



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 363 059**

51 Int. Cl.:
G09G 5/00 (2006.01)
G06F 3/14 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **08857033 .8**
96 Fecha de presentación : **08.05.2008**
97 Número de publicación de la solicitud: **2147532**
97 Fecha de publicación de la solicitud: **27.01.2010**

54 Título: **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 de la mención BOPI:
19.07.2011

45 Fecha de la publicación del folleto de la patente:
19.07.2011

73 Titular/es: **QUALCOMM INCORPORATED**
Attn: International IP Administration
5775 Morehouse Drive
San Diego, California 92121, US

72 Inventor/es: **Steele, Brian;**
Wiley, George, Alan y
Shekhar, Shashank

74 Agente: **Carpintero López, Mario**

ES 2 363 059 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

Estructura de paquetes para una interfaz digital de pantalla móvil

Antecedentes

Campo

- 5 La presente invención se refiere en general a enlaces de comunicación, y más particularmente a un procedimiento, un sistema, y un producto de programa de ordenador para proporcionar una estructura de paquetes mejorada para enlaces de interfaz digital de pantalla móvil (MDDI).

Antecedentes

- 10 En el campo de las tecnologías de interconexión, la demanda de cada vez mayores velocidades de datos, especialmente en lo relacionado con las presentaciones de vídeo, sigue creciendo.

La interfaz digital de pantalla móvil de (MDDI) es un mecanismo de transferencia rentable y de bajo consumo que permite la transferencia de datos a muy alta velocidad a través de un enlace de comunicación de corto alcance entre un servidor y un cliente. MDDI requiere un mínimo de sólo cuatro cables más potencia para la transferencia bidireccional de datos que proporciona un ancho de banda máximo de hasta 3,2 Gbits por segundo.

- 15 En una aplicación, MDDI aumenta la fiabilidad y reduce el consumo de energía en teléfonos de carcasa plegable al reducir significativamente el número de cables que se extienden a través de una articulación del terminal para interconectar el controlador de banda de base digital con una pantalla LCD y/o una cámara. Esta reducción de los cables también permite a los fabricantes de teléfonos disminuir los costes de desarrollo mediante la simplificación los diseños de teléfonos de carcasa plegable o deslizantes. Además, la señalización diferencial empleada con MDDI reduce la interferencia electromagnética que puede producirse en conexiones en paralelo tradicionales.

- 20 Hay algunas mejoras necesarias en los actuales sistemas MDDI. En la actualidad, los sub-marcos contienen una longitud de sub-marco fija e intervalos de tiempo. Esto limita el sistema a un número fijo de bits en cada sub-marco y opera a una velocidad fija. Esto evita que los paquetes se separen de un sub-marco al siguiente. Los paquetes más grandes deben retrasarse hasta que el siguiente sub-marco se transmita, desperdiciando ancho de banda y aumentando la latencia. Es necesario un sistema con una longitud de sub-marco flexible para transmitir más eficazmente estos paquetes de gran tamaño. Otra mejora en un sub-marco de longitud fija es la capacidad de utilizar una longitud ilimitada del sub-marco, cuando el enlace sale de la hibernación. Esto también ahorra ancho de banda debido a que el paquete marco de encabezado del sub-marco se transmite sólo una vez para permitir que el cliente se sincronice en el inicio.

- 25 Otra mejora necesaria para los sistemas existentes es un procedimiento para evitar la retransmisión repetitiva de ciertos datos de paquetes de vídeo cuando no se modifican algunos de los parámetros. Otra vez, esto ahorra ancho de banda. Esto se logra proporcionando un paquete de secuencia de vídeo sin ventanas. Además, se necesita un sistema para proporcionar una manera para especificar qué campos están contenidos dentro de un paquete de secuencia de vídeo cuando algunos valores no han cambiado. Sería desperdiciar ancho de banda retransmitir repetidamente los campos que contienen valores idénticos a los enviados en los paquetes anteriores. Esto se prevé en un campo de contenidos del paquete del paquete de secuencia de vídeo flexible.

- 30 Los sistemas existentes primero transmiten un paquete de medición de retraso de ida y vuelta, y a continuación transmiten un paquete encapsulado inverso separado para que el servidor reciba los datos desde el cliente. La invención actualmente reivindicada es una mejora significativa sobre los actuales sistemas y combina la funcionalidad de los dos paquetes en un solo paquete de encapsulación inversa mejorada.

Sumario

- 35 Un aspecto único introducido es un paquete de datos de vídeo sin ventanas. Este aspecto permite un tamaño de la ventana definida una primera vez, solamente para volver a utilizarse sin tener que redefinir la ventana. Esto se logra mediante la eliminación de coordenadas de campo X-izquierda, X-Derecha, Y-Arriba, Y-Abajo, X-Inicio e Y-Inicio del paquete de datos de vídeo. Un bit en el campo existente representa la sincronización vertical y se identifica con la primera línea de una pantalla de datos.

Otros aspectos, características y ventajas de la presente invención reivindicada, así como la estructura y el funcionamiento de los diversos aspectos de la presente invención reivindicada, se describen en detalle a continuación con referencia a los dibujos adjuntos.

Breve descripción de los dibujos

La figura 1 es un diagrama de bloques que ilustra un ejemplo de entorno usando una interfaz MDDI;

La figura 2A muestra una estructura de paquetes MDDI típica;

La figura 2B muestra una estructura típica de enlace delantero;

La figura 3 muestra el sub-marco del estado de la técnica con una longitud fija;

La figura 4 representa el sub-marco de longitud flexible;

La figura 5 representa el sub-marco de longitud ilimitada;

5 La figura 6 muestra el paquete de secuencia de vídeo sin ventanas, de acuerdo con la invención reivindicada;

La figura 7 muestra el paquete de secuencia de vídeo flexible;

La figura 8 muestra el paquete encapsulado de enlace inverso mejorado; y

La figura 9 representa la congelación del enlace.

Descripción detallada

10 La palabra “ejemplar” se utiliza aquí en el sentido de “servir como ejemplo, caso o ilustración”. Cualquier aspecto aquí descrito como “ejemplo” no debe interpretarse necesariamente como preferido o ventajoso respecto a otros aspectos.

15 Los aspectos descritos, y las referencias en la memoria a “un aspecto”, “un aspecto de ejemplo”, etc., indican que los aspectos descritos pueden incluir una característica o estructura particular, pero cada aspecto puede no incluir necesariamente la estructura o característica particular. Además, estas frases no se refieren necesariamente al mismo aspecto. Además, cuando una característica o estructura particular se describe en relación con un aspecto, se considera que está dentro del conocimiento de un experto en la materia que afecta a estas características o estructura en relación con otros aspectos, estén o no explícitamente descritos.

20 La interfaz digital de pantalla móvil (MDDI) es un mecanismo de transferencia rentable, de bajo consumo de energía que permite una transferencia de datos en serie a muy alta velocidad en un enlace de comunicación de corto alcance entre un servidor y un cliente para apreciar plenamente las nuevas características aquí introducidas, se proporciona una breve discusión sobre el sistema MDDI.

25 A continuación, ejemplos de MDDI se presentarán respecto a un módulo de cámara contenida en una carcasa plegable superior de un teléfono móvil. Sin embargo, sería evidente para las personas expertas en la materia relevante que cualquier módulo que tenga características funcionalmente equivalentes al módulo de la cámara puede ser fácilmente sustituido y utilizado en los aspectos de esta invención.

30 Además, de acuerdo a los aspectos de la invención, un servidor MDDI puede comprender uno de varios tipos de dispositivos que pueden beneficiarse del uso de la presente invención reivindicada. Por ejemplo, el servidor podría ser un ordenador portátil en forma de un dispositivo de ordenador portátil, de mano, o móvil similar. También podría ser un Asistente de Datos Personal (PDA), un dispositivo de paginación, o uno de los muchos teléfonos o módems inalámbricos.

35 Alternativamente, el servidor podría ser un dispositivo de entretenimiento portátil o de presentación, tal como un DVD portátil o un reproductor de CD, o un dispositivo de juegos. Además, el servidor puede residir como un dispositivo servidor o elemento de control en una variedad de otros productos comerciales ampliamente utilizados o planeados para los que se desea un enlace de comunicación de alta velocidad con un cliente. Por ejemplo, un servidor podría ser usado 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 grande de alta resolución para presentaciones. Un aparato, tal como un refrigerador, que incorpora un inventario a bordo o un sistema informático y/o conexiones Bluetooth a otros dispositivos del hogar, puede tener capacidades de visualización mejoradas cuando funciona en un modo de conexión a Internet o Bluetooth, o que tener necesidades de conexión reducidas para las pantallas en la puerta (un cliente) y teclados o escáneres (cliente), mientras que el equipo electrónico o los sistemas de control (servidor) residen en cualquier sitio del gabinete. En general, los expertos en la materia podrán apreciar la gran variedad de dispositivos electrónicos de módem y aparatos que pueden beneficiarse del uso de esta interfaz, así como la capacidad para adaptar los dispositivos más antiguos con una mayor velocidad de transporte de datos de información utilizando un número limitado de conductores disponibles en conectores o cables existentes o añadidos nuevos. Al mismo tiempo, un cliente MDDI puede comprender una gran variedad de dispositivos útiles para presentar la información a un usuario final, o presentar información desde un usuario al servidor. Por ejemplo, una micropantalla incorporada en anteojos o gafas, un aparato de proyección integrado en una gorra o un casco, una 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 la presentación de sonido o música de alta calidad. Otros dispositivos de presentación incluyen proyectores o dispositivos de proyección usados para presentar información para reuniones, o para películas e imágenes de televisión. Otro ejemplo sería el uso de pantallas táctiles o dispositivos sensibles, dispositivos de entrada de reconocimiento de voz, escáneres de seguridad, etc. que pueden ser llamados a la transferencia de una cantidad significativa de información desde un sistema o usuario del sistema

con una pequeña "entrada" real que no sea el tacto o el sonido del usuario. Además, estaciones de acoplamiento para ordenadores y equipos de automóvil o equipos de escritorio y soportes de teléfonos inalámbricos pueden actuar como dispositivos de interfaz para los usuarios finales o a otros dispositivos y equipos, y emplear clientes (dispositivos de entrada o salida tales como ratones) o servidores para ayudar en la transferencia de datos, especialmente cuando están involucradas redes de alta velocidad. Sin embargo, los expertos en la materia reconocerán fácilmente que la presente invención reivindicada no se limita a estos dispositivos, habiendo muchos otros dispositivos en el mercado, y propuestos para su uso, que tienen por objeto 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 reivindicada es útil para aumentar el rendimiento del procesamiento de datos entre distintos elementos o dispositivos para permitir las altas velocidades de datos necesarias para la realización de la experiencia de usuario deseada.

La figura 1 es un diagrama de bloques que ilustra un entorno de ejemplo usando una interfaz MDDI. En el ejemplo de la figura 1, MDDI se utiliza para interconectar los módulos a través de la articulación de un teléfono plegable **100**. Hay que señalar aquí que, si bien determinados aspectos de la invención reivindicada en la actualidad se describen en el contexto de ejemplos concretos, tales como interconexiones MDDI en un teléfono plegable, esto se hace sólo con fines ilustrativos y no deben utilizarse para limitar la presente invención a tales aspectos. Como se comprenderá por parte de un experto en la materia, sobre la base de las enseñanzas en este documento, aspectos de la invención reivindicada en la actualidad se pueden utilizar en otros dispositivos, incluidos los que pueden beneficiarse de tener interconexiones MDDI.

Con referencia a la figura 1, una sección inferior de cubierta plegable **104** de teléfono plegable **100** incluye un chip de base de banda de módem de estación móvil (MSM) **102**. El MSM **102** es un controlador de banda de base digital. Una sección de cubierta superior plegable **114** de un teléfono plegable **100** incluye un módulo de pantalla de cristal líquido (LCD) **116** y un módulo de cámara **118**.

Aún con referencia a la figura 1, un enlace MDDI **110** conecta un módulo de cámara **118** al MSM **102**. Típicamente, un controlador de enlace MDDI está integrado en cada módulo de cámara **118** y MSM **102**. En el ejemplo de la figura 1, un servidor MDDI **122** está integrado en el módulo de cámara **118**, mientras que un cliente MDDI **106** reside en el lado MSM del enlace MDDI **110**. Típicamente, el servidor MDDI es el controlador maestro del enlace MDDI. En el ejemplo de la figura 1, los datos de píxeles del módulo de la cámara **118** son recibidos y formateados en paquetes MDDI mediante el servidor MDDI **122** antes de ser transmitidos al enlace MDDI **110**. El cliente MDDI **106** recibe los paquetes MDDI y los vuelve a convertir en datos de los píxeles del mismo formato como generados por el módulo de la cámara **118**. Los datos de los píxeles luego son enviados a un bloque apropiado en MSM **102** para su procesamiento.

Aún con referencia a la figura 1, un enlace MDDI **112** conecta el módulo LCD **116** al MSM **102**. En el ejemplo de la figura 1, el enlace MDDI **112** interconecta un servidor MDDI **108**, integrado en MSM **102**, y un cliente MDDI **120** integrado en el módulo LCD **116**. En el ejemplo de la figura 1, los datos de visualización generados por un controlador gráfico del MSM **102** son recibidos y formateados en paquetes MDDI mediante el servidor MDDI **108** antes de ser transmitido al enlace MDDI **112**. Los clientes MDDI **120** reciben los paquetes MDDI y los vuelven a convertir en datos de visualización para su uso mediante el módulo LCD **116**.

Estructura del marco

La estructura del marco original se describe en la patente US Nº 6.760.772 B2, titulada "Generación e Implementación de un protocolo de comunicación e interfaz para la transferencia de señal de datos de alta velocidad", publicada el 06 de julio de 2004 (patente '772). Esta estructura de paquetes original **200** se muestra en la figura 2A. Los campos que se muestran en la figura 2A incluyen la longitud del paquete **202**, que suele ser un valor de 16 bits, que especifica el número total de bytes en el paquete, sin incluir el campo de la longitud de los paquetes **202**, tipo de paquete **204**, que es un 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 el CRC **208**, que es una comprobación de redundancia cíclica de 16 bits calculada sobre los bytes de datos **206**, el tipo de paquete **204**, y los campos de longitud de los paquetes **202**.

Tal como se muestra en la figura 2B, la información transmitida a través del enlace MDDI se agrupa en paquetes. Los paquetes multiplicados se agrupan en un sub-marco **210**, y varios sub-marcos constituyen un marco de medios **212**. Cada sub-marco **210** comienza con un paquete de cabecera del sub-marco **214**.

La figura 3 muestra el sub-marco de la técnica anterior con una longitud fija. Se muestra un paquete de encabezado del sub-marco **214**, los datos del paquete **216** seguido de paquetes llenos **218** y el límite del sub-marco **220**. El problema surge cuando el paquete saliente pendiente **222** es demasiado grande para caber dentro de la porción restante de un sub-marco **218**, tal como se muestra. Por lo tanto, el paquete de salida pendiente **222** debe esperar hasta el próximo sub-marco que se transmite. En su lugar, los paquetes llenos se transmiten por la duración del actual sub-marco. Esto gasta ancho de banda y consume innecesariamente energía adicional. Las nuevas estructuras de marco que se describen a continuación pueden operar del mismo modo tal como se describe en la patente '772; sin embargo, se proporcionan dos nuevos modos de funcionamiento que modifican la definición de la

longitud y el tiempo del sub-marco, mejorando así el rendimiento. Los dos modos de funcionamiento que se enumeran a continuación serán identificados con un campo de versión del protocolo contenido en el paquete de encabezado del sub-marco. Para asegurar la compatibilidad, para los dispositivos del cliente que no están conectados a un servidor, el enlace MDDI puede ser educado, para adherirse a la estructura del marco de la técnica anterior primero para verificar que el cliente puede soportar las nuevas estructuras del marco que se describen a continuación. Una vez verificado, el servidor puede moverse a la nueva estructura del marco. Todo esto se puede hacer en el primer sub-marco para proporcionar una transición rápida a cualquiera de los dos formatos que se describen a continuación.

Longitud flexible del sub-marco

El primer modo de operación prevé una longitud "flexible" del sub-marco, tal como se muestra en la figura 4. El sub-marco de longitud flexible **300** tiene un paquete de encabezado de sub-marco **304**, datos del paquete **316**, y un límite del sub-marco identificado **320**. El sub-marco de longitud flexible **300** envía un paquete de cabecera del sub-marco **304** al límite del sub-marco **320**. Cuando un paquete se pide que se transmita, nunca será bloqueado, incluso si el paquete atraviesa uno o más límites del sub-marco **320**. Este modo de funcionamiento permite que el servidor MDDI transmita el siguiente paquete de cabecera al sub-bastidor **304'** en la próxima oportunidad disponible después de que se haya completado el número de bytes transmitidos en el campo de la longitud del sub-marco, incluyendo los datos de vuelta **322**. La ventaja de este modo de operación es que el paquete ya no tendrá que dividirse entre los dos sub-marcos, si los datos están disponibles al final del primer sub-marco para la transmisión. Del mismo modo, también impide que se retrase hasta el siguiente sub-marco que se transmite para un paquete que no cabe en los bytes restantes del actual sub-marco. Los paquetes de encabezado del sub-marco **304'** deben ser el primer paquete transmitido después de que el paquete actual complete el número total de bytes transmitidos en el actual sub-marco sobre el sub-bastidor de longitud especificada en el final del paquete **324**. Este procedimiento sigue proporcionando paquetes de cabecera del sub-marco **304**, que proporcionan puntos de resincronización para los enlaces de transmisión que no son completamente fiables. El texto enviado en un sub-marco después de un largo sub-marco se acorta en la cantidad que el anterior sub-marco largo se acerca para crear un sub-marco de longitud media. El concepto de longitud flexible del sub-marco mantiene la temporización que debe promediar para ser similar al tiempo del sub-marco del sistema de longitud fija del sub-marco de la figura 3, pero no evita la transmisión de un paquete y no desperdicia ancho de banda.

Longitud ilimitada del sub-marco

Este segundo modo de funcionamiento permite que el servidor utilice sólo un único sub-marco para la duración del enlace activo MDDI, tal como se muestra en la figura 5. Esto significa que el servidor MDDI transmitirá un único paquete de encabezado del sub-marco **404** cuando el enlace sale de la hibernación y no transmite nada más. La ventaja de este modo de funcionamiento es que no hay ancho de banda adicional que se utilice para transmitir otro paquete de encabezado del sub-marco. Todavía es permisible transmitir paquetes de encabezado del sub-marco mientras en este modo de funcionamiento se permita una resincronización, sin embargo el número de bytes entre estos paquetes será arbitrario y se transmite a discreción del servidor MDDI.

Paquete de secuencia de vídeo sin ventanas

El paquete de secuencia de vídeo sin ventanas permite que la información de las ventanas quede fuera del paquete de vídeo. La información de ventanas en la versión de la técnica anterior del paquete de secuencia de vídeo incluía el borde izquierdo X, el borde superior Y, el borde derecho X, el borde inferior Y, el inicio X y el inicio Y. La figura 6 representa el paquete de secuencia de vídeo sin ventanas. Tal como puede verse, varios de los atributos son similares al paquete de secuencia de vídeo de la técnica anterior. El paquete de secuencia de vídeo sin ventanas **500** incluye la longitud del paquete **502**, que contiene 2 bytes que contienen un entero de 16 bits que especifica el número total de bytes, menos dos bytes, en el paquete de secuencia de vídeo sin ventanas **500**. Un tipo de paquete **504** consiste en 2 bytes que contienen un 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 la operación de paquete de secuencia de vídeo sin ventanas **500**. A continuación, se muestra el campo bClient ID **506**. Se trata de dos bytes que contienen un entero sin signo de 16 bits para la identificación de un ID de cliente. A continuación, su descriptor de formato de datos de vídeo **508**. El descriptor de formato de datos de vídeo **508** proporciona información para el inicio de un nuevo marco y es también un número entero sin signo de dos bytes y 16 bits. A continuación se muestran los atributos de los datos de los píxeles **510**, que son también un entero de dos bytes sin signo y 16 bits, que identifica los diversos atributos de los datos de los píxeles. El número de píxeles **512** comprende un entero de dos bytes sin signo y 16 bits que especifica el número de píxeles en el campo de los datos de los píxeles **516**. EL parámetro CRC **514** comprende dos bytes que contiene un CRC de 16 bits de todos los bytes a partir de la longitud del paquete **502** para el número de píxeles **512**. Los datos de los píxeles **516** contienen información de vídeo en bruto que se mostrará. El CRC de los datos de los píxeles **518** comprende dos bytes que contienen un CRC de 16 bits de sólo los datos de los píxeles **516**. Este paquete se utiliza para modos de funcionamiento en donde toda la región de la pantalla se actualiza constantemente. Un bit en el campo existente representa la sincronización vertical e identifica la primera línea de una pantalla de datos.

Paquete de secuencia de vídeo flexible

El paquete de secuencia de vídeo flexible, tal como se muestra en la figura 7, proporciona una forma de especificar qué campos están contenidos dentro de un paquete de secuencia de vídeo con la inclusión de los bits presentes del campo. Cada bit en este campo indica si el paquete contiene el campo correspondiente. Si un campo no está contenido en el paquete, entonces se asume que el valor debe ser el mismo que la última vez que el campo fue transmitido en un paquete de vídeo. Si este campo no se ha transmitido previamente, entonces el valor se puede asumir que es igual a cero.

El paquete de secuencia de vídeo flexible **600** tiene el siguiente contenido de los paquetes:

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

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

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

Los bits presentes en el campo **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 tal como se indica en la figura 7.

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

El número de píxeles **628** comprende un entero de dos bytes sin signo de 16 bits que especifica el número de píxeles en el campo de los datos de los píxeles **632**. El parámetro CRC **630** comprende dos bytes que contienen un CRC de 16 bits de todos los bytes de la longitud del paquete **602** al byte transmitido justo antes de este parámetro CRC **630**. Los datos de los píxeles **632** contienen la información de vídeo en bruto que se muestra. En este ejemplo, si el bit 5 de los atributos del campo de los datos de los píxeles **612** se establece en uno, entonces el campo de los datos de los 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. El CRC de los datos de los píxeles **634** comprende dos bytes que contienen un CRC de 16 bits de sólo los datos de los píxeles **632**.

Paquete de encapsulación de enlace inverso mejorado

El paquete de encapsulación de enlace inverso mejorado se muestra en la figura 8. Este paquete combina la funcionalidad del paquete de medición de la demora del viaje de ida y vuelta para ayudar a alinear el servidor con la secuencia de datos entrante con el paquete de encapsulación de enlace inverso usado para transferir datos desde el cliente al servidor, tal como se describe en la versión anterior del sistema MDDI. Este paquete utiliza un patrón de sincronización para encontrar la alineación de los datos de bytes de entrada. Una vez que el patrón de sincronización se encuentra en la secuencia de datos de entrada, el servidor puede tomar una muestra de manera fiable del resto de los bits de datos de enlace inverso para poner juntos un dato de enlace inverso y una secuencia de paquetes.

El contenido del paquete para el paquete de encapsulación de enlace inverso mejorado **700** es el siguiente:

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

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

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

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 ajusta en uno, entonces el servidor solicita la información especificada por el cliente. Si el bit es cero, entonces el servidor no necesita la información del cliente. Por ejemplo, el bit 0 podría

5 indicar que el servidor necesita un paquete de la capacidad del cliente. Será enviado 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 del cliente y el estado de los paquetes. Será enviado 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 que se transmita el primer byte de datos de un paquete de enlace inverso **724**. El bit 3 podría indicar que el servidor

10 requiere que el cliente transmita la cantidad de bytes inversos a esperar antes de comenzar la transmisión inversa de paquetes. Esto es para permitir que los paquetes de enlace inverso de tamaño dinámico que satisfacen exactamente los requisitos de los clientes en la actualidad pendientes del enlace inverso de actualización de los datos.

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 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 delantero dividido por la cantidad: dos veces el divisor de velocidad inversa **710**. La velocidad 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:

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

20 Tipo de interfaz 2 que indica que la velocidad de datos inversa es igual a dos veces el reloj de datos del enlace inverso;

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

25 Tipo de interfaz 4 que indica que la velocidad de datos inversa es igual a ocho veces el reloj de datos de acoplamiento inverso.

La longitud **712** de giro 1 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 el número de bytes necesarios para los controladores de datos MDDI en el servidor para deshabilitar sus salidas. Esto se basa en el tiempo de desactivación de la salida, la velocidad de datos del enlace directo, y la selección del tipo de interfaz de enlace directo que se utiliza. La longitud **714** del giro 2 comprende 1 byte que contiene un número entero sin signo de 8 bits que especifica el número total de bytes que se asignan al giro 2. La longitud recomendada del giro 2 es el número de bytes necesarios para el retardo de ida y vuelta, más el tiempo necesario para que el servidor active sus controladores de datos MDDI. La longitud del giro 2 puede ser también ser cualquier valor mayor que el valor mínimo calculado requerido para que haya tiempo suficiente para procesar los paquetes de enlace inverso en el servidor. El

30 número máximo de bytes inversos **716** comprende 2 bytes que indican cuántos bytes inversos se pueden transmitir desde el cliente de vuelta al servidor. Esto no incluye ningún byte necesario, tales como el patrón de sincronización, o los campos de longitud de bytes de transmisión al cliente que pueden preceder a los datos de enlace inverso cuando se solicite mediante los bits en el campo de indicaciones de enlace inverso **708**. Cuando el bit 3 se ajusta, el cliente puede solicitar enviar los datos que son menores que el valor máximo en el campo de bytes inversos

35 máximos **716**. Cuando el cliente transmite un número que es menor que el campo de bytes inversos máximos **716**, el MDDI acortará el período anticipado del campo de datos y sincronización inversa **724** para maximizar la solicitud de los clientes. El parámetro CRC **718** comprende 2 bytes que contienen un CRC de 16 bits de todos los bytes de longitud del paquete **702** a la longitud de giro **712** y el campo de bytes máximos inversos **716**. Si este CRC falla la comprobación, entonces todo el paquete debe ser descartado. Todos los ceros **720** comprenden 8 bytes que contienen cada uno un entero sin signo de 8 bits igual a cero. Este campo asegura que todas las señales de datos MDDI están en un nivel logc-cero durante un tiempo suficiente para permitir que el cliente inicie la recuperación del reloj utilizando sólo MDDI_Stb antes de la desactivación de los controladores de la línea del servidor durante el campo de giro 1 **722**. El giro 1 **722** comprende un primer período de giro. El número de bytes especificados mediante el parámetro la longitud de giro 1 **712** se asigna para permitir que los controladores de línea de datos

40 MODI en el cliente se activen antes de que los controladores de línea en el servidor estén desactivados. El cliente debe activar sus controladores de línea de datos MDDI durante el bit 0 del giro 1 **722** y el servidor debe desactivar sus salidas y desactivarse completamente antes del último bit del giro 1 **722**. La señal MDDI_Stb se comporta como si NMDI-Data0 estuviera en un nivel de lógica cero durante todo el período de giro 1 **722**.

La sincronización inversa, el recuento de bytes y los paquetes de datos **724** se muestran como un único campo en la figura 8. El primer byte en este campo debe ser el patrón de sincronización (0x053F) a petición del bit dos que se establecen en una lógica del campo de las indicaciones de enlace inverso **708**. Si el bit tres se establece en el siguiente campo de enlace de transmisión inversa debe ser el número de bytes que el cliente transmitirá en el enlace inverso. Si estos datos no se piden, el cliente puede transmitir de enlace de datos inverso hasta el número de bytes especificado en el campo bytes máximos inversos **716**. Este campo debe ser seguido por el campo de longitud del paquete del primer paquete de enlace inverso. Más de un paquete se puede transmitir en el periodo de datos inverso, si hay suficiente espacio. El cliente puede enviar paquetes de relleno o llevar las líneas de datos MDDI a un nivel de lógica cero cuando no hay datos para enviar al servidor. Si las líneas de datos MDDI se llevan a cero, el servidor interpretará esto como un paquete con una longitud cero (una longitud no válida) y el servidor no aceptará

ningún paquete adicional del cliente durante la duración del actual paquete de encapsulado de enlace inverso mejorado **700**. El giro **2 726** comprende el segunda período de giro. El número de bytes se especifica mediante el parámetro de longitud del giro **2 714**. El servidor debe activar los controladores de la línea de datos MDDI y estar completamente habilitado antes del último bit del giro **2 726** y el cliente debe deshabilitar sus salidas y estar completamente deshabilitado antes del último bit del giro **2 726**. El propósito del giro **2 726** es permitir que la cantidad restante de los datos del campo de paquetes de datos inversos **724** se transmita desde el cliente. Debido a las variaciones en los distintos sistemas y la cantidad de margen de seguridad asignado, es posible que ni el servidor ni el cliente lleven las señales de datos MDDI a un nivel de lógica cero en algunas partes del campo del giro **2 726** tal como se ve mediante los receptores de línea en el sistema principal. La señal MDDI_Stb se comporta como si MDDI_Data0 estuviera en un nivel de lógica cero durante todo el período de giro **2 726**. Todos los ceros **2 728** comprenden 8 bytes que contienen cada uno un entero sin signo de 8 bits igual a cero. Este campo asegura que todas las señales de datos MDDI estén en un nivel de lógica cero durante un tiempo suficiente para permitir que el cliente inicie la recuperación del reloj utilizando MDDI-DATA0 y MDDI_Stb después de habilitar los controladores de la línea del servidor después del campo de giro **2 726**.

15 **Congelación del enlace MDDI**

El servidor MDDI puede encontrar momentos en los que necesita detener el enlace de datos MDDI, o pausar la operación del enlace. La figura 9 muestra el aspecto de congelación del enlace de la invención reivindicada en la actualidad. La figura 9 muestra los datos MDDI **900**, el estrobo (STB) **902**, y el reloj recuperado **904**. Este aspecto permite que los datos MDDI **900** se detengan durante un corto período de tiempo **906** y congelen el estado actual del cliente MDDI. Tal como se muestra, el reloj recuperado **904** en el cliente se deriva de la secuencia de datos MDDI entrantes **900** y el estrobo MDDI **902**, y como resultado, al parar el enlace MDDI se detendrá cualquier ciclo del reloj adicional **906** que se vea en el cliente. El servidor debe mantener los niveles diferenciales correspondientes al último bit de datos transmitido al entrar en este modo. El servidor MDDI no está obligado a transmitir cualquier 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 puede ser usado para prevenir subdesbordamientos dentro de un diseño de servidor MDDI si otras fuentes de datos son brevemente incapaces de mantener la secuencia de datos MDDI de salida.

Debido al consumo de energía adicional para mantener los datos MDDI **900** y el estrobo **902** activos, este estado sólo debe usarse en situaciones de corta duración. Cuando no hay un contenido significativo que se transmite durante un período más largo de tiempo, se debe utilizar el modo de hibernación para mantener el consumo de energía al mínimo.

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

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

Esta memoria describe uno o más aspectos que incorporan las características de la invención reivindicada. Los aspectos descritos son meramente ejemplos de la invención reivindicada. El alcance de la invención reivindicada no se limita a los aspectos descritos. La invención se define mediante las reivindicaciones adjuntas.

Los expertos en la materia entenderán que la información y las señales pueden representarse usando cualquiera de una variedad de diferentes tecnologías y técnicas. Por ejemplo, los datos, las instrucciones, las órdenes, la información, las señales, los bits, los símbolos, y los chips a los que pueden hacer referencia a través de la descripción anterior puede ser representados por tensiones, corrientes, ondas electromagnéticas, campos magnéticos o partículas, campos ópticos o partículas, o cualquier combinación de los mismos.

Los expertos también apreciarán que los distintos bloques lógicos ilustrativos, módulos, circuitos, y etapas de algoritmo descrito en relación con las realizaciones descritas en este documento pueden ser implementados como hardware electrónico, software, o combinaciones de los mismos. Para ilustran claramente esta capacidad de intercambio de hardware y software, varios componentes ilustrativos, bloques, módulos, circuitos, y etapas se han descrito anteriormente generalmente en términos de su funcionalidad. Si dicha funcionalidad se implementa como hardware o software depende de la aplicación y de las limitaciones de diseño impuestas en el sistema global. Los expertos pueden implementar la funcionalidad descrita en diferentes formas para cada aplicación en particular, pero estas decisiones de implementación no deben ser interpretadas como causantes de apartarse del alcance de la presente invención.

Los distintos bloques, módulos y circuitos lógicos ilustrativos descritos en relación con las realizaciones descritas en este documento pueden ser implementadas o realizadas con un procesador de propósito general, un procesador de señal digital (DSP), un circuito integrado de aplicación específica (ASIC), una disposición de puerta programable de campo (FPGA) u otro dispositivo de lógica programable, puerta discreta o lógica de transistor, componentes de hardware discretos, o cualquier combinación de los mismos diseñada para llevar a cabo las funciones descritas en

este documento. Un procesador de propósito general puede ser un microprocesador, pero con carácter subsidiario, el procesador puede ser cualquier procesador convencional, controlador, microcontrolador, o máquina de estados. El procesador también puede ser implementado como una combinación de dispositivos informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, o varios microprocesadores en
5 conjunción con un núcleo DSP, o cualquier otra configuración de este tipo.

Las etapas de un procedimiento o algoritmo descrito en relación con las realizaciones descritas en este documento pueden ser incorporadas directamente en el hardware, en un módulo de software ejecutado por un procesador, o una combinación de los mismos. Un módulo de software puede residir en una memoria de acceso aleatorio (RAM), memoria flash, memoria de sólo lectura (ROM), ROM eléctricamente programable (EPROM), ROM eléctricamente
10 programable que se puede borrar (EEPROM), 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 de ejemplo se acopla con el procesador de modo que el procesador pueda leer la información y escribir información, en el medio de almacenamiento. En la alternativa, el medio de almacenamiento puede ser parte integral del procesador. El
15 procesador y el medio de almacenamiento pueden residir en un ASIC. El ASIC puede residir en un terminal del usuario. En la alternativa, el procesador y el medio de almacenamiento pueden residir como componentes discretos en un terminal del usuario.

La descripción anterior de las realizaciones descritas se proporciona para permitir que cualquier persona experta en la materia realice o utilice la invención. Varias modificaciones a estas realizaciones serán fácilmente evidentes para los expertos en la materia, y los principios genéricos definidos en este documento pueden aplicarse a otras formas
20 de realización sin apartarse del alcance de la invención.

REIVINDICACIONES

1. Procedimiento de envío de un paquete de datos de vídeo sin ventanas en un enlace de transmisión que acopla un cliente y un servidor en un dispositivo electrónico, comprendiendo el procedimiento las etapas de:
- 5 eliminar las coordenadas del campo X-Izquierda, X-Derecha, Y-Arriba, Y-Abajo, X-Inicio, e Y-Inicio de un paquete de datos de vídeo;
- proporcionar una sincronización vertical en un campo del paquete de datos de vídeo, indicando la sincronización vertical una primera línea de una pantalla de datos, en el que las etapas de eliminar y proporcionar comprenden la creación del paquete de datos de vídeo sin ventanas; y
- enviar el paquetes de datos de vídeo sin ventanas mediante el servidor al cliente.
- 10 2. Procedimiento según la reivindicación 1, que también comprende la etapa de actualizar una línea completa de la pantalla de datos.
3. Procedimiento según la reivindicación 2, que también comprende la etapa de enviar un segundo paquete de datos de vídeo sin ventanas mediante el servidor al cliente con la sincronización vertical eliminada para incrementar una posición de datos de píxeles en la pantalla de datos.
- 15 4. Sistema para el envío de un paquete de datos de vídeo sin ventanas en un enlace de transmisión que acopla un cliente y un servidor en un dispositivo electrónico, comprendiendo el sistema:
- medios para eliminar las coordenadas del campo X-Izquierda, X-Derecha, Y-Arriba, Y-Abajo, X Inicio, e Y-Inicio de un paquete de datos de vídeo;
- 20 medios para proporcionar una sincronización vertical en un campo del paquete de datos de vídeo, indicando la sincronización vertical una primera línea de una pantalla de datos, en el que los medios para eliminar y proporcionar comprenden unos medios para crear el paquetes de datos de vídeo sin ventanas; y
- medios para enviar el paquete de datos de vídeo sin ventanas mediante el servidor al cliente.
5. Sistema según la reivindicación 4, que también comprende unos medios para actualizar una línea completa de la pantalla de datos.
- 25 6. Sistema según la reivindicación 5, que también comprende unos medios para enviar un segundo paquetes de datos de vídeo sin ventanas mediante el servidor al cliente con la sincronización vertical eliminada para incrementar una posición de datos de píxeles en la pantalla de datos.
7. Producto del programa de ordenador, que comprende:
- medios legibles por ordenador que comprenden:
- 30 un código de hacer que un paquete de datos de vídeo sin ventanas se envíe a través de un enlace de transmisión que acopla un cliente y un servidor en un dispositivo electrónico, comprendiendo el código de ordenador:
- un código para provocar una eliminación de las coordenadas de campo X-Izquierda, X-Derecha, Y-Arriba, Y-Abajo, X-Inicio, e Y-Inicio de un paquete de datos de vídeo;
- 35 un código para provocar una sincronización vertical que se proporciona en un campo del paquete de datos de vídeo, indicando la sincronización vertical una primera línea de una pantalla de datos, en el que la eliminación de las coordenadas del campo y la provisión de la sincronización vertical comprende la creación del paquete de datos de vídeo sin ventanas; y
- un código para hacer que el paquetes de datos de video sin ventanas se envía mediante el servidor al cliente.
- 40 8. Producto de programa de ordenador según la reivindicación 7, que también comprende un código para hacer que una línea completa de la pantalla de datos se actualice.
9. Producto del programa de ordenador según la reivindicación 8, que también comprende un código para hacer que un segundo paquete de datos de vídeo sin ventanas se envíe mediante el servidor al cliente con la sincronización vertical eliminada para incrementar una posición de datos de píxeles en la pantalla de datos.

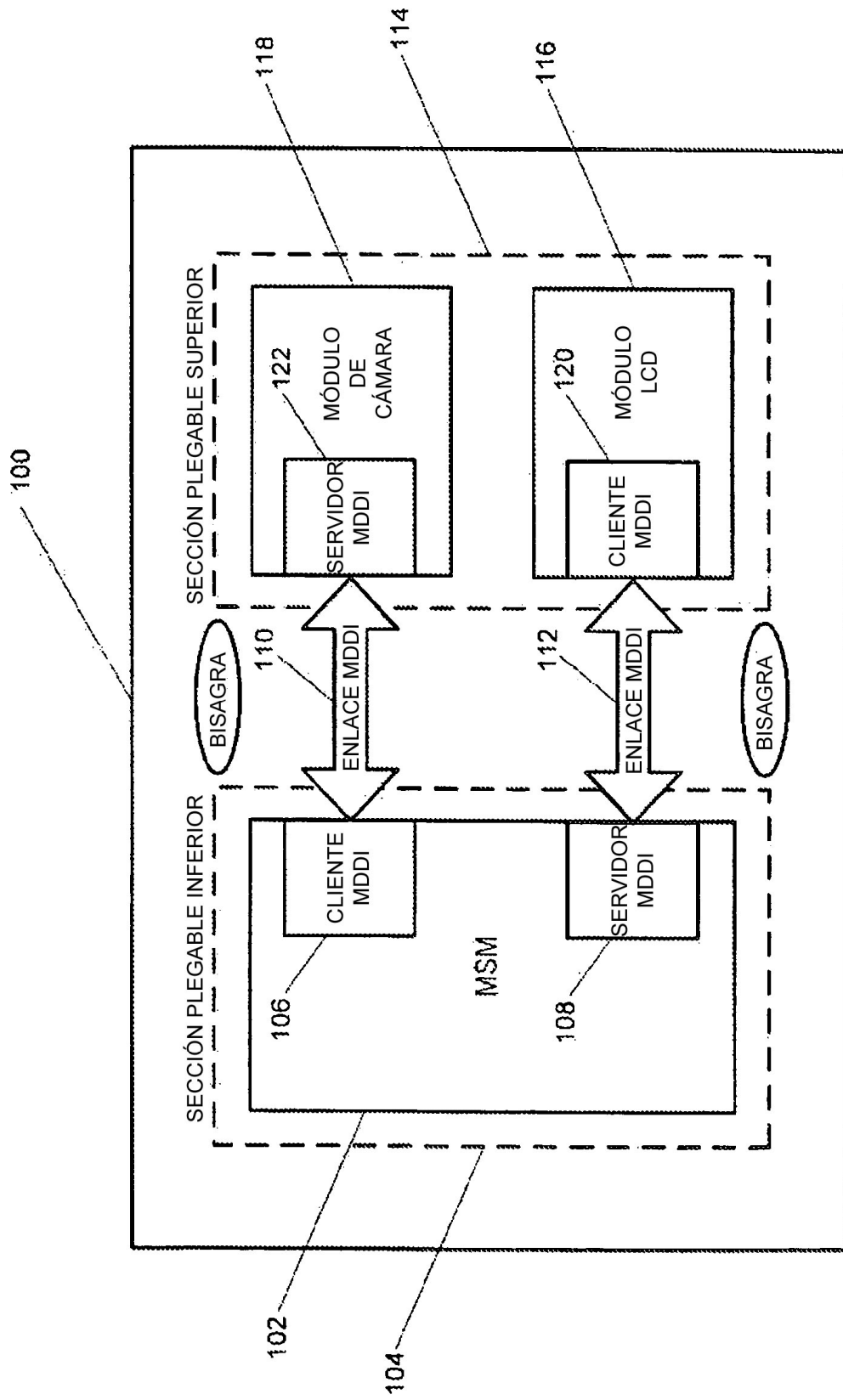


FIG. 1

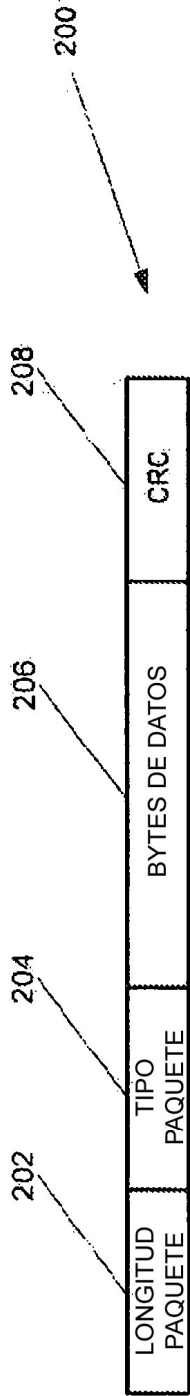


FIG. 2A

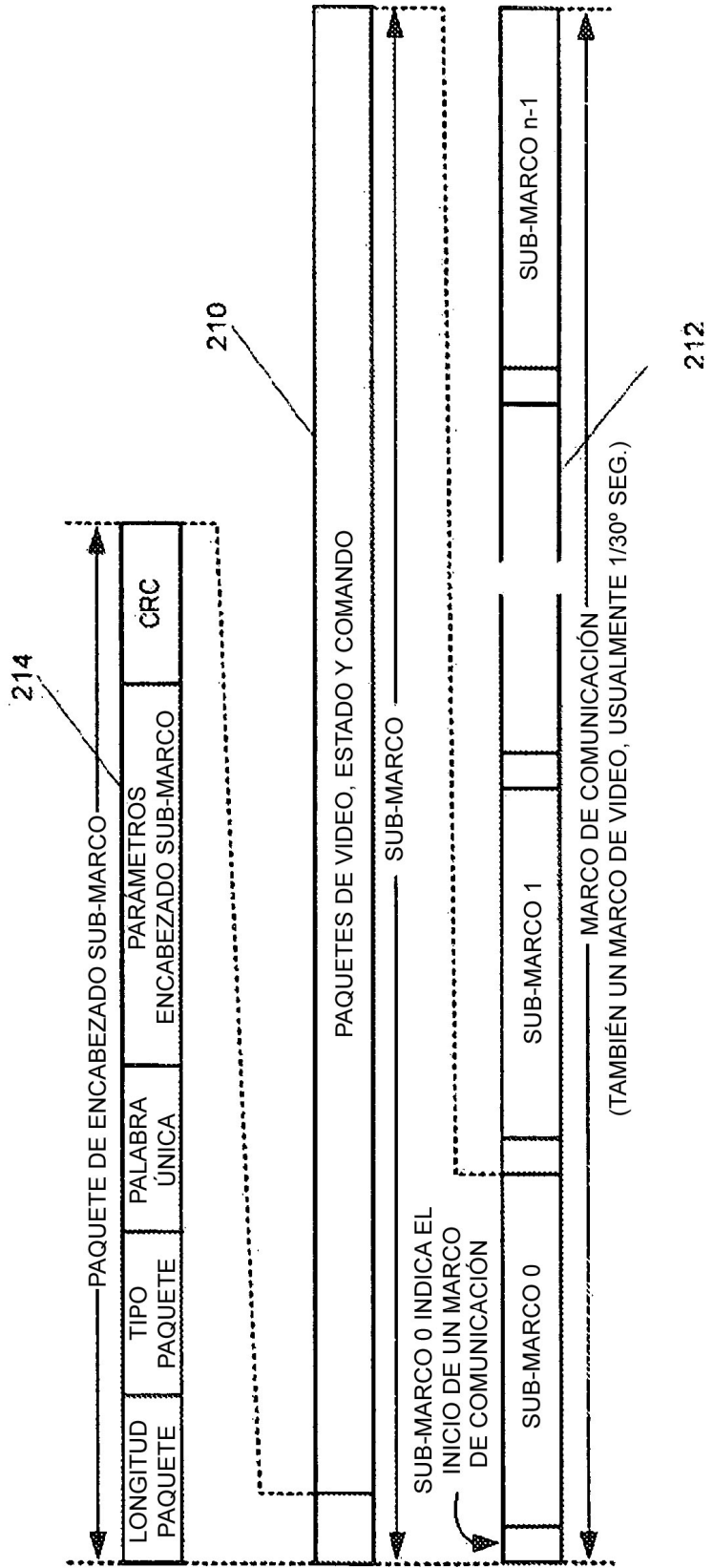


FIG. 2B

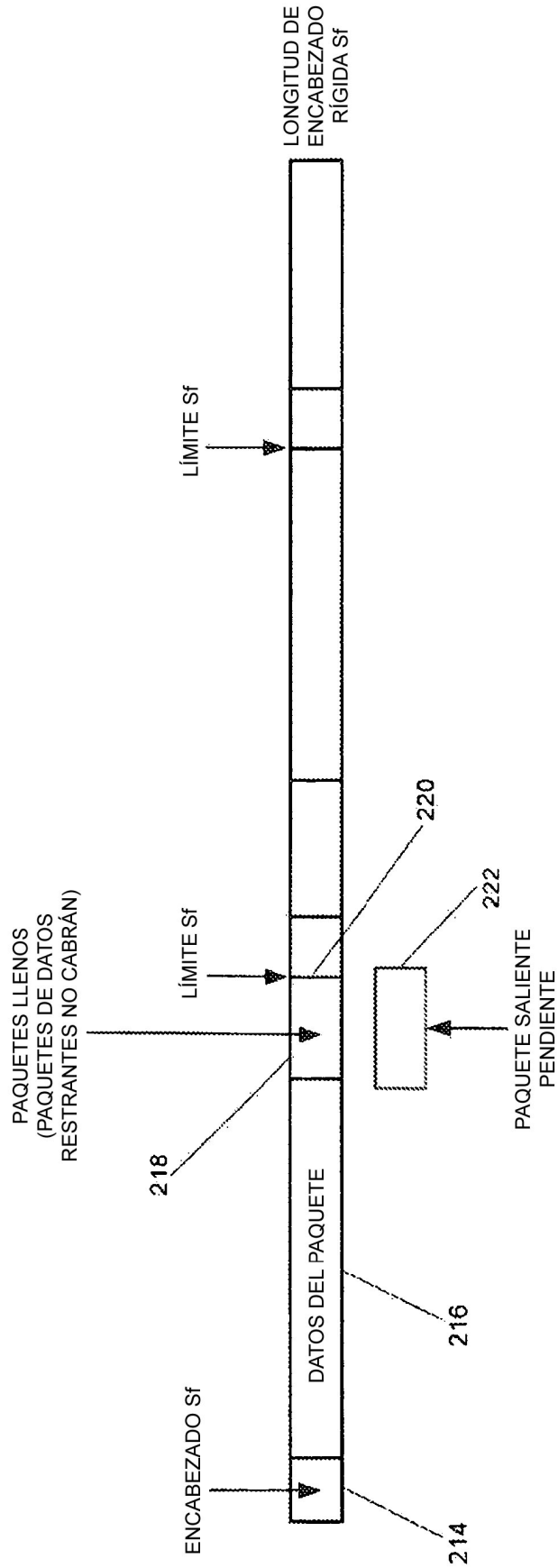


FIG. 3

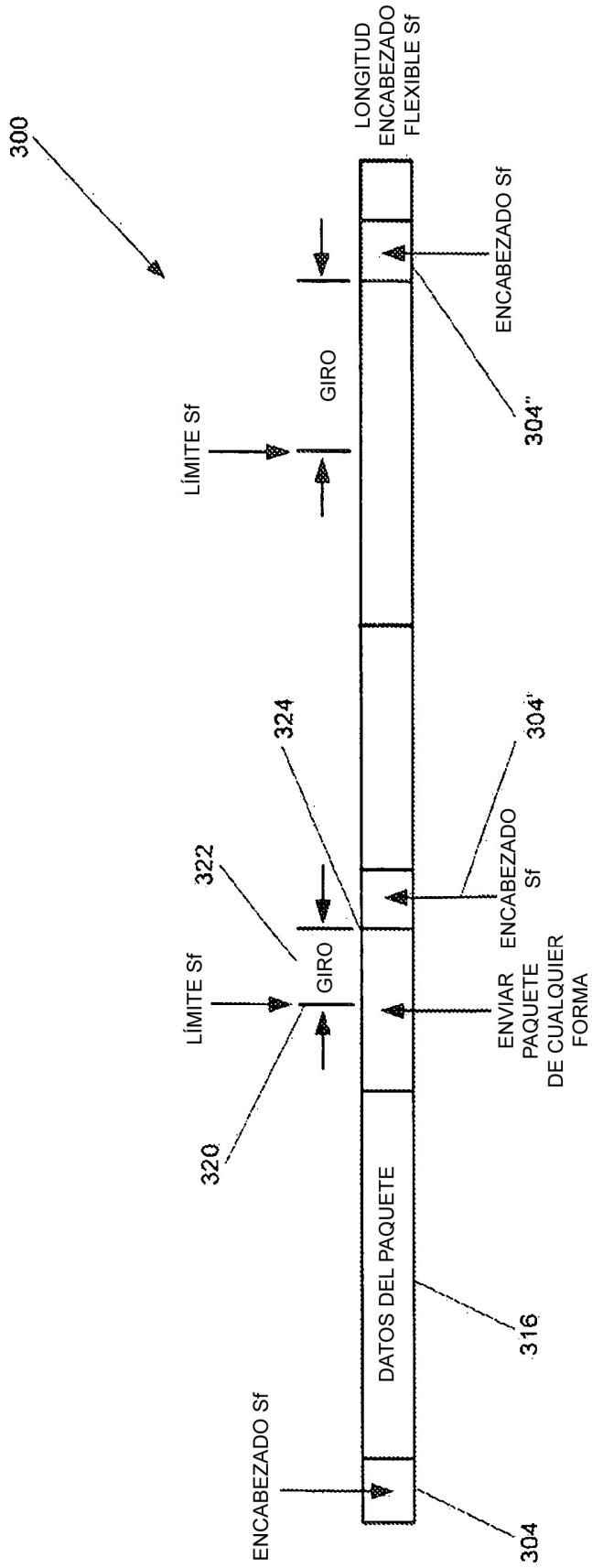


FIG. 4

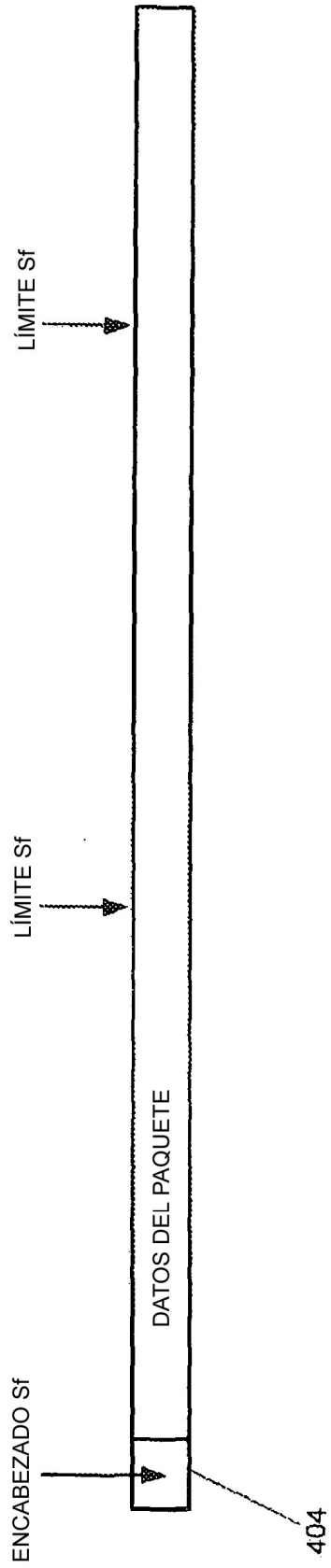


FIG. 5

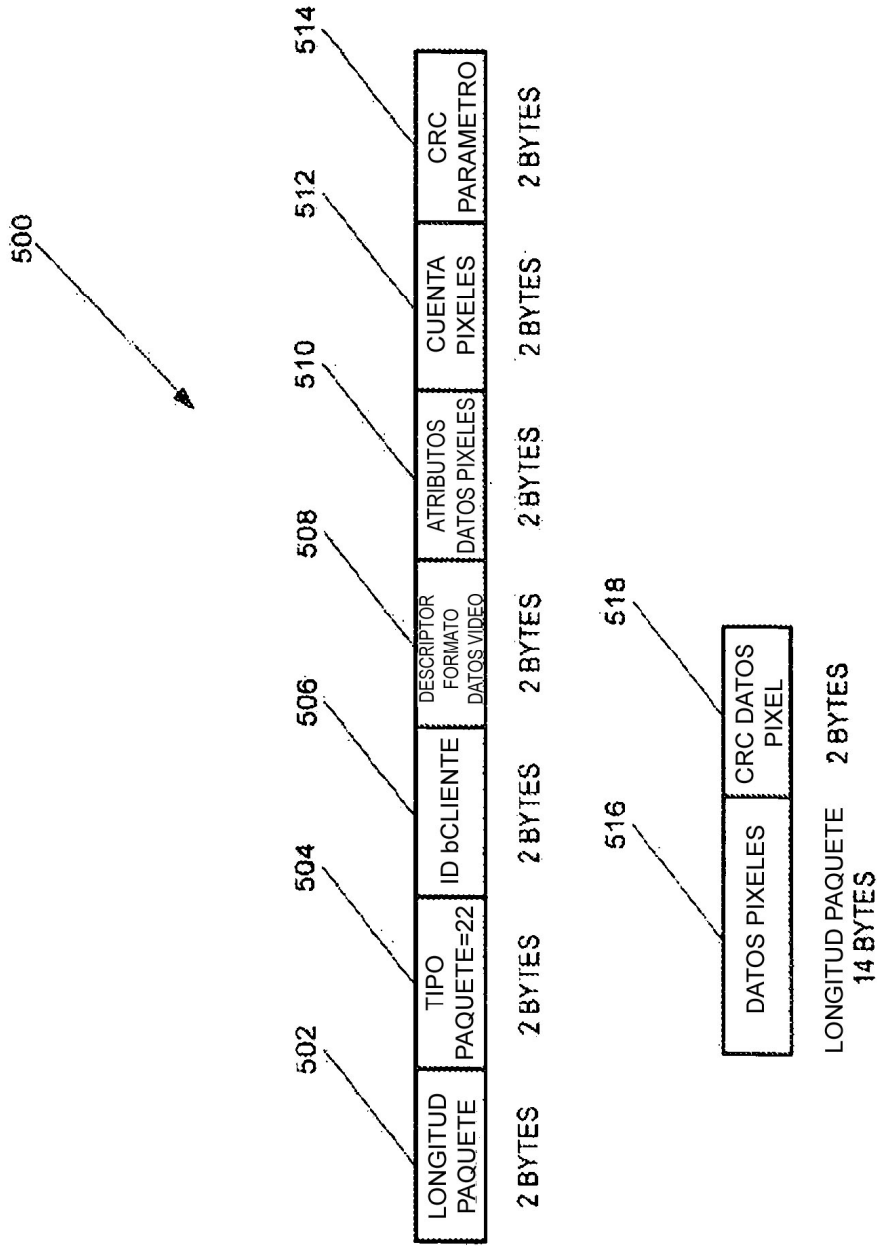


FIG. 6

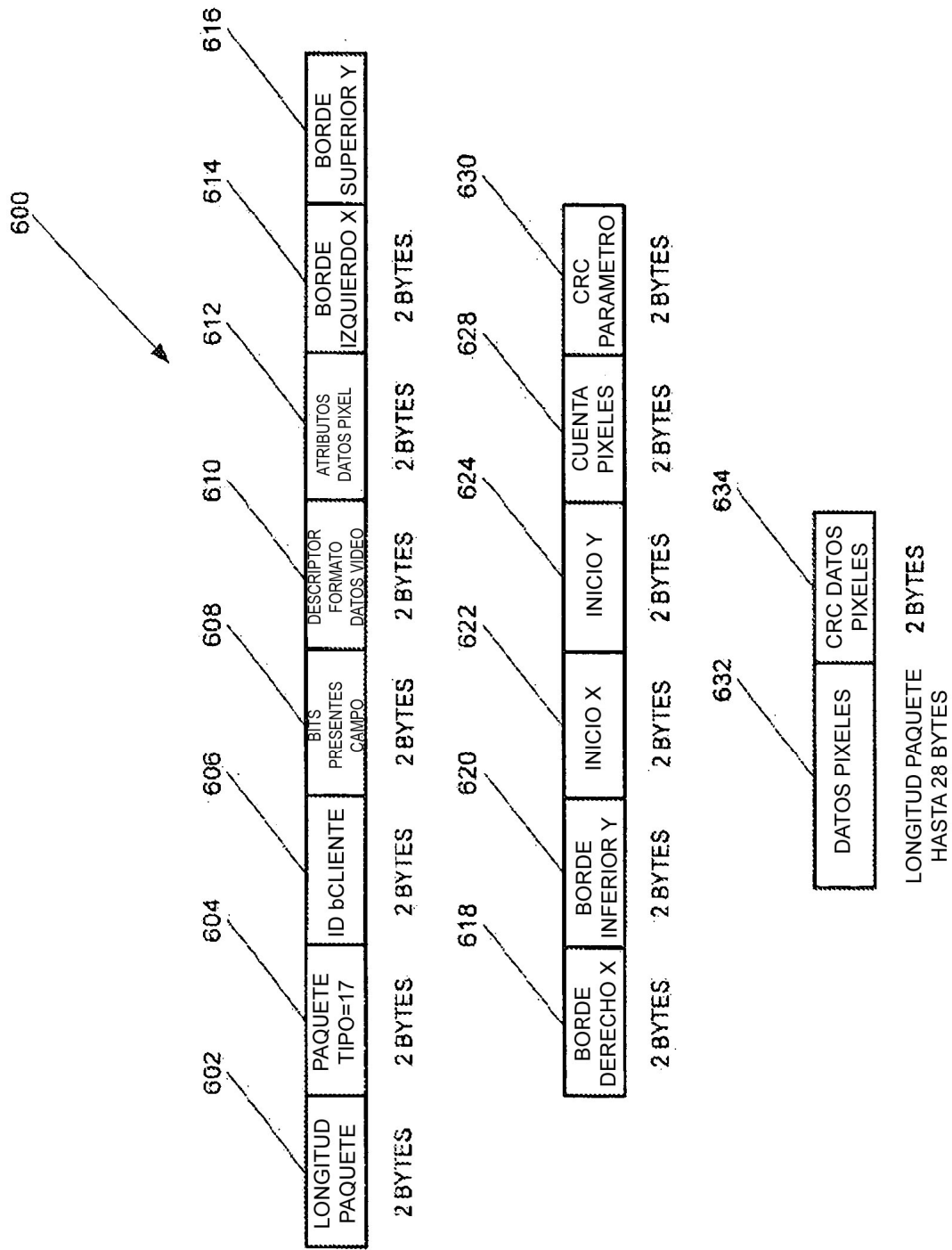


FIG. 7

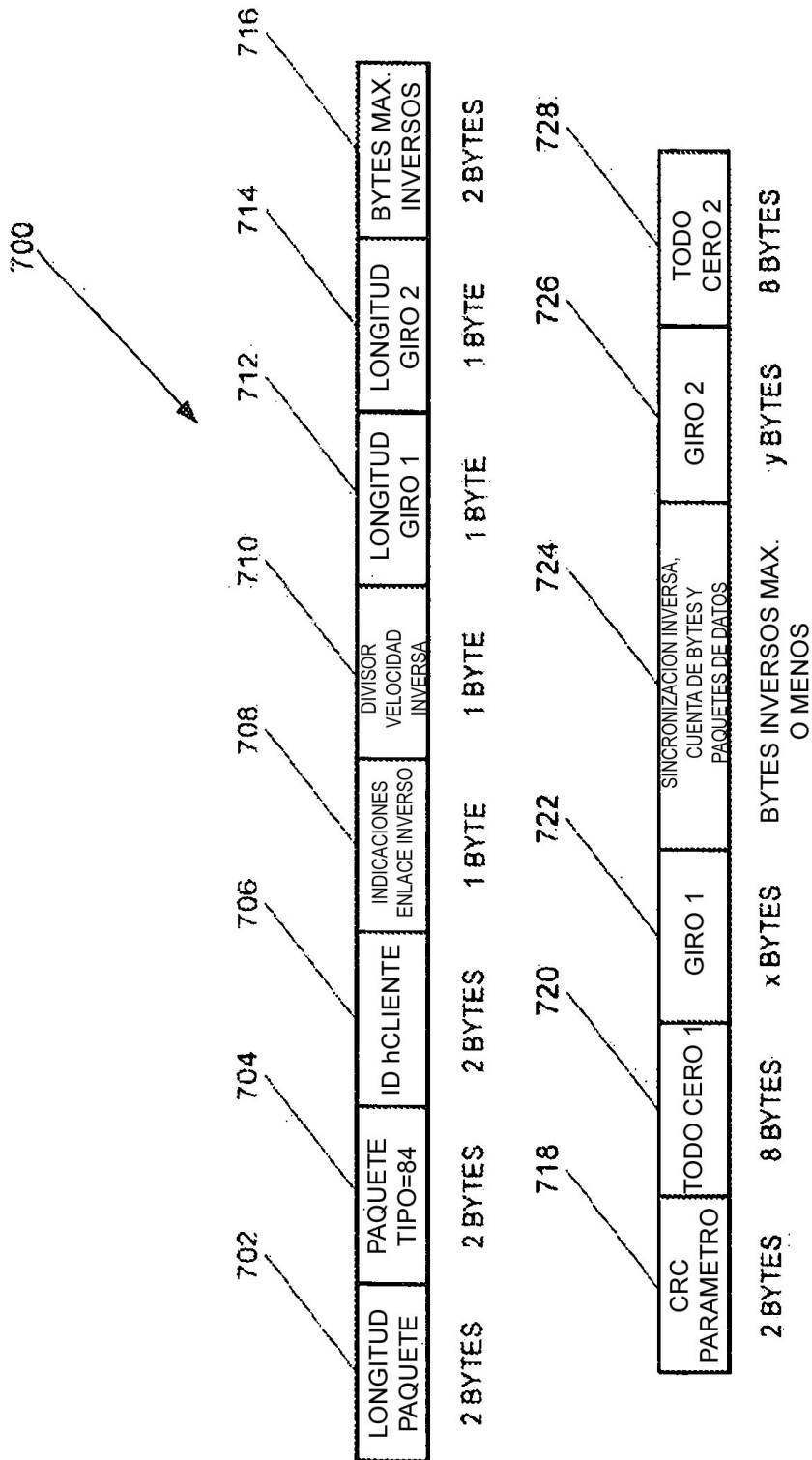


FIG. 8

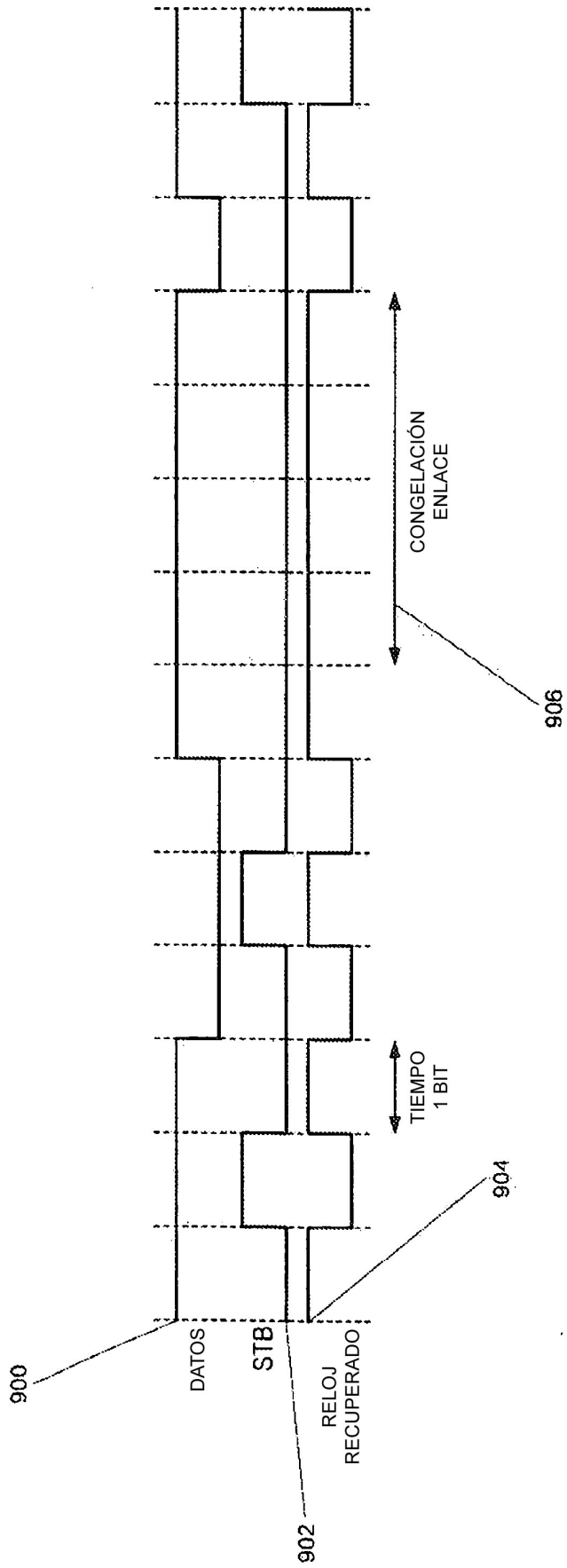


FIG. 9