



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 357 234**

51 Int. Cl.:
H04L 29/06 (2006.01)
H04L 12/64 (2006.01)
G06F 3/14 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **04754232 .9**
96 Fecha de presentación : **02.06.2004**
97 Número de publicación de la solicitud: **1629654**
97 Fecha de publicación de la solicitud: **01.03.2006**

54 Título: **Generación e implementación de un protocolo y una interfaz de señales para velocidades de transferencia de datos elevadas.**

30 Prioridad: **02.06.2003 US 475459 P**

45 Fecha de publicación de la mención BOPI:
20.04.2011

45 Fecha de la publicación del folleto de la patente:
20.04.2011

73 Titular/es: **QUALCOMM Incorporated**
5775 Morehouse Drive
San Diego, California 92121, US

72 Inventor/es: **Anderson, Jon, James;**
Steele, Brian;
Wiley, George, A. y
Shekhar, Shashank

74 Agente: **Fàbrega Sabaté, Xavier**

ES 2 357 234 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN**ANTECEDENTES DE LA INVENCION****I. Campo de la invención**

5 La presente invención se refiere a un protocolo y a un proceso de señales digitales para comunicar o transferir señales entre un dispositivo central y un dispositivo cliente de presentación audiovisual a altas velocidades de transferencia de datos. Más específicamente, la presente invención se refiere a una técnica para transferir señales multimedia y otros tipos de señales digitales desde un dispositivo inalámbrico hasta una microunidad de visualización u otro dispositivo de presentación utilizando un mecanismo de transferencia de alta velocidad de transferencia de datos y de baja potencia.

10 II. Técnica relacionada

15 Los ordenadores, productos relacionados con los juegos electrónicos y varias tecnologías de vídeo (por ejemplo, los DVD y los VCR de alta definición) han avanzado significativamente en los últimos años para proporcionar una presentación de imágenes fijas, de vídeo, de vídeo bajo demanda y gráficas con una resolución cada vez mayor, incluso cuando incluyen algún tipo de texto, a usuarios finales de tales equipos. Estos avances han requerido, a su vez, la utilización de dispositivos de visionado electrónicos de mayor resolución tales como monitores de vídeo de alta definición, monitores HDTV o elementos especializados de proyección de imágenes. La combinación de tales imágenes visuales con datos de audio de alta definición o de alta calidad, como cuando se utiliza una reproducción de sonido de tipo CD, DVD y otros dispositivos que también tienen salidas asociadas de señales de audio, se utiliza para crear una experiencia multimedia más realista, rica en contenido o verdadera para un usuario final. Además, mecanismos de transporte de música y sistemas de sonido de alta calidad con gran capacidad de movilidad, tales como los reproductores MP3, se han desarrollado solamente para presentaciones de audio a usuarios finales.

20 En un escenario de presentación de vídeo típico, los datos de vídeo se transfieren normalmente utilizando técnicas actuales a una velocidad que en el mejor de los casos podría calificarse como lenta o media, siendo del orden de un kilobit o de decenas de kilobits por segundo. Después, estos datos se guardan en memorias intermedias o se almacenan en dispositivos de memoria transitoria o de larga duración, para una reproducción aplazada (posterior), proporcionándose a un dispositivo de visionado deseado. Por ejemplo, las imágenes pueden transferirse "a través de" o utilizando Internet mediante un programa que reside en un ordenador que tiene un módem o un dispositivo de conexión a Internet, para recibir o transmitir datos útiles en la representación digital de una imagen. Una transferencia similar puede tener lugar utilizando dispositivos inalámbricos tales como ordenadores portátiles equipados con módems inalámbricos, o asistentes personales de datos (PDA) inalámbricos o teléfonos inalámbricos.

25 Una vez recibidos, los datos se almacenan localmente en elementos, circuitos o dispositivos de memoria, tales como RAM o memoria flash, incluyendo dispositivos de almacenamiento externos, para su reproducción. Dependiendo de la cantidad de datos y de la resolución de la imagen, la reproducción puede empezar de una manera relativamente rápida o presentarse con un retardo de mayor duración. Es decir, en algunos casos, la presentación de imágenes permite un cierto grado de reproducción en tiempo real para imágenes muy pequeñas o de baja resolución que no requieren muchos datos, o utilizar algún tipo de almacenamiento intermedio, de manera que después de un pequeño retardo se presente algún material mientras que está transfiriéndose más material. Siempre que no haya interrupciones en el enlace de transferencia, una vez que la presentación comienza la transferencia es razonablemente transparente para el usuario final del dispositivo de visionado.

30 Los datos utilizados para crear imágenes fijas o vídeo en movimiento se comprimen normalmente utilizando una de varias técnicas ampliamente conocidas, tales como las especificadas por el Grupo de Expertos Fotográfico Común (JPEG), el Grupo de Expertos de Imágenes en Movimiento (MPEG), y otras organizaciones y compañías de normalización ampliamente conocidas en la industria multimedia, informática y de telecomunicaciones para acelerar la transferencia de datos a través de un enlace de comunicaciones. Esto permite transferir imágenes o datos más rápidamente utilizando un menor número de bits para transferir una determinada cantidad de información.

35 Una vez que los datos se han transferido a un dispositivo "local" tal como un ordenador u otro dispositivo receptor, la información resultante se descomprime (o se reproduce utilizando reproductores de decodificación especiales), se descodifica si fuera necesario y se prepara para su presentación apropiada en función de la resolución de presentación disponible correspondiente y de elementos de control. Por ejemplo, una resolución de vídeo en un ordenador típico en lo que se refiere a una resolución de pantalla de X por Y píxeles varía normalmente desde valores tan bajos como 480x640 píxeles hasta 600x800 y 1024x1024, aunque una variedad de otras resoluciones es generalmente posible, según se desee o se necesite.

40 La presentación de imágenes también se ve afectada por el contenido de imagen y por la capacidad de los controladores de vídeo dados de manipular la imagen en lo que respecta a determinados niveles de color predefinidos o profundidad de color (bits por píxel utilizados para generar colores) e intensidades, y a cualquier bit suplementario adicional que esté utilizándose. Por ejemplo, una presentación en un ordenador típico necesitará entre 8 y 32, o más, bits por píxel,

aproximadamente, para representar varios colores (sombras y tonos), aunque otros valores son posibles.

A partir de los valores anteriores, puede observarse que una imagen de pantalla dada va a requerir la transferencia de entre 2,45 Megabits (Mb) y 33,55 Mb de datos, aproximadamente, en el intervalo de profundidad y de resoluciones típicas desde las más bajas hasta las más altas, respectivamente. Cuando se visualiza vídeo o imágenes en movimiento a una velocidad de 30 tramas por segundo, la cantidad de datos requerida está comprendida entre 73,7 y 1.006 Megabits de datos por segundo (Mbps) aproximadamente, o entre 9,21 y 125,75 Megabytes por segundo (MBps) aproximadamente. Además, puede desearse presentar datos de audio junto con imágenes, como para una presentación multimedia o como para otra presentación de audio de alta resolución, tal como música con calidad CD. También pueden utilizarse señales adicionales relacionadas con comandos, controles o señales interactivos. Cada una de estas opciones añade más datos que han de transferirse. En cualquier caso, cuando se desea transferir datos de imagen de alta calidad o alta resolución y señales de datos o información de audio de alta calidad a un usuario final para crear una experiencia rica en contenido, se requiere un enlace de una alta velocidad de transferencia de datos entre los elementos de presentación y el dispositivo fuente o central que está configurado para proporcionar tales tipos de datos.

Interfaces en serie con módems pueden manejar de manera rutinaria velocidades de transferencia de datos de 115 Kilobytes (KBps) aproximadamente o de 920 Kilobits por segundo (Kbps). Otras interfaces tales como interfaces en serie USB pueden permitir transferencias de datos a velocidades tan altas como 12 MBps, y transferencias de alta velocidad especializadas tales como las configuradas utilizando la norma 1394 del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) pueden producirse a velocidades del orden de 100 a 400 MBps. Desafortunadamente, estas velocidades se quedan cortas con respecto a las altas velocidades deseadas de transferencia de datos mencionadas anteriormente, cuya utilización se contempla para futuros dispositivos y servicios de datos inalámbricos para proporcionar señales de salida de alta resolución y ricas en contenido para su presentación en unidades de visualización o dispositivos de audio portátiles. Además, estas interfaces requieren la utilización de una cantidad significativa de software central o de sistema y de software cliente para su funcionamiento. Sus pilas de protocolo de software crean además una gran cantidad no deseable de datos suplementarios, especialmente cuando se contemplan dispositivos inalámbricos móviles o aplicaciones telefónicas. Tales dispositivos tienen importantes limitaciones de memoria y de consumo de potencia, así como una capacidad computacional ya tasada. Además, algunas de estas interfaces utilizan cables voluminosos que son muy pesados y poco adecuados para aplicaciones móviles con una orientación claramente estética, conectores complejos que añaden costes, o simplemente consumen mucha potencia.

Existen otras interfaces conocidas tales como la interfaz de adaptador analógico de gráficos de vídeo (VGA), la interfaz interactiva de vídeo digital (DVI) o la interfaz de vídeo de gigabits (GVIF). Las dos primeras de estas interfaces son interfaces de tipo en paralelo que procesan datos a velocidades de transferencia más altas, pero también utilizan cables pesados y consumen grandes cantidades de potencia, del orden de varios vatios. Ninguna de estas características es recomendable para utilizarse con dispositivos electrónicos portátiles de consumo. La tercera interfaz consume mucha potencia y utiliza conectores caros o voluminosos.

Para algunas de las interfaces anteriores, y para otros mecanismos de transferencia o sistemas/protocolos de gran velocidad de transferencia de datos asociados con transferencias de datos para equipos informáticos de instalación fija, hay otra importante desventaja. Para permitir las velocidades de transferencia de datos deseadas, también se requieren grandes cantidades de potencia y/o un funcionamiento a altos niveles de corriente. Esto reduce en gran medida la utilidad de tales técnicas en productos de consumo con una gran capacidad de movilidad.

Generalmente, para permitir tales velocidades de transferencia de datos utilizando alternativas como, por ejemplo, elementos de transferencia y conexiones de fibra óptica, también se requiere una pluralidad de convertidores y de elementos adicionales que introducen una complejidad y unos costes mucho mayores que los deseados para un producto de consumo meramente comercial. Aparte de la naturaleza generalmente cara, por el momento, de los sistemas ópticos, sus requisitos de potencia y su complejidad impiden su utilización general para aplicaciones ligeras, de baja potencia y portátiles.

Lo que se necesita en la industria de aplicaciones portátiles o móviles es una técnica que proporcione una experiencia de presentación de alta calidad, ya sea de audio, de vídeo o multimedia, para usuarios finales con una gran capacidad de movilidad. Es decir, cuando se utilizan ordenadores portátiles, teléfonos inalámbricos, PDA u otros equipos o dispositivos de comunicación con una gran capacidad de movilidad, los sistemas o dispositivos de presentación de vídeo y audio que se utilizan en la actualidad, simplemente no pueden suministrar una salida al alto nivel de calidad deseado. Normalmente, la baja calidad percibida es el resultado de altas velocidades de transferencia de datos que no pueden conseguirse y que son necesarias para transferir los datos de presentación de alta calidad. Por lo tanto, se necesita un nuevo mecanismo de transferencia para aumentar el caudal de datos entre los dispositivos centrales que proporcionan los datos y los dispositivos o elementos de visualización cliente que presentan una salida a los usuarios finales.

Los solicitantes han propuesto dichos nuevos mecanismos de transferencia en las publicaciones de solicitud de patente WO 03 023587 y US 6 760 772, ambas tituladas "*Generating And Implementing A Communication Protocol And Interface For High Data Rate Signal Transfer*", las cuales están asignadas al cesionario de la presente invención e

incorporadas en este documento como referencia. Las técnicas descritas en esas solicitudes pueden mejorar en gran medida la velocidad de transferencia para grandes cantidades de datos en señales de datos de alta velocidad. Sin embargo, la necesidad de velocidades de datos cada vez mayores, especialmente con relación a las presentaciones de vídeo, siguen creciendo. Incluso con otros desarrollos actuales en la tecnología de señales de datos, todavía existe la necesidad de conseguir velocidades de transferencia más rápidas. Por lo tanto, existe una necesidad continua de desarrollar un nuevo mecanismo o un mecanismo mejorado de transferencia, el cual es necesario para aumentar el caudal de datos entre dispositivos centrales y cliente.

RESUMEN

Las anteriores y otras desventajas existentes en la técnica se afrontan mediante las realizaciones de la presente invención, en las que se ha desarrollado un nuevo protocolo y mecanismo de transferencia de datos para transferir datos entre un dispositivo central y un dispositivo cliente receptor a altas velocidades de transferencia de datos.

Realizaciones de la invención están dirigidas a una interfaz digital de datos móviles (MDDI) para transferir datos digitales a una alta velocidad entre un dispositivo central y un dispositivo cliente a través de una trayectoria de comunicaciones que utiliza una pluralidad o una serie de estructuras de paquete enlazadas entre sí para formar un protocolo de comunicaciones para comunicar un conjunto preseleccionado de datos digitales de presentación y de control entre el dispositivo central y el dispositivo cliente. El protocolo de comunicaciones de señales o capa de enlace se utiliza por una capa física de controladores de enlace de dispositivo central o de dispositivo cliente. Al menos un controlador de enlace que reside en el dispositivo central está acoplado al dispositivo cliente a través de la trayectoria o enlace de comunicaciones y está configurado para generar, transmitir y recibir paquetes que forman el protocolo de comunicaciones, y para formar datos de presentación digitales en uno o más tipos de paquetes de datos. La interfaz proporciona una transferencia bidireccional de información entre el dispositivo central y el dispositivo cliente.

En aspectos adicionales de realizaciones de la invención, al menos un controlador de enlace de dispositivo cliente, o receptor de dispositivo cliente, está dispuesto en el dispositivo cliente y está acoplado al dispositivo central a través de la trayectoria o enlace de comunicaciones. El controlador de enlace de dispositivo cliente también está configurado para generar, transmitir y recibir paquetes que forman el protocolo de comunicaciones, y para formar datos de presentación digitales en uno o más tipos de paquetes de datos. Generalmente, el controlador de dispositivo central o de enlace utiliza una máquina de estados para procesar paquetes de datos utilizados en comandos o determinados tipos de procesamiento de preparación y consulta de señales, pero puede utilizar un procesador de propósito general más lento para manipular datos y algunos de los paquetes menos complejos utilizados en el protocolo de comunicaciones. El controlador de dispositivo central comprende uno o más excitadores de línea diferenciales, mientras que el receptor de dispositivo cliente comprende uno o más receptores de línea diferenciales acoplados a la trayectoria de comunicaciones.

Los paquetes están agrupados conjuntamente dentro de tramas multimedia, que se comunican entre el dispositivo central y el dispositivo cliente, que tienen una longitud fija predefinida con un número predeterminado de paquetes que tienen diferentes longitudes variables. Cada paquete comprende un campo de longitud de paquete, uno o más campos de datos de paquete y un campo de comprobación de redundancia cíclica. Un Paquete de Cabecera de Subtrama se transfiere o se coloca al principio de las transferencias de otros paquetes desde el controlador de enlace de dispositivo central. El protocolo de comunicaciones utiliza uno o más paquetes de tipo Flujo de Vídeo y paquetes de tipo Flujo de Audio para transferir datos de tipo vídeo y datos de tipo audio, respectivamente, desde el dispositivo central hasta el dispositivo cliente a través de un enlace directo para su presentación a un usuario de dispositivo cliente. El protocolo de comunicaciones utiliza uno o más paquetes de tipo Encapsulación de Enlace Inverso para transferir datos desde el dispositivo cliente hasta el controlador de enlace de dispositivo central.

El controlador de enlace de dispositivo central genera paquetes de tipo Relleno para ocupar periodos de transmisión de enlace directo que no tienen datos. El protocolo de comunicaciones utiliza una pluralidad de otros paquetes para transferir información de vídeo. Tales paquetes incluyen paquetes de tipo Mapa de Colores, Transferencia de Bloques de Bits, Relleno de Área de Mapa de Bits, Relleno de Patrón de Mapa de Bits y Habilidad de Color Transparente. El protocolo de comunicaciones utiliza paquetes de tipo Flujo Definido por el Usuario para transferir datos definidos por la interfaz y el usuario. El protocolo de comunicaciones utiliza paquetes de tipo Datos de Teclado y Datos de Dispositivo de Puntero para transferir datos a, o desde, dispositivos de entrada de usuario asociados con dicho dispositivo cliente. El protocolo de comunicaciones utiliza un paquete de tipo Interrupción de Enlace para finalizar la transferencia de datos en cualquier sentido a través de dicha trayectoria de comunicaciones.

La trayectoria de comunicaciones comprende o utiliza generalmente un cable que tiene una serie de cuatro o más conductores y un blindaje. En algunas realizaciones, los controladores de enlace comprenden una interfaz de datos USB y el cable utiliza una interfaz de tipo USB junto con los otros conductores. Además, puede utilizarse un cableado impreso o conductores flexibles, según se desee.

El controlador de enlace de dispositivo central solicita información de capacidades de visualización del dispositivo cliente con el fin de determinar qué tipo de datos y qué velocidades de transferencia de datos puede soportar dicho dispositivo cliente a través de dicha interfaz. El controlador de enlace de dispositivo cliente comunica capacidades de

5 visualización o presentación al controlador de enlace de dispositivo central utilizando al menos un paquete de tipo Capacidades de Dispositivo de Visualización. El protocolo de comunicaciones utiliza múltiples modos de transferencia, permitiendo cada uno la transferencia de un máximo número diferente de bits de datos en paralelo durante un periodo de tiempo dado, pudiendo seleccionarse cada modo mediante una negociación entre los controladores de enlace de dispositivo central y de dispositivo cliente. Estos modos de transferencia pueden ajustarse de manera dinámica durante la transferencia de datos, y no es necesario utilizar el mismo modo en el enlace inverso y en el enlace directo.

10 En otros aspectos de algunas realizaciones de la invención, el dispositivo central comprende un dispositivo de comunicaciones inalámbricas, tal como un teléfono inalámbrico, un PDA inalámbrico o un ordenador portátil que tiene un módem inalámbrico dispuesto en el mismo. Un dispositivo cliente típico comprende una unidad de visualización portátil tal como un microdispositivo de visualización, y/o un sistema de presentación de audio portátil. Además, el dispositivo central puede utilizar medios o elementos de almacenamiento para almacenar datos de presentación o multimedia que han de transferirse para presentarse a un usuario de un dispositivo cliente.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

15 Características y ventajas adicionales de la invención, así como de la estructura y el funcionamiento de varias realizaciones de la invención, se describen en detalle posteriormente con referencia a los dibujos adjuntos. En los dibujos, número de referencia similares indican generalmente etapas de procesamiento o elementos idénticos, funcionalmente similares y/o estructuralmente similares, y el dibujo en que aparece un primer elemento se indica mediante el (los) dígito(s) más a la izquierda del número de referencia.

20 La FIG. 1A ilustra un entorno básico en el que las realizaciones de la invención pueden funcionar, que incluye la utilización de un microdispositivo de visualización utilizado junto con un ordenador portátil.

La FIG. 1B ilustra un entorno básico en el que las realizaciones de la invención pueden funcionar, que incluye la utilización de un microdispositivo de visualización y de elementos de presentación de audio utilizados junto con un transceptor inalámbrico.

25 La FIG. 2 ilustra el concepto global de una interfaz digital de datos móviles con una interconexión entre un dispositivo central y un dispositivo cliente.

La FIG. 3 ilustra la estructura de un paquete útil para realizar transferencias de datos desde un dispositivo cliente hasta un dispositivo central.

30 La FIG. 4 ilustra la utilización de un controlador de enlace MDDI y los tipos de señales que se transmiten entre un dispositivo central y un dispositivo cliente a través de los conductores de enlace de datos físico para las interfaces de Tipo I y de Tipo U.

La FIG. 5 ilustra la utilización de un controlador de enlace MDDI y los tipos de señales que se transmiten entre un dispositivo central y un dispositivo cliente a través de los conductores de enlace de datos físico para las interfaces de Tipo II, II y IV.

La FIG. 6 ilustra la estructura de tramas y subtramas utilizadas para implementar el protocolo de interfaz.

35 La FIG. 7 ilustra la estructura general de paquetes utilizados para implementar el protocolo de interfaz.

La FIG. 8 ilustra el formato de un Paquete de Cabecera de Subtrama.

La FIG. 9 ilustra el formato y el contenido de un Paquete de Relleno.

La FIG. 10 ilustra el formato de un Paquete de Flujo de Vídeo.

La FIG. 11 ilustra el formato y el contenido para el Descriptor de Formato de Datos de Vídeo de la FIG. 10.

40 La FIG. 12 ilustra la utilización de formatos paquetizados y no paquetizados de datos.

La FIG. 13 ilustra el formato de un Paquete de Flujo de Audio.

La FIG. 14 ilustra la utilización de formatos PCM alineados por bytes y paquetizados de datos.

La FIG. 15 ilustra el formato de un Paquete de Flujo Definido por el Usuario.

La FIG. 16 ilustra el formato de un Paquete de Mapa de Colores.

45 La FIG. 17 ilustra el formato de un Paquete de Encapsulación de Enlace Inverso.

La FIG. 18 ilustra el formato de un Paquete de Capacidades de Dispositivo de Visualización.

La FIG. 19 ilustra el formato de un Paquete de Datos de Teclado.

- La FIG. 20 ilustra el formato de un Paquete de Datos de Dispositivo de Puntero.
- La FIG. 21 ilustra el formato de un Paquete de Interrupción de Enlace.
- La FIG. 22 ilustra el formato de un Paquete de Estado y Solicitud de Dispositivo de Visualización.
- La FIG. 23 ilustra el formato de un Paquete de Transferencia de Bloques de Bits.
- 5 La FIG. 24 ilustra el formato de un Paquete de Relleno de Área de Mapa de Bits.
- La FIG. 25 ilustra el formato de un Paquete de Relleno de Patrón de Mapa de Bits.
- La FIG. 26 ilustra el formato de un Paquete de Canal de Datos de Enlace de Comunicaciones.
- La FIG. 27 ilustra el formato de un Paquete de Solicitud de Traspaso de Tipo de Interfaz.
- La FIG. 28 ilustra el formato de un Paquete de Confirmación de Recepción de Tipo de Interfaz.
- 10 La FIG. 29 ilustra el formato de un Paquete de Traspaso de Tipo de Acción.
- La FIG. 30 ilustra el formato de un Paquete de Habilitación de Canal de Audio Directo.
- La FIG. 31 ilustra el formato de un Paquete de Frecuencia de Muestro de Audio Inverso.
- La FIG. 32 ilustra el formato de un Paquete de Datos Suplementarios de Protección de Contenido Digital.
- La FIG. 33 ilustra el formato de un Paquete de Habilitación de Color Transparente.
- 15 La FIG. 34 ilustra el formato de un Paquete de Medición de Retardo de Ida y Vuelta.
- La FIG. 35 ilustra la temporización de eventos durante el Paquete de Medición de Retardo de Ida y Vuelta.
- La FIG. 36 ilustra una implementación de muestra de un generador y verificador CRC útil para implementar la invención.
- 20 La FIG. 37A ilustra la temporización de señales CRC para el aparato de la FIG. 36 cuando se envían paquetes de datos.
- La FIG. 37B ilustra la temporización de señales CRC para el aparato de la FIG. 36 cuando se reciben paquetes de datos.
- La FIG. 38 ilustra etapas de procesamiento para una solicitud de servicio típica sin contienda.
- 25 La FIG. 39 ilustra etapas de procesamiento para una solicitud de servicio típica establecida después de que haya comenzado la secuencia de reinicio de enlace, compitiendo por el inicio del enlace.
- La FIG. 40 ilustra cómo una secuencia de datos puede transmitirse utilizando codificación DATA-STB.
- La FIG. 41 ilustra un sistema de circuitos útil para generar las señales DATA y STB a partir de datos de entrada en el dispositivo central, y para recuperar después los datos en el dispositivo cliente.
- La FIG. 42 ilustra excitadores y resistencias de terminación útiles para implementar una realización.
- 30 La FIG. 43 ilustra etapas y niveles de señal utilizados por un dispositivo cliente para garantizar el servicio del dispositivo central y utilizados por el dispositivo central para proporcionar tal servicio.
- La FIG. 44 ilustra una separación relativa entre transiciones en las líneas Data0, en otras líneas de datos (DataX) y en las líneas estroboscópicas (Stb).
- 35 La FIG. 45 ilustra la presencia de un retardo de respuesta que puede producirse cuando un dispositivo central inhabilita el excitador de dispositivo central después de transferir un paquete.
- La FIG. 46 ilustra la presencia de un retardo de respuesta que puede producirse cuando un dispositivo central permite que el excitador de dispositivo central transfiera un paquete.
- La FIG. 47 ilustra la relación en la entrada del receptor de dispositivo central entre la temporización de los datos que están transfiriéndose y los flancos de subida y de descenso de los impulsos estroboscópicos.
- 40 La FIG. 48 ilustra características de conmutación y un retardo de salida correspondiente de dispositivo cliente generado por la temporización de datos inversos.
- La FIG. 49 ilustra un diagrama de alto nivel de condiciones y etapas de procesamiento de señales mediante las

cuales puede implementarse la sincronización utilizando una máquina de estados.

La FIG. 50 ilustra cantidades de retardo típicas encontradas para el procesamiento de señales en las trayectorias directa e inversa en un sistema que utiliza la MDDI.

La FIG. 51 ilustra una medición marginal de retardo de ida y vuelta.

5 La FIG. 52 ilustra cambios en la velocidad de transferencia de datos de enlace inverso.

La FIG. 53 ilustra una representación gráfica de valores del Divisor de Velocidad Inversa frente la velocidad de transferencia de datos de enlace directo.

Las FIG. 54A y 54B ilustran etapas llevadas a cabo durante el funcionamiento de una interfaz.

La FIG. 55 ilustra una visión general de los paquetes de procesamiento de aparato de interfaz.

10 La FIG. 56 ilustra el formato de un Paquete de Enlace Directo.

La FIG. 57 ilustra valores típicos para el retardo y los desajustes de propagación en una interfaz de enlace de Tipo I.

La FIG. 58 ilustra una temporización de señales Data, Stb y de recuperación de reloj en un enlace de Tipo I para un procesamiento de señales a modo de ejemplo a través de la interfaz.

15 La FIG. 59 ilustra valores típicos para el retardo y los desajustes de propagación en interfaces de enlace de Tipo II, Tipo III o Tipo IV.

Las FIG. 60A, 60B y 60C ilustran diferentes posibilidades para la temporización de dos señales de datos y una señal MDDI_Stb entre sí, siendo, respectivamente, una señal ideal, una señal temprana y una señal tardía.

20 La FIG. 61 ilustra conectores a modo de ejemplo con asignaciones de clavijas de interfaz utilizados con interfaces de Tipo I y de Tipo II.

Las FIG. 62A y 62B ilustran posibles formas de onda de señales MDDI_Data y MDDI_Stb para interfaces de Tipo I y de Tipo II, respectivamente.

La FIG. 63 ilustra un diagrama de alto nivel de etapas y condiciones alternativas de procesamiento de señales mediante las cuales puede implementarse la sincronización utilizando una máquina de estados.

25 La FIG. 64 ilustra una temporización relativa a modo de ejemplo entre una serie de ciclos de reloj y la temporización de varios bits de paquetes de enlace inverso y valores de divisor.

La FIG. 65 ilustra un procesamiento de transferencia de código de error a modo de ejemplo.

La FIG. 66 ilustra un aparato útil para el procesamiento de transferencia de códigos de error.

La FIG. 67A ilustra un procesamiento de transferencia de códigos de error para la sobrecarga de códigos.

30 La FIG. 67B ilustra un procesamiento de transferencia de códigos de error para la recepción de códigos.

La FIG. 68A ilustra etapas de procesamiento para una activación iniciada por el dispositivo central.

La FIG. 68B ilustra etapas de procesamiento para una activación iniciada por el dispositivo cliente.

La FIG. 68C ilustra etapas de procesamiento para una activación iniciada por el dispositivo central y por el dispositivo cliente con contienda.

35 DESCRIPCIÓN DETALLADA DE LAS REALIZACIONES

I. Visión general

40 Un objetivo general de la invención es proporcionar una interfaz digital de visualización móvil (MDDI), tal y como se ha descrito anteriormente, que dé como resultado o proporcione un mecanismo de transferencia económico y de bajo consumo de potencia que permita una transferencia de datos de alta o muy alta velocidad a través de un enlace de comunicaciones de corto alcance entre un dispositivo central y un dispositivo de visualización utilizando un tipo "en serie" de canal o enlace de datos. Este mecanismo es apropiado para implementarse con conectores diminutos y con cables flexibles y delgados que son especialmente útiles para conectar elementos o dispositivos de visualización, tales como micropantallas ponibles (gafas protectoras o proyectores), a ordenadores portátiles, dispositivos de comunicaciones inalámbricas o dispositivos de entretenimiento.

45 Una ventaja de las realizaciones de la invención es que se proporciona una técnica de transferencia de datos que

tiene una baja complejidad, un bajo coste, gran fiabilidad, que se adapta bien a su entorno de utilización y que es muy robusta, siendo al mismo tiempo muy flexible.

La presente invención puede utilizarse en diversas situaciones para comunicar o transferir grandes cantidades de datos a gran velocidad, generalmente para aplicaciones de audio, vídeo o multimedia, desde un dispositivo central o fuente donde se generan o se almacenan datos hasta un dispositivo cliente de visualización o presentación. Una aplicación típica, que se describe posteriormente, es la transferencia de datos desde un ordenador portátil o un módem o teléfono inalámbrico hasta un dispositivo de visualización tal como una pequeña pantalla de vídeo o un microaparato de visualización pònible, tal como en forma de gafas o cascos protectores que contienen pequeñas lentes y pantallas de proyección, o desde un dispositivo central hasta un dispositivo cliente dentro de tales componentes. Es decir, desde un procesador hasta una pantalla interna u otro elemento de presentación.

Las características o atributos de la MDDI son independientes de una tecnología de visualización específica. Es un mecanismo altamente flexible para la transferencia de datos a alta velocidad en lo que respecta a la estructura interna de esos datos, no a los aspectos funcionales de los datos o comandos que implementa. Esto permite ajustar la temporización de los paquetes de datos que están transfiriéndose para adaptarse a las idiosincrasias de los dispositivos de visualización particulares, o a los deseos de visualización únicos para determinados dispositivos, o para cumplir con los requisitos de audio y vídeo combinados para algunos sistemas A-V. La interfaz no depende del elemento de visualización ni del dispositivo cliente, siempre que se siga el protocolo seleccionado. Además, los datos de enlace en serie agregados o la velocidad de transferencia de datos pueden variar en varios órdenes de magnitud, lo que permite a un diseñador de sistemas de comunicaciones o de dispositivos centrales optimizar el coste, los requisitos de potencia, la complejidad del dispositivo cliente y las frecuencias de actualización del dispositivo de visualización.

La interfaz de datos se presenta principalmente para utilizarse en la transferencia de grandes cantidades de datos a alta velocidad a través de un enlace de señales "cableado" o de un pequeño cable. Sin embargo, algunas aplicaciones también pueden utilizar un enlace inalámbrico, incluyendo enlaces ópticos, siempre que esté configurado para utilizar las mismas estructuras de paquetes y datos desarrolladas para el protocolo de interfaz, y pueden mantener el nivel deseado de transferencia con un consumo de potencia y una complejidad lo bastante bajos como para seguir siendo prácticas.

II. Entorno

Una aplicación típica puede observarse en las FIG. 1A y 1B, en las que un ordenador portátil o portable 100 y un teléfono inalámbrico o dispositivo PDA 102 se muestran comunicando datos con dispositivos de visualización 104 y 106, respectivamente, junto con sistemas de reproducción de audio 108 y 112. Además, la FIG. 1A muestra posibles conexiones a un dispositivo de visualización o pantalla 114 más grande o a un proyector de imágenes 116, que solo se muestran en una figura por motivos de claridad, pero también pueden conectarse a un dispositivo inalámbrico 102. El dispositivo inalámbrico puede estar recibiendo datos en este momento o tener ya almacenada una determinada cantidad de datos de tipo multimedia en un elemento o dispositivo de memoria para su posterior presentación para que un usuario final del dispositivo inalámbrico pueda verlos y/o escucharlos. Puesto que un dispositivo inalámbrico típico se utiliza para comunicaciones de voz y de texto simple la mayoría de las veces, tiene una pantalla de visualización bastante pequeña y un sistema de audio sencillo (altavoces) para comunicar información al usuario del dispositivo 102.

El ordenador 100 tiene una pantalla mucho más grande, pero un sistema de sonido externo inadecuado, y todavía se queda corto en comparación con otros dispositivos de presentación multimedia tales como una televisión de alta definición o pantallas de cine. El ordenador 100 se utiliza para fines ilustrativos, y otros tipos de procesadores, videojuegos interactivos o dispositivos electrónicos de consumo también pueden utilizarse con la invención. El ordenador 100 puede utilizar, pero sin limitarse a, o por, un módem inalámbrico u otro dispositivo incorporado para comunicaciones inalámbricas, o puede conectarse a tales dispositivos utilizando un enlace por cable o inalámbrico, según se desee.

Esto hace que la presentación de datos más complejos o "ricos" no sea una experiencia totalmente útil o satisfactoria. Por lo tanto, la industria está desarrollando otros mecanismos y dispositivos para presentar la información a usuarios finales y proporcionar un nivel mínimo de disfrute deseado o una experiencia positiva.

Tal y como se ha descrito anteriormente, varios tipos de dispositivos de visualización se han desarrollado, o están desarrollándose actualmente, para presentar información a usuarios finales del dispositivo 100. Por ejemplo, una o más compañías han desarrollado varias gafas protectoras ponibles que proyectan una imagen delante de los ojos de un usuario de dispositivo para mostrar una presentación visual. Cuando están colocados correctamente, tales dispositivos "proyectan" de manera eficaz una imagen virtual, percibida mediante los ojos de los usuarios, que es mucho mayor que el elemento que proporciona la salida visual. Es decir, un elemento de proyección muy pequeño permite que el ojo (los ojos) del usuario "vea(n)" imágenes a una escala mucho mayor que la posible con pantallas LCD típicas y similares. El uso de imágenes de pantalla virtual más grandes también permite la utilización de imágenes con una resolución mucho mayor que la posible con visualizadores de pantalla LCD más limitados. Otros dispositivos de visualización pueden incluir, pero sin limitarse a, pequeñas pantallas LCD o varios elementos de visualización de visualización de panel planos, lentes de proyección y controladores de pantalla para proyectar imágenes sobre una superficie, etc.

También puede haber elementos adicionales conectados a o asociados con el uso del dispositivo inalámbrico 102 o del ordenador 100 para presentar una salida a otro usuario o a otro dispositivo que a su vez transfiere las señales a otros sitios o las almacena. Por ejemplo, los datos pueden almacenarse en memoria flash, en forma óptica, utilizando por ejemplo un medio de CD grabable, o en un medio magnético tal como un grabador de cinta magnética y dispositivos similares, para su utilización posterior.

Además, muchos dispositivos y ordenadores inalámbricos tienen en la actualidad capacidades de decodificación incorporadas de música MP3, así como otros sistemas y decodificadores de sonido avanzados. Los ordenadores portátiles utilizan capacidades de reproducción de CD y de DVD como regla general, y algunos tienen pequeños lectores de memoria flash dedicados para recibir archivos de audio pregrabados. El problema de tener tales capacidades es que los archivos de música digitales prometen una experiencia rica con características altamente mejoradas, pero solo si los procesos de decodificación y de reproducción pueden llevar el mismo el ritmo. Esto también se aplica a los archivos de vídeo digitales.

Para ayudar en la reproducción del sonido, la FIG. 1a muestra altavoces externos 114 que también pueden estar acompañados de elementos adicionales tales como altavoces de bajos o altavoces de "sonido envolvente" para una proyección de sonido delantera y trasera. Al mismo tiempo, altavoces o auriculares 108 están indicados como incorporados en el mecanismo o montura de soporte del microdispositivo de visualización 106 de la FIG. 1b. Como se sabe, pueden utilizarse otros elementos de reproducción de audio o sonido, incluyendo dispositivos de amplificación de potencia o de ajustes de sonido.

En cualquier caso, tal y como se ha descrito anteriormente, cuando se desee transferir datos de imagen de alta calidad o de alta resolución e información de audio de alta calidad o señales de datos desde una fuente de datos hasta un usuario final a través de uno o más enlaces de comunicaciones 110, se requiere una alta velocidad de datos. Es decir, el enlace de transferencia 110 es claramente un posible cuello de botella en la comunicación de datos descrita anteriormente y limita el rendimiento del sistema ya que los mecanismos de transferencia actuales no consiguen las altas velocidades de transferencia de datos deseadas normalmente. Tal y como se ha descrito anteriormente, por ejemplo, para mayores resoluciones de imagen tales como 1024 por 1024 píxeles, con profundidades de color de 24 a 32 bits por píxel y a velocidades de transferencia de datos de 30 fps, las velocidades de transferencia de datos pueden aproximarse a velocidades superiores a 755 Mbps o más. Además, tales imágenes pueden presentarse como parte de una presentación multimedia que incluye datos de audio y posiblemente señales adicionales relacionadas con juegos interactivos o comunicaciones, o con varios comandos, controles o señales, incrementando adicionalmente la cantidad de datos y la velocidad de transferencia de datos.

También resulta evidente que necesitar menos cables o interconexiones para establecer un enlace de datos significa que los dispositivos móviles asociados con un dispositivo de visualización son más fáciles de utilizar y es más probable que se utilicen por un mayor número de usuarios. Esto es especialmente cierto cuando múltiples dispositivos se utilizan comúnmente para establecer una experiencia audiovisual completa, y más especialmente a medida que aumenta el nivel de calidad de los dispositivos de visualización y de los dispositivos de salida de audio.

Desafortunadamente, las velocidades de transferencia de datos más altas superan la tecnología actual disponible para la transferencia de datos. Lo que se necesita es una técnica para transferir datos a mayores velocidades para el enlace de transferencia de datos o trayectoria de comunicaciones entre los elementos de presentación y la fuente de datos, que permita por consiguiente una estructura de cableado de baja potencia, o de potencia más baja, ligera y lo más simple y económica posible. Los solicitantes han desarrollado una nueva técnica, o procedimiento y aparato, para conseguir estos y otros objetivos para permitir que una variedad de dispositivos móviles, portátiles o incluso de ubicación fija transfieran datos a dispositivos de visualización, microdispositivos de visualización o elementos de transferencia de audio deseados, a velocidades de transferencia de datos muy altas, manteniendo al mismo tiempo un bajo consumo de potencia deseado y una baja complejidad deseada.

III. Arquitectura de un sistema de interfaz de datos digitales de alta velocidad

Con el fin de crear y utilizar de manera eficaz una nueva interfaz de dispositivo, se ha formulado una arquitectura de sistema y de protocolo de señales que proporciona una velocidad de transferencia de datos muy alta utilizando señales de baja potencia. El protocolo se basa en una estructura de tramas comunes y de paquetes, o estructuras enlazadas entre sí para formar un protocolo para comunicar un conjunto preseleccionado de datos o tipos de datos junto con una estructura operativa o de comandos impuesta en la interfaz.

A. Visión general

Los dispositivos conectados por o que se comunican a través del enlace MDDI se denominan como el dispositivo central y el dispositivo cliente, siendo normalmente el dispositivo cliente un dispositivo de visualización de algún tipo. Los datos desde el dispositivo central hasta el dispositivo de visualización se transportan en el sentido directo (denominado como enlace o tráfico directo), y los datos desde el dispositivo de visualización hasta el dispositivo central se transportan en el sentido inverso (enlace o tráfico inverso), según permita el dispositivo central. Esto se ilustra en la configuración

básica mostrada en la FIG. 2. En la FIG. 2, un dispositivo central 202 está conectado a un dispositivo cliente 204 utilizando un canal de comunicación bidireccional 206, el cual se ilustra comprendiendo un enlace directo 208 y un enlace inverso 210. Sin embargo, estos canales están formados por un conjunto común de conductores cuya transferencia de datos conmuta de manera eficaz entre operaciones de enlace directo y de enlace inverso.

5 Tal y como se describe en este documento, el dispositivo central comprende uno de varios tipos de dispositivos que pueden resultar beneficiados al utilizar la presente invención. Por ejemplo, el dispositivo central 202 puede ser un ordenador portátil en forma de un dispositivo informático móvil manual, portátil o similar, puede ser un PDA, un dispositivo de radiomensajería o uno de muchos teléfonos o módems inalámbricos. Como alternativa, el dispositivo central 202 puede ser un dispositivo portátil de presentación o entretenimiento tal como un reproductor portátil de DVD o de CD, o un dispositivo de juegos. Al mismo tiempo, el dispositivo cliente 204 puede comprender una variedad de dispositivos útiles para presentar información a un usuario final. Por ejemplo, un microdispositivo de visualización incorporado en gafas o anteojos protectores, un dispositivo de proyección incorporado en un sombrero o en un casco, una pequeña pantalla o incluso un elemento holográfico incorporado en un vehículo, tal como en una ventana o parabrisas, o varios altavoces, auriculares o sistemas de sonido para presentar música o sonido de alta calidad. Sin embargo, los expertos en la materia reconocerán fácilmente que la presente invención no está limitada a estos dispositivos, habiendo otros muchos productos en el mercado y propuestos para su utilización, los cuales tienen como finalidad proporcionar a los usuarios finales imágenes y sonido de alta calidad en lo que respecta al almacenamiento y al transporte o en lo que respecta a la presentación durante la reproducción. La presente invención es útil para aumentar el caudal de datos entre varios dispositivos para permitir las altas velocidades de transferencia de datos necesarias para llevar a cabo la experiencia de usuario deseada.

B. Tipos de interfaz

La interfaz MDD está pensada para dirigirse a cinco o más tipos físicos, un tanto distintos, de interfaces encontradas en la industria informática y de telecomunicaciones. Hasta el momento están etiquetados simplemente como Tipo I, Tipo II, Tipo III, Tipo IV y Tipo U.

25 La interfaz de Tipo I está configurada como una interfaz de 6 hilos (conductores) que es adecuada para teléfonos móviles o inalámbricos, PDA, libros electrónicos, juegos electrónicos y reproductores multimedia portátiles, tales como reproductores de CD o reproductores MP3, y dispositivos con tipos similares de tecnología de consumo electrónico. La interfaz de Tipo U está configurada como una interfaz de 8 hilos (conductores) que es más adecuada para ordenadores personales portátiles, de tamaño agenda o de escritorio y dispositivos o aplicaciones similares que no requieran que el dispositivo de visualización se actualice rápidamente y que no tengan un controlador de enlace MDDI incorporado. Este tipo de interfaz también se distingue por el uso de una interfaz de bus serie universal (USB) de dos hilos, que es extremadamente útil para dar cabida a los sistemas operativos actuales o soporte software encontrado en la mayoría de ordenadores personales. Las interfaces de tipo U también pueden utilizarse en un modo solamente USB en el que el dispositivo de visualización tiene simplemente un conector USB que se conecta a un puerto USB estándar de un ordenador o dispositivo similar, por ejemplo un dispositivo electrónico de consumo equipado con un puerto de este tipo, tales como cámaras digitales o reproductores de vídeo.

Las interfaces de Tipo II, de Tipo III y de Tipo IV son adecuadas para dispositivos o unidades de visualización de alto rendimiento y utilizan un cableado más extenso y más complejo con conductores adicionales de tipo par trenzado para proporcionar la protección apropiada y transferencias de señales de datos con pocas pérdidas.

40 La interfaz de Tipo I transporta señales que pueden comprender información de visualización, de audio, de control y de señalización limitada, y se utiliza normalmente para dispositivos que no requieren datos de vídeo de velocidad completa con una alta resolución. Este tipo de interfaz está destinada principalmente a dispositivos, tales como dispositivos inalámbricos móviles, en los que una unidad central USB no está disponible normalmente en el dispositivo para la conexión y transferencia de señales. En esta configuración, el dispositivo móvil es un dispositivo central MDDI y actúa como el "maestro" que controla el enlace de comunicaciones desde el dispositivo central, que generalmente envía datos de visualización al dispositivo cliente (enlace o tráfico directo).

45 En esta interfaz, un dispositivo central permite la recepción de datos de comunicación en el dispositivo central desde el dispositivo cliente (enlace o tráfico inverso) enviando un tipo de paquete o comando espacial al dispositivo cliente que le permite tomar el control del bus (enlace) durante una duración especificada y enviar datos al dispositivo central como paquetes inversos. Esto se ilustra en la FIG. 3, donde un tipo de paquete denominado como un paquete de encapsulación (descrito posteriormente) se utiliza para permitir la transferencia de paquetes inversos a través del enlace de transferencia, creando el enlace inverso. El intervalo de tiempo asignado al dispositivo central para interrogar al dispositivo de visualización con respecto a los datos está predeterminado por el dispositivo central y está basado en los requisitos de cada aplicación específica. Este tipo de transferencia de datos bidireccional semidúplex es especialmente ventajosa cuando un puerto USB no está disponible para la transferencia de información o de datos desde el dispositivo cliente.

La interfaz de tipo U transfiere señales que son muy adecuadas para utilizarse en aplicaciones de un ordenador

de escritorio o de un ordenador portátil donde una interfaz USB puede soportarse ampliamente por un gran número de placas base u otro hardware y por el software de sistemas operativos. La utilización de una interfaz USB añadida permite la utilización de características "conectar y listo" y una configuración de aplicación sencilla. La inclusión de USB también permite un flujo bidireccional de propósito general de comandos, datos de estado, de audio, etc., mientras que los datos de vídeo y audio destinados al dispositivo cliente pueden transferirse usando los pares trenzados a baja potencia y alta velocidad. La potencia puede transferirse utilizando otros hilos, tal y como se describirá posteriormente. Realizaciones de la invención que utilizan una interfaz USB permiten transferencias de alta velocidad a través de un conjunto de conductores, implementando además, principalmente, una señalización y control a través de la conexión USB, la cual puede interrumpirse cuando no se utiliza y consume poca potencia.

La interfaz USB es una norma utilizada ampliamente en los modernos equipos informáticos personales, y los detalles de una interfaz USB y su funcionamiento son ampliamente conocidos en la técnica, por lo que no se explican en este documento. Para la interfaz USB, la comunicación entre el dispositivo central y el dispositivo de visualización es compatible con la especificación del bus serie universal, revisión 2.0. En aplicaciones que utilizan la interfaz de tipo U en las que USB es el canal de señalización primario y posiblemente un canal de retorno de voz, es opcional que el dispositivo central interroge al dispositivo cliente a través de señales de datos en serie MDDI.

Los dispositivos de visualización de alto rendimiento que soportan una resolución de tipo HDTV o altas resoluciones similares requieren flujos de datos a una velocidad de transferencia de datos de 1,5 Gbps aproximadamente con el fin de soportar vídeo de movimiento total. La interfaz de Tipo II soporta altas velocidades de transferencia de datos transmitiendo 2 bits en paralelo, el Tipo III transmitiendo 4 bits en paralelo y la interfaz de Tipo IV transfiriendo 8 bits en paralelo. El protocolo utilizado por la MDDI permite que cada dispositivo central de Tipo I, II, III o IV se comunique generalmente con cualquier dispositivo cliente o de visualización de Tipo I, II, III o IV negociando la velocidad de transferencia de datos más alta posible que puede utilizarse. Las capacidades o características disponibles de lo que puede denominarse como el dispositivo menos capaz se utilizan para fijar el rendimiento del enlace. Como regla, incluso para sistemas en los que tanto el dispositivo central como el dispositivo cliente pueden utilizar interfaces de Tipo II, Tipo III o Tipo IV, ambos inician su funcionamiento como una interfaz de Tipo I. Después, el dispositivo central determina la capacidad del dispositivo cliente o de visualización objetivo y negocia una operación de cambio o de reconfiguración al modo de Tipo II, de Tipo III o de Tipo IV, según sea apropiado para la aplicación particular.

Generalmente es posible que el dispositivo central utilice el protocolo de capa de enlace apropiado (descrito posteriormente en mayor detalle) y en cualquier momento para ralentizar o reconfigurar de nuevo el funcionamiento a un modo más lento para ahorrar potencia o acelerarlo o hacerlo pasar a un modo más rápido para soportar transferencias de mayor velocidad, como para un contenido de visualización de mayor resolución. Por ejemplo, un dispositivo central puede cambiar los modos de visualización cuando el sistema de visualización conmuta de una fuente de alimentación tal como una batería a potencia de CA, o cuando la fuente del medio de visualización conmuta a un formato de resolución más bajo o más alto, o una combinación de estas y otras condiciones o eventos puede considerarse como una base para cambiar un modo de visualización o de transferencia de datos.

También es posible que un sistema comunique datos utilizando un modo en un sentido y otro modo en otro sentido. Por ejemplo, un modo de interfaz de Tipo IV puede utilizarse para transferir datos a un dispositivo de visualización a una alta velocidad, mientras que un modo de Tipo I o de Tipo U se utiliza cuando se transfieren datos a un dispositivo central desde dispositivos periféricos tales como un teclado o un dispositivo de puntero.

C. Estructura de interfaz física

La disposición general de un controlador de enlace o de dispositivo para establecer comunicaciones entre el dispositivo central y el dispositivo cliente se muestra en las FIG. 4 y 5. En las FIG. 4 y 5, un controlador de enlace MDDI 402 y 502 se muestra instalado en un dispositivo central 202 y un controlador de enlace MDDI 404 y 504 se muestra instalado en un dispositivo cliente 204. Como antes, el dispositivo central 202 está conectado a un dispositivo cliente 204 utilizando un canal de comunicaciones bidireccional 406 que comprende una serie de conductores. Tal y como se describirá posteriormente, los controladores de enlace de dispositivo central y de dispositivo cliente pueden fabricarse como un circuito integrado utilizando un diseño de circuito único que puede fijarse, ajustarse o programarse para responder como un controlador de dispositivo central (excitador) o un controlador de dispositivo cliente (receptor). Esto permite menores costes debido a una fabricación a mayor escala de un único dispositivo de circuito.

En la FIG. 4, un dispositivo central USB 408 y un dispositivo cliente USB 410 también se muestran para utilizarse en la implementación de versiones de interfaz de Tipo U de la MDDI. Circuitos y dispositivos para implementar tales funciones se conocen ampliamente en la técnica y no se describen en mayor detalle en este documento.

En la FIG. 5, un controlador de enlace MDDI 502 se muestra instalado en un dispositivo central 202' y un controlador de enlace MDDI 504 se muestra instalado en un dispositivo cliente 204'. Como antes, el dispositivo central 202' está conectado a un dispositivo cliente 204' utilizando un canal de comunicaciones bidireccional 506 que comprende una serie de conductores. Tal y como se ha descrito anteriormente, los controladores de enlace de dispositivo central y de dispositivo cliente pueden fabricarse utilizando un diseño de circuito único.

Las señales transmitidas entre un dispositivo central y un dispositivo cliente, tal como un dispositivo de visualización, a través del enlace MDDI o los conductores físicos utilizados, también se ilustran en las FIG. 4 y 5. Tal y como puede observarse en las FIG. 4 y 5, la trayectoria o mecanismo primario para transferir datos a través de la MDDI utiliza señales de datos etiquetadas como MDDI_Data0+/- y MDDI_Stb+/- . Cada una de las mismas son señales de datos de baja tensión que se transfieren a través de un par diferencial de hilos en un cable. Solo hay una transición en el par MDDI_Data0 o en el par MDDI_Stb para cada bit enviado a través de la interfaz. Este es un mecanismo de transferencia basado en tensión no basado en corriente, de manera que el consumo de corriente estática es prácticamente cero. El dispositivo central envía las señales MDDI_Stb al dispositivo de visualización cliente.

Aunque los datos pueden fluir tanto en el sentido directo como en el inverso a través de los pares MDDI_Data, es decir, es una trayectoria de transferencia bidireccional, el dispositivo central es el maestro o controlador del enlace de datos. Las trayectorias de las señales MDDI_Data0 y MDDI_Stb funcionan en un modo diferencial para maximizar la inmunidad al ruido. La velocidad de transferencia de datos para las señales en estas líneas se determina por la velocidad del reloj enviada por el dispositivo central, y puede variar en un intervalo comprendido entre 1 kbps y 400 Mbps o más.

La interfaz de Tipo II contiene un par más de datos o conductores o trayectorias que la de Tipo I, denominado como MDDI_Data1+/- . La interfaz de Tipo III contiene dos pares más de datos o de trayectorias de señal que la interfaz de Tipo II denominados como MDDI_Data2+/- y MDDI_Data3+/- . La interfaz de Tipo IV contiene cuatro pares más de datos o de trayectorias de señal que la interfaz de Tipo III denominados como MDDI_Data4+/-, MDDI_Data5+/-, MDDI_Data6+/- y MDDI_Data7+/-, respectivamente. En cada una de las configuraciones de interfaz anteriores, un dispositivo central puede enviar potencia al dispositivo cliente o dispositivo de visualización utilizando el par de hilos o señales designados como MDDI_Pwr y MDDI_Gnd.

Un tipo de transferencia disponible generalmente sólo para la configuración de Tipo U es la conexión o trayectoria de señal USB MDDI. La conexión USB MDDI comprende una trayectoria secundaria para la comunicación entre un dispositivo central y un dispositivo de visualización cliente. En determinadas aplicaciones puede ser más ventajoso enviar determinada información a una velocidad de transferencia de datos relativamente baja entre un dispositivo central y un dispositivo cliente. La utilización del enlace de transferencia USB permite que dispositivos sin un controlador de enlace MDDI que tengan una unidad central USB o una capacidad de dispositivo central limitada se comuniquen con un dispositivo cliente o dispositivo de visualización compatible con MDDI equipado con la interfaz de Tipo U. Ejemplos de información que puede transferirse de manera útil a través de una interfaz USB a un dispositivo de visualización son: mapas de bits estáticos, flujos de audio digitales, datos de dispositivo de puntero, datos de teclado e información de estado y de control. La funcionalidad soportada a través de la interfaz USB también puede implementarse utilizando la trayectoria de datos primaria en serie de alta velocidad MDDI. Aunque los datos (véanse más abajo los paquetes) definidos anteriormente pueden enviarse a través de una interfaz de tipo USB, los requisitos para encadenar datos en forma de paquetes consecutivos no se aplican a la interfaz USB, ni el uso de paquetes que soportan un cambio de Tipo MDDI.

Un resumen de las señales transmitidas entre el dispositivo central y el dispositivo cliente (de visualización) a través del enlace MDDI se ilustra a continuación en la tabla I, según el tipo de interfaz.

Tabla I

<u>Tipo I</u>	<u>Tipo II</u>	<u>Tipo III</u>	<u>Tipo IV</u>
MDDI_Pwr/Gnd	MDDI_Pwr/Gnd	MDDI_Pwr/Gnd	MDDI_Pwr/Gnd
MDDI_Stb+/-	MDDI_Stb+/-	MDDI_Stb+/-	MDDI_Stb+/-
MDDI_Data0+/-	MDDI_Data0+/-	MDDI_Data0+/-	MDDI_Data0+/-
	MDDI_Data1+/-	MDDI_Data1+/-	MDDI_Data1+/-
		MDDI_Data2+/-	MDDI_Data2+/-
		MDDI_Data3+/-	MDDI_Data3+/-
<u>Tipo U</u>			MDDI_Data4+/-
MDDI_Pwr/Gnd			MDDI_Data5+/-
MDDI_Stb+/-			MDDI_Data6+/-
MDDI_Data0+/-			MDDI_Data7+/-
MDDI_USB+/-			

El cableado utilizado generalmente para implementar la estructura y el funcionamiento anteriores tiene nominalmente una longitud del orden de 1,5 metros y tiene tres pares trenzados de conductores, donde cada uno es a su

vez un hilo AWG 30 de múltiples filamentos. Un blindaje de lámina está envuelto o formado de otro modo por encima de los tres pares trenzados, como un hilo de drenaje adicional. Los pares trenzados y el conductor de drenaje de blindaje terminan en el conector de dispositivo de visualización con el blindaje conectado al blindaje del dispositivo de visualización (cliente), y hay un capa aislante, que cubre a todo el cable, como es bien sabido en la técnica. Los hilos están emparejados de la siguiente manera: MDDI_Gnd con MDDI_Pwr; MDDI_Stb+ con MDDI_Stb-; MDDI_Data0+ con MDDI_Data0-; MDDI_Data1+ con MDDI_Data1-, y así sucesivamente. El diámetro de cable nominal es del orden de 3,0 mm con una impedancia nominal de 85 ohmios \pm 10%, y con una resistencia de CC nominal de 110 ohmios por 1000 pies. La velocidad de propagación de señal debe ser nominalmente de 0,66c, con un retardo máximo a través del cable inferior a 8,0 nseg aproximadamente.

5

10 **D. Tipos de datos y velocidades**

Para conseguir una interfaz útil para una gran variedad de experiencias y aplicaciones de usuario, la interfaz digital de datos móviles (MDDI) proporciona soporte para una variedad de dispositivos de visualización e información de visualización, transductores de audio, teclados, dispositivos de puntero, y otros muchos dispositivos de entrada que pueden estar integrados en o funcionar en colaboración con un dispositivo de visualización móvil, junto con información de control, y combinaciones de los mismos. La interfaz MDD está diseñada para poder permitir una variedad de posibles tipos de flujos de datos que se transmiten entre el dispositivo central y el dispositivo cliente en el sentido de enlace directo o en el sentido de enlace inverso utilizando un número mínimo de cables o conductores. Se soportan flujos tanto isócronos como asíncronos (actualizaciones). Muchas combinaciones de tipos de datos son posibles siempre que la velocidad de transferencia de datos agregados sea inferior o igual a la máxima velocidad de enlace MDDI deseada. Éstas pueden incluir, pero sin limitarse a, los elementos enumerados a continuación en las tablas II y III.

15

20

Tabla II

Transferencia desde dispositivo central hasta dispositivo cliente		
datos de vídeo isócronos	720x480, 12bit, 30f/s	~124,5 Mbps
datos de audio estéreo isócronos	44,1kHz, 16bit, estéreo	~1,4 Mbps
datos gráficos asíncronos	800x600, 12bit, 10f/s,	estéreo ~115.2 Mbps
control asíncrono	mínimo	<< 1,0 Mbps

Tabla III

Transferencia desde dispositivo cliente hasta dispositivo central		
datos de voz isócronos	8 kHz, 8bit	<< 1,0 Mbps
datos de vídeo isócronos	640x480, 12bit, 24f/s ~	88,5 Mbps
estado asíncrono, entrada de usuario, etc.	mínimo	<< 1,0 Mbps

25

La interfaz no es fija, sino que puede extenderse para que pueda soportar la transferencia de una variedad de "tipos" de información que incluye datos definidos por el usuario, para la futura flexibilidad del sistema. Ejemplos específicos de datos que se permiten son: vídeo de movimiento total, ya sea en forma de vídeo comprimido o de campos de mapa de bits de pantalla total o parcial; mapas de bits estáticos a bajas velocidades para conservar la potencia y reducir los costes de implementación; PCM o datos de audio comprimidos en una variedad de resoluciones o velocidades; datos de posición y selección de dispositivo de puntero y datos que pueden definirse por el usuario para capacidades no definidas todavía. Tales datos también pueden transferirse junto con información de control o de estado para detectar la capacidad del dispositivo o fijar parámetros de funcionamiento.

30

La presente invención hace avanzar la técnica ya que se utiliza en transferencias de datos que incluyen, pero sin limitarse a, ver una película (pantalla de vídeo y audio), utilizar un ordenador personal con una visualización personal limitada (pantalla de gráficos, algunas veces combinada con vídeo y audio), jugar a un videojuego en un PC, consola o dispositivo personal (pantalla de gráficos en movimiento, o vídeo y audio sintéticos), "navegar" por Internet, utilizando dispositivos en forma de videoteléfono (vídeo y audio bidireccional de baja velocidad), una cámara para imágenes digitales fijas, o una cámara de vídeo para capturar imágenes de vídeo digitales, y para una mejora en la productividad o para el mero entretenimiento con teléfonos celulares, teléfonos inteligentes o PDA.

35

40

La interfaz de datos móviles descrita posteriormente se presenta en lo que se refiere a proporcionar grandes cantidades de datos de tipo A-V a través de un enlace de comunicaciones o de transferencia que está configurado

generalmente como un enlace de tipo cable o línea cableada. Sin embargo, resultará fácilmente evidente que la estructura, los protocolos, la temporización o el mecanismo de transferencia de señales pueden ajustarse para proporcionar un enlace en forma de un medio óptico o inalámbrico, si puede mantener el nivel deseado de transferencia de datos.

5 Las señales de interfaz MDD utilizan un concepto conocido como la trama común (CF) para el protocolo o la estructura de señales básicos. La idea que subyace a la utilización de una trama común es proporcionar un impulso de sincronización para flujos de datos isócronos simultáneos. Un dispositivo de visualización puede utilizar esta velocidad de trama común como una referencia de tiempo. Una baja velocidad CF aumenta la eficacia del canal al reducir los datos suplementarios para transmitir la cabecera de subtrama. Por otro lado, una alta velocidad CF reduce la latencia y permite
10 inventiva puede programarse de manera dinámica y puede fijarse a uno de muchos valores que son apropiados para los flujos isócronos utilizados en una aplicación particular. Es decir, se selecciona el valor CF que mejor se adapte al dispositivo de visualización y a la configuración de dispositivo central dados, según se desee.

15 El número de bytes requerido generalmente por trama común, que puede ajustarse o programarse, para flujos de datos isócronos cuya utilización es más probable con una aplicación, tal como para un microvisualizador en forma de visor, se muestra en la tabla IV.

Tabla IV

Velocidad de trama común (CFR) = 1200 Hz							
	X	Y	Bit	Velocidad de trama	Canal	Velocidad (Mbps)	Byte/ CFR
Película de DVD	720	480	12	30	1	124,4	12960
Gráficos estéreo	800	600	12	10	2	115,2	12000
Cámara de vídeo	640	480	12	24	1	88,5	9216
Audio de CD	1	1	16	44100	2	1,4	147
Voz	1	1	8	8000	1	0,1	6,7

20 Los cálculos fraccionales de bytes por trama común se obtienen fácilmente utilizando una estructura de contador M/N simple programable. Por ejemplo, un cómputo de 26-2/3 bytes por CF se implementa transfiriendo 2 tramas de 27 bytes, seguidas cada una por una trama de 26 bytes. Una velocidad CF inferior puede seleccionarse para producir un número entero de bytes por CF. Sin embargo, en términos generales, implementar un contador M/N simple en hardware requiere menos espacio dentro de un chip de circuito integrado o módulo electrónico utilizado para implementar parte de o toda la invención que el espacio necesario para una memoria intermedia FIFO más grande de muestras de audio.

25 Una aplicación a modo de ejemplo que ilustra el impacto de diferentes velocidades de transferencia de datos y de tipos de datos es un sistema de Karaoke. Para el Karaoke, un usuario del sistema canta junto con un programa de vídeo musical. Las letras de la canción se muestran en la parte inferior de una pantalla para que el usuario conozca las palabras que va a cantar y, de manera aproximada, la distribución de tiempo de la canción. Esta aplicación requiere una unidad de visualización con actualizaciones gráficas poco frecuentes y mezclar la voz del usuario con un flujo de audio estéreo.

30 Si se supone una velocidad de trama común de 300 Hz, entonces cada CF consistirá en: 92.160 bytes de contenido de vídeo y 588 bytes de contenido de audio (en base a 147 muestras de 16 bits, en estéreo) a través del enlace directo hacia el dispositivo de visualización, y una media de 29,67 (26-2/3) bytes de voz se envían desde un micrófono hasta la máquina de Karaoke móvil. Se envían paquetes asíncronos entre el dispositivo central y el dispositivo de visualización. Esto incluye un máximo de 768 bytes de datos gráficos (la altura de un cuarto de pantalla) y menos de 200 bytes aproximadamente (varios bytes) para diversos comandos de control y de estado.

35 La tabla V muestra cómo están asignados los datos en una trama común para el ejemplo del Karaoke. La velocidad total que está utilizándose se selecciona para que sea de 225 Mbps aproximadamente. Una velocidad ligeramente superior de 226 Mbps permite transferir otros 400 bytes de datos por subtrama aproximadamente, lo que permite la utilización de mensajes de control y de estado ocasionales.

Tabla V

Velocidad de elemento	Bytes/CF
Vídeo musical a 640 x 480 píxeles y 30 fps	92160
Letra de canción a 640 x 120 píxeles y 1 fps	768
Audio de CD a 44.100 sps, estéreo, 16 bits	588
Voz a 8.000 sps, mono, 8 bits	26,67
Cabecera de subtrama	19
Datos suplementarios de enlace inverso	26,67+2*9+20
Bytes/CF totales	93626,33
Velocidad total (Mbps)	224,7032

III. Arquitectura de sistema de interfaz de datos digitales de alta velocidad

E. Capa de enlace

5 Los datos transferidos utilizando las señales de datos en serie de alta velocidad de la interfaz MDD consisten en un flujo de paquetes multiplexados en el tiempo que están enlazados uno detrás de otro. Incluso cuando un dispositivo de transmisión no tiene datos que enviar, un controlador de enlace MDDI envía generalmente de manera automática paquetes de relleno, manteniendo de ese modo un flujo de paquetes. La utilización de una estructura de paquetes simple garantiza una temporización isócrona fiable para señales de vídeo y audio o flujos de datos.

10 Grupos de paquetes están contenidos dentro de elementos o estructuras de señal denominados como subtramas, y grupos subtramas están contenidos dentro de elementos o estructuras de señal denominados como una trama multimedia. Una subtrama contiene uno o más paquetes, dependiendo de su tamaño respectivo y sus usos de transferencia de datos, y una trama multimedia contiene una o más subtramas. La subtrama más grande proporcionada por el protocolo utilizado por la presente invención es del orden de 232-1 ó 4.294.967.295 bytes, y el tamaño más grande de trama multimedia se vuelve entonces del orden de 216-1 ó 65.535 subtramas.

15 Un paquete de cabecera especial contiene un identificador único que aparece al principio de cada subtrama, tal y como se describirá posteriormente. Ese identificador también se utiliza para adquirir la temporización de trama en el dispositivo cliente cuando se inicia la comunicación entre el dispositivo central y el dispositivo cliente. La adquisición de la temporización de enlace se describe posteriormente en mayor detalle.

20 Normalmente, una pantalla de visualización se actualiza una vez por trama multimedia cuando está visualizándose un vídeo de movimiento total. La velocidad de trama de visualización es la misma que la velocidad de trama multimedia. El protocolo de enlace soporta vídeo de movimiento total a lo largo de una visualización completa, o solo una pequeña región de contenido de vídeo de movimiento total rodeado por una imagen estática, dependiendo de la aplicación deseada. En algunas aplicaciones móviles de baja potencia, tal como el visionado de páginas web o de correo electrónico, la pantalla de visualización solo puede necesitar actualizarse ocasionalmente. En esas situaciones, es ventajoso transmitir una única subtrama y después interrumpir o inactivar el enlace para minimizar el consumo de potencia. La interfaz también soporta efectos tales como una visión en estéreo, y maneja primitivas gráficas.

30 Las subtramas existen para permitir la transmisión de paquetes de alta prioridad de manera periódica. Esto permite que flujos isócronos simultáneos coexistan con una cantidad mínima de almacenamiento intermedio de datos. Esto es una ventaja que la presente invención proporciona al proceso de visualización, permitiendo que múltiples flujos de datos (comunicación a alta velocidad de datos de vídeo, de voz, de control, de estado, de dispositivo de puntero, etc.) compartan esencialmente un canal común. Transfiere información utilizando relativamente pocas señales. También permite que haya acciones específicas de la tecnología de visualización, tales como impulsos síncronos horizontales e intervalos de borrado para un monitor CRT.

F. Controlador de enlace

35 El controlador de enlace MDDI mostrado en las FIG. 4 y 5 se fabrica o se ensambla para que sea una implementación completamente digital con la excepción de los receptores de línea diferenciales que se utilizan para recibir datos MDDI y señales estroboscópicas. Sin embargo, incluso los excitadores y receptores de línea diferenciales pueden implementarse en los mismos circuitos integrados digitales con el controlador de enlace. No se requieren funciones analógicas o bucles de enganche de fase (PLL) para implementar el hardware del controlador de enlace. Los controladores

de enlace de dispositivo central y de dispositivo cliente contienen funciones muy similares, con la excepción de la interfaz de visualización que contiene una máquina de estados para la sincronización de enlace. Por lo tanto, la presente invención permite la ventaja práctica de poder crear un diseño o circuito de controlador único que puede configurarse como un dispositivo central o como un dispositivo cliente, que puede reducir los costes de fabricación para los controladores de enlace en su totalidad.

IV. Protocolo de enlace de interfaz

A. Estructura de trama

El protocolo de señales o la estructura de tramas utilizados para implementar la comunicación de enlace directo para la transferencia de paquetes se ilustra en la FIG. 6. Tal y como se muestra en la FIG. 6, la información o los datos digitales se agrupan en elementos conocidos como paquetes. Múltiples paquetes se agrupan a su vez para formar lo que se denomina como una "subtrama", y múltiples subtramas se agrupan a su vez para formar una trama "multimedia". Para controlar la formación de tramas y la transferencia de subtramas, cada subtrama comienza con un paquete predefinido de manera especial denominado como un Paquete de Cabecera de Subtrama (SHP).

El dispositivo central selecciona la velocidad de transferencia de datos que va a utilizarse para una transferencia dada. El dispositivo central puede modificar dinámicamente esta velocidad en función de la capacidad de transferencia máxima del dispositivo central, o de los datos que está recuperando el dispositivo central desde una fuente, y de la capacidad máxima del dispositivo de visualización o de otro dispositivo al que estén transfiriéndose los datos.

Un dispositivo cliente receptor diseñado para, o que puede, funcionar con la MDDI o el protocolo de señales inventivo puede consultarse por el dispositivo central para determinar la velocidad de transferencia de datos máxima, o la máxima actual, que puede utilizar, o puede utilizarse una velocidad mínima más lenta por defecto, así como tipos de datos y características útiles soportados. Esta información puede transferirse utilizando un Paquete de Capacidades de Dispositivo de Visualización (CDP), descrito posteriormente en mayor detalle. El dispositivo de visualización cliente puede transferir datos o comunicarse con otros dispositivos utilizando la interfaz a una velocidad de transferencia de datos mínima preseleccionada o dentro de un intervalo de velocidades mínimas de transferencia de datos, y el dispositivo central realizará una consulta utilizando una velocidad de transferencia de datos dentro de este intervalo para determinar las capacidades totales de los dispositivos cliente.

Otra información de estado que define la naturaleza del mapa de bits y las capacidades de velocidad de trama de vídeo del dispositivo de visualización puede transferirse en un paquete de estado al dispositivo central para que el dispositivo central pueda configurar la interfaz para que sea tan eficaz u óptima como práctica, o según se desee con relación a cualquier limitación del sistema.

El dispositivo central envía paquetes de relleno cuando no hay (más) paquetes de datos a transferir en la presente subtrama, o cuando el dispositivo central no puede transferir a una velocidad suficiente para seguir el mismo ritmo que la velocidad de transmisión de datos escogida para el enlace directo. Puesto que cada subtrama comienza con un paquete de cabecera de subtrama, el final de la subtrama anterior contiene un paquete (de manera más probable un paquete de relleno) y llena exactamente la subtrama anterior. En caso de no haber sitio para paquetes con datos, un paquete de relleno será con gran probabilidad el último paquete en una subtrama, o estará al final de una subtrama siguiente/anterior y antes de un paquete de cabecera de subtrama. La tarea de las operaciones de control en un dispositivo central es garantizar que quede suficiente espacio en una subtrama para cada paquete que va a transmitirse dentro de esa subtrama. Al mismo tiempo, una vez que un dispositivo central haya iniciado el envío de un paquete de datos, el dispositivo central debe poder completar de manera satisfactoria un paquete de ese tamaño dentro de una trama sin incurrir en un estado de subalimentación de datos.

En un aspecto de las realizaciones, la transmisión de subtramas tiene dos modos. Un modo es un modo de subtrama periódico utilizado para transmitir flujos de vídeo y audio en directo. En este modo, la longitud de subtrama se define con valor distinto de cero. El segundo modo es un modo asíncrono o no periódico en el que las tramas se utilizan para proporcionar datos de mapas de bits a un dispositivo de visualización solamente cuando hay nueva información disponible. Este modo se define fijando la longitud de subtrama a cero en el Paquete de Cabecera de Subtrama. Cuando se utiliza el modo periódico, la recepción de paquetes de subtrama puede comenzar cuando el dispositivo de visualización se haya sincronizado para la estructura de tramas de enlace directo. Esto corresponde a los estados "sincronizados" definidos según el diagrama de estados descrito posteriormente con respecto a la FIG. 49 o la FIG. 63. En el modo de subtrama no periódica y no sincronizada, la recepción comienza después de que se haya recibido el primer Paquete de Cabecera de Subtrama.

B. Estructura de paquete global

A continuación se presenta el formato o la estructura de los paquetes utilizados para formular el protocolo de señalización implementado por la presente invención, teniendo en cuenta que la interfaz es extensible y pueden añadirse estructuras de paquete adicionales según se desee. Los paquetes están etiquetados como, o divididos en, diferentes "tipos de paquete" en lo que respecta a su función en la interfaz, es decir, a los comandos o datos que transfieren. Por lo tanto,

5 cada tipo de paquete denota una estructura de paquete predefinida para un paquete dado que se utiliza en la manipulación de los paquetes y los datos que están transfiriéndose. Como resultará fácilmente evidente, los paquetes pueden tener longitudes preseleccionadas o tener longitudes variables o modificables dinámicamente, dependiendo de sus funciones respectivas. Los paquetes también pueden tener distintos nombres, aunque se siga realizando la misma función, como puede ocurrir cuando los protocolos se cambian durante la aceptación de una norma. Los bytes o valores de byte utilizados en los diversos paquetes están configurados como enteros sin signo de múltiples bits (8 ó 16 bits). Un resumen de los paquetes que se utilizan junto con su designaciones de "tipo", enumerados por orden de tipo, se muestra en la tabla VI. También se indica el sentido en el que la transferencia de un paquete se considera válida, indicando también se si utiliza o no para una interfaz de tipo U.

10

Tabla VI

Nombre de Paquete	Tipo de paquete	Válido en sentido		
		Directo	Inverso	Tipo U
Paquete de Cabecera de Subtrama	255	x		x
Paquete de Relleno	0	x	x	
Paquete de Flujo de Vídeo	1	x	x	x
Paquete de Flujo de Audio	2	x	x	x
Paquetes de Flujo Reservados	3 - 55			
Paquetes de Flujo Definidos por el Usuario	56 - 63	x	x	x
Paquete de Mapa de Color	64	x	x	x
Paquete de Encapsulación de Enlace Inverso	65	x		
Paquete de Capacidades de Dispositivo de Visualización	66		x	x
Paquete de Datos de Teclado	67	x	x	x
Paquete de Datos de Dispositivo de Puntero	68	x	x	x
Paquete de Interrupción de Enlace	69	x		
Paquete de Estado y Solicitud de Visualización	70		x	x
Paquete de Transferencia de Bloques de Bits	71	x		x
Paquete de Relleno de Área de Mapa de Bits	72	x		x
Paquete de Relleno de Patrón de Mapa de Bits	73	x		x

Nombre de Paquete	Tipo de paquete	Válido en sentido		
		Directo	Inverso	Tipo U
Canal de Datos de Enlace de Comunicaciones	74	x	x	x
Paquete				
Paquete de Solicitud de traspaso de Tipo de Interfaz	75	x		
Paquete de Confirmación de Recepción de Tipo de Interfaz	76		x	
Paquete de Traspaso de Tipo de Acción	77	x		
Paquete de Habilitación de Canal de Audio Directo	78	x		x

Paquete de Frecuencia de Muestreo de Audio Inverso	79	x		x
Datos Suplementarios de Protección de Contenido Digital	80	x	x	x
Paquete				
Paquete de Habilitación de Color Transparente	81	x		x
Paquete de Medición de Retardo de Ida y Vuelta	82	x		
Paquete de Calibración de Desajustes de Enlace Directo	83	x		

5 Los paquetes tienen una estructura básica común o un conjunto global de campos mínimos que comprenden un campo Longitud de Paquete, un campo Tipo de Paquete, un (varios) campo(s) Bytes de Datos y un campo CRC, que se ilustran en la FIG. 7. Tal y como se muestra en la FIG. 7, el campo Longitud de Paquete contiene información, en forma de un valor de múltiples bits o múltiples bytes, que especifica el número total de bits en el paquete, o su longitud entre el campo de longitud de paquete y el campo CRC. En una realización, el campo de longitud de paquete contiene un ancho de 16 bits o de 2 bytes, entero sin signo, que especifica la longitud del paquete. El campo Tipo de Paquete es otro campo de múltiples bits que designa el tipo de información que está contenida dentro del paquete. En una realización a modo de ejemplo, es un valor con un ancho de 8 bits o 1 byte, en forma de un entero sin signo de 8 bits, y especifica tipos de datos tales como capacidades del dispositivo de visualización, cambios, flujos de vídeo o audio, estado, etc.

10 Un tercer campo es el campo Bytes de Datos, que contiene los bits o datos que están transfiriéndose o enviándose entre el dispositivo central y el dispositivo cliente como parte de ese paquete. El formato de los datos se define específicamente para cada tipo de paquete según el tipo específico de datos que está transfiriéndose, y puede separarse en una serie de campos adicionales, cada uno con sus propios requisitos de formato. Es decir, cada tipo de paquete tendrá un formato definido para esta parte o campo. El último campo es el campo CRC que contiene los resultados de una comprobación de redundancia cíclica de 16 bits calculada sobre los campos Bytes de Datos, Tipo de Paquete y Longitud de Paquete, que se utiliza para confirmar la integridad de la información del paquete. Dicho de otro modo, se calcula a lo largo de todo el paquete excepto para el propio campo CRC. El dispositivo cliente mantiene generalmente un cómputo total de los errores CRC detectados y notifica este cómputo al dispositivo central en el Paquete de Estado y Solicitud de Visualización (véase posteriormente en mayor detalle).

15 Durante la transferencia de los paquetes, los campos se transmiten empezando con el primer bit menos significativo (LSB) y terminando con el bit más significativo (MSB) transmitido en último lugar. Los parámetros que tienen una longitud de más de un byte se transmiten utilizando el primer byte menos significativo, lo que da como resultado el mismo patrón de transmisión de bits utilizado para un parámetro con una longitud mayor que 8 bits, tal y como se utiliza para un parámetro más corto donde el LSB se transmite en primer lugar. Los datos en la trayectoria de señal MDDI_Data0 están alineados con el bit '0' de bytes transmitidos a través de la interfaz en cualquiera de los modos, Tipo I, Tipo II, Tipo In, o Tipo IV.

20 Cuando se manipulan datos para los dispositivos de visualización, los datos de conjuntos de píxeles se transmiten por filas en primer lugar y después por columnas, tal y como se realiza tradicionalmente en la electrónica. Dicho de otro modo, todos los píxeles que aparecen en la misma fila en un mapa de bits se transmiten en orden con el píxel más a la izquierda transmitido en primer lugar y con el píxel más a la derecha transmitido en último lugar. Después de que se haya transmitido el píxel más a la derecha de una fila, entonces el siguiente píxel en la secuencia es el píxel más a la izquierda de la siguiente fila. Las filas de píxeles se transmiten generalmente en orden de arriba a abajo para la mayoría de dispositivos de visualización, aunque pueden permitirse otras configuraciones según sea necesario. Además, en el tratamiento de mapas de bits, el enfoque convencional, el cual se sigue en este documento, define un punto de referencia etiquetando la esquina superior izquierda de un mapa de bits como ubicación o desplazamiento "0,0". Las coordenadas X e Y utilizadas para definir o determinar una posición en el mapa de bits aumentan de valor a medida que desplazan hacia la parte derecha e inferior del mapa de bits, respectivamente. La primera fila y la primera columna empiezan con un valor índice de cero.

40 **C. Definiciones de paquete**

1. Paquete de Cabecera de Subtrama

El paquete de cabecera de subtrama es el primer paquete de cada subtrama y tiene una estructura básica ilustrada en la FIG. 8. Tal y como puede observarse en la FIG. 8, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Palabra Única, Longitud de Subtrama, Versión de Protocolo, Cómputo de

Subtramas y Cómputo de Tramas Multimedia, generalmente en ese orden. Este tipo de paquete está identificado generalmente como un paquete de Tipo 255 (0xff en hexadecimal) y utiliza una longitud fija preseleccionada de 17 bytes.

Mientras que el campo Tipo de Paquete utiliza un valor de 1 byte, el campo Palabra Única utiliza un valor de 3 bytes. La combinación de 4 bytes de estos dos campos forma una palabra única de 32 bits con una buena autocorrelación. En una realización a modo de ejemplo, la palabra única real es 0x005a3bff, donde los 8 bits menos significativos se transmiten en primer lugar como el Tipo de Paquete, y los 24 bits más significativos se transmiten después.

El campo Longitud de Subtrama contiene 4 bytes de información que especifica el número de bytes por subtrama. La longitud de este campo puede fijarse igual a cero para indicar que el dispositivo central solo transmitirá una subtrama antes de que el enlace se interrumpa pasando a un estado inactivo. El valor de este campo puede modificarse dinámicamente "sobre la marcha" cuando se pase de una subtrama a la siguiente. Esta capacidad es útil con el fin de realizar ajustes de temporización mínimos en los impulsos de sincronización para permitir flujos de datos isócronos. Si la CRC del paquete Cabecera de Subtrama no es válida, entonces el controlador de enlace debe utilizar la Longitud de Subtrama de un paquete Cabecera de Subtrama anterior que se sepa que es correcto para estimar la longitud de la subtrama actual.

El campo Versión de Protocolo contiene 2 bytes que especifican la versión de protocolo utilizada por el dispositivo central. El campo Versión de Protocolo se fija a '0' para especificar la primera versión o la versión actual del protocolo que está utilizándose. Este valor cambiará a lo largo del tiempo a medida que se creen nuevas versiones. El campo Cómputo de Subtramas contiene 2 bytes que especifican un número de secuencia que indica el número de subtramas que se han transmitido desde el inicio de la trama multimedia. La primera subtrama de la trama multimedia tiene un Cómputo de Subtramas de cero. La última subtrama de la trama multimedia tiene un valor de n-1, donde n es el número de subtramas por trama multimedia. Debe observarse que si la Longitud de Subtrama está fijada igual a cero (indicando una subtrama no periódica), entonces el Cómputo de Subtramas también debe fijarse igual a cero.

El campo Cómputo de Trama Multimedia contiene 3 bytes que especifican un número de secuencia que indica el número de tramas multimedia que se han transmitido desde el inicio del presente elemento o dato multimedia que está transfiriéndose. La primera trama multimedia de un elemento multimedia tiene un Cómputo de Tramas Multimedia de cero. El Cómputo de Tramas Multimedia se incrementa justo antes de la primera subtrama de cada trama multimedia y vuelve a cero después de utilizar el máximo Cómputo de Tramas Multimedia (por ejemplo, número de tramas multimedia 224-1 = 16.777.215). El dispositivo central puede reajustar el valor Cómputo de Tramas Multimedia generalmente en cualquier momento para satisfacer las necesidades de una aplicación final.

2. Paquete de Relleno

Un paquete de relleno es un paquete que se transfiere a, o desde, un dispositivo cliente cuando ninguna otra información está disponible para enviarse en el enlace directo o en el enlace inverso. Se recomienda que los paquetes de relleno tengan una longitud mínima con el fin de permitir una máxima flexibilidad a la hora de enviar otros paquetes cuando se necesiten. Al final de una subtrama o de un paquete de encapsulación de enlace inverso (véase más abajo), un controlador de enlace fija el tamaño del paquete de relleno para llenar el espacio restante y mantener la integridad del paquete.

El formato y los contenidos de un Paquete de Relleno se muestran en la FIG. 9. Tal y como se muestra en la FIG. 9, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Bytes de Relleno y CRC. Este tipo de paquete está identificado generalmente como un Tipo 0, el cual se indica en el campo de tipo de 1 byte. Los bits o bytes del campo Bytes de Relleno comprenden un número variable de valores de bit todos a cero para permitir que el paquete de relleno tenga la longitud deseada. El paquete de relleno más pequeño no contiene ningún byte en este campo. Es decir, el paquete consiste solamente en la longitud de paquete, el tipo de paquete y la CRC, y utiliza una longitud fija preseleccionada de 3 bytes.

3. Paquete de Flujo de Vídeo

Los Paquetes de Flujo de Vídeo transportan datos de vídeo para actualizar normalmente regiones rectangulares de un dispositivo de visualización. El tamaño de esta región puede ser tan pequeño como un único píxel o tan grande como todo el dispositivo de visualización. Puede haber un número casi ilimitado de flujos visualizados simultáneamente, limitados por los recursos del sistema, porque todo el contexto requerido para visualizar un flujo está contenido dentro del Paquete de Flujo de Vídeo. El formato del paquete de flujo de vídeo (descriptor de formato de datos de vídeo) se muestra en la FIG. 10. Tal y como se observa en la FIG. 10, este tipo de paquete está estructurado para tener los campos Longitud de Paquete (2 bytes), Tipo de Paquete, Descriptor de Datos de Vídeo, Atributos de Visualización, Borde Izquierdo X, Borde Superior Y, Borde Derecho X, Borde Inferior Y, Inicio X e Y, Cómputo de Píxeles, CRC de Parámetros, Datos de Píxel y CRC. Este tipo de paquete está identificado generalmente como un Tipo 1, el cual se indica en el campo de tipo de 1 byte.

El concepto de trama común descrito anteriormente es una manera eficaz de minimizar el tamaño de memoria intermedia de audio y de reducir la latencia. Sin embargo, para los datos de vídeo puede ser necesario distribuir los píxeles de una trama de vídeo a través de múltiples Paquetes de Flujo de Vídeo dentro de una trama multimedia. También es muy

probable que los píxeles de un solo Paquete de Flujo de Vídeo no se correspondan exactamente a una ventana rectangular perfecta en el dispositivo de visualización. Para la velocidad de trama de vídeo a modo de ejemplo de 30 tramas por segundo, hay 300 subtramas por segundo, lo que da como resultado 10 subtramas por trama multimedia. Si hay 480 filas de píxeles en cada trama, cada Paquete de Flujo de Vídeo de cada subtrama contendrá 48 filas de píxeles. En otras situaciones, el Paquete de Flujo de Vídeo puede no contener un número entero de filas de píxeles. Esto es cierto para otros tamaños de trama de vídeo donde el número de subtramas por trama multimedia no se divide de manera equitativa en el número de filas (también conocidas como líneas de vídeo) por trama de vídeo. Cada Paquete de Flujo de Vídeo debe contener generalmente un número entero de píxeles, incluso aunque no contenga un número entero de filas de píxeles. Esto es importante si cada píxel es mayor que un byte, o si están en un formato paquetizado como el mostrado en la FIG. 12.

El formato y los contenidos utilizados para realizar la operación de un campo Descriptor de Datos de Vídeo a modo de ejemplo, mencionado anteriormente, se muestran en las FIG. 11a a 11d. En las FIG. 11 a 11d, el campo Descriptor de Formato de Datos de Vídeo contiene 2 bytes en forma de un entero sin signo de 16 bits que especifica el formato de cada píxel en los Datos de Píxel en el presente flujo del presente paquete. Es posible que diferentes paquetes de Flujo de Vídeo puedan utilizar diferentes formatos de datos de píxel, es decir, utilizar un valor diferente en el Descriptor de Formatos de Datos de Vídeo y, de manera similar, un flujo (región del dispositivo de visualización) puede cambiar su formato de datos sobre la marcha. El Descriptor de Formato de Datos de Vídeo define el formato de píxel solamente para el paquete actual, lo que no implica que se utilice un formato constante durante la vía útil de un flujo de vídeo particular.

Las FIG. 11a a 11d ilustran cómo está codificado el Descriptor de Formato de Datos de Vídeo. Tal y como se utiliza en estas figuras, y en esta realización, cuando los bits [15:13] son iguales a '000', tal y como se muestra en la FIG. 11a, entonces los datos de vídeo consisten en un conjunto de píxeles monocromo donde el número de bits por píxel está definido por los bits 3 a 0 de la palabra Descriptor de Formato de Datos de Vídeo. Los bits 11 a 4 están fijados a cero en esta situación. En cambio, cuando los bits [15:13] son iguales a '001', tal y como se muestra en la FIG. 11b, entonces los datos de vídeo consisten en un conjunto de píxeles de color donde cada uno especifica un color a través de un mapa de colores. En esta situación, los bits 5 a 0 de la palabra Descriptor de Formato de Datos de Vídeo definen el número de bits por píxel, y los bits 11 a 6 están fijados a cero. En cambio, cuando los bits [15:13] son iguales a '010', tal y como se muestra en la FIG. 11c, entonces los datos de vídeo consisten en un conjunto de píxeles de color donde el número de bits por píxel de rojo está definido por los bits 11 a 8, el número de bits por píxel de verde está definido por los bits 7 a 4, y el número de bits por píxel de azul está definido por los bits 3 a 0. En esta situación, el número total de bits en cada píxel es la suma del número de bits utilizados para el rojo, el verde y el azul.

Sin embargo, cuando los bits [15:13] son iguales a '011', tal y como se muestra en la FIG. 11d, entonces los datos de vídeo consisten en un conjunto de datos de vídeo en formato 4:2:2 con información de luminancia y crominancia, donde el número de bits por píxel de luminancia (Y) está definido por los bits 11 a 8, el número de bits del componente Cr está definido por los bits 7 a 4, y el número de bits del componente Cb está definido por los bits 3 a 0. El número total de bits en cada píxel es la suma del número de bits utilizado para el rojo, el verde y el azul. Los componentes Cr y Cb se envían a la mitad de velocidad de Y. Además, las muestras de vídeo en la parte de Datos de Píxel de este paquete se organizan de la siguiente manera: $Y_n, Cr_n, Cb_n, Y_{n+1}, Y_{n+2}, Cr_{n+2}, Cb_{n+2}, Y_{n+3}, \dots$, donde Cr_n y Cb_n están asociados con Y_n e Y_{n+1} , y Cr_{n+2} y Cb_{n+2} están asociados con Y_{n+2} e Y_{n+3} , etc. Si hay un número impar de píxeles en una fila (Borde Derecho X - Borde Izquierdo X + 1) en el flujo actual, entonces el valor Cb correspondiente al último píxel en cada fila estará seguido por el valor Y del primer píxel de la siguiente fila.

Para cada uno de los cuatro formatos mostrados en las figuras, el bit 12, que está designado como "P", especifica si las muestras de Datos de Píxel son datos de píxel empaquetados o alineados por bytes. Un valor de '0' en este campo indica que cada píxel y cada color de cada píxel en el campo Datos de Píxel está alineado por bytes con un límite de byte de interfaz MDD. Un valor de '1' indica que cada píxel y cada color de cada píxel de los Datos de Píxel están empaquetados contra el píxel o color anterior de un píxel sin dejar ningún bit sin utilizar.

El primer píxel del primer paquete de flujo de vídeo de una trama multimedia para una ventana de visualización particular irá en la esquina superior izquierda de la ventana de flujo definida por un Borde Izquierdo X y un Borde Superior Y, y el siguiente píxel recibido se coloca en la siguiente ubicación de píxel de la misma fila, y así sucesivamente. En este primer paquete de una trama multimedia, el valor de inicio X será normalmente igual al Borde Izquierdo X, y el valor de inicio Y será normalmente igual al Borde Superior Y. En posteriores paquetes correspondientes a la misma ventana de pantalla, los valores de inicio X e Y se fijarán normalmente a la ubicación de píxel de la ventana de pantalla que va normalmente detrás del último píxel enviado en el Paquete de Flujo de Vídeo que se transmitió en la subtrama anterior.

4. Paquete de Flujo de Audio

Los paquetes de flujo de audio transportan datos de audio que van a reproducirse a través del sistema de audio del dispositivo de visualización, o para un dispositivo de presentación de audio autónomo. Diferentes flujos de datos de audio pueden asignarse a distintos canales de audio en un sistema de sonido, por ejemplo: delantero-izquierdo, delantero-derecho, central, trasero-izquierdo y trasero-derecho, dependiendo del tipo de sistema de audio que esté utilizándose. Un

complemento total de canales de audio se proporciona para auriculares que contienen un procesamiento mejorado de señales espaciales y acústicas. El formato de los Paquetes de Flujo de Audio se ilustra en la FIG. 13. Tal y como se muestra en la FIG. 13, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, ID de Canal de Audio, Cómputo de Muestras de Audio, Bits por Muestra y Empaquetado, Frecuencia de Muestreo de Audio, CRC de Parámetros, Datos de Audio Digitales y CRC de Datos de Audio. En una realización, este tipo de paquete está identificado generalmente como un paquete de Tipo 2.

El campo Bits por Muestra y Empaquetado contiene un 1 byte en forma de un entero sin signo de 8 bits que especifica el formato de empaquetado de los datos de audio. El formato utilizado generalmente se indica en los bits 4 a 0 que definen el número de bits por muestra de audio PCM. El bit 5 especifica si las muestras de Datos de Audio Digitales están empaquetadas o no. La diferencia entre muestras de audio empaquetadas y alineadas por bytes se ilustra en la FIG. 14. Un valor de '0' indica que cada muestra de audio PCM del campo Datos de Audio Digitales está alineada por bytes con un límite de byte de interfaz MDDI, y un valor '1' indica que cada muestra de audio PCM sucesiva está empaquetada contra la muestra de audio anterior. Este bit solo es eficaz cuando el valor definido en los bits 4 a 0 (el número de bits por muestra de audio PCM) no es un múltiplo de ocho. Los bits 7 a 6 están reservados para un uso futuro y están fijados generalmente a un valor de cero.

5. Paquetes de Flujo Reservados

En una realización, los tipos de paquete 3 a 55 están reservados para paquetes de flujo que se definirán para utilizarse en futuras versiones o variaciones de los protocolos de paquete, según se desee para varias aplicaciones encontradas. Nuevamente, esto es parte de hacer la interfaz MDD más flexible y útil frente a los diseños de sistema y la tecnología en continua evolución en comparación con otras técnicas.

6. Paquetes de Flujo Definidos por el Usuario

Ocho tipos de flujo de datos, conocidos como Tipos 56 a 63, están reservados para utilizarse en aplicaciones propietarias, que pueden definirse por fabricantes de equipos, para utilizarse con un enlace MDDI. Se conocen como Paquetes de Flujo Definidos por el Usuario. Los paquetes de flujo de vídeo transportan datos de vídeo para actualizar (o no) una región rectangular del dispositivo de visualización. La definición de los parámetros de flujo y de los datos para estos tipos de paquete se deja a los fabricantes de equipos específicos que solicitan su utilización. El formato de los paquetes de flujo definidos por el usuario se ilustra en la FIG. 15. Tal y como se muestra en la FIG. 15, este tipo de paquete está estructurado para tener los campos Longitud de Paquete (2 bytes), Tipo de Paquete, número de ID de Flujo, Parámetros de Flujo, CRC de Parámetros, Datos de Flujo y CRC de Datos De Flujo.

7. Paquetes de Mapa de Colores

Los paquetes de mapas de colores especifican los contenidos de una tabla de consulta de mapa de colores utilizada para presentar colores a un dispositivo de visualización. Algunas aplicaciones pueden requerir un mapa de colores que sea más grande que la cantidad de datos que puede transmitirse en un único paquete. En estos casos, pueden transferirse múltiples paquetes de Mapa de Colores, cada uno con un subconjunto diferente del mapa de colores utilizando los campos de desplazamiento y de longitud descritos posteriormente. El formato del Paquete de Mapa de Colores se ilustra en la FIG. 16. Tal y como se muestra en la FIG. 16, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Tamaño de Datos de Mapa de Colores, Desplazamiento de Mapa de Colores, CRC de Parámetros, Datos de Mapa de Colores y CRC de Datos. Este tipo de paquete está identificado generalmente como un paquete de Tipo 64.

8. Paquetes de Encapsulación de Enlace Inverso

En una realización a modo de ejemplo, los datos se transfieren en el sentido inverso utilizando un Paquete de Encapsulación de Enlace Inverso. Se envía un paquete de enlace directo y la operación de enlace MDDI (dirección de transferencia) se modifica o se invierte a la mitad de este paquete para que los paquetes puedan enviarse en el sentido inverso. El formato del Paquete de Encapsulación de Enlace Inverso se ilustra en la FIG. 17. Tal y como se muestra en la FIG. 17, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Banderas de Enlace Inverso, Longitud de Inversión, CRC de Parámetro, Inversión 1, Paquetes de Datos Inversos e Inversión 2. Este tipo de paquete está identificado generalmente como un paquete de Tipo 65.

El controlador de enlace MDDI se comporta de una manera especial cuando envía un Paquete de Encapsulación de Enlace Inverso. La interfaz MDD tiene una señal estroboscópica que siempre se excita por el dispositivo central. El dispositivo central se comporta como si estuviera transmitiendo un cero para cada bit de las partes de Inversión y de Paquetes de Datos Inversos del Paquete de Encapsulación de Enlace Inverso. El dispositivo central hace oscilar una señal MDDI_Strobe en cada límite de bit durante los dos tiempos de inversión y durante el tiempo asignado a los paquetes de datos inversos. (Este es el mismo comportamiento que si estuviera transmitiendo datos todos a cero). El dispositivo central inhabilita sus excitadores de línea de señal de datos MDDI durante el periodo de tiempo especificado por Inversión 1, y el dispositivo cliente vuelve a habilitar sus excitadores de línea durante el campo Rehabilitación de Excitador que sigue al periodo de tiempo especificado por el campo Inversión 2. El dispositivo de visualización lee el parámetro Longitud de

Inversión y excita las señales de datos hacia el dispositivo central inmediatamente después del último bit del campo Inversión 1. El dispositivo de visualización utiliza los parámetros Longitud de Paquete y Longitud de Inversión para conocer la cantidad de tiempo que tiene disponible para enviar paquetes al dispositivo central. El cliente puede enviar paquetes de relleno o excitar las líneas de datos a un estado de cero cuando no tiene datos que enviar al dispositivo central. Si las líneas de datos se excitan a cero, el dispositivo central interpreta esto como un paquete con una longitud cero (una longitud no válida) y el dispositivo central no acepta ningún paquete más del dispositivo cliente para la duración del Paquete de Encapsulación de Enlace Inverso actual.

El dispositivo de visualización cliente excita las líneas de datos MDDI al nivel de cero durante al menos un periodo de reloj de enlace inverso antes del inicio del campo Inversión 2. Esto mantiene las líneas de datos en un estado determinista durante el periodo de tiempo de Inversión 2. Si el dispositivo cliente no tiene más paquetes que enviar, puede incluso inhabilitar las líneas de datos después de excitarlas a un nivel de cero ya que las resistencias de polarización de hibernación (descritas posteriormente) mantienen las líneas de datos a un nivel de cero para el resto del campo Paquetes de Datos Inversos.

El campo de Solicitud de Enlace Inverso del Paquete de Estado y Solicitud de Visualización puede utilizarse para informar al dispositivo central sobre el número de bytes que necesita el dispositivo de visualización en el Paquete de Encapsulación de Enlace Inverso para enviar datos al dispositivo central. El dispositivo central intenta conceder la solicitud asignando al menos ese número de bytes en el Paquete de Encapsulación de Enlace Inverso. El dispositivo central puede enviar más de un Paquete de Encapsulación de Enlace Inverso en una subtrama. El dispositivo de visualización pueden enviar un Paquete de Estado y Solicitud de Visualización casi en cualquier momento, y el dispositivo central interpretará el parámetro Solicitud de Enlace Inverso como el número total de bytes solicitados en una subtrama.

9. Paquetes de Capacidades de Dispositivo de Visualización

Un dispositivo central necesita conocer la capacidad del dispositivo de visualización (cliente) con el que está comunicándose para configurar el enlace de dispositivo central a dispositivo de visualización de una manera generalmente óptima o deseada. Se recomienda que un dispositivo de visualización envíe un Paquete de Capacidades de Dispositivo de Visualización al dispositivo central después de adquirir la sincronización de enlace directo. La transmisión de un paquete de este tipo se considera necesaria cuando la solicita el dispositivo central utilizando las Banderas de Enlace Inverso del Paquete de Encapsulación de Enlace Inverso. El formato del Paquete de Capacidades de Dispositivo de Visualización se ilustra en la FIG. 18. Tal y como se muestra en la FIG. 18, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Versión de Protocolo, Versión Mínima de Protocolo, Anchura de Mapa de Bits, Altura de Mapa de Bits, Capacidad Monocromo, Capacidad de Mapa de Colores, Capacidad RGB, Capacidad Y Cr Cb, Capacidad de Características de Visualización, Capacidad de Velocidad de Transferencia de Datos, Capacidad de Velocidad de Trama, Profundidad de Memoria Intermedia de Audio, Capacidad de Flujo de Audio, Capacidad de Velocidad de Audio, Velocidad Mínima de Subtrama y CRC. En una realización a modo de ejemplo, este tipo de paquete está identificado generalmente como un paquete de Tipo 66.

10. Paquetes de Datos de Teclado

Un paquete de datos de teclado se utiliza para enviar datos de teclado desde el dispositivo cliente hasta el dispositivo central. Un teclado inalámbrico (o cableado) puede utilizarse junto con varios dispositivos de visualización o dispositivos de audio, incluyendo, pero sin limitarse a, un visualizador en forma de visor/dispositivo de presentación de audio. El Paquete de Datos de Teclado transmite al dispositivo central los datos de teclado recibidos desde uno de varios dispositivos conocidos a modo de teclado. Este paquete también puede usarse en el enlace directo para enviar datos al teclado. El formato de un Paquete de Datos de Teclado se muestra en la FIG. 19 y contiene un número variable de bytes de información de o para un teclado. Tal y como se muestra en la FIG. 19, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Datos de Teclado y CRC. En este documento, este tipo de paquete está identificado generalmente como un paquete de Tipo 67.

11. Paquetes de Datos de Dispositivo de Puntero

Un paquete de datos de dispositivo de puntero se utiliza para enviar información de posición de un ratón inalámbrico u otro dispositivo de puntero desde el dispositivo de visualización hasta el dispositivo central. Los datos también pueden enviarse al dispositivo de puntero a través del enlace directo utilizando este paquete. Un formato a modo de ejemplo de un Paquete de Datos de Dispositivo de Puntero se muestra en la FIG. 20 y contiene un número variable de bytes de información de o para un dispositivo de puntero. Tal y como se muestra en la FIG. 20, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Datos de Dispositivo de Puntero y CRC. En una realización a modo de ejemplo, este tipo de paquete está identificado generalmente como un paquete de Tipo 68 en el campo de tipo de 1 byte.

12. Paquetes de Interrupción de Enlace

Un Paquete de Interrupción de Enlace se envía desde el dispositivo central hasta el dispositivo de visualización cliente para indicar que las señales de datos MDDI y las señales estroboscópicas se interrumpirán y pasarán a un estado

de "hibernación" de bajo consumo de potencia. Este paquete es útil para interrumpir el enlace y conservar la potencia después de que se envíen mapas de bits estáticos desde un dispositivo de comunicaciones móviles hasta el dispositivo de visualización, o cuando no hay más información que transferir por el momento desde un dispositivo central hasta un dispositivo cliente. El funcionamiento normal se reanuda cuando el dispositivo central envía paquetes de nuevo. El primer paquete enviado después de la hibernación es un paquete de cabecera de subtrama. El formato de un Paquete de Estado de Dispositivo de Visualización se muestra en la FIG. 21. Tal y como se muestra en la FIG. 21, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete y CRC. En una realización, este tipo de paquete está identificado generalmente como un paquete de Tipo 69 en el campo de tipo de 1 byte y utiliza una longitud fija preseleccionada de 3 bytes.

En el estado de hibernación de baja potencia, el excitador de señal MDDI_Data está inhabilitado en un estado de alta impedancia, y las señales MDDI_Data se fijan a un estado de cero lógico utilizando una red de polarización de alta impedancia que puede sobreexcitarse por el dispositivo de visualización. La señal estroboscópica utilizada por la interfaz se fija a un nivel de cero lógico en el estado de hibernación para minimizar el consumo de potencia. El dispositivo central o el dispositivo cliente puede hacer que el enlace MDDI "se despierte" del estado de hibernación descrito en este documento, lo que es un avance clave para y una ventaja de la presente invención.

13. Paquetes de Estado y Solicitud de Dispositivo de Visualización

El dispositivo central necesita una pequeña cantidad de información del dispositivo de visualización para que pueda configurar el enlace de dispositivo central a dispositivo de visualización de una manera generalmente óptima. Se recomienda que el dispositivo de visualización envíe un Paquete de Estado de Dispositivo de Visualización al dispositivo central en cada subtrama. El dispositivo de visualización debe enviar este paquete como el primer paquete en el Paquete de Encapsulación de Enlace Inverso para garantizar que se entregue de manera fiable al dispositivo central. El formato de un Paquete de Estado de Dispositivo de Visualización se muestra en la FIG. 22. Tal y como se muestra en la FIG. 22, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Solicitud de Enlace Inverso, Cómputo de Errores de CRC y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 70 en el campo de tipo de 1 byte y utiliza una longitud fija preseleccionada de 8 bytes.

El campo Solicitud de Enlace Inverso puede utilizarse para informar al dispositivo central sobre el número de bytes que necesita el dispositivo de visualización en el Paquete de Encapsulación de Enlace Inverso para enviar datos al dispositivo central. El dispositivo central debe intentar conceder la solicitud asignando al menos ese número de bytes en el Paquete de Encapsulación de Enlace Inverso. El dispositivo central puede enviar más de un Paquete de Encapsulación de Enlace Inverso en una subtrama con el fin de permitir datos. El dispositivo cliente puede enviar un Paquete de Estado y Solicitud de Dispositivo de Visualización en cualquier momento y el dispositivo central interpretará el parámetro de Solicitud de Enlace Inverso como el número total de bytes solicitados en una subtrama. Detalles adicionales y ejemplos específicos de cómo se envían datos de enlace inverso al dispositivo central se muestran posteriormente.

14. Paquetes de Transferencia de Bloques de Bits

El Paquete de Transferencia de Bloques de Bits proporciona un medio para desplazar regiones del dispositivo de visualización en cualquier dirección. Los dispositivos de visualización que tienen esta capacidad notificarán la capacidad en el bit 0 del campo Indicadores de Capacidad de Características de Dispositivo de Visualización del Paquete de Capacidades de Dispositivo de Visualización. El formato de un Paquete de Transferencia de Bloques de Bits se muestra en la FIG. 23. Tal y como se muestra en la FIG. 23, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Valor X Superior Izquierdo, Valor Y Superior Izquierdo, Anchura de Ventana, Altura de Ventana, Movimiento X de Ventana, Movimiento Y de Ventana y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 71 y utiliza una longitud fija preseleccionada de 15 bytes.

Los campos se utilizan para especificar los valores X e Y de la coordenada de la esquina superior izquierda de la ventana que va a moverse, la anchura y la altura de la ventana que va a moverse y el número de píxeles en los que la ventana va a moverse horizontalmente y verticalmente, respectivamente. Valores positivos para los dos últimos campos hacen que la ventana se mueva hacia la derecha y hacia abajo, y valores negativos provocan un movimiento hacia la izquierda y hacia arriba, respectivamente.

15. Paquetes de Relleno de Área de Mapa de Bits

El Paquete de Relleno de Área de Mapa de Bits proporciona un medio para inicializar fácilmente una región del dispositivo de visualización a un único color. Los dispositivos de visualización que tienen esta capacidad notificarán la capacidad en el bit 1 del campo Indicadores de Capacidad de Características de Dispositivo de Visualización del Paquete de Capacidades de Dispositivo de Visualización. El formato de un Paquete de Relleno de Área de Mapa de Bits se muestra en la FIG. 24. Tal y como se muestra en la FIG. 24, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Valor X Superior Izquierdo, Valor Y Superior Izquierdo, Anchura de Ventana, Altura de Ventana, Descriptor de Formato de Datos, Valor de Relleno de Área de Píxeles y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 72 en el campo de tipo de 1 byte y utiliza una longitud fija

preseleccionada de 17 bytes.

16. Paquetes de Relleno de Patrón de Mapa de Bits

El Paquete de Relleno de Patrón de Mapa de Bits proporciona un medio para inicializar fácilmente una región del dispositivo de visualización a un patrón preseleccionado. Los dispositivos de visualización que tienen esta capacidad notificarán la capacidad en el bit 2 del campo Indicadores de Capacidad de Características de Dispositivo de Visualización del Paquete de Capacidades de Dispositivo de Visualización. La esquina superior izquierda del patrón de relleno se alinea con la esquina superior izquierda de la ventana que va a rellenarse. Si la ventana que va a rellenarse es más ancha o más alta que el patrón de relleno, entonces el patrón puede repetirse de manera horizontal o vertical varias veces para rellenar la ventana. La parte derecha o inferior del último patrón repetido se trunca según sea necesario. Si la ventana es más pequeña que el patrón de relleno, entonces el lado derecho o la parte inferior del patrón de relleno puede truncarse para ajustarse a la ventana.

El formato de un Paquete de Relleno de Patrón de Mapa de Bits se muestra en la FIG. 25. Tal y como se muestra en la FIG. 25, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Valor X Superior Izquierdo, Valor Y Superior Izquierdo, Anchura de Ventana, Altura de Ventana, Anchura de Patrón, Altura de Patrón, Descriptor de Formato de Datos, CRC de Parámetros, Datos de Píxel de Patrón y CRC de Datos de Píxel. Este tipo de paquete está identificado generalmente como un paquete de Tipo 73 en el campo de tipo de 1 byte.

17. Paquetes de Canal de Datos de Enlace de Comunicaciones

El Paquete de Canal de Datos de Enlace de Comunicaciones proporciona un medio para que un dispositivo de visualización con una capacidad computacional de alto nivel, tal como un PDA, se comunique con un transceptor inalámbrico tal como un teléfono celular o un dispositivo de puerto de datos inalámbrico.

En esta situación, el enlace MDDI actúa como una interfaz adecuada de alta velocidad entre el dispositivo de comunicaciones y el dispositivo informático con el dispositivo de visualización móvil, donde este paquete transporta datos en una Capa de Enlace de Datos de un sistema operativo para el dispositivo. Por ejemplo, este paquete puede utilizarse si un navegador web, un cliente de correo electrónico o un PDA están incorporados en un dispositivo de visualización móvil. Los dispositivos de visualización que tienen esta capacidad notificarán la capacidad en el bit 3 del campo Indicadores de Capacidad de Características de Dispositivo de Visualización del Paquete de Capacidades de Dispositivo de Visualización.

El formato de un Paquete de Canal de Datos de Enlace de Comunicaciones se muestra en la FIG. 26. Tal y como se muestra en la FIG. 26, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, CRC de Parámetros, Datos de Enlace de Comunicaciones y CRC de Datos de Comunicaciones. Este tipo de paquete está identificado generalmente como un paquete de Tipo 74 en el campo de tipo.

18. Paquetes de Solicitud de Traspaso de Tipo de Interfaz

El Paquete de Solicitud de Traspaso de Tipo de Interfaz permite al dispositivo central solicitar que el dispositivo cliente o de visualización pase del modo actual o existente a los modos de Tipo I (serie), Tipo II (paralelo de 2 bits), Tipo III (paralelo de 4 bits) o Tipo IV (paralelo de 8 bits). Antes de que el dispositivo central solicite un modo particular, debe confirmar que el dispositivo de visualización puede funcionar en el modo deseado examinando los bits 6 y 7 del campo Indicadores de Capacidad de Características de Dispositivo de Visualización del Paquete de Capacidades de Dispositivo de Visualización. El formato de un Paquete de Solicitud de Traspaso de Tipo de Interfaz se muestra en la FIG. 27. Tal y como se muestra en la FIG. 27, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Tipo de Interfaz y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 75 y utiliza una longitud fija preseleccionada de 4 bytes.

19. Paquetes de Confirmación de Recepción de Tipo de Interfaz

El Paquete de Confirmación de Recepción de Tipo de Interfaz se envía por el dispositivo de visualización para confirmar la recepción del Paquete de Traspaso de Tipo de Interfaz. El modo solicitado, Tipo I (serie), Tipo II (paralelo de 2 bits), Tipo III (paralelo de 4 bits) o Tipo IV (paralelo de 8 bits), se transmite al dispositivo central como un parámetro en este paquete. El formato de un Paquete de Confirmación de Recepción de Tipo de Interfaz se muestra en la FIG. 28. Tal y como se muestra en la FIG. 28, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Tipo de Interfaz y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 76 y utiliza una longitud fija preseleccionada de 4 bytes.

20. Paquetes de Traspaso de Tipo de Acción

El Paquete de Traspaso de Tipo de Acción es un medio para que el dispositivo central ordene al dispositivo de visualización pasar al modo especificado en este paquete. Éste será el mismo modo que el solicitado y confirmado anteriormente mediante el Paquete de Solicitud de Traspaso de Tipo de Interfaz y el Paquete de Confirmación de Recepción de Tipo de Interfaz. El dispositivo central y el dispositivo de visualización deben pasar al modo acordado después de que se envíe este paquete. El dispositivo de visualización puede perder y recuperar la sincronización de

enlace durante el traspaso de modo. El formato de un Paquete de Traspaso de Tipo de Acción se muestra en la FIG. 29. Tal y como se muestra en la FIG. 29, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Tipo de paquete y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 77 en el campo de tipo de 1 byte y utiliza una longitud fija preseleccionada de 4 bytes.

5 21. Paquetes de Habilitación de Canal de Audio Directo

10 Este paquete permite al dispositivo central habilitar o inhabilitar canales de audio en el dispositivo de visualización. Esta capacidad es útil para que el dispositivo de visualización (cliente) pueda desconectar los amplificadores de audio o elementos de circuito similares para ahorrar potencia cuando el dispositivo central no transmite audio. Esto es mucho más difícil de implementar de manera implícita utilizando simplemente la presencia o ausencia de flujos de audio como un indicador. El estado por defecto cuando el sistema de visualización se enciende es que todos los canales de audio están habilitados. El formato de un Paquete de Habilitación de Canal de Audio Directo se muestra en la FIG. 30. Tal y como se muestra en la FIG. 30, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Máscara de Habilitación de Canal de Audio y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 78 en el campo de tipo de 1 byte y utiliza una longitud fija preseleccionada de 4 bytes.

15 22. Paquetes de Frecuencia de Muestreo de Audio Inverso

20 Este paquete permite al dispositivo central habilitar o inhabilitar el canal de audio de enlace inverso y fijar la frecuencia de muestreo de datos de audio de este flujo. El dispositivo central selecciona una frecuencia de muestreo que se define para ser válida en el Paquete de Capacidades de Dispositivo de Visualización. Si el dispositivo central selecciona una frecuencia de muestro no válida, entonces el dispositivo de visualización no enviará un flujo de audio al dispositivo central. El dispositivo central puede inhabilitar el flujo de audio de enlace inverso fijando la frecuencia de muestreo a 255. El estado por defecto asumido cuando el sistema de visualización se enciende o se conecta inicialmente es con el flujo de audio de enlace inverso inhabilitado. El formato de un Paquete de Frecuencia de Muestreo de Audio Inverso se muestra en la FIG. 31. Tal y como se muestra en la FIG. 31, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Frecuencia de Muestreo de Audio y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 79 y utiliza una longitud fija preseleccionada de 4 bytes.

25 23. Paquetes de Datos Suplementarios de Protección de Contenido Digital

30 Este paquete permite que el dispositivo central y el dispositivo de visualización intercambien mensajes relacionados con el procedimiento de protección de contenido digital que está usándose. Actualmente se contemplan dos tipos de protección de contenido, protección de contenido de transmisión digital (DTCP) o sistema de protección de contenido digital de gran ancho de banda (HDCP), con espacio reservado para futuras designaciones de esquemas de protección alternativos. El procedimiento que se utiliza está especificado por un parámetro Tipo de Protección de Contenido en este paquete. El formato de un Paquete de Datos Suplementarios de Protección de Contenido Digital se muestra en la FIG. 32. Tal y como se muestra en la FIG. 32, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Tipo de Protección de Contenido, Mensajes de Datos Suplementarios de Protección de Contenido y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 80.

35 24. Paquetes de Habilitación de Color Transparente

40 El Paquete de Habilitación de Color Transparente se utiliza para especificar qué color es transparente en un dispositivo de visualización y para habilitar o inhabilitar la utilización de un color transparente para visualizar imágenes. Los dispositivos de visualización que tienen esta capacidad notificarán esa capacidad en el bit 4 del campo Indicadores de Capacidad de Características de Dispositivo de Visualización del Paquete de Capacidades de Dispositivo de Visualización. Cuando un píxel con el valor para el color transparente se escribe en el mapa de bits, el color no cambia con respecto al valor anterior. El formato de un Paquete de Habilitación de Color Transparente se muestra en la FIG. 33. Tal y como se muestra en la FIG. 33, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, Habilitación de Color Transparente, Descriptor de Formato de Datos, Valor de Pixel Transparente y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 81 en el campo de tipo de 1 byte y utiliza una longitud fija preseleccionada de 10 bytes.

45 25. Paquetes de Medición de Retardo de Ida y Vuelta

50 El Paquete de Medición de Retardo de Ida y Vuelta se utiliza para medir el retardo de propagación desde el dispositivo central hasta un dispositivo cliente (de visualización) más el retardo desde el dispositivo cliente (de visualización) al dispositivo central. Esta medición incluye de manera intrínseca los retardos que existen en los excitadores y receptores de línea y en un subsistema de interconexión. Esta medición se utiliza para fijar el retardo de inversión y los parámetros de divisor de velocidad de enlace inverso en el Paquete de Encapsulación de Enlace Inverso, descrito de manera genérica anteriormente. Este paquete es muy útil cuando el enlace MDDI está funcionando a la máxima velocidad prevista para una aplicación particular. La señal MDDI_Stb se comporta como si estuvieran enviándose datos todos a cero durante los siguientes campos: Todo a Cero, Tiempos de Vigilancia y Periodo de Medición. Esto hace que la señal MDDI_Stb oscile a la mitad de la velocidad de transferencia de datos para que pueda utilizarse como un reloj periódico en

el dispositivo de visualización durante el Periodo de Medición.

El formato de un Paquete de Medición de Retardo de Ida y Vuelta se muestra en la FIG. 34. Tal y como se muestra en la FIG. 34, este tipo de paquete está estructurado para tener los campos Longitud de Paquete, Tipo de Paquete, CRC de Parámetros, Todo a Cero, Tiempo de Vigilancia 1, Periodo de Medición, Tiempo de Vigilancia 2 y Rehabilitación de Excitador. Este tipo de paquete está identificado generalmente como un paquete de Tipo 82 y utiliza una longitud fija preseleccionada de 533 bits.

La temporización de eventos que tienen lugar durante el Paquete de Medición de Retardo de Ida y Vuelta se ilustra en la FIG. 35. En la FIG. 35, el dispositivo central transmite el Paquete de Medición de Retardo de Ida y Vuelta, mostrado mediante la presencia de los campos CRC de Parámetros y Alineación Estroboscópica seguidos por los campos Todo a Cero y Tiempo de Vigilancia 1. Un retardo 3502 se produce antes de que el paquete llegue al dispositivo de visualización cliente o sistema de circuitos de procesamiento. Cuando el dispositivo de visualización recibe el paquete, transmite el patrón 0xff, 0xff, 0x0 de manera precisa y práctica al principio del Periodo de Medición según determine el dispositivo de visualización. El tiempo real en que el dispositivo de visualización comienza a transmitir esta secuencia está retardado con respecto al inicio del Periodo de Medición desde el punto de vista del dispositivo central. La cantidad de este retardo es sustancialmente el tiempo que tarda el paquete en propagarse a través de los excitadores y receptores de línea y del sistema de interconexión. Una cantidad similar de retardo 3504 se produce cuando el patrón se propaga desde el dispositivo de visualización al dispositivo central.

Con el fin de determinar de manera precisa el tiempo de retardo de ida y vuelta para las señales que van y vienen del dispositivo cliente, el dispositivo central cuenta el número de periodos de tiempo de bit que se producen después del inicio del Periodo de Medición hasta que se detecta el comienzo de la secuencia 0xff, 0xff, 0x0 tras su llegada. Esta información se utiliza para determinar la cantidad de tiempo que tarda una señal de ida y vuelta en ir desde el dispositivo central hasta el dispositivo cliente y volver. Después, la mitad aproximadamente de esta cantidad se atribuye a un retardo creado para el paso unidireccional de una señal al dispositivo cliente.

El dispositivo de visualización inhabilita sus excitadores de línea casi justo después de enviar el último bit del patrón 0xff, 0xff, 0x0. El Tiempo de Guardia 2 da tiempo a que los excitadores de línea del dispositivo de visualización pasen completamente al estado de alta impedancia antes de que el dispositivo central transmita la Longitud de Paquete del siguiente paquete. Las resistencias de polarización de hibernación a nivel lógico alto y a nivel lógico bajo (véase la FIG. 42) garantizan que las señales MDDI_Data se mantengan en un nivel bajo válido en los intervalos en los que los excitadores de línea están inhabilitados tanto en el dispositivo central como en el dispositivo de visualización.

26. Paquete de Calibración de Desajustes de Enlace Directo

El Paquete de Calibración de Desajustes de Enlace Directo permite a un dispositivo cliente o de visualización calibrarse para las diferencias en el retardo de propagación de las señales MDDI_Data con respecto a la señal MDDI_Stb. Sin compensación del retardo diferencial, la velocidad de transferencia de datos máxima está limitada generalmente para dar cuenta del peor caso de variación posible en estos retardos. Generalmente, este paquete solo se envía cuando la velocidad de transferencia de datos del enlace directo está configurada a una velocidad de 50 Mbps aproximadamente o inferior. Después de enviar este paquete para calibrar el dispositivo de visualización, la velocidad de transferencia de datos puede aumentar por encima de los 50 Mbps. Si la velocidad de transferencia de datos se fija demasiado alta durante el proceso de calibración de desajustes, el dispositivo de visualización puede sincronizarse con un alias del periodo de bit, lo que podría provocar que el ajuste de compensación de retardo diferencial estuviera inactivo durante más del tiempo de un bit, dando como resultado una sincronización de datos errónea. El tipo de interfaz con la velocidad de transferencia de datos más alta o el mayor tipo de interfaz posible se selecciona antes de enviar el Paquete de Calibración de Desajustes de Enlace Directo para que todos los bits de datos existentes estén calibrados.

El formato de un Paquete de Calibración de Desajustes de Enlace Directo se muestra en la FIG. 56. Tal y como se muestra en la FIG. 56, este tipo de paquete está estructurado para tener los campos Longitud de Paquete (2 bytes), Tipo de Paquete, CRC de Parámetros, Secuencia de Datos de Calibración y CRC. Este tipo de paquete está identificado generalmente como un paquete de Tipo 83 en el campo de tipo y tiene una longitud preseleccionada de 515.

D. CRC de Paquete

Los campos CRC aparecen al final de los paquetes y algunas veces después de determinados parámetros más críticos en un paquete que puede tener un campo de datos significativamente grande y, por lo tanto, una mayor probabilidad de errores durante su transferencia. En paquetes que tienen dos campos CRC, el generador CRC, cuando solo se utiliza uno, se reinicializa después de la primera CRC de manera que los cálculos CRC siguientes a un campo de datos largo no se ven afectados por los parámetros del principio del paquete.

En una realización a modo de ejemplo, el polinomio utilizado para el cálculo CRC se conoce como el CRC-16, o $X^{16} + X^{15} + X^2 + X^0$. Una implementación sencilla de un generador y verificador CRC 3600 útil para implementar la invención se muestra en la FIG. 36. En la FIG. 36, un registro CRC 3602 se inicializa a un valor de 0x0001 justo antes de transferir el primer bit de un paquete que se introduce en la línea Tx_MDDI_Data_Antes_CRC; después, los bytes del

paquete se desplazan dentro del registro empezando por el primer LSB. Debe observarse que los números de bit de registro en esta figura corresponden al orden del polinomio que está utilizándose y no a las posiciones de bit utilizadas por la MDDI. Es más eficaz desplazar el registro CRC en un único sentido, y esto da como resultado que el bit 15 de la CRC aparezca en la posición de bit 0 del campo CRC MDDI, y el bit de registro 14 de la CRC en la posición de bit 1 del campo CRC MDDI, y así sucesivamente, hasta que se llegue a la posición de bit 14 de la MDDI.

Como un ejemplo, si los contenidos de paquete de los Paquetes de Estado y Solicitud de Dispositivo de Visualización son: 0x07, 0x46, 0x000400, 0x00 (o representados como una secuencia de bytes como: 0x07, 0x00, 0x46, 0x00, 0x04, 0x00, 0x00), y se envían utilizando las entradas de los multiplexores 3604 y 3606, y la puerta NAND 3608, la salida CRC resultante en la línea Tx_MDDI_Data_Con_CRC es 0x0ea1 (o representada como una secuencia como 0xa1, 0x0e).

Cuando el generador y verificador CRC 3600 está configurado como un verificador CRC, la CRC que se recibe en la línea Rx_MDDI_Data se introduce en el multiplexor 3604 y en la puerta NAND 3608, y se compara bit a bit con el valor encontrado en el registro CRC utilizando la puerta NOR 3610, la puerta OR-exclusiva (XOR) 3612 y la puerta AND 3614. Si hay algún error, proporcionado por la puerta AND 3614, la CRC se incrementa una vez para cada paquete que contenga un error CRC conectando la salida de la puerta 3614 a la entrada del registro 3602. Debe observarse que el circuito de ejemplo mostrado en el diagrama de la FIG. 36 puede proporcionar más de una señal de error CRC dentro de una ventana COMPROBAR_CRC_AHORA dada (véase la FIG. 37B). Por lo tanto, el contador de errores CRC solo cuenta generalmente el primer caso de error CRC dentro de cada intervalo cuando COMPROBAR_CRC_AHORA está activa. Si está configurado como un generador CRC, la CRC se sincroniza a partir del registro CRC en el momento que coincide con el final del paquete.

La temporización para las señales de entrada y de salida, y para las señales de habilitación, se ilustra gráficamente en las FIG. 37A y 37B. La generación de una CRC y la transmisión de un paquete de datos se muestran en la FIG. 37A con el estado (0 ó 1) de las señales Gen_Reajuste, Comprobar_CRC_Ahora, Generar_CRC_Ahora y Enviar_MDDI_Data, junto con las señales Tx_MDDI_Data_Antes_CRC y Tx_MDDI_Data_Con_CRC. La recepción de un paquete de datos y la comprobación del valor CRC se muestran en la FIG. 37B, con el estado de las señales Gen_Reajuste, Comprobar_CRC_Ahora, Generar_CRC_Ahora y Enviar_MDDI_Data, junto con las señales Rx_MDDI_Data y de error CRC.

E. Sobrecarga de Códigos de Error para CRC de Paquetes

Cuando se transfieren solamente paquetes de datos y la CRC entre el dispositivo central y el dispositivo cliente, no se incluyen códigos de error. El único error es una pérdida de sincronización. En caso contrario, hay que esperar a que el enlace salga de un periodo en que la trayectoria o línea de transferencia de datos no es óptima y después reajustar el enlace y continuar. Desafortunadamente, esto es muy lento y en cierto modo ineficiente.

Para utilizarse en una realización, se ha desarrollado una nueva técnica en la que la parte CRC de los paquetes se utiliza para transferir información de códigos de error. Esto se muestra de manera genérica en el esquema de la FIG. 65 y de manera más detalla en las FIG. 66A, 66B y 67. Es decir, uno o más códigos de error se generan por los procesadores o dispositivos que se encargan de la transferencia de datos, que indican errores o fallos predefinidos específicos que pueden producirse en el procesamiento o enlace de comunicaciones. Cuando se encuentra un error, se genera el código de error apropiado y se transfiere utilizando los bits para la CRC de un paquete. Es decir, el valor CRC se sobrecarga, o se sobrescribe, con el código de error deseado, o con algún otro valor preseleccionado para representar la presencia de un error, que puede detectarse en el extremo receptor mediante un dispositivo de supervisión o verificador de errores que supervisa los valores del campo CRC. Para casos en los que el código de error coincide con el valor CRC por algún motivo, se transfiere el complemento del error para evitar confusiones.

En una realización, para proporcionar un sistema robusto de detección y de aviso de errores, el código de error puede transferirse varias veces, utilizando una serie de paquetes, generalmente todos, que se transfieren o se envían después de que se haya detectado el error. Esto se produce hasta el momento en que la condición que crea el error se elimine del sistema, momento en el que los bits CRC regulares se transfieren sin sobrecargarse por otro valor.

Esta técnica de sobrecargar el valor CRC proporciona una respuesta mucho más rápida a los errores del sistema utilizando al mismo tiempo una cantidad mínima de bits o campos adicionales.

Tal y como se muestra en la FIG. 66, un mecanismo o aparato de sobrescritura CRC 6600 se muestra utilizando un detector o un medio de detección de errores 6602, que puede formar parte de otro sistema de circuitos conocido o descrito anteriormente, que detecta la presencia o la existencia de errores en el enlace o proceso de comunicaciones. Un generador o medio de generación de códigos de error 6604, que puede formarse como parte de otro sistema de circuitos o utilizar técnicas tales como tablas de consulta para almacenar mensajes de error preseleccionados, genera uno o más códigos de error para indicar errores o fallos predefinidos específicos cuya presencia se ha detectado. Se entenderá fácilmente que los dispositivos 6602 y 6604 pueden formarse como un único circuito o dispositivo, según se desee, o como parte de una secuencia programada de etapas para otros procesadores y elementos conocidos.

Se conoce un comparador o medio de comparación de valores CRC 6606 para comprobar si el código o códigos de error seleccionados son el mismo que el valor CRC que está transfiriéndose. Si ese es el caso, entonces un generador o medio o dispositivo de generación de complementos de código se utiliza para proporcionar el complemento de los códigos de error para no confundirlos con el patrón o valor CRC original y para no hacer más difícil o complicar el esquema de detección. Después, un medio, elemento o dispositivo de selección o selector de códigos de error 6610 selecciona el código o valor de error que se desea insertar o sobrescribir, o sus complementos respectivos, según sea apropiado. Un medio, mecanismo o dispositivo de sobrescritura CRC de códigos de error 6612 es un dispositivo que recibe el flujo de datos, los paquetes y los códigos deseados que van a insertarse y sobrescribe los valores CRC correspondientes o apropiados con el fin de transferir los códigos de error deseados a un dispositivo receptor.

Tal y como se ha mencionado, el código de error puede transferirse varias veces, utilizando una serie de paquetes, de manera que dispositivo de sobrescritura 6612 puede utilizar elementos de almacenamiento en memoria con el fin de mantener copias de los códigos durante el procesamiento o recuperar estos códigos a partir de elementos anteriores u otras ubicaciones de almacenamiento conocidas que pueden utilizarse para almacenar o guardar sus valores, según se necesite o se desee.

El procesamiento general del mecanismo de sobrescritura de la FIG. 66 se implementa y se muestra en mayor detalle en las FIG. 67A y 67B. En la FIG. 67A, uno o más errores se detectan en la etapa 6702 en los datos o en el proceso de comunicaciones y un código de error se selecciona en la etapa 6704 para indicar esta condición. Al mismo tiempo, o en un momento adecuado, el valor CRC que va a sustituirse se comprueba en la etapa 6706 y se compara con el código de error deseado en la etapa 6708. El resultado de esta comparación, tal y como se ha indicado anteriormente, es una determinación de si el código deseado, u otros valores representativos, es igual o no al actual valor CRC. Si es así, entonces el procesamiento avanza hasta la etapa 6712 donde el complemento, o en algunos casos otro valor representativo, según se desee, se selecciona como el código a insertar. Una vez que se haya determinado qué códigos o valores de error van a insertarse en las etapas 6710 y 6714, ese código apropiado se selecciona para su inserción. Estas etapas se ilustran por separado para una mayor claridad, pero generalmente representan una única opción basada en la salida de la etapa 6708 de decisión. Finalmente, en la etapa 6716, los valores apropiados se sobrescriben en la posición CRC para transferirse con los paquetes determinados por el proceso.

En el lado de recepción de paquetes, tal y como se muestra en la FIG. 67B, los valores CRC de paquete se supervisan en la etapa 6722. Generalmente, los valores CRC se supervisan mediante uno o más procesos del sistema para determinar si se ha producido un error en la transferencia de datos y si hay que solicitar o no una retransmisión del paquete o paquetes, o inhabilitar operaciones adicionales, etc., algunas de las cuales se han descrito anteriormente. Como parte de tal supervisión, la información también puede utilizarse para comparar los valores con códigos de error conocidos o preseleccionados, o con valores representativos, y para detectar la presencia de errores. Como alternativa, puede implementarse un dispositivo de supervisión y un proceso de detección de errores aparte. Si un código está presente, se extrae o se indica de otro modo en la etapa 6724 para un procesamiento adicional. En la etapa 6726 puede determinarse si es el código real o un complemento, en cuyo caso se utiliza una etapa adicional 6728 para convertir el valor al valor de código deseado. En cualquier caso, el código extraído resultante, complemento u otros valores recuperados se utilizan posteriormente para detectar el error que se ha producido a partir del código transmitido en la etapa 6730.

V. Reinicio de enlace desde el estado de hibernación

Cuando el dispositivo central reinicia el enlace directo desde un estado de hibernación, excita la señal MDDI_Data a un estado de uno lógico durante 150 μ seg aproximadamente y después activa la señal MDDI_Stb y excita simultáneamente la señal MDDI_Data a un estado de cero lógico durante 50 μ seg, y después inicia el tráfico de enlace directo enviando un paquete de cabecera de subtrama. Generalmente, esto permite resolver las contiendas de bus antes de enviar el paquete de cabecera de subtrama proporcionando un tiempo de asentamiento suficiente entre las señales.

Cuando el dispositivo cliente, en este caso un dispositivo de visualización, necesita datos o comunicaciones desde el dispositivo central, excita la línea MDDI_Data0 a un estado de uno lógico durante 70 μ seg aproximadamente, aunque pueden utilizarse otros periodos según se desee, y después inhabilita el excitador llevándolo a un estado de alta impedancia. Esta acción hace que dispositivo central inicie o reinicie el tráfico de datos en el enlace directo (208) y que interroge al dispositivo cliente con relación a su estado. El dispositivo central debe detectar la presencia del impulso de solicitud en un tiempo límite de 50 μ seg y después iniciar la secuencia de activación para excitar la señal MDDI_Data0 a un uno lógico durante 150 μ seg y a un cero lógico durante 50 μ seg. El dispositivo de visualización no debe enviar un impulso de solicitud de servicio si detecta la línea MDDI_Data0 en el estado de uno lógico durante más de 50 μ seg. La naturaleza de la selección de los tiempos y tolerancias de los intervalos de tiempo relacionados con el proceso de hibernación y la secuencia de activación se describe posteriormente en mayor detalle.

Un ejemplo de las etapas de procesamiento para un evento de solicitud de servicio típico 3800 sin contienda se ilustra en la FIG. 38, donde los eventos están etiquetados para una mayor claridad de ilustración utilizando las letras A, B, C, D, E, F y G. El proceso comienza en el punto A, cuando el dispositivo central envía un Paquete de Interrupción de Enlace al dispositivo cliente para informarle de que el enlace pasará a un estado de hibernación de baja potencia. En una

etapa siguiente, el dispositivo central entra en el estado de hibernación de baja potencia inhabilitando el excitador de señal MDDI_Data0 y fijando el excitador de señal MDDI_Stb a un cero lógico, tal y como se muestra en el punto B. La señal MDDI_Data0 se excita a un nivel de cero mediante una red de polarización de alta impedancia. Después de un periodo de tiempo, el cliente envía un impulso de solicitud de servicio al dispositivo central excitando la señal MDDI_Data0 a un nivel de uno lógico, tal y como se observa en el punto C. El dispositivo central mantiene el nivel de cero utilizando la red de polarización de alta impedancia, pero el excitador del dispositivo cliente hace que la línea pase a un nivel de uno lógico. En un tiempo límite de 50 μ seg, el dispositivo central reconoce el impulso de solicitud de servicio y establece un nivel de uno lógico en la señal R4DDI_Data0 habilitando su excitador, tal y como puede observarse en el punto D. Después, el dispositivo cliente deja de intentar establecer el impulso de solicitud de servicio y hace pasar su excitador a un estado de alta impedancia, tal y como se observa en el punto E. El dispositivo central excita la señal MDDI_Data0 a un nivel de cero lógico durante 50 μ seg, tal y como se muestra en el punto F, y también empieza a generar la señal MDDI_Stb de una manera compatible con el nivel de cero lógico de la señal NMDI_Data0. Después de establecer la señal MDDI_Data0 en un nivel de cero y excitar la señal MDDI_Stb durante 50 μ seg, el dispositivo central comienza a transmitir datos en el enlace directo enviando un Paquete de Cabecera de Subtrama, tal y como se muestra en el punto G.

Un ejemplo similar se ilustra en la FIG. 39, donde una solicitud de servicio se establece después de que se haya iniciado la secuencia de reinicio de enlace, y los eventos están etiquetados nuevamente utilizando las letras A, B, C, D, E, F y G. Este ejemplo representa el peor caso de escenario, donde un impulso o señal de solicitud del dispositivo cliente llega casi a corromper el Paquete de Cabecera de Subtrama. El proceso comienza en el punto A, cuando el dispositivo central envía nuevamente un Paquete de Interrupción de Enlace al dispositivo cliente para informarle de que el enlace pasará a un estado de hibernación de baja potencia. En una etapa siguiente, el dispositivo central entra en el estado de hibernación de baja potencia inhabilitando el excitador de señal MDDI_Data0 y fijando el excitador de señal MDDI_Stb a un cero lógico, tal y como se muestra en el punto B. Como antes, la señal NMDI_Data0 se excita a un nivel de cero mediante una red de polarización de alta impedancia. Después de un periodo de tiempo, el dispositivo central inicia la secuencia de reinicio de enlace excitando la señal MDDI_Data0 a un nivel de uno lógico durante 150 μ seg, tal y como se observa en el punto C. Antes de que pasen 50 μ seg después de que comience la secuencia de reinicio de enlace, el dispositivo de visualización también establece la señal MDDI_Data0 durante 70 μ seg, tal y como se observa en el punto D. Esto sucede porque el dispositivo de visualización necesita solicitar un servicio del dispositivo central y no reconoce que el dispositivo central ya ha iniciado la secuencia de reinicio de enlace. Después, el dispositivo cliente deja de intentar establecer el impulso de solicitud de servicio y hace pasar su excitador a un estado de alta impedancia, tal y como se observa en el punto E. El dispositivo central sigue excitando la señal MDDI_Data0 a un nivel de uno lógico. El dispositivo central excita la señal MDDI_Data0 a un nivel de cero lógico durante 50 μ seg, tal y como se muestra en el punto F, y también empieza a generar la señal MDDI_Stb de una manera compatible con el nivel de cero lógico de la señal MDDI_Data0. Después de establecer la señal MDDI_Data0 en un nivel de cero y excitar la señal MDDI_Stb durante 50 μ seg, el dispositivo central comienza a transmitir datos en el enlace directo enviando un Paquete de Cabecera de Subtrama, tal y como se muestra en el punto G.

A partir del análisis anterior, puede observarse que la solución anterior requiere que el dispositivo central pase por dos estados como parte de una secuencia de activación. Para el primer estado, el dispositivo central excita la señal MDDI_Data0 a un estado alto durante 150 μ seg y después excita la señal MDDI_Data0 a un estado bajo durante 50 μ seg activando al mismo tiempo la línea MDDI_Stb, y después comienza a transmitir paquetes MDDI. Este proceso es adecuado para hacer avanzar el estado de la técnica en lo que respecta a las velocidades de transferencia de datos que pueden conseguirse utilizando el aparato y los procedimientos MDDI. Sin embargo, tal y como se ha indicado anteriormente, siempre se requiere más velocidad en lo que se refiere a un menor tiempo de respuesta a las condiciones, o la capacidad de seleccionar más rápidamente la siguiente etapa o proceso, o la capacidad de simplificar el procesamiento o los elementos.

Los solicitantes han descubierto un nuevo enfoque inventivo para activar el procesamiento y la temporización, en donde el dispositivo central utiliza una temporización basada en ciclos de reloj para la oscilación de las señales. En esta configuración, el dispositivo central comienza a hacer oscilar la señal MDDI_Stb entre 0 y 10 μ seg después de que el dispositivo central haya excitado la señal MDDI_Data0 a un estado alto al principio de la secuencia de activación, y no espera a que la señal se excite a un estado bajo. Durante una secuencia de activación, el dispositivo central hace oscilar la señal MDDI_Stb aunque la señal MDDI_Data0 esté siempre a un nivel de cero lógico. Esto elimina de manera eficaz el concepto de tiempo desde el lado del dispositivo cliente, y el dispositivo central pasa de los periodos anteriores de 150 μ seg y 50 μ seg para los dos primeros estados, a 150 ciclos de reloj y a 50 ciclos de reloj para esos periodos.

El dispositivo central se hace ahora responsable de excitar esa línea de datos a un estado alto, y en un tiempo límite de 10 ciclos de reloj comienza a transmitir una señal estroboscópica como si la línea de datos estuviera a cero. Después de que el dispositivo central haya excitado la línea de datos a un estado alto durante 150 ciclos de reloj, el dispositivo central excita la línea de datos a un estado bajo durante 50 ciclos de reloj mientras sigue transmitiendo la señal estroboscópica. Después de que hayan completado ambos procesos, el dispositivo central puede empezar a transmitir el primer paquete de cabecera de subtrama.

En el lado del dispositivo cliente, la implementación del dispositivo cliente puede usar ahora el reloj generado para

5 calcular el número de ciclos de reloj en que la línea de datos está en un estado alto en primer lugar y después en un estado bajo. El número de ciclos de reloj que es necesario producir en la línea de datos excitada a un estado alto es de 150 y en línea de datos excitada a un estado bajo es de 50. Esto significa que para una secuencia de activación apropiada, el dispositivo cliente debe poder contar al menos 150 ciclos de reloj continuos de la línea de datos en un estado alto, seguidos de al menos 50 ciclos de reloj continuos de la línea de datos en un estado bajo. Una vez que se cumplan estas dos condiciones, el dispositivo cliente puede empezar a buscar la palabra única de la primera subtrama. Una interrupción en este patrón se utiliza como base para hacer que los contadores vuelvan a un estado inicial en el que el cliente busca de nuevo los 150 primeros ciclos de reloj continuos de la línea de datos en un estado alto.

10 Una implementación de dispositivo cliente de la invención para una activación basada en dispositivo cliente desde el estado de hibernación es muy similar al caso de activación inicial excepto que no es obligatorio que la velocidad del reloj empiece a 1 Mbps, como se ha descrito anteriormente. En cambio, la velocidad del reloj puede fijarse para que se reanude a la velocidad anterior que estuviera activa cuando el enlace de comunicaciones pasó al estado de hibernación. Si el dispositivo central inicia la transmisión de una señal estroboscópica como se ha descrito anteriormente, el dispositivo cliente debe poder contar nuevamente al menos 150 ciclos de reloj continuos de la línea de datos en un estado alto, seguidos de al menos 50 ciclos de reloj continuos de la línea de datos en un estado bajo. Una vez que se hayan cumplido estas dos condiciones, el cliente puede empezar a buscar la palabra única.

15 Una implementación de dispositivo cliente de la invención para una activación basada en dispositivo cliente desde el estado de hibernación es similar a la activación basada en dispositivo central excepto que comienza con el dispositivo cliente excitando la línea de datos. El cliente puede excitar de manera asíncrona la línea de datos sin un reloj para activar el dispositivo central. Una vez que el dispositivo central reconozca que el dispositivo cliente está excitando la línea de datos en estado alto, puede iniciar su secuencia de activación. El dispositivo cliente puede contar el número de ciclos de reloj generados por el dispositivo central al comenzar o durante su proceso de activación. Una vez que el dispositivo cliente haya contado 70 ciclos de reloj continuos de los datos en estado alto, puede dejar de excitar la línea de datos a un estado alto. En este momento, el dispositivo central ya debe estar excitando también la línea de datos a un estado alto. Después, el dispositivo cliente puede contar otros 80 ciclos de reloj continuos de la línea de datos en estado alto hasta alcanzar los 150 ciclos de reloj de la línea de datos en estado alto, y después puede buscar 50 ciclos de reloj de la línea de datos en estado bajo. Una vez que se hayan cumplido estas tres condiciones, el dispositivo cliente puede empezar a buscar la palabra única.

20 Una ventaja de esta nueva implementación de procesamiento de activación es que elimina la necesidad de un dispositivo de medición de tiempo. Ya sea un oscilador, un circuito de descarga de condensadores u otros dispositivos conocidos de este tipo, el dispositivo cliente ya no necesita tales dispositivos externos para determinar las condiciones de activación. Esto ahorra costes y espacio de circuito en la implementación de controladores, contadores, etc., en una tarjeta de dispositivo cliente. Aunque esto puede no ser tan ventajoso para el dispositivo cliente, para el dispositivo central esta técnica también debe simplificar potencialmente el dispositivo central en lo que respecta a una lógica de muy alta densidad (VHDL) que esté utilizándose para el sistema de circuitos núcleo. El consumo de potencia por utilizar las líneas estroboscópicas y de datos como la fuente de notificación y medición de activación también será menor puesto que no se necesita que ningún sistema de circuitos externo esté funcionando para que los elementos principales esperen una activación basada en dispositivo central.

25 El número de ciclos o de periodos de reloj utilizados son a modo de ejemplo y otros periodos pueden usarse tal y como resultará evidente para un experto en la técnica.

30 Una ventaja de esta nueva implementación de procesamiento de activación es que elimina la necesidad de un dispositivo de medición de tiempo. Ya sea un oscilador, un circuito de descarga de condensadores u otros dispositivos conocidos de este tipo, el dispositivo cliente ya no necesita tales dispositivos externos para determinar las condiciones de activación. Esto ahorra costes y espacio de circuito en la implementación de controladores, contadores, etc., en una tarjeta de dispositivo cliente. Aunque esto puede no ser tan ventajoso para el dispositivo cliente, para el dispositivo central esta técnica también debe simplificar potencialmente el dispositivo central en lo que respecta a una lógica de muy alta densidad (VHDL) que esté utilizándose para el sistema de circuitos núcleo. El consumo de potencia por utilizar las líneas estroboscópicas y de datos como la fuente de notificación y medición de activación también será menor puesto que no se necesita que ningún sistema de circuitos externo esté funcionando para que los elementos principales esperen una activación basada en dispositivo central.

35 Para aclarar e ilustrar el funcionamiento de esta nueva técnica, la temporización de las señales MDDI_Data0 y MDDI_Stb y varias operaciones relativas a los ciclos de reloj se muestran en las FIG. 68A, 68B y 68C.

40 Un ejemplo de las etapas de procesamiento para una activación típica iniciada por dispositivo central sin contienda se ilustra en la FIG. 68A, donde los eventos están etiquetados nuevamente para una mayor claridad de ilustración utilizando las letras A, B, C, D, E, F y G. El proceso comienza en el punto A, cuando el dispositivo central envía un Paquete de Interrupción de Enlace al dispositivo cliente para informarle de que el enlace va a pasar a un estado de hibernación de baja potencia. En una etapa siguiente, el punto B, el dispositivo central hace oscilar la señal MDDI_Stb

5 durante 64 ciclos aproximadamente (o según se desee para el diseño del sistema) para permitir que el procesamiento del dispositivo cliente se complete antes de detener la oscilación de la señal MDDI_Stb, lo que detiene el reloj recuperado en el dispositivo cliente. El dispositivo central también fija inicialmente la señal MDDI_Data0 a un nivel de cero lógico y después inhabilita la salida MDDI_Data0 en el intervalo de 16 a 48 ciclos (incluyendo generalmente retardos de propagación de inhabilitación de salida) después de la CRC. Puede ser deseable poner receptores de alta velocidad para las señales MDDI_Data0 y MDDI_Stb en el dispositivo cliente en un estado de baja potencia algún tiempo después de los 48 ciclos, después de la CRC y antes de la siguiente etapa (C).

10 El dispositivo central entra en el estado de hibernación de baja potencia en el punto o etapa C, inhabilitando los excitadores de las señales MDDI_Data0 y MDDI_Stb y haciendo que un controlador de dispositivo central pase a un estado de hibernación de baja potencia. También puede fijarse el excitador de señal MDDI_Stb a un nivel de cero lógico (utilizando una red de polarización de alta impedancia) o seguir con la oscilación durante la hibernación, según se desee. El dispositivo cliente también está en un estado de hibernación de bajo nivel de potencia.

15 Después de un periodo de tiempo, el dispositivo central comienza la secuencia de reinicio de enlace en el punto D, habilitando la salida de excitador de las señales MDDI_Data0 y MDDI_Stb. El dispositivo central excita la señal MDDI_Data0 a un nivel de uno lógico y la señal MDDI_Stb a un nivel de cero lógico durante el tiempo que tardan los excitadores en habilitar completamente sus respectivas salidas. El dispositivo central espera normalmente 200 nanosegundos aproximadamente después de que estas salidas hayan alcanzado los niveles lógicos deseados antes de excitar los impulsos de la señal MDDI_Stb. Esto da tiempo a que el dispositivo cliente se prepare para la recepción.

20 Con los excitadores de dispositivo central habilitados y la señal MDDI_Data0 excitándose a un nivel de uno lógico, el dispositivo central empieza a hacer oscilar la señal MDDI_Stb durante 150 ciclos de señal MDDI_Stb, tal y como se observa en el punto E. El dispositivo central excita la señal MDDI_Data0 a un nivel de cero lógico durante 50 ciclos, tal y como se muestra en el punto F, y el dispositivo cliente empieza a buscar el Paquete de Cabecera de Subtrama después de que la señal MDDI_Data0 haya estado en un nivel de cero lógico durante 40 ciclos de señal MDDI_Stb. El dispositivo central comienza a transmitir datos en el enlace directo enviando un Paquete de Cabecera de Subtrama, tal y como se muestra en el punto G.

25 Un ejemplo de las etapas de procesamiento de una activación típica iniciada por dispositivo cliente sin contienda se ilustra en la FIG. 68B, donde los eventos están etiquetados nuevamente para una mayor claridad de ilustración utilizando las letras A, B, C, D, E, F, G, H e I. Como antes, el proceso comienza en el punto A, cuando el dispositivo central envía un Paquete de Interrupción de Enlace para informar al dispositivo cliente de que el enlace va a pasar al estado de baja potencia.

30 En el punto B, el dispositivo central hace oscilar la señal MDDI_Stb durante 64 ciclos aproximadamente (o según se desee para el diseño del sistema) para permitir que el procesamiento del dispositivo cliente se complete antes de detener la oscilación de la señal MDDI_Stb, lo que detiene el reloj recuperado en el dispositivo cliente. El dispositivo central también fija inicialmente la señal MDDI_Data0 a un nivel de cero lógico y después inhabilita la salida MDDI_Data0 en el intervalo de 16 a 48 ciclos (incluyendo generalmente retardos de propagación de inhabilitación de salida) después de la CRC. Puede ser deseable poner receptores de alta velocidad para las señales MDDI_Data0 y MDDI_Stb en el dispositivo cliente en un estado de baja potencia algún tiempo después de los 48 ciclos, después de la CRC y antes de la siguiente etapa (C).

35 El dispositivo central entra en el estado de hibernación de baja potencia en el punto o etapa C, inhabilitando los excitadores de las señales MDDI_Data0 y MDDI_Stb y haciendo que un controlador de dispositivo central pase a un estado de hibernación de baja potencia. También puede fijarse el excitador de línea MDDI_Stb a un nivel de cero lógico (utilizando una red de polarización de alta impedancia) o seguir con la oscilación durante la hibernación, según se desee. El dispositivo cliente también está en un estado de hibernación de bajo nivel de potencia.

40 Después de un periodo de tiempo, el dispositivo cliente comienza la secuencia de reinicio de enlace en el punto D, habilitando el receptor de señal MDDI_Stb y también habilitando un desfase en el receptor de señal MDDI_Stb para garantizar que el estado de la versión recibida de MDDI_Stb está en un nivel de cero lógico en el dispositivo cliente antes de que dispositivo central habilite su excitador de señal MDDI_Stb. Puede ser deseable que el dispositivo cliente habilite el desfase poco antes de habilitar el receptor para garantizar la recepción de una señal diferencial válida e inhibir señales erróneas, según se desee. El dispositivo cliente habilita el excitador de señal MDDI_Data0 excitando al mismo tiempo la línea de MDDI_Data0 a un nivel de uno lógico.

45 En un tiempo límite de 1 mseg aproximadamente, punto E, el dispositivo central reconoce el impulso de solicitud de servicio del dispositivo cliente, y el dispositivo central inicia la secuencia de reinicio de enlace habilitando las salidas de excitador de las señales MDDI_Data0 y MDDI_Stb. El dispositivo central excita la señal MDDI_Data0 a un nivel de uno lógico y la señal MDDI_Stb a un nivel de cero lógico durante el tiempo que tardan los excitadores en habilitar sus respectivas salidas. El dispositivo central espera normalmente 200 nanosegundos aproximadamente después de que estas salidas alcancen los niveles lógicos deseados antes de excitar los impulsos de la señal MDDI_Stb. Esto da tiempo a que el dispositivo cliente se prepare para la recepción.

Con los excitadores de dispositivo central habilitados y la señal MDDI_Data0 excitándose a un nivel de uno lógico, el dispositivo central empieza a transmitir impulsos en la señal MDDI_Stb durante 150 ciclos de señal MDDI_Stb, tal y como se observa en el punto F. Cuando el dispositivo cliente reconoce el primer impulso de la señal MDDI_Stb, inhabilita el desfase en su receptor de señal MDDI_Stb. El dispositivo cliente sigue excitando la señal MDDI_Data0 a un nivel de uno lógico durante 70 ciclos de señal MDDI_Stb e inhabilita su excitador de señal MDDI_Data0 en el punto G.

Tal y como se observa en los puntos G y H, el dispositivo central excita la señal MDDI_Data0 a un nivel de cero lógico durante 50 ciclos, y el dispositivo cliente empieza a buscar el Paquete de Cabecera de Subtrama después de que la señal MDDI_Data0 haya estado en un nivel de cero lógico durante 40 ciclos de señal MDDI_Stb. El dispositivo central empieza a transmitir datos en el enlace directo enviando un Paquete de Cabecera de Subtrama, tal y como se muestra en el punto I.

Un ejemplo de las etapas de procesamiento para una activación típica iniciada por dispositivo central con contienda del dispositivo cliente, es decir, el dispositivo cliente también desea activar el enlace, se ilustra en la FIG. 68C. Los eventos están etiquetados de nuevo para una mayor claridad de ilustración utilizando las letras A, B, C, D, E, F, G, H e I. Como antes, el proceso comienza en el punto A, cuando el dispositivo central envía un Paquete de Interrupción de Enlace para informar al dispositivo cliente de que el enlace va a pasar al estado de baja potencia, continúa con el punto B donde la señal MDDI_Stb se hace oscilar durante 64 ciclos aproximadamente (o según se desee para el diseño del sistema) para permitir completar el procesamiento del dispositivo cliente, y después con el punto C donde el dispositivo central entra en el estado de hibernación de baja potencia inhabilitando los excitadores de las señales MDDI_Data0 y MDDI_Stb y haciendo que un controlador de dispositivo central pase a un estado de hibernación de baja potencia. Después de un periodo de tiempo, el dispositivo central comienza la secuencia de reinicio de enlace en el punto D habilitando la salida de excitador de las señales MDDI_Data0 y MDDI_Stb y empieza a hacer oscilar la señal MDDI_Stb durante 150 ciclos de señal MDDI_Stb, tal y como se observa en el punto E.

A los 70 ciclos de señal MDDI_Stb después del punto E, en este caso el punto F, el dispositivo cliente todavía no ha reconocido que el dispositivo central está excitando la señal MDDI_Data0 a un nivel de uno lógico, por lo que el dispositivo cliente también excita la señal MDDI_Data0 a un nivel de uno lógico. Esto sucede porque el dispositivo cliente desea solicitar un servicio pero no reconoce que el dispositivo central está intentando comunicarse y que ya ha iniciado la secuencia de reinicio de enlace. En el punto G, el dispositivo cliente deja de excitar la señal MDDI_Data0 y lleva a su excitador a un estado de alta impedancia inhabilitando su salida. El dispositivo central sigue excitando la señal MDDI_Data0 a un nivel de uno lógico durante 80 ciclos adicionales.

El dispositivo central excita la señal MDDI_Data0 a un nivel de cero lógico durante 50 ciclos, tal y como se muestra en el punto H, y el dispositivo cliente empieza a buscar el Paquete de Cabecera de Subtrama después de que la señal MDDI_Data0 haya estado en un nivel de cero lógico durante 40 ciclos de señal MDDI_Stb. El dispositivo central empieza a transmitir datos en el enlace directo enviando un Paquete de Cabecera de Subtrama, tal y como se muestra en el punto I.

VI. Especificaciones eléctricas de interfaz

En las realizaciones de ejemplo, los datos en un formato sin retorno a cero (NRZ) se codifican utilizando una señal de datos-estroboscópica o formato DATA-STB, que permite incluir información de reloj en los datos y en las señales estroboscópicas. El reloj puede recuperarse sin un sistema de circuitos complejo de bucle de enganche de fase. Los datos se transportan a través de un enlace diferencial bidireccional que se implementa generalmente utilizando un cable de línea alámbrica, aunque pueden utilizarse otros conductores, hilos impresos o elementos de transferencia, tal y como se ha indicado anteriormente. La señal estroboscópica (STB) se transporta a través de un enlace unidireccional que se excita solamente por el dispositivo central. La señal estroboscópica hace oscilar su valor (0 ó 1) cuando hay un estado continuo, 0 ó 1, que no cambia en la línea o señal de datos.

Un ejemplo de cómo puede transmitirse una secuencia de datos, tales como los bits "1110001011", utilizando la codificación DATA-STB se muestra de manera gráfica en la FIG. 40. En la FIG. 40, una señal DATA 4002 se muestra en la línea superior de un cronograma de señales y una señal STB 4004 se muestra en una segunda línea, alineadas cada vez según sea apropiado (punto inicial común). A medida que transcurre el tiempo, cuando hay un cambio de estado que se produce en la línea (señal) DATA 4002, entonces la línea (señal) STB 4004 mantiene un estado anterior, por lo que el primer estado '1' de la señal DATA se correlaciona con el primer estado '0' de la señal STB, su valor inicial. Sin embargo, si o cuando el nivel de estado de la señal DATA no cambia, entonces la señal STB oscila al estado opuesto o '1' en el presente ejemplo, como es el caso en la FIG. 40 donde la señal DATA proporciona otro valor '1'. Es decir, solo hay una única transición por ciclo de bit entre la señal DATA y la señal STB. Por lo tanto, la señal STB cambia de nuevo, esta vez a '0' cuando la señal DATA está a '1' y permanece en este nivel o valor cuando la señal DATA cambia el nivel a '0'. Cuando la señal DATA está a '1', la señal STB oscila al estado opuesto o '1' en el presente ejemplo, y así sucesivamente, cuando la señal DATA cambia o mantiene los niveles o valores.

Tras recibirse estas señales se lleva a cabo una operación OR-exclusiva (XOR) en las señales DATA y STB para generar una señal de reloj 4006, que se muestra en la parte inferior del cronograma para una comparación relativa con la

señal de datos y con la señal estroboscópica deseadas. Un ejemplo de sistema de circuitos útil para generar las salidas o señales DATA y STB a partir de datos de entrada en el dispositivo central y después recuperar o recapturar los datos de las señales DATA y STB en el dispositivo cliente se muestra en la FIG. 41.

5 En la FIG. 41, una parte de transmisión 4100 se utiliza para generar y transmitir las señales DATA y STB originales a través de una trayectoria de señales intermedia 4102, mientras que una parte de recepción 4120 se utiliza para recibir las señales y recuperar los datos. Tal y como se muestra en la FIG. 41, con el fin de transferir datos desde un dispositivo central a un dispositivo cliente, la señal DATA se introduce en dos elementos de circuito biestables de tipo D 4104 y 4106 junto con una señal de reloj para activar los circuitos. Después, las dos salidas de circuito biestable (Q) se dividen en un par diferencial de señales MDDI_Data0+, MDDI_Data0- y MDDI_Stb+, MDDI_Stb-, respectivamente, 10 utilizando dos excitadores de línea diferenciales 4108 y 4110 (modo de tensión). Un elemento lógico, circuito o puerta NOR-exclusiva (XNOR) de tres entradas 4112 está conectado para recibir la señal DATA y las salidas de ambos biestables, y genera una salida que proporciona la entrada de datos al segundo biestable, que a su vez genera las señales MDDI_Stb+ y MDDI_Stb-. De manera conveniente, la puerta XNOR tiene la burbuja de inversión colocada para indicar que está invirtiendo de manera eficaz la salida Q del biestable que genera la señal estroboscópica.

15 En la parte de recepción 4120 de la FIG. 41, las señales MDDI_Data0+, MDDI_Data0- y MDDI_Stb+, MDDI_Stb-, se reciben mediante dos receptores de línea diferenciales 4122 y 4124 que generan salidas únicas a partir de las señales diferenciales. Después, las salidas de los amplificadores se introducen en cada una de las entradas de un elemento lógico, circuito o puerta OR-exclusiva (XOR) de dos entradas 4126 que genera la señal de reloj. La señal de reloj se utiliza para activar cada uno de los dos circuitos biestables de tipo D 4128 y 4130 que reciben una versión retardada de la señal DATA, a través del elemento de retardo 4132, donde uno de los cuales (4128) genera valores '0' de datos y el otro (4130) valores '1' de datos. El reloj tiene además una salida independiente con respecto a la lógica XOR. Puesto que la información de reloj está distribuida entre las líneas DATA y STB, ninguna señal conmuta entre estados más rápido que la mitad de la velocidad de reloj. Puesto que el reloj se reproduce utilizando el procesamiento OR-exclusiva de las señales DATA y STB, el sistema tolera de manera eficaz el doble de la cantidad de desajuste entre los datos de entrada y el reloj en comparación con la situación en la que una señal de reloj se envía directamente a través de una única línea de datos dedicada.

20 Los pares de datos MDDI y las señales MDDI_Stb+ y MDDI_Stb- funcionan en un modo diferencial para maximizar la inmunidad frente a los efectos negativos del ruido. Cada parte de la trayectoria de señales diferenciales termina en una fuente, utilizándose la mitad de la impedancia característica del cable o conductor para transferir señales. Los pares de datos MDDI terminan en una fuente en los extremos tanto del dispositivo central como del dispositivo cliente. Puesto que solo uno de estos dos excitadores está activo en un momento dado, en la fuente existe de manera continua una terminación para el enlace de transferencia. Las señales MDDI_Stb+ y MDDI_Stb- solo se excitan por el dispositivo central.

30 Una configuración a modo de ejemplo de elementos útiles para conseguir que los excitadores, los receptores y las terminaciones transfieran señales como parte de la interfaz MDD inventiva se muestra en la FIG. 42, mientras que las correspondientes especificaciones eléctricas de CC de las señales MDDI_Data y MDDI_Stb se muestran en la tabla VII. Esta interfaz a modo de ejemplo utiliza una detección de baja tensión, en este caso de 20 mV, con oscilaciones de potencia de menos de 1 voltio y drenaje de baja potencia.

TABLA VII

Parámetro	Descripción	Min.	Tipo	Max.	Unidades
R_{term}	Terminación en serie	41,3	42,2	43,0	Ohmios
$R_{hibernación}$	Terminación de polarización de estado de hibernación	8	10	12	K-Ohmios
$V_{hibernación}$	Tensión de circuito abierto de estado de hibernación	0,5		2,8	V
$V_{Intervalo-salida}$	Intervalo permitido de tensión de salida de excitador con respecto a TIERRA	0		2,8	V
V_{OD+}	Alta tensión de salida diferencial de excitador	0,5			V
V_{OD-}	Baja tensión de salida diferencial de excitador			-0,5	V
V_{IT+}	Alta tensión umbral de entrada diferencial de receptor			10	mV
V_{IT-}	Baja tensión umbral de entrada diferencial de receptor	-10			mV

40

Parámetro	Descripción	Min.	Tipo	Max.	Unidades
$V_{\text{Intervalo_Entrada}}$	Intervalo permitido de tensión de entrada de receptor con respecto a TIERRA	0		3,0	V
I_{entrada}	Corriente de fuga de entrada (excluyendo la polarización de hibernación)	-25		25	μA

5 Los parámetros eléctricos y las características de los excitadores de línea diferenciales y de los receptores de línea diferenciales describen en la tabla VIII. De manera funcional, el excitador transfiere el nivel lógico de la entrada directamente a una salida positiva, y la inversa de la entrada a una salida negativa. El retardo desde la entrada hasta las salidas está bien ajustado con la línea diferencial que se excita de manera diferencial. En la mayoría de implementaciones, la oscilación de tensión en las salidas es inferior a la oscilación en la entrada para minimizar el consumo de potencia y las emisiones electromagnéticas. La tabla VIII presenta una oscilación de tensión mínima de 0,5 V aproximadamente. Sin embargo, pueden utilizarse otros valores, tal y como saben los expertos en la técnica, y se contempla un valor más pequeño en algunas realizaciones, dependiendo de las limitaciones de diseño.

10 Los receptores de línea diferenciales tienen las mismas características que un comparador de tensión de alta velocidad. En la FIG. 41, la entrada sin la burbuja es la entrada positiva y la entrada con la burbuja es la entrada negativa. La salida es un uno lógico si $(V_{\text{entrada}+}) - (V_{\text{entrada}-})$ es mayor que cero. Otro modo de describir esto es mediante un amplificador diferencial con una ganancia muy grande (prácticamente infinita) con la salida limitada a niveles de tensión de 0 y 1 lógicos.

15 El retardo diferencial entre los diferentes pares debe minimizarse para hacer funcionar el sistema de transmisión diferencial a la mayor velocidad posible.

20 En la FIG. 42, un controlador de dispositivo central 4202 y un controlador de dispositivo de visualización o dispositivo cliente 4204 se muestran transfiriendo paquetes a través del enlace de comunicaciones 4206. El controlador de dispositivo central utiliza una serie de tres excitadores 4210, 4212 y 4214 para recibir las señales DATA y STB de dispositivo central que van a transferirse, así como para recibir las señales de datos de dispositivo cliente que van a transferirse. Generalmente, el excitador responsable del paso de la señal DATA de dispositivo central utiliza una entrada de señal de habilitación para permitir la activación del enlace de comunicaciones solamente cuando se desea la transferencia desde el dispositivo central hasta el dispositivo cliente. Puesto que la señal STB se forma como parte de la transferencia de datos, no se emplea ninguna señal de habilitación adicional para ese excitador (4212). Las salidas de cada uno de los excitadores de las señales DATA y STB están conectadas a impedancias o resistencias de terminación 4216a, 4216b, 4216c y 4216d, respectivamente.

25 Las resistencias de terminación 4216a y 4216b también actuarán como impedancias en la entrada del receptor 4220 en el lado de dispositivo cliente para el procesamiento de la señal STB, mientras que resistencias de terminación adicionales 4216e y 4216f están colocadas en serie con resistencias 4216c y 4216d, respectivamente, en la entrada del receptor de procesamiento de datos de dispositivo cliente 4222. Un sexto excitador 4226 en el controlador de dispositivo cliente se utiliza para preparar las señales de datos que están transfiriéndose desde el dispositivo cliente hasta el dispositivo central, donde el excitador 4214, a través de resistencias de terminación 4216c y 4216d, en el lado de entrada, procesa los datos para transferirlos al dispositivo central para su procesamiento.

30 Dos resistencias adicionales 4218a y 4218b están colocadas entre las resistencias de terminación y tierra y una fuente de tensión 4220, respectivamente, como parte del control de hibernación descrito en este documento. La fuente de tensión se utiliza para excitar las líneas de transferencia a los niveles alto o bajo descritos anteriormente para el tratamiento de los flujos de datos.

35 Los excitadores y las impedancias anteriores pueden formarse como componentes discretos o como parte de un módulo de circuito, o como un circuito integrado de aplicación específica (ASIC) que actúa como una solución de codificador o de decodificador más económica.

40 Puede observarse fácilmente que la potencia se transfiere al dispositivo cliente, o de visualización, desde el dispositivo central utilizando las señales etiquetadas como MDDI_Pwr y MDDI_Gnd a través de un par de conductores. La parte MDDI_Gnd de la señal actúa como la tierra de referencia y como la trayectoria de retorno de fuente de alimentación o señal para el dispositivo de visualización. La señal MDDI_Pwr actúa como la fuente de alimentación del dispositivo de visualización, la cual se excita mediante el dispositivo central. En una configuración a modo de ejemplo, para aplicaciones de baja potencia, el dispositivo de visualización puede necesitar hasta 500 mA. La señal MDDI_Pwr puede proporcionarse desde fuentes de alimentación portátiles tales como, pero sin limitarse a, una batería o conjunto de baterías de tipo iones de litio que residan en el dispositivo central, y puede oscilar entre 3,2 y 4,3 voltios con respecto a la señal MDDI_Gnd.

VII. Características de temporización

A. Visión general

5 Las etapas y niveles de señal utilizados por un dispositivo cliente para garantizar el servicio del dispositivo central y utilizados por el dispositivo central para proporcionar tal servicio, se ilustran en la FIG. 43. En la FIG. 43, la primera parte de las señales ilustradas muestra un Paquete de Interrupción de Enlace que está transfiriéndose desde el dispositivo central, y la línea de datos se excita después a un estado de cero lógico utilizando el circuito de polarización de alta impedancia. Ningún dato está transmitiéndose mediante el dispositivo de visualización cliente, o el dispositivo central, el cual tiene su excitador inhabilitado. Una serie de impulsos estroboscópicos para la línea de la señal MDDI_Stb puede observarse en la parte inferior, ya que la señal MDDI_Stb está activa durante el Paquete de Interrupción de Enlace. Una vez que este paquete finaliza y el nivel lógico pasa a cero cuando el dispositivo central excita el circuito de polarización y la lógica a cero, la línea de la señal MDDI_Stb también pasa a un nivel de cero. Esto representa la finalización de la última transferencia de señales o servicio desde el dispositivo central, y puede haber ocurrido anteriormente en cualquier momento, y se incluye para mostrar el cese anterior de servicio y el estado de las señales antes del comienzo del servicio. Si se desea, una señal de este tipo puede enviarse solamente para reajustar el enlace de comunicaciones al estado apropiado sin que este dispositivo central haya establecido una comunicación anterior 'conocida'.

15 Tal y como se muestra en la FIG. 43, la señal transmitida desde el dispositivo cliente se fija inicialmente a un nivel lógico de cero. Dicho de otro modo, la salida del dispositivo cliente está a una alta impedancia, y el excitador está inhabilitado. Cuando está solicitándose un servicio, el dispositivo cliente habilita su excitador y envía una solicitud de servicio al dispositivo central en un periodo de tiempo, denominado *t*servicio, durante el cual la línea se excita a un nivel de uno lógico. Después, un determinado periodo de tiempo, denominado como *t*detección-dispositivo central, transcurre o puede necesitarse antes de que el dispositivo central detecte la solicitud, después del cual el dispositivo central responde con una secuencia de inicio de enlace excitando la señal a un nivel de uno lógico. En este momento, el dispositivo cliente desestima la solicitud e inhabilita el excitador de solicitud de servicio para que la línea de salida del dispositivo cliente pase de nuevo a un nivel de cero lógico. Durante este momento, la señal MDDI_Stb está en un nivel de cero lógico.

25 El dispositivo central excita la salida de datos de dispositivo central al nivel '1' durante un periodo denominado como *t*reinicio-alto, después del cual el dispositivo central excita el nivel lógico a cero y activa la señal MDDI_Stb durante un periodo denominado como *t*reinicio-bajo, después del cual el primer tráfico directo comienza con un Paquete de Cabecera de Subtrama, y los paquetes de tráfico directo se transfieren posteriormente. La señal MDDI_Stb se activa durante el periodo *t*reinicio-bajo y el Paquete de Cabecera de Subtrama posterior.

30 La tabla VIII muestra tiempos representativos para la longitud de los diversos periodos descritos anteriormente y la relación con velocidades de transferencia datos máximas y mínimas, donde:

$$t_{bit} = \frac{1}{Velocidad_Datos_Enlace}$$

Tabla VIII

Parámetro	Descripción	Min.	Tipo	Max.	Unidades
<i>t</i> servicio	Duración de impulso de solicitud de servicio de dispositivo de visualización	60	70	80	µseg
<i>t</i> reinicio-alto	Duración de impulso alto de reinicio de enlace de dispositivo central	140	150	160	µseg
<i>t</i> reinicio-bajo	Duración de impulso bajo de reinicio de enlace de dispositivo central	40	50	60	µseg
<i>t</i> detección-dispositivo visualización	Tiempo en que el dispositivo de visualización detecta la secuencia de reinicio de enlace	1		50	µseg
<i>t</i> detección-dispositivo central	Tiempo en que el dispositivo central detecta un impulso de solicitud de servicio	1		50	µseg
1/ <i>t</i> rend-min-bit	Velocidad de transferencia de datos de enlace para un dispositivo de rendimiento mínimo	0,001		1	Mbps
1/ <i>t</i> rend-max-bit	Intervalo máximo de velocidad de transferencia de datos de enlace para un	0,001		450	Mbps

	dispositivo				
	Velocidad de transferencia de datos de enlace inverso	0,0005		50	Mbps
t_{bit}	Periodo de un bit de datos de enlace directo	2,2		10^6	nsec

Los expertos en la técnica entenderán fácilmente que las funciones de los elementos individuales ilustrados en las FIG. 41 y 42 son ampliamente conocidas, y la función de los elementos de la FIG. 42 se confirma mediante el cronograma de la FIG. 43. Los detalles acerca de las terminaciones en serie y de las resistencias de hibernación que se muestran en la FIG. 42 se han omitido con respecto a la FIG. 41 ya que esa información no es necesaria para una descripción de cómo llevar a cabo la codificación *Data-Strobe* y de cómo recuperar el reloj a partir de la misma.

B. Enlace directo de temporización *Data-Strobe*

Las características de conmutación para la transferencia de datos en el enlace directo desde la salida de excitador de dispositivo central se muestran en la tabla IX. La tabla IX presenta una forma tabular de los valores mínimos y máximos deseados, frente a tiempos típicos en los que se producen determinadas transiciones de señal. Por ejemplo, el periodo de tiempo típico para que se produzca una transición desde el principio hasta el final de un valor de datos (salida de '0' ó '1'), una transición de *Data0* a *Data0*, denominado como t_{dd} -(salida-dispositivo central), es t_{bit} , mientras que el tiempo mínimo es de $t_{bit}-0,5$ nseg aproximadamente, y el tiempo máximo es de $t_{bit}+0,5$ nseg aproximadamente. La separación relativa entre transiciones en la línea *Data0*, en otras líneas de datos (*DataX*) y en las líneas estroboscópicas (*Stb*), se ilustra en la FIG. 44, donde se muestran las transiciones *Data0* a *Strobe*, *Strobe* a *Strobe*, *Strobe* a *Data0*, *Data0* a *no-Data0*, *no-Data0* a *no-Data0*, *no-Data0* a *Strobe* y *Strobe* a *no-Data0*, las cuales se denominan como t_{dd} -(salida-dispositivo central), t_{ds} -(salida-dispositivo central), t_{ss} -(salida-dispositivo central), t_{sd} -(salida-dispositivo central), t_{ddx} -(salida-dispositivo central), t_{dx} -(salida-dispositivo central), t_{dsx} -(salida-dispositivo central) y t_{sdx} -(salida-dispositivo central), respectivamente.

Tabla IX

Parámetro	Descripción	Min.	Tip.	Max.	Unidades
t_{dd} -(salida-dispositivo central)	Transición <i>Data0</i> a <i>Data0</i>	$t_{bit} - 0,5$	t_{bit}	$t_{bit} + 0,5$	nseg
t_{ds} -(salida-dispositivo central)	Transición <i>Data0</i> a <i>Strobe</i>	$t_{bit} - 0,8$	t_{bit}	$t_{bit} + 0,8$	nseg
t_{ss} -(salida-dispositivo central)	Transición <i>Strobe</i> a <i>Strobe</i>	$t_{bit} - 0,5$	t_{bit}	$t_{bit} + 0,5$	nseg
t_{sd} -(salida-dispositivo central)	Transición <i>Strobe</i> a <i>Data0</i>	$t_{bit} - 0,8$	t_{bit}	$t_{bit} + 0,8$	nseg
t_{ddx} -(salida-dispositivo central)	Transición <i>Data0</i> a <i>no-Data0</i>		t_{bit}		nseg
t_{dx} -(salida-dispositivo central)	Transición <i>no-Data0</i> a <i>no-Data0</i>	$t_{bit} - 0,5$	t_{bit}	$t_{bit} + 0,5$	nseg
t_{dsx} -(salida-dispositivo central)	Transición <i>no-Data0</i> a <i>Strobe</i>		t_{bit}		nseg
t_{sdx} -(salida-dispositivo central)	Transición <i>Strobe</i> a <i>no-Data0</i>		t_{bit}		nseg

Los requisitos de temporización MDDI típicos para la entrada de receptor de dispositivo cliente para las mismas señales que transfieren datos en el enlace directo se muestran en la tabla X. Puesto que se describen las mismas señales pero retardadas en el tiempo, no se necesita una nueva figura para ilustrar las características o el significado de señal de las respectivas etiquetas, tal y como entenderán los expertos en la técnica.

Tabla X

Parámetro	Descripción	Min.	Tip.	Max.	Unidades
t_{dd} -(entrada-dispositivo visualización)	Transición <i>Data0</i> a <i>Data0</i>	$t_{bit} - 1,0$	t_{bit}	$t_{bit} + 1,0$	nseg
t_{ds} -(entrada-dispositivo visualización)	Transición <i>Data0</i> a <i>Strobe</i>	$t_{bit} - 1,5$	t_{bit}	$t_{bit} + 1,5$	nseg
t_{ss} -(entrada-dispositivo visualización)	Transición <i>Strobe</i> a <i>Strobe</i>	$t_{bit} - 1,0$	t_{bit}	$t_{bit} + 1,0$	nseg

t_{tsd} -(entrada-dispositivo visualización)	Transición <i>Strobe</i> a <i>Data0</i>	$t_{tbit} - 1,5$	t_{tbit}	$t_{tbit} + 1,5$	nseg
t_{tddx} -(salida- dispositivo central)	Transición <i>Data0</i> a <i>no-Data0</i>		t_{tbit}		nseg
t_{tdxdx} -(salida- dispositivo central)	Transición <i>no-Data0</i> a <i>no-Data0</i>		t_{tbit}		nseg
t_{tdxs} -(salida- dispositivo central)	Transición <i>no-Data0</i> a <i>Strobe</i>		t_{tbit}		nseg
t_{tsdx} -(salida- dispositivo central)	Transición <i>Strobe</i> a <i>no-Data0</i>		t_{tbit}		nseg

Las FIG. 45 y 46 ilustran la presencia de un retardo de respuesta que puede producirse cuando el dispositivo central inhabilita o habilita el excitador de dispositivo central, respectivamente. En el caso en que un dispositivo central envía determinados paquetes, tal como el Paquete de Encapsulación de Enlace Inverso o el Paquete de Medición de Retardo de Ida y Vuelta, el dispositivo central inhabilita el excitador de línea después de que los paquetes deseados se hayan enviado, tal como los paquetes CRC de Parámetros, Alineación Estroboscópica y Todo a Cero ilustrados en la FIG. 45 como habiéndose transferido. Sin embargo, tal y como se muestra en la FIG. 45, el estado de la línea no conmuta necesariamente de '0' a un valor deseado más alto instantáneamente, aunque esto puede conseguirse potencialmente con determinados elementos de control o de circuito actuales, pero tarda en responder un periodo de tiempo denominado como periodo de Retardo de Inhabilitación de Excitador de Dispositivo Central. Aunque puede producirse casi instantáneamente de manera que este periodo de tiempo tiene una longitud de 0 nanosegundos (nseg), puede extenderse más fácilmente a lo largo de un periodo algo más largo, siendo 10 nseg una longitud de periodo máxima deseada, que se produce durante los periodos de paquete Tiempo de Vigilancia 1 e Inversión 1.

Haciendo referencia a la FIG. 46, puede observarse el cambio de nivel de señal experimentado cuando el excitador de dispositivo central se habilita para transferir un paquete tal como el Paquete de Encapsulación de Enlace Inverso o el Paquete de Medición de Retardo de Ida y Vuelta. En este caso, después de los periodos de paquete Tiempo de Vigilancia 2 o Inversión 2, el excitador de dispositivo central se habilita y empieza a excitar un nivel, en este caso '0', cuyo valor se aproxima o se alcanza a lo largo de un periodo de tiempo denominado como periodo de Retardo de Habilitación de Excitador de Dispositivo Central, el cual se produce durante el periodo de Rehabilitación de Excitador, antes de que se envíe el primer paquete.

Un proceso similar se produce para los excitadores y las transferencias de señal para el dispositivo cliente, en este caso un dispositivo de visualización. Las pautas generales para la longitud de estos periodos y sus respectivas relaciones se muestran a continuación en la tabla XI.

Tabla XI

Descripción	Min.	Max.	Unidades
Retardo de Inhibición de Excitador de Dispositivo Central	0	10	nseg
Retardo de Habilitación de Excitador de Dispositivo Central	0	2,0	nseg
Retardo de Inhabilitación de Excitador de Dispositivo de Visualización	0	10	nseg
Retardo de Habilitación de Excitador de dispositivo de Visualización	0	2,0	nseg

C. Enlace inverso de temporización *Data-Strobe*

Las características de conmutación y las relaciones de tiempo para las señales estroboscópicas y de datos utilizadas para transferir datos en el enlace inverso desde la salida de excitador de dispositivo cliente se muestran en las FIG. 47 y 48. Los tiempos típicos para determinadas transiciones de señal se describen posteriormente. La FIG. 47 ilustra la relación en la entrada de receptor de dispositivo central entre la temporización de los datos que están transfiriéndose y los flancos de subida y de descenso de los impulsos estroboscópicos. Es decir, lo que denomina como el tiempo de configuración para el flanco de subida o delantero de las señales estroboscópicas, tsu_sr , y el tiempo de configuración para el flanco de descenso o trasero de las señales estroboscópicas, tsu_sf . Una longitud de tiempo típica para estos periodos de configuración es del orden de un mínimo de 8 nanosegundos.

La FIG. 48 ilustra las características de conmutación y el retardo de salida de dispositivo cliente correspondiente generado por la temporización de datos inversos. En la FIG. 48 puede observarse la relación entre la temporización de los datos que están transfiriéndose y los flancos de subida y de descenso de los impulsos estroboscópicos que representan el retardo inducido. Es decir, lo que se denomina como el retardo de propagación entre el flanco de subida o delantero de la señales estroboscópicas y los datos (válidos), tpd_sr , y el retardo de propagación entre los datos y el flanco de descenso o

trasero de las señales estroboscópicas, tpd-sf. Una longitud de tiempo máxima típica para estos periodos de retardo de propagación es del orden de 8 nanosegundos.

VIII. Implementación del control de enlace (Funcionamiento del controlador de enlace)

A. Procesador de paquetes de máquina de estado

5 Los paquetes que están transfiriéndose a través de un enlace MDDI se despachan muy rápidamente, normalmente a una velocidad del orden de 300 Mbps o más, tal como 400 Mbps, aunque obviamente se permiten velocidades más bajas, según se desee. Este tipo de velocidad de enlace de transferencia o de bus es demasiado bueno como para que lo controlen los microprocesadores de propósito general (económicos) disponibles actualmente en el mercado o similares. Por lo tanto, una implementación práctica para conseguir este tipo de transferencia de señal es utilizar una máquina de estados programable para analizar sintácticamente el flujo de paquetes de entrada para generar paquetes que se transfieran o se redirijan al subsistema audiovisual apropiado para el que están destinados. Tales dispositivos son ampliamente conocidos y utilizan circuitos dedicados generalmente a un número limitado de operaciones, funciones o estados para conseguir un funcionamiento deseado a alta velocidad o a muy alta velocidad.

15 Pueden utilizarse controladores, procesadores o elementos de procesamiento de propósito general para tratar o manipular de manera más apropiada algún tipo de información tal como paquetes de control o de estado, los cuales demandan una velocidad más baja. Cuando se reciben estos paquetes (paquetes de control, de estado u otros paquetes predefinidos), la máquina de estados debe enviarlos, a través de una memoria intermedia de datos o un elemento de procesamiento similar, al procesador de propósito general para que los paquetes puedan tratarse para proporcionar un resultado (efecto) deseado, mientras que los paquetes de audio y vídeo se transfieren a su destino apropiado para su procesamiento. Si futuros microprocesadores u otros controladores, procesadores o elementos de procesamiento de propósito general se fabrican para conseguir capacidades de procesamiento con una mayor velocidad de transferencia de datos, entonces los estados o la máquina de estados descritos a continuación también pueden implementarse utilizando el control de software de tales dispositivos, normalmente como programas almacenados en un elemento o medio de almacenamiento.

25 La función del procesador de propósito general puede realizarse en algunas realizaciones aprovechando la potencia de procesamiento o los ciclos de exceso disponibles para los microprocesadores (CPU) en aplicaciones informáticas, o para los controladores, procesadores, procesadores de señales digitales (DSP), circuitos especializados o ASIC encontrados en los dispositivos inalámbricos, casi de la misma forma en que algunos módems o procesadores gráficos utilizan la potencia de procesamiento de las CPU encontradas en los ordenadores para llevar a cabo algunas funciones y reducir la complejidad y el coste del hardware. Sin embargo, esta compartición o utilización de ciclos puede afectar negativamente a la velocidad de procesamiento, a la temporización o al funcionamiento global de tales elementos, por lo que en muchas aplicaciones se prefieren circuitos o elementos dedicados para este procesamiento general.

35 Para que los datos de imagen se visualicen en un dispositivo de visualización (microdispositivo de visualización), o para recibir de manera fiable todos los paquetes enviados por el dispositivo central, el procesamiento de señales de dispositivo de visualización se sincroniza con la temporización de canal de enlace directo. Es decir, las señales que llegan al dispositivo de visualización y a los circuitos del dispositivo de visualización necesitan estar sustancialmente sincronizadas en el tiempo para que se lleve a cabo un procesamiento de señales apropiado. Un diagrama de estados de alto nivel obtenido por las etapas de procesamiento de señales o por un procedimiento mediante el cual puede implementarse una sincronización de este tipo se presenta en la ilustración de la FIG. 49. En la FIG. 49, los posibles "estados" de sincronización de enlace directo para una máquina de estados 4900 se muestran categorizados como un ESTADO DE TRAMAS NO SINCRONIZADAS 4904, dos ESTADOS DE ADQUISICIÓN DE SINCRONIZACIÓN 4902 y 4906, y tres ESTADOS SINCRONIZADOS 4908, 4910 y 4912.

45 Tal y como se muestra mediante la etapa o estado inicial 4902, el dispositivo cliente o de visualización, tal como un dispositivo de presentación, empieza en un estado "no sincronizado" preseleccionado y busca una palabra única en el primer paquete de cabecera de subtrama que se detecta. Debe observarse que este estado no sincronizado representa el ajuste de comunicación mínimo o ajuste de "retroceso" en el que se selecciona una interfaz de Tipo I. Cuando la palabra única se encuentra durante la búsqueda, el dispositivo de visualización guarda el campo de longitud de subtrama. No se comprueban los bits CRC para su procesamiento en esta primera trama o hasta que se obtenga la sincronización. Si esta longitud de subtrama es cero, entonces el procesamiento de estado sincronizado avanza por consiguiente a un estado 4904 etiquetado en este caso como el estado de "tramas no sincronizadas", que indica que todavía no se ha conseguido la sincronización. Esta etapa del procesamiento está etiquetada como habiendo cumplido la cond 3, o condición 3, en la FIG. 49. En caso contrario, si la longitud de trama es mayor que cero, entonces el procesamiento de estado sincronizado avanza hasta un estado 4906 donde el estado de la interfaz se fija como "encontrada una trama sincronizada". Esta etapa del procesamiento está etiquetada como cumpliendo la cond 5, o condición 5, en la FIG. 49. Además, si la máquina de estados ve un paquete de cabecera de trama y una determinación CRC correcta para una longitud de trama mayor que cero, el procesamiento avanza hasta el estado "encontrada una trama sincronizada". Esto está etiquetado como cumpliendo la cond 6, o condición 6, en la FIG. 49.

En cada situación en la que el sistema está en un estado distinto al “no sincronizado”, cuando se detecta la palabra única y se determina un resultado CRC correcto para el paquete de cabecera de subtrama, y la longitud de subtrama es mayor que cero, entonces el estado de interfaz se cambia al estado “sincronizado” 4908. Esta etapa del procesamiento está etiquetada como habiendo cumplido la cond 1, o condición 1, en la FIG. 49. Por otro lado, si la palabra única o la CRC del Paquete de Cabecera de Subtrama no son correctas, entonces el procesamiento de estado sincronizado avanza o vuelve al estado de interfaz 4902 de estado de "TRAMA NO SINCRONIZADA". Esta parte del procesamiento está etiquetada como cumpliendo la cond 2, o condición 2, en el diagrama de estados de la FIG. 49.

B. Tiempo de adquisición para la sincronización

La interfaz puede configurarse para permitir un determinado número de “errores de sincronización” antes de decidir que la sincronización se ha perdido y volver al estado "TRAMA NO SINCRONIZADA". En la FIG. 49, una vez que la máquina de estados haya alcanzado el “ESTADO SINCRONIZADO” y no se encuentren errores, obtiene constantemente el resultado de la cond 1 y permanece en el estado “SINCRONIZADO”. Sin embargo, una vez que se detecte el resultado de la cond 2, el procesamiento cambia el estado a un estado de “error de sincronización” 4910. En este momento, si el procesamiento da como resultado la detección de otro resultado de cond 1, entonces la máquina de estados vuelve al estado “sincronizado”; en caso contrario, obtiene otro resultado de cond 2 y avanza hasta un estado “DOS ERRORES DE SINCRONIZACIÓN” 4912. Nuevamente, si se produce la cond 1, el procesamiento hace volver la máquina de estados al estado “SINCRONIZADO”. En caso contrario, se obtiene otra cond 2 y la máquina de estados vuelve al estado “no sincronizado”. Debe entenderse además que si la interfaz encuentra un “paquete de interrupción de enlace”, entonces esto hará que el enlace deje de transferir datos y se volverá al estado “trama no sincronizada” ya que no hay nada con lo que sincronizarse, lo que se denomina como cumplir la cond 4, o condición 4, en el diagrama de estados de la FIG. 49.

Debe entenderse que es posible que haya una “copia falsa” repetitiva de la palabra única que puede aparecer en alguna ubicación fija dentro de la subtrama. En esta situación, es altamente improbable que la máquina de estados se sincronice con la subtrama, ya que la CRC del Paquete de Cabecera de Subtrama también debe validarse cuando se procesa para que el procesamiento de interfaz MDD avance hasta el estado “SINCRONIZADO”.

La longitud de subtrama del Paquete de Cabecera de Subtrama puede fijarse a cero para indicar que el dispositivo central transmitirá solamente una subtrama antes de que el enlace se interrumpa, y la interfaz MDD se lleva a o se configura en un estado de hibernación inactivo. En este caso, el dispositivo de visualización debe recibir inmediatamente paquetes a través del enlace directo después de detectar el Paquete de Cabecera de Subtrama porque solo se envía una única subtrama antes de que el enlace pase al estado inactivo. En operaciones normales o típicas, la longitud de subtrama no es cero y el dispositivo de visualización solo procesa paquetes de enlace directo mientras que la interfaz está en los estados mostrados de manera colectiva como estados "SINCRONIZADOS", en la FIG. 49.

El tiempo requerido para que un dispositivo de visualización se sincronice con la señal de enlace directo varía dependiendo del tamaño de subtrama y de la velocidad de transferencia de datos de enlace directo. La probabilidad de detectar una "copia falsa" de la palabra única como parte de los datos aleatorios, o más aleatorios, en el enlace directo es mayor cuando el tamaño de subtrama es más grande. Al mismo tiempo, la capacidad de recuperarse de una detección falsa es menor, y el tiempo que tarda en hacerlo es mayor, cuando la velocidad de transferencia de datos de enlace directo es más baja.

C. Inicialización

Tal y como se ha indicado anteriormente, en el momento del “encendido”, el dispositivo central configura el enlace directo para funcionar a o por debajo de una velocidad de transferencia de datos mínima requerida, o deseada, de 1 Mbps, y configura la longitud de subtrama y la velocidad de trama multimedia de manera apropiada para una aplicación dada. Es decir, tanto el enlace directo como el enlace inverso inician su funcionamiento utilizando la interfaz de Tipo I. Estos parámetros solo se utilizan generalmente de manera temporal cuando el dispositivo central determina la capacidad o la configuración deseada para el dispositivo de visualización cliente (u otro tipo de dispositivo cliente). El dispositivo central envía o transfiere un Paquete de Cabecera de Subtrama a través del enlace directo seguido de un Paquete de Encapsulación de Enlace Inverso que tiene el bit '0' de las Banderas de Solicitud fijado a un valor de uno (1) con el fin de solicitar que el dispositivo de visualización o dispositivo cliente responda con un Paquete de Capacidades de Dispositivo de Visualización. Una vez que el dispositivo de visualización haya adquirido la sincronización en (o con) el enlace directo, envía un Paquete de Capacidades de Dispositivo de Visualización y un Paquete de Estado y Solicitud de Dispositivo de Visualización a través del enlace o canal inverso.

El dispositivo central examina los contenidos del Paquete de Capacidades de Dispositivo de Visualización con el fin de determinar la reconfiguración del enlace para un nivel de rendimiento óptimo o deseado. El dispositivo central examina los campos Versión de Protocolo y Versión Mínima de Protocolo para confirmar que el dispositivo central y el dispositivo de visualización utilizan versiones del protocolo que son compatibles entre sí. Las versiones de protocolo permanecen generalmente como los dos primeros parámetros del Paquete de Capacidades de Dispositivo de Visualización, de manera que la compatibilidad puede determinarse incluso cuando otros elementos del protocolo puedan no ser compatibles o considerarse completamente compatibles.

D. Procesamiento CRC

Para todos los tipos de paquete, la máquina de estados de procesador de paquetes garantiza que el verificador CRC se controle de manera apropiada o correcta. También incrementa un contador de errores CRC cuando una comparación CRC da como resultado la detección de uno o más errores, y reajusta el contador CRC al principio de cada subtrama que esté procesándose.

E. Pérdida alternativa de comprobación de sincronización

Aunque la anterior serie de etapas o estados funciona para producir velocidades de transferencia de datos o de procesamiento más altas, se ha descubierto que una disposición alternativa o cambio en las condiciones que el dispositivo cliente utiliza para declarar que hay una pérdida de sincronización con el dispositivo central, puede utilizarse de manera eficaz para conseguir un procesamiento o velocidades de transferencia de datos incluso mayores. La nueva realización inventiva tiene la misma estructura básica pero con las condiciones de cambio de estado modificadas. Además, un nuevo contador está implementado para ayudar a realizar las comprobaciones para la sincronización de subtramas. Estas etapas y condiciones se presentan con respecto a la FIG. 63, que ilustra una serie de estados y condiciones útiles para establecer las operaciones del procedimiento o máquina de estados. Sólo se muestran las partes de "ESTADOS DE ADQUISICIÓN DE SINCRONIZACIÓN" y "ESTADOS SINCRONIZADOS" por motivos de claridad. Además, puesto que los estados resultantes son sustancialmente los mismos, ya que es la misma máquina de estados, utilizan la misma numeración. Sin embargo, las condiciones para cambiar de estado (y el funcionamiento de la máquina de estados) varían en cierto modo, de manera que todas se han vuelto a numerar por motivos de claridad entre las dos figuras (1, 2, 3, 4, 5 y 6 frente a 61, 62, 63, 64 y 65), para una mayor claridad para identificar las diferencias. Puesto que el estado TRAMA NO SINCRONIZADA no se considera en este análisis, el estado (4904) y la condición (6) no se utilizan en esta figura.

En la FIG. 63, el sistema o dispositivo cliente (para la visualización o la presentación) comienza con la máquina de estados 500 en el estado "no sincronizado" 4902 preseleccionado, como en la FIG. 49. El primer cambio de estado para cambiar los estados desde el estado no sincronizado 4902 está en la condición 64 que es el descubrimiento del patrón de sincronización. Suponiendo que la CRC de la cabecera de subtrama también se transmite en este paquete (se cumple la condición 61), el estado de la máquina de estados de procesador de paquetes puede cambiarse al estado sincronizado 4908. Un error de sincronización, condición 62, hará que la máquina de estados pase al estado 4910, y una segunda ocurrencia al estado 4912. Sin embargo, se ha descubierto que cualquier fallo CRC de un paquete MDDI hará que la máquina de estados salga del estado sincronizado 4908 al estado de error de sincronización 4910. Otro fallo CRC de cualquier paquete MDDI provocará un cambio al estado de dos errores de sincronización 4912. Un paquete descodificado con un valor CRC correcto hará que la máquina de estados vuelva al estado sincronizado 4908.

Lo que se ha cambiado es utilizar el valor o determinación CRC para 'cada' paquete. Es decir, hacer que la máquina de estados compruebe el valor CRC para cada paquete para determinar una pérdida de sincronización en lugar de solo observar paquetes de cabecera de subtrama. En esta configuración o proceso, una pérdida de sincronización no se determina utilizando la palabra única ni solamente los valores CRC de cabecera de subtrama.

Esta nueva implementación de interfaz permite al enlace de interfaz MDD reconocer los fallos de sincronización mucho más rápidamente y, por lo tanto, recuperarse de los mismos más rápidamente.

Para hacer este sistema más robusto, el dispositivo cliente también debe añadir o utilizar un contador de subtramas. Después, el dispositivo cliente comprueba la presencia de la palabra única en el momento en que se espera que llegue o se produzca en una señal. Si la palabra única no se produce en el instante correcto, el dispositivo cliente puede reconocer mucho más rápidamente que se ha producido un fallo de sincronización que si tuviera que esperar varios (en este caso, tres) tiempos o periodos de paquete que fueran mayores que una longitud de subtrama. Si la prueba para la palabra única indica que no está presente, dicho de otro modo, que la temporización es incorrecta, entonces el dispositivo cliente puede declarar inmediatamente una pérdida de sincronización de enlace y pasar al estado no sincronizado. El proceso de comprobar la presencia de la palabra única adecuada añade una condición 65 (cond 65) a la máquina de estados que indica que la palabra única es incorrecta. Si se espera recibir un paquete de subtrama en el dispositivo cliente y no coincide, el dispositivo cliente puede pasar inmediatamente al estado no sincronizado 4902 ahorrando tiempo de espera adicional de múltiples errores de sincronización (condición 62) encontrados normalmente pasando entre los estados 4910 y 4912.

Este cambio utiliza un contador adicional o función de cómputo en el núcleo del dispositivo cliente para contar la longitud de subtrama. En una realización se utiliza una función de cómputo descendente y la transferencia de cualquier paquete que estuviera procesándose actualmente se interrumpe para comprobar la palabra única de subtrama si el contador ha expirado. Como alternativa, el contador puede contar de manera ascendente, comparándose el cómputo con un valor máximo o particular deseado, momento en que se comprueba el paquete actual. Este proceso impide que el dispositivo cliente descodifique paquetes que se hayan recibido de manera incorrecta en el dispositivo cliente con longitudes de paquete extraordinariamente largas. Si el contador de longitud de subtrama necesitó interrumpir algún otro paquete que estaba descodificándose, puede determinarse una pérdida de sincronización ya que ningún paquete debe atravesar un límite de subtrama.

IX. Procesamiento de paquetes

Para cada tipo de paquete descrito anteriormente que recibe la máquina de estados, éste experimenta una etapa o serie de etapas de procesamiento particular para implementar el funcionamiento de la interfaz. Los paquetes de enlace directo se procesan generalmente según el procesamiento a modo de ejemplo enumerado a continuación en la tabla XII.

Tabla XII

<i>Tipo de paquete</i>	<i>Respuesta de la máquina de estados del procesador de paquetes</i>
Cabecera de Subrama (SH)	Confirma paquete correcto, captura campo de longitud de subrama y envía parámetros de paquete a un procesador de propósito general.
Relleno (F)	Ignora los datos.
Flujo de vídeo (VS)	Interpreta el parámetro Descriptor de Formato de Datos de Vídeo y otros parámetros, desempaqueta los datos de píxel empaquetados cuando sea necesario, convierte píxeles a través del mapa de colores si fuera necesario y escribe datos de píxel en ubicaciones apropiadas del mapa de bits.
Flujo de Audio (AS)	Envía ajuste de frecuencia de muestro de audio a un generador de reloj de muestras de audio, separa muestras de audio de tamaño especificado, desempaqueta datos de muestras de audio cuando sea necesario y encamina muestras de audio hacia una FIFO de muestras de audio apropiada.
Mapa de Colores (CM)	Lee el tamaño de mapa de colores y parámetros de desplazamiento, y escribe los datos de mapa de colores en una ubicación de almacenamiento o memoria de mapa de colores.

<i>Tipo de paquete</i>	<i>Respuesta de la máquina de estados del procesador de paquetes</i>
Encapsulación de Enlace Inverso (REL)	Facilita el envío de paquetes en el sentido inverso en el momento apropiado. Se examinan las banderas de enlace inverso y se envían Paquetes de Capacidades de Dispositivo de Visualización según sea necesario. También se envían paquetes de Estado y Solicitud de Dispositivo de Visualización según sea apropiado.
Capacidades de Dispositivo de Visualización (DC)	Envía este tipo de paquete cuando lo solicita un dispositivo central, usando el campo de banderas de enlace inverso del Paquete de Encapsulación de Enlace Inverso
Teclado (K)	Transmite estos paquetes desde y hasta un procesador de propósito general que se comunica con un dispositivo de tipo teclado, si está presente, y se utiliza si se desea.
Dispositivo de Puntero (PD)	Transmite estos paquetes desde y hasta un procesador de propósito general que se comunica con un dispositivo de tipo puntero, si está presente, y se utiliza si se desea.
Interrupción de Enlace (LS)	Registra la acción de interrupción de enlace e informa a un procesador de propósito general.
Estado y Solicitud de Servicio de Dispositivo Cliente (DSRS)	Envía este paquete como el primer paquete en el Paquete de Encapsulación de Enlace Inverso.
Transferencia de Bloques de Bits (BPT)	Interpreta parámetros de paquete, tales como el Descriptor de Formato de Datos de Vídeo, determina qué píxeles mover primero y mueve los píxeles en el mapa de bits según sea necesario.
Relleno de Área de Mapa de Bits (BAF)	Interpreta parámetros de paquete, convierte píxeles a través del mapa de colores si fuera necesario y escribe datos de píxel en ubicaciones apropiadas del mapa de bits.
Relleno de Patrón de Mapa de Bits (BPF)	Interpreta parámetros de paquete, desempaqueta datos de píxel empaquetados si fuera necesario, convierte píxeles a través del mapa de colores si fuera necesario y escribe datos de píxel en ubicaciones apropiadas

	del mapa de bits.
Canal de Enlace de Comunicaciones (CLC)	Envía estos datos directamente a un procesador de propósito general.

<i>Tipo de paquete</i>	<i>Respuesta de la máquina de estados del procesador de paquetes</i>
Solicitud de Servicio de Dispositivo de Visualización (DSR) durante la hibernación	El procesador de propósito general controla las funciones de bajo nivel de envío de solicitud y detecta por sí mismo contienda por el reinicio de enlace.
Solicitud de Traspaso de Tipo de Interfaz (ITHR) y Confirmación de Recepción de Tipo de Interfaz (ITA)	Puede pasar estos paquetes desde y hasta el procesador de propósito general. La lógica para recibir este tipo de paquete y formular una respuesta con una confirmación de recepción es sustancialmente mínima. Por lo tanto, esta operación también puede implementarse en la máquina de estados del procesador de paquetes. El cambio resultante se produce como una acción de capa física de bajo nivel y no es probable que afecte a la funcionalidad o funcionamiento del procesador de propósito general.
Traspaso de Tipo de Acción (PTH)	Puede actuar en tales paquetes ya sea directamente o transfiriéndolos al procesador de propósito general, haciendo también que el hardware experimente un cambio de modo.

X. Reducción de la velocidad de transferencia de datos de enlace inverso

5 Se ha observado que determinados parámetros utilizados para el controlador de enlace de dispositivo central pueden ajustarse o configurarse de una determinada manera con el fin de conseguir una velocidad de transferencia de datos de enlace inverso máxima o más optimizada (escala), lo que es muy deseable. Por ejemplo, durante el tiempo utilizado para transferir el campo Paquetes de Datos Inversos del Paquete de Encapsulación de Enlace Inverso, el par de señales MDDI_Stb oscila para crear un reloj de datos periódico a la mitad de la velocidad de transferencia de datos de enlace directo. Esto sucede porque el controlador de enlace de dispositivo central genera la señal MDDI_Stb que corresponde a la señal MDDI_Data0 como si estuviera enviando todo a cero. La señal MDDI_Stb se transfiere desde el dispositivo central hasta el dispositivo cliente donde se utiliza para generar una señal de reloj para transferir datos de enlace inverso desde el dispositivo de visualización con la que los datos inversos se envían al dispositivo central. Una ilustración de cantidades típicas de retardo encontradas para la transferencia y el procesamiento de señales en las trayectorias directa e inversa en un sistema que utiliza la MDDI se muestra en la FIG. 50. En la FIG. 50 se muestra una serie de valores de retardo de 1,5 nsec, 8,0 nseg, 2,5 nseg, 2,0 nseg, 1,0 nseg, 1,5 nseg, 8,0 nseg y 2,5 nseg cerca de partes de procesamiento para la generación de Stb+/-, el cable de transferencia al dispositivo de visualización, el receptor de dispositivo de visualización, la generación de reloj, la sincronización de señales, la generación de Data0+/-, el cable de transferencia al dispositivo central y las etapas del receptor de dispositivo central, respectivamente.

20 Dependiendo de la velocidad de transferencia de datos de enlace directo y de los retardos de procesamiento de señal encontrados, puede requerirse más tiempo que un ciclo en la señal MDDI_Stb para este efecto o conjunto de eventos de "ida y vuelta" que han de completarse, lo que da como resultado el consumo no deseable de cantidades de tiempo o ciclos. Para sortear este problema, el Divisor de Velocidad Inversa hace posible que el tiempo de un bit en el enlace inverso abarque múltiples ciclos de la señal MDDI_Stb. Esto significa que la velocidad de transferencia de datos de enlace inverso es inferior a la velocidad de enlace directo.

25 Debe observarse que la longitud real de los retardos de señal a través de la interfaz puede variar dependiendo de cada sistema o hardware específico de dispositivo central y de dispositivo cliente que esté utilizándose. Aunque no es necesario, generalmente puede hacerse que cada sistema funcione mejor utilizando el Paquete de Medición de Retardo de Ida y Vuelta para medir el retardo real en un sistema, de manera que el Divisor de Velocidad Inversa puede fijarse a un valor óptimo.

30 Un retardo de ida y vuelta se mide haciendo que el dispositivo central envíe un Paquete de Medición de Retardo de Ida y Vuelta al dispositivo de visualización. El dispositivo de visualización responde a este paquete enviando una secuencia de unos al dispositivo central dentro de, o durante, una ventana de medición preseleccionada en ese paquete denominada como el campo Periodo de Medición. La temporización detallada de esta medición se ha descrito anteriormente. El retardo de ida y vuelta se utiliza para determinar la velocidad a la que los datos de enlace inverso pueden muestrearse de manera segura.

35 La medición de retardo de ida y vuelta consiste en determinar, detectar o contar el número de intervalos de reloj

de datos de enlace directo que se producen entre el inicio del campo Periodo de Medición y el inicio del periodo de tiempo cuando la secuencia de respuesta 0xff, 0xff, 0x00 se recibe en el dispositivo central desde el dispositivo cliente. Debe observarse que es posible que la respuesta del dispositivo cliente pueda recibirse en una pequeña fracción de un periodo de reloj de enlace directo antes de que el cómputo de medición esté a punto de incrementarse. Si este valor no modificado se utiliza para calcular el Divisor de Velocidad inversa, puede generar errores de bit en el enlace inverso debido a un muestreo de datos no fiable. Un ejemplo de esta situación se ilustra en la FIG. 51, donde las señales que representan MDDI_Data en el dispositivo central, MDDI_Stb en el dispositivo central, el reloj de datos de enlace directo en el dispositivo central y el Cómputo de Retardo se ilustran de manera gráfica. En la FIG. 51, la secuencia de respuesta se recibió desde el dispositivo de visualización en una fracción de un periodo de reloj de enlace directo antes de que el Cómputo de Retardo estuviera a punto de incrementarse de 6 a 7. Si se supone que el retardo es 6, entonces el dispositivo central muestrearía los datos inversos justo después de una transición de bit o posiblemente a mitad de una transición de bit. Esto puede dar como resultado un muestreo erróneo en el dispositivo central. Por esta razón, el retardo medido debe incrementarse normalmente en uno antes de que se utilice para calcular el Divisor de Velocidad inversa.

El Divisor de Velocidad inversa es el número de ciclos de señal MDDI_Stb que el dispositivo central debe esperar antes de muestrear los datos de enlace inverso. Puesto que la señal MDDI_Stb pasa por ciclos a una velocidad que es la mitad de la velocidad de enlace directo, la medición corregida de retardo de ida y vuelta necesita dividirse por 2 y después redondearse por exceso al siguiente entero. Expresada como una fórmula, esta relación es:

$$divisor_velocidad_inversa = \text{redondearPorExcesoSiguieteEntero} \left(\frac{retardo_ida_y_vuelta + 1}{2} \right)$$

Para el ejemplo dado, se convierte en:

$$divisor_velocidad_inversa = \text{redondearPorExcesoSiguieteEntero} \left(\frac{6+1}{2} \right) = 4$$

Si la medición de retardo de ida y vuelta utilizada en este ejemplo fue de 7 en lugar de 6, entonces el Divisor de Velocidad inversa también sería igual a 4.

Los datos de enlace inverso se muestrean por el dispositivo central en el flanco de subida del reloj de enlace inverso. Hay presente un contador o un circuito o dispositivo conocido similar tanto en el dispositivo central como en el dispositivo cliente (de visualización) para generar el reloj de enlace inverso. Los contadores se inicializan de manera que el primer flanco de subida del reloj de enlace inverso se produce al principio del primer bit en el campo Paquetes de Enlace Inverso del Paquete de Encapsulación de Enlace Inverso. Esto se ilustra, para el ejemplo proporcionado posteriormente, en la FIG. 52. Los contadores se incrementan en cada flanco de subida de la señal MDDI_Stb, y el número de cálculos producidos hasta que vuelven a cero se fija por el parámetro Divisor de Velocidad Inversa del Paquete de Encapsulación de Enlace Inverso. Puesto que la señal MDDI_Stb oscila a la mitad de la velocidad de enlace directo, la velocidad de enlace inverso es la mitad de la velocidad de enlace directo dividida por el Divisor de Velocidad Inversa. Por ejemplo, si la velocidad de enlace directo es de 200 Mbps y el Divisor de Velocidad inversa es 4, entonces la velocidad de datos de enlace inverso se expresa como:

$$\frac{1}{2} \cdot \frac{200Mbps}{4} = 25Mbps$$

Un ejemplo que muestra la temporización de las líneas de las señales MDDI_Data0 y MDDI_Stb en un Paquete de Encapsulación de Enlace Inverso se muestra en la FIG. 52, donde los parámetros de paquete utilizados a modo de ilustración tienen los valores:

Longitud de Paquete = 1024 (0x0400)	Longitud de Inversión 1 = 1
Tipo de Paquete = 65 (0x41)	Longitud de Inversión 2 = 1
Banderas de Enlace Inverso = 0	Divisor de Velocidad Inversa = 2
CRC de Parámetros = 0xdb43	Todo a Cero es 0x00

Los datos de paquete entre los campos Longitud de Paquete y CRC de Parámetros son:

0x00, 0x04, 0x41, 0x00, 0x02, 0x01, 0x01, 0x43, 0xdb, 0x00,...

El primer paquete de enlace inverso devuelto desde el dispositivo de visualización es el Paquete de Estado y Solicitud de Dispositivo de Visualización, que tiene una Longitud de Paquete de 7 y un tipo de paquete de 70. Este paquete comienza con los valores de byte 0x07, 0x00, 0x46, etc. Sin embargo, solamente el primer byte (0x07) es visible en la FIG. 52. Este primer paquete de enlace inverso está desplazado en el tiempo en casi un periodo de reloj de enlace inverso en la figura para ilustrar un retardo de enlace inverso real. Una forma de onda ideal con un retardo cero de ida y vuelta desde el dispositivo central hasta el dispositivo de visualización se muestra como una traza de línea de puntos.

Se transfiere el byte MS del campo CRC de Parámetros, precedido por el tipo de paquete, y después el campo de todo a cero. La señal estroboscópica del dispositivo central conmuta de uno a cero y de nuevo a uno a medida que los datos del dispositivo central cambian de nivel, formando impulsos más anchos. Cuando los datos pasan a cero, la señal estroboscópica conmuta a la velocidad superior, y solo el cambio en los datos de la línea de datos provoca un cambio cerca del final del campo de alineación. La señal estroboscópica conmuta a la velocidad superior durante el resto de la figura debido a niveles fijos de 0 ó 1 de la señal de datos para mayores periodos de tiempo, y las transiciones siguen el patrón de impulsos (flanco).

El reloj de enlace inverso para el dispositivo central está a cero hasta el final del periodo Inversión 1, cuando el reloj se activa para permitir paquetes de enlace inverso. Las flechas de la parte inferior de la figura indican cuándo se muestrean los datos, como resultará evidente a partir del resto de la descripción. El primer byte del campo de paquete que está transfiriéndose (en este caso, 11000000) se muestra empezando después de Inversión 1, y el nivel de línea se ha estabilizado por el excitador de dispositivo central que está inhabilitándose. El retardo en el paso del primer bit, y como puede observarse para el tercer bit, puede verse en las líneas de puntos para la señal Data.

En la FIG. 53 pueden observarse valores típicos del Divisor de Velocidad Inversa en función de la velocidad de transferencia de datos de enlace directo. El Divisor de Velocidad Inversa real se determina como resultado de una medición de enlace de ida y vuelta para garantizar un funcionamiento de enlace inverso apropiado. Una primera región 5302 corresponde a un área de funcionamiento seguro, una segunda región 5304 corresponde a un área de rendimiento marginal, mientras que una tercera región 5306 indica ajustes que es improbable que funcionen correctamente.

La medición de retardo de ida y vuelta y el ajuste del Divisor de Velocidad Inversa son idénticos cuando funcionan con cualquiera de los ajustes de Tipo de Interfaz en el enlace directo o en el enlace inverso ya que están expresados y se utilizan en términos de unidades de periodos de reloj reales en lugar de números de bits transmitidos o recibidos.

XI. Tiempos de inversión y de vigilancia

Tal y como se ha descrito anteriormente, el campo Inversión 1 del Paquete de Encapsulación de Enlace Inverso y el campo Tiempo de Vigilancia 1 del Paquete de Medición de Retardo de Ida y Vuelta designan valores para longitudes de tiempo que permiten inhabilitar los excitadores de interfaz de dispositivo central antes de habilitar los excitadores de interfaz de dispositivo de visualización. Los campos Inversión 2 y Tiempo de Vigilancia 2 proporcionan valores de tiempo que permiten inhabilitar los excitadores de dispositivo de visualización antes de habilitar los excitadores de dispositivo central. Los campos Tiempo de Vigilancia 1 y Tiempo de Vigilancia 2 se rellenan generalmente con valores prefijados o preseleccionados para longitudes cuyo ajuste no está previsto. Dependiendo del hardware de interfaz que esté utilizándose, estos valores pueden desarrollarse utilizando datos empíricos y ajustarse en algunos casos para mejorar el funcionamiento.

Varios factores contribuyen a determinar la longitud de Inversión 1, los cuales son la velocidad de transferencia de datos de enlace directo y el tiempo de inhabilitación máximo de los excitadores de señal MDDI_Data en el dispositivo central. El tiempo de inhabilitación máximo de excitador de dispositivo central está especificado en la Tabla XI, donde se muestra que los excitadores tardan aproximadamente 10 nseg como máximo en inhabilitarse y 2 nseg aproximadamente en habilitarse. El número mínimo de ciclos de reloj de enlace directo requerido para que el excitador de dispositivo central se deshabilite se expresa según la relación:

$$Ciclos_reloj_Inhabilitar_{INV1} = \frac{VelocidadDatosEnlaceDirecto}{FactorTipoInterfaz_{DIR}} \cdot retardoInhabilitarExcitadorDispositivoCentral_{max}$$

El intervalo de valores permitidos de Inversión 1 se expresa según la relación:

$$Inversión_1 \geq \text{redondearPorExcesoSiguiendoEntero} \left(\frac{Ciclos_reloj_Inhabilitar_{INV1} \cdot FactorTipoInterfaz_{DIR}}{8} \right)$$

donde el Factor de Tipo de Interfaz es 1 para el Tipo I, 2 para el Tipo II, 4 para el Tipo III y 8 para el Tipo IV.

Combinando las dos ecuaciones anteriores, puede observarse que el término Factor de Tipo de Interfaz se

simplifica, e Inversión 1 se define como:

$$Inversión_1 = \text{redondearPorExcesoSiguieteEntero} \left(\frac{VelocidadDatosEnlaceDirecto \cdot \text{retardoInhabilitarExcitadorDispositivoCentral}_{\max}}{8} \right)$$

Por ejemplo, un enlace directo de Tipo III de 1500 Mbps usará un retardo Inversión 1 de:

5

$$Inversión_1 = \text{redondearPorExcesoSiguieteEntero} \left(\frac{1500 \text{ Mbps} \cdot 10 \text{ nseg}}{8} \right) = 2 \text{ Bytes}$$

A medida que aumenta el retardo de ida y vuelta, el margen de temporización mejora desde el instante en que el dispositivo central se inhabilita cuando se habilita el dispositivo de visualización.

10

Los factores que determinan la longitud de tiempo utilizado generalmente para Inversión 2 son la velocidad de transferencia de datos de enlace directo, el tiempo de inhabilitación máximo de los excitadores de señal MDDI_Data del dispositivo de visualización y el retardo de ida y vuelta del enlace de comunicaciones. El cálculo del tiempo requerido para inhabilitar el excitador de dispositivo de visualización es esencialmente el mismo que para el excitador de dispositivo central descrito anteriormente, y se define según la relación:

15

$$Ciclos_reloj_Inhabilitar_{INV2} = \frac{VelocidadDatosEnlaceDirecto}{FactorTipoInterfaz_{DIR}} \cdot \text{retardoInhabilitarExcitadorDispositivoVisualización}_{\max}$$

y el intervalo de valores permitidos para Inversión 2 se expresa como:

20

$$Inversión_2 \geq \text{redondearPorExcesoSiguieteEntero} \left(\frac{Ciclos_reloj_Inhabilitar_{INV2} + \text{retardo_ida_y_vuelta} + 1}{\left(\frac{8}{FactorTipoInterfaz_{DIR}} \right)} \right)$$

Por ejemplo, un enlace directo de Tipo III de 1500 Mbps con un retardo de ida y vuelta de 10 ciclos de reloj de enlace directo utiliza normalmente un retardo Inversión 2 del orden de:

25

$$Ciclos_reloj_inhabilitar_{INV2} = \frac{1500 \text{ Mbps}}{4} \cdot 10 \text{ nseg} = 3,75$$

30

$$Inversión_2 \geq \text{redondearPorExcesoSiguieteEntero} \left(\frac{3,75 + 10 + 1}{\left(\frac{8}{4} \right)} \right) = 8$$

XII. Temporización alternativa de enlace inverso

35

Aunque el uso de las bandas de temporización y de vigilancia descritas anteriormente funciona para obtener una interfaz con una alta velocidad de transferencia de datos, se ha descubierto una técnica para permitir que las longitudes de bit inversas sean más cortas que el tiempo de ida y vuelta, modificando el descubrimiento de la temporización inversa.

40

Tal y como se ha mencionado anteriormente, el enfoque anterior relacionado con la temporización del enlace inverso está configurado de manera que el número de ciclos de reloj se cuente desde el último bit del Tiempo de Vigilancia 1 de un paquete de temporización inversa hasta que se muestree el primer bit en el flanco de subida de un reloj IO. Es decir, la(s) señal(es) de reloj utilizada(s) para controlar el tiempo de las entradas y salidas de la interfaz MDD. El cálculo del divisor de velocidad inversa viene dado por:

$$divisor_velocidad_inversa = \text{redondearPorExcesoSiguieteEntero} \left(\frac{\text{retardo_ida_y_vuelta} + 1}{2} \right)$$

Esto proporciona un ancho de bit igual al retardo de ida y vuelta, lo que da como resultado un enlace inverso muy fiable. Sin embargo, se ha demostrado que el enlace inverso puede funcionar más rápidamente, o a una mayor velocidad de transferencia de datos, lo cual debe aprovecharse. Una nueva técnica inventiva permite utilizar capacidades adicionales de la interfaz para obtener mayores velocidades.

5 Esto se consigue haciendo que el dispositivo central cuente el número de ciclos de reloj hasta que se muestree un uno, pero haciendo que el dispositivo central muestree la línea de datos tanto en el flanco de subida como en el flanco de descenso durante el paquete de temporización inversa. Esto permite al dispositivo central escoger el punto de muestreo más útil o incluso óptimo en el bit inverso para garantizar que el bit sea estable. Es decir, encontrar el flanco de subida más útil u óptimo para muestrear datos de paquetes de encapsulación inversos de tráfico inverso. El punto de muestreo óptimo depende del divisor de enlace inverso y de si el primer uno se detectó en un flanco de subida o en un flanco de descenso. El nuevo procedimiento de temporización permite al dispositivo central buscar simplemente el primer flanco del patrón 0xFF 0xFF 0x00 enviado por el dispositivo cliente para la temporización de enlace inverso para determinar dónde muestrear en un paquete de encapsulación inverso.

15 Ejemplos del bit inverso entrante y de cómo ese bit busca varios divisores de velocidad inversa se ilustran en la FIG. 64, junto con una pluralidad de ciclos de reloj que se han producido desde el último bit de Tiempo de Vigilancia 1. En la FIG. 64, puede observarse que si el primer flanco se produce entre un flanco de subida y un flanco de descenso (etiquetado como subida/descenso), el punto de muestreo óptimo para un divisor de velocidad inversa de uno, el punto de muestreo óptimo es un flanco de ciclo de reloj etiquetado como 'b', ya que es el único flanco de subida que se produce dentro del periodo del bit inverso. Para un divisor de velocidad inversa de dos, el punto de muestreo óptimo sigue siendo probablemente el flanco delantero 'b' de ciclo de reloj ya que el flanco de ciclo 'c' está más cerca de un flanco de bit que 'b'. Para un divisor de velocidad inversa de cuatro, el punto de muestreo óptimo es probablemente el flanco 'd' de ciclo de reloj ya que está más cerca del flanco trasero del bit inverso donde probablemente el valor se ha estabilizado.

25 Sin embargo, volviendo a la FIG. 64, si el primer flanco se produce entre un flanco de descenso y un flanco de subida (etiquetado como descenso/subida), el punto de muestreo óptimo para un divisor de velocidad inversa de uno es el punto de muestreo 'a' de flanco de ciclo de reloj ya que es el único flanco de subida dentro del periodo de tiempo de bit inverso. Para un divisor de velocidad inversa de dos, el punto de muestreo óptimo es el flanco 'b', y para un divisor de velocidad inversa de cuatro, el punto de muestreo óptimo es el flanco 'c'.

30 Puede observarse que a medida que los divisores de velocidad inversa son cada vez mayores, el punto de muestreo óptimo se vuelve más fácil de averiguar o seleccionar, y debe ser el flanco de subida que esté más cerca del centro.

35 El dispositivo central puede usar esta técnica para encontrar el número de flancos de subida de reloj antes del flanco de subida de datos de los datos de paquetes de temporización observados en la línea de datos. Después puede decidir, basándose en si el flanco se produce entre un flanco de subida y un flanco de descenso o entre un flanco de descenso y un flanco de subida, cuál es el divisor de velocidad inversa y cuántos ciclos de reloj adicionales añadir a un contador numérico para garantizar de manera razonable que el bit siempre se muestree lo más cerca posible del centro.

40 Una vez que el dispositivo central haya seleccionado o determinado el número de ciclos de reloj, puede "explorar" varios divisores de velocidad inversa con el dispositivo cliente para determinar si valdrá un divisor de velocidad inversa particular. El dispositivo central (y el dispositivo cliente) pueden comenzar con un divisor de uno y comprobar la CRC del paquete de estado inverso recibido desde el dispositivo cliente para determinar si esta velocidad inversa funciona de manera apropiada para transferir datos. Si la CRC está corrupta, probablemente hay un error de muestreo y el dispositivo central puede incrementar el divisor de velocidad inversa e intentar solicitar de nuevo un paquete de estado. Si el segundo paquete solicitado está corrupto, el divisor puede incrementarse de nuevo y volverse a realizar la solicitud. Si este paquete se descodifica correctamente, este divisor de velocidad inversa puede usarse para todos los futuros paquetes inversos.

45 Este procedimiento es eficaz y útil ya que la temporización inversa no debe cambiar con respecto a la estimación de temporización inicial de ida y vuelta. Si el enlace directo es estable, el dispositivo cliente debe continuar descodificando los paquetes de enlace directo incluso si hay fallos de enlace inverso. Por supuesto, el dispositivo central es todavía responsable de fijar un divisor de enlace inverso para el enlace, ya que este procedimiento no garantiza un enlace inverso perfecto. Además, el divisor dependerá principalmente de la calidad del reloj que se utilice para generar un reloj IO. Si ese reloj tiene una cantidad importante de fluctuaciones, hay una mayor probabilidad de un error de muestreo. Esta probabilidad de error aumenta con la cantidad de ciclos de reloj en el retardo de ida y vuelta.

50 Esta implementación parece que funciona mejor para los datos inversos de Tipo I, pero puede presentar problemas para los datos inversos de Tipo II a Tipo IV debido a que los desajustes entre las líneas de datos son posiblemente muy grandes para hacer funcionar el enlace a la velocidad que funciona mejor para un solo par de datos. Sin embargo, la velocidad de datos no necesita reducirse probablemente al procedimiento anterior incluso con los Tipos II a IV para el funcionamiento. Este procedimiento también puede funcionar mejor si se aplica del mismo modo en cada línea de datos para seleccionar la ubicación de muestreo de reloj ideal u óptima. Si están en el mismo tiempo de muestreo para cada par de datos, este procedimiento seguirá funcionando. Si están en diferentes periodos de muestreo, pueden utilizarse

dos enfoques diferentes. El primero es seleccionar una ubicación de muestreo deseada o más optimizada para cada punto de datos, incluso si no es la misma para cada par de datos. Después, el dispositivo central puede reconstruir el flujo de datos después de muestrear todos los bits del conjunto de pares de datos: dos bits para el Tipo II, cuatro bits para el Tipo III y ocho bits para el Tipo IV. La otra opción es que el dispositivo central incremente el divisor de velocidad inversa de manera que los bits de datos para cada par de datos puedan muestrearse en el mismo flanco de reloj.

XIII. Efectos de retardo y desajustes de enlace

El retardo diferencial en el enlace directo entre los pares de señal MDD_Data y la señal MDDI_Stb puede limitar la máxima velocidad de transferencia de datos posible a no ser que se utilice una compensación de retardo diferencial. Las diferencias de retardo que provocan los desajustes de temporización se deben a la lógica de controlador, a los excitadores y receptores de línea, y al cable y los conectores, tal y como se describe a continuación.

A. Análisis de temporización de enlace limitado por desajustes (Tipo I de MDDI)

1. Ejemplo de retardo y desajustes de un enlace de Tipo I

Un circuito de interfaz típico similar al mostrado en la FIG. 41 se muestra en la FIG. 57 para permitir un enlace de interfaz de Tipo I. En la FIG. 57, valores típicos o a modo de ejemplo para el retardo y desajustes de propagación se muestran para cada una de varias fases de procesamiento o de interfaz de un enlace directo de Tipo I MDDI. Los desajustes en el retardo entre la señal MDDI_Stb y la señal MDDI_Data0 hacen que el ciclo de trabajo del reloj de salida se distorsione. Los datos en la entrada D de la etapa de biestable de receptor (RXFF) que utiliza los biestables 5728, 5732, deben cambiar ligeramente después del flanco de reloj para que puedan muestrearse de manera fiable. La figura muestra dos líneas de retardo en cascada 5732a y 5732b que se utilizan para solucionar dos problemas diferentes relacionados con la creación de esta relación de temporización. En la implementación real, pueden combinarse en un único elemento de retardo.

La FIG. 58 ilustra señales Data, Stb y la temporización de recuperación de reloj en un enlace de Tipo I para un procesamiento de señales a modo de ejemplo a través de la interfaz.

El retardo diferencial total que es significativo surge o se obtiene generalmente a partir de la suma de los desajustes en las siguientes etapas: biestable de transmisor (TXFF) con biestables 5704, 5706; excitador de transmisor (TXDRVR) con excitadores 5708, 5710; el cable 5702; receptor de línea de recepción (RXRCVR) con receptores 5722, 5724; y lógica XOR de receptor (RXXOR). El Retardo 1 5732a debe coincidir con o superar el retardo de la puerta XOR 5736 en la etapa RXXOR que se determina mediante la relación:

$$t_{PD-\min(\text{retard}d)} \geq t_{PD-\max(XOR)}$$

Es deseable cumplir este requisito para que la entrada D del biestable de receptor 5728, 5732 no cambie antes de su entrada de reloj. Esto es válido si el tiempo de retención de RXFF es cero.

La finalidad o función de Retardo 2 es compensar el tiempo de retención del biestable RXFF según la relación:

$$t_{PD-\min(\text{retard}2)} = t_{H(RXFF)}$$

En muchos sistemas esto será cero porque el tiempo de retención es cero y, obviamente, en ese caso, el retardo máximo de Retardo 2 también puede ser cero.

El peor caso de contribución a los desajustes en la etapa XOR de receptor está en el caso 'señales de datos tardías/señales estroboscópicas tempranas', donde el Retardo 1 está a un valor máximo y la salida de reloj de la puerta XOR llega lo antes posible según la relación:

$$t_{DESAJUSTES-\max(RXXOR)} = t_{PD-\max(\text{retard}d)} - t_{PD-\min(XOR)}$$

En esta situación, los datos pueden cambiar entre dos periodos de bit, n y n+1, muy cerca del instante en que el bit n+1 está sincronizado en el biestable de receptor.

La velocidad de transferencia de datos máxima (periodo de bit mínimo) de un enlace de Tipo I MDDI es una función del máximo desajuste encontrado en todos los excitadores, cables y receptores del enlace MDDI más la configuración de datos total en la etapa RXFF. El retardo diferencial total en el enlace hasta la salida de la etapa RXRCVR puede expresarse como:

$$t_{DESAJUSTES\text{-max}(ENLACE)} = t_{DESAJUSTES\text{-max}(TXFF)} + t_{DESAJUSTES\text{-max}(TXDRVR)} + t_{DESAJUSTES\text{-max}(CABLE)} + t_{DESAJUSTES\text{-max}(RXRCVR)}$$

y el periodo de bit mínimo viene dado por:

$$t_{BIT\text{-min}} = t_{DESAJUSTES\text{-max}(ENLACE)} + t_{DESAJUSTES\text{-max}(RXXOR)} + t_{PD\text{-max}(retard\delta)} + t_{SU}(RXFF)$$

En el ejemplo mostrado en la FIG. 57, $t_{DESAJUSTES\text{-max}(ENLACE)} = 1,4$ nseg y el periodo de bit mínimo puede expresarse como:

$$t_{BIT\text{-min}} = 1,4 + 0,3 + 0,2 + 0,5 = 2,4nseg$$

o fijarse a 416 Mbps aproximadamente.

B. Análisis de temporización de enlace para los Tipos II, III y IV MDDI

Un circuito de interfaz típico similar al mostrado en las FIG. 41 y 57 se muestra en la FIG. 59 para permitir enlaces de interfaz de Tipo II, III y IV. En las etapas TXFF (5904), TXDRVR (5908), RXRCVCR (5922) y RXFF (5932, 5928, 5930) se utilizan elementos adicionales para permitir un procesamiento de señales adicional. En la FIG. 59, valores típicos o a modo de ejemplo para el retardo y desajustes de propagación se muestran para cada una de varias etapas de procesamiento o de interfaz de un enlace directo MDDI de Tipo II. Además de los desajustes en el retardo entre la señal MDDI_Stb y la señal MDDI_Data0 que afectan al ciclo de trabajo del reloj de salida, también hay desajustes entre estas dos señales y las otras señales MDDI_Data. Los datos en la entrada D de la etapa de biestable B de receptor (RXFFB) que consiste en los biestables 5928 y 5930, se modifican ligeramente después del flanco de reloj para que puedan muestrearse de manera más fiable. Si la señal MDDI_Data1 llega antes que la señal MDDI_Stb o MDDI_Data0, entonces la señal MDDI_Data1 debe retardarse para muestrearse por al menos la cantidad del retardo diferencial. Para conseguir esto, los datos se retardan utilizando la línea de retardo Retardo3. Si la señal MDDI_Data1 llega más tarde que las señales MDDI_Stb y MDDI_Data0 y también está retardada por el Retardo3, entonces el instante en que la señal MDDI_Data1 cambia se acerca al siguiente flanco de reloj. Este proceso determina un límite superior de la velocidad de transferencia de datos de un enlace MDDI de Tipo II, III o IV. Algunas posibilidades diferentes a modo de ejemplo para la relación de temporización o de desajustes entre dos señales de datos y la señal MDDI_Stb se ilustran en las FIG. 60A, 60B y 60C.

Con el fin de muestrear datos de manera fiable en RXFFB cuando la señal MDDI_DataX llega lo más pronto posible, Retardo3 se fija según la relación:

$$t_{PD\text{-min}(retard\delta)} \geq t_{DESAJUSTES\text{-max}(ENLACE)} + t_H(RXFFB) + t_{PD\text{-max}(XOR)}$$

La velocidad de enlace máxima se determina por el mínimo periodo de bit permitido. Esto se ve más afectado cuando la señal MDDI_DataX llega lo más tarde posible. En ese caso, el mínimo tiempo de ciclo permitido viene dado por:

$$t_{BIT\text{-min}} = t_{DESAJUSTES\text{-max}(ENLACE)} + t_{PD\text{-max}(retard\delta)} + t_{SU}(RXFFB) - t_{PD\text{-min}(XOR)}$$

El límite superior de la velocidad de enlace es entonces:

$$t_{PD\text{-max}(retard\delta)} = t_{PD\text{-min}(retard\delta)}$$

y de ese modo:

$$t_{BIT\text{-min}(Limite\text{-Inferior})} = 2 \cdot t_{DESAJUSTES\text{-max}(ENLACE)} + t_{PD\text{-max}(XOR)} + t_{SU}(RXFFB) + t_H(RXFFB)$$

En el ejemplo mostrado anteriormente, el límite inferior del mínimo periodo de bit viene dado por la relación:

$$t_{BIT\text{-min}(Nivel\text{-Inferior})} = 2 \cdot 1,4 + 1,5 + 0,5 + 0,1 = 4,8nseg$$

que son 208 Mbps aproximadamente.

Esto es mucho más lento que la máxima velocidad de transferencia de datos que puede utilizarse con un enlace de Tipo I. La capacidad de compensación automática de retardo diferencial de la MDDI reduce significativamente el efecto que tiene el retardo diferencial en la velocidad de enlace máxima.

XIV. Descripción de interconexiones de capa física

Las conexiones físicas útiles para implementar una interfaz según la presente invención pueden realizarse utilizando piezas disponibles comercialmente tal como el número de pieza 3260-8S2(01) fabricada por Hirose Electric Company Ltd. en el lado de dispositivo central, y el número de pieza 3240-8P-C fabricada por Hirose Electric Company Ltd. en el lado de dispositivo de visualización. Una asignación de clavijas de interfaz a modo de ejemplo o "configuración de clavijas" para tales conectores utilizados con interfaces de Tipo I y de Tipo II se enumera en la tabla XIII y se ilustra en la FIG. 61.

Tabla XIII

Nombre de señal	Número de clavija	Color	Nombre de señal	Número de clavija	Color
MDDI_Pwr	1	Rojo	MDDI_Gnd	2	Negro emparejado con rojo
MDDI_Stb+	3	Verde	MDDI_Stb-	4	Negro emparejado con verde
MDDI_Data0+	5	Azul	MDDI_Data0-	6	Negro emparejado con azul
MDDI_Data1+	7	Blanco	MDDI_Data1-	8	Negro emparejado con blanco
			Blindaje		

El blindaje está conectado a la línea MDDI_Gnd en la interfaz de dispositivo central, y un hilo de drenaje y blindaje del cable está conectado al blindaje del conector del dispositivo de visualización. Sin embargo, el hilo de drenaje y blindaje no está conectado a la conexión a tierra de circuito de un dispositivo de visualización.

Los elementos o dispositivos de interconexión se eligen o se diseñan con el fin de que sean lo bastante pequeños como para utilizarse con dispositivos informáticos y de comunicaciones móviles, tales como PDA y teléfonos inalámbricos, o dispositivos de juego portátiles, sin ser llamativos o antiestéticos en comparación con un tamaño de dispositivo relativo. Cualquier conector y cableado debe ser lo bastante duradero como para utilizarse en el entorno de consumo típico y permitir un tamaño reducido, especialmente para el cableado, así como un coste relativamente bajo. Los elementos de transferencia deben permitir señales de datos y estroboscópicas que sean datos NRZ diferenciales que tengan una velocidad de transferencia de hasta 450 Mbps aproximadamente para el Tipo I y el Tipo II y de hasta 3,6 Gbps para la versión de Tipo IV en paralelo de 8 bits.

XV. Funcionamiento

Un resumen de las etapas generales llevadas a cabo en el procesamiento de datos y paquetes durante el funcionamiento de una interfaz que utiliza las realizaciones de la invención se muestra en las FIG. 54A y 54B, junto con una visión general del aparato de interfaz que procesa los paquetes en la FIG. 55. En estas figuras, el proceso comienza en una etapa 5402 con la determinación de si el dispositivo cliente y el dispositivo central están conectados o no utilizando una trayectoria de comunicaciones, en este caso un cable. Esto puede llevarse a cabo utilizando un sondeo periódico por parte del dispositivo central, utilizando software o hardware que detecte la presencia de conectores o cables o señales en las entradas del dispositivo central (tal y como se realiza para las interfaces USB), u otras técnicas conocidas. Si no hay ningún dispositivo cliente conectado al dispositivo central, entonces puede entrar simplemente en un estado de espera de una longitud predeterminada, dependiendo de la aplicación, pasar a un modo de hibernación, o estar inactivo para esperar un uso posterior, lo que podría requerir que un usuario lleve a cabo una acción para volver a activar el dispositivo central. Por ejemplo, cuando un dispositivo central reside en un dispositivo de tipo ordenador, un usuario puede tener que hacer clic sobre un icono de la pantalla o solicitar que un programa active el procesamiento de dispositivo central para buscar el dispositivo cliente. Nuevamente, enchufando simplemente una conexión de tipo USB, tal como la utilizada para la interfaz de Tipo U, se puede activar el procesamiento de dispositivo central, dependiendo de las capacidades y de la configuración del dispositivo central o del software que reside en el dispositivo central.

Una vez que un cliente se haya conectado al dispositivo central, o viceversa, o se detecte su presencia, el dispositivo cliente o el dispositivo central envía paquetes apropiados que solicitan servicio en las etapas 5404 y 5406. El dispositivo cliente puede enviar Paquetes de Estado o de Solicitud de Servicios de Dispositivo de Visualización en la etapa 5404. Debe observarse que el enlace, tal y como se ha mencionado anteriormente, puede haberse interrumpido anteriormente o estar en un modo de hibernación, por lo que esto puede no ser una inicialización completa del enlace de

comunicaciones resultante. Una vez que el enlace de comunicaciones esté sincronizado y el dispositivo central esté intentando comunicarse con el dispositivo cliente, el dispositivo cliente también proporciona un Paquete de Capacidades de Dispositivo de Visualización al dispositivo central en la etapa 5408. El dispositivo central puede empezar a determinar en este momento el tipo de soporte, incluyendo las velocidades de transferencia, permitido por el dispositivo cliente.

5 Generalmente, el dispositivo central y el dispositivo cliente también negocian el tipo (tasa/velocidad) del modo de servicio que va a utilizarse, por ejemplo el Tipo I, el Tipo U, el Tipo II, etc., en una etapa 5410. Una vez que se haya establecido el tipo de servicio, el dispositivo central puede empezar a transferir información. Además, el dispositivo central puede utilizar Paquetes de Medición de Retardo de Ida y Vuelta para optimizar la temporización de los enlaces de comunicaciones en paralelo con otro procesamiento de señales, tal y como se muestra en la etapa 5411.

10 Como se ha indicado anteriormente, todas las transferencias comienzan con un Paquete de Cabecera de Subtrama, el cual se muestra transfiriéndose en la etapa 5412, seguido del tipo de datos, en este caso paquetes de flujo de vídeo y audio, y de paquetes de relleno, los cuales se muestran transfiriéndose en la etapa 5414. Los datos de audio y vídeo se habrán preparado o correlacionado anteriormente en paquetes, y los paquetes de relleno se insertan según sea necesario o deseable para rellenar un número requerido de bits para las tramas multimedia. El dispositivo central puede enviar paquetes tales como Paquetes de Habilitación de Canal de Audio Directo para activar dispositivos de sonido. Además, el dispositivo central puede transferir comandos e información utilizando otros tipos de paquete descritos anteriormente, en este caso mostrados como la transferencia de los paquetes Mapa de Colores, Transferencia de Bloques de Bits u otros paquetes, en la etapa 5416. Además, el dispositivo cliente y el dispositivo central pueden intercambiar datos relacionados con un teclado o con dispositivos de puntero utilizando los paquetes apropiados.

20 Durante el funcionamiento, puede producirse uno de varios eventos diferentes que hace que el dispositivo central o el dispositivo cliente desee una velocidad de transferencia de datos o un tipo de modo de interfaz diferentes. Por ejemplo, un ordenador u otro dispositivo de comunicación de datos puede encontrar condiciones de carga en el procesamiento de datos que provoquen una reducción en la preparación o presentación de paquetes. Un dispositivo de visualización que reciba los datos puede cambiar una fuente de alimentación dedicada de CA por una fuente de alimentación de batería más limitada, y o bien no poder transferir los datos tan rápidamente, procesar comandos tan fácilmente, o no poder utilizar el mismo grado de resolución o profundidad de color con los ajustes de potencia más limitados. Como alternativa, una condición restrictiva puede reducirse o eliminarse permitiendo que cualquiera de los dispositivos transfiera datos a mayores velocidades. Al ser esto más deseable, puede crearse una solicitud para pasar a un modo con una mayor velocidad de transferencia de datos.

25 Si se producen o cambian estos y otros tipos de condiciones conocidas, el dispositivo central o el dispositivo cliente pueden detectarlos e intentar renegociar el modo de interfaz. Esto se muestra en la etapa 5420, donde el dispositivo central envía Paquetes de Solicitud de Traspaso de Tipo de Interfaz al dispositivo cliente que solicita un cambio a otro modo, el dispositivo cliente envía Paquetes de Confirmación de Recepción de Tipo de Interfaz que confirman que se está buscando un cambio, y el dispositivo central envía Paquetes de Traspaso de Tipo de Acción para realizar el cambio al modo especificado.

30 Aunque no se requiere un orden particular de procesamiento, el dispositivo cliente y el dispositivo central también pueden intercambiar paquetes relacionados con los datos destinados a o recibidos desde dispositivos de puntero, teclados u otros dispositivos de entrada de tipo usuario asociados principalmente con el dispositivo cliente, aunque tales elementos también pueden estar presentes en el lado del dispositivo central. Estos paquetes se procesan normalmente utilizando un elemento de tipo procesador general y no la máquina de estados (5502). Además, algunos de los comandos mencionados anteriormente también se procesarán mediante el procesador general (5504, 5508).

35 Después de que los datos y comandos se hayan intercambiado entre el dispositivo central y el dispositivo cliente, en algún momento se decide si van a transferirse o no datos adicionales o si el dispositivo central o el dispositivo cliente va a finalizar la transferencia. Esto se muestra en la etapa 5422. Si el enlace va a entrar en un estado de hibernación o va a interrumpirse completamente, el dispositivo central envía un Paquete de Interrupción de Enlace al dispositivo cliente y ambos lados finalizan la transferencia de datos.

40 Los paquetes que se transfieren en las anteriores operaciones de procesamiento se transferirán utilizando los excitadores y receptores descritos anteriormente con relación a los controladores de dispositivo central y dispositivo cliente. Estos excitadores de línea y otros elementos lógicos están conectados a la máquina de estados y a los procesadores generales descritos anteriormente, tal y como se ilustra en el esquema de la FIG. 55. En la FIG. 55, una máquina de estados 5502 y los procesadores generales 5504 y 5508 pueden conectarse además a otros elementos no mostrados tales como una interfaz USB dedicada, elementos de memoria u otros componentes que residan fuera del controlador de enlace con el que interactúan, incluyendo, pero sin limitarse a, la fuente de datos y chips de control de vídeo para dispositivos de visualización.

45 Los procesadores y la máquina de estados proporcionan control sobre la habilitación e inhabilitación de los excitadores descritos anteriormente con relación a los tiempos de vigilancia, etc., para garantizar un establecimiento o una finalización eficaces del enlace de comunicaciones y de la transferencia de paquetes.

XVI. Anexo

Además de los formatos, estructuras y contenidos descritos anteriormente para los diversos paquetes utilizados para implementar la arquitectura y el protocolo para realizaciones de la invención, a continuación se presentan los contenidos u operaciones de los campos en mayor detalle para algunos de los tipos de paquete. Se presentan para aclarar adicionalmente su uso u operaciones respectivos para permitir que los expertos en la técnica entiendan más fácilmente y utilicen la invención para una variedad de aplicaciones. Solamente se describen adicionalmente algunos de los campos que todavía no se han descrito en este documento. Además, estos campos se presentan con definiciones y valores a modo de ejemplo con relación a las realizaciones presentadas anteriormente. Sin embargo, no debe considerarse que tales valores limitan la invención, sino que representan una o más realizaciones útiles para implementar la interfaz y el protocolo, y no es necesario que todas las realizaciones se lleven a la práctica conjuntamente o al mismo tiempo. Pueden usarse otros valores en otras realizaciones para conseguir la presentación deseada de datos o de resultados de velocidades de transferencia de datos, tal y como entenderán los expertos en la técnica.

A. Para Paquetes de Flujo de Vídeo

En una realización, el campo Atributos de Dispositivo de Visualización (1 byte) tiene una serie de valores de bit que se interpretan de la siguiente manera. Los bits 1 y 0 seleccionan cómo se encaminan los datos de píxel de visualización. Para los valores de bit de '00' o '11' los datos se muestran para los dos ojos, para los valores de bit '10' los datos se encaminan solamente hacia el ojo izquierdo, y para los valores de bit '01' los datos se encaminan solamente hacia el ojo derecho. El bit 2 indica si los Datos de Píxel se presentan o no en un formato entrelazado, donde el valor '0' significa que los datos de píxel están en el formato progresivo estándar y que el número de fila (coordenada Y de píxel) se incrementa en 1 cuando se avanza desde una fila a la siguiente. Cuando este bit tiene el valor '1', los datos de píxel están en el formato entrelazado, y el número de fila se incrementa en 2 cuando se avanza de una fila a la siguiente. El bit 3 indica que los Datos de Píxel están en el formato de píxel alternativo. Éste es similar al modo entrelazado estándar habilitado por el bit 2, pero el entrelazado es vertical en lugar de horizontal. Cuando el bit 3 vale 0, los Datos de Píxel están en el formato progresivo estándar y el número de columna (coordenada X de píxel) se incrementa en 1 cuando se recibe cada píxel sucesivo. Cuando el bit 3 vale '1', los Datos de Píxel están en un formato de píxel alternativo, y el número de columna se incrementa en 2 cuando se recibe cada píxel. Los bits 7 a 4 están reservados para un uso futuro y están fijados generalmente a cero.

Los campos Inicio X e Inicio Y de dos bytes especifican las coordenadas X e Y absolutas del punto (Inicio X, Inicio Y) para el primer píxel del campo Datos de Píxel. Los campos Borde Izquierdo X y Borde Superior Y de 2 bytes especifican la coordenada X del borde izquierdo y la coordenada Y del borde superior de la ventana de visualización rellena por el campo Datos de Píxel, mientras que los campos Borde Derecho X y Borde Inferior Y especifican la coordenada X del borde derecho y la coordenada Y del borde inferior de la ventana que está actualizándose.

El campo Cómputo de Píxeles (2 bytes) especifica el número de píxeles del campo Datos de Píxel posterior.

El campo CRC de Parámetros (2 bytes) contiene una CRC de todos los bytes desde Longitud de Paquete hasta el Cómputo de Píxeles. Si esta CRC proporciona un valor incorrecto, se descarta todo el paquete.

El campo Datos de Píxel contiene información de vídeo sin procesar que va a visualizarse, la cual está formateada de la manera descrita por el campo Descriptor de Formato de Datos de Vídeo. Los datos se transmiten una "fila" a la vez tal y como se ha mencionado en este documento.

El campo CRC de Datos de Píxel (2 bytes) contiene una CRC de 16 bits de solamente los Datos de Píxel. Si una verificación CRC de este valor falla, entonces los Datos de Píxel pueden seguir utilizándose pero se incrementa el cómputo de errores de CRC.

B. Para Paquetes de Flujo de Audio

En una realización, el campo ID de Canal de Audio (1 byte) identifica un canal de audio particular al que el dispositivo cliente envía datos de audio. Los canales de audio físicos se especifican o se correlacionan mediante este campo como valores de 0, 1, 2, 3, 4, 5, 6 ó 7 que indican el canal delantero izquierdo, el canal delantero derecho, el canal trasero izquierdo, el canal trasero derecho, el canal central delantero, el canal de bajos, el canal envolvente izquierdo y el canal envolvente derecho, respectivamente. Un valor 254 de ID de canal de audio indica que el único flujo de muestras de audio digital se envía tanto al canal delantero izquierdo como al canal delantero derecho. Esto simplifica las aplicaciones en las que se utiliza un auricular estéreo para la comunicación de voz, se utilizan aplicaciones de mejora de productividad en un PDA, u otras aplicaciones donde una Interfaz de Usuario simple genera tonos de aviso. Los valores del campo ID que van del 8 al 253, y el 255, están reservados actualmente para utilizarse cuando nuevos diseños deseen designaciones adicionales.

El campo Cómputo de Muestras de Audio (2 bytes) especifica el número de muestras de audio en este paquete.

El campo Bits por Muestra y Empaquetado contiene 1 byte que especifica el formato de avance de los datos de

5 audio. El formato utilizado generalmente es uno en el que los bits 4 a 0 definen el número de bits por muestra de audio PCM. El bit 5 especifica si las muestras de Datos de Audio Digitales están empaquetadas o no. Tal y como se mencionado anteriormente, la FIG. 12 ilustra la diferencia entre muestras de audio empaquetadas y alineadas por bytes. Un valor de '0' para el bit 5 indica que cada muestra de audio PCM del campo Datos de Audio Digitales está alineada por bytes con el límite de byte de interfaz, y un valor de '1' indica que cada muestra de audio PCM sucesiva está empaquetada contra la muestra de audio anterior. Este bit solo es eficaz cuando el valor definido en los bits 4 a 0 (el número bits por muestra de audio PCM) no es un múltiplo de ocho. Los bits 7 a 6 están reservados para utilizarse cuando los diseños de sistema deseen designaciones adicionales y están fijados generalmente a un valor de cero.

10 El campo Frecuencia de Muestreo de Audio (1 byte) especifica la frecuencia de muestro PCM de audio. El formato utilizado es uno en el que un valor de 0 indica una frecuencia de 8.000 muestras por segundo (sps), un valor de 1 indica 16.000 sps, un valor de 2 indica 24.000 sps, un valor de 3 indica 32.000 sps, un valor de 4 indica 40.000 sps, un valor de 5 indica 48.000 sps, un valor de 6 indica 11.025 sps, un valor de 7 indica 22.050 sps y un valor de 8 indica 44.100 sps, respectivamente, donde los valores de 9 a 15 están reservados para un uso futuro, por lo que está fijados actualmente a cero.

15 El campo CRC de Parámetros (2 bytes) contiene una CRC de 16 bits de todos los bytes desde Longitud de Paquete hasta la Frecuencia de Muestreo de Audio. Si esta CRC no proporciona una comprobación correcta, entonces se descarta todo el paquete. El campo Datos de Audio Digitales contiene las muestras de audio sin procesar que van a reproducirse y tiene normalmente un formato lineal a modo de enteros sin signo. El campo CRC de Datos de Audio (2 bytes) contiene una CRC de 16 bits de solo los Datos de Audio. Si esta CRC falla en su comprobación, entonces los Datos de Audio pueden seguir utilizándose pero se incrementa el cómputo de errores de CRC.

20 C. Para Paquetes de Flujo Definidos por el Usuario

25 En una realización, el campo Número de ID de Flujo de 2 bytes se utiliza para identificar un flujo particular definido por el usuario. Los contenidos de los campos Parámetros de Flujo y Datos de Flujo se definen normalmente por el fabricante de equipos MDDI. El campo CRC de Parámetros de Flujo de 2 bytes contiene una CRC de 16 bits de todos los bytes de los parámetros de flujo empezando desde la Longitud de Paquete hasta el byte de Codificación de Audio. Si esta CRC falla en su comprobación, entonces se descarta todo el paquete. El campo CRC de Datos de Flujo de 2 bytes contiene una CRC de solo los Datos de Flujo. Si esta CRC no proporciona una comprobación correcta, entonces la utilización de los Datos de Flujo es opcional, dependiendo de los requisitos de la aplicación. La utilización de los datos de flujo que dependen de una CRC correcta, requiere generalmente que los datos de flujo se almacenen en una memoria intermedia hasta que se confirme que la CRC es correcta. El cómputo de errores CRC se incrementa si la comprobación CRC no es correcta.

30 D. Para Paquetes de Mapa de Colores

35 El campo Tamaño de Datos de Mapa de Colores (2 bytes) especifica el número total de entradas de tabla de mapa de colores que existen en los Datos de Mapa de Colores de este paquete. En esta realización, el número de bytes de los Datos de Mapa de Colores es 3 veces el Tamaño de Mapa de Colores. El Tamaño de Mapa de Colores se fija a cero para no enviar datos de mapa de colores. Si el Tamaño de Mapa de Colores es cero, entonces se envía generalmente un valor Desplazamiento de Mapa de Colores, pero es ignorado por el dispositivo de visualización. El campo Desplazamiento de Mapa de Colores (2 bytes) especifica el desplazamiento de los Datos de Mapa de Colores en este paquete desde el principio de la tabla de mapa de colores del dispositivo de visualización.

40 Un campo CRC de Parámetros de 2 bytes contiene una CRC de todos los bytes desde la Longitud de Paquete hasta el byte Codificación de Audio. Si esta CRC falla en su comprobación, entonces se descarta todo el paquete.

45 Para el campo Datos de Mapa de Colores, cada ubicación del mapa de colores es un valor de 3 bytes, donde el primer byte especifica la magnitud de azul, el segundo byte especifica la magnitud de verde y el tercer byte especifica la magnitud de rojo. El campo Tamaño de Mapa de Colores especifica el número de elementos de tabla de mapa de colores de 3 bytes que existen en el campo Datos de Mapas de Colores. Si un único mapa de colores no cabe en un Paquete de Mapa de Colores y de Formato de Datos de Vídeo, entonces la totalidad del mapa de colores puede especificarse enviando múltiples paquetes con diferentes Desplazamientos de Mapa de Colores y diferentes Datos de Mapa de Colores en cada paquete.

50 Un campo CRC de Datos de Mapa de Colores de 2 bytes contiene una CRC de solamente los Datos de Mapa de Colores. Si esta CRC falla en su comprobación, entonces los Datos de Mapa de Colores pueden seguir utilizándose pero se incrementa el cómputo de errores de CRC.

E. Para Paquetes de Encapsulación de Enlace Inverso

55 En una realización, el campo Banderas de Enlace Inverso (1 byte) contiene un conjunto de banderas para solicitar información del dispositivo de visualización. Si un bit (por ejemplo, el bit 0) está fijado a uno, entonces el dispositivo central solicita la información especificada del dispositivo de visualización utilizando el Paquete de Capacidades de Dispositivo de

Visualización. Si el bit vale cero, entonces el dispositivo central no necesita la información del dispositivo de visualización. Los bits restantes (en este caso, los bits 1 a 7) están reservados para un uso futuro y están fijados a cero. Sin embargo, pueden utilizarse más bits según se desee para fijar banderas para el enlace inverso.

5 El campo Divisor de Velocidad Inversa (1 byte) especifica el número de ciclos MDDI_Stb que se producen con relación al reloj de datos de enlace inverso. El reloj de datos de enlace inverso es igual al reloj de datos de enlace directo dividido por dos veces el Divisor de Velocidad Inversa. La velocidad de transferencia de datos de enlace inverso está relacionada con el reloj de datos de enlace inverso y con el Tipo de Interfaz del enlace inverso. Para una interfaz de Tipo I, la velocidad de transferencia de datos inversa es igual al reloj de datos de enlace inverso; para las interfaces de Tipo II, Tipo III y Tipo IV, las velocidades de transferencia de datos inversas son iguales a dos veces, cuatro veces y ocho veces el reloj de datos de enlace inverso, respectivamente.

10 El campo Longitud de Inversión 1 (1 byte) especifica el número total de bytes que están asignados para la Inversión 1. La longitud recomendada de Inversión 1 es el número de bytes requeridos para los excitadores de señal MDDI_Data en un dispositivo central que tiene las salidas inhabilitadas. Esto se basa en el tiempo de inhabilitación de salida descrito anteriormente, en la velocidad de transferencia de datos de enlace directo y en la selección de Tipo de Interfaz de enlace directo que estén utilizándose. Una descripción más completa del ajuste de Inversión 1 se ha proporcionado anteriormente.

15 El campo Longitud de Inversión 2 (1 byte) especifica el número total de bytes que están asignados a la Inversión. La longitud recomendada de Inversión 2 es el número de bytes requeridos para los excitadores de señal MDDI_Data en el dispositivo de visualización para inhabilitar sus salidas más el retardo de ida y vuelta. Una descripción del ajuste de Inversión 2 se ha proporcionado anteriormente.

20 El campo CRC de Parámetros (2 bytes) contiene una CRC de 16 bits para todos los bytes desde la Longitud de Paquete hasta la Longitud de Inversión. Si esta CRC falla en su comprobación, entonces se descarta todo el paquete.

25 El campo Todo a Cero (1 byte) está fijado igual a cero y se utiliza para garantizar que todas las señales MDDI_Data estén en el estado de cero antes de inhabilitar los excitadores de línea durante el primer periodo Tiempo de Vigilancia.

30 El campo Inversión 1 se utiliza para establecer el primer periodo de inversión. El número de bytes especificado por el parámetro Longitud de Inversión está asignado por este campo para permitir que los excitadores de línea MDDI_Data del dispositivo central se inhabiliten antes de que los excitadores de línea del dispositivo cliente (de visualización) se habiliten. El dispositivo central inhabilita sus excitadores de línea MDDI_Data durante el bit 0 de Inversión 1 y el dispositivo cliente (de visualización) habilita sus excitadores de línea inmediatamente después del último bit de Inversión 1. La señal MDDI_Stb se comporta como si el periodo de Inversión fuera todo ceros.

El campo Paquetes de Datos Inversos contiene una serie de paquetes de datos que se transfieren desde el dispositivo cliente hasta el dispositivo central. Tal y como se ha indicado anteriormente, paquetes de Relleno se envían para rellenar el espacio restante que no se utiliza por otros tipos de paquetes.

35 El campo Inversión 2 se utiliza para establecer el segundo periodo de inversión. El número de bytes especificado por el parámetro Longitud de Inversión está asignado por este campo.

El campo Rehabilitación de Excitadores utiliza 1 byte que es igual a cero para garantizar que todas las señales MDDI_Data vuelvan a habilitarse antes del campo Longitud de Paquete del siguiente paquete.

F. Para Paquetes de Capacidades de Dispositivo de Visualización

40 En una realización, el campo Versión de Protocolo utiliza 2 bytes para especificar la versión de protocolo utilizada por el dispositivo cliente. La versión inicial se fija igual a cero, mientras que el campo Versión Mínima de Protocolo utiliza 2 bytes para especificar la versión de protocolo mínima que el dispositivo cliente puede emplear o interpretar. El campo Capacidad de Velocidad de Transferencia de Datos de Dispositivo de Visualización (2 bytes) especifica la velocidad máxima de transferencia de datos que el dispositivo de visualización puede recibir en el enlace directo de la interfaz, y está especificado en forma de megabits por segundo (Mbps). El campo Capacidad de Tipo de Interfaz (1 byte) especifica los tipos de interfaz que se soportan en los enlaces directo e inverso. Esto se indica actualmente seleccionando el bit 0, el bit 1 o el bit 2 para seleccionar el modo de Tipo II, de Tipo III o de Tipo IV en el enlace directo, respectivamente, y el bit 3, el bit 4 o el bit 5 para seleccionar el modo de Tipo II, de Tipo III o de Tipo IV en el enlace inverso, respectivamente; los bits 6 y 7 están reservados y fijados a cero. Los campos Anchura y Altura de Mapa de Bits (2 bytes) especifican la anchura y la altura del mapa de bits en píxeles.

50 El campo de Capacidad Monocromo (1 byte) se utiliza para especificar el número de bits de resolución que pueden visualizarse en un formato monocromo. Si un dispositivo de visualización no puede utilizar un formato monocromo, entonces este valor está fijado a cero. Los bits 7 a 4 están reservados para un uso futuro y, por lo tanto, están fijados a cero. Los bits 3 a 0 definen el número máximo de bits de escala de grises que puede haber para cada píxel. Estos cuatro

bits hacen posible especificar valores de 1 a 15 para cada píxel. Si el valor es cero, entonces el dispositivo de visualización no soporta el formato monocromo.

5 El campo Capacidad de Mapa de Colores (3 bytes) especifica el número máximo de elementos de tabla que hay en la tabla de mapa de colores del dispositivo de visualización. Si el dispositivo de visualización no puede utilizar el formato de mapa de colores, entonces este valor es cero.

10 El campo Capacidad RGB (2 bytes) especifica el número de bits de resolución que pueden visualizarse en el formato RGB. Si el dispositivo de visualización no puede utilizar el formato RGB, entonces este valor es igual a cero. La palabra de Capacidad RGB está compuesta por tres valores sin signo distintos: los bits 3 a 0 definen el número máximo de bits de azul, los bits 7 a 4 definen el número máximo de bits de verde y los bits 11 a 8 definen el número máximo de bits de rojo en cada píxel. Actualmente, los bits 15 a 12 están reservados para un uso futuro y están fijados generalmente a cero.

15 El Campo Capacidad Y Cr Cb (2 bytes) especifica el número de bits de resolución que pueden visualizarse en el formato Y Cr Cb. Si el dispositivo de visualización no puede utilizar el formato Y Cr Cb, entonces este valor está fijado igual a cero. La palabra de Capacidad Y Cr Cb está compuesta por tres valores sin signo distintos: los bits 3 a 0 definen el número máximo de bits en la muestra Cb, los bits 7 a 4 definen el número máximo de bits en la muestra Cr, los bits 11 a 8 definen el número máximo de bits en la muestra Y, y los bits 15 a 12 están actualmente reservados para un uso futuro y están fijados a cero.

20 El campo Indicadores de Capacidades de Características de Dispositivo de Visualización utiliza 4 bytes que contienen un conjunto de banderas que indican características específicas que soporta el dispositivo de visualización. Un bit fijado a uno indica que se soporta la capacidad, y un bit fijado a cero indica que no se soporta la capacidad. El valor del bit 0 indica si se soporta o no el Paquete de Transferencia de Bloques de Mapa de Bits (tipo de paquete 71). El valor de los bits 1, 2 y 3 indica si se soportan o no el Paquete de Relleno de Área de Mapa de Bits (tipo de paquete 72), el Paquete de Relleno de Patrón de Mapa de Bits (tipo de paquete 73) o el Paquete de Canal de Datos de Enlace de Comunicaciones (tipo de paquete 74), respectivamente. El valor del bit 4 indica si el dispositivo de visualización tiene o no la capacidad de hacer un color transparente, mientras que los valores de los bits 5 y 6 indican si el dispositivo de visualización puede aceptar datos de vídeo o datos de audio en un formato paquetizado, respectivamente, y el valor del bit 7 indica si el dispositivo de visualización puede enviar un flujo de vídeo de enlace inverso desde una cámara. El valor de los bits 11 y 12 indican si el dispositivo cliente está comunicándose con un dispositivo de puntero, pudiendo enviar y recibir Paquetes de Datos de Dispositivo de Puntero, o con un teclado, pudiendo enviar y recibir Paquetes de Datos de Teclado, respectivamente. Los bits 13 a 31 están reservados actualmente para un uso futuro o para designaciones alternativas útiles para los diseñadores de sistemas, y están fijados generalmente a cero.

25 El campo Capacidad de Velocidad de Trama de Vídeo de Dispositivo de Visualización (1 byte) especifica la máxima capacidad de actualización de tramas de vídeo del dispositivo de visualización en tramas por segundo. Un dispositivo central puede elegir actualizar la imagen a una velocidad más baja que el valor especificado en este campo.

30 El campo Profundidad de Memoria Intermedia de Audio (2 bytes) especifica la profundidad de la memoria intermedia elástica en un dispositivo de visualización que está dedicado a cada flujo de audio.

35 El campo Capacidad de Canal de Audio (2 bytes) contiene un grupo de banderas que indican qué canales de audio soporta el dispositivo de visualización (cliente). Un bit fijado a uno indica que el canal se soporta y un bit fijado a cero indica que el canal no se soporta. Las posiciones de bit están asignadas a diferentes canales, por ejemplo las posiciones de bit 0, 1, 2, 3, 4, 5, 6 y 7 indican el canal delantero izquierdo, el canal delantero derecho, el canal trasero izquierdo, el canal trasero derecho, el canal delantero central, el canal de bajos, el canal envolvente izquierdo y el canal envolvente derecho, respectivamente. Los bits 8 a 15 están reservados actualmente para un uso futuro y están fijados generalmente a cero.

40 Un campo Capacidad de Frecuencia de Muestreo de Audio de 2 bytes, para el enlace directo, contiene un conjunto de banderas para indicar la capacidad de frecuencia de muestreo de audio del dispositivo cliente. Las posiciones de bit están asignadas por consiguiente a las diferentes frecuencias, de manera que los bits 0, 1, 2, 3, 4, 5, 6, 7 y 8 están asignados a 8.000, 16.000, 24.000, 32.000, 40.000, 48.000, 11.025, 22.050 y 44.100 muestras por segundo (SPS), respectivamente, donde los bits 9 a 15 están reservados para un uso futuro o para usos de frecuencias alternativas, según se desee, por lo que actualmente están fijados a '0'. Fijar un valor de bit para uno de estos bits a '1' indica que se soporta esa frecuencia de muestreo particular, y fijar el bit a '0' indica que no se soporta esa frecuencia de muestreo.

45 El campo Velocidad Mínima de Subtrama (2 bytes) especifica la velocidad mínima de subtrama en tramas por segundo. La velocidad mínima de subtrama mantiene una velocidad de actualización de estado de dispositivo de visualización suficiente para leer determinados sensores o dispositivos de puntero del dispositivo de visualización.

50 Un campo Capacidad de Frecuencia de Muestreo de Micrófono de 2 bytes, para el enlace inverso, contiene un conjunto de banderas que indican la capacidad de frecuencia de muestro de audio de un micrófono del dispositivo cliente. Para los propósitos de la MDDI, un micrófono de dispositivo cliente está configurado para soportar como mínimo al menos 8.000 muestras por segundo. Las posiciones de bit para este campo están asignadas a las diferentes frecuencias, donde

5 las posiciones de bit 0, 1, 2, 3, 4, 5, 6, 7 y 8, por ejemplo, se utilizan para representar 8.000, 16.000, 24.000, 32.000, 40.000, 48.000, 11.025, 22.050 y 44.100 muestras por segundo (SPS), respectivamente, donde los bits 9 a 15 están reservados para un uso futuro o para usos de frecuencias alternativas, según se desee, por lo que actualmente están fijados a '0'. Fijar un valor de bit para uno de estos bits a '1' indica que se soporta esa frecuencia de muestreo particular, y fijar el bit a '0' indica que no se soporta esa frecuencia de muestreo. Si no hay ningún micrófono conectado, entonces cada bit de Capacidad de Frecuencia de Muestreo de Micrófono está fijado a cero.

10 El campo Tipo de Protección de Contenido (2 bytes) contiene un conjunto de banderas que indican el tipo de protección de contenido digital soportado por el dispositivo de visualización. Actualmente, la posición de bit 0 se utiliza para indicar si se soporta DTCP y la posición de bit 1 se utiliza para indicar si se soporta HDCP, donde las posiciones 2 a 15 están reservadas para utilizarse con otros esquemas de protección según se desee o estén disponibles, por lo que actualmente están fijados a cero.

G. Para Paquetes de Estado y Solicitud de Dispositivo de Visualización

15 El campo Solicitud de Enlace Inverso (3 bytes) especifica el número de bytes que el dispositivo de visualización necesita en el enlace inverso en la siguiente subtrama para enviar información al dispositivo central.

20 El campo Cómputo de Errores de CRC (1 byte) indica cuántos errores CRC han ocurrido desde el inicio de la trama multimedia. El cómputo CRC se reajusta cuando se envía un paquete de cabecera de subtrama con un Cómputo de Subtrama de cero. Si el número real de errores CRC es mayor que 255, entonces este valor se satura generalmente en 255.

25 El campo Cambiar Capacidad utiliza en 1 byte para indicar un cambio en la capacidad del dispositivo de visualización. Esto puede producirse si un usuario conecta un dispositivo periférico, tal como un micrófono, un teclado o un dispositivo de visualización, o por alguna otra razón. Cuando los bits [7:0] son iguales a 0, entonces la capacidad no ha cambiado desde que se envió el último Paquete de Capacidades de Dispositivo de Visualización. Sin embargo, cuando los bits [7:0] tienen un valor comprendido entre 1 y 255, la capacidad ha cambiado. El Paquete de Capacidades de Dispositivo de Visualización se examina para determinar las nuevas características del dispositivo de visualización.

H. Para Paquetes de Transferencia de Bloques de Bits

30 Los campos Valor X y Valor Y de Coordenada Superior Izquierda de Ventana utilizan 2 bytes cada uno para especificar el valor X e Y de las coordenadas de la esquina superior izquierda de la ventana que va a moverse. Los campos Anchura y Altura de Ventana utilizan 2 bytes cada uno para especificar la anchura y la altura de la ventana que va a moverse. Los campos Movimiento X y Movimiento Y de Ventana utilizan 2 bytes cada uno para especificar el número de píxeles en que va a moverse la ventana horizontalmente y verticalmente, respectivamente. Normalmente, estas coordenadas están configuradas de manera que valores positivos de X hacen que la ventana se mueva hacia la derecha, y valores negativos provocan un movimiento hacia la izquierda, mientras que valores positivos de Y hacen que la ventana se mueva hacia abajo, y valores negativos provocan un movimiento hacia arriba.

I. Para Paquetes de Relleno de Área de Mapa de Bits

35 Los campos Valor X y Valor Y de Coordenada Superior Izquierda de Ventana utilizan 2 bytes cada uno para especificar el valor X e Y de las coordenadas de la esquina superior izquierda de la ventana que va a rellenarse. Los campos Anchura y Altura de Ventana (2 bytes cada uno) especifican la anchura y la altura de la ventana que va a rellenarse. El campo Descriptor de Formato de Datos de Vídeo (2 bytes) especifica el formato del Valor de Relleno de Área de Píxel. El formato es el mismo que el del mismo campo del Paquete de Flujo de Vídeo. El campo Valor de Relleno de Área de Píxel (4 bytes) contiene el valor de píxel que va a rellenarse en la ventana especificada por los campos descritos anteriormente. El formato de este píxel está especificado en el campo Descriptor de Formato de Datos de Vídeo.

J. Para Paquetes de Relleno de Patrón de Mapa de Bits

45 Los campos Valor X y Valor Y de Coordenada Superior Izquierda de Ventana utilizan 2 bytes cada uno para especificar el valor X e Y de las coordenadas de la esquina superior izquierda de la ventana que va a rellenarse. Los campos Anchura y Altura de Ventana (2 bytes cada uno) especifican la anchura y la altura de la ventana que va a rellenarse. Los campos Anchura de Patrón y Altura de Patrón (2 bytes cada uno) especifican la anchura y la altura, respectivamente, del patrón de relleno. El campo Descriptor de Formato de Datos de Vídeo de 2 bytes especifica el formato del Valor de Relleno de Área de Píxel. La FIG. 11 ilustra cómo está codificado el Descriptor de Formato de Datos de Vídeo. El formato es el mismo que el del mismo campo del Paquete de Flujo de Vídeo.

50 El campo CRC de Parámetros (2 bytes) contiene una CRC de todos los bytes desde la Longitud de Paquete hasta el Descriptor de Formato de Vídeo. Si esta CRC falla en su comprobación, entonces se descarta todo el paquete. El campo Datos de Píxel de Patrón contiene información de vídeo sin procesar que especifica el patrón de relleno en el formato especificado por el Descriptor de Formato de Datos de Vídeo. Los datos están empaquetados en bytes, y el primer píxel de cada fila debe estar alineado por bytes. Los datos de patrón de relleno se transmiten una fila cada vez. El campo CRC de

Datos de Píxel de Patrón (2 bytes) contiene una CRC de solamente los Datos de Píxel de Patrón. Si esta CRC falla en su comprobación, entonces los Datos de Píxel de Patrón pueden seguir usándose pero se incrementa el cómputo de errores de CRC.

K. Paquetes de Canal de Datos de Enlace de Comunicaciones

5 El campo CRC de Parámetros (2 bytes) contiene una CRC de 16 bits de todos los bytes desde la Longitud de Paquete hasta el Tipo de Paquete. Si esta CRC falla en su comprobación, entonces se descarta todo el paquete.

El campo Datos de Enlace de Comunicaciones contiene datos sin procesar del canal de comunicaciones. Estos datos se transmiten simplemente al dispositivo informático del dispositivo de visualización.

10 El campo CRC de Datos de Enlace de Comunicaciones (2 bytes) contiene una CRC de 16 bits de solamente los Datos de Enlace de Comunicaciones. Si esta CRC falla en su comprobación, entonces los Datos de Enlace de Comunicaciones se siguen utilizando o siguen siendo útiles, pero se incrementa el cómputo de errores de CRC.

L. Para Paquetes de Solicitud de Traspaso de Tipo de Interfaz

15 El campo Tipo de Interfaz (1 byte) especifica el nuevo tipo de interfaz a utilizar. El valor de este campo especifica el tipo de interfaz de la siguiente manera. Si el valor del bit 7 es igual a '0', la solicitud de cambio de tipo es para el enlace directo, si es igual a '1', entonces la solicitud de cambio de tipo es para el enlace inverso. Los bits 6 a 3 están reservados para un uso futuro y están fijados generalmente a cero. Los bits 2 a 0 se utilizan para definir el tipo de interfaz a utilizar, donde un valor de 1 significa un cambio al modo de Tipo I, un valor de 2 un cambio al modo de Tipo II, un valor de 3 un cambio al modo de tipo III y un valor de 4 un cambio al modo de Tipo IV. Los valores de '0' y 5 a 7 están reservados para una futura designación de modos alternativos o combinaciones de modos.

20 M. Para Paquetes de Confirmación de Recepción de Tipo de Interfaz

25 El campo Tipo de Interfaz (1 byte) tiene un valor que confirma el nuevo tipo de interfaz a utilizar. El valor en este campo especifica el tipo de interfaz de la siguiente manera. Si el bit 7 es igual a '0', la solicitud de cambio de tipo es para el enlace directo; alternativamente, si es igual a '1', la solicitud de cambio de tipo es para el enlace inverso. Las posiciones de bit 6 a 3 están reservadas actualmente para utilizarse en la designación de otros tipos de cambio, según se desee, y están fijadas generalmente a cero. Sin embargo, las posiciones de bit 2 a 0 se utilizan para definir el tipo de interfaz a utilizar, donde un valor de '0' indica una confirmación de recepción negativa, o que no puede llevarse a cabo el cambio solicitado, y los valores de '1', '2', '3' y '4' indican un cambio a los modos de Tipo I, Tipo II, Tipo III y Tipo IV, respectivamente. Los valores de 5 a 7 están reservados para utilizarse con designaciones alternativas de modos, según se desee.

N. Para Paquetes de Traspaso de Tipo de Acción

30 El campo Tipo de Interfaz de 1 byte indica el nuevo tipo de interfaz a utilizar. El valor presente en este campo especifica el tipo de interfaz utilizando en primer lugar el valor de bit 7 para determinar si el cambio de tipo es para los enlaces directo o inverso. Un valor de '0' indica que la solicitud de cambio de tipo es para el enlace directo, y un valor de '1' para el enlace inverso. Los bits 6 a 3 están reservados para un uso futuro y, por lo tanto, están fijados generalmente a un valor de cero. Sin embargo, los bits 2 a 0 se utilizan para definir el tipo de interfaz a utilizar, donde los valores 1, 2, 3 y 4 especifican un cambio a los modos de Tipo I, Tipo II, Tipo III y Tipo IV, respectivamente. El uso de los valores 0 y 5 a 7 para estos bits está reservado para un uso futuro.

O. Para Paquetes de Habilitación de Canal de Audio Directo

40 El campo Máscara de Habilitación de Canal de Audio (1 byte) contiene un grupo de banderas que indican qué canales de audio van a habilitarse en un dispositivo cliente. Un bit fijado a uno habilita el canal correspondiente, y un bit fijado a cero inhabilita el canal correspondiente. Los bits 0 a 5 designan canales 0 a 5 que se corresponden con el canal delantero izquierdo, el canal delantero derecho, el canal trasero izquierdo, el canal trasero derecho, el canal delantero central y el canal de bajos, respectivamente. Los bits 6 y 7 están reservados para un uso futuro y por el momento están fijados generalmente a cero.

P. Para Paquetes de Frecuencia de Muestreo de Audio Inverso

45 El campo Frecuencia de Muestreo de Audio (1 byte) especifica la frecuencia de muestreo de audio digital. Los valores para este campo están asignados a las diferencias frecuencias, donde los valores de 0, 1, 2, 3, 4, 5, 6, 7 y 8 se utilizan para designar 8.000, 16.000, 24.000, 32.000, 40.000, 48.000, 11.025, 22.050 y 44.100 muestras por segundo (SPS), respectivamente, donde los valores de 9 a 254 están reservados para utilizarse con frecuencias alternativas, según se desee, por lo que están fijados actualmente a '0'. Un valor de 255 se utiliza para inhabilitar el flujo de audio de enlace inverso.

50 El campo Formato de Muestra (1 byte) especifica el formato de las muestras de audio digitales. Cuando los bits [1:0] son iguales a '0', las muestras de audio digitales están en un formato lineal; cuando son iguales a 1, las muestras de

audio digitales están en el formato de Ley μ ; y cuando son iguales a 2, las muestras de audio digitales están en el formato de ley A. Los bits [7:2] están reservados para un uso alternativo en la designación de formatos de audio, según se desee, y están fijados generalmente a cero.

Q. Para los Paquetes de Datos Suplementarios de Protección de Contenido Digital

5 El campo Tipo de Protección de Contenido (1 byte) especifica el procedimiento de protección de contenido digital que está utilizándose. Un valor de '0' indica la protección de contenido de transmisión digital (DTCP), mientras que un valor de 1 indica el sistema de protección de contenido digital de gran ancho de banda (HDCP). El intervalo de valores de 2 a 255 no está especificado actualmente sino que está reservado para utilizarse con esquemas de protección alternativos, según se desee. El campo Mensajes de Datos Suplementarios de Protección de Contenido es un campo de longitud variable que contiene mensajes de protección de contenido enviados entre el dispositivo central y el dispositivo cliente.

R. Para Paquetes de Habilitación de Color Transparente

15 El campo Habilitación de Color Transparente (1 byte) especifica si el modo de color transparente está habilitado o inhabilitado. Si el bit 0 es igual a 0, entonces el modo de color transparente está inhabilitado, si es igual a 1, entonces el modo de color transparente está habilitado y el color transparente está especificado por los dos parámetros siguientes. Los bits 1 a 7 de este byte están reservados para un uso futuro y están fijados normalmente a cero.

El campo Descriptor de Formato de Datos de Vídeo (2 bytes) especifica el formato del Valor de Relleno de Área de Pixel. La FIG. 11 ilustra cómo está codificado el Descriptor de Formato de Datos de Vídeo. El formato es generalmente el mismo que el del mismo campo del Paquete de Flujo de Vídeo.

20 El campo Valor de Relleno de Área de Pixel utiliza 4 bytes asignados para el valor de píxel que va a rellenarse en la ventana especificada anteriormente. El formato de este píxel está especificado en el campo Descriptor de Formato de Datos de Vídeo.

S. Para Paquetes de Medición de Retardo de Ida y Vuelta

25 En una realización, el campo CRC de Parámetros (2 bytes) contiene una CRC de 16 bits de todos los bytes desde la Longitud de Paquete hasta el Tipo de Paquete. Si esta CRC falla en su comprobación, entonces se descarta todo el paquete.

El campo Todo a Cero (1 byte) contiene ceros para garantizar que todas las señales MDDI_Data estén en estado de cero antes de inhabilitar los excitadores de línea durante el primer periodo Tiempo de Vigilancia.

30 El campo Tiempo de Vigilancia 1 (8 bytes) se utiliza para permitir que los excitadores de línea MDDI_Data del dispositivo central se inhabiliten antes de que los excitadores de línea del dispositivo cliente (de visualización) se habiliten. El dispositivo central inhabilita sus excitadores de línea MDDI_Data durante el bit 0 de Tiempo de Vigilancia 1 y el dispositivo de visualización habilita sus excitadores de línea inmediatamente después del último bit de Tiempo de Vigilancia 1.

35 El campo Periodo de Medición es una ventana de 512 bytes utilizada para permitir que el dispositivo de visualización responda con 0xff, 0xff, 0x0 a la mitad de la velocidad de transferencia de datos utiliza en el enlace directo. Esta velocidad se corresponde con un Divisor de Velocidad Inversa de 1. El dispositivo de visualización devuelve esta respuesta inmediatamente al principio del Periodo de Medición. Esta respuesta se recibirá en un dispositivo central precisamente en el retardo de ida y vuelta del enlace después del principio del primer bit del Periodo de Medición en el dispositivo central. Los excitadores de línea MDDI_Data del dispositivo de visualización se inhabilitan justo antes y justo después de la respuesta 0xff, 0xff, 0x00 del dispositivo de visualización.

40 El valor del campo Tiempo de Vigilancia 2 (8 bytes) permite que los excitadores de línea MDDI_Data del dispositivo cliente se inhabiliten antes de que los excitadores de línea del dispositivo central se habiliten. El Tiempo de Vigilancia 2 está siempre presente pero solo se requiere cuando el retardo de ida y vuelta está en la cantidad máxima que puede medirse en el Periodo de Medición. El dispositivo cliente inhabilita sus excitadores de línea durante el bit 0 del Tiempo de Vigilancia 2 y el dispositivo central habilita sus excitadores de línea justo después del último bit del Tiempo de Vigilancia 2.

45 El campo Rehabilitación de Excitadores (1 byte) está fijado a cero para garantizar que todas las señales MDDI_Data vuelvan a habilitarse antes del campo Longitud de Paquete del siguiente paquete.

T. Para Paquetes de Calibración de Desajustes de Enlace Directo

50 En una realización, el campo CRC de Parámetros (2 bytes) contiene una CRC de 16 bits de todos los bytes desde la Longitud de Paquete hasta el Tipo de Paquete. Si esta CRC falla en su comprobación, entonces se descarta todo el paquete.

El campo Secuencia de Datos de Calibración contiene una secuencia de datos de 512 bytes que hace que las

señales MDDI_Data oscilen en cada periodo de datos. Durante el procesamiento de la Secuencia de Datos de Calibración, el controlador de dispositivo central MDDI iguala todas las señales MDDI_Data a la señal estroboscópica. El circuito de recuperación de reloj de dispositivo de visualización solo debe utilizar la señal MDDI_Stb en lugar del resultado de aplicar una función Xor a las señales MDDI_Stb y MDDI_Data0 para recuperar el reloj de datos cuando el dispositivo de visualización cliente está recibiendo el campo Secuencia de Datos de Calibración. Dependiendo de la fase exacta de la señal MDDI_Stb al principio del campo Secuencia de Datos de Calibración, la Secuencia de Datos de Calibración será generalmente una de las siguientes en función del tipo de interfaz que esté utilizándose cuando se envíe este paquete:

Tipo I - 0xaa, 0xaa ... ó 0x55, 0x55...

Tipo II - 0xcc, 0xcc ... ó 0x33, 0x33...

Tipo III - 0xf0, 0xf0 ... ó 0x0f, 0x0f ...

Tipo IV - 0xff, 0x00, 0xff, 0x00 ... ó 0x00, 0xff, 0x00, 0xff ...

Un ejemplo de las posibles formas de onda de las señales MDDI_Data y MDDI_Stb para las interfaces de Tipo I y de Tipo II se muestra en las FIG. 62A y 62B, respectivamente.

XVII. Conclusión

Aunque varias realizaciones de la presente invención se han descrito anteriormente, debe entenderse que se han presentado solamente a modo de ejemplo y de manera no limitativa. Por lo tanto, el espíritu y el alcance de la presente invención no deben limitarse por ninguna de las realizaciones descritas anteriormente a modo de ejemplo, sino que solo deben definirse según las siguientes reivindicaciones y sus equivalentes.

REIVINDICACIONES

1. Un procedimiento para transferir datos digitales a una alta velocidad entre un dispositivo central y un dispositivo cliente a través de una trayectoria de comunicaciones para su presentación a un usuario, que comprende:
- 5 generar una o más de una pluralidad de estructuras de paquete predefinidas y enlazarlas entre sí para formar un protocolo de comunicaciones predefinido;
- comunicar un conjunto preseleccionado de datos digitales de control y de presentación entre dicho dispositivo central y dicho dispositivo cliente a través de dicha trayectoria de comunicaciones utilizando dicho protocolo de comunicaciones;
- 10 acoplar al menos un controlador de enlace de dispositivo central que reside en dicho dispositivo central a dicho dispositivo cliente a través de dicha trayectoria de comunicaciones, estando configurado el controlador de enlace de dispositivo central para generar, transmitir y recibir paquetes que forman dicho protocolo de comunicaciones, y para formar datos digitales de presentación en uno o más de tipos de paquetes de datos;
- transferir datos en forma de paquetes a través de dicha trayectoria de comunicaciones utilizando dichos controladores de enlace;
- 15 y caracterizado por:
- activar un enlace de comunicaciones excitando una línea de datos a un estado alto durante al menos 150 ciclos de reloj y empezar a transmitir en un tiempo límite de 10 ciclos de reloj una señal estroboscópica como si la línea de datos estuviera a cero, mediante dicho dispositivo central; y
- 20 excitar la línea de datos a un estado bajo durante 50 ciclos de reloj mediante dicho dispositivo central mientras se continúa con la transmisión de una señal estroboscópica después de que el dispositivo central haya excitado la línea de datos a un estado alto durante 150 ciclos de reloj.
2. El procedimiento según la reivindicación 1, que comprende además empezar a transmitir un primer paquete de cabecera de subtrama mediante dicho dispositivo central.
3. El procedimiento según la reivindicación 1, que comprende además contar al menos 150 ciclos de reloj continuos de la línea de datos en estado alto, seguidos de al menos 50 ciclos de reloj continuos de la línea de datos en estado bajo, mediante dicho dispositivo cliente.
- 25 4. El procedimiento según la reivindicación 3, que comprende además buscar una palabra única de la primera subtrama mediante dicho dispositivo cliente.
5. El procedimiento según la reivindicación 3, que comprende además detener la excitación de la línea de datos en estado alto mediante dicho dispositivo cliente después de que el dispositivo cliente haya contado 70 ciclos de reloj continuos de los datos en estado alto.
- 30 6. El procedimiento según la reivindicación 5, que comprende además contar otros 80 ciclos de reloj continuos de la línea de datos en estado alto para llegar a los 150 ciclos de reloj de la línea de datos en estado alto mediante dicho dispositivo cliente, y buscar 50 ciclos de reloj de la línea de datos en estado bajo y buscar la palabra única.
- 35 7. Un aparato para transferir datos digitales a una alta velocidad entre un dispositivo central y un dispositivo cliente a través de una trayectoria de comunicaciones para su presentación a un usuario, que comprende:
- medios para generar una o más de una pluralidad de estructuras de paquete predefinidas y enlazarlas entre sí para formar un protocolo de comunicaciones predefinido;
- 40 medios para comunicar un conjunto preseleccionado de datos digitales de control y de presentación entre dicho dispositivo central y dicho dispositivo cliente a través de dicha trayectoria de comunicaciones utilizando dicho protocolo de comunicaciones;
- medios para acoplar al menos un controlador de enlace de dispositivo central que reside en dicho dispositivo central a dicho dispositivo cliente a través de dicha trayectoria de comunicaciones, estando configurado el controlador de enlace de dispositivo central para generar, transmitir y recibir paquetes que forman dicho protocolo de comunicaciones, y para formar datos digitales de presentación en uno o más de tipos de paquetes de datos;
- 45 medios para transferir datos en forma de paquetes a través de dicha trayectoria de comunicaciones utilizando dichos controladores de enlace;
- y caracterizado por:
- medios para activar un enlace de comunicaciones excitando una línea de datos a un estado alto durante al

menos 150 ciclos de reloj y para empezar a transmitir en un tiempo límite de 10 ciclos de reloj una señal estroboscópica como si la línea de datos estuviera a cero, mediante dicho dispositivo central; y

medios para excitar la línea de datos a un estado bajo durante 50 ciclos de reloj mediante dicho dispositivo central mientras se continúa con la transmisión de una señal estroboscópica después de que el dispositivo central haya excitado la línea de datos a un estado alto durante 150 ciclos de reloj.

5

8. El aparato según la reivindicación 7, que comprende además medios para empezar a transmitir un primer paquete de cabecera de subtrama mediante dicho dispositivo central.

10

9. El aparato según la reivindicación 7, que comprende además medios para contar al menos 150 ciclos de reloj continuos de la línea de datos en estado alto, seguidos de al menos 50 ciclos de reloj continuos de la línea de datos en estado bajo, mediante dicho dispositivo cliente.

10. El aparato según la reivindicación 9, que comprende además medios para buscar una palabra única de la primera subtrama mediante dicho dispositivo cliente.

15

11. El aparato según la reivindicación 9, que comprende además medios para detener la excitación de la línea de datos en estado alto mediante dicho dispositivo cliente después de que el dispositivo cliente haya contado 70 ciclos de reloj continuos de los datos en estado alto.

12. El aparato según la reivindicación 11, que comprende además medios para contar otros 80 ciclos de reloj continuos de la línea de datos en estado alto para llegar a los 150 ciclos de reloj de la línea de datos en estado alto mediante dicho dispositivo cliente, y buscar 50 ciclos de reloj de la línea de datos en estado bajo y buscar la palabra única.

20

13. Medio legible por ordenador que comprende instrucciones para llevar a cabo un procedimiento según cualquiera de las reivindicaciones 1 a 6.

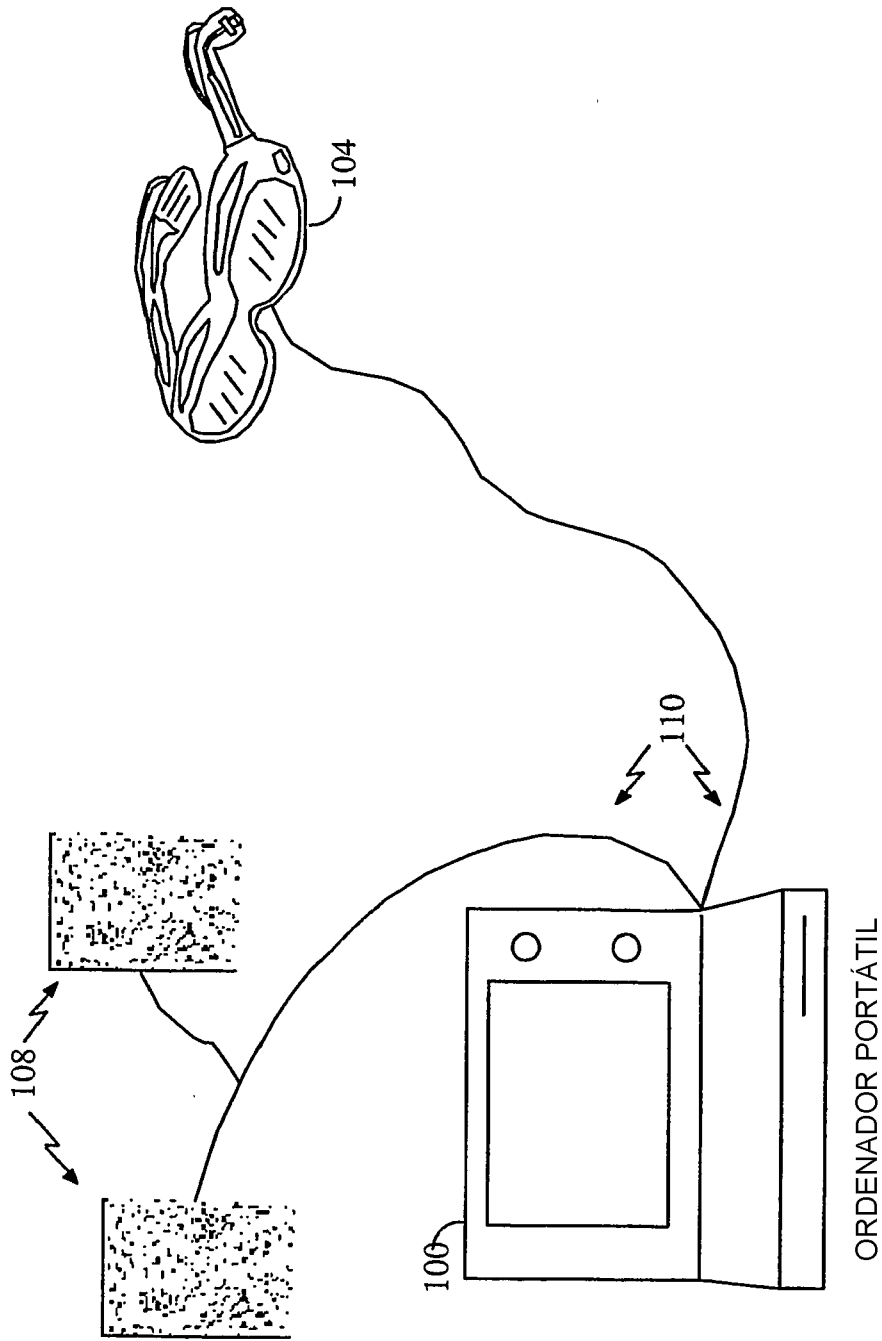


FIG. 1A

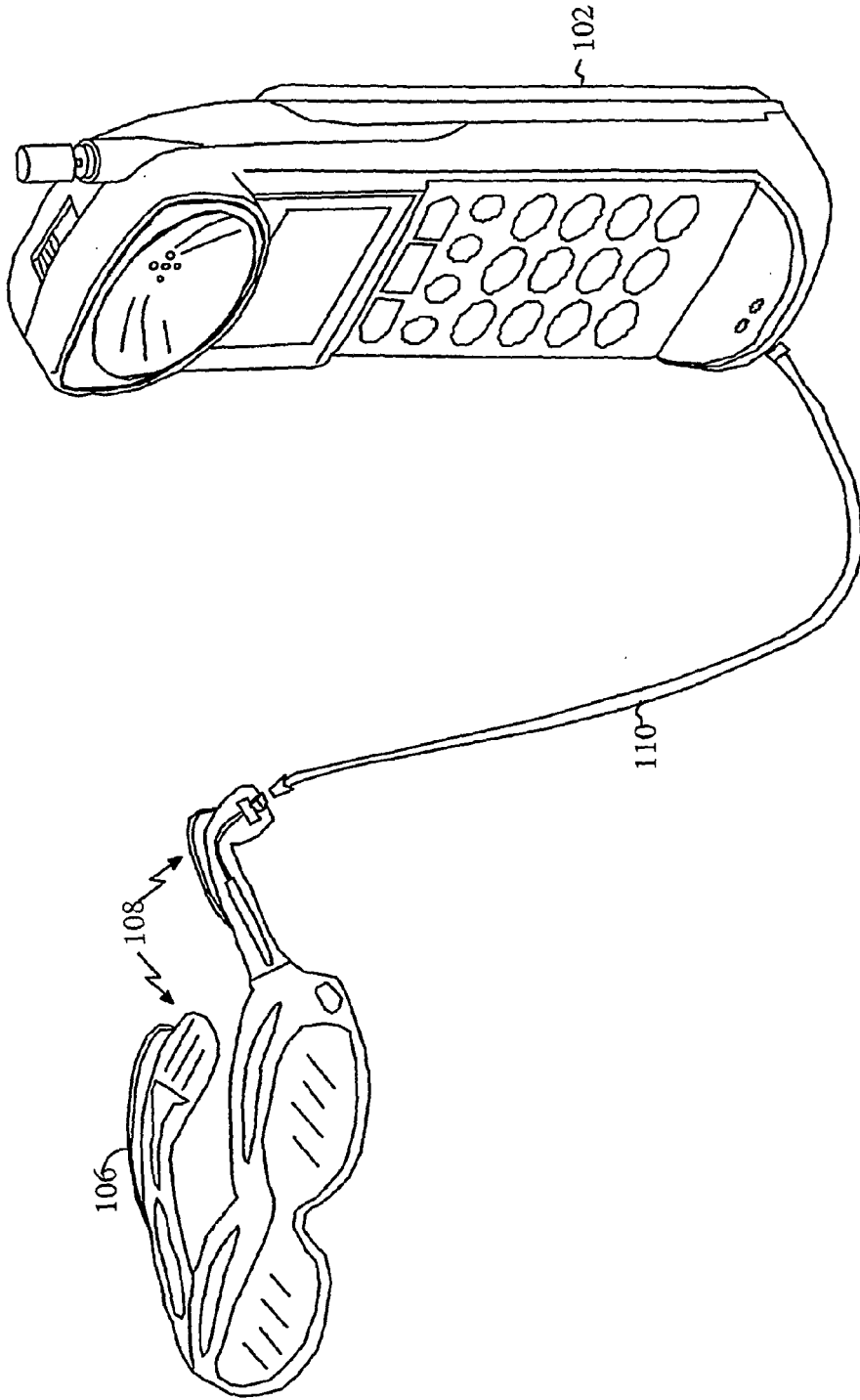


FIG. 1B

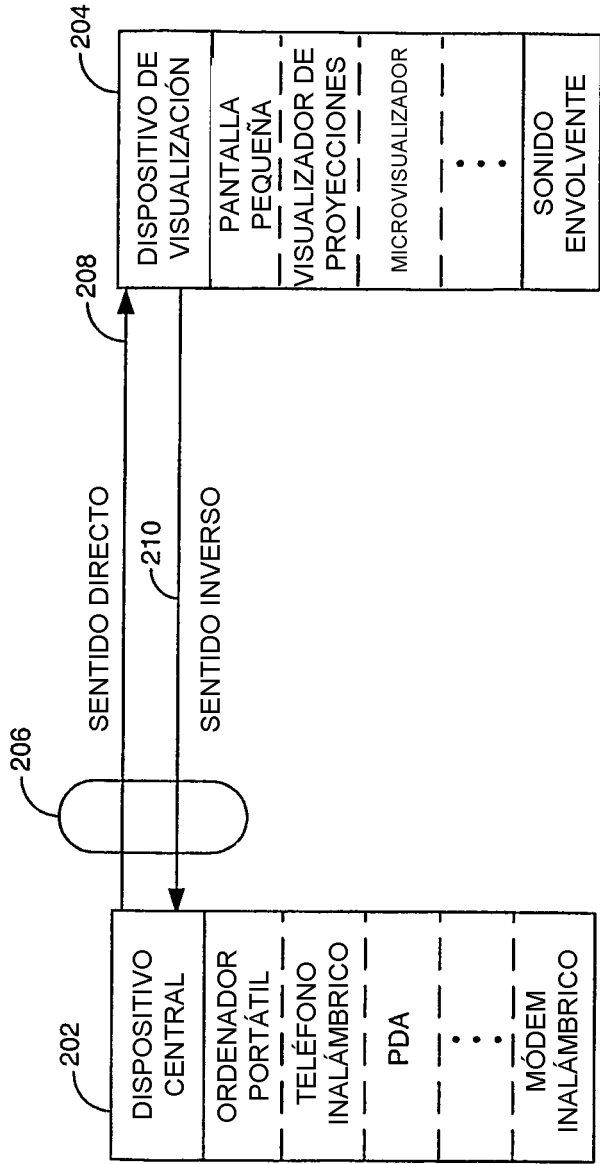


FIG. 2

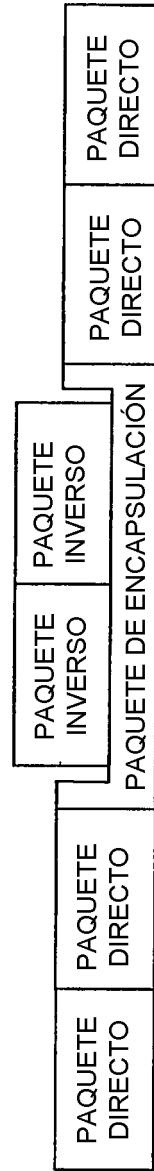


FIG. 3

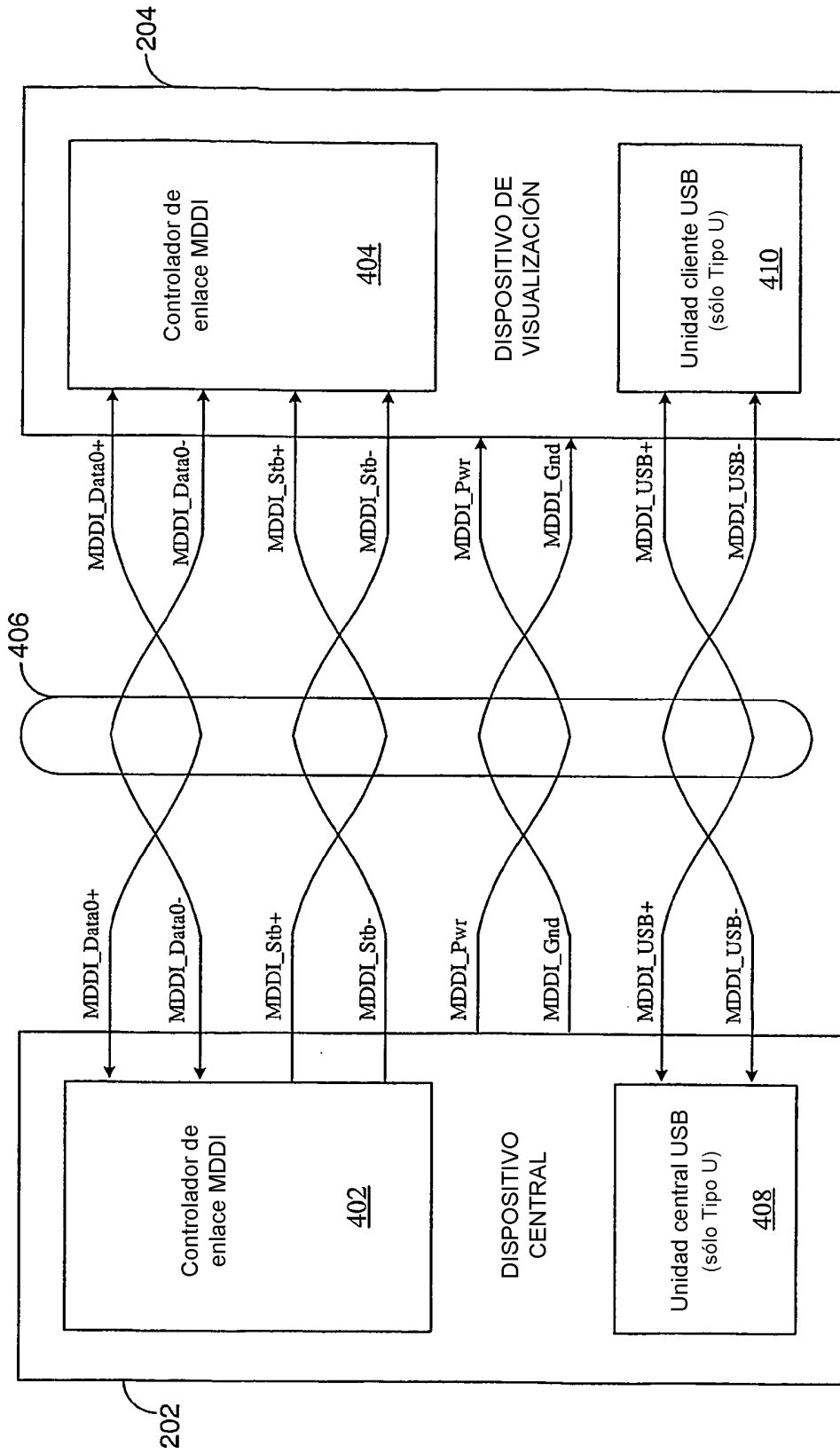


FIG. 4

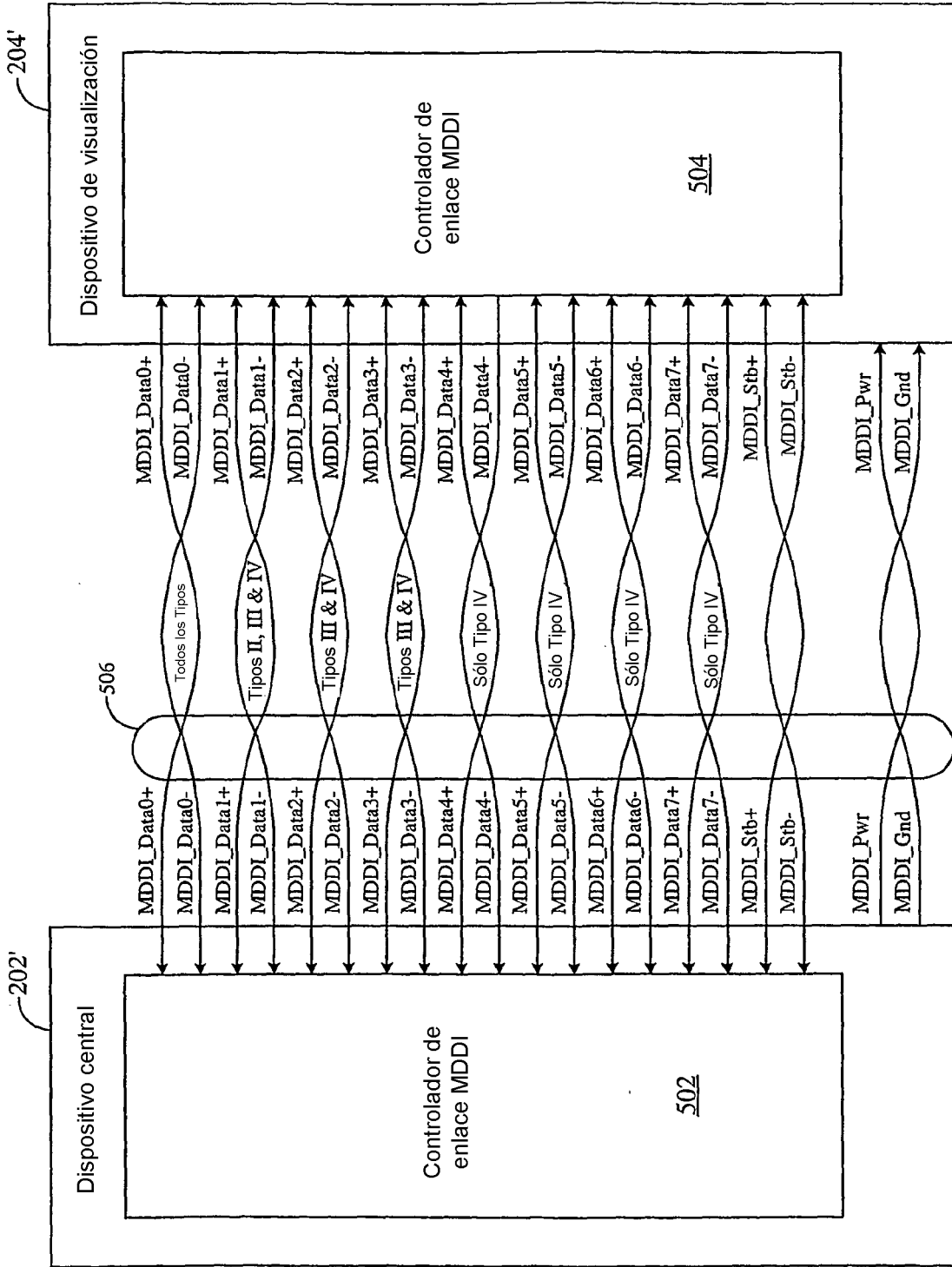


FIG. 5

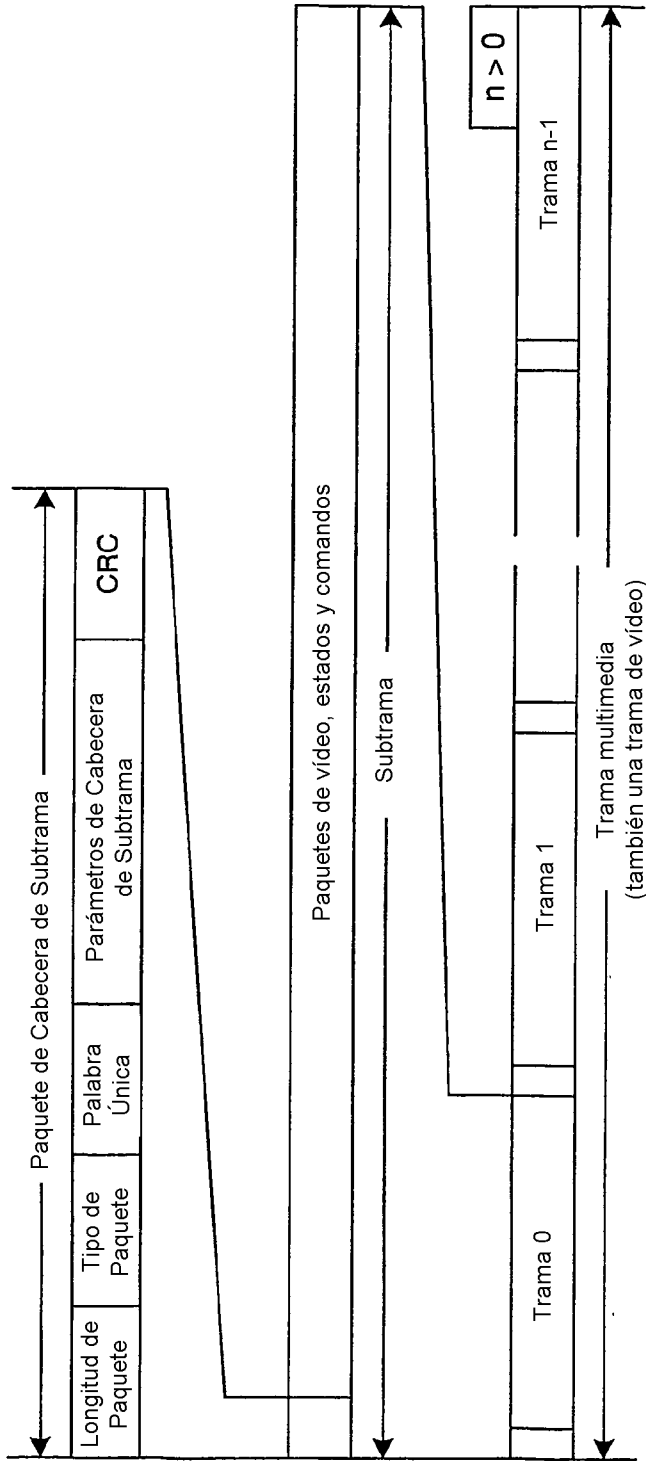


FIG. 6



FIG. 7

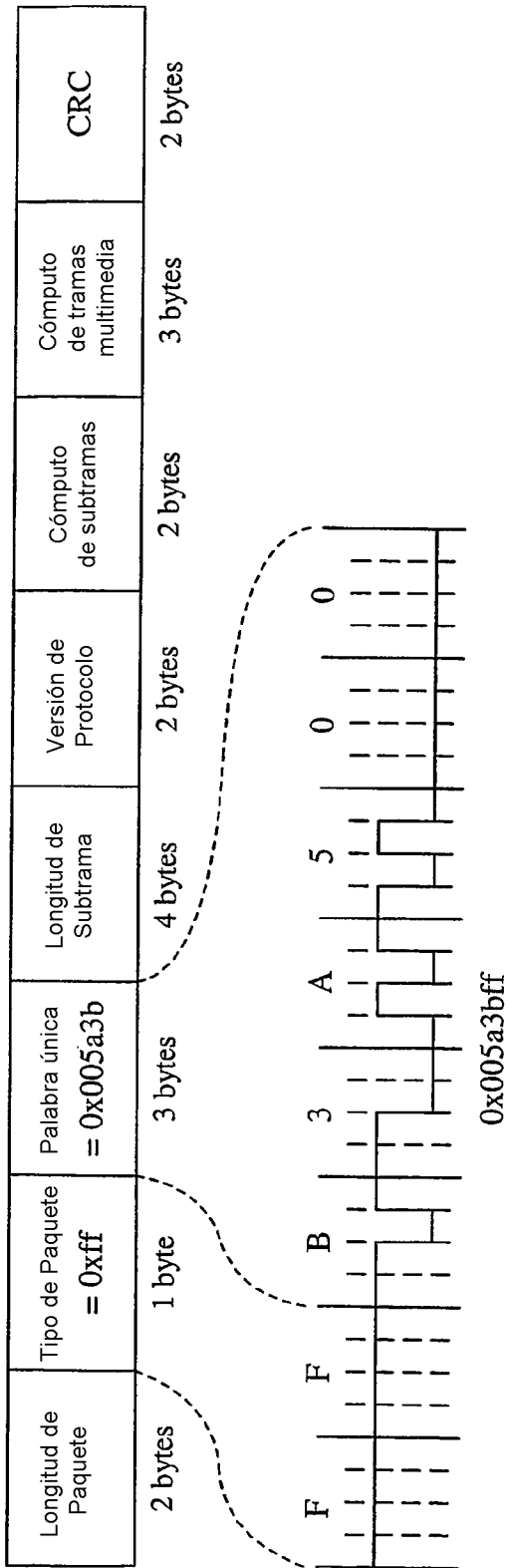


FIG. 8

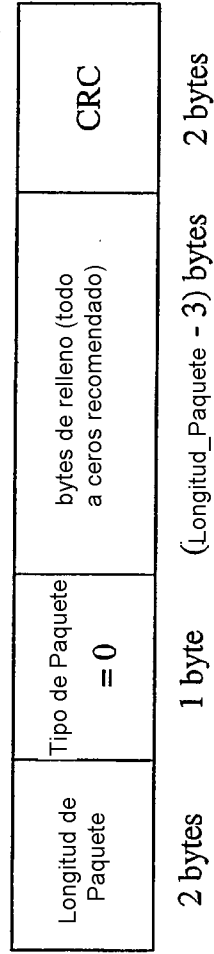


FIG. 9

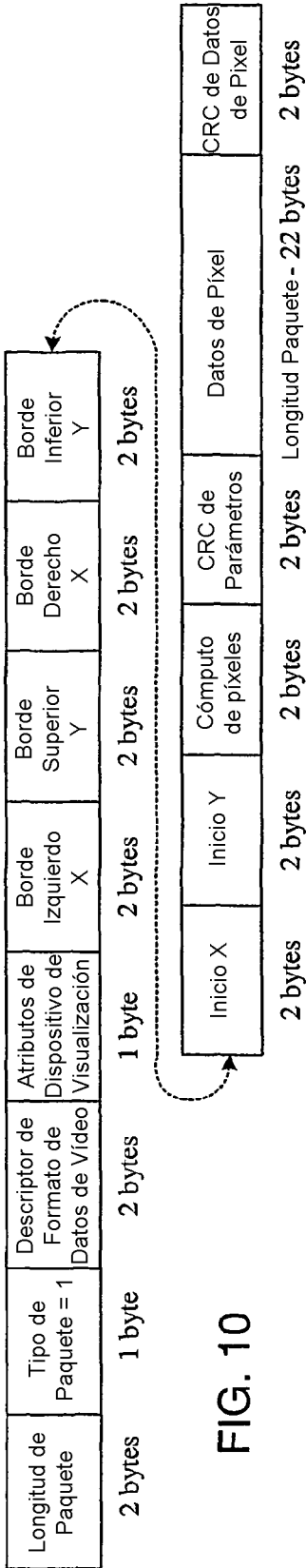


FIG. 10

Descriptor de Formato de Datos de Vídeo - Monocromo

0	0	0	P	no utilizados = 0	número de bits por píxel
15	14	13	12	11	4 3 0

Descriptor de Formato de Datos de Vídeo - Color utilizando Mapa de Colores

0	0	1	P	no utilizados = 0	número de bits por píxel
15	14	13	12	11	6 5 0

Descriptor de Formato de Datos de Vídeo - Color, RGB sin procesar

0	1	0	P	número de bits de rojo por píxel	número de bits de verde por píxel	número de bits de azul por píxel
15	14	13	12	11	8 7	4 3 0

Descriptor de Formato de Datos de Vídeo - Vídeo de Componente a Color, Y Pr Pb

0	1	1	P	número de bits de Y por píxel	número de bits de Pr por píxel	número de bits de Pb por píxel
15	14	13	12	11	8 7	4 3 0

FIG. 11

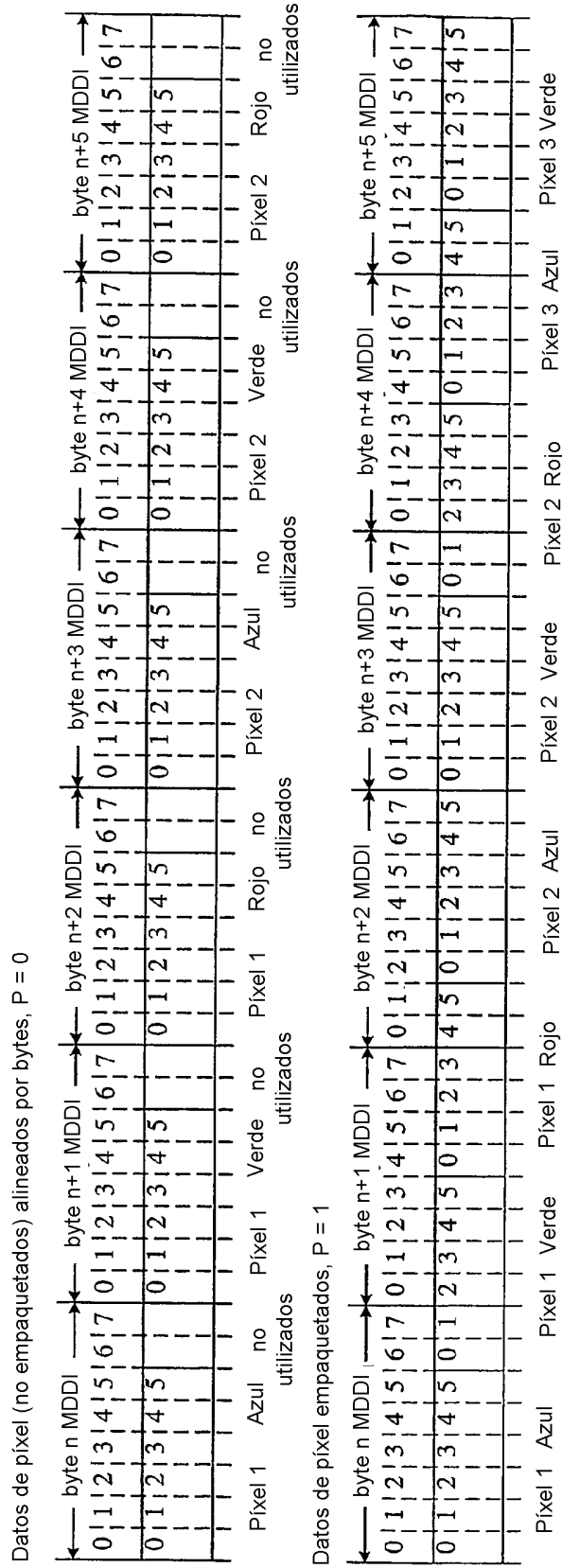


FIG. 12

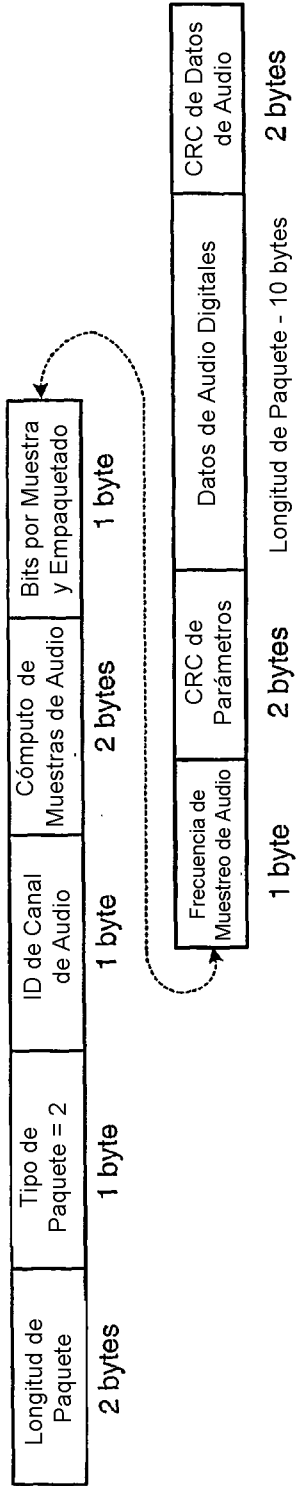


FIG. 13

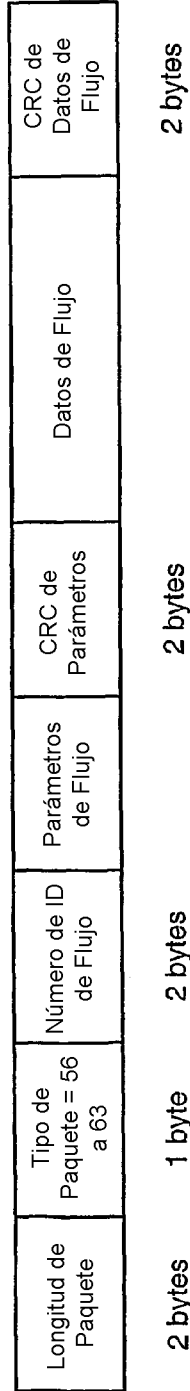


FIG. 15

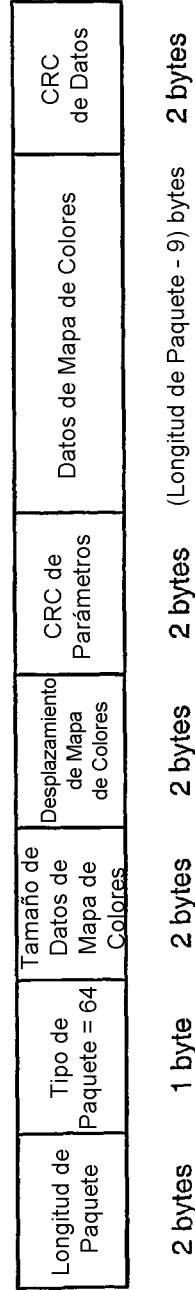


FIG. 16

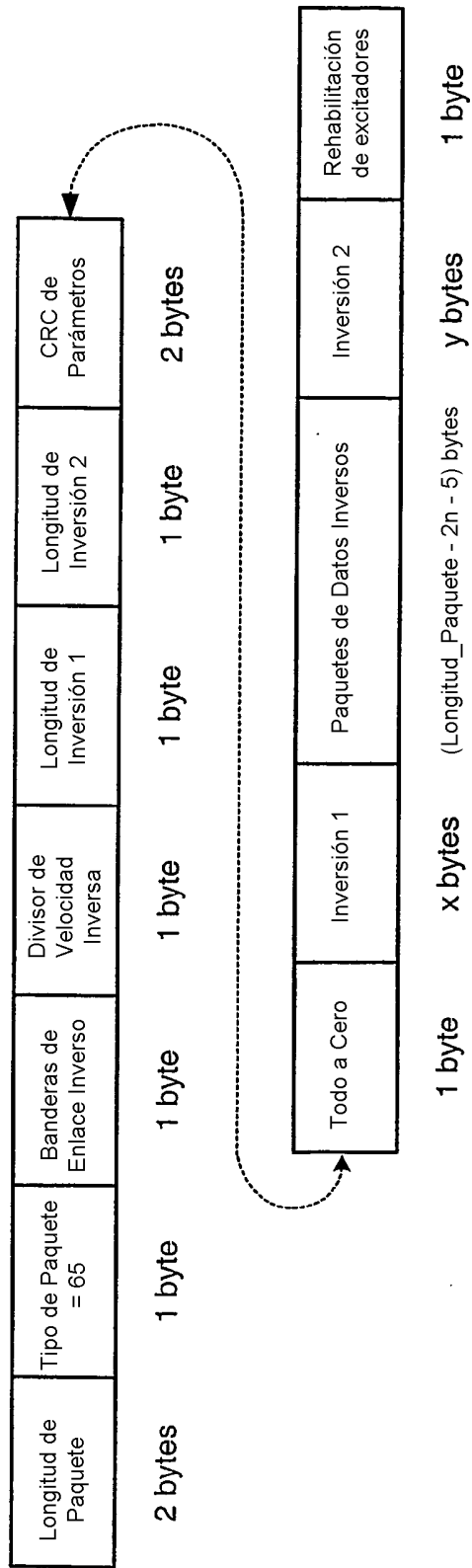


FIG. 17

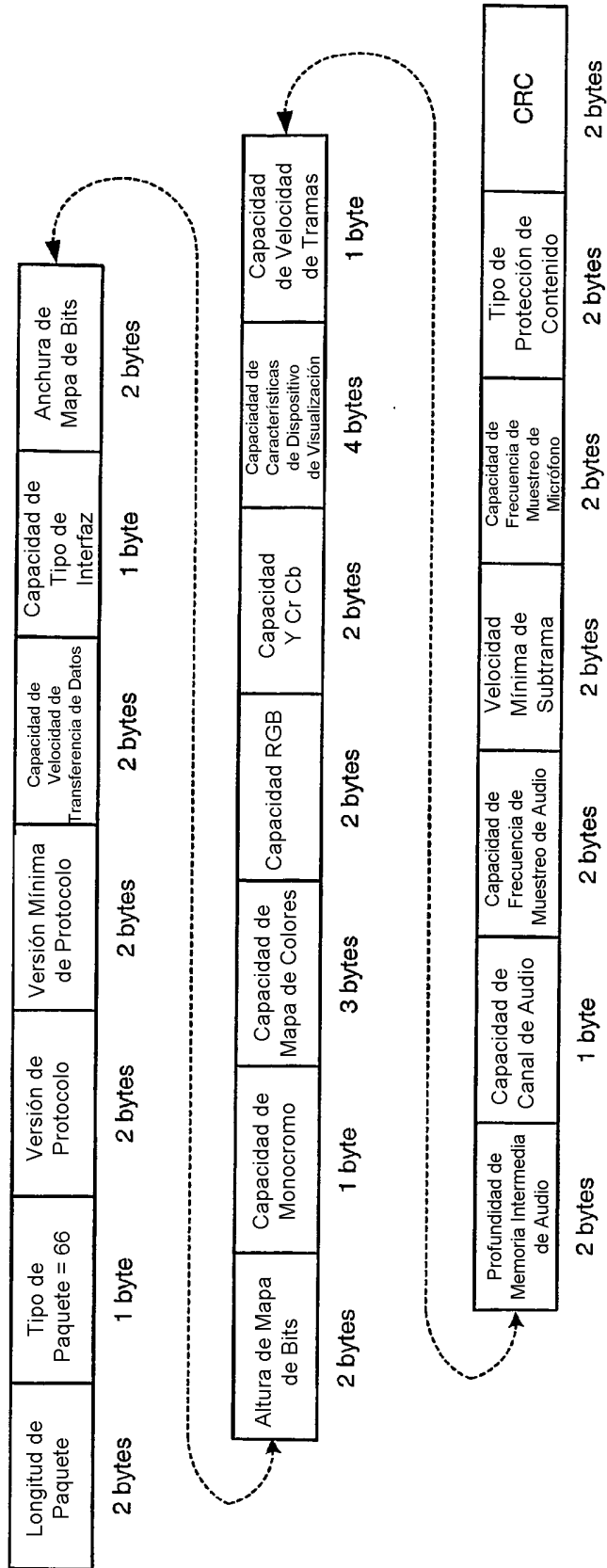


FIG. 18

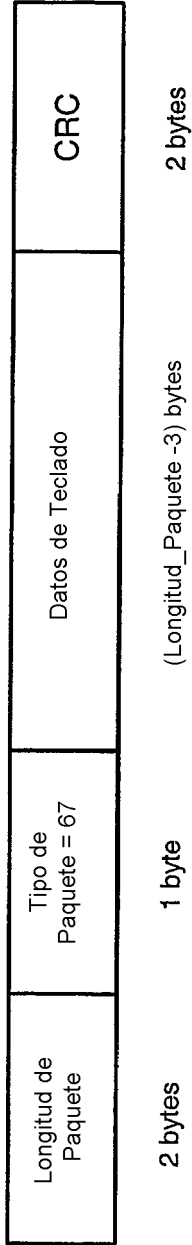


FIG. 19

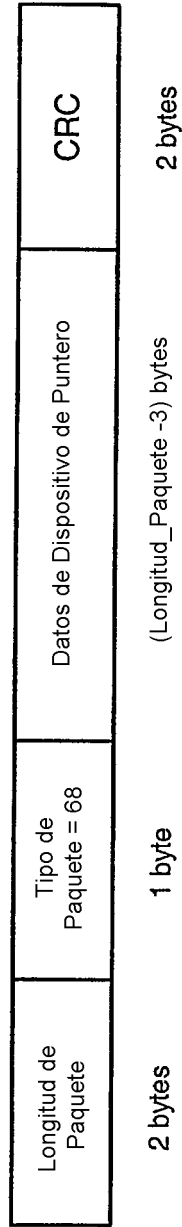


FIG. 20

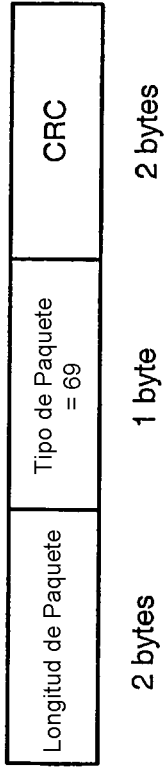


FIG. 21

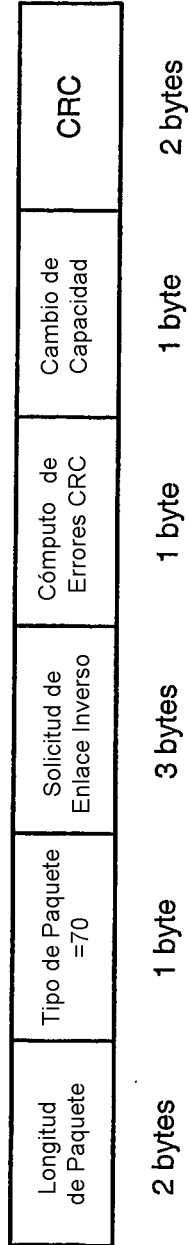


FIG. 22

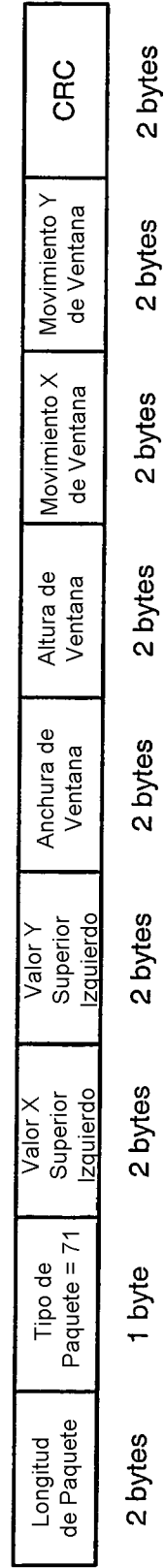


FIG. 23

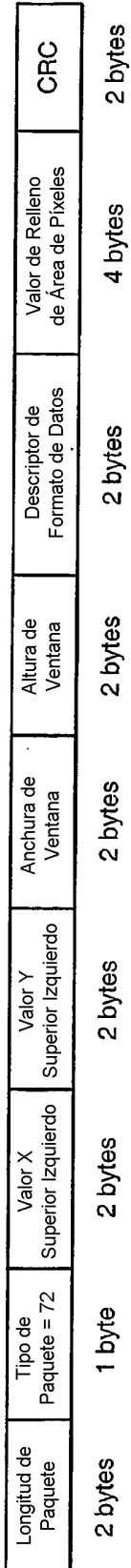


FIG. 24

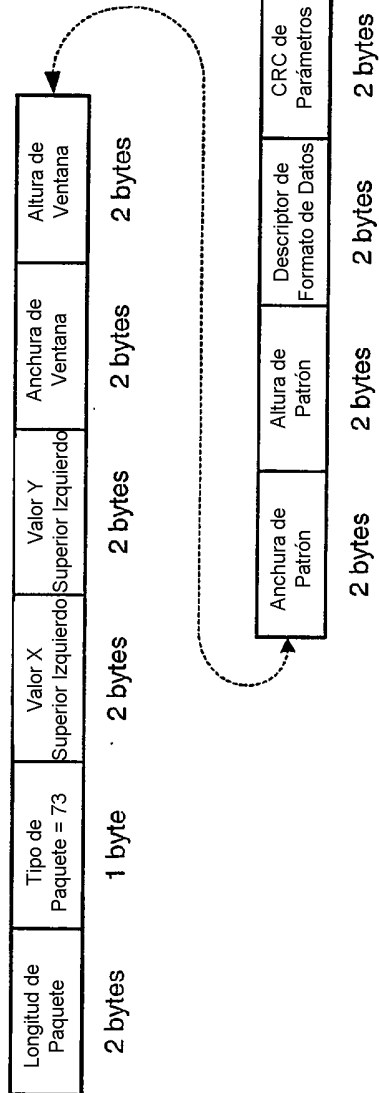


FIG. 25

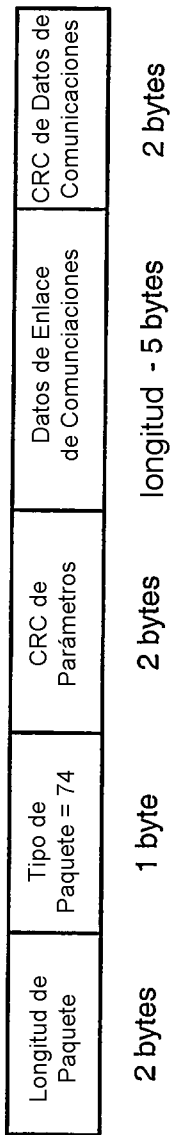


FIG. 26

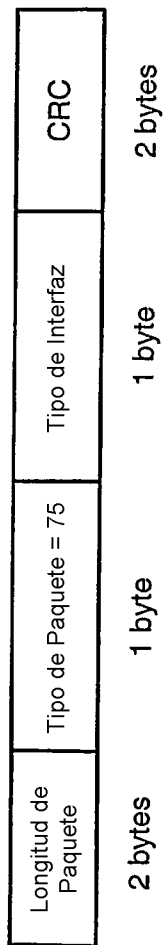


FIG. 27

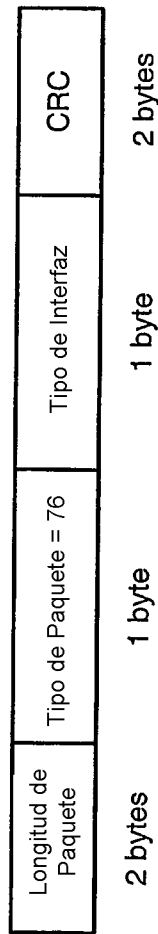


FIG. 28

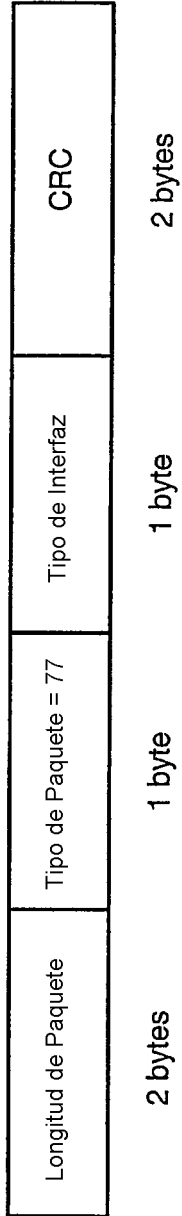


FIG. 29

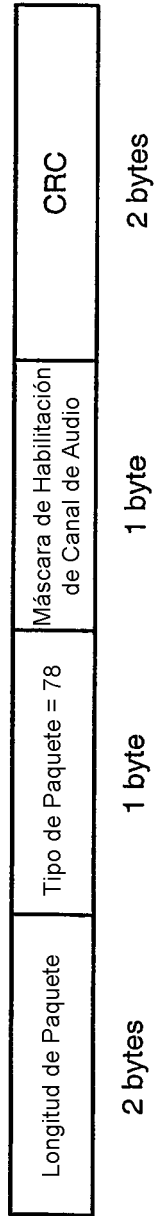


FIG. 30

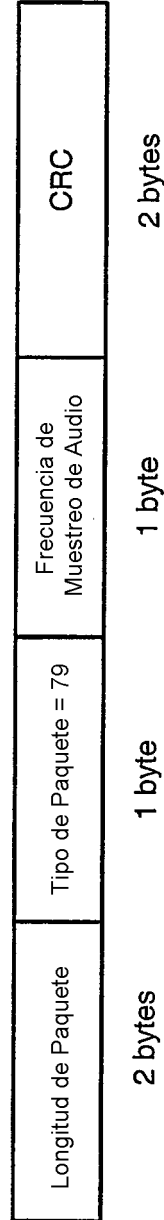


FIG. 31

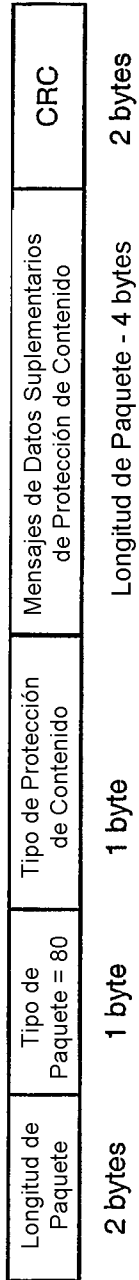


FIG. 32

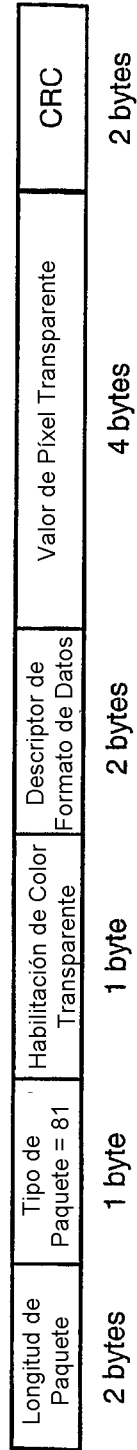


FIG. 33

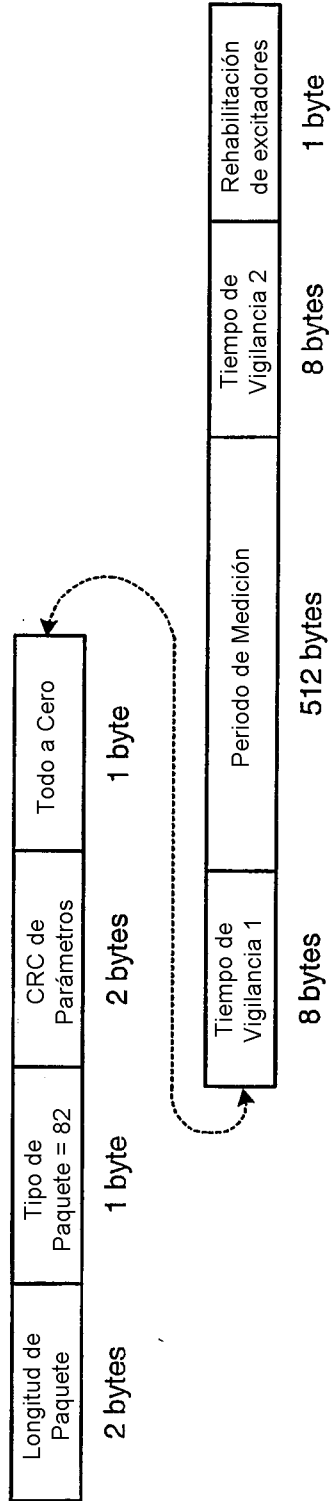


FIG. 34

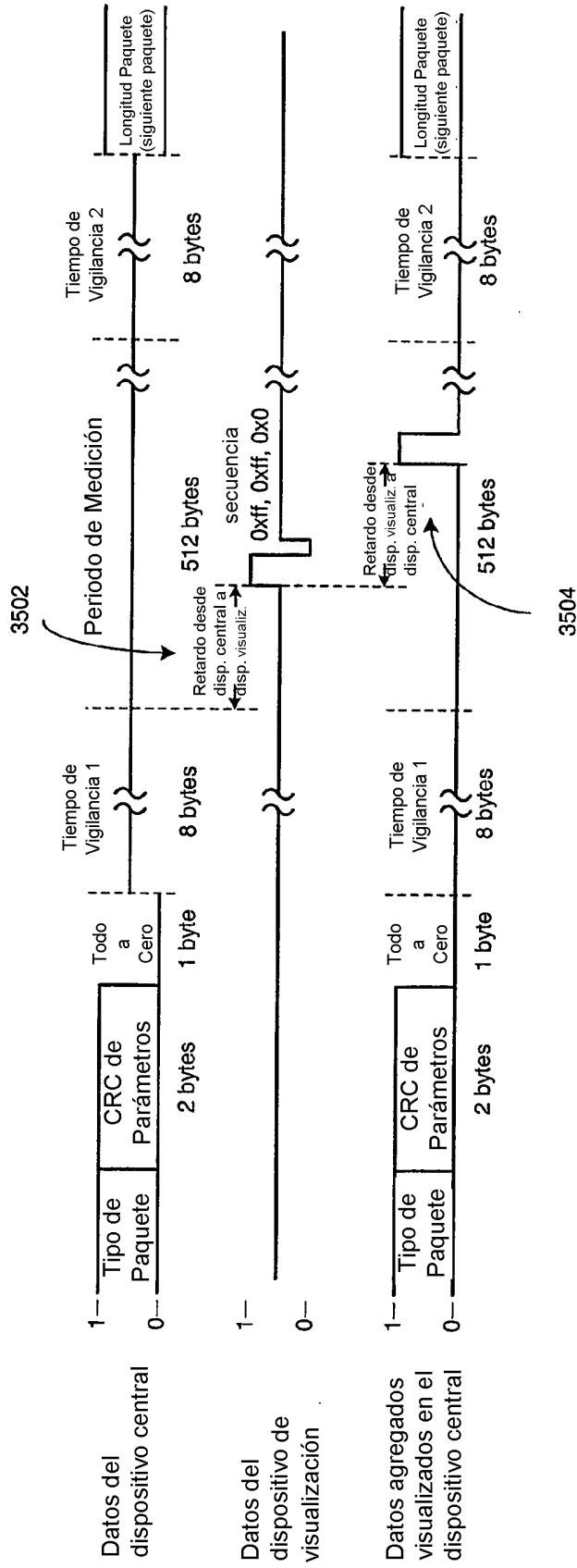


FIG. 35

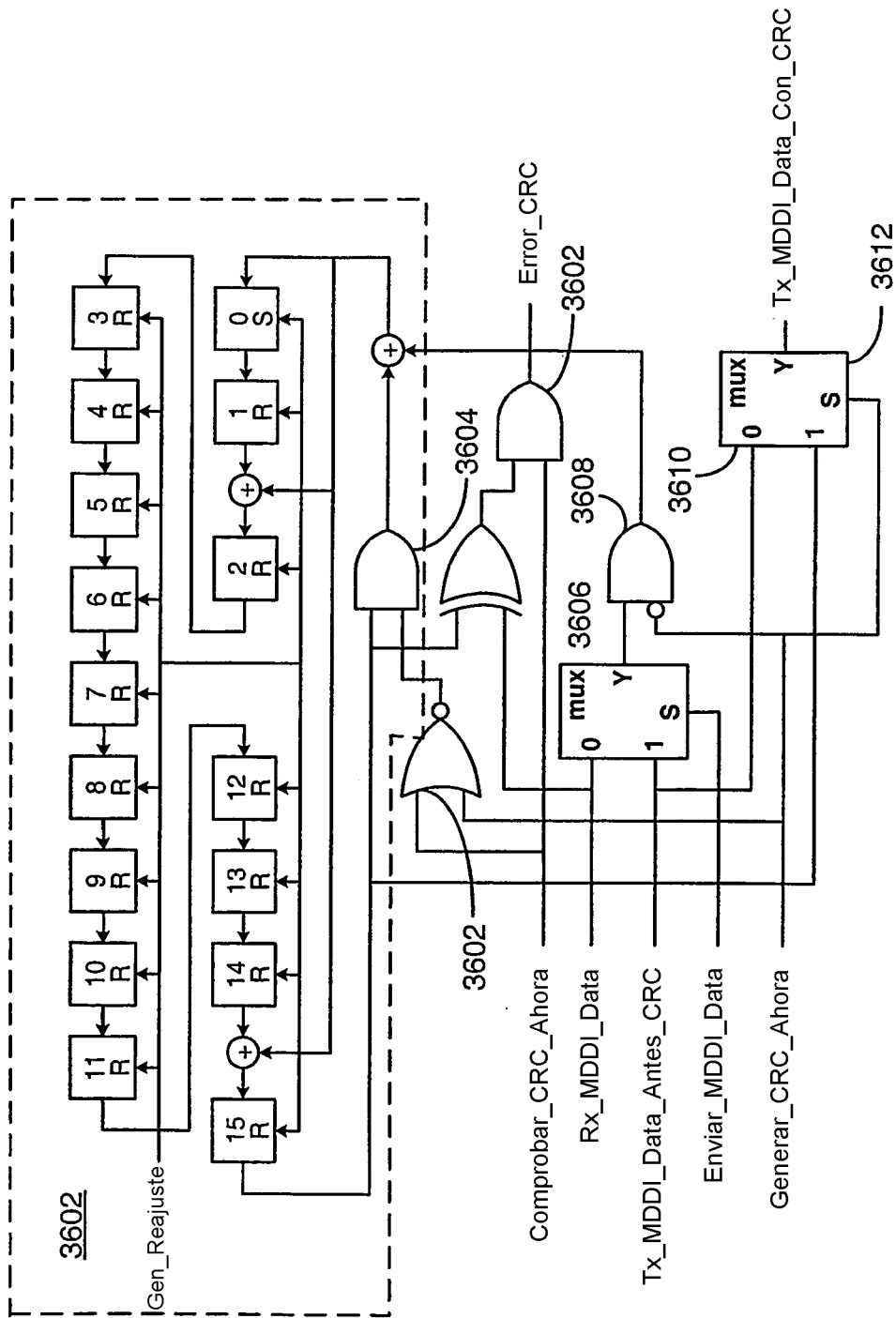


FIG. 36

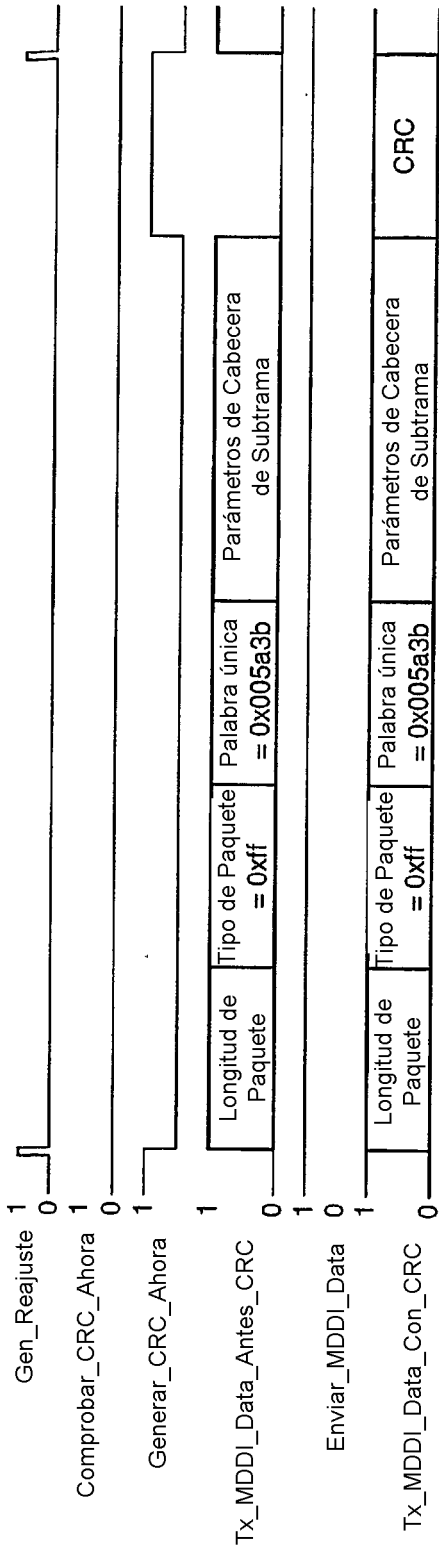


FIG. 37A

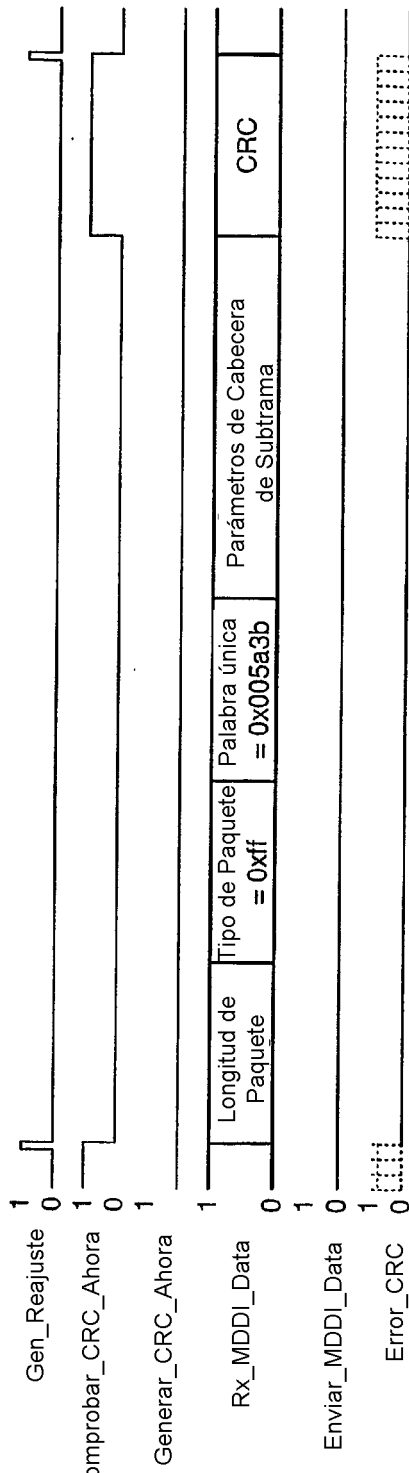


FIG. 37B

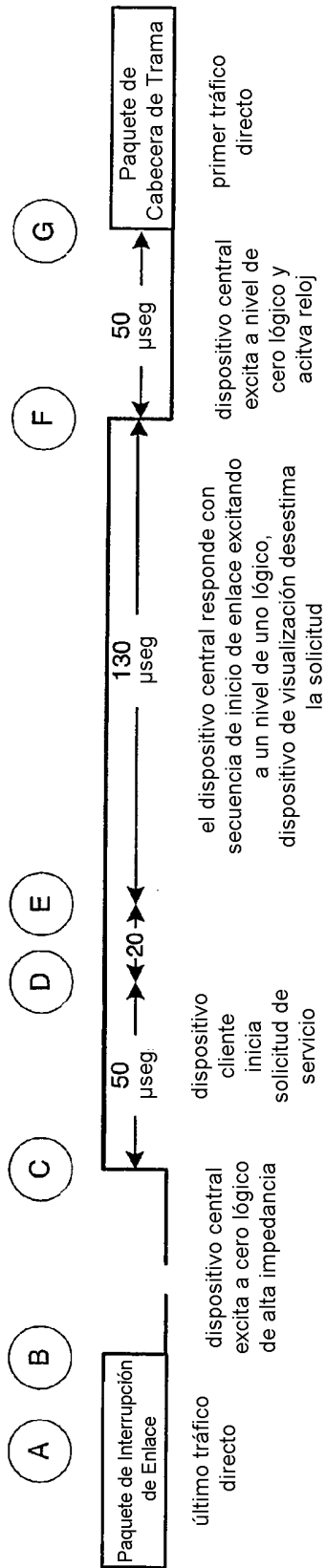


FIG. 38

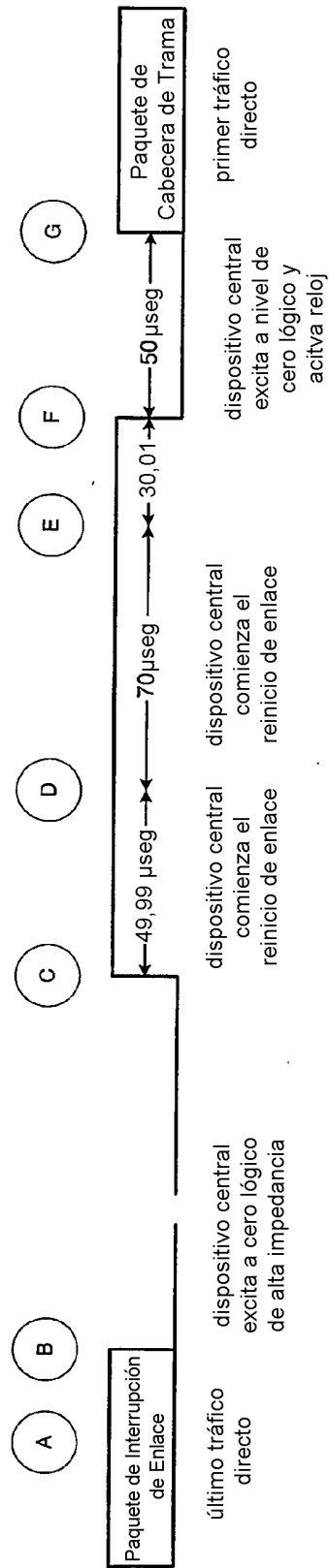


FIG. 39

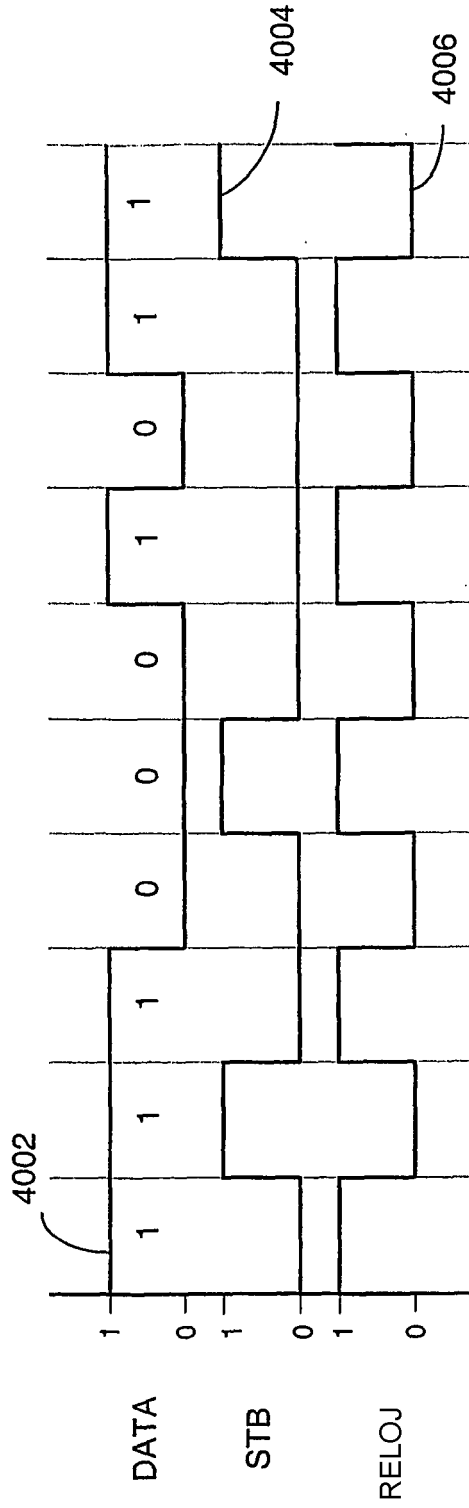


FIG. 40

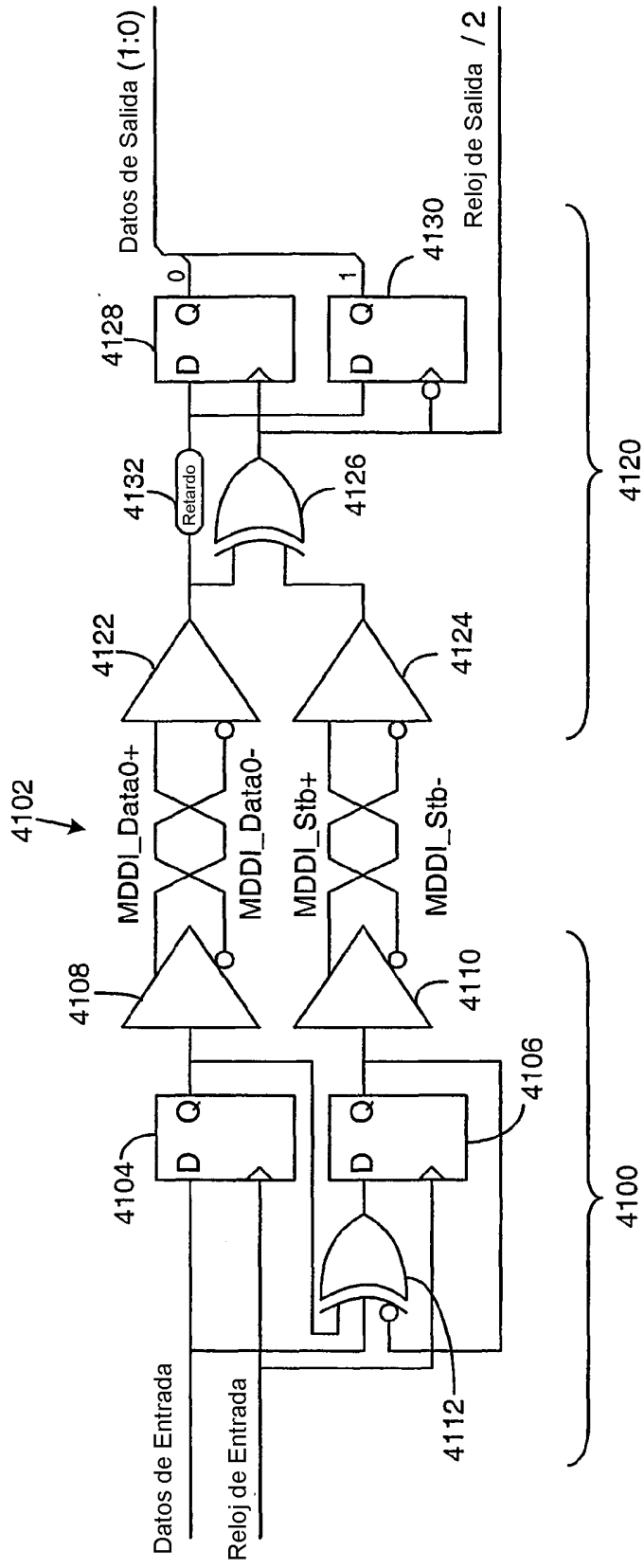


FIG. 41

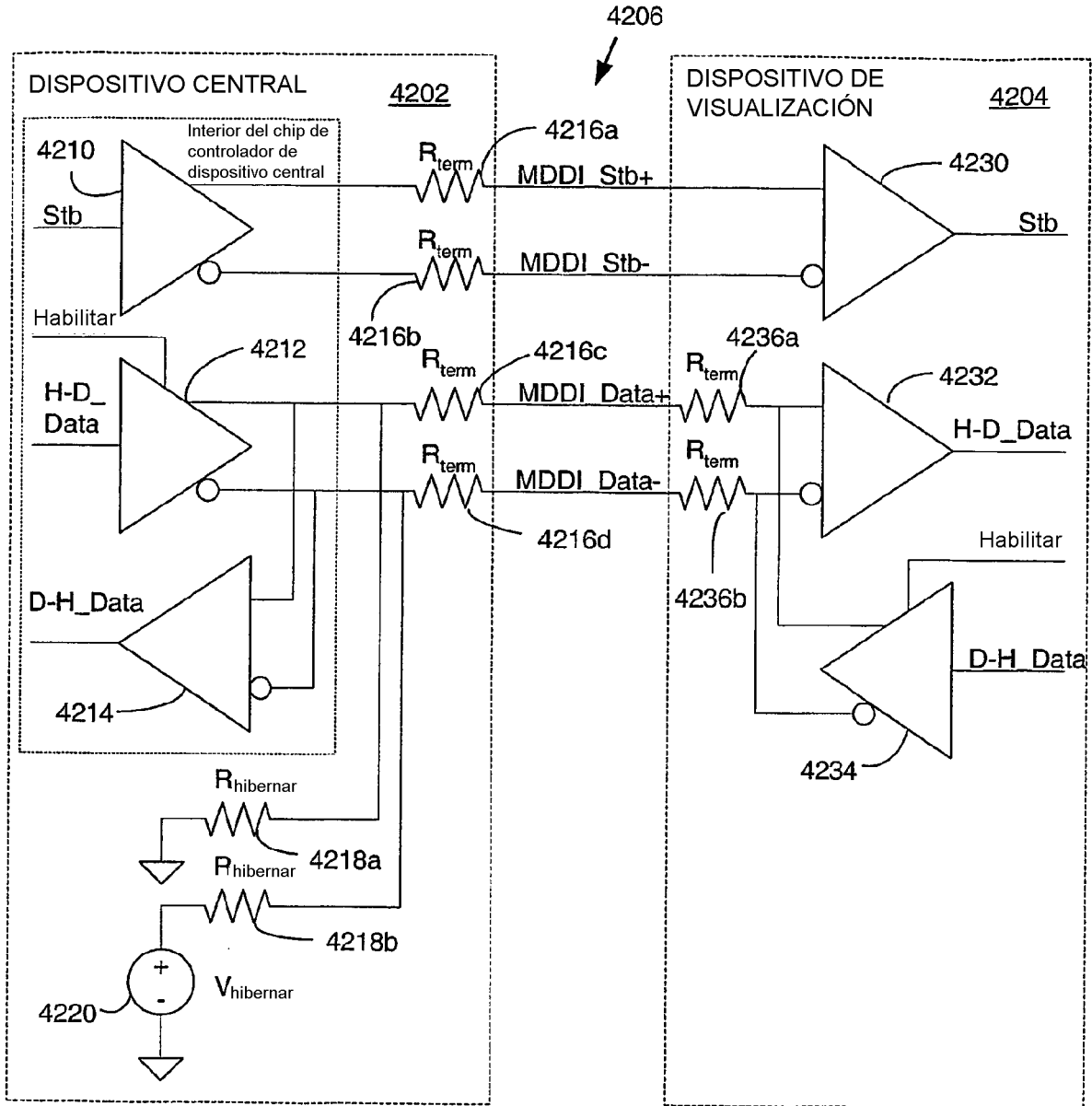


FIG. 42

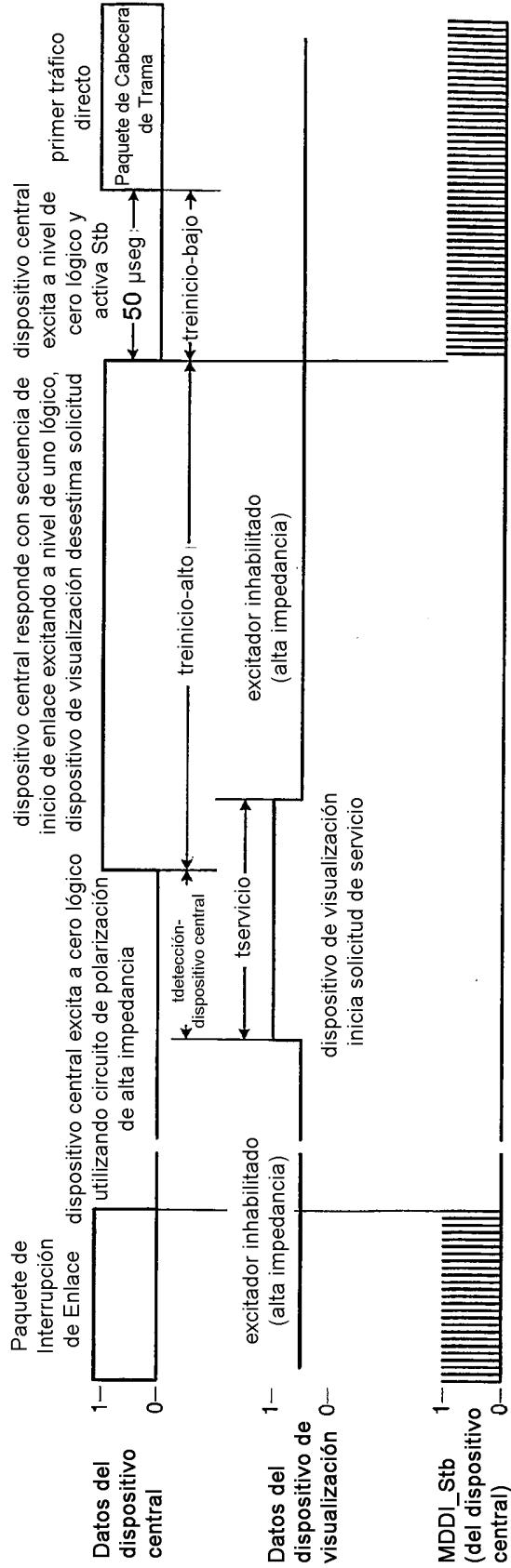


FIG. 43

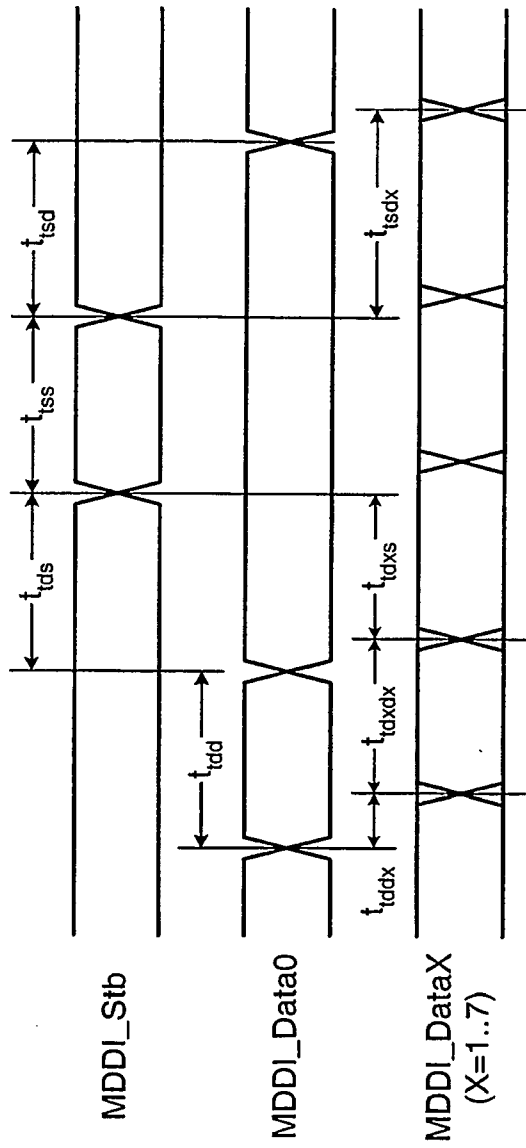


FIG. 44

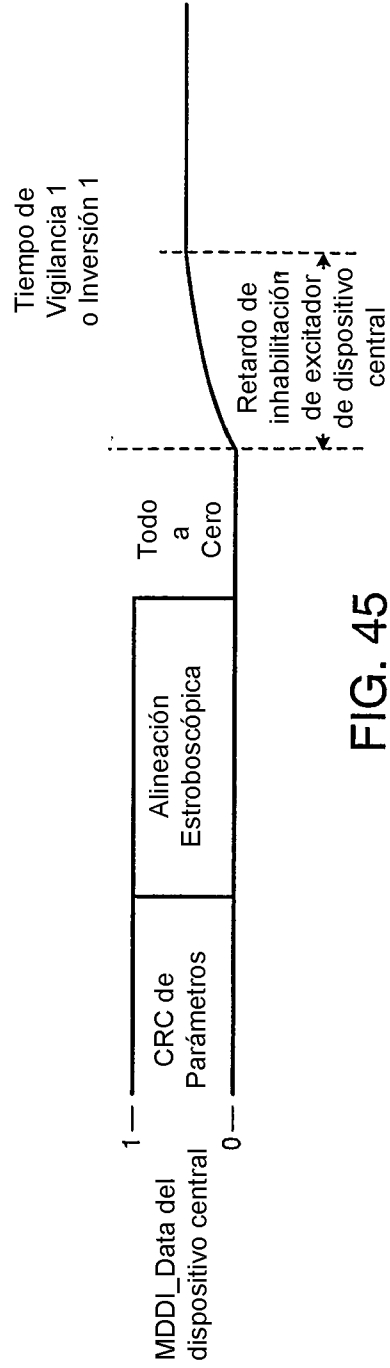


FIG. 45

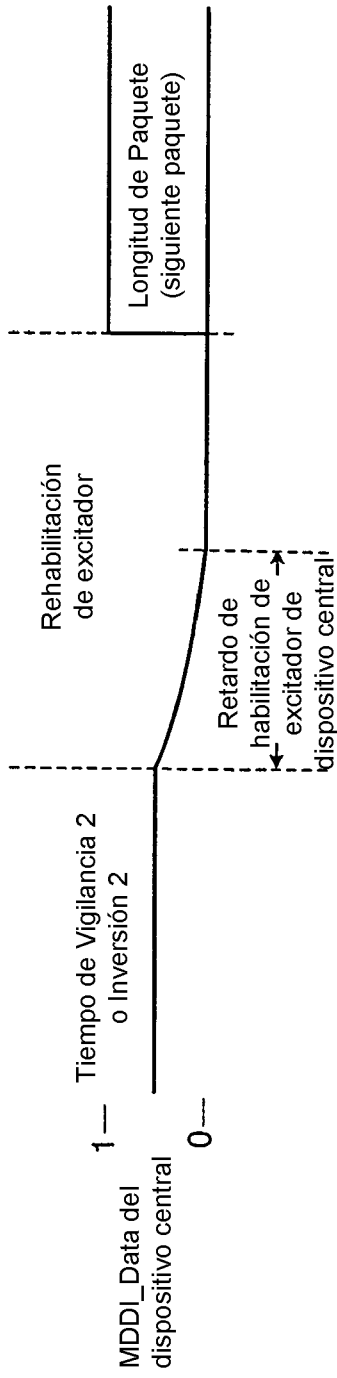


FIG. 46

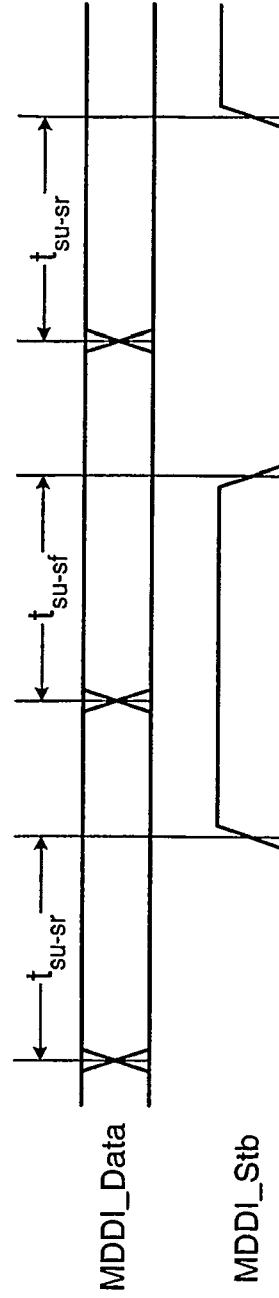


FIG. 47

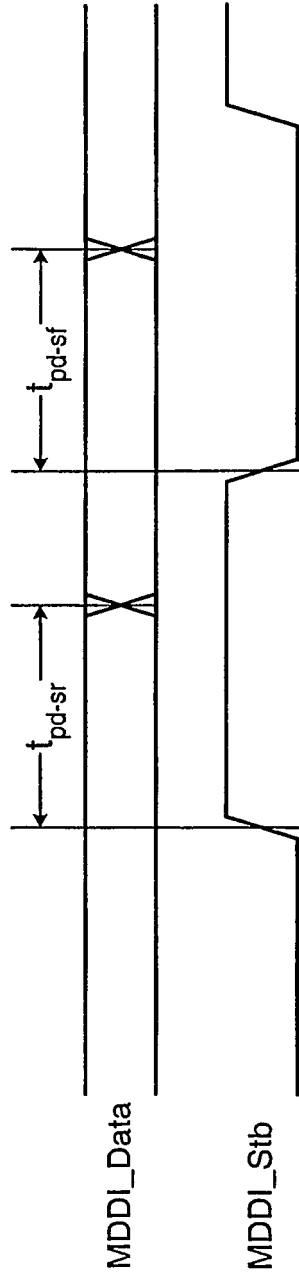


FIG. 48

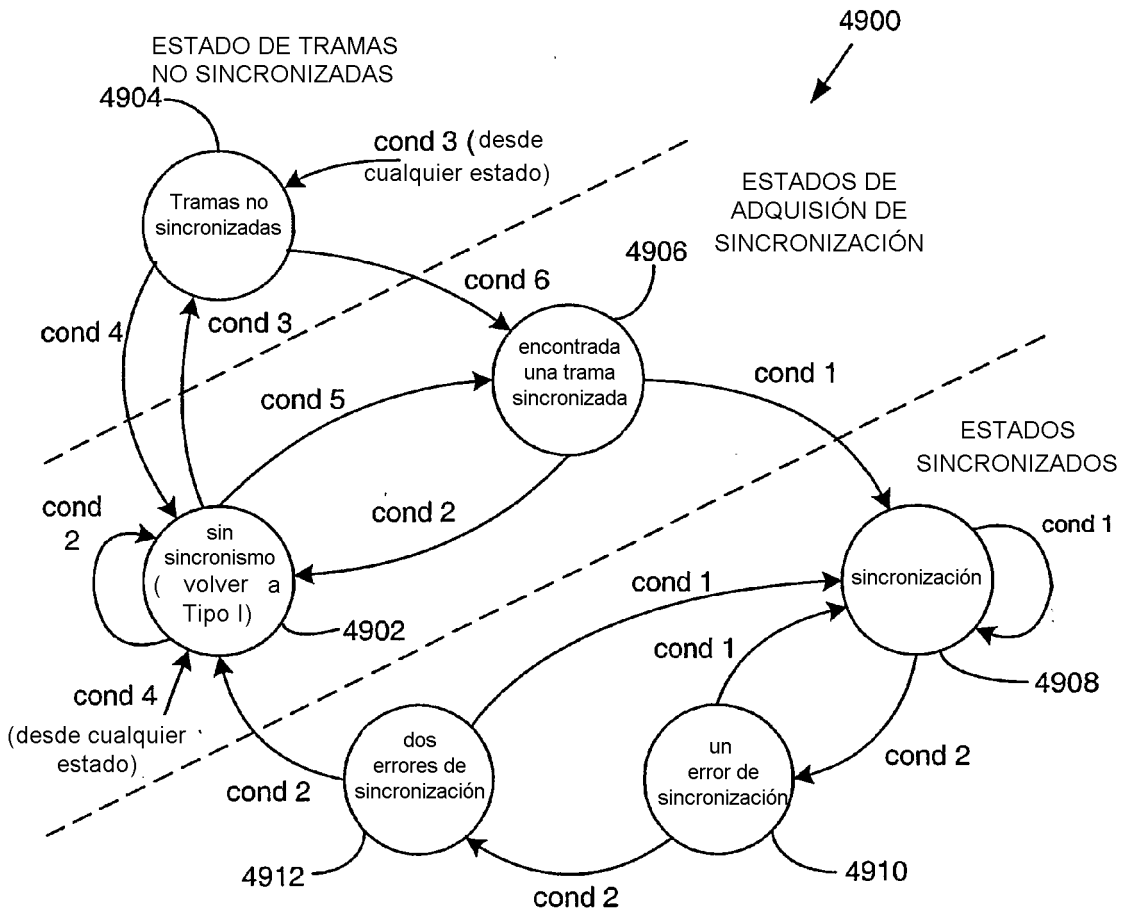


FIG. 49

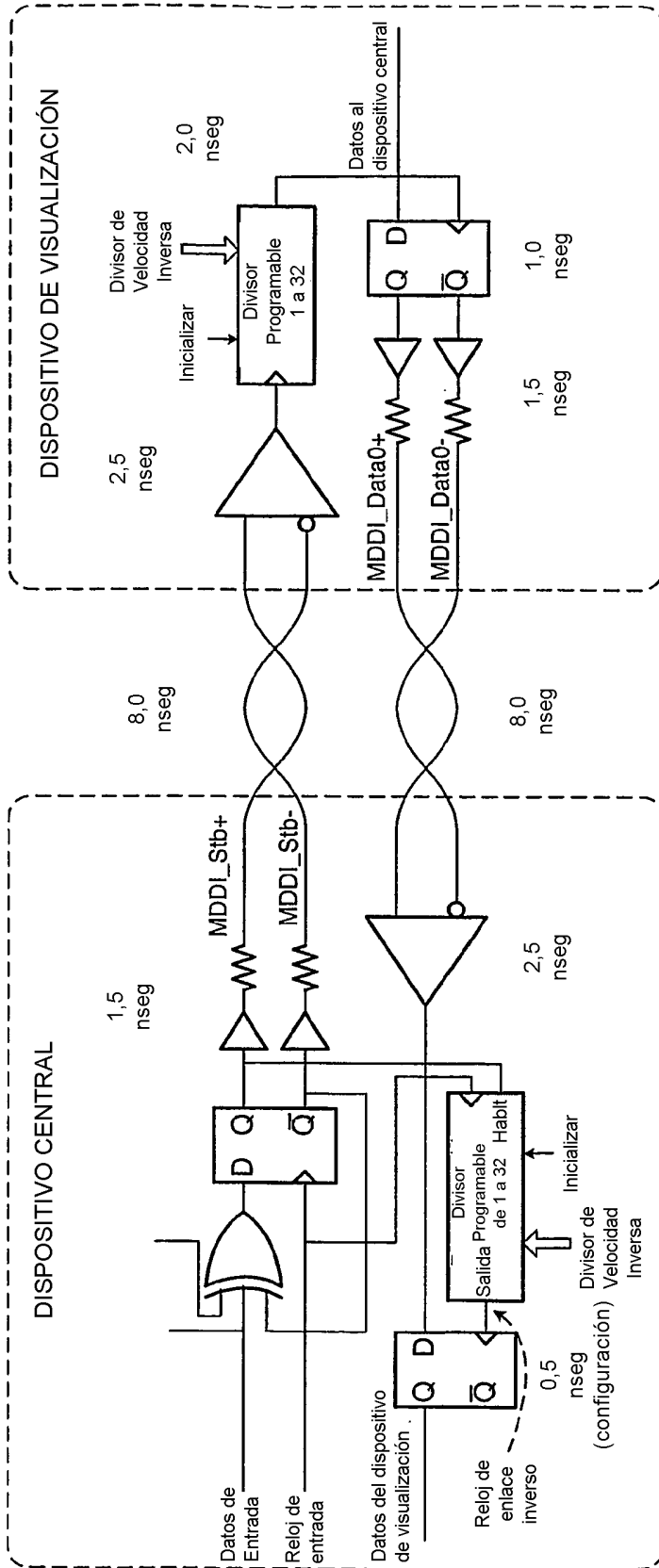


FIG. 50

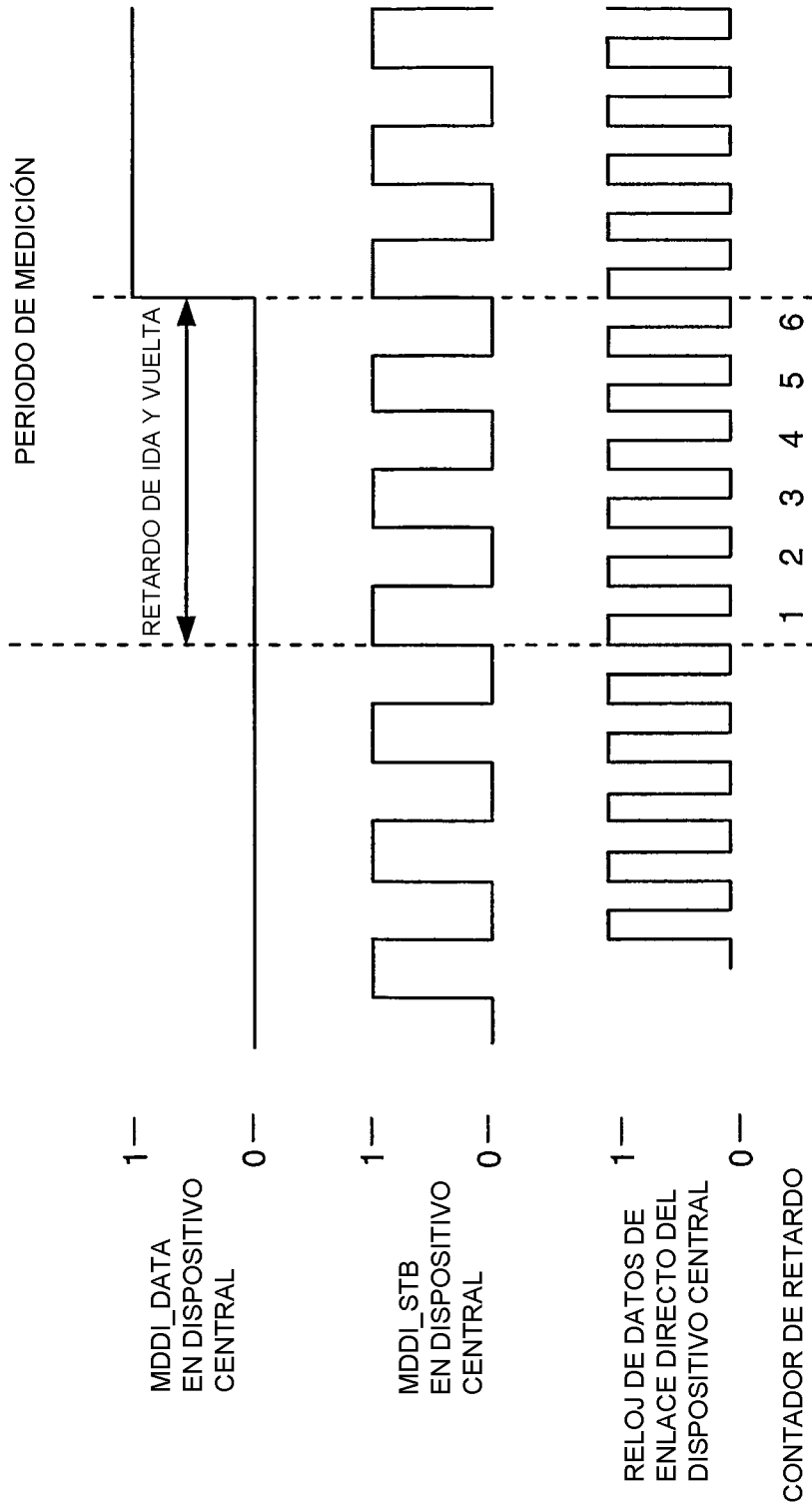


FIG. 51

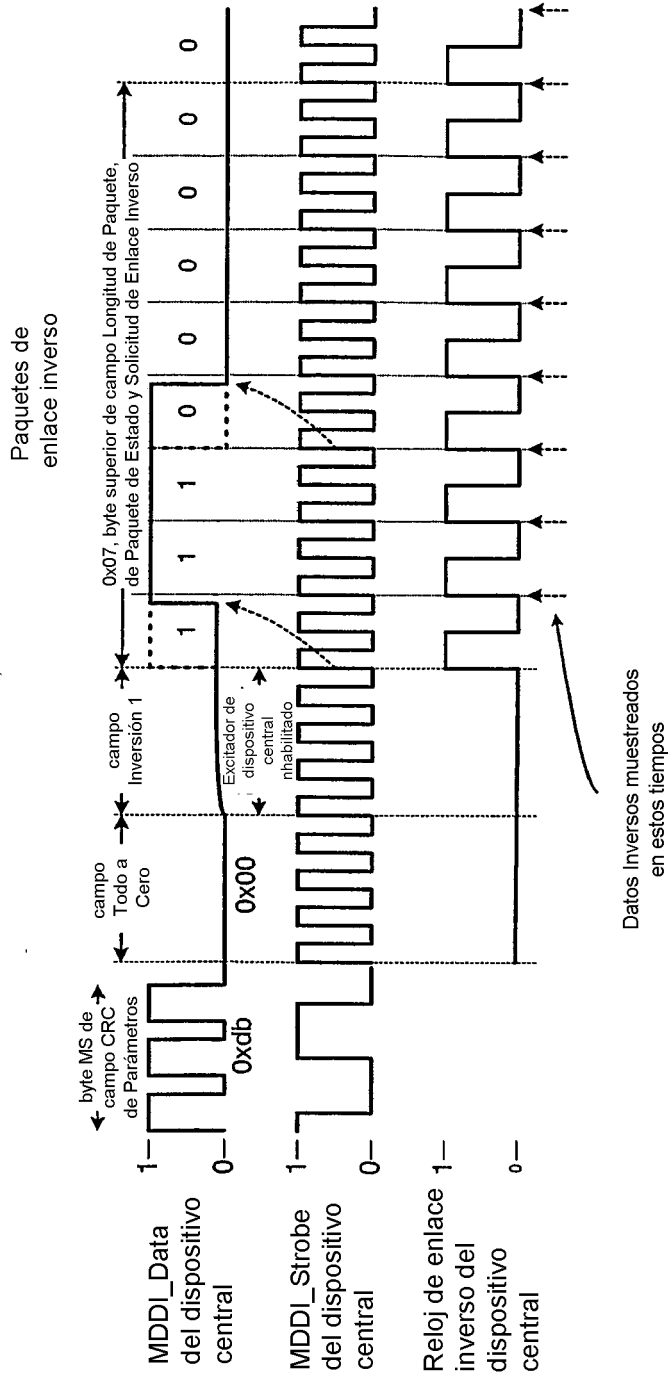


FIG. 52

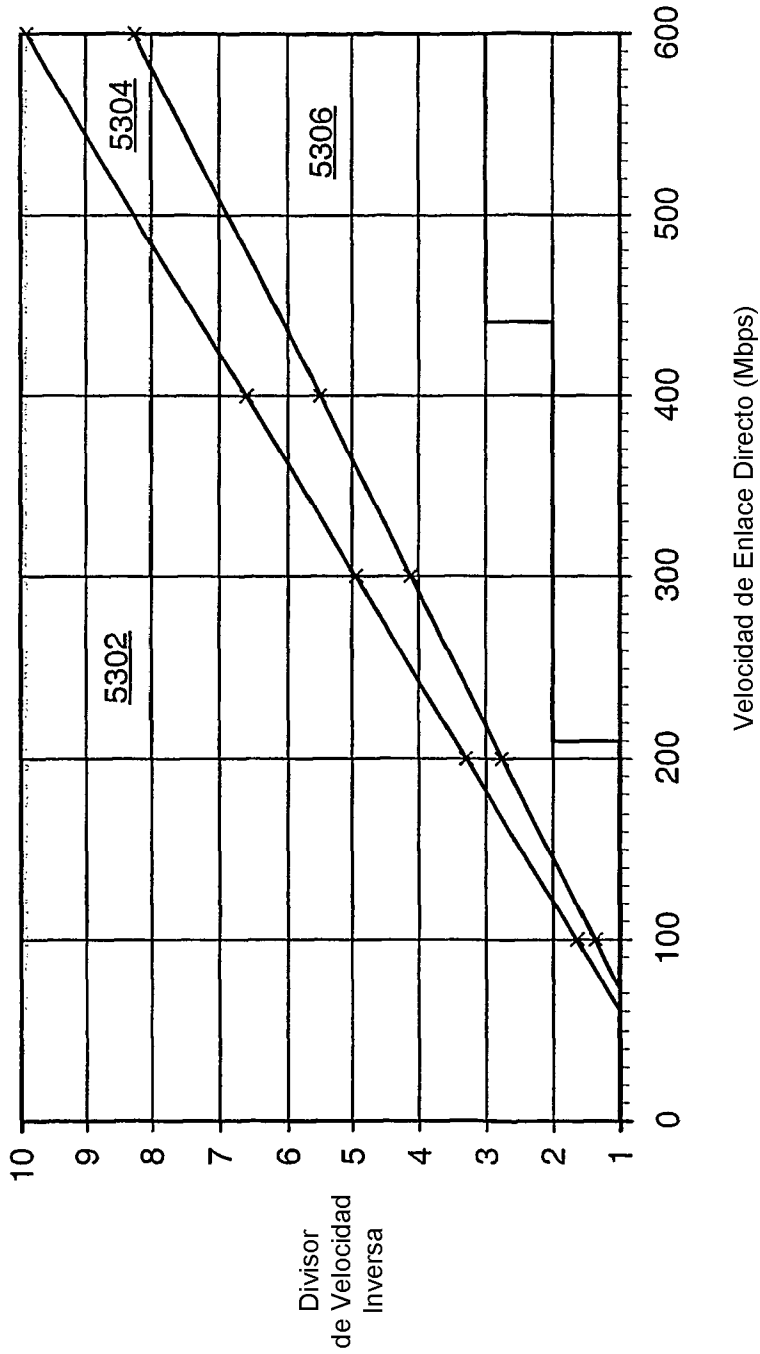


FIG. 53

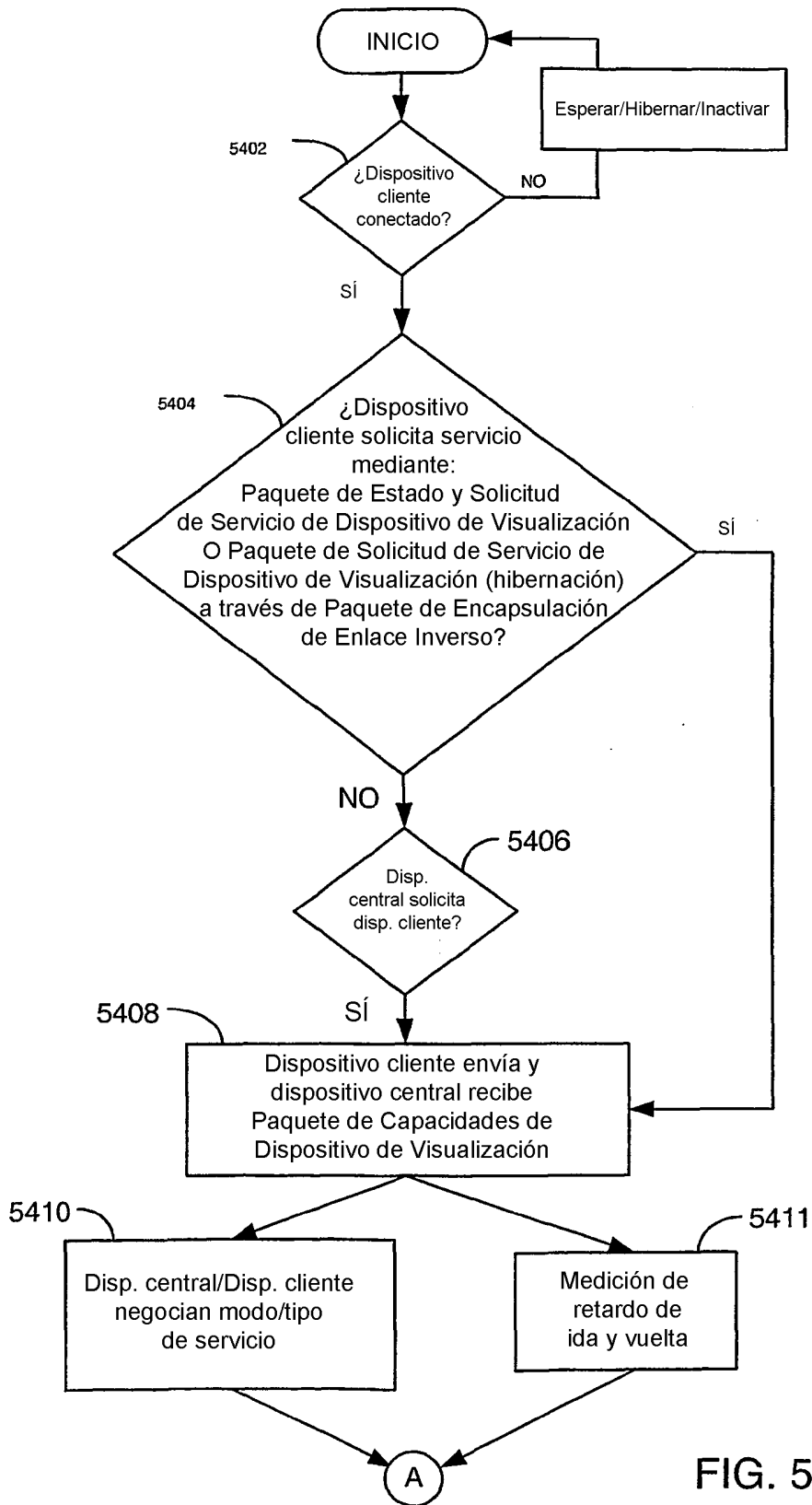


FIG. 54A

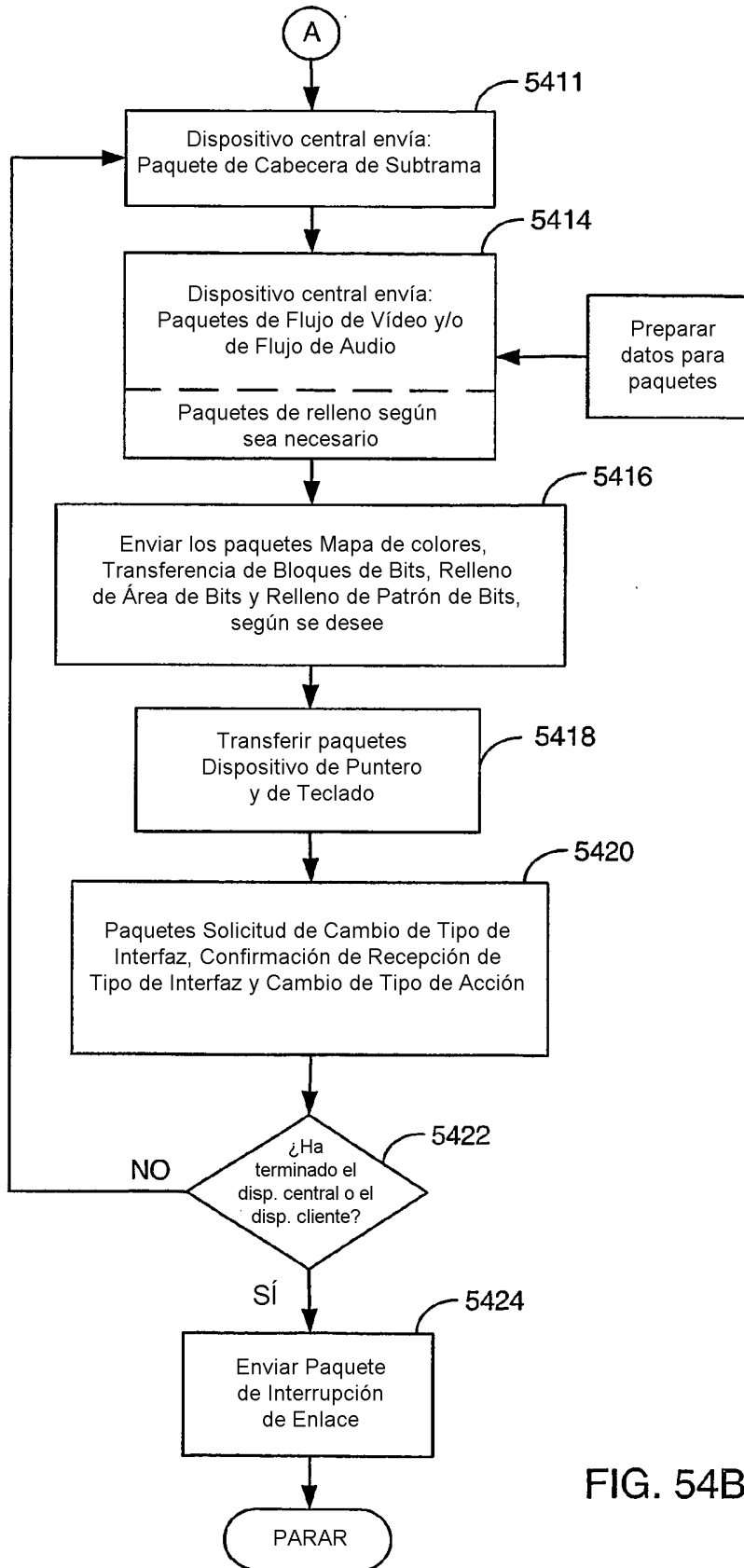


FIG. 54B

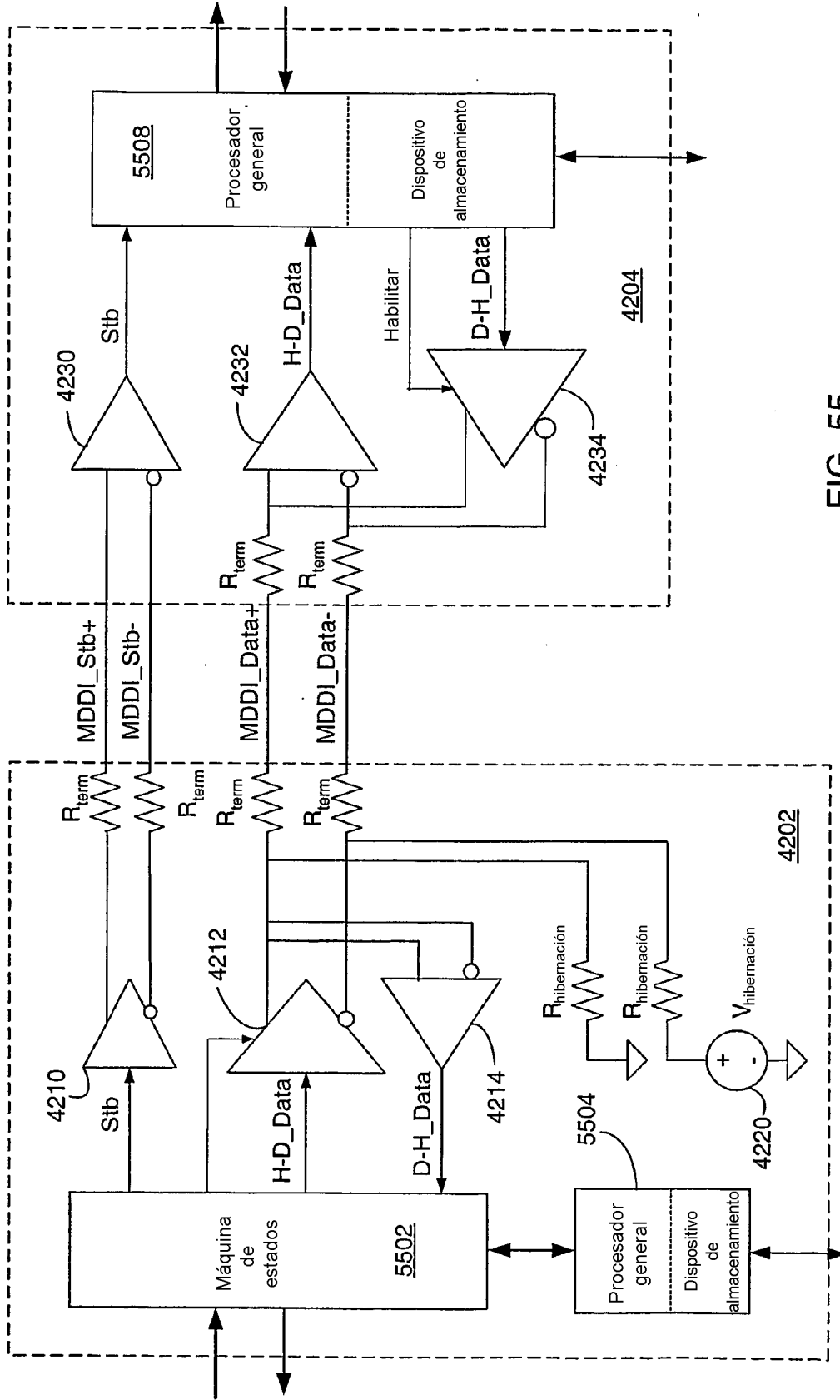


FIG. 55

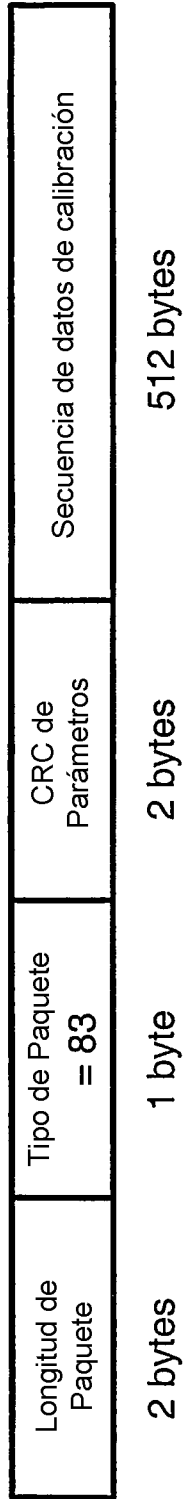


FIG. 56

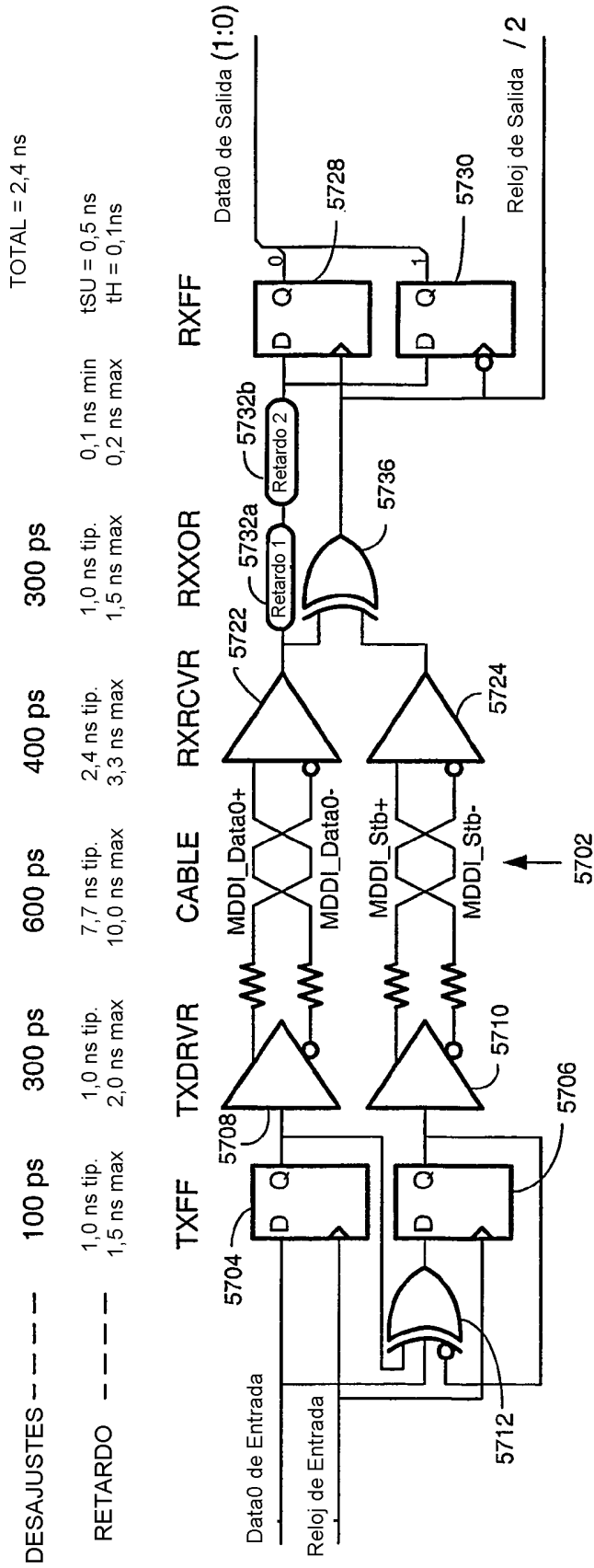


FIG. 57

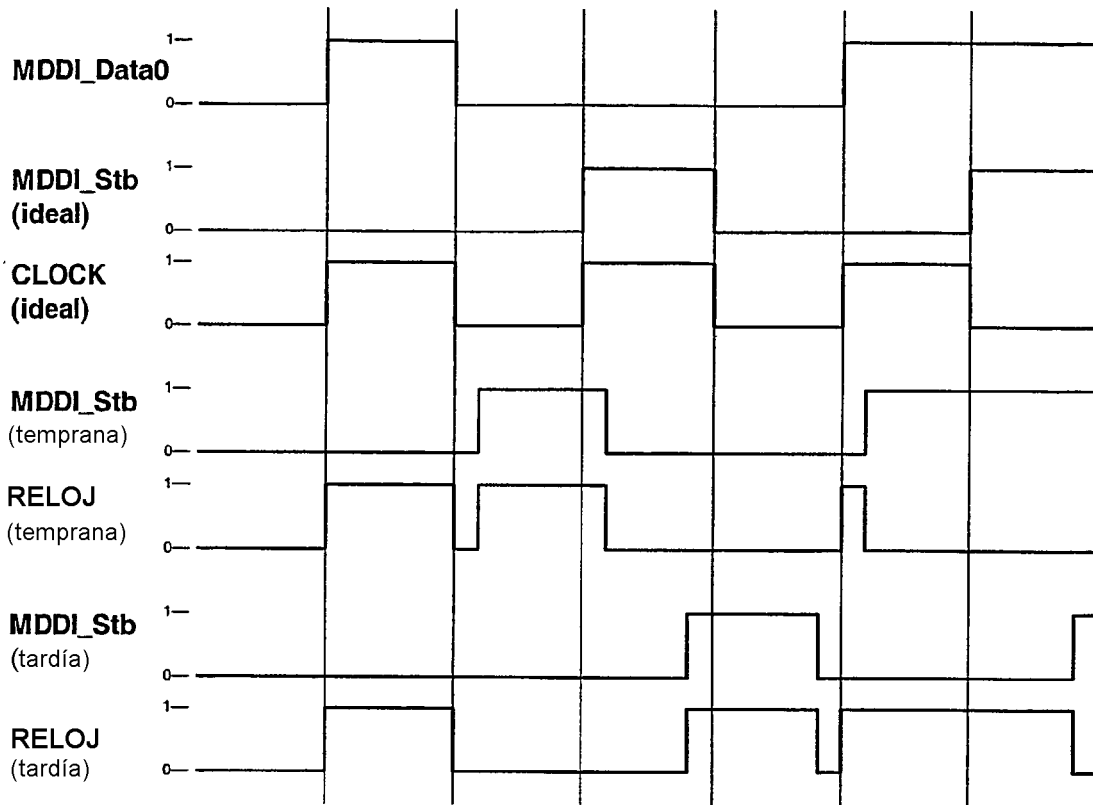


FIG. 58

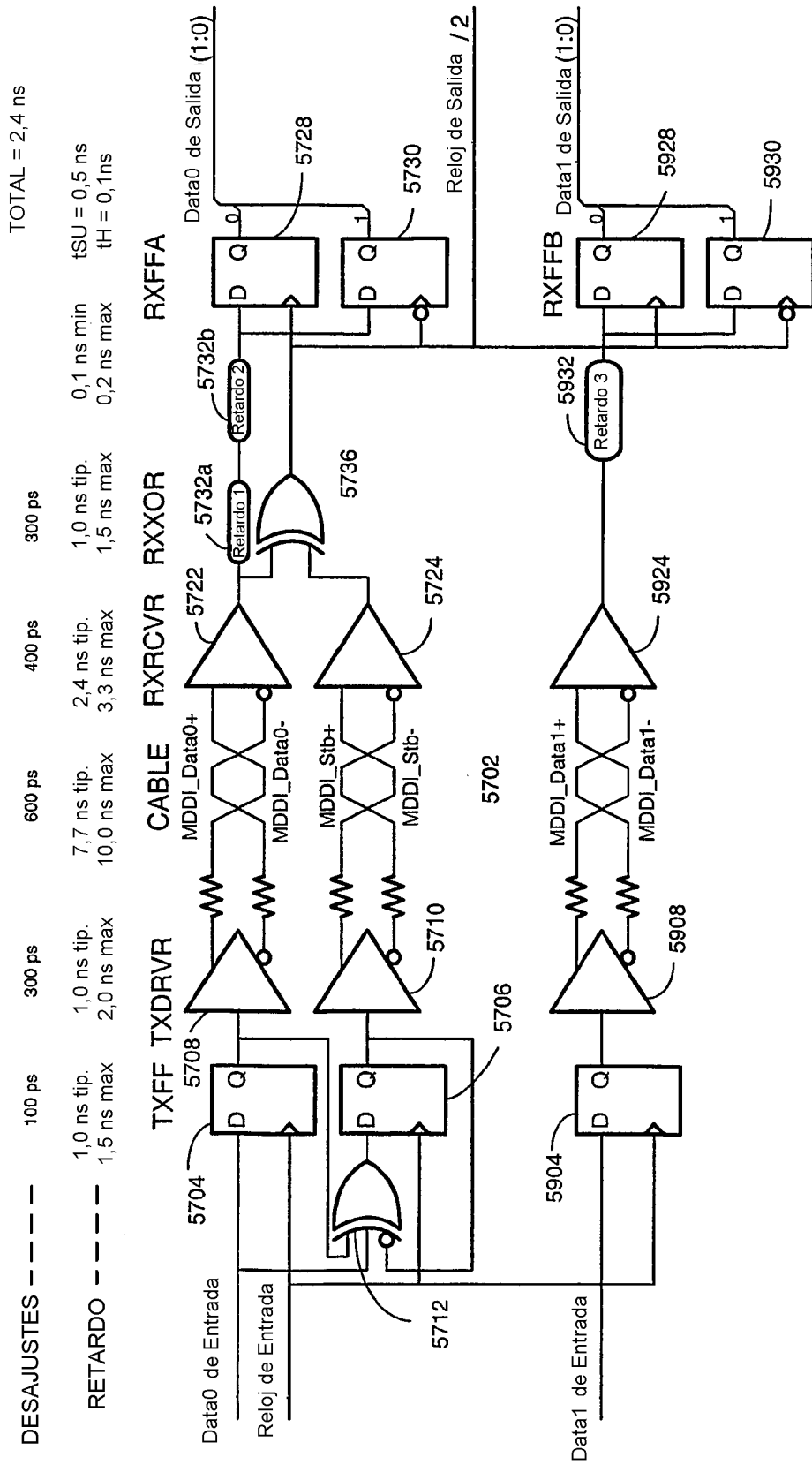


FIG. 59

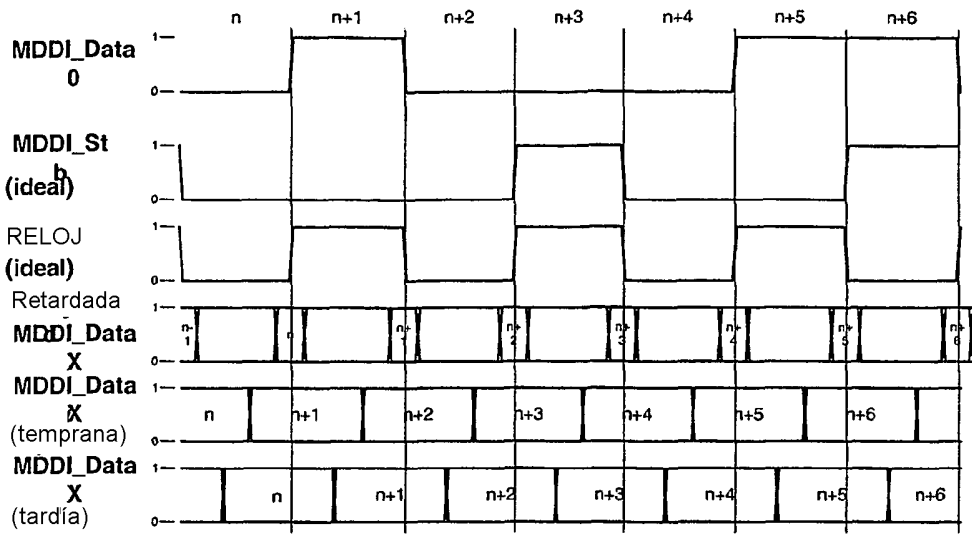


FIG. 60A

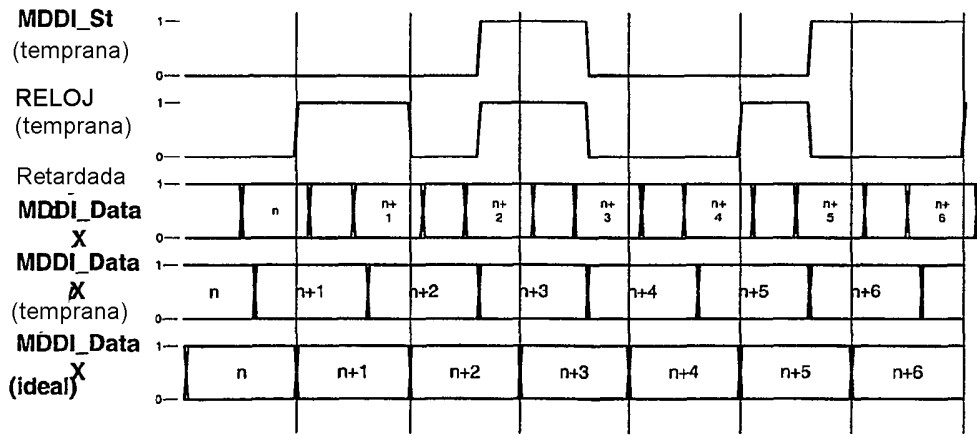


FIG. 60B

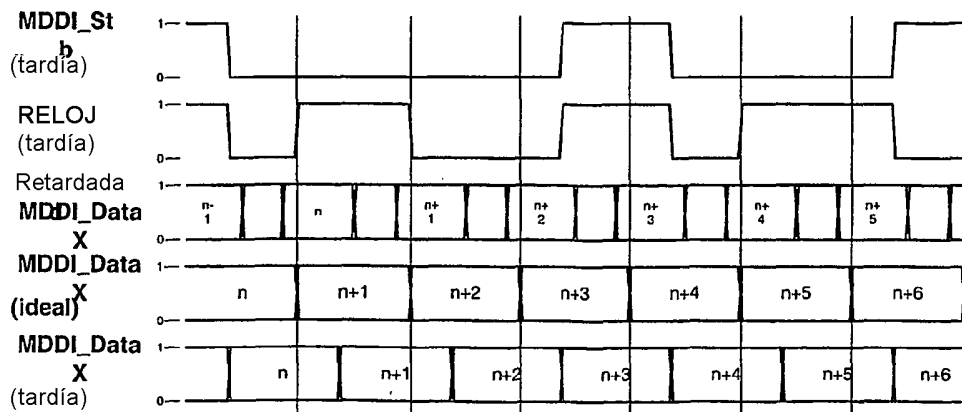


FIG. 60C

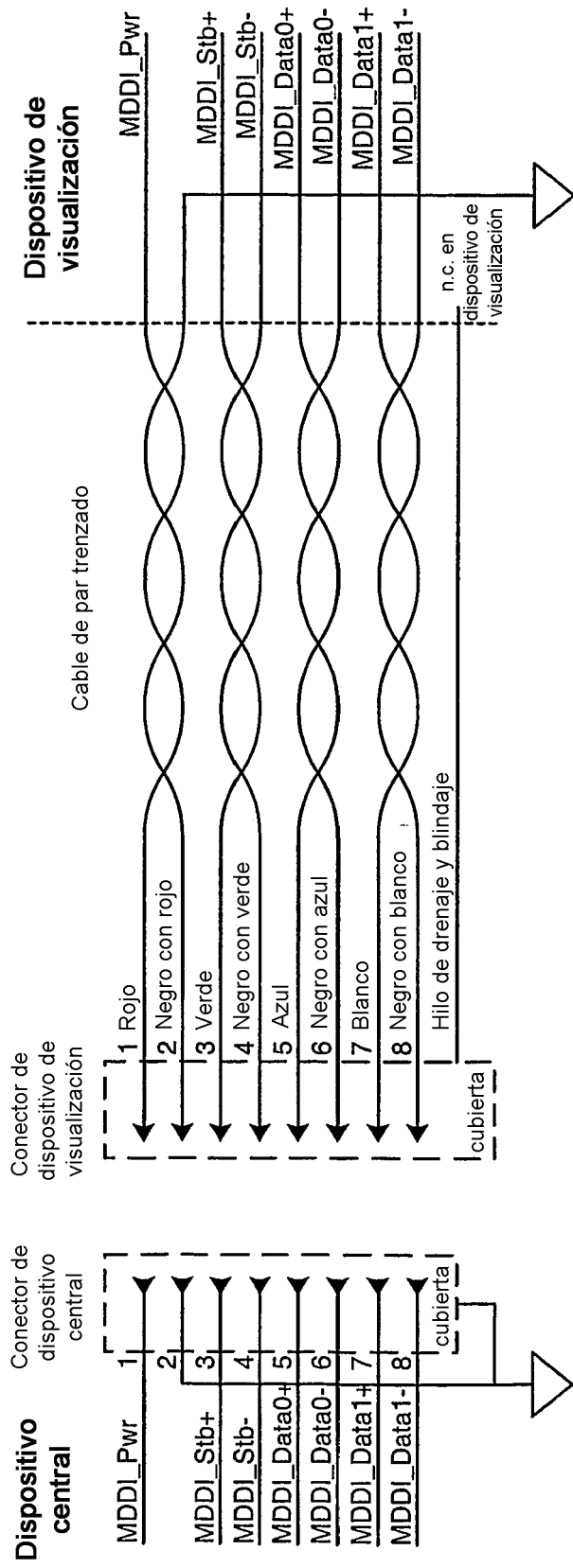
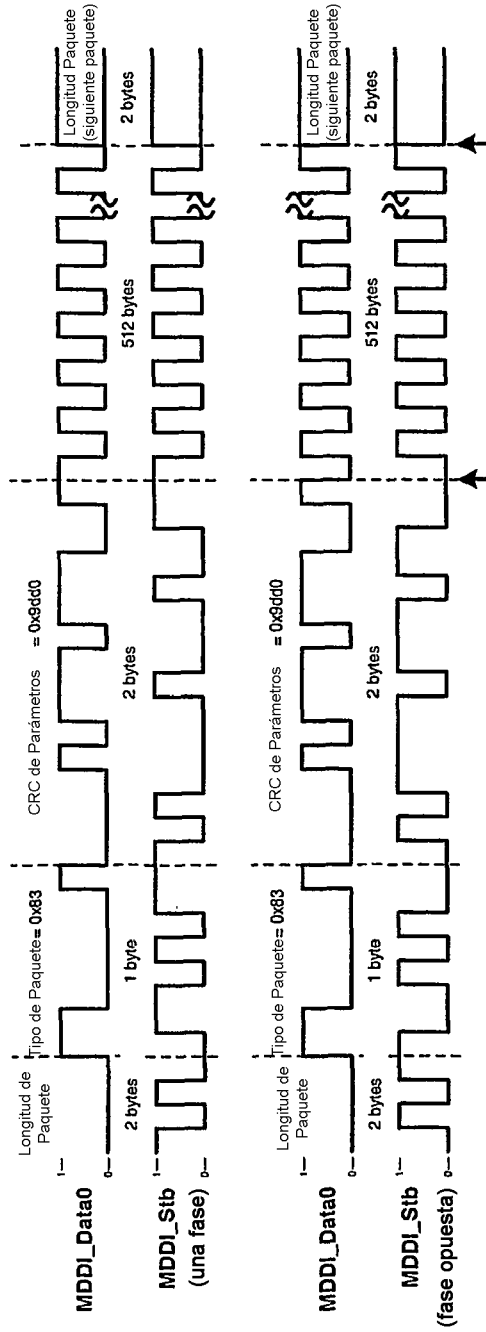


FIG. 61

Ejemplo de datos de Tipo I



MDDI_Sib se comporta como si tuviera todos los datos a cero durante los Datos de Calibración, no hay transición MDDI_Sib porque MSB CRC es uno

Si el LSB de la siguiente Longitud de Paquete es uno, entonces la MDDI_sib oscila

FIG. 62A

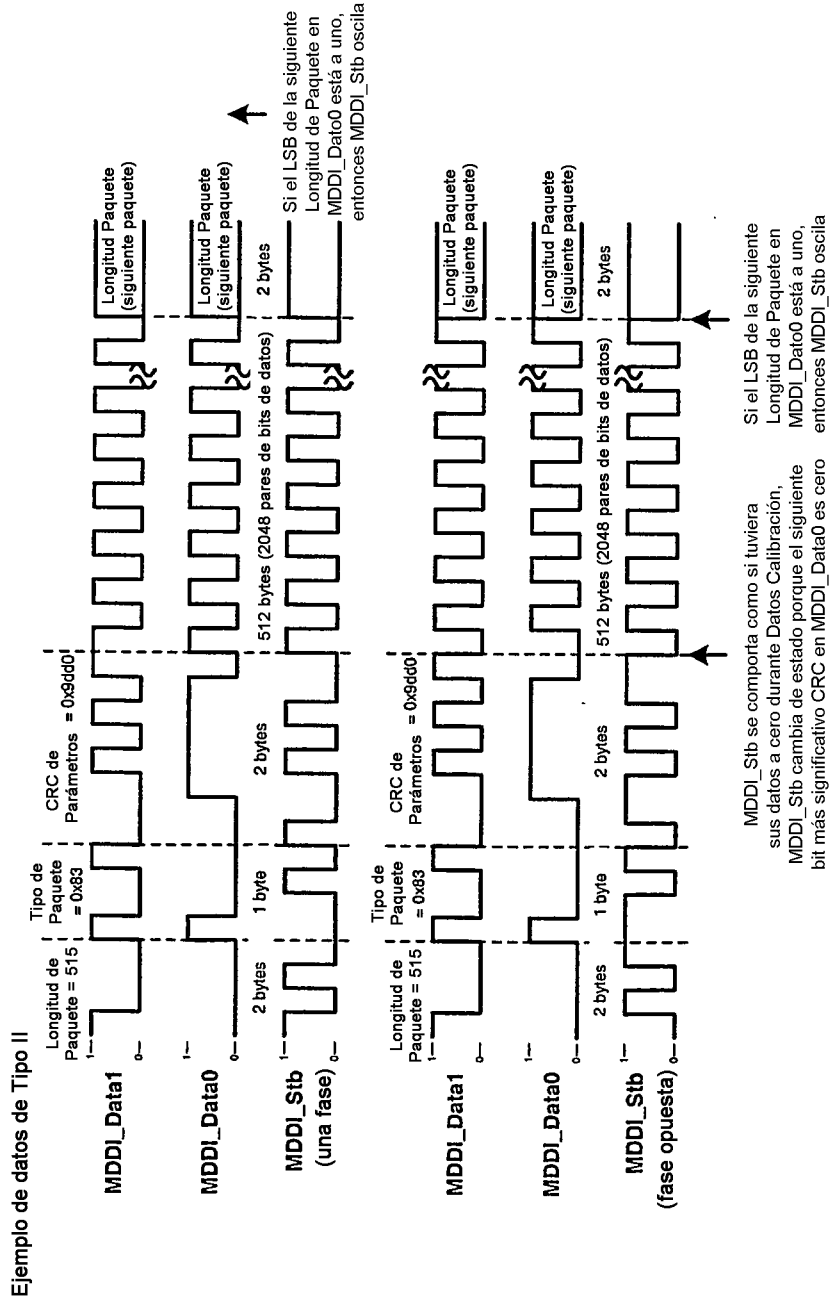
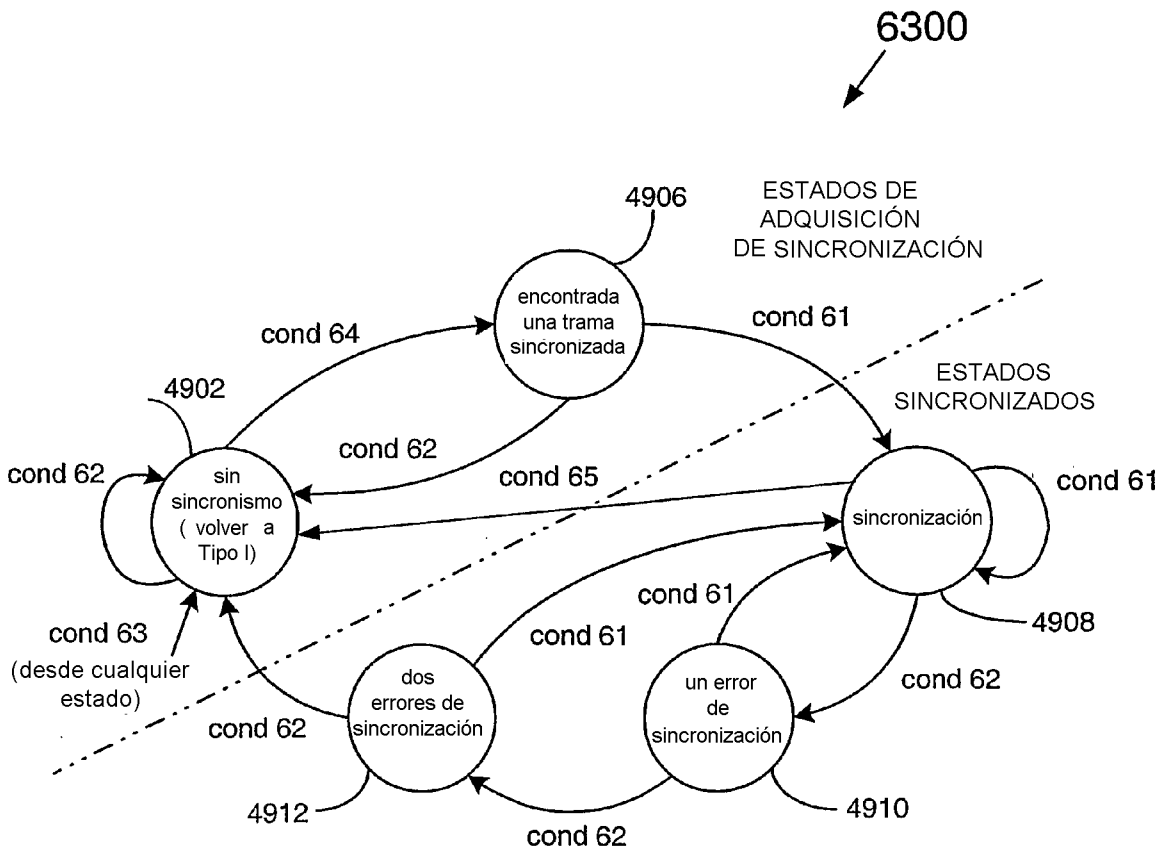


FIG. 62B



cond 61 = paquete de cabecera de subtrama y CRC correcta en límite de subtrama
 cond 62 = patrón de no sincronismo o CRC errónea en límite de subtrama
 cond 63 = paquete de interrupción de enlace recibido
 cond 64 = patrón de sincronismo encontrado
 cond 65 = palabra única incorrecta

FIG. 63

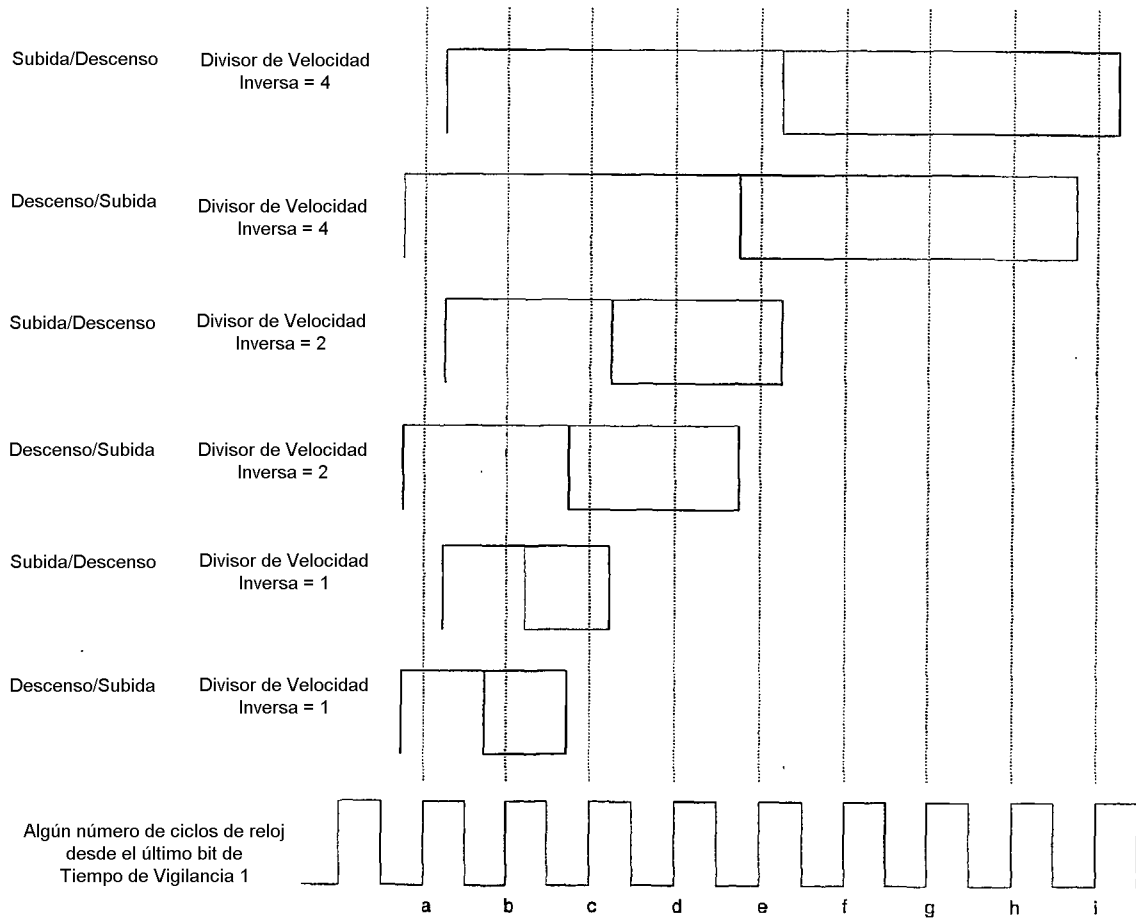


FIG. 64

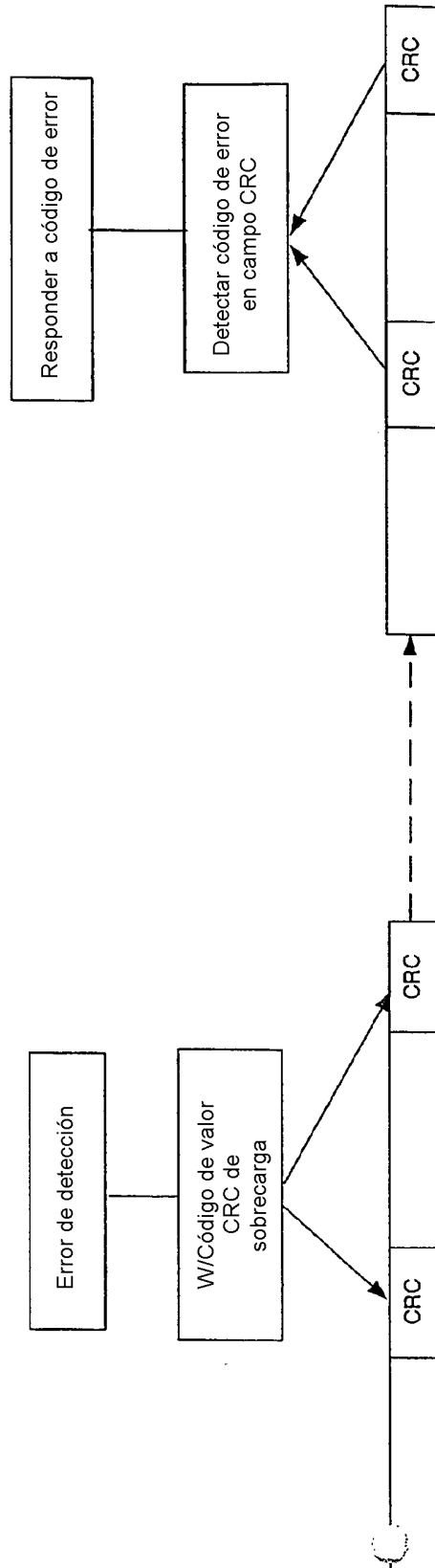


FIG. 65

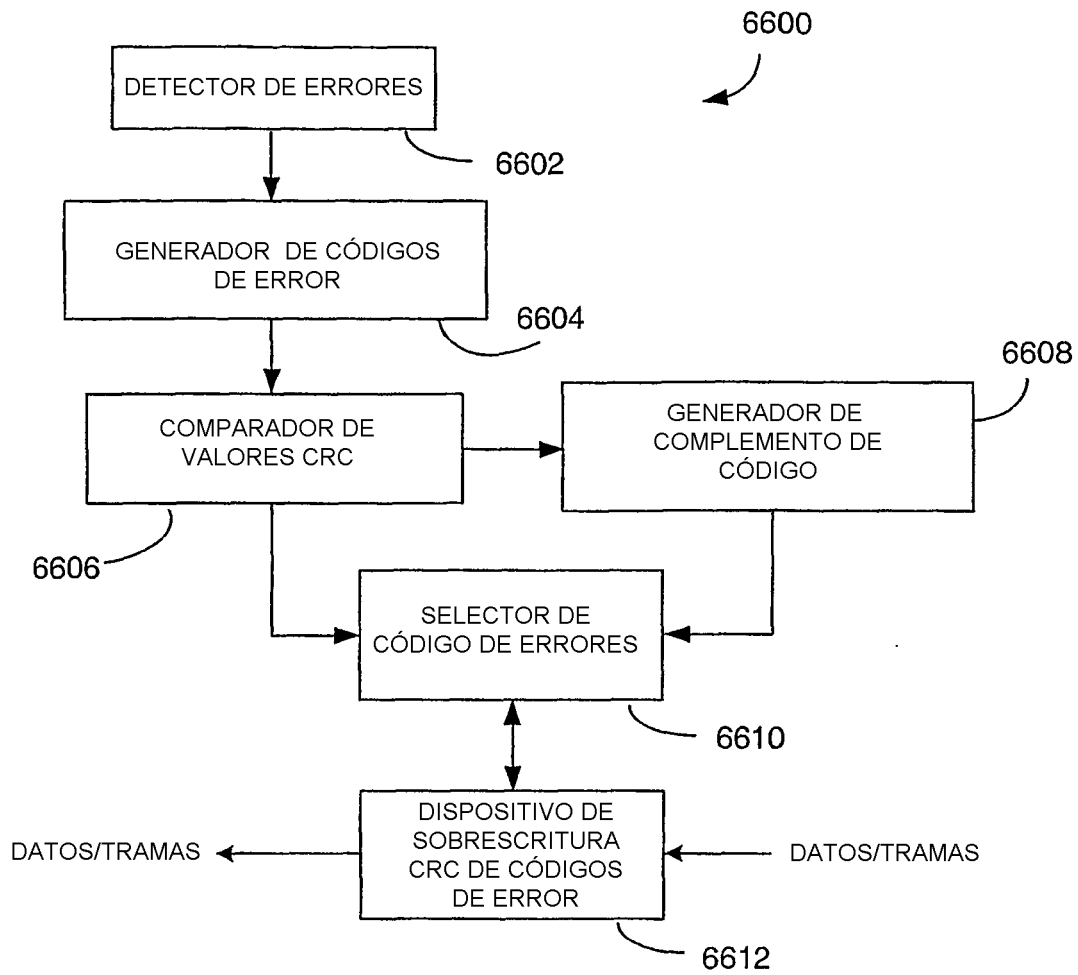


FIG. 66

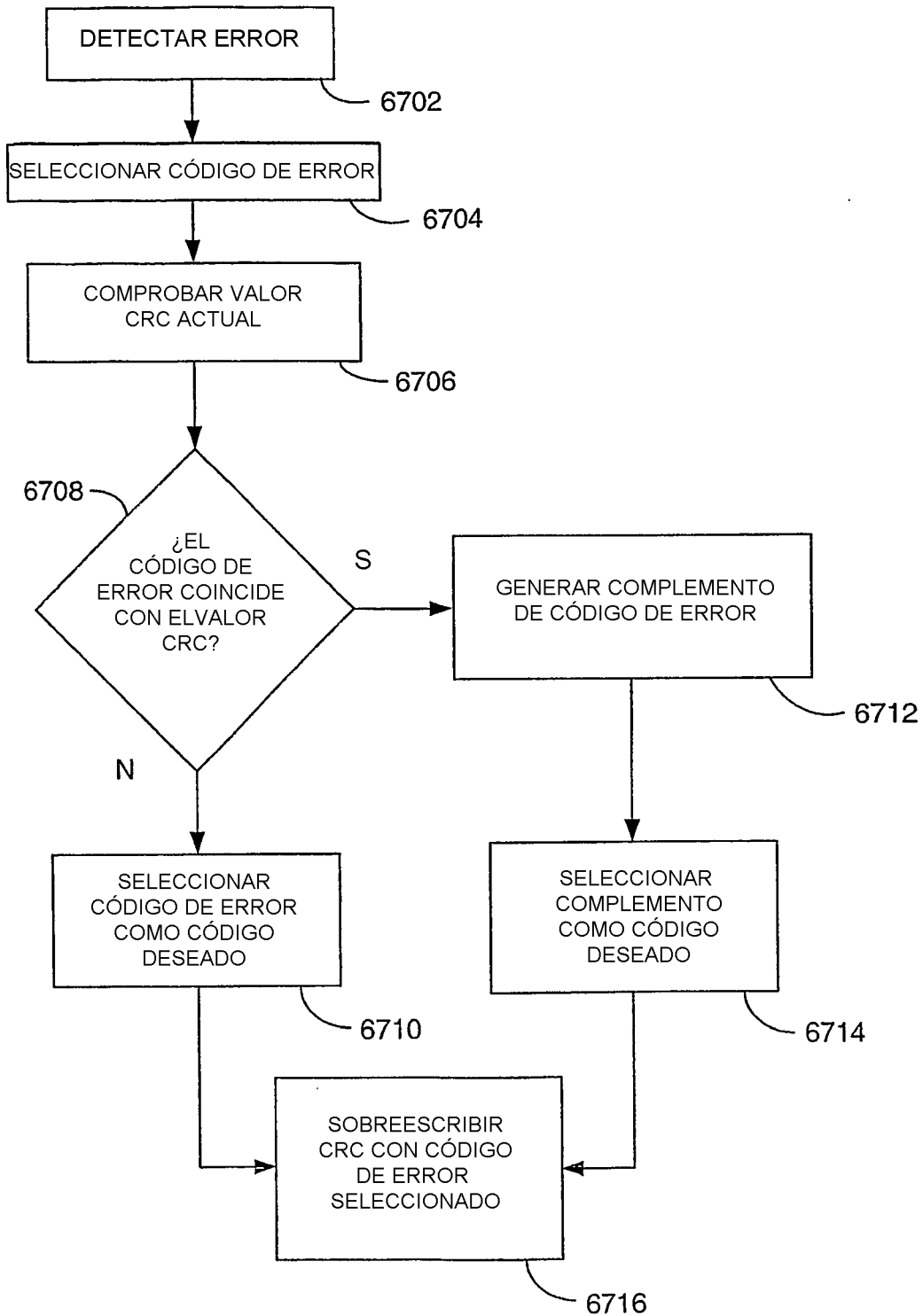


FIG. 67A

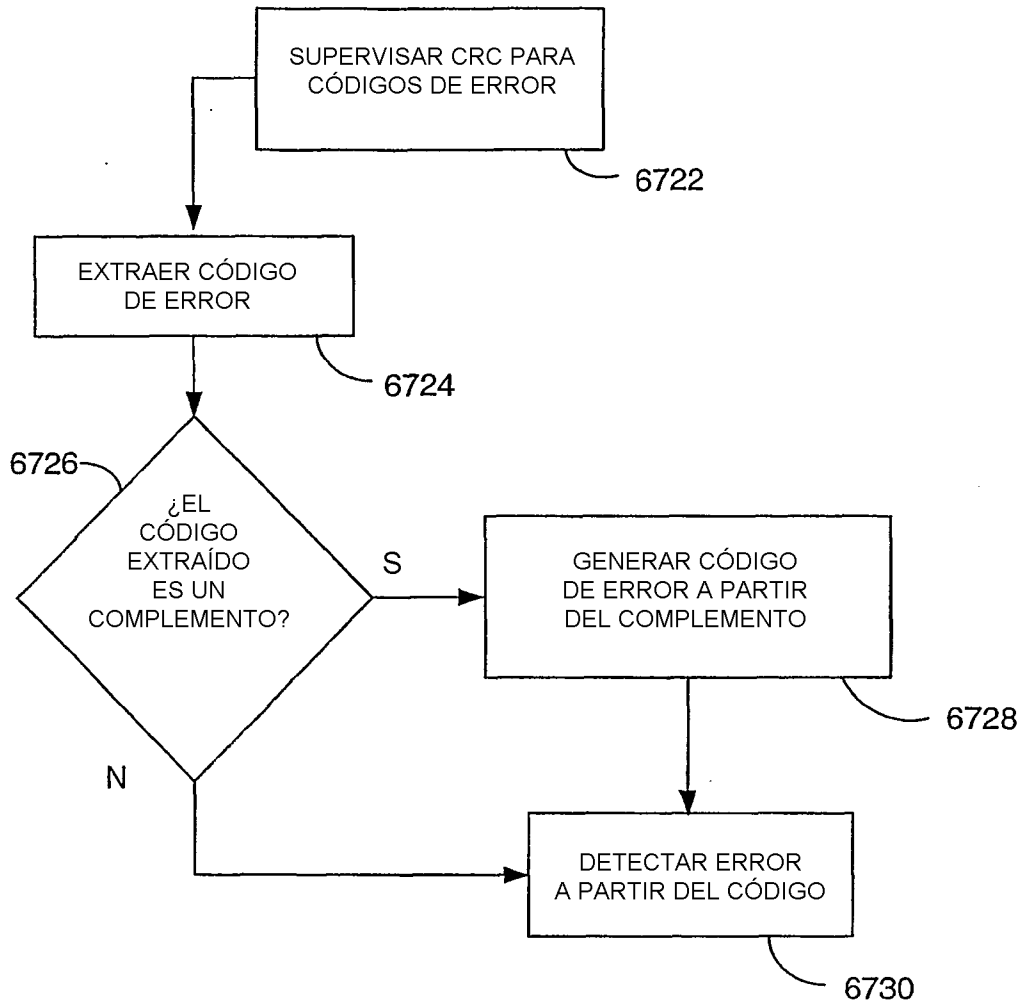


FIG. 67B

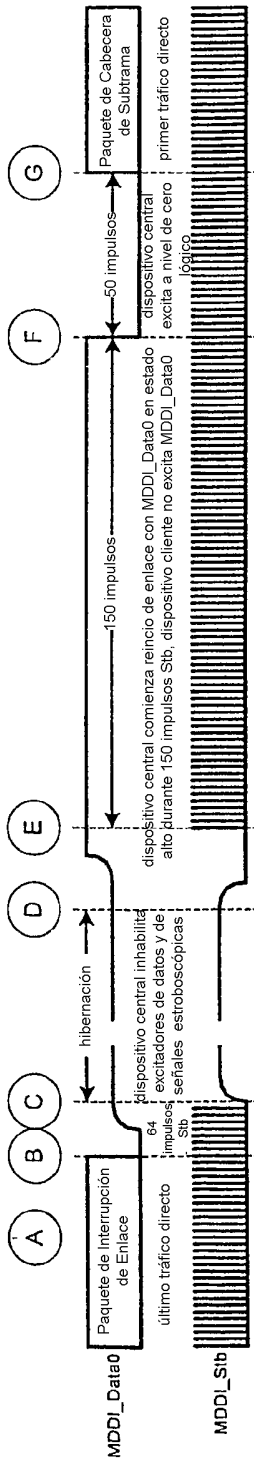


FIG. 68A

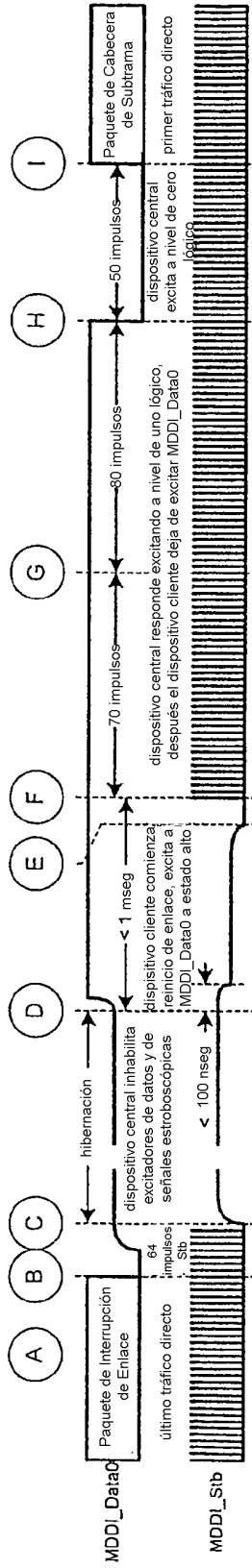


FIG. 68B

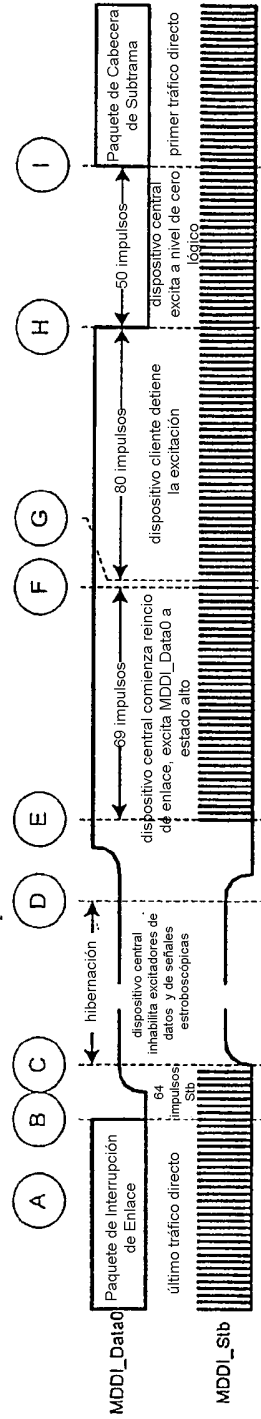


FIG. 68C