM = \begin{cases} I \,\&\, 1, & IS = 0 \\ (3 - I) \,\&\, 1, & IS = 1 \end{cases}$$

$$WI = \begin{cases} 439, & IM = 1 \\ 4074, & IM = 0 \end{cases} \quad \left| \begin{array}{l} \\ \text{Scale factor multipliers} \\ \\ \end{array} \right.$$

LIMB

Input: YUT
Output: YUP
Function: Limit quantizer scale factor.

$GEUL = ((YUT + 11264)\ \&\ 16383) >> 13$
$GELL = ((YUT + 15840)\ \&\ 16383) >> 13$

$$YUP = \begin{cases} 544, & GELL = 1 \\ 5120, & GEUL = 0 \\ \text{YUT}, & \text{otherwise} \end{cases}$$

| Set lower limit to 1.06
| Set upper limit to 10.00

MIX

Inputs: AL, YU, YL
Output: Y
Function: Form linear combination of fast and slow quantizer scale factors.

$DIF = (YU + 16384 - (YL >> 6))\ \&\ 16383$
$DIFS = DIF >> 13$

| Compute difference
|

$$DIFM = \begin{cases} DIF, & DIFS = 0 \\ (16384 - DIF)\ \&\ 8191, & DIFS = 1 \end{cases}$$

| Compute magnitude
| of difference
|

$PRODM = (DIFM * AL) >> 6$

| Compute magnitude
| of product

$$PROD = \begin{cases} PRODM, & DIFS = 0 \\ (16384 - PRODM)\ \&\ 16383, & DIFS = 1 \end{cases}$$

| Convert magnitude to
| two's complement
|

$Y = ((YL >> 6) + PROD)\ \&\ 8191$

4.2.5 *Adaptation speed control*

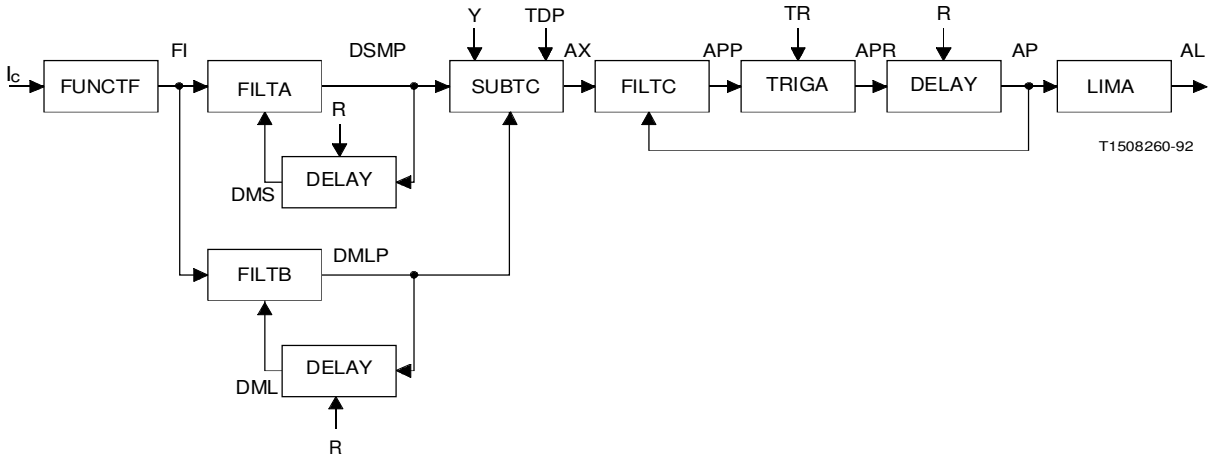

FIGURE 8/G.726
Adaptation speed control

DELAY

See § 4.2.4 for specification.

FILTA

Inputs: FI, DMS
Output: DMSP
Function: Update of short-term average of F(I).

$DIF = ((FI << 9) + 8192 - DMS) \,\&\, 8191$ | Compute difference
$DIFS = DIF >> 12$ |

$$DIFSX = \begin{cases} DIF >> 5, & DIFS = 0 \\ (DIF >> 5) + 3840, & DIFS = 1 \end{cases}$$

| Time constant is 1/32,
| Sign extension
|

$DMSP = (DIFSX + DMS) \,\&\, 4095$

FILTB

Inputs: FI, DML
Output: DMLP
Function: Update of long-term average of F(I).

$DIF = ((FI << 11) + 32768 - DML) \,\&\, 32767$ | Compute difference
$DIFS = DIF >> 14$ |

$$DIFSX = \begin{cases} DIF >> 7, & DIFS = 0 \\ (DIF >> 7) + 16128, & DIFS = 1 \end{cases}$$

| Time constant is 1/28,
| Sign extension
|

$DMLP = (DIFSX + DML) \,\&\, 16383$

FILTC

Inputs: AX, AP
Output: APP
Function: Low pass filter of speed control parameter.

$DIF = ((AX << 9) + 2048 - AP) \,\&\, 2047$ | Compute difference
$DIFS = DIF >> 10$ |

$$DIFSX = \begin{cases} DIF >> 4, & DIFS = 0 \\ (DIF >> 4) + 896, & DIFS = 1 \end{cases}$$

| Time constant is 1/16,
| Sign extension
|

$APP = (DIFSX + AP) \,\&\, 1023$

Input: I
Output: FI
Function: Map quantizer output into the F(I) function.

For 40 kbit/s ADPCM:

$IS = I >> 4$

$$IM = \begin{cases} I \,\&\, 15, & IS = 0 \\ (31 - I) \,\&\, 15, & IS = 1 \end{cases}$$

$$FI = \begin{cases} 0, & 0 \le IM \le 4 \\ 1, & 5 \le IM \le 9 \\ 2, & IM = 10 \\ 3, & IM = 11 \\ 4, & IM = 12 \\ 5, & IM = 13 \\ 6, & IM = 14 \\ 6, & IM = 15 \end{cases}$$

For 32 kbit/s ADPCM:

$IS = I >> 3$

$$IM = \begin{cases} I \,\&\, 7, & IS = 0 \\ (15 - I) \,\&\, 7, & IS = 1 \end{cases}$$

$$FI = \begin{cases} 0, & 0 \le IM \le 2 \\ 1, & 3 \le IM \le 5 \\ 3, & IM = 6 \\ 7, & IM = 7 \end{cases}$$

For 24 kbit/s ADPCM:

$IS = I >> 2$

$$IM = \begin{cases} I \,\&\, 3, & IS = 0 \\ (7 - I) \,\&\, 3, & IS = 1 \end{cases}$$

$$FI = \begin{cases} 0, & IM = 0 \\ 1, & IM = 1 \\ 2, & IM = 2 \\ 7, & IM = 3 \end{cases}$$

For 16 kbit/s ADPCM:

$IS = I >> 1$

$$IM = \begin{cases} I \,\&\, 1, & IS = 0 \\ (3 - I) \,\&\, 1, & IS = 1 \end{cases}$$

$$FI = \begin{cases} 7, & IM = 1 \\ 0, & IM = 0 \end{cases}$$

LIMA

Input: AP
Output: AL
Function: Limit speed control parameter.

$$AL = \begin{cases} 64, & AP \geq 256 \\ AP >> 2, & AP \leq 255 \end{cases}$$

SUBTC

Inputs: DMSP, DMLP, TDP, Y
Output: AX
Function: Compute magnitude of the difference of short and long term functions of quantizer output sequence and then perform threshold comparison for quantizing speed control parameter.

$$DIF = ((DMSP << 2) + 32768 - DMLP) \,\&\, 32767$$
$$DIFS = DIF >> 14$$

| Compute difference

$$DIFM = \begin{cases} DIF, & DIFS = 0 \\ (32768 - DIF) \,\&\, 16383, & DIFS = 1 \end{cases}$$

| Compute magnitude, of difference

$$DTHR = DMLP >> 3$$

$$AX = \begin{cases} 0, & Y \geq 1536 \text{ and } DIFM < DTHR \text{ and } TDP = 0 \\ 1, & \text{otherwise} \end{cases}$$

TRIGA

Inputs: TR, APP
Output: APR
Function: Speed control trigger block.

$$APR = \begin{cases} APP, & TR = 0 \\ 256, & TR = 1 \end{cases}$$

4.2.6 *Adaptative predictor and reconstructed signal calculator*

ACCUM

Inputs: WA1, WA2, WB1, WB2, WB3, WB4, WB5, WB6
Outputs: SE, SEZ
Function: Addition of predictor outputs to form the partial signal estimate (from the sixth order predictor) and the signal estimate.

$SEZI = (((((((WB1 + WB2) \,\&\, 65535) + WB3) \,\&\, 65535) + WB4) \,\&\, 65535) + WB5) \,\&\, 65535) + WB6) \,\&\, 65535$ | Sum for partial signal estimate

$SEI = (((SEZI + WA2) \,\&\, 65535) + WA1) \,\&\, 65535$ | Complete sum for signal estimate

$SEZ = SEZI >> 1$

$SE = SEI >> 1$

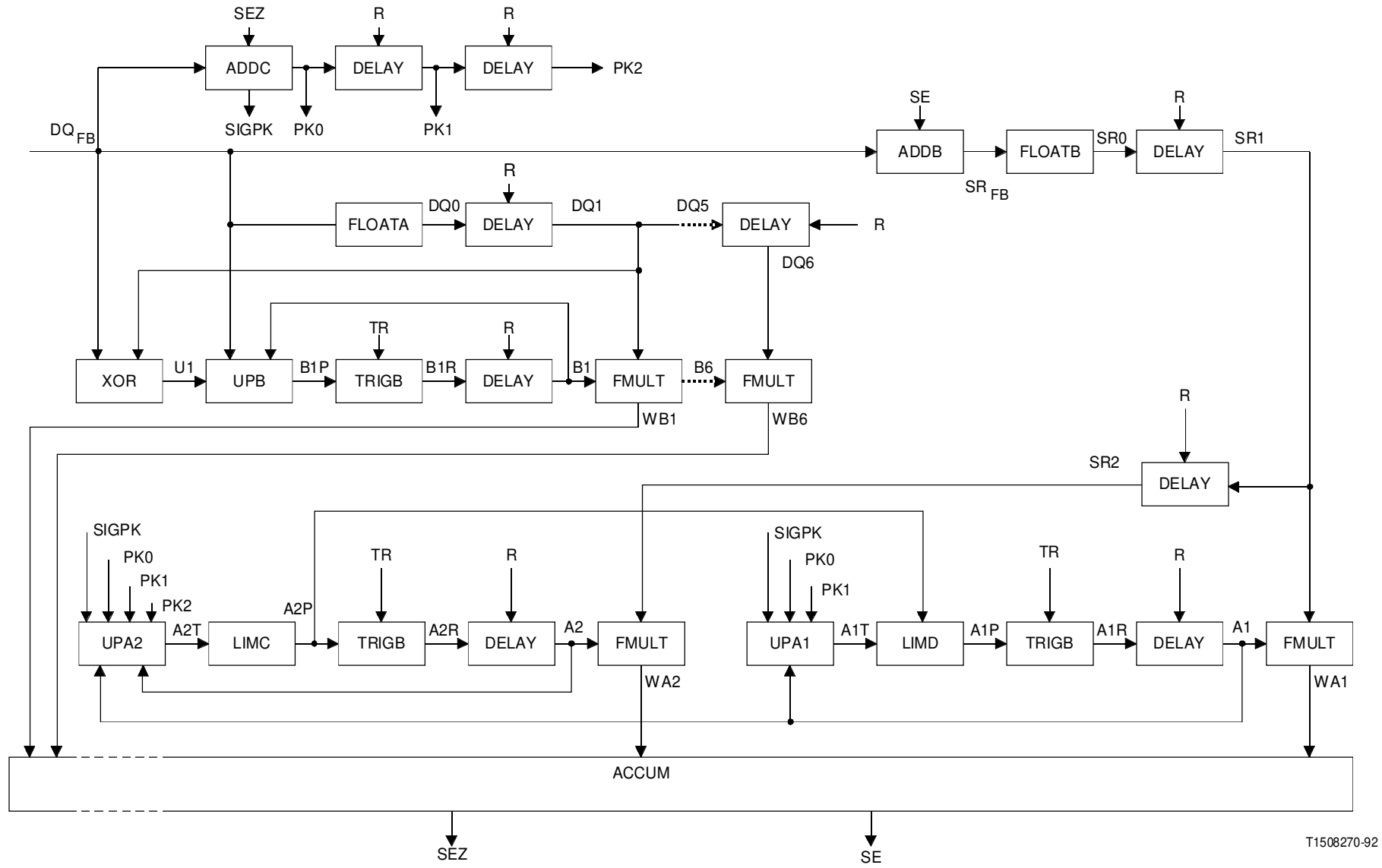

FIGURE 9/G.726

Adaptive predictor and reconstructed signal calculator

ADDB

Inputs: DQ, SE
Output: SR
Function: Addition of quantized difference signal and signal estimate to form reconstructed signal.

$$DQS = \begin{cases} DQ >> 14: \textit{for } 15 \textit{ SM DQ} \\ DQ >> 15: \textit{for } 16 \textit{ SM DQ} \end{cases}$$

$$DQI = \begin{cases} DQ, & DQS = 0 \\ (65536 - (DQ \,\&\, 16383)) \,\&\, 65535, & DQS = 1: \textit{for } 15 \textit{ SM DQ} \\ (65536 - (DQ \,\&\, 32767)) \,\&\, 65535, & DQS = 1: \textit{for } 16 \textit{ SM DQ} \end{cases}$$

| Convert signed magnitude to two's complement

$$SES = SE >> 14$$

$$SEI = \begin{cases} SE, & SES = 0 \\ (1 << 15) + SE, & SES = 1 \end{cases}$$

| Sign extension

$$SR = (DQI + SEI) \,\&\, 65535$$

ADDC

Inputs: DQ, SEZ
Output: PK0, SIGPK
Function: Obtain sign of addition of quantized difference signal and partial signal estimate.

$$DQS = \begin{cases} (DQ >> 14): \textit{for } 15 \textit{ SM DQ} \\ (DQ >> 15): \textit{for } 16 \textit{ SM SQ} \end{cases}$$

$$DQI = \begin{cases} DQ, & DQS = 0 \\ (65536 - (DQ \,\&\, 16383)) \,\&\, 65535, & DQS = 1\text{: } \textit{for } 15 \textit{ SM DQ} \\ (65536 - (DQ \,\&\, 32767)) \,\&\, 65535, & DQS = 1\text{: } \textit{for } 16 \textit{ SM DQ} \end{cases}$$

| Convert signed magnitude to two's complement

$$SEZS = SEZ >> 14$$

$$SEZI = \begin{cases} SEZ, & SEZS = 0 \\ (1 << 15) + SEZ, & SEZS = 1 \end{cases}$$

| Sign extension

$$DQSEZ = (DQI + SEZI) \,\&\, 65535$$

$$PK0 = DQSEZ >> 15$$

$$SIGPK = \begin{cases} 1, & DQSEZ = 0 \\ 0, & \text{otherwise} \end{cases}$$

DELAY

See § 4.2.4 for specification.

FLOATA

Input: DQ
Output: DQ0
Function: Convert 15-bit or 16-bit signed magnitude to floating point.

$$DQS = \begin{cases} DQ >> 14: \textit{for } 15 \textit{ SM DQ} \\ DQ >> 15: \textit{for } 16 \textit{ SM DQ} \end{cases}$$

$$MAG = \begin{cases} DQ \,\&\, 16383: \textit{for } 15 \textit{ SM DQ} \\ DQ \,\&\, 32767: \textit{for } 16 \textit{ SM DQ} \end{cases}$$

| Compute magnitude

$$EXP = \begin{cases} 15, & 16384 \le MAG: \textit{for } 16 \textit{ SM DQ} \\ 14, & 8192 \le MAG \le 16383: \textit{ for } 16 \textit{ SM DQ} \\ 14, & 8192 \le MAG: \textit{for } 15 \textit{ SM DQ} \\ 13, & 4096 \le MAG \le 8191 \\ . & . \\ . & . \\ . & . \\ 2, & 2 \le MAG \le 3 \\ 1, & MAG = 1 \\ 0, & MAG = 0 \end{cases}$$

| Compute exponent

$$MANT = \begin{cases} 1 << 5, & MAG = 0 \\ (MAG << 6) >> EXP, & \text{otherwise} \end{cases}$$

| Compute mantissa with a
| 1 in the most
| significant bit

$$DQ0 = (DQS << 10) + (EXP << 6) + MANT$$

| Combine sign bit, 4 exponent
| bits and 6 mantissa bits
| into one 11-bit word

Input: SR
Output: SR0
Function: Convert 16-bit two's complement to floating point.

$SRS = SR >> 15$

$$MAG = \begin{cases} SR, & SRS = 0 \\ (65536 - SR) \;\&\; 32767, & SRS = 1 \end{cases}$$

| Compute magnitude

$$EXP = \begin{cases} 15, & 16384 \le MAG \\ 14, & 8192 \le MAG \le 16383 \\ . & . \\ . & . \\ . & . \\ 2, & 2 \le MAG \le 3 \\ 1, & MAG = 1 \\ 0, & MAG = 0 \end{cases}$$

| Compute exponent

$$MANT = \begin{cases} 1 << 5, & MAG = 0 \\ (MAG << 6) >> EXP, & \text{otherwise} \end{cases}$$

| Compute mantissa with a 1 in the most significant bit

$SR0 = (SRS << 10) + (EXP << 6) + MANT$

| Combine sign bit, 4 exponent bits and 6 mantissa bits into one 11-bit word

FMULT

Inputs: An or Bn, SRn or DQn
Output: WAn or WBn
Note: Equations are given for An, SRn and WAn. The equations are also valid when substituting Bn for An, DQn for SRn and WBn for WAn.
Function: Multiply predictor coefficients with corresponding quantized difference signal or reconstructed signal. Multiplication is done in floating point format.

$$AnS = An >> 15$$

$$AnMAG = \begin{cases} An >> 2 & AnS = 0 \\ (16384 - (An >> 2)) \,\&\, 8191 & AnS = 1 \end{cases}$$

| Convert two's complement to signed magnitude

$$AnEXP = \begin{cases} 13, & 4096 \le AnMAG \\ 12, & 2048 \le AnMAG \le 4095 \\ . & . \\ . & . \\ . & . \\ 2; & 2 \le AnMAG \le 3 \\ 1, & AnMAG = 1 \\ 0, & AnMAG = 0 \end{cases}$$

| Compute exponent

$$AnMANT = \begin{cases} 1 << 5, & AnMAG = 0 \\ (AnMAG << 6) >> AnEXP, & \text{otherwise} \end{cases}$$

Error!

$$SRnS = SRn >> 10$$
$$SRnEXP = (SRn >> 6) \,\&\, 15$$
$$SRnMANT = SRn \,\&\, 63$$

| Split floating point word into sign bit, exponent and mantissa

$$WAnS = SRnS ** AnS$$
$$WAnEXP = SRnEXP + AnEXP$$
$$WAnMANT = ((SRnMANT * AnMANT) + 48) >> 4$$

| Perform floating point multiplication

$$WAnMAG = \begin{cases} (WAnMANT << 7) >> (26 - WAnEXP), & WAnEXP \le 26 \\ ((WAnMANT << 7) << (WAnEXP - 26)) \,\&\, 32767, & WAnEXP > 26 \end{cases}$$

| Convert floating point to magnitude

$$WAn = \begin{cases} WAnMAG, & WAnS = 0 \\ (65536 - WAnMAG) \,\&\, 65535, & WAnS = 1 \end{cases}$$

| Convert mag. to two's complement

LIMC

Input: A2T
Output: A2P
Function: Limits on a_2 coefficient of second order predictor.

$A2UL = 12288$ | Upper limit of +0.75

$A2LL = 53248$ | Lower limit of –0.75

$$A2P = \begin{cases} A2LL, & 32768 \le A2T \le A2LL \\ A2UL, & A2UL \le A2T \le 32767 \\ A2T, & \text{otherwise} \end{cases}$$

LIMD

Inputs: A1T, A2P
Output: A1P
Function: Limits on a_1 coefficient of second order predictor.

$OME = 15360$ | (1 – epsilon) where
| epsilon = 1/16

$A1UL = (OME + 65536 - A2P)$ & 65535 | Compute upper limit

$A1LL = (A2P + 65536 - OME)$ & 65535 | Compute lower limit

$$A1P = \begin{cases} A1LL, & 32768 \le A1T \text{ and } A1T \le A1LL \\ A1UL, & A1UL \le A1T \text{ and } A1T \le 32767 \\ A1T, & \text{otherwise} \end{cases}$$

TRIGB

Inputs: TR, AnP or BnP or TDP
Output: AnR or BnR or TDR
Note: Equation is given for AnP and AnR. Equation is also valid when substituting BnP and BnR or TDP and TDR for AnP and AnR respectively.
Function: Predictor trigger block.

$$AnR = \begin{cases} AnP, & TR = 0 \\ 0, & TR = 1 \end{cases}$$

UPA1

Inputs: PK0, PK1, A1, SIGPK
Output: A1T
Function: Update a_1 coefficient of second order predictor.

$PKS = PK0 \ast\ast PK1$ | 1-bit "exclusive or"

$$UGA1 = \begin{cases} 192, & PKS = 0 \text{ and } SIGPK = 0 \\ 65344, & PKS = 1 \text{ and } SIGPK = 0 \\ 0, & SIGPK = 1 \end{cases}$$ |,| Gain = ± 3/256,|

$A1S = A1 >> 15$

$$ULA1 = \begin{cases} (65536 - (A1 >> 8)) \,\&\, 65535, & A1S = 0 \\ (65536 - ((A1 >> 8) + 65280)) \,\&\, 65535, & A1S = 1 \end{cases}$$ | Leak factor = 1/256 |

$UA1 = (UGA1 + ULA1) \,\&\, 65535$ | Compute update
$A1T = (A1 + UA1) \,\&\, 65535$ |

UPA2

Inputs: PK0, PK1, PK2, A1, A2, SIGPK
Output: A2T
Function: Update a_2 coefficient of second order predictor.

$PKS1 = PK0 \ast\ast PK1$ | 1-bit "exclusive or"

$PKS2 = PK0 \ast\ast PK2$ | 1-bit "exclusive or"

$$UGA2A = \begin{cases} 16384, & PKS2 = 0 \\ 114688 & PKS2 = 1 \end{cases}$$

$A1S = A1 >> 15$

If $A1S = 0$ then

$$FA1 = \begin{cases} A1 << 2, & A1 \le 8191 \\ 8191 << 2, & A1 \ge 8192 \end{cases}$$

| Implement $f(a_1)$
| with limiting
| at +1/2

If $A1S = 1$, then

$$FA1 = \begin{cases} (A1 << 2) \,\&\, 131071, & A1 \ge 57345 \\ 24577 << 2, & A1 \le 57344 \end{cases}$$

| Implement $f(a_1)$
| with limiting
| at −1/2

$$FA = \begin{cases} FA1, & PKS1 = 1 \\ (131072 - FA1) \,\&\, 131071, & PKS1 = 0 \end{cases}$$

| Attach sign to
| result of $f(a_1)$
|

$UGA2B = (UGA2A + FA) \,\&\, 131071$ |
$UGA2S = UGA2B >> 16$ |

$$UGA2 = \begin{cases} UGA2B >> 7, & UGA2S = 0 \text{ and } SIGPK = 0 \\ (UGA2B >> 7) + 64512, & UGA2S = 1 \text{ and } SIGPK = 0 \\ 0, & SIGPK = 1 \end{cases}$$

|
| Gain calculation,
| gain = ± 1/128
|
|
|

$A2S = A2 >> 15$

$$ULA2 = \begin{cases} (65536 - (A2 >> 7)) \,\&\, 65535, & A2S = 0 \\ (65536 - ((A2 >> 7) + 65024)) \,\&\, 65535, & A2S = 1 \end{cases}$$

|
| Leak factor is
| 1/128
|

$UA2 = (UGA2 + ULA2) \,\&\, 65535$ | Compute update
$A2T = (A2 + UA2) \,\&\, 65535$ |

UPB

Inputs: Un, Bn, DQ
Output: BnP
Function: Update for coefficients of sixth order predictor.

For 40 kbit/s ADPCM (16 SM DQ):

$DQMAG = DQ$ & 32767

$$UGBn = \begin{cases} 128, & Un = 0 \text{ and } DQMAG \neq 0 \\ 65408, & Un = 1 \text{ and } DQMAG \neq 0 \\ 0, & DQMAG = 0 \end{cases}$$

| Gain = ± 1/128 or 0

$BnS = Bn >> 15$

$$ULBn = \begin{cases} (65536 - (Bn >> 9)) \,\&\, 65535, & BnS = 0 \\ (65536 - ((Bn >> 9) + 65408)) \,\&\, 65535 & BnS = 1 \end{cases}$$

| Leak factor = 1/512

$UBn = (UGBn + ULBn)$ & 65535
$BnP = (Bn + UBn)$ & 65535

| Compute update

For 32, 24 and 16 kbit/s ADPCM (15 or 16 SM DQ):

$$DQMAG = \begin{cases} DQ \,\&\, 16383: \textit{for } 15 \textit{ SM DQ} \\ DQ \,\&\, 32767: \textit{for } 16 \textit{ SM DQ} \end{cases}$$

$$UGBn = \begin{cases} 128, & Un = 0 \text{ and } DQMAG \neq 0 \\ 65408, & Un = 1 \text{ and } DQMAG \neq 0 \\ 0, & DQMAG = 0 \end{cases}$$

| Gain = ± 1/128 or 0

$BnS = Bn >> 15$

$$ULBn = \begin{cases} (65536 - (Bn >> 8)) \,\&\, 65535, & BnS = 0 \\ (65536 - ((Bn >> 8) + 65280)) \,\&\, 65535 & BnS = 1 \end{cases}$$

| Leak factor = 1/256

$UBn = (UGBn + ULBn)$ & 65535
$BnP = (Bn + UBn)$ & 65535

| Compute update

XOR

Inputs: DQn, DQ
Output: Un
Function: One bit "exclusive or" of sign of difference signal and sign of delayed difference signal.

$$DQS = \begin{cases} DQ >> 14: \textit{for } 15 \textit{ SM } DQ \\ DQ >> 15: \textit{for } 16 \textit{ SM } DQ \end{cases}$$

$$DQnS = DQn >> 10$$

$$Un = DQS \text{ ** } DQnS \qquad \text{| 1-bit "exclusive or"}$$

4.2.7 *Tone and transition detector*

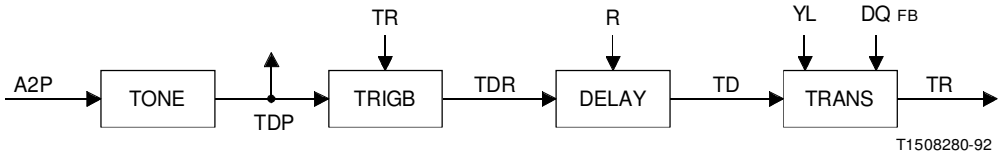

FIGURE 10/G.726
Tone and transition detector

DELAY

See § 4.2.4 for specification.

TONE

Input: A2P
Output: TDP
Function: Partial band signal detection.

$$TDP = \begin{cases} 1, & 32768 \le A2P \text{ and } A2P < 53760 \\ 0, & \text{otherwise} \end{cases}$$

TRANS

Inputs: TD, YL, DQ
Output: TR
Function: Transition detector.

$$DQMAG = \begin{cases} DQ \ \& \ 16383: & for\ 15\ SM\ DQ \\ DQ \ \& \ 32767 \ : & for\ 16\ SM\ DQ \end{cases}$$

$$YLINT = YL >> 15$$

$$YLFRAC = (YL >> 10) \ \& \ 31$$

$$THR1 = (32 + YLFRAC) << YLINT$$

$$THR2 = \begin{cases} 31 << 9, & YLINT > 8: \ for\ 15\ SM\ DQ \\ 31 << 10, & YLINT > 9: \ for\ 16\ SM\ DQ \\ THR1, & \text{otherwise} \end{cases}$$

$$DQTHR = (THR2 + (THR2 >> 1)) >> 1$$

$$TR = \begin{cases} 1, & DQMAG > DQTHR \text{ and } TD = 1 \\ 0, & \text{otherwise} \end{cases}$$

TRIGB

See § 4.2.6 for specification.

4.2.8 *Output PCM format conversion and synchronous coding adjustment*

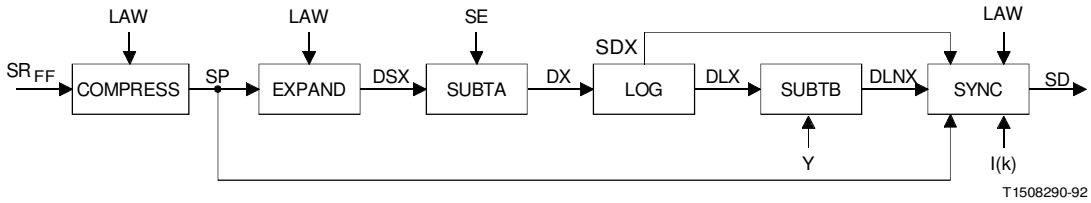

FIGURE 11/G.726

Output PCM format conversion and synchronous coding adjustment

COMPRESS (decoder only)

Inputs: SR, LAW
Output: SP
Function: Convert from uniform PCM to either A-law or μ-law PCM.

$IS = SR >> 15$

$$IM = \begin{cases} SR, & IS = 0 \\ (65536 - SR)\ \&\ 32767, & IS = 1 \end{cases}$$

| Convert two's
| complement to
| signed magnitude
|

$$IMAG = \begin{cases} IM, & LAW = 0 \\ IM >> 1 & LAW = 1 \text{ and } IS = 0 \\ (IM + 1) >> 1 & LAW = 1 \text{ and } IS = 1 \end{cases}$$

| μ-law
| A-law
|

then quantize IMAG (see note below) according to Recommendation G.711 using decision values (column 5 of Tables 1a, 1b, 2a and 2b/G.711) in the following way:

$$SP = \begin{cases} \text{character signal after even bit inversion deduced from Table 1a/G.711 (column 6),} & IS = 0 \text{ and } LAW = 1 \\ \text{character signal after even bit inversion deduced from Table 1b/G.711 (column 6),} & IS = 1 \text{ and } LAW = 1 \\ \text{character signal of Table 2a/G.711 (column 6),} & IS = 0 \text{ and } LAW = 0 \\ \text{character signal of Table 2b/G.711 (colum 6),} & IS = 1 \text{ and } LAW = 0 \end{cases}$$

Note – When IMAG is outside the range defined by the virtual decision level, SP must be set equal to the maximum PCM code word. For the purpose of clarification, examples of conversion for both A-law (after even bit inversion) and μ-law in the vicinity of the origin are given in the table below:

TABLE 15/G.726

Conversion for A-law and μ-law examples

IS	IMAG	PCM code word SP	
		A-law	μ-law
0	3	11010100	11111101
0	2	11010100	11111110
0	1	11010101	11111110
0	0	11010101	11111111
1	1	01010101	01111110
1	2	01010101	01111110
1	3	01010100	01111101

EXPAND

See § 4.2.1 for specification. Substitute SP for S as input and SLX for SL as output.

LOG

See § 4.2.2 for specification. Substitute DX for D as input, DLX for DL and DSX for DS as outputs.

SUBTA

See § 4.2.1 for specification. Substitute SLX for SL as input and DX for D as output.

SUBTB

See § 4.2.2 for specification. Substitute DLX for DL as input and DLNX for DLN as output.

SYNC (decoder only)

Inputs: I, SP, DLNX, DSX, LAW
Output: SD
Function: Re-encode output PCM sample in decoder for synchronous tandem coding.

For 40 kbit/s ADPCM:

$IS = \mathrm{I} >> 4$

$$IM = \begin{cases} I + 16, & IS = 0 \\ I \ \& \ 15, & IS = 1 \end{cases}$$

$$SD = \begin{cases} SP^{+}, & ID < IM \\ SP, & ID = IM \\ SP^{-}, & ID > IM \end{cases}$$

where

SP^{+} = the PCM code word that represents the next more positive PCM output level (when SP represents the most positive PCM output level, then SP^{+} is constrained to be SP).

and

SP^{-} = the PCM code word that represents the next more negative PCM output level (when SP represents the most negative PCM output level, then SP^{-} is constrained to be SP).

For 32 kbit/s ADPCM:

$IS = \mathrm{I} >> 3$

$$IM = \begin{cases} I + 8, & IS = 0 \\ I \ \& \ 7, & IS = 1 \end{cases}$$

ID is defined according to the following table:

TABLE 16/G.726

ID definition for 40 kbit/s ADPCM

DSX	DLNX	ID	
0	553-2047	31	
0	528- 552	30	
0	502- 527	29	
0	475- 501	28	
0	445- 474	27	
0	413- 444	26	
0	378- 412	25	
0	339- 377	24	
0	298- 338	23	
0	250- 297	22	
0	198- 249	21	
0	139- 197	20	
0	68- 138	19	
0	0- 67	18	– – | Positive portion of decision interval
0	4080-4095	18	– – | Negative portion of decision interval
0	3974-4079	17	
0	2048-3973	15	
1	2048-3973	15	
1	3974-4079	14	
1	4080-4095	13	– – | Negative portion of decision interval
1	0- 67	13	– – | Positive portion of decision interval
1	68- 138	12	
1	139- 197	11	
1	198- 249	10	
1	250- 297	9	
1	298- 338	8	
1	339- 377	7	
1	378- 412	6	
1	413- 444	5	
1	445- 474	4	
1	475- 501	3	
1	502- 527	2	
1	528- 552	1	
1	553-2047	0	

ID is defined according to the following table:

TABLE 17/G.726

ID definition for 32 kbit/s ADPCM

DSX	DLNX	ID	
0	400- 2047	15	
0	349- 399	14	
0	300- 348	13	
0	246- 299	12	
0	178- 245	11	
0	80- 177	10	
0	0- 79	9	– – | Positive portion of decision interval
0	3972-4095	9	– – | Negative portion of decision interval
0	2048-3971	7	
1	2048-3971	7	
1	3972-4095	6	– – | Negative portion of decision interval
1	0- 79	6	– – | Positive portion of decision interval
1	80- 177	5	
1	178- 245	4	
1	246- 299	3	
1	300- 348	2	
1	349- 399	1	
1	400-2047	0	

$$SD = \begin{cases} SP^{+}, & ID < IM \\ SP, & ID = IM, SP^{-}, \\ & ID > IM \end{cases}$$

where

SP^{+} = the PCM code word that represents the next more positive PCM output level (when SP represents the most positive PCM output level, then SP^{+} is constrained to be SP).

and

SP^{-} = the PCM code word that represents the next more negative PCM output level (when SP represents the most negative PCM output level, then SP^{-} is constrained to be SP).

For 24 kbit/s ADPCM:

$$IS = I >> 2$$

$$IM = \begin{cases} I + 4, & IS = 0 \\ I \;\&\; 3, & IS = 1 \end{cases}$$

ID is defined according to the following table:

TABLE 18/G.726

ID definition for 24 kbit/s ADPCM

DSX	DLNX	ID	
0	331-2047	7	
0	218- 330	6	
0	8- 217	5	
0	0- 7	3	– – | Positive portion of decision interval
0	2048-4095	3	– – | Negative portion of decision interval
1	2048-4095	3	– – | Negative portion of decision interval
1	0- 7	3	– – | Positive portion of decision interval
1	8- 217	2	
1	218- 330	1	
1	331-2047	0	

$$SD = \begin{cases} SP^{+}, & ID < IM \\ SP, & ID = IM \\ SP^{-}, & ID > IM \end{cases}$$

where

SP^{+} = the PCM code word that represents the next more positive PCM output level (when SP represents the most positive PCM output level, then SP^{+} is constrained to be SP).

and

SP^{-} = the PCM code word that represents the next more negative PCM output level (when SP represents the most negative PCM output level, then SP^{-} is constrained to be SP).

For 16 kbit/s ADPCM:

$$IS = \mathrm{I} >> 1$$

$$IM = \begin{cases} I + 2, & IS = 0 \\ I \ \& \ 1, & IS = 1 \end{cases}$$

ID is defined according to the following table:

TABLE 19/G.726

ID definition for 16 kbit/s ADPCM

DSX	DLNX	ID	
0	261-2047	3	
0	0- 260	2	– – | Positive portion of decision interval
0	2048-4095	2	– – | Negative portion of decision interval
1	2048-4095	1	– – | Negative portion of decision interval
1	0- 260	1	– – | Positive portion of decision interval
1	261-2047	0	

For the purposes of clarification, examples of re-encoding for both A-law (after even bit inversion) and μ-law in the vicinity of the origin are given in the table below:

TABLE 20/G.726

Re-encoding for A-law and μ-law: ADPCM

	A-law		μ-law	
Comparison of ID and IM	SP	SD	SP	SD
ID > IM	11010101	01010101	11111110	11111111
ID = IM	"	11010101	"	11111110
ID < IM	"	11010100	"	11111101
ID > IM	01010101	01010100	11111111	01111110
ID = IM	"	01010101	"	11111111
ID < IM	"	11010101	"	11111110
ID > IM	01010100	01010111	01111110	01111101
ID = IM	"	01010100	"	01111110
ID < IM	"	01010101	"	01111111

Note – SP (and SD) represent character signals defined according to Tables 1/G.711 and 2/G.711. See sub-block COMPRESS (§ 4.2.8) for the exact representation of SP (and SD).

APPENDIX I

(to Recommendation G.726)

Network aspects

The purpose of this Appendix is to give a broad outline of the interaction of 16, 24, 32 and 40 kbit/s ADPCM with other devices that are found in the telephony network and also the effect of specific signals found in the network. Some general guidance is also offered.

I.1 *General transmission considerations*

Both 24 and 16 kbit/s codings are intended for use with DCME overload channels. It is recommended that 32 kbit/s and 16 kbit/s or 24 kbit/s coding be alternated rapidly such that at least 3.5 to 3.7 bits/sample are used on average. The rate of alternation is for further study. The method of alternation is beyond the scope of this Recommendation. The effect on speech quality of this alternation is not expected to be significant. The use of 24 or 16 kbit/s coding for data transmission is not recommended.

The 40 kbit/s coding is intended for use with DCME and packet circuit multiplication equipment (PCME) data modem channels, especially for modem operation at speeds of 7200, 9600 and 12 000 bit/s.

Consideration will have to be given to appropriate corrective action with, for example, the use of bit stealing techniques for the provision of a limited speed signalling facility. Otherwise, serious performance degradation will occur.

Conversely a 64 kbit/s channel which is conveyed by an ADPCM channel (or channels) will not exhibit bit integrity.

I.2 *Interaction with other processes*

The synchronous coding adjustment is described in §§ 1.2 and 3.7 of this Recommendation. The favourable operation of this adjustment is dependent on the signals on the ADPCM path and on the intermediate 64 kbit/s path both being uncorrupted by other digital processes. For example, the use of digital pads, A-law to μ-law converters, echo cancellers or digital speech interpolation (DSI) at these intermediate points will inhibit the correct functioning of this adjustment. However, the performance will still be better than that achieved when an asynchronous connection is employed.

The use of an ADPCM link to interconnect 64 kbit/s A-law PCM signals and 64 kbit/s μ-law signals has been found to be satisfactory for speech even though this will inhibit the correct operation of the synchronous coding adjustment between the ADPCM link so used and the subsequent ADPCM link.

The interactions between ADPCM and processes such as DSI and echo cancellation (e.g. quantization noise in the echo path) are for further study.

The effect of large d.c. offsets (arising from PCM encoders) on the performance of ADPCM for low level signals is for further study.

I.3 *Interaction with coding laws other than PCM*

Interconnection with coding laws other than PCM is not the subject of the Recommendation and analogue interconnections may need to be employed.

It follows that great care must be exercised when interconnection is made to coding laws which are not the subject of CCITT Recommendations.

I.4 *Encoder/decoder synchronization*

The encoder and its respective decoder must always operate at the same bit rate (i.e 16, 24, 32 or 40 kbit/s), or otherwise severe mistracking may occur.

I.5 *Synchronous coding adjustment*

The synchronous coding adjustment will work correctly when an ADPCM encoder/decoder pair is connected by a bit-transparent 64 kbit/s PCM path to another encoder/decoder pair operating at the same rate. When two encoder/decoder pairs are operating at different rates, the synchronous tandeming property is not guaranteed to be established.

I.6 *Speech performance*

Under error free transmission conditions the perceived quality of speech over 32 kbit/s ADPCM links is only slightly lower than that over 64 kbit/s PCM links. This will only be significant when numbers of such links are used in tandem and not when single links are used. Hence the numbers of such 32 kbit/s ADPCM links must be controlled on an international connection. With transmission error ratios higher than $1 \cdot 10^{-4}$ the perceived quality of speech over 32 kbit/s ADPCM links is better than that over 64 kbit/s PCM links. Precise limits for the international portion of the connection and the national extensions may be found in Recommendation G.113. Preliminary tests indicate that for voice, the 40 kbit/s ADPCM coding performs approximately as well as 64 kbit/s PCM according to Recommendation G.711.

I.7 *Voice frequency telegraph performance*

Twenty-four-channel voice frequency telegraph of Recommendation R.35 cannot be satisfactorily conveyed over 32 kbit/s ADPCM links and it is, therefore, desirable to implement routing rules to avoid this combination.

I.8 *Data performance*

Voiceband data performance up to 2400 bit/s using, for example modems conforming to Recommendation V.21, V.22 *bis*, V.23 and V.26 *ter*, will not be subject to significant degradation over 32 kbit/s ADPCM links provided the numbers of such links do not exceed the limits of Recommendation G.113.

Voiceband data performance at 4800 bit/s using, for example modems conforming to Recommendation V.27 *bis*, can be accommodated with 32 kbit/s ADPCM but will be subject to additional degradations over and above that expected from standard 64 kbit/s PCM links. More care will need to be exercised in using such a service.

Voiceband data at speeds up to 12 000 bit/s can be accommodated by 40 kbit/s ADPCM. The performance of V.33 modems operating at 14 400 bit/s over 40 kbit/s ADPCM is for further study.

I.9 *Dual tone multi-frequency (DTMF) signalling*

No major difficulties are likely to be experienced with DTMF signalling conveyed over 32 kbit/s or 40 kbit/s ADPCM links. The use of DTMF for end-to-end signalling is limited by the number of links in tandem. DTMF performance for 16 kbit/s or 24 kbit/s ADPCM is for further study.

I.10 *Facsimile*

No degradation is to be expected when using 40 kbit/s ADPCM with Group 2 or Group 3 facsimile apparatus according to Recommendations T.3 or T.4 at rates up to 12 000 bit/s. Performance of Group 3 facsimile when using 40 kbit/s ADPCM at 14 400 bit/s is for further study. No serious degradation is to be expected when using 32 kbit/s ADPCM with Group 2 facsimile apparatus according to Recommendations T.3 or T.4 at rates up to 12 000 bit/s.

APPENDIX II

(to Recommendation G.726)

Digital test sequences for the verification of the algorithms in Recommendation G.726

This Appendix gives information on the digital test sequences which have been chosen to verify implementations of the algorithms in Rec. G.726. Copies of the sequences on flexible diskettes together with a detailed description can be ordered from the ITU sales services (Please refer to collective letter No. 11/XV, 1991).

II.1 *Purpose of digital test sequences*

Digital sequences are used to verify the conformance of an implementation to a digital transcoding algorithm. The sequences are chosen to exercise the major arithmetic components and thus give a reasonable level of confidence of the compliance of an implementation with this Recommendation. Note that with a limited number of test sequences it is not possible to demonstrate 100% coverage of all states of the implementation. The more general issues involved in testing such algorithms are the subject of active research in the areas of VLSI testing and protocol conformance testing.

II.2 *Diskette interface and format*

Copies of the digital test sequences are available from the ITU on four 5 ¼″ diskettes. The diskettes were created under MS-DOS operating system (version 3.2 or newer), and 1.2 Mbyte high-density doubled-sided 96 tracks per inch 5¼ MS-DOS format.