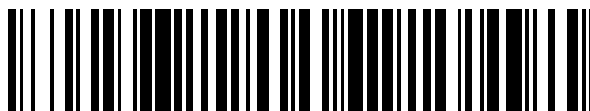


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 759 598**

51 Int. Cl.:

**H04J 3/06** (2006.01)

**G09G 5/00** (2006.01)

**H04W 84/18** (2009.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.05.2008 E 11161950 (8)**

97 Fecha y número de publicación de la concesión europea: **04.09.2019 EP 2337253**

54 Título: **Una estructura de paquetes para una interfaz digital de pantalla móvil**

30 Prioridad:

**08.05.2007 US 928488 P**  
**06.05.2008 US 116018**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**11.05.2020**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**  
**5775 Morehouse Drive**  
**San Diego, CA 92121-1714, US**

72 Inventor/es:

**STEELE, BRIAN;**  
**WILEY, GEORGE ALAN y**  
**SHEKHAR, SHASHANK**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

**ES 2 759 598 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

Una estructura de paquetes para una interfaz digital de pantalla móvil

5 **[0001]** La presente Solicitud de Patente se refiere a la Patente de EE. UU. comúnmente asignada n.º 6.760.772 B2, titulada "Generating and Implementing a Communication Protocol and Interface for High Speed Data Transfer [Generación e implementación de un protocolo de comunicación e interfaz para transferencia de datos de alta velocidad]", emitida el 6 de julio de 2004, y la Patente de EE. UU. n.º 7.315.265 titulada "Double Data Rate Serial Encode [Codificador en serie de velocidad de transferencia de datos doble]", emitida el 1 de enero de 2008.

10

## ANTECEDENTES

## Campo

15 **[0002]** La presente invención se refiere en general a enlaces de comunicación y más en particular a un procedimiento, sistema y producto de programa informático para proporcionar una estructura de paquetes mejorada para enlaces de Interfaz Digital de Pantalla Móvil (MDDI).

## Antecedentes

20

**[0003]** En el campo de las tecnologías de interconexión, la demanda de velocidades de transferencia de datos cada vez mayores, especialmente en relación con las presentaciones de vídeo, continúa creciendo.

25

**[0004]** La Interfaz Digital de Pantalla Móvil (MDDI) es un mecanismo de transferencia rentable y de bajo consumo de energía, que permite la transferencia de datos en serie a muy alta velocidad a través de un enlace de comunicación de corto alcance entre un servidor y un cliente. La MDDI requiere un mínimo de solo cuatro cables más potencia para la transferencia de datos bidireccional que suministra un ancho de banda máximo de hasta 3,2 Gbits por segundo.

30

**[0005]** En una aplicación, la MDDI aumenta la confiabilidad y disminuye el consumo de energía en los teléfonos con cubierta tipo concha al reducir significativamente el número de cables que atraviesan la bisagra de un teléfono para interconectar el controlador digital de banda base con una pantalla LCD y/o una cámara. Esta reducción de cables también permite a los fabricantes de teléfonos móviles reducir los costes de desarrollo al simplificar los diseños de teléfono con cubierta tipo concha o deslizante. Además, la señalización diferencial empleada con la MDDI reduce la Interferencia electromagnética que se puede producir a través de conexiones paralelas tradicionales.

35

**[0006]** Existen algunas mejoras necesarias para los sistemas MDDI actuales. Actualmente, las subtramas contienen una longitud de subtrama fija e intervalos de tiempo. Esto limita el sistema a un número fijo de bits en cada subtrama y funciona a una velocidad fija. Esto evita que los paquetes se extiendan de una subtrama a la siguiente. Los paquetes grandes se deben retrasar hasta que se transmita la siguiente subtrama, desperdiciando el ancho de banda e incrementando la latencia. Se necesita un sistema con longitud de subtrama flexible para transmitir eficazmente estos paquetes grandes. Otra mejora sobre una longitud de subtrama fija es la capacidad de usar una longitud de subtrama ilimitada cuando el enlace salga de la hibernación. Esto también ahorra ancho de banda porque el paquete de encabezado de subtrama se transmite solo una vez para permitir que el cliente se sincronice al inicio.

40

45 **[0007]** Otra mejora necesaria para los sistemas existentes es un procedimiento para evitar la retransmisión repetitiva de determinados datos de paquetes de vídeo cuando algunos de los parámetros no cambien. Nuevamente, esto ahorrará en ancho de banda. Esto se logra al proporcionar un paquete de corriente de vídeo sin ventanas. Adicionalmente, se necesita un sistema para proporcionar una manera de especificar qué campos están contenidos dentro de un paquete de corriente de vídeo cuando algunos valores no han cambiado. Perdería ancho de banda retransmitir repetidamente los campos que contienen valores idénticos a los enviados en paquetes previos. Esto se proporciona en un campo de contenido de paquete del paquete de corriente de vídeo flexible.

50

**[0008]** Los sistemas existentes transmiten primero un paquete de medición de retardo de ida y vuelta y transmiten a continuación un paquete de encapsulación inversa separado para que el servidor reciba datos del cliente. La invención reivindicada actualmente es una mejora significativa sobre los sistemas actuales y combina la funcionalidad de los dos paquetes en un único paquete de encapsulación inversa mejorado.

55

60 **[0009]** El documento WO 2005/091593 A (QUALCOMM INC [EE. UU.]; ANDERSON JON JAMES [EE. UU.]; STEELE BRIAN [EE. UU.]; WILEY G) 29 de septiembre de 2005 (29-09-2005) divulga y mejora el paquete de encapsulación de enlace inverso.

60

## SUMARIO

65 **[0010]** Los aspectos de la invención reivindicada, divulgada en el presente documento, abordan las necesidades indicadas anteriormente proporcionando un procedimiento, sistema y producto de programa informático que usa una estructura de trama que comprende una longitud de subtrama flexible. La subtrama flexible envía un paquete de

encabezado de subtrama en el límite de subtrama con una indicación de una longitud de subtrama. Cuando se solicita que se transmita un paquete a través de la interfaz MDDI, no se bloquea su transmisión debido a la falta de espacio restante en la subtrama actual. Esto puede causar que un paquete cruce uno o más límites de subtrama. Si se cruza un límite de subtrama, otro paquete de encabezado de subtrama es el primer paquete transmitido después del paquete que cruzó el límite. Este segundo subtrama se acorta una cantidad igual a la cantidad que la subtrama previa va más allá de la longitud de subtrama. Esto mantiene una temporización que, en promedio, es similar a la temporización de subtrama usando una longitud de subtrama fija, pero no evita la transmisión de ninguna longitud de paquete. Además, esto permite que los paquetes de encabezado de subtrama se transmitan de forma semiperiódica en caso de que el cliente pierda la sincronización. También se puede implementar una longitud de subtrama ilimitada, con lo que solo se transmite un paquete de encabezado de subtrama cuando un enlace sale de la hibernación y la subtrama que contiene los datos del paquete comprende una longitud ilimitada.

**[0011]** Otro aspecto único introducido es un paquete de datos de vídeo sin ventanas. Este aspecto permite que un tamaño de ventana definido por primera vez se reutilice sin tener que redefinir la ventana. Esto se logra retirando las coordenadas de campo X Izquierda, X Derecha, Y Superior, Y Inferior, X Inicio e Y Inicio del paquete de datos de vídeo. Un bit dentro del campo existente representa la sincronización vertical e identifica la primera línea de una pantalla de datos.

**[0012]** También se introduce una subtrama flexible para transmitir eficazmente paquetes grandes. Adicionalmente, se divulga un paquete de datos de vídeo flexible que contiene un campo que indica qué campos opcionales del paquete de vídeo flexible están presentes en el paquete transmitido.

**[0013]** La invención reivindicada comprende un paquete de encapsulación de enlace inverso mejorado. El paquete de encapsulación de enlace inverso mejorado combina las características de un paquete de retardo de ida y vuelta con un paquete de encapsulación inversa en un solo paquete. La primera parte de la transmisión inversa es un preámbulo que permite que el servidor se sincronice con los datos de enlace inverso de modo que pueda muestrear con exactitud los datos inversos. La segunda porción de los datos inversos contiene un recuento de bytes. Esto permite que el ancho de banda dinámico del enlace inverso se asigne en base a las necesidades del cliente. El servidor puede establecer un umbral de límite superior de estos datos inversos con el campo de bytes máximos.

**[0014]** La presente invención está definida por las reivindicaciones adjuntas y limitada por el alcance de las reivindicaciones. El(los) modo(s) de realización o el(los) aspecto(s) al(a los) que se hace referencia en esta descripción y que no cae(n) completamente dentro del alcance de las reivindicaciones adjuntas es(son) un(os) ejemplo(s) adecuado(s) para comprender la presente invención.

**[0015]** Otro aspecto introducido en el presente documento es una congelación de enlaces. Esto detiene o congela la transmisión de la corriente de datos en cualquier punto dentro de la corriente de datos por parte del servidor. El cliente se desconecta por medio de la corriente de datos de MDDI entrantes, por lo que el resultado de detener el enlace MDDI es que los ciclos de reloj ya no existen dentro del cliente. El servidor mantiene los niveles diferenciales correspondientes al último bit de datos transmitido al entrar en este modo. La corriente de datos se puede reanudar a continuación por el servidor.

**[0016]** Otros aspectos, rasgos característicos y ventajas de la presente invención reivindicada, así como la estructura y el funcionamiento de los diversos modos de realización de la presente invención reivindicada, se describen en detalle a continuación con referencia a los dibujos adjuntos.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

**[0017]**

La Fig. 1 es un diagrama de bloques que ilustra un entorno de ejemplo que usa una interfaz MDDI;

la Fig. 2A muestra una estructura de paquete de MDDI típica;

la Fig. 2B representa una estructura de enlace directo típica;

la Fig. 3 muestra la subtrama de la técnica anterior con una longitud fija;

la Fig. 4 representa la subtrama de longitud flexible;

la Fig. 5 representa la subtrama de longitud ilimitada;

la Fig. 6 muestra el paquete de corriente de vídeo sin ventanas;

la Fig. 7 muestra el paquete de corriente de vídeo flexible;

la Fig. 8 muestra el paquete de encapsulación de enlace inverso mejorado; y

la Fig. 9 representa la congelación de enlaces.

5 **DESCRIPCIÓN DETALLADA**

[0018] El término "ejemplar" se usa en el presente documento para significar "que sirve de ejemplo, caso o ilustración". Cualquier aspecto descrito en el presente documento como "ejemplar" no ha de interpretarse necesariamente como preferente o ventajoso con respecto a otros aspectos.

10 [0019] Los aspectos descritos, y las referencias en la memoria descriptiva a "un aspecto", "un aspecto de ejemplo", etc., indican que los aspectos descritos pueden incluir un rasgo característico, una estructura o una característica particular, pero cada aspecto puede no incluir necesariamente el rasgo característico, la estructura o la característica particular. Además, dichas frases no se refieren necesariamente al mismo modo de realización. Además, cuando un rasgo característico, una estructura o una característica particular se describe en relación con un aspecto, se afirma que está dentro del conocimiento de un experto en la técnica afectar dicho rasgo característico, dicha estructura o dicha característica en relación con otros aspectos, ya se describan o no explícitamente.

15 [0020] La Interfaz Digital de Pantalla Móvil (MDDI) es un mecanismo de transferencia rentable y de bajo consumo de energía que permite la transferencia de datos en serie a muy alta velocidad a través de un enlace de comunicación de corto alcance entre un servidor y un cliente. Para apreciar completamente los nuevos rasgos característicos introducidos en el presente documento, se proporciona un breve análisis sobre el sistema MDDI.

20 [0021] A continuación, se presentarán ejemplos de MDDI con respecto a un módulo de cámara contenido en una cubierta tipo concha superior de un teléfono móvil. Sin embargo, sería evidente para las personas expertas en la(s) técnica(s) pertinente(s) que cualquier módulo que tenga características funcionalmente equivalentes al módulo de cámara se podría sustituir fácilmente y usar en aspectos de la presente invención.

25 [0022] Además, de acuerdo con los aspectos de la invención, un servidor MDDI puede comprender uno de varios tipos de dispositivos que se pueden beneficiar del uso de la presente invención reivindicada. Por ejemplo, el servidor podría ser un ordenador portátil en forma de teléfono, ordenador portátil o dispositivo informático móvil similar. También podría ser un Asistente de Datos Personal (PDA), un dispositivo de paginación o uno de los muchos teléfonos o módems inalámbricos.

30 [0023] De forma alternativa, el servidor podría ser un dispositivo portátil de entretenimiento o presentación, tal como un reproductor portátil de DVD o CD, o un dispositivo de reproducción de juegos. Además, el servidor puede residir como un dispositivo de servidor o elemento de control entre una variedad de productos comerciales, ampliamente usados o planificados, para los cuales se desea un enlace de comunicación de alta velocidad con un cliente. Por ejemplo, un servidor se podría usar para transferir datos a altas velocidades desde un dispositivo de grabación de vídeo a un cliente basado en el almacenamiento para mejorar la respuesta, o a una pantalla más grande de alta resolución para presentaciones. Un electrodoméstico, tal como un frigorífico, que incorpora un inventario a bordo o un sistema informático y/o conexiones Bluetooth a otros dispositivos domésticos, puede tener capacidades de visualización mejores cuando funcione en un modo conectado a Internet o Bluetooth, o puede tener una menor necesidad de cableado para pantallas de puerta (un cliente) y teclados o escáneres (cliente) mientras el ordenador electrónico o los sistemas de control (servidor) residen en otra parte del armario. En general, los expertos en la técnica apreciarán la amplia variedad de dispositivos y accesorios electrónicos modernos que se puedan beneficiar del uso de esta interfaz, así como la capacidad de retroadaptar dispositivos más antiguos con un transporte de información a una mayor velocidad de transferencia de datos utilizando números limitados de conductores disponibles en conectores o cables recién añadidos o existentes. Al mismo tiempo, un cliente de MDDI puede comprender una variedad de dispositivos útiles para presentar información a un usuario final, o presentar información desde un usuario al servidor. Por ejemplo, una micropantalla incorporada en anteojos o gafas, un dispositivo de proyección incorporado en un sombrero o casco, una pantalla pequeña o incluso un elemento holográfico incorporado en un vehículo, tal como en una ventana o parabrisas, o diversos altavoces, auriculares o sistemas de sonido para presentar sonido o música de alta calidad. Otros dispositivos de presentación incluyen proyectores o dispositivos de proyección usados para presentar información para conferencias, o para películas e imágenes de televisión. Otro ejemplo podría ser el uso de almohadillas táctiles o dispositivos sensibles, dispositivos de entrada de reconocimiento de voz, escáneres de seguridad, etc., que se pueden solicitar para transferir una cantidad significativa de información desde un dispositivo o usuario del sistema con poca "entrada" real que no sea tacto o sonido del usuario. Además, las estaciones de acoplamiento para ordenadores y los equipos de automóviles o equipos de sobremesa y los soportes para teléfonos inalámbricos pueden actuar como dispositivos de interfaz para usuarios finales u otros dispositivos y equipos, y emplear clientes (dispositivos de salida o de entrada, tal como ratones) o servidores para ayudar a la transferencia de datos, especialmente cuando se trata de redes de alta velocidad. Sin embargo, los expertos en la técnica admitirán fácilmente que la presente invención no se limita a estos dispositivos, ya que existen muchos otros dispositivos en el mercado, y propuestos para su uso, que están previstos para proporcionar a los usuarios finales imágenes y sonido de alta calidad, ya sea en términos de almacenamiento y transporte o en términos de presentación en reproducción.

La presente invención es útil para incrementar el rendimiento de los datos entre diversos elementos o dispositivos, para alojar las altas velocidades de transferencia de datos necesarias para realizar la experiencia de usuario deseada.

**[0024]** La FIG. 1 es un diagrama de bloques que ilustra un entorno de ejemplo que usa una interfaz MDDI. En el ejemplo de la Fig. 1, la MDDI se usa para interconectar módulos a través de la bisagra de un teléfono con cubierta tipo concha 100. Debe señalarse aquí que, mientras que determinados aspectos de la invención reivindicada actualmente se describirán en el contexto de ejemplos específicos, tales como las interconexiones de MDDI en un teléfono con cubierta tipo concha, esto se hace solo con propósitos ilustrativos y no debe usarse para limitar la presente invención a dichos aspectos. Como entenderá una persona experta en la(s) técnica(s) pertinente(s) en base a las enseñanzas en el presente documento, los aspectos de la invención reivindicada actualmente se pueden usar en otros dispositivos que incluyan cualquiera que se pueda beneficiar de tener interconexiones de MDDI.

**[0025]** En referencia a la Fig. 1, una sección de cubierta tipo concha inferior 104 del teléfono con cubierta tipo concha 100 incluye un chip de banda base de módem de estación móvil (MSM) 102. El MSM 102 es un controlador de banda base digital. Una sección de cubierta tipo concha superior 114 del teléfono con cubierta tipo concha 100 incluye un módulo de pantalla de cristal líquido (LCD) 116 y un módulo de cámara 118.

**[0026]** Todavía en referencia a la Fig. 1, un enlace MDDI 110 conecta el módulo de cámara 118 al MSM 102. Típicamente, un controlador de enlace MDDI está integrado en cada uno del módulo de cámara 118 y del MSM 102. En el ejemplo de la Fig. 1, un servidor MDDI 122 está integrado en el módulo de cámara 118, mientras que un cliente de MDDI 106 reside en el lado de MSM del enlace MDDI 110. Típicamente, el servidor MDDI es el controlador maestro del enlace MDDI. En el ejemplo de la Fig. 1, los datos de píxeles del módulo de cámara 118 se reciben y formatean en paquetes de MDDI por el servidor MDDI 122 antes de transmitirse al enlace MDDI 110. El cliente de MDDI 106 recibe los paquetes de MDDI y los vuelve a convertir en datos de píxeles del mismo formato que el generado por el módulo de cámara 118. Los datos de píxeles se envían a continuación a un bloque apropiado en el MSM 102 para su procesamiento.

**[0027]** Todavía en referencia a la Fig. 1, un enlace MDDI 112 conecta el módulo de LCD 116 al MSM 102. En el ejemplo de la Fig. 1, el enlace MDDI 112 interconecta un servidor MDDI 108, integrado en el MSM 102, y un cliente de MDDI 120 integrado en el módulo LCD 116. En el ejemplo de la Fig. 1, los datos de visualización generados por un controlador de gráficos del MSM 102 se reciben y formatean en paquetes de MDDI por el servidor MDDI 108 antes de transmitirse al enlace MDDI 112. El cliente de MDDI 120 recibe los paquetes de MDDI y los vuelve a convertir en datos de visualización para su uso por el módulo LCD 116.

### **Estructura de trama**

**[0028]** La estructura de trama original se describe en la patente de EE. UU. n.º 6.760.772 B2, titulada "Generating and Implementing a Communication Protocol and Interface for High Data Rate Signal Transfer", emitida el 6 de julio de 2004 (patente '772). Esta estructura de paquete original 200 se muestra en la Fig. 2A. Los campos representados en la Fig. 2A incluyen la longitud de paquete 202, que típicamente es un valor de 16 bits que especifica el número total de bytes en el paquete, sin incluir el campo de longitud de paquete 202, el tipo de paquete 204, que es un número entero sin signo de 16 bits que especifica el tipo de información contenida en el paquete 200, bytes de datos 206, que son los datos enviados entre el servidor y el cliente, y la CRC 208, que es una verificación de redundancia cíclica de 16 bits calculada sobre los bytes de datos 206, el tipo de paquete 204 y los campos de longitud de paquete 202.

**[0029]** Como se muestra en la Fig. 2B, la información transmitida a través del enlace MDDI se agrupa en paquetes. Múltiples paquetes se agrupan en una subtrama 210, y múltiples subtramas forman una trama de medios 212. Cada subtrama 210 comienza con un paquete de encabezado de subtrama 214.

**[0030]** La Fig. 3 muestra la subtrama de la técnica anterior con una longitud fija. Se muestra el paquete de encabezado de subtrama 214, el paquete de datos 216 seguido de los paquetes de relleno 218 y el límite de subtrama 220. El problema surge cuando el paquete saliente pendiente 222 es demasiado grande para caber dentro de la porción restante de una subtrama 218, como se muestra. Por tanto, el paquete saliente pendiente 222 debe esperar hasta que se transmita la siguiente subtrama. En cambio, los paquetes de relleno se transmiten durante la subtrama actual. Esto desperdicia el ancho de banda y consume innecesariamente potencia adicional. Las nuevas estructuras de trama descritas a continuación pueden funcionar en el mismo modo que se divulga en la patente '772; sin embargo, se proporcionan dos modos operativos nuevos que modifican la definición de longitud y la temporización de subtrama, mejorando por tanto el rendimiento. Los dos modos operativos enumerados a continuación se identificarán con un campo de versión de protocolo contenido dentro del paquete de encabezado de subtrama. Para garantizar la compatibilidad, para los dispositivos del cliente que no están conectados a un servidor, se puede abrir el enlace MDDI, adhiriéndose primero a la estructura de trama de la técnica anterior para verificar que el cliente puede admitir las nuevas estructuras de trama presentadas a continuación. Una vez verificado, el servidor se puede mover a la nueva estructura de trama. Todo esto se puede hacer en la primera subtrama para proporcionar una transición rápida a cualquiera de los dos formatos que se describen a continuación.

### Longitud de subtrama flexible

[0031] El primer modo de funcionamiento proporciona una longitud de subtrama "flexible", como se muestra en la Fig. 4. La subtrama de longitud flexible **300** tiene un paquete de encabezado de subtrama **304**, un paquete de datos **316** y un límite de subtrama identificado **320**. El subtrama de longitud flexible **300** envía un paquete de encabezado de subtrama **304** en el límite de subtrama **320**. Cuando se solicita que se transmita un paquete, nunca se bloqueará, incluso aunque el paquete cruce uno o más límites de subtrama **320**. Este modo de funcionamiento permite que el servidor MDDI transmita el siguiente paquete de encabezado de subtrama **304'** en la próxima oportunidad disponible después de que se haya completado el número de bytes transmitidos en el campo de longitud de subtrama, incluidos los datos de prórroga **322**. La ventaja de este modo de funcionamiento es que ya no será necesario dividir un paquete entre dos subtramas si los datos se vuelven disponibles al final para la primera subtrama para la transmisión. Del mismo modo, también evita retardos hasta que la siguiente subtrama se transmita para un paquete que no cabrá en los bytes restantes para la subtrama actual. Los paquetes de encabezado de subtrama **304'** deberían ser el primer paquete transmitido después de que el paquete actual complete el número total de bytes transmitidos en la subtrama actual a través de la longitud de subtrama especificada en el extremo de paquete **324**. Este procedimiento sigue proporcionando paquetes de encabezado de subtrama **304** que proporcionan puntos de resincronización para enlaces de transmisión que no sean completamente confiables. El texto enviado en una subtrama después de una larga subtrama se acorta por la cantidad que la subtrama larga previa pasa para crear una longitud promedio de subtrama. El concepto de longitud de subtrama flexible mantiene la temporización que, en promedio, debería ser similar a la temporización de subtrama del sistema de longitud de subtrama fija de la Fig. 3, pero nunca evita la transmisión de un paquete y no desperdicia el ancho de banda.

### Longitud de subtrama ilimitada

[0032] Este segundo modo operativo permite que el servidor use solo una subtrama durante el enlace MDDI activo, como se muestra en la Fig. 5. Esto significa que el servidor MDDI transmitirá solo un paquete de encabezado de subtrama **404** cuando el enlace salga de la hibernación y ya no transmita. La ventaja de este modo operativo es que no se usa ningún ancho de banda adicional para transmitir otros paquetes de encabezado de subtrama. Todavía está permitido transmitir paquetes de encabezado de subtrama mientras está en este modo operativo para permitir una resincronización, sin embargo, el número de bytes entre estos paquetes será arbitrario y se transmitirá a discreción del servidor MDDI.

### Paquete de corriente de vídeo sin ventanas

[0033] El paquete de corriente de vídeo sin ventanas permite que la información de ventanas quede fuera del paquete de vídeo. La información de ventanas en la versión de la técnica anterior del paquete de corriente de vídeo incluía borde izquierdo X, borde superior Y, borde derecho X, borde inferior Y, inicio X e inicio Y. La Fig. 6 representa el paquete de corriente de vídeo sin ventanas. Como se puede observar, varios de los atributos son similares al paquete de corriente de vídeo de la técnica anterior. El paquete de corriente de vídeo sin ventanas **500** incluye la longitud de paquete **502**, que contiene 2 bytes que contienen un número entero de 16 bits que especifica el número total de bytes, menos dos bytes, en el paquete de corriente de vídeo sin ventanas **500**. Un tipo de paquete **504** consiste en 2 bytes que contienen un número entero de 16 bits que identifica en dos bytes el tipo de paquete. En este ejemplo, el tipo de paquete se identifica como 22, para el **funcionamiento** del paquete de corriente de vídeo sin ventanas **500**. A continuación, se muestra el campo de ID de bClient **506**. Estos son dos bytes que contienen un número entero sin signo de 16 bits para identificar una ID de cliente. A continuación, se encuentra el descriptor de formato de datos de vídeo **508**. El descriptor de formato de datos de vídeo **508** proporciona información para el comienzo de una nueva trama y también es un número entero sin signo de 16 bits y dos bytes. A continuación se encuentran los atributos de datos de píxeles **510**, que también son un número entero sin signo de 16 bits y dos bytes que identifica los diversos atributos de los datos de píxeles. El recuento de píxeles **512** comprende un número entero sin signo de 16 bits y dos bytes que especifica el número de píxeles en el campo de datos de píxeles **516**. El parámetro CRC **514** comprende dos bytes que contienen una CRC de 16 bits de todos los bytes desde la longitud de paquete **502** hasta el recuento de píxeles **512**. Los datos de píxeles **516** contienen la información de vídeo sin procesar que se visualizará. La CRC de datos de píxeles **518** comprende dos bytes que contienen una CRC de 16 bits de solo datos de píxeles **516**. Este paquete se usa para modos operativos donde la región de visualización completa se actualiza constantemente.

### Paquete de corriente de vídeo flexible

[0034] El paquete de corriente de vídeo flexible, como se muestra en la Fig. 7, proporciona una forma de especificar qué campos están contenidos dentro de un paquete de corriente de vídeo con la inclusión de bits de campo presentes. Cada bit en este campo indica si el paquete contiene el campo correspondiente. Si un campo no está contenido en el paquete, entonces se supone que el valor debería permanecer igual que la última vez que se transmitió ese campo en un paquete de vídeo. Si ese campo no se ha transmitido previamente, entonces se puede suponer que el valor es cero.

[0035] El paquete de corriente de vídeo flexible **600** tiene los siguientes contenidos de paquete:

La longitud de paquete **602** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica el número total de bytes en el paquete sin incluir el campo de longitud de paquete. Este valor dependerá del tamaño de datos de píxeles, así como de los paquetes que se incluirán.

5 El tipo de paquete **604** comprende 2 bytes que contienen un número entero sin signo de 16 bits. En este ejemplo, un tipo de paquete de 17 identifica el paquete como un paquete de corriente de vídeo flexible **600**.

El siguiente campo es ID de bClient **606** que comprende 2 bytes que contienen un número entero sin signo de 16 bits reservado para la ID de cliente.

10 Los bits de campo presentes **608**, un valor de '1' para cada bit indica que el campo está presente en el paquete. Un valor de '0' para el bit indica que el campo no está presente. El orden de los campos es como se establece en la Fig. 7.

15 **[0036]** El descriptor de formato de datos de vídeo **610** proporciona información para el comienzo de una nueva trama y también es un número entero sin signo de 16 bits y dos bytes. A continuación están los atributos de datos de píxeles **612**, que también son un número entero sin signo de 16 bits y dos bytes que identifican los diversos atributos de los datos de píxeles. El borde izquierdo X **614** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica la coordenada X del borde izquierdo de la ventana de pantalla llena por el campo de los datos de píxeles **632**. El borde superior Y **616** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica la coordenada Y del borde superior de la ventana de pantalla llena por el campo de los datos de píxeles **632**. El borde derecho X **618** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica la coordenada X del borde derecho de la ventana de pantalla llena por el campo de los datos de píxeles **632**. El borde inferior Y **620** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica la coordenada Y del borde inferior de la ventana de pantalla llena por el campo de los datos de píxeles **632**. El inicio X **622** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica la coordenada X absoluta, donde el punto (inicio X **622** e inicio Y **624**) es el primer píxel en el campo de datos de píxeles **632**. El inicio Y **624** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica la coordenada Y absoluta, donde el punto (inicio X **622** e inicio **624**) es el primer píxel en el campo de datos de píxeles **632**.

30 **[0037]** El recuento de píxeles **628** comprende un número entero sin signo de 16 bits y dos bytes que especifica el número de píxeles en el campo de datos de píxeles **632**. El parámetro CRC **630** comprende dos bytes que contienen una CRC de 16 bits de todos los bytes desde la longitud de paquete **602** hasta el byte transmitido justo antes de este parámetro CRC **630**. Los datos de píxeles **632** contienen la información de vídeo sin procesar que se visualizará. En este ejemplo, si el bit 5 del campo de atributos de datos de píxeles **612** se establece en uno, entonces el campo de datos de píxeles **632** contiene exactamente una fila de píxeles, donde el primer píxel transmitido corresponde al píxel más a la izquierda y el último píxel transmitido corresponde al píxel más a la derecha. La CRC de datos de píxeles **634** comprende dos bytes que contienen una CRC de 16 bits de solo datos de píxeles **632**.

40 Paquete de **Encapsulación de Enlace Inverso** Mejorado que divulga modos de realización de acuerdo con la invención reivindicada

45 **[0038]** El paquete de encapsulación de enlace inverso mejorado se muestra en la Fig. 8. Este paquete combina la funcionalidad del paquete de medición de retardo de ida y vuelta para ayudar a alinear el servidor con la corriente de datos entrantes con el paquete de encapsulación de enlace inverso usado para transferir datos del cliente al servidor, como se describe en la versión anterior del sistema MDDI. Este paquete usa un patrón de sincronización para encontrar la alineación de los datos de bytes entrantes. Una vez que se encuentra el patrón de sincronización en la corriente de datos entrantes, el servidor puede muestrear de manera confiable los bits de datos de enlace inverso restantes para juntar un dato de enlace inverso y una corriente de paquetes.

50 **[0039]** El contenido del paquete para el paquete de encapsulación de enlace inverso mejorado **700** es como sigue:

La longitud de paquete **702** comprende 2 bytes que contienen un número entero sin signo de 16 bits que especifica el número total de bytes en el paquete, sin incluir el campo de longitud de paquete **702**.

55 El tipo de paquete **704** comprende 2 bytes que contienen un número entero sin signo de 16 bits. En este ejemplo, un tipo de paquete **704** de 84 identifica el paquete como un paquete de encapsulación de enlace inverso mejorado **700**.

60 El siguiente campo es ID de hClient **706** que comprende 2 bytes que contienen un número entero sin signo de 16 bits reservado para la ID de cliente.

65 **[0040]** Los indicadores de enlace inverso **708** comprenden 1 byte que contiene un número entero sin signo de 8 bits que contiene un conjunto de indicadores para solicitar información del cliente y especificar el tipo de interfaz de enlace inverso. En este ejemplo, si un bit se establece en uno, entonces el servidor solicita la información especificada al cliente. Si el bit es cero, entonces el servidor no necesita la información del cliente. Por ejemplo, el Bit 0 podría indicar

que el servidor necesita un paquete de capacidad de cliente. Se enviará por el cliente al servidor en el campo de paquetes de datos inversos **724**. El bit 1 podría indicar que el servidor necesita la solicitud de cliente y el paquete de estado. Se enviará por el cliente al servidor en el campo de paquetes de datos inversos **724**. El bit 2 podría indicar que el servidor necesita que el cliente transmita un byte de sincronización antes de transmitir el primer byte de datos de un paquete de enlace inverso **724**. El bit 3 podría indicar que el servidor requiere que el cliente transmita la cantidad de bytes inversos para esperar antes de comenzar la transmisión inversa de paquetes. Esto es para permitir paquetes de enlace inverso de tamaño dinámico que cumplan exactamente los requisitos de los clientes actualmente pendientes de actualización de datos de enlace inverso.

**[0041]** El divisor de velocidad inversa **710** comprende 1 byte que contiene un número entero sin signo de 8 bits que especifica el número de ciclos de MDDI\_Stb que se producen por reloj de datos de enlace inverso. El reloj de datos de enlace inverso es igual al reloj de datos de enlace directo dividido por la cantidad: dos veces el divisor de velocidad inversa **710**. La velocidad de transferencia de datos de enlace inverso está relacionada con el reloj de datos de enlace inverso y el tipo de interfaz en el enlace inverso en el siguiente ejemplo:

la Interfaz Tipo 1 que indica la velocidad de transferencia de datos inversa es igual al reloj de datos de enlace inverso;

la Interfaz Tipo 2 que indica que la velocidad de transferencia de datos inversa es igual a dos veces el reloj de datos de enlace inverso;

la Interfaz Tipo 3 que indica que la velocidad de transferencia de datos inversa es igual a cuatro veces el reloj de datos de enlace inverso; y

la Interfaz Tipo 4 que indica que la velocidad de transferencia de datos inversa es igual a ocho veces el reloj de datos de enlace inverso.

**[0042]** La longitud de giro 1 **712** comprende 1 byte que contiene un número entero sin signo de 8 bits que especifica el número total de bytes que se asignan para el giro 1. La longitud recomendada del giro 1 es la cantidad de bytes necesarios para que los drivers de MDDI\_Data en el servidor desactiven sus salidas. Esto se basa en el tiempo de desactivación de salida, en la velocidad de transferencia de datos de enlace directo y en la selección del tipo de interfaz de enlace directo que se esté usando. La longitud de giro 2 **714** comprende 1 byte que contiene un número entero sin signo de 8 bits que especifica el número total de bytes que se asignan para el giro 2. La longitud recomendada del giro 2 es la cantidad de bytes necesarios para el retardo de ida y vuelta más el tiempo requerido para que el servidor active sus drivers de MDDI\_Data. La longitud de giro 2 también puede ser cualquier valor mayor que el valor mínimo requerido calculado para permitir tiempo suficiente para procesar paquetes de enlace inverso en el servidor. Los bytes inversos máximos **716** comprenden 2 bytes que indican cuántos bytes inversos se pueden transmitir desde el cliente de vuelta al servidor. Esto no incluye ningún byte requerido tal como el patrón de sincronización, o el cliente transmite campos de longitud de bytes que pueden preceder a los datos de enlace inverso cuando lo soliciten los bits en el campo de indicadores de enlace inverso **708**. Cuando se establece el Bit 3, el cliente puede solicitar enviar datos que sean inferiores al valor en el campo de bytes inversos máximos **716**. Cuando el cliente transmite un número que es menor que el campo de bytes inversos máximos **716**, la MDDI acortará el período anticipado del campo de datos inversos y la sincronización **724** para maximizar la solicitud de los clientes. El parámetro CRC **718** comprende 2 bytes que contienen una CRC de 16 bits de todos los bytes desde la longitud de paquete **702** hasta la longitud de giro **712** y el campo de bytes inversos máximos **716**. Si esta CRC no se verifica, entonces el paquete completo debería descartarse. Todo cero 1 **720** comprende 8 bytes, cada uno de los cuales contiene un número entero sin signo de 8 bits igual a cero. Este campo garantiza que todas las señales MDDI\_Data estén en un nivel lógico cero durante un tiempo suficiente para permitir que el cliente comience a recuperar el reloj usando solo MDDI\_Stb antes de desactivar los drivers de línea del servidor durante el campo de giro 1 **722**. El giro 1 **722** comprende un primer período de giro. El número de bytes especificado por el parámetro de longitud de giro 1 **712** se asigna para permitir que los drivers de línea de MDDI\_Data en el cliente se activen antes de que los drivers de línea en el servidor se desactiven. El cliente activará sus drivers de línea de MDDI\_Data durante el bit 0 del giro 1 **722** y el servidor desactivará sus salidas y se desactivará completamente antes del último bit del giro 1 **722**. La señal MDDI\_Stb se comporta como si MDDI\_Data0 estuviera en un nivel lógico cero durante todo el período de giro 1 **722**.

**[0043]** La sincronización inversa, el recuento de bytes y los paquetes de datos **724** se muestran como un solo campo en la Fig. 8. El primer byte en este campo debería ser el patrón de sincronización (0x053F) si el Bit 2 lo solicita para que se establezca en uno lógico en el campo de indicadores de enlace inverso **708**. Si se establece el Bit 3, el siguiente campo de enlace inverso transmitido debería ser el número de bytes que el cliente transmitirá en el enlace inverso. Si no se solicitan estos datos, el cliente puede transmitir datos de enlace inverso hasta el número de bytes especificado en el campo de bytes inversos máximos **716**. Este campo debería seguirse por el campo de longitud de paquete del primer paquete de enlace inverso. Se puede transmitir más de un paquete en el período de datos inverso si hay suficiente espacio. El cliente puede enviar paquetes de relleno o llevar las líneas MDDI\_Data a un nivel lógico cero cuando no tenga datos que enviar al servidor. Si las líneas MDDI\_Data se llevan a cero, el servidor interpretará esto como un paquete con una longitud cero (no una longitud válida) y el servidor no aceptará ningún paquete adicional del cliente durante el paquete de encapsulación de enlace inverso mejorado actual **700**. El giro 2 **726** comprende el



segundo período de giro. El número de bytes se especifica mediante el parámetro de longitud de giro 2 714. El servidor activará sus drivers de línea de MDDI\_Data y se activará completamente antes del último bit del giro 2 726 y el cliente desactivará sus salidas y se desactivará completamente antes del último bit del giro 2 726. El propósito del giro 2 726 es permitir que la cantidad restante de datos del campo de paquetes de datos inversos 724 se transmita desde el cliente. Debido a las variaciones en los diferentes sistemas y a la cantidad del margen de seguridad asignada, es posible que ni el servidor ni el cliente lleven las señales de MDDI\_Data a un nivel de cero lógico durante algunas partes del campo de giro 2 726 como se ve por los receptores de línea en el servidor. La señal MDDI\_Stb se comporta como si MDDI\_Data0 estuviera en un nivel de cero lógico durante todo el período de giro 2 726. Todo cero 2 728 comprende 8 bytes que contienen cada uno un número entero sin signo de 8 bits igual a cero. Este campo garantiza que todas las señales MDDI\_Data estén en un nivel lógico cero durante un tiempo suficiente para permitir que el cliente comience a recuperar el reloj usando MDDI\_Data0 y MDDI\_Stb después de activar los drivers de línea del servidor después del campo de giro 2 726.

### Congelación de enlace MDDI

[0044] El servidor MDDI puede encontrar momentos donde necesite detener el enlace de datos MDDI o pausar el funcionamiento del enlace. La Fig. 9 muestra el aspecto de congelación de enlace de la invención reivindicada actualmente. La Fig. 9 muestra datos de MDDI 900, la luz estroboscópica (STB) 902 y el reloj recuperado 904. Este aspecto permite que los datos de MDDI 900 se detengan durante un corto período de tiempo 906 y congelen el estado actual del cliente de MDDI. Como se muestra, el reloj recuperado 904 en el cliente se deriva de la corriente de datos de MDDI entrantes 900 y de la luz estroboscópica de MDDI 902, y, como resultado, la detención del enlace MDDI detendrá más ciclos de reloj 906 dentro del cliente. El servidor debe mantener los niveles diferenciales correspondientes al último bit de datos transmitido al entrar en este modo. No es necesario que el servidor MDDI transmita ningún paquete especial que indique que está entrando en este modo, y puede congelar el enlace en el medio de un paquete saliente si es necesario. Esto se puede usar para evitar desbordamientos dentro de un diseño de servidor MDDI si otras fuentes de datos no se pueden mantener al día brevemente con la corriente de datos saliente de MDDI.

[0045] Debido al consumo de energía adicional de mantener los datos de MDDI 900 y las señales de la luz estroboscópica 902 accionadas, este estado solo debería usarse en situaciones de corta duración. Cuando no se transmita contenido significativo durante un período de tiempo más largo, el modo de hibernación debería usarse para mantener el consumo de energía al mínimo.

[0046] En muchos clientes habrá un retardo en la canalización de procesamiento para decodificar los paquetes entrantes. La detención de la MDDI justo después de que un paquete se transmite desde el servidor no cumple con los requisitos del cliente, y el cliente debería tener la oportunidad de procesar los datos contenidos en el último paquete.

[0047] Las señales que salgan del cliente de MDDI también se congelarán en un estado particular debido a la falta de un reloj. Cualquier diseño que haga uso del cliente de MDDI debe ser consciente de la posibilidad de esta condición.

[0048] Esta memoria descriptiva divulga uno o más aspectos que incorporan las características de la invención reivindicada. Los aspectos divulgados simplemente ejemplifican la invención reivindicada. El alcance de la invención no está limitado a los aspectos divulgados. La invención se define por las reivindicaciones adjuntas a la misma.

[0049] Los expertos en la técnica entenderían que la información y las señales se pueden representar usando cualquiera de una variedad de tecnologías y técnicas diferentes. Por ejemplo, los datos, las instrucciones, los comandos, la información, las señales, los bits, los símbolos y los chips que se puedan haber mencionado a lo largo de la descripción anterior se pueden representar mediante tensiones, corrientes, ondas electromagnéticas, campos o partículas magnéticos, campos o partículas ópticos o cualquier combinación de los mismos.

[0050] Los expertos en la técnica apreciarían además que los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos en relación con los modos de realización divulgados en el presente documento se pueden implementar como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, anteriormente se han descrito diversos componentes, bloques, módulos, circuitos y etapas ilustrativas, en general, en lo que respecta a su funcionalidad. Que dicha funcionalidad se implemente como hardware o software depende de la solicitud particular y de las restricciones de diseño impuestas en el sistema general. Los expertos en la técnica pueden implementar la funcionalidad descrita de formas distintas para cada solicitud particular, pero no debería interpretarse que dichas decisiones de implementación suponen apartarse del alcance de la presente invención.

[0051] Los diversos bloques lógicos, módulos y circuitos ilustrativos descritos en relación con los modos de realización divulgados en el presente documento se pueden implementar o realizar con un procesador de uso general, con un procesador de señales digitales (DSP), con un circuito integrado específico de la aplicación (ASIC), con una matriz de puertas programables por campo (FPGA) o con otro dispositivo de lógica programable, lógica de transistor o de puertas discretas, componentes de hardware discretos, o con cualquier combinación de los mismos diseñada

para realizar las funciones descritas en el presente documento. Un procesador de uso general puede ser un microprocesador, pero, como alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también se puede implementar como una combinación de dispositivos informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra configuración de este tipo.

**[0052]** Las etapas de un procedimiento o algoritmo descrito en relación con los modos de realización divulgados en el presente documento se pueden realizar directamente en hardware, en un módulo de software ejecutado por un procesador o en una combinación de los dos. Un módulo de software puede residir en una Memoria de Acceso Aleatorio (RAM), una memoria flash, una Memoria de Solo Lectura (ROM), una memoria ROM Programable Eléctricamente (EPROM), una memoria ROM Programable y Borrable Eléctricamente (EEPROM), unos registros, un disco duro, un disco extraíble, un CD-ROM o cualquier otra forma de medio de almacenamiento conocido en la técnica. Un medio de almacenamiento ejemplar está acoplado al procesador de modo que el procesador pueda leer información de, y escribir información en, el medio de almacenamiento. Como alternativa, el medio de almacenamiento puede estar integrado en el procesador. El procesador y el medio de almacenamiento pueden residir en un ASIC. El ASIC puede residir en un terminal de usuario. Como alternativa, el procesador y el medio de almacenamiento de troqueles pueden residir como componentes discretos en un terminal de usuario.

**[0053]** La descripción previa de los modos de realización divulgados se proporciona para permitir que cualquier experto en la técnica realice o use la presente invención. Diversas modificaciones de estos modos de realización resultarán fácilmente evidentes a los expertos en la técnica, y los principios genéricos definidos en el presente documento se pueden aplicar a otros modos de realización sin apartarse del alcance de la presente invención. Por tanto, la presente invención no está prevista para limitarse a los modos de realización mostrados en el presente documento, sino que se le concede el alcance más amplio consecuente con los principios y las características novedosas divulgados en el presente documento.

**REIVINDICACIONES**

- 5 1. Un procedimiento para proporcionar un paquete de encapsulación de enlace inverso mejorado (700) a través de un enlace de transmisión que acopla un cliente (106) y un servidor (122) dentro de un dispositivo electrónico (100), que comprende las etapas de:
- 10 establecer un primer indicador (708) dentro del paquete de encapsulación de enlace inverso mejorado (700) y transmitir el indicador desde el servidor (122) al cliente (106) solicitando que se transmita un patrón de sincronización (724) desde el cliente (106) al servidor (122);
- 15 establecer un segundo indicador (708) dentro del paquete de encapsulación de enlace inverso mejorado (700) y transmitir el indicador desde el servidor (122) al cliente (106) solicitando un recuento de bytes de un número de bytes que se vayan a transmitir desde el cliente (106) al servidor (122);
- 20 proporcionar un período fijo (722) de tiempo dentro del paquete de encapsulación de enlace inverso mejorado (700) para que el servidor (122) desactive sus salidas de driver en el enlace de transmisión;
- proporcionar el patrón de sincronización (724) transmitido por el cliente (106) en una porción de datos inversa (724) del paquete de encapsulación de enlace inverso mejorado (700);
- 25 proporcionar el recuento de bytes dentro de la porción de datos inversa (724) del paquete de encapsulación de enlace inverso mejorado (700) que indica un número total de bytes que se espera transmitir desde el cliente (106) al servidor (122); y
- medir un retardo de ida y vuelta entre el servidor (122) y el cliente (106) usando el patrón de sincronización, el procedimiento **caracterizado por**:
- 30 proporcionar un giro (726) para permitir que la cantidad restante de datos de la porción de datos inversa (724) se transmita desde el cliente;
- 35 alinear el servidor (122) con el paquete de encapsulación de enlace inverso mejorado (700) en base al retardo de ida y vuelta y al patrón de sincronización (724); y
- enviar un valor de recuento de bytes desde el cliente (106) al servidor (122) que indique que el cliente (106) enviará menos bytes que los actualmente asignados en una porción de datos inversa (724) del paquete de encapsulación de enlace inverso mejorado (700), en el que el valor de recuento de bytes no es mayor que el valor indicado dentro de un campo de bytes máximos (716) del paquete de encapsulación de enlace inverso mejorado.
- 40 2. El procedimiento de la reivindicación 1 que comprende además la etapa de transmitir un paquete de datos desde el cliente al servidor.
- 45 3. El procedimiento de la reivindicación 1 en el que la etapa de proporcionar el patrón de sincronización comprende transmitir una secuencia específica de bytes desde el cliente para permitir que el servidor se sincronice con los datos inversos antes de que los paquetes reales se transmitan en un enlace inverso.
- 50 4. El procedimiento de la reivindicación 1, que comprende además la etapa de uso del retardo de ida y vuelta previo medido para un siguiente paquete de encapsulación de enlace inverso mejorado transmitido en lugar de establecer el primer indicador y proporcionar nuevamente el patrón de sincronización.
- 55 5. El procedimiento de la reivindicación 1 que comprende la etapa del servidor y del cliente de avanzar al siguiente campo en el paquete de encapsulación de enlace inverso mejorado después de que el cliente haya transmitido el número de bytes indicado en el valor de bytes.
- 60 6. El procedimiento de la reivindicación 1 que comprende además la etapa de permitir que se transmita hasta un número máximo de bytes, siendo el número máximo igual a un valor en un campo de bytes máximos, en lugar de establecer el segundo indicador para que se transmita un recuento de bytes en el paquete de encapsulación de enlace inverso mejorado.
- 65 7. Un sistema para proporcionar un paquete de encapsulación de enlace inverso mejorado (700) a través de un enlace de transmisión que acopla un cliente (106) y un servidor (122) dentro de un dispositivo electrónico (100), que comprende:
- medios para establecer un primer indicador (708) dentro del paquete de encapsulación de enlace inverso mejorado (700) y transmitir el indicador desde el servidor (122) al cliente (106) solicitando que se transmita un patrón de sincronización (724) desde el cliente (106) al servidor (122);

medios para establecer un segundo indicador (708) dentro del paquete de encapsulación de enlace inverso mejorado (700) y transmitir el indicador desde el servidor (122) al cliente (106) solicitando un recuento de bytes de un número de bytes que se vayan a transmitir desde el cliente (106) al servidor (122);

5 medios para proporcionar un período fijo (722) de tiempo dentro del paquete de encapsulación de enlace inverso mejorado (700) para que el servidor (122) desactive sus salidas de driver en el enlace de transmisión;

10 medios para proporcionar el patrón de sincronización (724) transmitido por el cliente (106) en una porción de datos inversa (724) del paquete de encapsulación de enlace inverso mejorado (700);

15 medios para proporcionar el recuento de bytes dentro de la porción de datos inversa (724) del paquete de encapsulación de enlace inverso mejorado (700) que indica un número total de bytes que se espera transmitir desde el cliente (106) al servidor (122); y

medios para medir un retardo de ida y vuelta entre el servidor (122) y el cliente (106) usando el patrón de sincronización, el sistema **caracterizado por:**

20 medios para proporcionar un giro (726) para permitir que la cantidad restante de datos de la porción de datos inversa (724) se transmita desde el cliente;

medios para alinear el servidor (122) con el paquete de encapsulación de enlace inverso mejorado (700), en base al retardo de ida y vuelta y al patrón de sincronización (724); y

25 medios para enviar un valor de recuento de bytes desde el cliente (106) al servidor (122) que indica que el cliente (106) enviará menos bytes que los asignados actualmente en una porción de datos inversa del paquete de encapsulación de enlace inverso mejorado (700), en el que el valor de recuento de bytes no es mayor que el valor indicado dentro de un campo de bytes máximos (716) del paquete de encapsulación de enlace inverso mejorado.

30 **8.** El sistema de la reivindicación 7 que comprende además medios para transmitir un paquete de datos desde el cliente al servidor.

35 **9.** El sistema de la reivindicación 7 en el que los medios para proporcionar un patrón de sincronización comprenden medios para transmitir una secuencia específica de bytes desde el cliente para permitir que el servidor se sincronice con los datos inversos antes de que los paquetes reales se transmitan en un enlace inverso.

40 **10.** El sistema de la reivindicación 7, que comprende además medios para usar el retardo de ida y vuelta previo medido para el siguiente paquete de encapsulación de enlace inverso mejorado transmitido en lugar de establecer el primer indicador y proporcionar nuevamente el patrón de sincronización.

45 **11.** El sistema de la reivindicación 7 que comprende además medios para que el servidor y el cliente avancen al siguiente campo en el paquete de encapsulación de enlace inverso mejorado después de que el cliente haya transmitido el número de bytes indicado en el valor de bytes.

50 **12.** El sistema de la reivindicación 7 que comprende además medios para permitir que se transmita hasta un número máximo de bytes, siendo el número máximo igual a un valor en un campo de bytes máximos, en lugar de establecer el segundo indicador para que se transmita un recuento de bytes el paquete de encapsulación de enlace inverso mejorado.

**13.** Un medio legible por ordenador que comprende instrucciones que, cuando se ejecutan en un procesador, causan que el procesador lleve a cabo las etapas del procedimiento de cualquiera de las reivindicaciones 1 a 6.

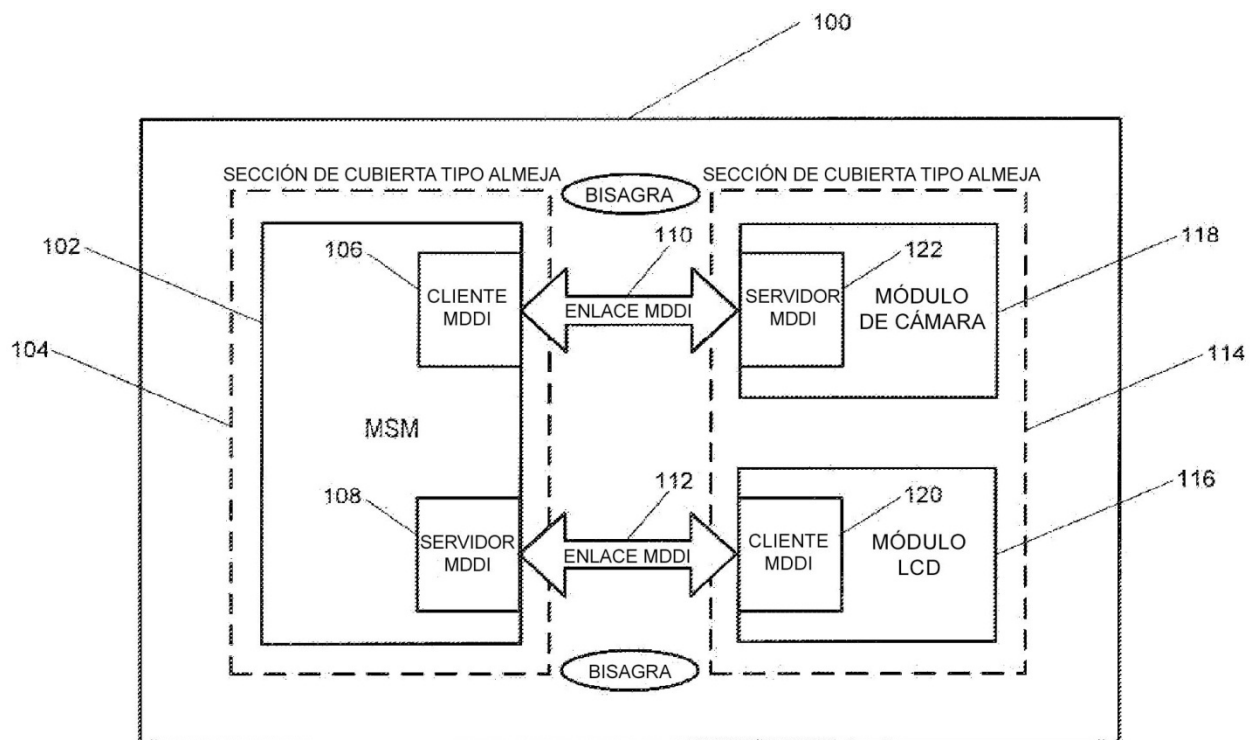


FIG. 1

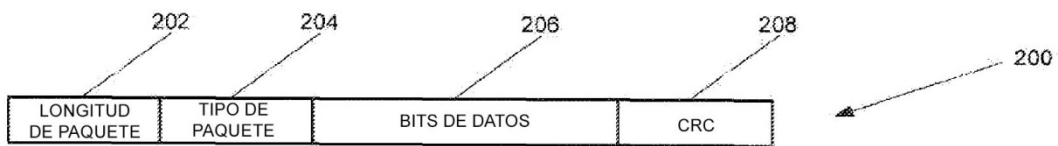


FIG. 2A

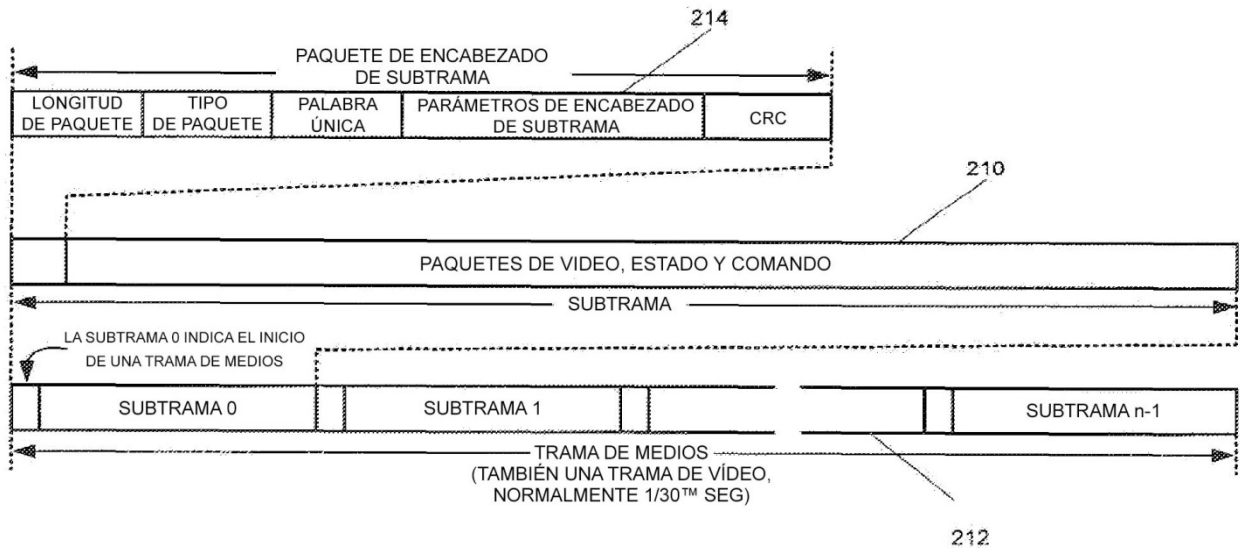


FIG. 2B

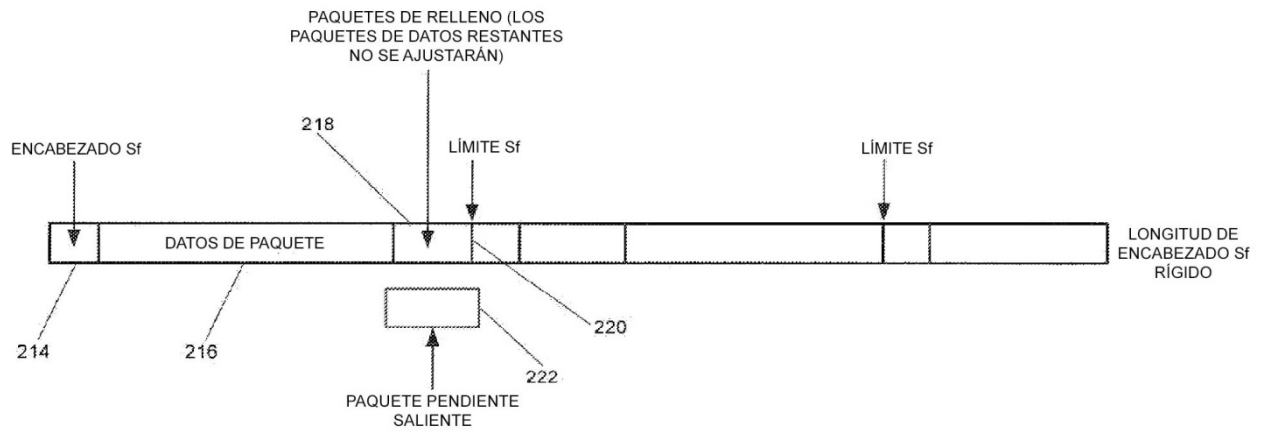


FIG. 3

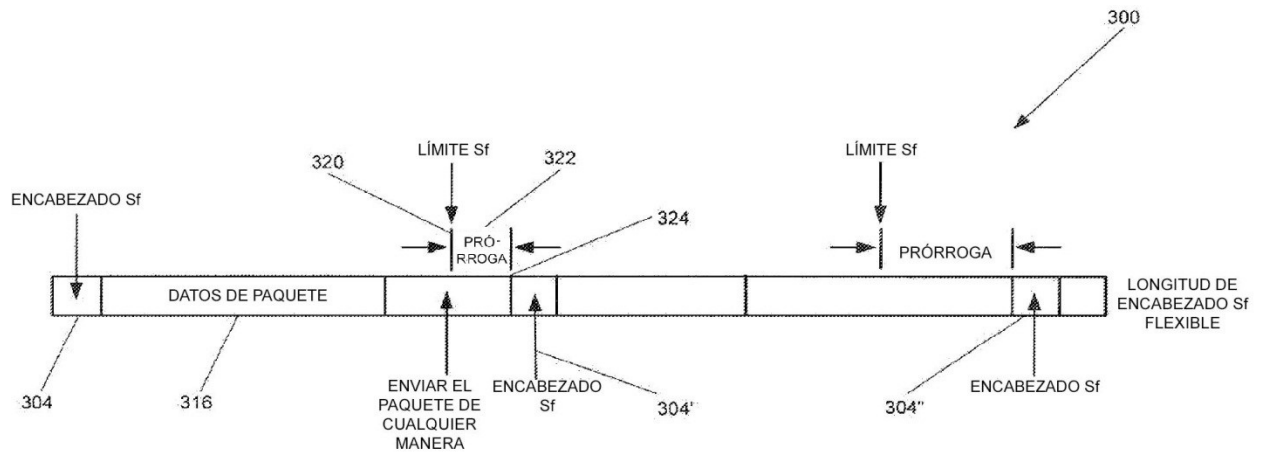


FIG. 4



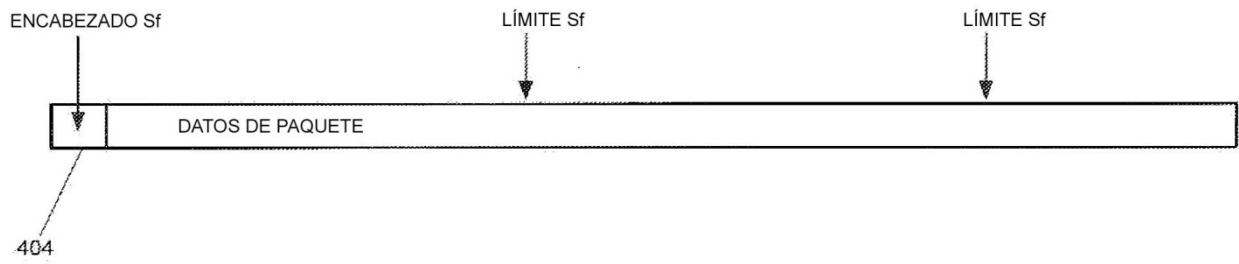


FIG. 5

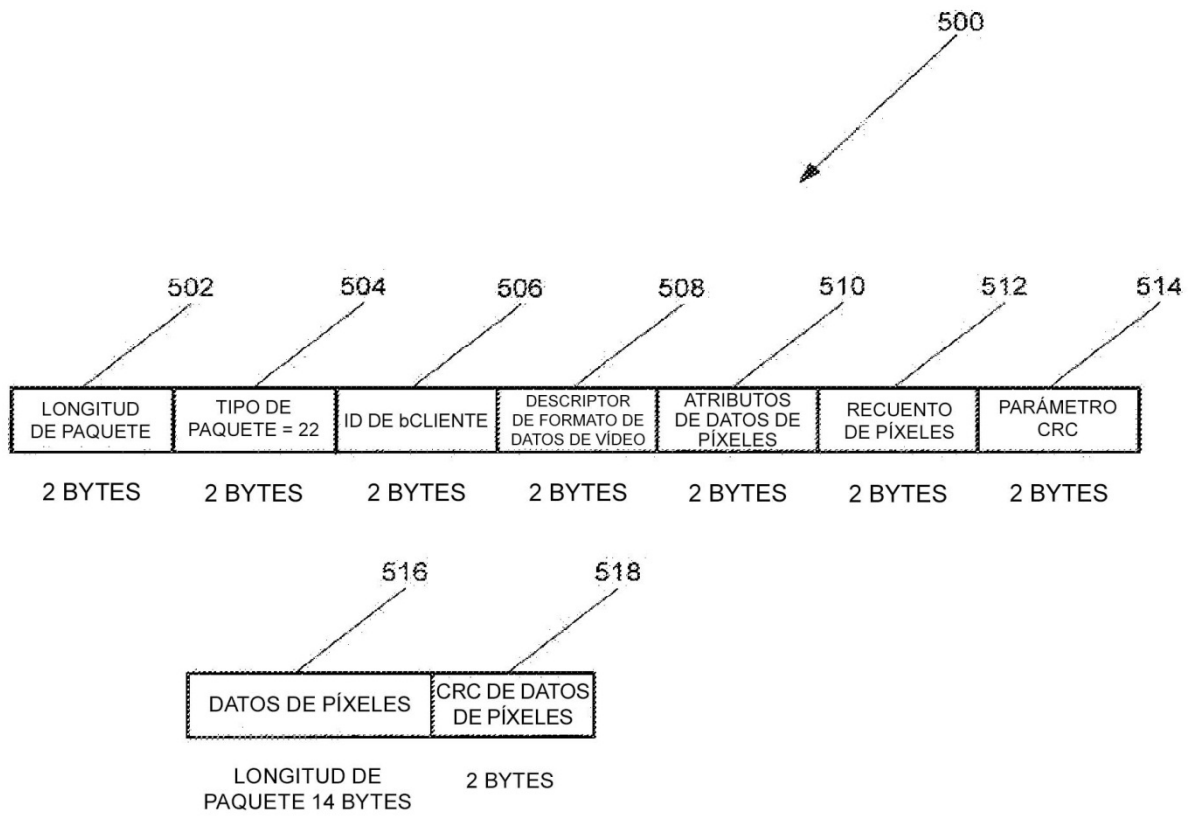


FIG. 6

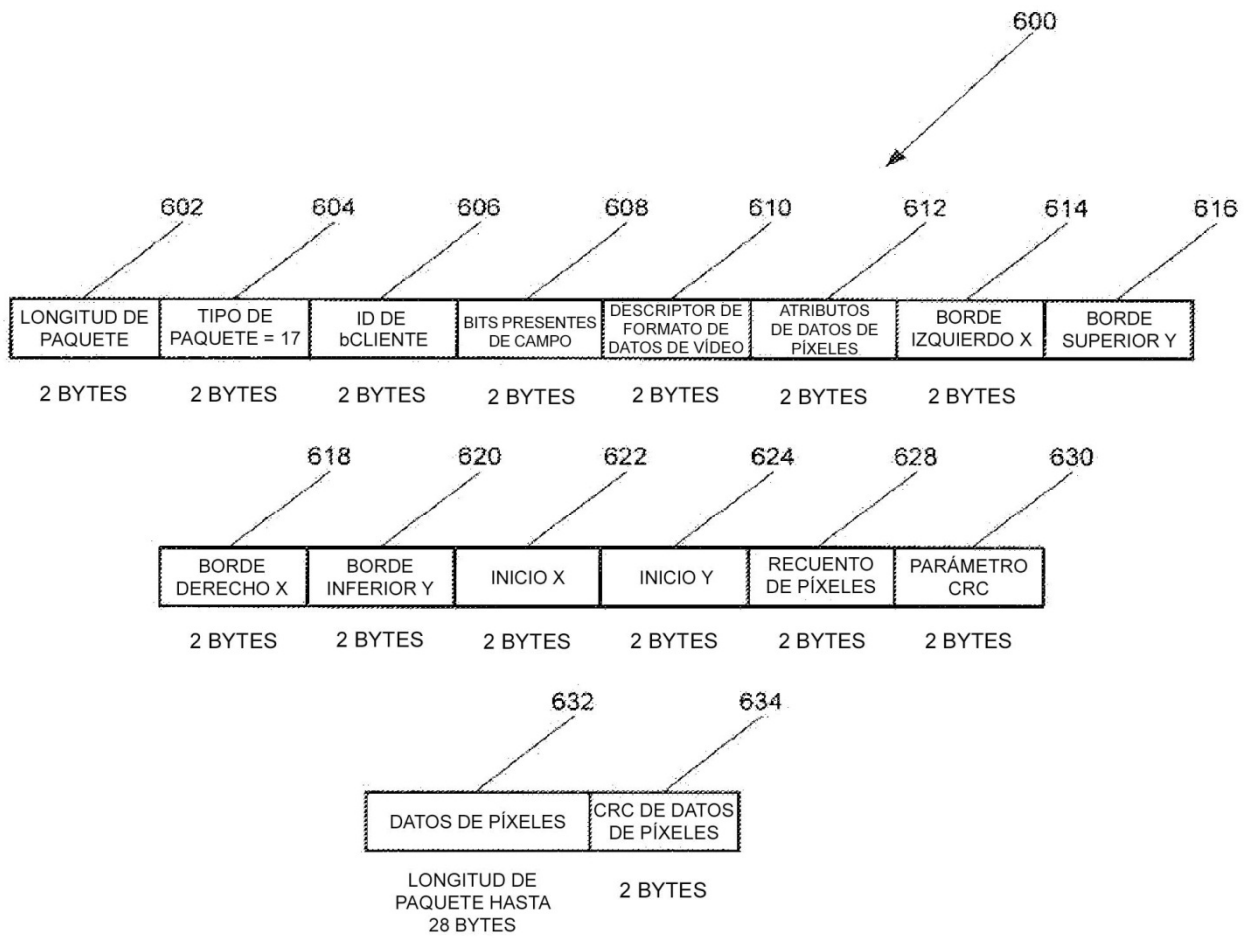


FIG. 7

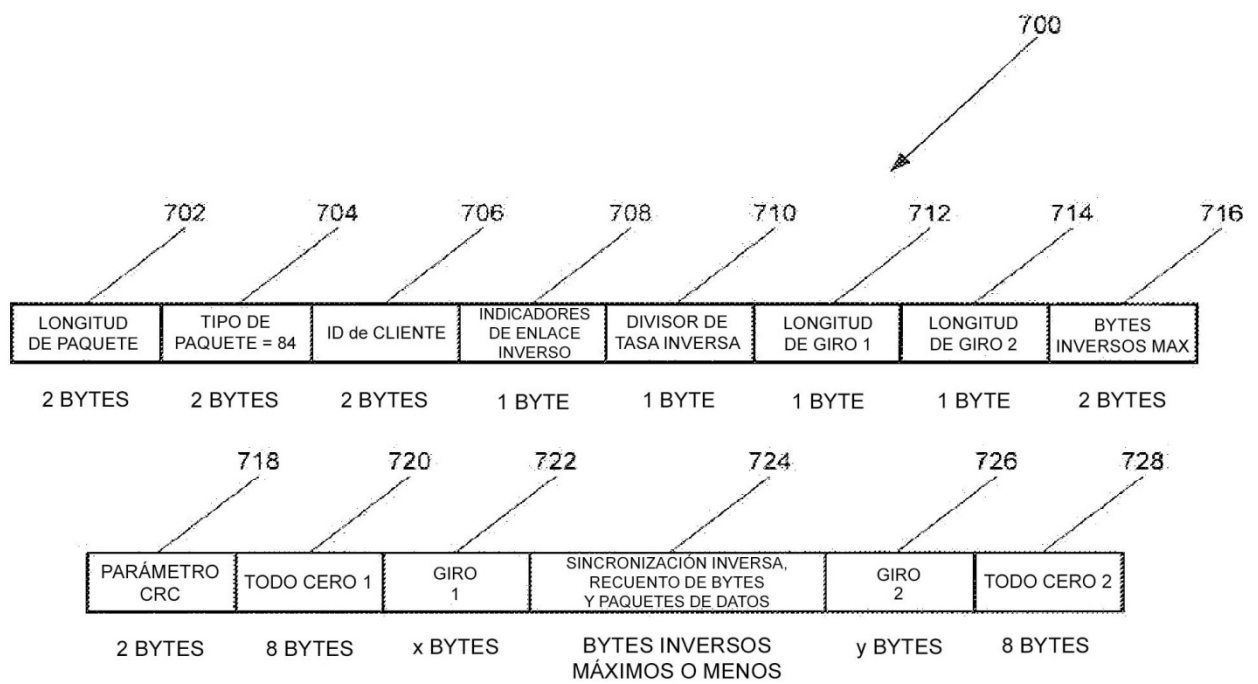


FIG. 8

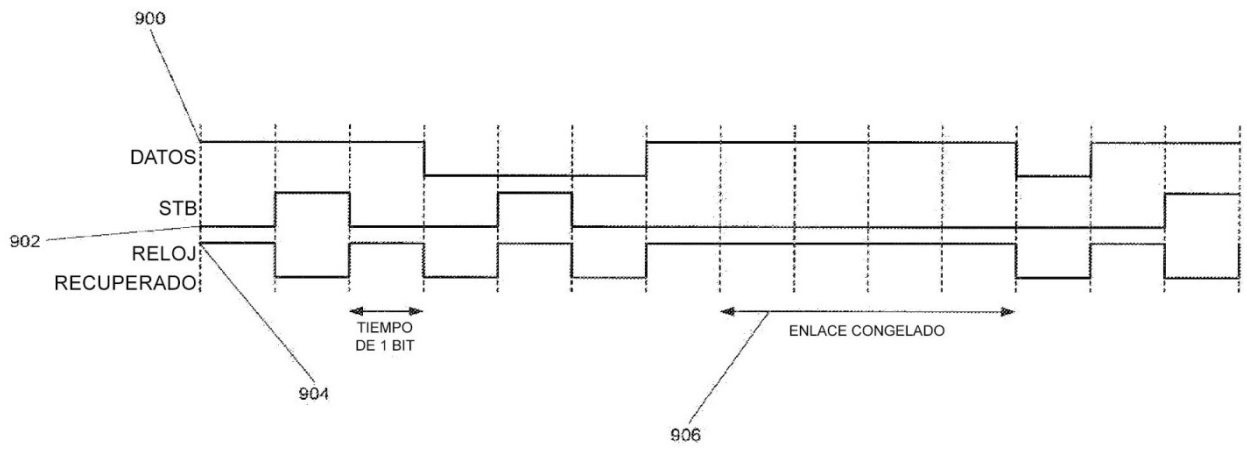


FIG. 9