



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

# UIT-T

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

# G.728

**Apéndice I**  
**Soporte lógico**  
(07/95)

SERIE G: SISTEMAS Y MEDIOS DE TRANSMISIÓN,  
SISTEMAS Y REDES DIGITALES

Sistemas de transmisión digital – Equipos terminales –  
Codificación de señales analógicas mediante métodos  
diferentes de la MIC

---

**Programas y secuencias de prueba para  
verificación de la implementación del algoritmo  
del codificador de voz utilizando predicción  
lineal con excitación por código de bajo retardo  
a 16 kbit/s de la Recomendación G.728**

Recomendación UIT-T G.728 – Apéndice I

(Anteriormente Recomendación del CCITT)

---

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

## NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

## PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 1998

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

**PROGRAMAS Y SECUENCIAS DE PRUEBA PARA VERIFICACIÓN  
DE LA IMPLEMENTACIÓN DEL ALGORITMO DEL CODIFICADOR  
DE VOZ UTILIZANDO PREDICCIÓN LINEAL CON EXCITACIÓN  
POR CÓDIGO DE BAJO RETARDO A 16 kbit/s  
DE LA RECOMENDACIÓN G.728**

## **1 Consideraciones generales**

Este documento describe la secuencia de pruebas digitales y los soportes lógicos de medición que ha de utilizarse para verificar la realización de la Recomendación G.728. Se incluyen disposiciones para realizaciones en coma flotante, basadas en el texto principal de la Recomendación G.728, y en coma fija exacta a nivel de bit, basadas en el anexo G/G.728.

### **1.1 Principios de verificación para realizaciones en coma flotante**

El texto principal de la especificación del algoritmo LD-CELP se formula de manera no exacta a nivel de bit para permitir la realización simple en diferentes tipos de soporte físico. Esto implica que el procedimiento de verificación no puede suponer que existe una realización de referencia exactamente igual a la aplicación que está probándose. Por consiguiente, se requieren mediciones objetivas para establecer el grado de desviación entre la realización probada y la referencia. Si se halla que esta desviación medida es suficientemente pequeña, se supone que la realización sometida a prueba puede interfuncionar con cualquier otra realización que pase la prueba. Dado que ninguna prueba de duración finita puede probar todos los aspectos de una realización, nunca puede garantizarse de modo absoluto que una realización es correcta. No obstante, el procedimiento de prueba descrito pone en juego todas las partes importantes del algoritmo LD-CELP, y debería constituir una herramienta valiosa para los encargados de realizarlo.

Los procedimientos de verificación de coma flotante descritos en este documento han sido concebidos teniendo en cuenta realizaciones de coma flotante de 32 bits. Si bien podrían aplicarse a cualquier realización LD-CELP, se necesitará probablemente el formato de coma flotante de 32 bits para cumplir los requisitos de las pruebas.

### **1.2 Principios de verificación para realizaciones en coma fija**

El anexo G/G.728 describe el algoritmo LD-CELP de coma fija de una forma exacta a nivel de bit. Ello supone que cuando dos realizaciones de codificador o decodificador arrancan desde sus estados iniciales con señales de entrada idénticas, todas las variables de estado deben ser exactamente idénticas en instantes equivalentes a lo largo del procesamiento de las señales de entrada. En consecuencia, pueden utilizarse secuencias de prueba de entrada en coma flotantes dando lugar a secuencias de salida exactas a nivel de bit.

Se ha utilizado como señal de entrada de prueba adicional un segmento breve de señal vocal de entrada. También se proporcionan todas las variables de estado interno asociadas durante el procesamiento de esta entrada. De esa forma, el encargado de la aplicación puede verificar que todo el procesamiento en la realización se adapta con precisión al procesamiento descrito en la especificación. Para considerar que cumple exactamente lo dispuesto en el anexo G/G.728, una implementación debe adaptarse exactamente a las salidas prescritas para las señales de entrada de prueba de coma flotante y a todas las variables de estado interno para el segmento breve de señal vocal.

Al igual que en el caso del procesamiento de verificación de la realización de coma flotante, como no hay ninguna prueba de longitud finita capaz de verificar todos y cada uno de los aspectos de una

realización, nunca puede garantizarse al 100% que la realización es correcta. Sin embargo, el procedimiento de prueba descrito verifica todas las partes principales del algoritmo LD-CELP y constituye una herramienta muy útil para el implementador.

## 2 Configuraciones de prueba

En esta cláusula se describe cómo deben combinarse las diferentes secuencias de prueba y los programas de medición para llevar a cabo las prueba de verificación. El procedimiento se basa en pruebas realizadas en las interfaces SU e ICHAN del codificador sometido a prueba, e ICHAN y SPF del decodificador sometido a prueba. Las señales SU y SPF se representan con precisión de coma fijo de 16 bits, tal como se describe en la subcláusula 4.2. Debería darse la posibilidad de desactivar el posfiltro adaptativo en el realización del decodificador sometido a prueba. Toda secuencia de pruebas debería empezar con la realización sometida a prueba en el estado inicial definido en la Recomendación G.728. Se necesitan tres programas de mediciones, CWCOMP, SNR y WSNR, para llevar a cabo las evaluaciones de las secuencias de salida de las pruebas. Estos programas se describen más detalladamente en la cláusula 3. En las subcláusulas 2.1 a 2.6 figuran las descripciones de las diferentes configuraciones de prueba que han de utilizarse.

### 2.1 Prueba del codificador

El funcionamiento básico del codificador es probado mediante la configuración representada en la figura 1. Se aplica una secuencia de prueba de señales de entrada, IN, al codificador sometido a prueba. Las palabras clave de salida se comparan directamente a las palabras clave de referencia, INCW o INCW\*G, utilizando el programa CWCOMP.

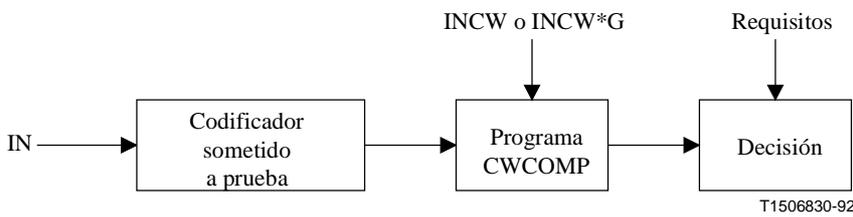


Figura 1 – Configuración de prueba N.º 1 – Prueba del codificador

### 2.2 Prueba del decodificador

El funcionamiento básico del decodificador es probado mediante la configuración representada en la figura 2. Se aplica una secuencia de prueba de palabras clave, CW, al decodificador sometido a prueba con el posfiltro adaptativo desactivado. La señal de salida es entonces comparada a la señal de salida de referencia, OUTA, con el programa SNR.

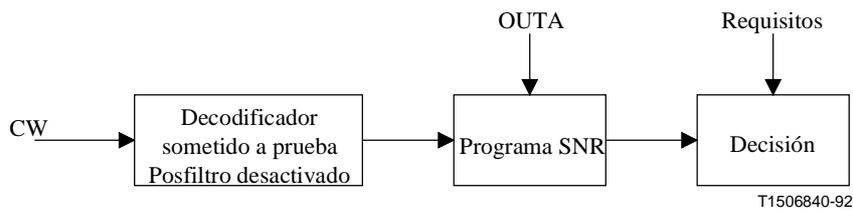


Figura 2 – Configuración de prueba N.º 2 – Prueba del decodificador

### 2.3 Prueba del filtro de ponderación perceptual

El filtro de ponderación perceptual del codificador es probado con la configuración ilustrada en la figura 3. Se hace pasar una secuencia de prueba de señales de entrada, IN5, a través del codificador sometido a prueba, y la calidad de las palabras clave de salida se mide con el programa WSNR. Este programa necesita también la secuencia de entrada a fin de calcular la medida de distancia correcta.

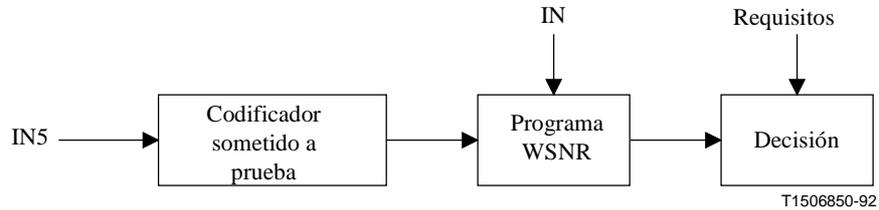


Figura 3 – Configuración de prueba N.º 3 – Prueba del codificador

### 2.4 Prueba del posfiltro

El posfiltro adaptativo del decodificador es probado con la configuración ilustrada en la figura 4. Se aplica una secuencia de prueba de palabras clave, CW, al decodificador sometido a prueba con el posfiltro adaptativo activado. La señal de salida es entonces comparada a la señal de salida de referencia, OUTB, con el programa SNR.

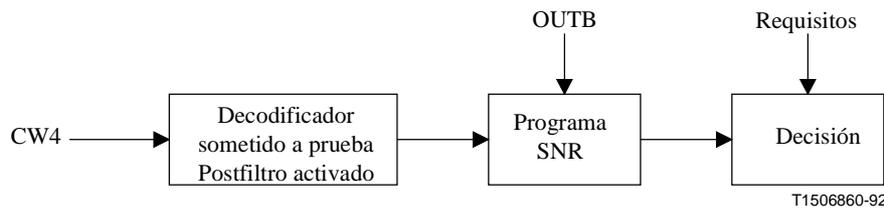


Figura 4 – Configuración de prueba N.º 4 – Prueba del decodificador

### 2.5 Prueba del decodificador de coma fija

El funcionamiento básico del decodificador del anexo G es probado mediante la configuración representada en la figura 5. Se aplica una secuencia de palabras clave, CW, al decodificador sometido a prueba con el posfiltro adaptativo desactivado. La señal de salida se compara a continuación con la señal de salida de referencia, OUTA\*G, mediante un programa *diff* en Unix ® o un programa FC MS-DOS ®. No debe apreciarse ninguna diferencia.

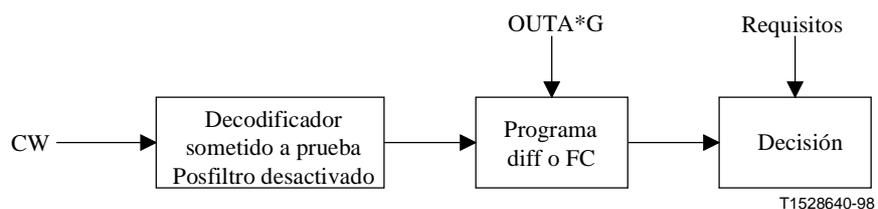
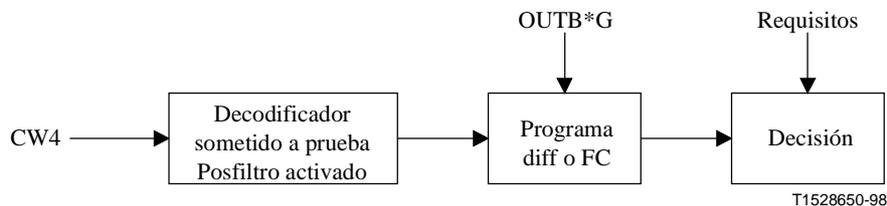


Figura 5 – Configuración de prueba N.º 5 – Prueba del decodificador de coma fija

## 2.6 Prueba del posfiltro de coma fija

El posfiltro adaptativo del decodificador de coma fija es probado mediante la configuración presentada en la figura 6. Se aplica una secuencia de prueba de palabras clave, CW, al decodificador sometido a prueba con el posfiltro adaptativo activado. La señal de salida se compara a continuación con la señal de salida de referencia, OUTB\*G, mediante un programa *diff* en Unix <sup>®1</sup> o un programa FC en MS-DOS <sup>®1</sup>. No debe apreciarse ninguna diferencia.



**Figura 6 – Configuración de prueba N.º 6 – Prueba del posfiltro de coma fija**

## 2.7 Variables de estado interno de coma fija

Para realizaciones de coma fija, se aplica como señal de entrada de prueba adicional un segmento breve de señal vocal de entrada. También se proporcionan todas las variables de estado interno asociadas necesarias para procesar esta entrada (tanto en el codificador como en el decodificador). Esta señal la utilizará el encargado de la aplicación para verificar que todo el procesamiento en la realización se adapta exactamente al procesamiento descrito en la especificación. Para considerar que cumple exactamente lo dispuesto en el anexo G, una implementación debe adaptarse exactamente a las señales prescritas para las señales de entrada de prueba de coma flotante, indicadas en las subcláusulas 2.1, 2.5 y 2.6 anteriores, y a todas las variables de estado interno para el segmento breve de la señal vocal.

## 3 Programas de verificación

En esta subcláusula se describen los programas CWCOMP, SNR y WSNR, mencionados en el punto relativo a la configuración de las pruebas, así como el programa LDCDEC, proporcionado en calidad de herramienta de depuración para los encargados de las aplicaciones.

El soporte lógico de verificación está escrito en Fortran, lo más cerca posible de la norma ANSI Fortran 77. Se utiliza en gran medida la resolución de doble precisión en coma flotante a fin de reducir al mínimo los errores numéricos en los módulos LD-CELP de referencia. Los programas fueron compilados con un compilador Fortran disponible en el mercado, a fin de elaborar versiones ejecutables para los PC basados en 386/87. El fichero READ.ME describe cómo crear programas ejecutables en otros computadores.

### 3.1 CWCOMP

El programa CWCOMP es una herramienta sencilla para comparar el contenido de dos ficheros de palabras de código. El usuario debe suministrar dos nombres de ficheros de palabras clave, a saber, la salida del codificador de referencia (nombre de fichero que figura en la última columna del cuadro 1)

<sup>1</sup> ® Unix es la marca comercial registrada de Unix Systems Laboratories y MS-DOS es la marca comercial registrada de Microsoft Corporation.

y la salida del codificador sometido a prueba. El programa compara las palabras clave de estos ficheros y escribe el resultado de la comparación en el terminal. El requisito para la configuración de prueba 1 es que no debe haber ninguna diferencia.

### 3.2 SNR

El programa SNR (signal-to-noise ratio, relación señal/ruido) realiza una medición de la relación señal/ruido entre dos ficheros de señales. El primero es un fichero de referencia proporcionado por el programa decodificador de referencia, y el segundo es el fichero de salida del decodificador sometido a prueba. Una relación SNR global, a saber GLOB, es calculada como relación señal/ruido de la totalidad del fichero. Una relación SNR parcial, a saber SEG256, es calculada como media de la relación señal/ruido de todos los segmentos de 256 muestras cuya potencia de señal de referencia supera determinado umbral. Se hallan las relaciones SNR de segmento mínimas para segmentos de longitud 256, 128, 64, 32, 16, 8 y 4 con potencia por encima del mismo umbral.

Para ejecutar el programa SNR, el usuario debe introducir los nombres de dos ficheros de entrada. El primero es el fichero de salida del decodificador de referencia descrito en la última columna del cuadro 4. El segundo es el fichero de salida decodificada producido por el decodificador sometido a prueba. Después de procesar los ficheros, el programa envía las diferentes SNR al terminal. Los valores necesarios para las configuraciones de prueba 2 y 4 se dan con respecto a estos números SNR.

### 3.3 WSNR

El algoritmo WSNR se basa en la realización de un decodificador de referencia y una medición de distancia para calcular la distorsión ponderada perceptual media de una secuencia de palabras clave. Se calcula una relación logarítmica señal/distorsión para cada vector de señal de 5 muestras, y las relaciones se promedian teniendo en cuenta todos los vectores de señales cuya energía está por encima de determinado umbral.

Para ejecutar el programa WSNR, el usuario tiene que introducir los nombres de dos ficheros de entrada. El primero es el fichero de señales de entrada del decodificador (primera columna del cuadro 1) y el segundo es el fichero de palabras clave de salida del decodificador. Después de procesar la secuencia, el programa WSNR escribe el valor de salida WSNR en el terminal. Los valores necesarios para la configuración de prueba 3 se dan con respecto a este número WSNR.

### 3.4 LDCDEC

Además de los tres programas de medición, se incluye también un programa de demostración del decodificador de referencia, a saber, el programa LDCDEC. Este programa se basa en la misma subrutina de decodificador que el WSNR, y podría modificarse para comprobar variables en el decodificador a efectos de depuración. El usuario debe indicar el fichero de palabras clave de entrada, el fichero de señales de salida, y si ha de incluirse el posfiltro adaptativo o no.

### 3.5 *diff* o FC

Además del soporte lógico distribuido con los disquetes, se supone que el encargado de la aplicación cuenta con el sistema operativo Unix ® o MS-DOS ®. Las instrucciones *diff* y FC comparan dos ficheros e indican al usuario si son el mismo o son distintos. Para comprobar ficheros binarios en DOS, la instrucción correcta es FC /B FILE1 FILE2. Para comparar dos ficheros en Unix la instrucción es diff file1 file2.

## 4 Secuencias de prueba

A continuación se da una descripción de las secuencias de prueba que deben aplicarse. La descripción incluye los requisitos específicos de cada secuencia.

### 4.1 Convenios

Las secuencias de prueba se numeran secuencialmente, con un prefijo que identifica el tipo de señal:

IN	Señal de entrada del decodificador ( <i>encoder input signal</i> )
INCW	Palabra de código de salida del codificador en coma flotante ( <i>floating point encoder output codewords</i> )
INCW*G	Palabra de código de salida del codificador en coma fija ( <i>fixed point encoder output codewords</i> )
CW	Palabra de código de entrada del decodificador ( <i>decoder input codewords</i> )
OUTA	Señal de salida del decodificador sin posfiltro ( <i>decoder output signal without postfilter</i> )
OUTA*G	Señal de salida del decodificador en coma fija sin posfiltro ( <i>fixed point decoder output signal without postfilter</i> )
OUTB	Señal de salida del decodificador con posfiltro ( <i>decoder output signal with postfilter</i> )
OUTB*G	Señal de salida del decodificador en coma fija con posfiltro ( <i>fixed point decoder output signal with postfilter</i> )

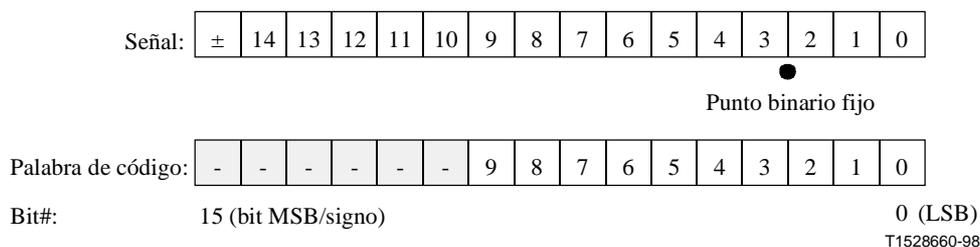
Todos los ficheros de secuencias de prueba tienen la extensión \*.BIN.

### 4.2 Formatos de los ficheros

Los ficheros de señales, de conformidad con los interfaces SU y SPF del LD-CELP (prefijo de fichero IN, OUTA y OUTB) están todos en formato binario de 16 bits con complemento a 2 y deben interpretarse con un punto binario fijo entre el bit #2 y el bit #3, como se muestra en la figura 5. Obsérvese que deben usarse en, su totalidad, los 16 bits disponibles para alcanzar el máximo de presión en las mediciones de prueba.

Los ficheros de palabras clave (señal ICHAN del LD-CELP, prefijo de fichero CW o INCW) están almacenados en el mismo formato binario de 16 bits que los ficheros de señales. Los 10 bits menos significativos de cada palabra de 16 bits representan la palabra de código de 10 bits, tal como se indica en la figura 7. Los otros bits (#12-#15) se ponen a cero.

Tanto los ficheros de señales como los de palabras de código están almacenados en el formato de almacenamiento de palabras en que se coloca primero el byte de menor peso y que es usual en los computadores IBM/DOS y VAX/VMS. Para la utilización en otras máquinas, tales como la mayor parte de las UNIX, puede tener que modificarse este orden mediante una operación de trueque de bytes.



**Figura 7 – Formato binario de los ficheros de señales y palabras de código**

### 4.3 Secuencias de prueba y requisitos

Los cuadros de esta subcláusula describen el conjunto completo de pruebas que deben realizarse para verificar que una aplicación en coma flotante de LD-CELP es conforme a la especificación e interfuerona con otras aplicaciones correctas. El cuadro 1 es un resumen de las secuencias de prueba del codificador. Los requisitos correspondientes se expresan en los cuadros 2 y 3. Los cuadros 4, 5 y 6 contienen el resumen y los requisitos de las secuencias de prueba del decodificador.

**Cuadro 1 – Pruebas del codificador**

Señal de entrada	Longitud vectores	Descripción de la prueba	Configuración de la prueba	Señal de salida
IN1	1 536	Probar que las 1024 palabras de código posible están todas adecuadamente realizadas	1	INCW1, INCW1G
IN2	1 536	Comprobar la gama dinámica de la función de autocorrelación de ganancia logarítmica	1	INCW2, INCW2G
IN3	1 024	Comprobar la gama dinámica de la función de autocorrelación de las señales decodificadas	1	INCW3, INCW3G
IN4	10 240	Barrido de frecuencia en la gama típica de timbre de señal vocal	1	INCW4, INCW4G
IN5	84 480	Señal vocal real con diferentes niveles de entrada y micrófonos	3	– INCW5G
IN6	256	Probar los limitadores del codificador	1	INCW6, INCW6G

**Cuadro 2 – Requisitos de prueba del codificador en coma flotante**

Señal de entrada	Señal de salida	Requisitos
IN1	INCW1	0 palabras de código diferentes detectadas por CWCOMP
IN2	INCW2	0 palabras de código diferentes detectadas por CWCOMP
IN3	INCW3	0 palabras de código diferentes detectadas por CWCOMP
IN4	INCW4	0 palabras de código diferentes detectadas por CWCOMP
IN5	–	WSNR > 20,55 dB
IN6	INCW6	0 palabras de código diferentes detectadas por CWCOMP

**Cuadro 3 – Requisitos de prueba del codificador en coma fija**

Señal de entrada	Señal de salida	Requisitos
IN1	INCW1G	0 palabras de código diferentes detectadas para CWCOMP
IN2	INCW2G	0 palabras de código diferentes detectadas para CWCOMP
IN3	INCW3G	0 palabras de código diferentes detectadas para CWCOMP
IN4	INCW4G	0 palabras de código diferentes detectadas para CWCOMP
IN5	INCW5G	0 palabras de código diferentes detectadas para CWCOMP
IN6	INCW6G	0 palabras de código diferentes detectadas para CWCOMP

**Cuadro 4 – Pruebas del decodificador**

Señal de entrada	Longitud vectores	Descripción de la prueba	Configuración de la prueba	Señal de salida
CW1	1 536	Probar que las 1024 palabras de código posible están todas adecuadamente realizadas	2, 5	OUTA1, OUTA1G
CW2	1 792	Comprobar la gama dinámica de la función de autocorrelación de ganancia logarítmica	2, 5	OUTA2, OUTA2G
CW3	1 280	Comprobar la gama dinámica de la función de autocorrelación de las señales decodificadas	2, 5	OUTA3, OUTA3G
CW4	10 240	Probar el decodificador con barrido de frecuencia en la gama típica de timbre de señal vocal	2, 5	OUTA4, OUTA4G
CW4	10 240	Probar el posfiltro con barrido de frecuencia en la gama típica de timbre de señal vocal	4, 6	OUTB4, OUTB4G
CW5	84 480	Señal vocal real con diferentes niveles de entrada y micrófonos	2, 5	OUTA5, OUTA5G
CW6	256	Probar los limitadores del decodificador	2, 5	OUTA6, OUTA6G

**Cuadro 5 – Requisitos de prueba del decodificador en coma flotante**

Nombre del fichero de salida	Requisitos (valores mínimos de SNR, en dB)								
	SEG256	GLOB	MIN256	MIN128	MIN64	MIN32	MIN16	MIN8	MIN4
OUTA1	75,00	74,00	68,00	68,00	67,00	64,00	55,00	50,00	41,00
OUTA2	94,00	85,00	67,00	58,00	55,00	50,00	48,00	44,00	41,00
OUTA3	79,00	76,00	70,00	28,00	29,00	31,00	37,00	29,00	26,00
OUTA4	60,00	58,00	51,00	51,00	49,00	46,00	40,00	35,00	28,00
OUTB4	59,00	57,00	50,00	50,00	49,00	46,00	40,00	34,00	26,00
OUTA5	59,00	61,00	41,00	39,00	39,00	34,00	35,00	30,00	26,00
OUTA6	69,00	67,00	66,00	64,00	63,00	63,00	62,00	61,00	60,00

### Requisitos de prueba del decodificador en coma fija

No hay diferencias entre el vector de prueba de salida, OUTA\*G o OUTB4G y la salida real para cualquier vector de prueba de entrada, CW\*, medido por *diff* o FC o un programa de comparación de ficheros equivalente. Además, para poder considerar que se adapta exactamente al anexo G/G.728, una realización debe seguir con precisión todas las variables de estado interno para el segmento breve de señal vocal.

## 5 Distribución de las herramientas de verificación

Se incluye un fichero READ.ME en el disquete #1, que describe el contenido de cada fichero y los procedimientos necesarios para compilar y encadenar los programas. Se emplean extensiones para distinguir los diferentes tipos de ficheros. Los ficheros \*.FOR son códigos fuente de programas Fortran, los ficheros \*.EXE son programas ejecutables con procesadores 386/87, y los ficheros \*.BIN son ficheros de secuencias de pruebas binarias. Se indica el contenido de cada disquete en los cuadros 6, 7, 8 y 9.

**Cuadro 6 – Lista de distribución, disco #1**

<b>Disquete</b>	<b>Nombre del fichero</b>	<b>Número de bytes</b>
Disquete #1	READ.ME	10 430
	CWCOMP.FOR	2 642
Tamaño total: 1 289 859 bytes	CWCOMP.EXE	25 153
	SNR.FOR	5 536
	SNR.EXE	36 524
	WSNR.FOR	3 554
	WSNR.EXE	103 892
	LDCDEC.FOR	3 016
	LDCDEC.EXE	101 080
	LDCSUB.FOR	37 932
	FILSUB.FOR	1 740
	DSTRUCT.FOR	2 968
	IN1.BIN	15 360
	IN2.BIN	15 360
	IN3.BIN	10 240
	IN5.BIN	844 800
	IN6.BIN	2 560
	INCW1.BIN	3 072
	INCW2.BIN	3 072
	INCW3.BIN	2 048
	INCW6.BIN	512
	CW1.BIN	3 072
	CW2.BIN	3 584
	CW3.BIN	2 560
	CW6.BIN	512
	OUTA1.BIN	15 360
	OUTA2.BIN	17 920
	OUTA3.BIN	12 800
	OUTA6.BIN	2 560

**Cuadro 7 – Lista de distribución, disco #2**

<b>Disquete</b>	<b>Nombre del fichero</b>	<b>Número de bytes</b>
Disquete #2	IN4.BIN	102 400
	INCW4.BIN	20 480
Tamaño total: 1 361 920 bytes	CW4.BIN	20 480
	CW5.BIN	168 960
	OUTA4.BIN	102 400
	OUTB4.BIN	102 400
	OUTA5.BIN	844 800

**Cuadro 8 – Lista de distribución, disco #3**

Disquete	Nombre del fichero	Número de bytes
Disquete #3  Tamaño total: 1 297 280 bytes	INCW1G.BIN	3 072
	INCW2G.BIN	3 072
	INCW3G.BIN	2 048
	INCW4G.BIN	20 480
	INCW5G.BIN	168 960
	INCW6G.BIN	512
	OUTA1G.BIN	15 360
	OUTA2G.BIN	17 920
	OUTA3G.BIN	12 800
	OUTA4G.BIN	102 400
	OUTB4G.BIN	102 400
	OUTA5G.BIN	844 800
	OUTA6G.BIN	2 560
	READ.ME	896

**Disquete #4 – Variables de estado interno**

La especificación en coma fija de la Recomendación G.728, en su anexo G/G.728, es exacta a nivel de bit. Como señal de entrada de prueba se utiliza un segmento breve de señal vocal a fin de comparar las representaciones internas de todas las variables de estado entre dos realizaciones distintas. Todos los vectores de salida deben coincidir. Para evitar cualquier confusión todas estas salidas se almacenan en ASCII en el disco #4.

**Cuadro 9 – Lista de distribución, disco #4**

Tamaño	Nombre del fichero	Observaciones
36 100	a.q14	a(2) a a(51) en Q14. "-" significa que no hay actualización para dicho vector.
7 700	ap.q14	ap() en Q14
8 400	apf.bf	apf() como salida intermedia del bloque 50, a continuación formato Q
7 700	apf.q13	el apf() final en Q13 (convertido a partir de apf.bf)
36 900	atmp.bf	atmp(2) a atmp(51), a continuación formato IAQ Q. (salida del bloque 50)
7 700	awp.q14	awp(2) a awp(11) en Q14. "-" significa que no hay actualización para dicho vector.
7 700	awz.q14	awz(2) a awz(11) en Q14. "-" significa que no hay actualización para dicho vector.
8 400	awztmp.bf	awztmp(2) a awztmp(11) en Q13, Q14 o Q15, seguido por el formato Q en la última columna (salida del bloque 37)
7 700	az.q14	az() en Q14
1 200	b.q16	b en Q16 (coeficiente del posfiltro a largo plazo calculado en el bloque 84)
14 400	d.q1	el vector más reciente del conjunto d() en Q1
4 200	dec.q1	dec(21:25) en Q1 (nuevo dominio residual LPC diezmado para la trama actual)
15 600	et.bf	et() en la coma flotante de bloque; 5 mantisas y nlset en cada línea

**Cuadro 9 – Lista de distribución, disco #4 (continuación)**

<b>Tamaño</b>	<b>Nombre del fichero</b>	<b>Observaciones</b>
3 600	gain.sf	ganancia lineal utilizada para escalar el vector código (mantisa, a continuación nlsgain)
4 000	gaininv.sf	1/GAIN utilizado para formalizar el vector objeto (mantisa, a continuación NLS)
1 400	gl.q14	gl en Q14
1 400	glb.q14	glb en Q14
7 700	gp.q14	gp(2) a gp(11) en Q14. "-" significa que no hay actualización para dicho vector
8 400	gptmp.bf	gptmp(2) a gptmp(11), a continuación formato gptmp Q. (salida del bloque 44)
2 800	gstate.q9	gstate(1) en Q9. (Los otros 9 gstate() están en las líneas precedentes)
3 500	gtmp.q9	gtmp() en Q9. Obsérvese que el primer vector gtmp() tiene tres -16384.
4 200	h.q13	vector h() en Q13. "-" significa que no hay actualización para dicho vector.
2 000	ichan.q0	índice de canal de salida del codificador "ichan" (uno por línea)
14 400	input.q3	vector de entrada PCM lineal de 16 bits (Q3 fijo, un vector en línea)
2 400	isig.q0	el índice de forma "is" seguido por el índice de ganancia "ig" en cada línea
1 400	kp.q0	el periodo de tono kp en Q0
2 800	loggain.q9	ganancia logarítmica antes de conversión a ganancia lineal (entrada del bloque 48)
14 800	lpfiir.q1	los 20 elementos de lpfiir() correspondientes a la trama en curso
14 400	output.q3	vector de salida (con posfiltro) del decodificador en PCM lineal de 16 bits
14 400	pn.q7	pn() en Q7 (salida del bloque 13)
1 200	ptap.q14	ptap en Q14 (salida del bloque 83)
8 400	r_b36.bf	r(1) a r(11) en la salida del bloque 36
8 400	r_b43.bf	r(1) a r(11) en la salida del bloque 43
1 400	rc1.q15	rc1 del bloque 50 en Q15 (el utilizado para obtener tiltz)
5 821	readme	describe el contenido del disco #4
37 600	rexp.bf	rexp(1) a rexp(51), a continuación nlsrexp. (salida del bloque 49)
9 200	rexplg.bf	rexplg(1) a rexplg(11), a continuación nlsrexplg. (salida del bloque 43)
9 200	rexpw.bf	rexpw(1) a rexpw(11), a continuación nlsrexpw. (salida del bloque 36)
36 900	rtmp.bf	rtmp(1) a rtmp(51) en la salida del bloque 49
14 400	s.q2	vector s() de entrada tras convertir input.q3 a Q2 con redondeo
3 600	scale.sf	scale en coma flotante escalar (salida del bloque 75)
14 400	scalefil.q14	scalefil en Q14 (salida del bloque 76)
14 400	sst.q0	Q0 sst(-4:0) tras el desplazamiento de memoria intermedia SST() (es decir sst(1:5) >> 2)
14 400	sst.q2	sst(1:5) en Q2
14 400	st.bf	st() en el formato de coma flotante del bloque
15 600	statelc.sbf	los 5 elementos más recientes de statelc() y nlsstate(10) para el vector vigente
15 600	stmp.q2	stmp() en Q2. Su contenido va hasta el vector 2 de la trama vigente.
14 800	stpfir.q2	stpfir(1:5) tras el posfiltrado del vector vigente

**Cuadro 9 – Lista de distribución, disco #4 (fin)**

<b>Tamaño</b>	<b>Nombre del fichero</b>	<b>Observaciones</b>
14 400	stpfir.q2	stpfir(1:5) tras el posfiltrado del vector vigente
14 400	sttmp.sbf	sttmp(), 20 mantisas seguidas de 4 exponentes (nlssttmp()).
17 700	sw.q2	sw() en Q2 (salida del bloque 4)
3 200	sumfil.q2	sumfil en Q2 a la salida del bloque 74 (AA1 en pseudocódigo)
3 200	sumunfil.q2	sumunfil en Q2 a la salida del bloque 73 (AA0 en pseudocódigo)
14 400	target.q2	vector objetivo sin normalizar en Q2 (salida del bloque 11)
14 400	targetn.bf	vector objetivo de ganancia normalizada en la coma flotante del bloque
15 600	temp_b72.q2	temp() a la salida del bloque 72, en Q2
14 400	tiltz.q14	tiltz en Q14.
1 400	wiir.q2	los elementos 5 más recientes de wiir() en Q2 tras el filtrado de ponderación
14 400	y2.q5	conjunto y2() en Q5. "-" significa que no hay actualización para dicho vector.
78 700	zir.q2	vector zir() en Q2
14 400	zirwfir.q2	los elementos 5 más recientes de zirwfir() tras la actualización de memoria del bloque 10
14 400	zirwiir.q2	los elementos 5 más recientes de zirwiir() tras la actualización de memoria del bloque 10



## SOPORTES LÓGICOS DEL UIT-T

G. 191 (11.96)	Biblioteca de herramientas de soporte lógico 96 (STL-96) y Manual STL-96 (únicamente en inglés)
G.722 apéndice II (03.87)	Secuencias de pruebas digitales para la verificación del códec de 7 kHz MICDA-SB a 64 kbit/s de la Recomendación G.722
G.723.1 anexo A (11.96)	Código de referencia C, señales de prueba y secuencias de prueba para el codificador de voz de doble velocidad a 5,3 y 6,3 kbit/s en coma fija, versión 5.1
G.723.1 anexo B (11.96)	Código de referencia C y señales de prueba para el codificador de voz de doble velocidad a 5,3 y 6,3 kbit/s en coma flotante, versión 5.1F
G.723.1 anexo C (11.96)	Código de referencia C y señales de prueba para el códec de canales escalables, versión 3.1
G.726 apéndice II (03.91)	Secuencias de pruebas digitales para la verificación de los algoritmos MICDA a 40, 32, 24 y 16 kbit/s de la Recomendación G.726
G.727 apéndice I (03.91)	Secuencias de pruebas digitales para la verificación del algoritmo MICDA jerarquizada con 5, 4, 3 y 2 bits/muestra de la Recomendación G.727
G.728 apéndice I (07.95)	Programas y secuencias de prueba para la verificación del algoritmo del codificador de voz LD-CELP a 16 kbit/s de la Recomendación G.728
G.729 (03.96)	Código fuente C y vectores de prueba para la verificación de la implementación del codificador de voz CS-ACELP a 8 kbit/s de la Recomendación G.729
G.729 anexo A (11.96)	Código fuente C y vectores de prueba para la verificación de la implementación del codificador de voz CS-ACELP a 8 kbit/s de complejidad reducida de la Recomendación G.729
G.729 anexo B (10.96)	Código fuente C y vectores de prueba para la verificación de la implementación del algoritmo del esquema de compresión de silencios de la Recomendación G.729
P.501 (08.96)	Señales de prueba para telefonometría
P.861 (08.96)	Código de referencia C de la medición de calidad vocal perceptual (PSQM)
Q.921 <i>bis</i> (03.93)	Sucesión de pruebas abstractas para pruebas de conformidad del procedimiento de acceso al enlace en el canal D – Parte I: Velocidad básica del lado usuario
Q.931 <i>bis</i> (02.95)	PICS y sucesión de pruebas abstractas para la capa 3 de señalización digital de abonado N.º 1 (DSS 1) de la RDSI - Prueba de conformidad de control de la llamada básica en modo circuito
Q.933 <i>bis</i> (10.95)	PICS y sucesión de pruebas abstractas para las pruebas de conformidad del control de llamada básica en modo trama para conexiones virtuales permanentes (PVC) – Sección I: lados usuario y red de la interfaz usuario-red
T.24 (11.94)	Conjunto normalizado de imágenes digitalizadas
T.83 (11.94)	Datos de pruebas de conformidad para el codificador y decodificador genéricos para la compresión digital y codificación de imágenes fijas de tonos continuos