(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 454 545**

(51) Int. Cl.:

***G06F 9/38*** (2006.01)
***G06F 1/10*** (2006.01)
*G06F 5/08* (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **17.12.2003** **E 03790522 (1)**

(97) Fecha y número de publicación de la concesión europea: **19.02.2014** **EP 1573516**

(54) Título: **Aparato y método de compensación de desfase de reloj**

(30) Prioridad:

**19.12.2002 US 434841 P**
**27.01.2003 US 352372**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**10.04.2014**

(73) Titular/es:

**MOSAID TECHNOLOGIES INCORPORATED (100.0%)**
**11 Hines Road, Suite 203**
**Ottawa, ON K2K 2X1 , CA**

(72) Inventor/es:

**ROTH, ALAN;**
**BECCA, OSWALD y**
**OVALLE, PEDRO**

(74) Agente/Representante:

**CURELL AGUILÁ, Mireia**

**ES 2 454 545 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

Aparato y método de compensación de desfase de reloj.

### Solicitud relacionada

La presente solicitud reivindica prioridad según la 35 U.S.C. § 119 o 365 con respecto a la solicitud US nº 10/352.372, presentada el 27 de enero de 2003, y reivindica los derechos de la solicitud provisional US nº 60/434.841, presentada el 19 de diciembre de 2002.

### Antecedentes de la invención

La velocidad de datos doble (DDR y DDRII) y la velocidad de datos cuádruple (QDR y QDRII) son arquitecturas industriales normalizadas para la memoria estática de acceso aleatorio (SRAM) en redes de alta velocidad. La arquitectura DDR duplica la velocidad de datos de la SRAM convencional al llevar a cabo dos accesos a memoria por cada ciclo de reloj. En la arquitectura QDR, el puerto de entrada y el puerto de salida están separados y funcionan de manera independiente permitiendo dos lecturas de memoria y dos escrituras en memoria por cada ciclo de reloj. Con dos lecturas y escrituras de memoria por cada ciclo de reloj, la arquitectura QDR cuadriplica la velocidad de datos de la SRAM convencional permitiendo cuatro accesos a memoria por cada ciclo de reloj.

La arquitectura QDR se diseñó originalmente para interfaces de SRAM de alta velocidad. No obstante, la arquitectura QDR ha sido adoptada para otras aplicaciones de alta frecuencia, por ejemplo, como una interfaz normalizada hacia coprocesadores basados en memoria.

La arquitectura QDR define un par de relojes maestros que se usa para controlar accesos de lectura y escritura a la SRAM. Por ejemplo, todos los datos leídos de la SRAM se alinean con los flancos de subida del par de relojes maestros.

Cuando se trabaja a una frecuencia de funcionamiento baja, por ejemplo, por debajo de 133 MHz, se dispone del tiempo suficiente para que un elemento maestro del bus, tal como un ASIC o un microprocesador acoplado al dispositivo de QDR, use los flancos de subida del par de relojes maestros para capturar los datos sincronizados con el par de relojes maestros. No obstante, a medida que la frecuencia del funcionamiento del dispositivo de QDR se incrementa, las ventanas válidas de datos y los tiempos de retención disminuyen de forma correspondiente. Los datos sincronizados con el par de relojes maestros por el coprocesador basado en memoria pueden no ser válidos cuando son capturados por el elemento maestro del bus usando el par de relojes maestros. Con el fin de permitir que el elemento maestro del bus capture datos válidos cuando se trabaja a frecuencias superiores, la arquitectura de QR define también un par de relojes de datos. El par de relojes de datos es una versión desplazada en fase del par de relojes maestros.

La arquitectura de QDR permite que el elemento maestro del bus use el par de relojes de datos para capturar los datos en lugar del par de relojes maestros con el fin de ajustarse a la configuración de datos y los tiempos de retención en el elemento maestro del bus. De este modo, el coprocesador basado en memoria debe sincronizar los datos con el par de relojes de datos después de que los mismos hayan sido leídos de los medios de almacenamiento de datos. Se puede producir una diferencia de fase (desfase) significativa entre el par de relojes maestros y el par de relojes de datos.

En relación con esto, se hace referencia al documento US nº 6.097.775 el cual describe un circuito sincronizador para transferir señales entre dos dominios de reloj en los cuales una primera unidad sincronizadora y una segunda unidad sincronizadora forman un protocolo de señalización de entrada en contacto. Las disposiciones del documento US nº 6.097.775 presentan las mismas dificultades que se han expuesto anteriormente en líneas generales con la arquitectura de QDR.

### Sumario de la invención

Se presenta un circuito de compensación de desfase, el cual es compatible con los requisitos de la interfaz de QDR II y acepta una diferencia de fase significativa entre un reloj de entrada y un reloj de salida.

Un circuito de retención transparente presenta dos estados, abierto y cerrado. Mientras está abierto, el circuito de retención transparente deja pasar datos de la entrada a la salida. Mientras está cerrado, el circuito de retención transparente retiene los datos presentes en la entrada cuando se produjo la transición del estado abierto al cerrado. Mientras está abierto, el circuito de retención transparente proporciona una ventana para capturar los datos presentes en la entrada con el fin de evitar la espera de un flanco de reloj sucesivo para dejar pasar datos de la entrada a la salida.

Un circuito de sincronización para resincronizar datos desde un reloj de entrada a un reloj de salida incluye un primer circuito de retención transparente, un segundo circuito de retención transparente y un circuito de retención de salida.

El primer circuito de retención transparente recibe los datos y se activa mediante impulsos de reloj por medio del reloj de entrada. El segundo circuito de retención transparente recibe datos del primer circuito de retención transparente y se activa mediante impulsos de reloj por medio de un reloj de salida retardado. El reloj de salida retardado es una versión retardada del reloj de salida. El circuito de retención de salida recibe datos del segundo circuito de retención transparente y se activa mediante impulsos de reloj por medio del reloj de salida. El reloj de salida retardado puede incluir un retardo de inserción. El reloj de salida puede ser una versión de bucle enganchado por retardo del reloj de salida retardado con el retardo de inserción eliminado.

El reloj de entrada puede ser un reloj K# de un par de relojes maestros y el reloj de salida un reloj C# de un par de relojes de datos. El circuito de retención de salida se puede disparar por flancos. Se puede dar salida a datos desde el circuito de retención de salida con una velocidad de datos doble.

El primer circuito de retención transparente y el segundo circuito de retención transparente dejan pasar datos recibidos cuando están abiertos y retienen unos últimos datos recibidos cuando están cerrados. En una forma de realización, el primer circuito de retención transparente está abierto cuando el reloj de entrada es “1” lógico y está cerrado cuando el reloj de entrada es “0” lógico, y el segundo circuito de retención transparente está abierto cuando el reloj de salida retardado es “1” lógico y está cerrado cuando el reloj de salida es “0” lógico.

**Breve descripción de los dibujos**

Los anteriores y otros objetivos, características y ventajas de la invención resultarán evidentes a partir de la siguiente descripción más particular de formas de realización preferidas de la invención, según se ilustra en los dibujos adjuntos en los cuales los caracteres de referencia iguales remiten a las mismas partes en la totalidad de las diferentes vistas. Los dibujos no están necesariamente a escala, poniéndose énfasis, en cambio, en la ilustración de los fundamentos de la invención.

La figura 1 es un diagrama de bloques de un dispositivo que incluye un circuito de compensación de desfase para sincronizar datos recibidos desde medios de almacenamiento de datos de acuerdo con los fundamentos de la presente invención;

las figuras 2A a 2B son un diagrama de bloques con mayor detalle de la interfaz de salida de datos acoplada a los medios de almacenamiento de datos representados en la figura 1;

la figura 3 es un diagrama de temporización que ilustra el retardo de inserción;

la figura 4 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase mostrado en las figuras 2A a 2B para datos anticipados y desfase moderado entre los relojes;

la figura 5 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase representado en las figuras 2A a 2B para datos tardíos y un desfase moderado entre los relojes;

la figura 6 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase mostrado en las figuras 2A a 2B para datos anticipados y un desfase en el peor de los casos entre los relojes;

la figura 7 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase representados en las figuras 2A a 2B para datos anticipados y un desfase en el peor de los casos entre los relojes;

la figura 8 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase mostrado en las figuras 2A a 2B para datos anticipados y sin desfase entre los relojes;

la figura 9 es un esquema de una forma de realización de uno cualquiera de los circuitos de retención transparentes representados en las figuras 2A a 2B;

la figura 10 es un esquema del detector de reloj representado en las figuras 2A a 2B;

la figura 11 es un diagrama de bloques de uno cualquiera de los bucles enganchados por retardo representados en la figura 2; y

la figura 12 es un esquema de una forma de realización del detector de flancos y el biestable de SR representados en las figuras 2A a 2B.

**Descripción detallada de la invención**

A continuación se proporciona una descripción de formas de realización preferidas de la invención.

La figura 1 es un diagrama de bloques de un dispositivo 100 que incluye un circuito de compensación de desfase 106 para sincronizar datos recibidos desde los medios de almacenamiento de datos 110 de acuerdo con los fundamentos de la presente invención. El dispositivo 100 proporciona datos almacenados en los medios de almacenamiento de datos 110 como respuesta a una solicitud de leer los datos recibidos desde un elemento maestro de bus 101. El elemento maestro de bus 101 puede ser un microprocesador o un circuito integrado de aplicación específica (ASIC) con capacidad de emitir una orden al dispositivo 100.

Los datos obtenidos a la salida de los medios de almacenamiento de datos 110 se sincronizan con el par de relojes maestros. El circuito de salida de datos 104 resincroniza los datos recibidos desde los medios de almacenamiento de datos con un reloj de salida 115 seleccionado por el circuito selector de relojes 108. Los datos procedentes de los medios de almacenamiento de datos, sincronizados con el par de relojes maestros 106, son acondicionados por el circuito de compensación de desfase 106, de manera que los datos transmitidos al circuito de retención de salida 102 se pueden sincronizar con el reloj de salida.

El circuito de compensación de desfase 106 compensa el desfase entre el par de relojes maestros y el reloj de salida seleccionado, y el desfase entre los datos y el par de relojes maestros. Haciendo referencia a las figuras 2A a 2B, el circuito de compensación de desfase 106 incluye dos circuitos de retención transparentes 120, 130. Cada circuito de retención transparente 120, 130 presenta dos estados; abierto y cerrado. Cuando el circuito de retención está en el estado abierto, los datos presentes en la entrada pasan a la salida. Cuando el circuito de retención está en el estado cerrado, los datos presentes en la entrada al producirse la transición del estado abierto a cerrado son retenidos en la salida del circuito de retención.

En la forma de realización representada, los circuitos de retención 120, 130 están abiertos cuando la señal de reloj respectiva acoplada a la entrada del reloj es “1” y están cerrados cuando la señal de reloj respectiva es “0”. Cuando está abierto, el circuito de retención transparente proporciona una ventana para capturar datos en la entrada en lugar de esperar a un flanco de reloj sucesivo.

Haciendo referencia de nuevo a la figura 1, los medios de almacenamiento de datos 110 pueden incluir memoria o registros para almacenar datos y presentan un puerto de entrada 118 y un puerto de salida 116 separados que funcionan de manera independiente, permitiendo la lectura y escritura simultánea de datos. Los datos obtenidos a la salida, recibidos por el circuito de salida de datos 104 en el puerto de salida 116 se sincronizan con un par de relojes maestros 112.

El circuito de salida de datos 104 sincroniza datos recibidos desde los medios de almacenamiento de datos 110 con un reloj de salida. Un circuito selector de relojes 108 selecciona el reloj de salida para sincronizar la salida de datos 122.

En una forma de realización, los datos a los que se da salida a través del puerto de salida 116 se sincronizan con los flancos de subida del par de relojes maestros 112. No obstante, en unas formas de realización alternativas, la salida de datos se puede sincronizar con los flancos de caída del par de relojes maestros 112. Después de que se haya dado salida a los datos sincronizados con el par de relojes maestros desde los medios de almacenamiento de salida 110, los datos se pueden sincronizar con un reloj de salida 115. El par de relojes de datos 114 es una versión desplazada en fase del par de relojes maestros 112. El circuito de compensación de desfase 106 gestiona un desplazamiento de fase (desfase) de hasta 180 grados entre el par de relojes maestros 112 y el par de relojes de datos 114.

El circuito selector de relojes 108 incluye un detector de reloj para detectar una señal de reloj en el par de relojes de datos 114. El detector de reloj se describe a continuación junto con la figura 5. Si se detecta una señal de reloj en el par de relojes de datos 114, una de las señales de reloj del par de relojes de datos 114 se selecciona como el reloj de salida retardado 115 para que el circuito de compensación de desfase 106 acondicione los datos de manera que los mismos se puedan sincronizar con el reloj de salida por parte del circuito de retención de salida 102. En caso contrario, uno de entre el par de relojes maestros 112 se selecciona como el reloj de salida retardado 115 para el circuito de compensación de desfase 106.

Después de que los datos se hayan acondicionado basándose en el reloj de salida retardado 115, los datos acondicionados 123 a los que ha dado salida el circuito de compensación de desfase 106 se acoplan a un circuito de retención de salida 102. El circuito de retención de salida 102 sincroniza los datos acondicionados con el reloj de salida (reloj de salida de DLL) 117 para proporcionar una salida de datos sincronizada con el reloj de salida 117. Las figuras 2A a 2B son un diagrama de bloques más detallado de la interfaz de salida de datos 104 acoplada a los medios de almacenamiento de datos 110 representados en la figura 1. En la forma de realización representada, el circuito de retención de salida 102 incluye una circuitería para generar una salida de datos de velocidad de datos dual (DDR). No obstante, en una forma de realización alternativa, se puede proporcionar una salida de velocidad de

datos única acoplando la entrada del biestable de tipo D 150 directamente a la salida del circuito de retención transparente 130 en el circuito de compensación de desfase 106 y activando mediante impulsos de reloj el biestable de tipo D 150 con la salida DLL#_CK del DLL 210.

Tal como se ha descrito anteriormente, el par de relojes de datos 114 es una versión desplazada en fase del par de relojes maestros 112. En la forma de realización representada para la arquitectura de QDR, el par de relojes maestros 112 incluye una señal K_CLK y una señal K#_CLK (figura 3). La señal K#_CLK es la señal K_CLK desplazada en fase 180 grados. El par de relojes de datos 114 incluye una señal C_CLK y una señal C#_CLK. La señal C#_CLK es la señal C_CLK desplazada en fase 180 grados.

En la forma de realización representada, los medios de almacenamiento de datos 110 son una memoria estática de acceso aleatorio (SRAM) de puerto dual, con puertos de entrada y salida independientes separados. Cada uno de entre el puerto de entrada 118 y el puerto de salida 116 incluye un bus de datos de 36 bits. El puerto de entrada 118 incluye también señales de dirección y de control. Todos los datos y órdenes que se introducen a través del puerto de entrada 118 y los datos a los que se da salida a través del puerto de salida 116 se sincronizan con el par de relojes maestros (K_CLK, K#_CLK) 112.

En una forma de realización alternativa, los medios de almacenamiento de datos 110 pueden ser una memoria direccionable por contenido (CAM) o una memoria dinámica de acceso aleatorio (DRAM). Los medios de almacenamiento de datos también pueden ser un bloque lógico, por ejemplo, un bloque de registros para almacenar datos.

El puerto de entrada 118 acepta datos de velocidad de datos doble, es decir, se puede recibir una orden o datos nuevos dos veces cada periodo de K_CLK. Por ejemplo, en una forma de realización, se recibe una orden o datos nuevos en cada flanco (de bajada y de subida) de la señal de K-CLK capturando la orden o datos tanto en el flanco de subida de K_CLK como en el flanco de subida de K#_CLK. Los medios de almacenamiento de datos pueden aceptar una orden o datos nuevos dos veces cada periodo de reloj aun cuando la orden pueda tardar en completarse más de un periodo de K_CLK.

Los datos reenviados desde el puerto de salida 116 se sincronizan con el par de relojes maestros. El circuito de compensación de desfase 106 transmite los datos reenviados desde el puerto de salida 116 en función del reloj de salida retardado 115. K_CLK y K#_CLK son versiones retardadas del reloj K y el reloj K# recibidos en los pines de entrada del dispositivo. Las señales de C_CLK y C#_CLK son versiones retardadas del reloj C y el reloj C# recibidos en los pines de entrada del dispositivo. Los bloques de retardo 231, 232, 233, 234 se refieren al retardo debido a memorias intermedias de entrada, pistas de señales y otros componentes del dispositivo. El reloj de salida retardado 115 es o bien una versión retardada del reloj K# o bien una versión retardada del reloj C# en función de si el detector de reloj 240 detecta una señal de reloj sobre el par de relojes de datos 114.

El detector de reloj 240 puede ser cualquier detector de reloj conocido en la técnica. A continuación se describe una forma de realización de un detector de reloj junto con la figura 4. La salida 202 del detector de reloj 240 controla los multiplexores 200 y 220. El detector de reloj 240 detecta si se recibe una señal de reloj en el par de relojes de datos 114. El estado de la señal de detección C_Clock obtenida a la salida del detector de reloj 240 y acoplada a los multiplexores 220, 200 selecciona si se transmite el K#_CLK o el C#_CLK al reloj de salida retardado. Si se detecta una señal de reloj 202 en el par de relojes de datos, la C#_CLK se reenvía como reloj de salida retardado, a través del multiplexor 200, y a la entrada del bucle enganchado por retardo (DLL) 210, y C_CLK se acopla a la entrada del bucle enganchado por retardo (DLL) 230 a través del multiplexor 220. Si no se detecta una señal de reloj en el par de relojes de datos, K#_CLK y K_CLK se reenvían a través de los multiplexores 200, 220.

Tal como se ha descrito anteriormente, el circuito de compensación de desfase 106 incluye dos circuitos de retención transparentes 120, 130. El puerto de salida 116 de los medios de almacenamiento de datos 110 está acoplado (datos A) al circuito de retención transparente de 36 bits 120. Las salidas de datos (datos B) del circuito de retención transparente 120 están acopladas a las entradas de datos del circuito de retención transparente 130. El circuito de retención transparente 120 es controlado por K#_CLK y el circuito de retención transparente 130 es controlado por el reloj de salida retardado 115. Mientras K#_CLK es “1” lógico, el circuito de retención transparente 120 está abierto y se transfieren datos desde las entradas de datos (bus de datos A) a las salidas de datos (bus de datos B). Mientras K#_CLK es “0” lógico, el circuito de retención 120 está cerrado y los datos capturados al producirse el flanco de caída de K#_CLK son almacenados por el circuito de retención 120 y se da salida a los mismos en el bus de datos B. Mientras el circuito de retención 120 está cerrado, los cambios en el bus de datos A de entrada no dan como resultado cambios en el bus de datos B de salida. El circuito de retención transparente 130 funciona de la misma manera como respuesta al reloj de salida retardado. Sin desfase entre K#_CLK y el reloj de salida retardado, los datos en el bus A se transmiten tal como son recibidos en el bus A a través de los dos circuitos de retención 120, 130 al bus de datos C. Si existe un desfase entre K#_CLK y el reloj de salida retardado, los datos recibidos en el bus A se transmiten al circuito de retención 120 tal como son recibidos y son almacenados por el circuito de retención 120 para transmitir datos válidos en el bus de datos B para su transferencia al bus de datos C mientras el circuito de retención 130 está abierto. El funcionamiento de los circuitos de retención transparentes se describe a continuación junto con las figuras 4 a 8.

El reloj de salida retardado 115 no es una señal enganchada por DLL y por lo tanto padece el problema bien conocido del retardo de inserción. Un retardo de inserción es el tiempo que tarda una señal en desplazarse desde un PIN de entrada en un circuito integrado al lugar en el que la señal es usada en el circuito integrado. El retardo de inserción se produce debido a retardos resistivos y capacitivos de los hilos y componentes físicos del sistema así como al tiempo de transición a través de las memorias intermedias de entrada.

Los bucles enganchados por retardo (DLL) 210, 230 se sintonizan de manera fina para un intervalo particular de frecuencias de reloj y compensan el retardo de inserción. La figura 3 ilustra el par de relojes de datos (reloj C, reloj C#) según es recibido en los pines de entrada del dispositivo 100. Tal como se muestra en la figura 3, los flancos (de caída y de subida) del reloj de salida retardado se producen con un retardo de inserción 300 después de los flancos respectivos de caída y subida del par de relojes de datos. El DLL compensa el retardo de inserción proporcionando relojes de salida de DLL (DLL_CK#, DLL_CK) sin el retardo de inserción. Por lo tanto, las salidas de DLL están alineadas en fase con las señales del par de relojes de datos en los pines. El funcionamiento de los DLL se describe con mayor detalle a continuación junto con la figura 11.

Haciendo referencia de nuevo a las figuras 2A a 2B, las salidas de los DLLs 230, 210 están acopladas a un detector de flancos 190. El detector de flancos 190 puede ser cualquier detector de flancos bien conocido para los expertos en la materia. El detector de flancos da salida a un impulso positivo sobre la señal de reloj de DDR 191 al producirse una detección de un flanco de subida de la señal DLL_CK o el flanco de subida de la señal DLL_CK#. Un flanco de subida sobre la señal de reloj de DDR 191 activa por impulsos de reloj el biestable de tipo D 150 para producir la salida de velocidad doble de datos. Un biestable de activación-reinicialización 180 está acoplado también al detector de flancos 190. El estado del biestable de SR 180 cambia con cada flanco de subida de la señal de salida 192 del detector de flancos. La salida 193 del biestable de activación-reinicialización está acoplada al elemento de retardo 185. La salida del elemento de retardo 185 está acoplada al multiplexor 140 para seleccionar a qué 18 bits de los datos de 36 bits se les da salida en cada flanco de los relojes de salida de DLL 191. A los bits de datos 35 a 18 se les da salida como respuesta al flanco de subida de C_CLK y a los bits de datos 17 a 0 se les da salida como respuesta al flanco de subida de C#_CLK.

La figura 4 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase representado en las figuras 2A a 2B para datos anticipados y un desfase moderado entre el par de relojes maestros y el par de relojes de datos. En la forma de realización representada, el detector de reloj ha detectado una señal de reloj en el par de relojes de datos, y la salida de datos se sincroniza con los flancos de subida del par de relojes de datos (C_CLK, C#_CLK). La C_CLK es una versión retardada de la K_CLK, y la C#_CLK es una versión retardada de la K#_CLK. El diagrama de temporización se describe en combinación con el diagrama de bloques de las figuras 2A a 2B. Se produce un desfase moderado 800 entre los flancos de subida y de caída de las versiones retardadas respectivas de los relojes y un desfase de datos entre los datos válidos y el flanco de subida de la K_CLK.

En el instante de tiempo 801, se da salida de manera anticipada a datos válidos, desde los medios de almacenamiento de datos, en el puerto 116 sobre el bus de datos A antes que el flanco de subida de la K#_CLK. Los datos recibidos desde los medios de almacenamiento de datos son válidos para un periodo del reloj K. Los datos válidos se muestran de manera que se producen como respuesta al primer flanco de subida de K_CLK, aunque los expertos en la materia apreciarán que puede tardarse varios ciclos de reloj K en producir esta salida. En el instante de tiempo 802, el flanco de subida de K#_CLK abre el circuito de retención transparente 120 y se transfieren datos al bus de datos B. En el instante de tiempo 803, el flanco de subida de C#_CLK abre el circuito de retención transparente 130 y los datos válidos se transfieren al bus de datos C. Mientras K#_CLK está a un nivel bajo, los últimos datos recibidos sobre el bus de datos A se almacenan en el circuito de retención 120. De modo similar, cuando la C#_CLK está en el nivel bajo los últimos datos recibidos sobre el bus de datos B mientras C#_CLK está en el nivel alto se almacenan en el circuito de retención 130.

Haciendo de nuevo referencia a las figuras 2A a 2B, después de que se hayan transferido los datos de 36 bits al bus de datos C, se transmiten a la vez 18 bits a través del biestable de tipo D 150 con una velocidad de datos doble. La señal de control 186 del multiplexor retardado controla si se van a transmitir los 18 bits inferiores o superiores. En el instante de tiempo 804, el siguiente flanco de subida de C#_CLK (y su señal derivada enganchada por DLL DLL_CK#) retiene los bits de datos 35 a 18 en la entrada del biestable de tipo D 170 del bus 133 hacia el bus 171. El impulso resultante en la salida 191 del detector de flancos activa mediante impulsos el biestable de tipo D 150 y se da salida a los bits de datos 17 a 0 a través de la memoria intermedia 160. El impulso conmuta también el estado del biestable de activación reinicialización 180. La salida del biestable de activación reinicialización se retarda a través del retardo 185 y conmuta el estado de la habilitación del multiplexor para permitir que los bits de datos 35 a 17 en el bus 171 pasen a través del multiplexor al bus 141.

En el instante de tiempo 805, el siguiente flanco de subida de C_CLK (y su señal derivada enganchada por DLL DLL_CK) también provoca la generación de un impulso en la señal 191, que activa mediante impulsos de reloj el biestable 150 para retener bits de datos 35 a 18. A continuación, la memoria intermedia 160 da salida a los bits de datos 35 a 18. Un experto en la materia apreciará que el sistema está diseñado de manera que se cumplan los

requisitos de configuración y retención de los biestables 170 y 150. El circuito de compensación de desfase acondiciona los datos de tal manera que se da salida a datos válidos en el bus de datos C antes del flanco respectivo (de subida o caída) del reloj de salida, de modo que los datos válidos se sincronizan con el reloj de salida.

La figura 5 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase representado en las figuras 2A a 2B para datos tardíos y un desfase moderado entre el par de relojes maestros y el par de relojes de datos. Estos datos son válidos sobre el bus de datos A en el instante de tiempo 900 después del flanco de subida de K#_CLK. Con K#_CLK en estado alto, el circuito de retención 120 es transparente y los datos se transfieren desde el bus de datos A a B. Adicionalmente, puesto que C#_CLK está en estado alto, el circuito de retención transparente 130 está abierto y los datos válidos se transfieren desde el bus de datos B a C. Poco tiempo después de esto, en el instante de tiempo 901, K#_CLK realiza una transición al estado bajo y el circuito de retención 120 almacena los últimos datos recibidos sobre el bus de datos A y transmite los datos almacenados en el bus de datos B. En el instante de tiempo 902, C_CLK pasa al nivel alto y C#_CLK pasa al nivel bajo, el circuito de retención transparente 130 almacena los últimos datos recibidos en el bus de datos B y transmite los datos almacenados en el bus de datos C.

En el instante de tiempo 903, el flanco de subida de C#_CLK (y su señal derivada enganchada por DLL DLL_CK#), a través del detector de flancos 190, retiene los 18 bits inferiores (D[17:0]) del bus de datos de 36 bits en el biestable de tipo D 150 para transmitir los 18 bits inferiores sobre el bus de salida.

En el instante de tiempo 904, el flanco de subida de C_CLK (y su señal derivada enganchada por DLL DLL_CK), a través del detector de flancos 190, retiene los 18 bits superiores (D[35:18]) del bus de datos de 36 bits en el biestable 150 para transmitir los 18 bits superiores sobre el bus de salida.

La figura 6 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase mostrado en las figuras 2A a 2B para datos anticipados y un desfase en el peor de los casos (180 grados) entre el par de relojes maestros y el par de relojes de datos. Esta es la condición de desfase del peor de los casos. En el instante de tiempo 1000, los datos son válidos en el bus de datos A. En el instante de tiempo 1001, el flanco de subida de K#_CLK abre el circuito de retención transparente 120 y se transfieren datos válidos desde el bus de datos A al bus de datos B. En el instante de tiempo 1001, el flanco de subida de C#_CLK (por medio del reloj de salida retardado por la señal) abre el circuito de retención transparente 130 y los datos válidos se transfieren desde el bus de datos B al bus de datos C. Al mismo tiempo, K#_CLK realiza una transición al estado bajo, el cual retiene los datos del bus de datos B en el circuito de retención 120. En el instante de tiempo 1003, el siguiente flanco de caída de C#_CLK cierra el circuito de retención 130 y retiene los datos en el bus de datos C.

En el instante de tiempo 1004, el flanco de subida de C#_CLK (y su señal derivada enganchada por DLL DLL_CK#) a través del detector de flancos 190, retiene los 18 bits inferiores (D[17:0]) del bus de datos de 36 bits en el biestable de tipo D 150 para transmitir los 18 bits inferiores sobre el bus de salida.

En el instante de tiempo 1005, el flanco de subida de C_CLK (y su señal derivada enganchada por DLL DLL_CK), a través del detector de flancos 190, retiene los 18 bits superiores (D[35:18]) del bus de datos de 36 bits en el biestable de tipo D 150 para transmitir los 18 bits superiores sobre el bus de salida.

La figura 7 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase representado en las figuras 2A a 2B para datos tardíos y un desfase del peor de los casos entre el par de relojes maestros y el par de relojes de datos. En el instante de tiempo 1100, los datos son válidos en el bus de datos A en la entrada del circuito de retención 120. Cuando el circuito de retención 120 se abre debido al “1” lógico en K#_CLK, los datos en el bus de datos A se transmiten a través del circuito de retención 120 sobre el bus de datos B.

En el instante de tiempo 1101, el “0” lógico en K#_CLK cierra el circuito de retención 120, y los datos sobre el bus de datos A se almacenan en el circuito de retención 120 y son transmitidos al bus de datos B. El “1” lógico en C#_CLK abre el circuito de retención 130 y los datos sobre el bus de datos B son transmitidos al bus de datos C.

En el instante de tiempo 1102, el “0” lógico en C_CLK cierra el circuito de retención 130, y los datos sobre el bus de datos B son almacenados en el circuito de retención 130 y transmitidos sobre el bus de datos C. El “1” lógico en K#_CLK abre el circuito de retención 120 y los datos sobre el bus de datos A son transmitidos al bus de datos B.

En el instante de tiempo 1103, el flanco de subida de C#_CLK (y su señal derivada enganchada por DLL DLL_CK#), a través del detector de flancos 190, retiene los 18 bits inferiores (D[17:0]) del bus de datos de 36 bits en el biestable de tipo D 150 para transmitir los 18 bits inferiores sobre el bus de salida.

En el instante de tiempo 1104, el flanco de subida de C_CLK (y su señal derivada enganchada por DLL DLL_CK), a través del detector de flancos 190, retiene los 18 bits superiores (D[35:18]) del bus de datos de 36 bits C en el biestable de tipo D 150 para transmitir los 18 bits superiores sobre el bus de salida.

La figura 8 es un diagrama de temporización que ilustra la relación entre los datos y relojes en el circuito de compensación de desfase mostrado en las figuras 2A a 2B con datos anticipados y sin desfase entre el par de relojes maestros y el par de relojes de datos. En el instante de tiempo 1200, los datos son válidos en el bus de datos A en la entrada del circuito de retención 120. En el instante de tiempo 1201, el “1” lógico en K#_CLK abre el circuito de retención 120, y los datos sobre el bus de datos A son transmitidos a través del circuito de retención 120 al bus de datos B. Además, en el instante de tiempo 1201, el “1” lógico sobre C#_CLK abre el circuito de retención 130 y los datos sobre el bus de datos B son transferidos a través del circuito de retención 130 al bus de datos C.

En el instante de tiempo 1202, el “0” lógico sobre K#_CLK cierra el circuito de retención 120, y los datos sobre el bus de datos A son almacenados en el circuito de retención 120 y transmitidos sobre el bus de datos B. Además, el “0” lógico en C#_CLK cierra el circuito de retención 130, y los datos sobre el bus de datos A son almacenados en el circuito de retención 130 y transmitidos al bus de datos C.

En el instante de tiempo 1203, el flanco de subida de C#_CLK (y su señal derivada enganchada por DLL DLL_CK#), a través del detector de flancos 190, retiene los 18 bits inferiores (D[17:0]) del bus de datos de 36 bits en el biestable de tipo D 150 para transmitir los 18 bits inferiores sobre el bus de salida.

En el instante de tiempo 1204, el flanco de subida de C_CLK (y su señal derivada enganchada por DLL DLL_CK), a través del detector de flancos 190, retiene los 18 bits superiores D (D[35:18]) del bus de datos de 36 bits en el biestable de tipo D 150 para transmitir los 18 bits superiores sobre el bus de salida. Tal como se ha descrito, los dos circuitos de retención 120, 130 están abiertos durante el mismo periodo de tiempo (instante de tiempo 1201 a instante de tiempo 1202) y los datos se transfieren a través del circuito de retención 120 y 130 a medida que se reciben desde el bus de datos A al bus de datos C mientras K#_CLK y C#_CLK están ambos en estado alto.

Puede apreciarse que la invención permite un desfase amplio (0 grados a 180 grados) entre los relojes K y C. Los datos válidos que llegan de forma tardía o anticipada con respecto al flanco de subida de K#_CLK se transfieren desde un dominio de reloj al otro dominio de reloj pasando por un amplio desfase entre los relojes.

La figura 9 es un esquema de una forma de realización de uno cualquiera de los circuitos de retención transparentes 120, 130 representados en las figuras 2A a 2B. Mientras la señal de control 420 es “1” lógico, el circuito de retención 120, 130 está abierto y los datos recibidos en la entrada 412 se transfieren directamente a la salida 414. Mientras la señal de control es “1” lógico, el circuito de retención está cerrado y los datos de entrada almacenados, retenidos al producirse la transición de la señal de control desde el “1” lógico al “0” lógico, se transfieren a la salida 414.

El circuito de retención transparente incluye puertas de transmisión 400, 402. Tal como es bien conocido por los expertos en la materia, una puerta de transmisión incluye un transistor PMOS y un transistor NMOS acoplados de tal manera que los dos transistores están en ON u OFF en función del estado de una señal de control acoplada a las puertas de los transistores. Mientras los dos transistores están en OFF, el circuito de retención está cerrado y no se transmiten datos a través de la puerta de transmisión. Mientras los dos transistores están en ON, el circuito de retención está abierto y se transmiten datos a través de la puerta de transmisión.

Solamente una de las puertas de transmisión 400, 402 está abierta al mismo tiempo. La puerta de transmisión 402 está abierta mientras la señal de control es un “1” lógico y está cerrada mientras la señal de control es un “0” lógico. La puerta de transmisión 400 está abierta mientras la señal de control es un “0” lógico y está cerrada mientras la señal de control es un “1” lógico.

Mientras la puerta de transmisión 402 está abierta, la puerta de transmisión 400 está cerrada. Los datos recibidos en el puerto de entrada 412 se transmiten a través de la puerta de transmisión 402, y a través de inversores 408, 410 al puerto de salida 414. Los datos transmitidos a través del inversor 408 se transmiten también a través del inversor 416 a la entrada de la puerta de transmisión 400. Mientras la señal de control es un “0” lógico, la puerta de transmisión 402 está cerrada, los datos recibidos en el puerto de entrada 412 no se pueden transmitir al puerto de salida 414. En cambio, debido a que la puerta de transmisión 400 está abierta, los datos presentes en la entrada del inversor 416 en el momento en el que el estado de la señal de control cambia de “1” lógico a “0” lógico se transmiten a través de la puerta de transmisión 400, el inversor 408 y 410 al puerto de salida 414. De este modo, los últimos datos recibidos a través del puerto de entrada mientras la señal de control es un “1” lógico se almacenan (retienen) en el circuito de retención mientras la señal de control es un “0” lógico y son transmitidos a través del puerto de salida 414.

La figura 10 es un esquema del detector de reloj 240 representado en las figuras 2A a 2B. En la forma de realización representada, el detector de reloj incluye cuatro circuitos de retención (biestables) de tipo D 501, 502, 503, 504 conectados en serie. La entrada D del circuito de retención 501 está unida a $V_{DD}$ y las entradas de reinicialización de todos los circuitos de retención están conectadas a una señal de reinicialización RSTB. La señal de salida 202 de detección de reloj se obtiene a la salida del circuito de retención 504.

La fijación de la señal de reinicialización RSTB a “0” lógico reinicializa la totalidad de los circuitos de retención 501,

502, 503, 504. Después de la reinicialización, las salidas Q de cada circuito de retención 501, 502, 503, 504 se fijan al “0” lógico, incluyendo la salida Q del circuito de retención 504, la señal de salida de detección de reloj 202.

El detector de reloj 240 detecta si existe una señal de reloj en el par de relojes de datos. En la forma de realización representada, la señal C_CLK está acoplada a las entradas de reloj de los circuitos de retención. No obstante, las entradas de reloj de los circuitos de retención 501, 502, 503, 504 se pueden conectar a cualquiera de las señales del par de relojes de datos, es decir, a la señal C_CLK o C#_CLK. El detector de reloj 240 indica que ha detectado un reloj de datos válidos después de detectar cuatro flancos de subida en C_CLK.

El circuito de retención 501 detecta el primer flanco de subida de C_CLK. Con la entrada D conectada a $V_{DD}$, se retiene un “1” lógico en 501, y la salida Q 506 del circuito de retención 501 cambia del “0” lógico al “1” lógico. En el segundo flanco de subida de C_CLK, el “1” lógico en la entrada D del circuito de retención 502 es retenido por el circuito de retención 502, y la salida Q 507 del circuito de retención 502 cambia del “0” lógico al “1” lógico.

En el tercer flanco de subida de C_CLK, el “1” lógico en la entrada D del circuito de retención 503 es retenido por el circuito de retención 503, y la salida Q 508 del circuito de retención 503 cambia del “0” lógico al “1” lógico. En el cuarto flanco de subida de C_CLK, el “1” lógico en la entrada D del circuito de retención 504 es retenido por el circuito de retención 504 y la salida Q 505 del circuito de retención 504 cambia del “0” lógico al “1” lógico.

Después de detectarse cuatro flancos de subida en C_CLK, la salida de detección de reloj se fija a un “1” lógico indicando que existe una señal de reloj en el par de relojes de datos, y la totalidad de la salida de datos se sincronizará con el par de relojes de datos. La señal de salida de detección de reloj permanece fijada a un “1” lógico hasta que se detecte una señal de reinicialización.

La figura 11 es un diagrama de bloques de uno cualquiera de los bucles de enganche por retardo 210, 230 representados en las figuras 2A a 2B. El bucle de enganche por retardo 210, 230 incluye un detector de fase 600, una bomba de carga 602, una línea de retardo controlada por voltaje 604 y un trayecto de realimentación con retardo de inserción 606.

El detector de fase 600 detecta la diferencia de fase entre el reloj de entrada y el reloj de salida. Cuando se detecta una diferencia de fase, el detector de fase indica la diferencia de fase accionando las señales apropiadas de subir/bajar en la salida del detector de fase 600. Las señales de subir/bajar están acopladas a una bomba de carga 602. La bomba de carga 602 aumenta o reduce el voltaje de control 608 para una línea de retardo controlada por voltaje de manera apropiada para modificar el retardo añadido al reloj de entrada con el fin de minimizar la diferencia de fase.

El retardo se añade al reloj de entrada basándose en la diferencia de fase detectada entre el reloj de entrada y el reloj de salida. El retardo se añade también basándose en un retardo de inserción conocido mediante el trayecto de la alimentación con el circuito de retardo de inserción 606.

El trayecto de realimentación con retardo de inserción 606 incluye retardos de réplica para garantizar que el reloj de salida de DLL está enganchado de forma precisa al par de relojes seleccionado (C, C# o K, K#) según se muestra en la figura 3. El retardo de réplica duplica los componentes y trayectos que producen el retardo de inserción 231, 232, 233, 234 (figuras 2A a 2B) entre el pin de entrada (C, C# o K, K#) y el lugar en el que la señal de reloj (C_CLK, C#_CLK o K_CLK, K#_CLK) se usa en el dispositivo. El retardo de réplica es un grupo de circuitos que son una réplica exacta del retardo de inserción. Por ejemplo, el retardo de réplica incluye los mismos componentes, tales como los transistores con la misma distribución y configuración. Además, en el retardo de réplica se usan las mismas anchuras y longitudes del cableado.

Tal como se ha descrito haciendo referencia a la figura 3, la señal de reloj de entrada en la entrada al DLL tiene un retardo de inserción con respecto a la señal de reloj recibida en el pin de entrada del dispositivo. La línea de retardo controlada por voltaje 604 retarda el reloj de entrada en prácticamente un periodo de reloj completo y genera un reloj de salida. El reloj de salida está acoplado al trayecto de realimentación con el retardo de inserción 606. El retardo de inserción de réplica retarda el reloj de salida. El detector de fase 600 compara el reloj de entrada con el reloj de salida retardado (reloj de realimentación) y ajusta la bomba de carga 602. El DLL continúa ajustando la línea de retardo controlada por voltaje 604 hasta que el reloj de realimentación y el reloj de entrada están en fase. El reloj de salida obtenido a la salida del DLL es el reloj de entrada menos el retardo de inserción. El DLL es estable cuando el reloj de entrada y el reloj de realimentación están en fase. Después de ajustar la diferencia de fase y el retardo de inserción, el reloj de salida está alineado o bien con el reloj K# o bien con el reloj C# según se reciban en el pin del dispositivo.

Haciendo referencia de nuevo a las figuras 2A a 2B, el DLL 210 engancha DLL_CK# con el reloj K# o el reloj C#. El DLL 220 engancha DLL_CK con el reloj K o el reloj C. Continuando con la figura 11, el trayecto de realimentación 606 en el DLL 210 crea una réplica del retardo 232, 234 (figuras 2A a 2B) para K#_CLK y C#_CLK, y el trayecto de realimentación en el DLL 220 crea una réplica del retardo 231, 233 (figuras 2A a 2B) para K_CLK y C_CLK.

La figura 12 es un esquema de una forma de realización del detector de flancos 190 y el biestable de SR 180 mostrados en las figuras 2A a 2B. El detector de flancos 190 genera un impulso positivo sobre el reloj de DDR 191 como respuesta a la detección de un flanco de subida en cualquiera de las señales del par de relojes seleccionado. En la forma de realización representada, una de las señales del par de relojes (DLL_CK#) está acoplada a una entrada de la puerta NAND 700 y a un circuito de retardo de inversión 704. La salida del circuito de retardo 704 está acoplada a la otra entrada de la puerta NAND 700. La otra señal del par de relojes (DLL_CK) está acoplada a una entrada de la puerta NAND 702.

Un flanco de subida en la entrada DLL_CK a la puerta NAND 702 genera un impulso negativo en la salida de la puerta NAND 702. La longitud del impulso depende del retardo de inversión 706. El impulso negativo en la salida de la puerta NAND 702 genera un impulso positivo en el reloj de DDR 191 y en la salida del inversor 710. De manera similar, un flanco de subida en la entrada DLL_CK# a la puerta NAND 700 genera un impulso positivo en el reloj de DLR 191.

El biestable de SR 180 está acoplado a las salidas de las puertas NAND 700, 702 en el detector de flancos 190 para fijar el estado de la señal de control al multiplexor 140 en función de si, en la salida de DDR, se va a dar salida los primeros 18 bits o los segundos 18 bits del bus de datos de 36 bits. El funcionamiento de un biestable de SR es bien conocido por los expertos en la materia. Un impulso positivo en la salida del inversor 710 como respuesta a un flanco de subida de DLL_CK# reinicializa la salida del biestable de SR a un “0” lógico. Un impulso positivo en la salida del inversor 712 como respuesta a un flanco de subida de DLL_CK fija la salida del biestable de SR a un “1” lógico.

La invención anterior se ha descrito para ser usada en un sistema embebido. La invención se aplica también a un componente discreto que funcione en un sistema con un reloj de entrada y un reloj de salida.

Aunque esta invención se ha mostrado y descrito particularmente haciendo referencia a las formas de realización preferidas de la misma, los expertos en la materia apreciarán que pueden aplicarse a la misma varios cambios en cuanto a forma y detalles sin apartarse del alcance de la invención delimitado por las reivindicaciones adjuntas.

## REIVINDICACIONES

1. Circuito de sincronización para resincronizar datos desde un reloj de entrada a un reloj de salida, estando caracterizado el circuito porque presenta:

un primer circuito de retención transparente (120) que recibe los datos y se activa mediante impulsos de reloj por medio del reloj de entrada;

un segundo circuito de retención transparente (130) que recibe datos desde el primer circuito de retención transparente y se activa mediante impulsos de reloj por medio de un reloj de salida retardado, siendo el reloj de salida retardado una versión retardada del reloj de salida; y

un circuito de retención de salida (102) que recibe datos desde el segundo circuito de retención transparente y se activa mediante impulsos de reloj por medio del reloj de salida.

2. Circuito de sincronización según la reivindicación 1, en el que el reloj de salida retardado incluye un retardo de inserción y el reloj de salida es una versión de bucle de enganche por retardo del reloj de salida retardado con el retardo de inserción eliminado.

3. Circuito de sincronización según la reivindicación 1, en el que una diferencia de fase entre el reloj de entrada y el reloj de salida es inferior o igual a ciento ochenta grados.

4. Circuito de sincronización según la reivindicación 1, en el que el reloj de entrada es un reloj K# de un par de relojes maestros y el reloj de salida es un reloj C# de un par de relojes de datos.

5. Circuito de sincronización según la reivindicación 1, en el que el circuito de retención de salida se activa por flancos.

6. Circuito de sincronización según la reivindicación 1, en el que se da salida a los datos desde el circuito de retención de salida a una velocidad de datos doble.

7. Circuito de sincronización según la reivindicación 1, en el que el primer circuito de retención transparente y el segundo circuito de retención transparente dejan pasar los datos recibidos cuando se encuentran abiertos y retienen unos datos últimos recibidos cuando se encuentran cerrados.

8. Circuito de sincronización según la reivindicación 7, en el que el primer circuito de retención transparente se encuentra abierto cuando el reloj de entrada es un “1” lógico y se encuentra cerrado cuando el reloj de entrada es un “0” lógico.

9. Circuito de sincronización según la reivindicación 8, en el que el segundo circuito de retención transparente se encuentra abierto cuando el reloj de salida retardado es un “1” lógico y se encuentra cerrado cuando el reloj de salida retardado es un “0” lógico.

10. Método de sincronización de datos desde un reloj de entrada a un reloj de salida, estando caracterizado el método porque:

se reciben los datos por parte de un primer circuito de retención transparente (120) activado mediante impulsos de reloj por el reloj de entrada;

se reciben datos desde el primer circuito de retención transparente por parte de un segundo circuito de retención transparente (130) activado mediante impulsos de reloj por un reloj de salida retardado, siendo el reloj de salida retardado una versión retardada del reloj de salida; y

se reciben datos desde el segundo circuito de retención transparente por parte de un circuito de retención de salida (102) activado mediante impulsos de reloj por el reloj de salida.

11. Método según la reivindicación 10, en el que el reloj de salida retardado incluye un retardo de inserción y el reloj de salida es una versión de bucle de enganche por retardo del reloj de salida retardado con el retardo de inserción eliminado.

12. Método según la reivindicación 10, en el que una diferencia de fase entre el reloj de entrada y el reloj de salida es inferior o igual a ciento ochenta grados.

13. Método según la reivindicación 10, en el que el reloj de entrada es un reloj K# de un par de relojes maestros y el reloj de salida es un reloj C# de un par de relojes de datos.

14. Método según la reivindicación 10, en el que el circuito de retención de salida se activa por flancos.

15. Método según la reivindicación 10, en el que se da salida a datos desde el circuito de retención de salida a una velocidad de datos doble.

16. Método según la reivindicación 10, en el que el primer circuito de retención transparente y el segundo circuito de retención transparente dejan pasar los datos recibidos cuando se encuentran abiertos y retienen unos datos últimos recibidos cuando se encuentran cerrados.

17. Método según la reivindicación 16, en el que el primer circuito de retención transparente se encuentra abierto cuando el reloj de entrada es un “1” lógico y se encuentra cerrado cuando el reloj de entrada es un “0” lógico.

18. Método según la reivindicación 17, en el que el segundo circuito de retención transparente se encuentra abierto cuando el reloj de salida retardado es un “1” lógico y se encuentra cerrado cuando el reloj de salida retardado es un “0” lógico.

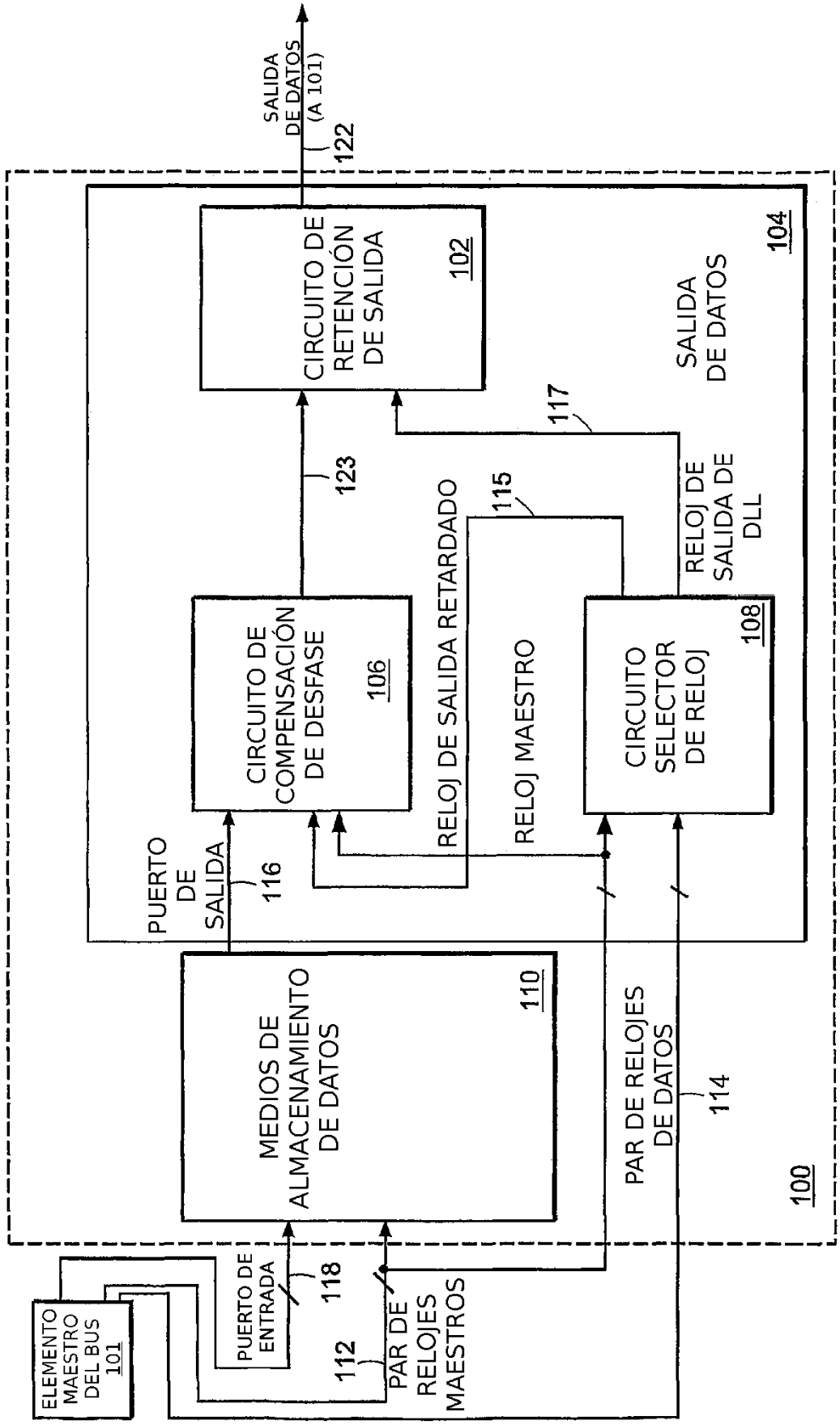
SALIDA DE DATOS (A 101)
122
CIRCUITO DE RETENCIÓN DE SALIDA
102
SALIDA DE DATOS
104
117
123
115
RELOJ DE SALIDA RETARDADO
RELOJ DE SALIDA DE DLL
CIRCUITO DE COMPENSACIÓN DE DESFASE
106
RELOJ MAESTRO
CIRCUITO SELECTOR DE RELOJ
108
PUERTO DE SALIDA
116
MEDIOS DE ALMACENAMIENTO DE DATOS
110
PAR DE RELOJES DE DATOS
114
100
ELEMENTO MAESTRO DEL BUS 101
PUERTO DE ENTRADA
118
112
PAR DE RELOJES MAESTROS

FIG. 1

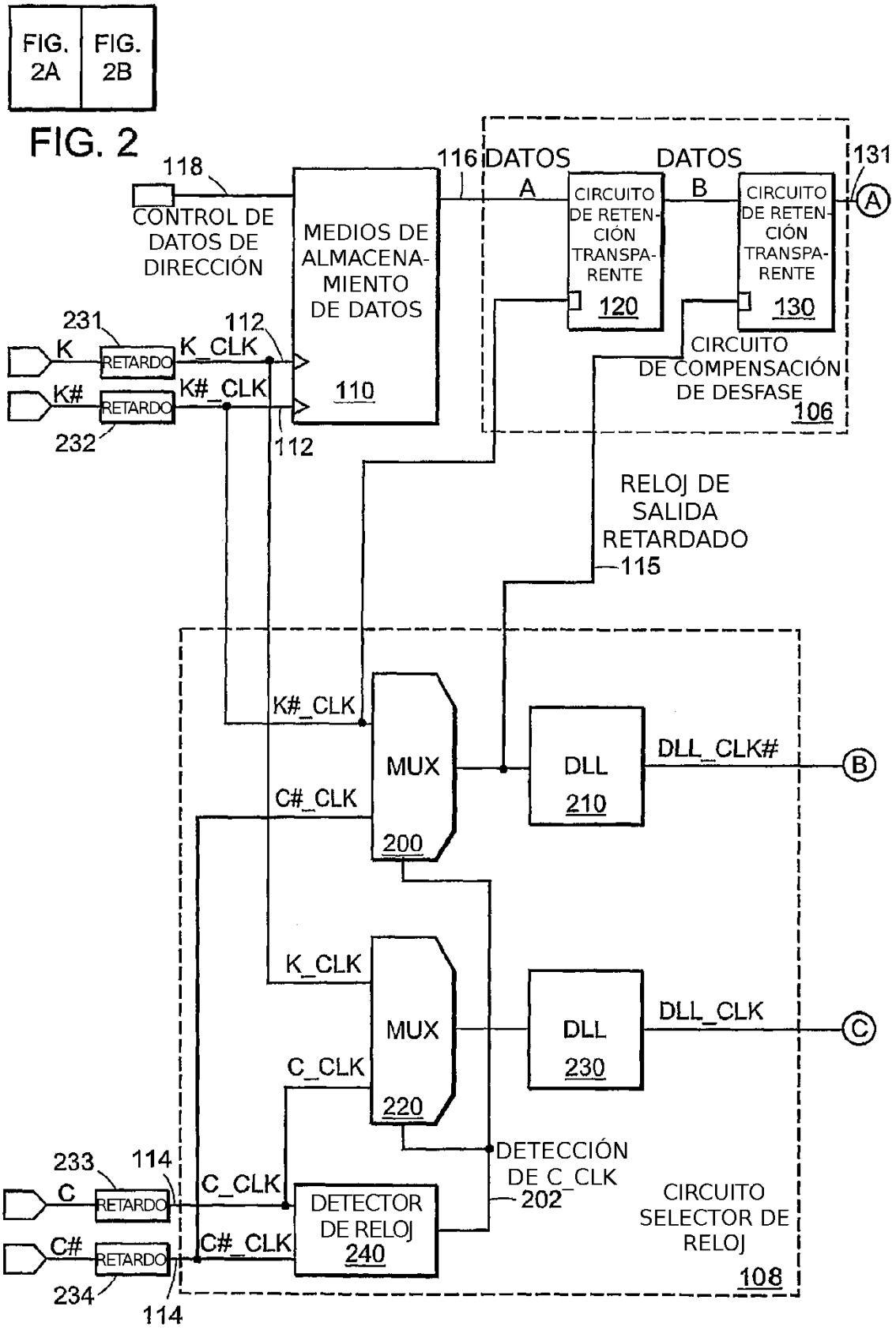
FIG. 2A
FIG. 2B
FIG. 2
118
CONTROL DE DATOS DE DIRECCIÓN
MEDIOS DE ALMACENA-MIENTO DE DATOS
110
116
DATOS A
DATOS B
CIRCUITO DE RETEN-CIÓN TRANSPA-RENTE
120
CIRCUITO DE RETEN-CIÓN TRANSPA-RENTE
130
131
A
CIRCUITO DE COMPENSACIÓN DE DESFASE
106
231
K
RETARDO
K_CLK
112
K#
RETARDO
K#_CLK
232
112
RELOJ DE SALIDA RETARDADO
115
K#_CLK
MUX
200
DLL
210
DLL_CLK#
B
C#_CLK
K_CLK
MUX
220
DLL
230
DLL_CLK
C
C_CLK
DETECCIÓN DE C_CLK
202
233
114
C
RETARDO
C_CLK
DETECTOR DE RELOJ
240
C#
RETARDO
C#_CLK
234
114
CIRCUITO SELECTOR DE RELOJ
108

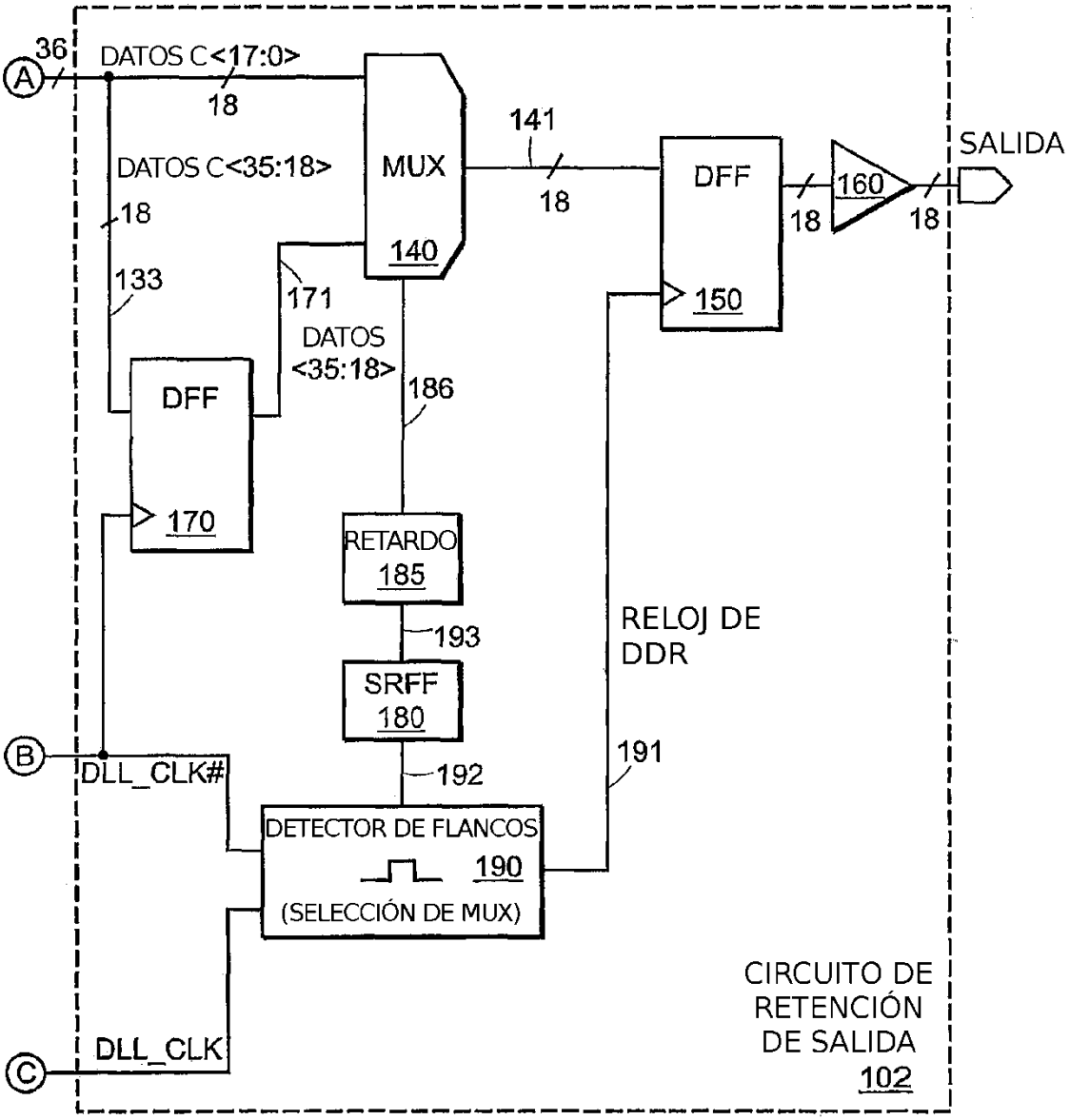
36
A
DATOS C<17:0>
18
DATOS C<35:18>
18
133
MUX
140
141
18
DFF
150
160
18
18
SALIDA
171
DATOS
<35:18>
186
DFF
170
RETARDO
185
193
SRFF
180
192
RELOJ DE
DDR
191
B
DLL_CLK#
DETECTOR DE FLANCOS
190
(SELECCIÓN DE MUX)
C
DLL_CLK
CIRCUITO DE
RETENCIÓN
DE SALIDA
102

FIG. 2B

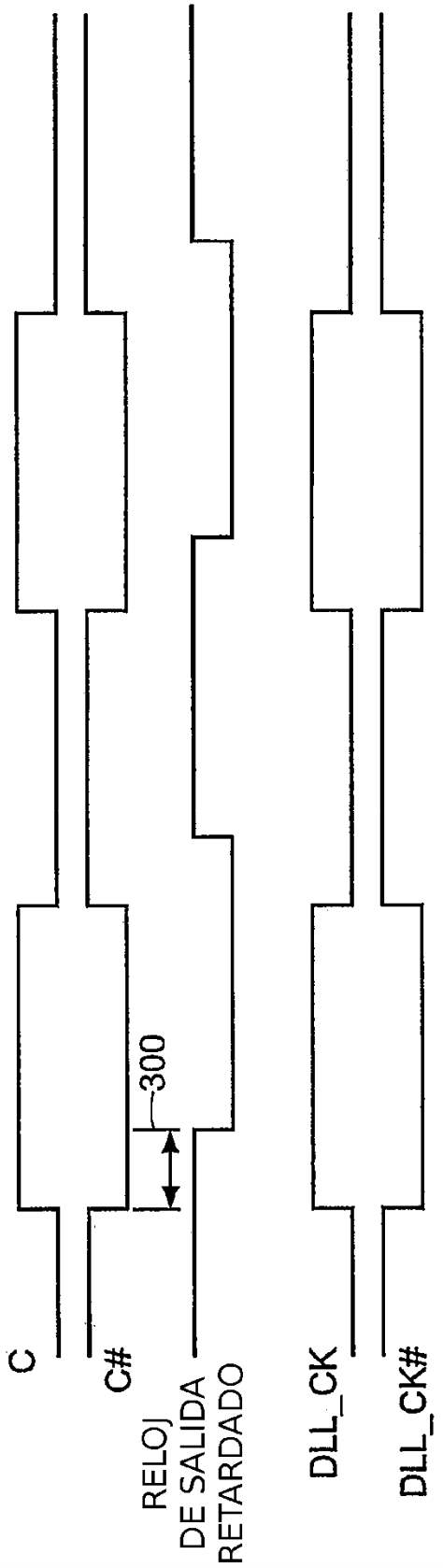


FIG. 3

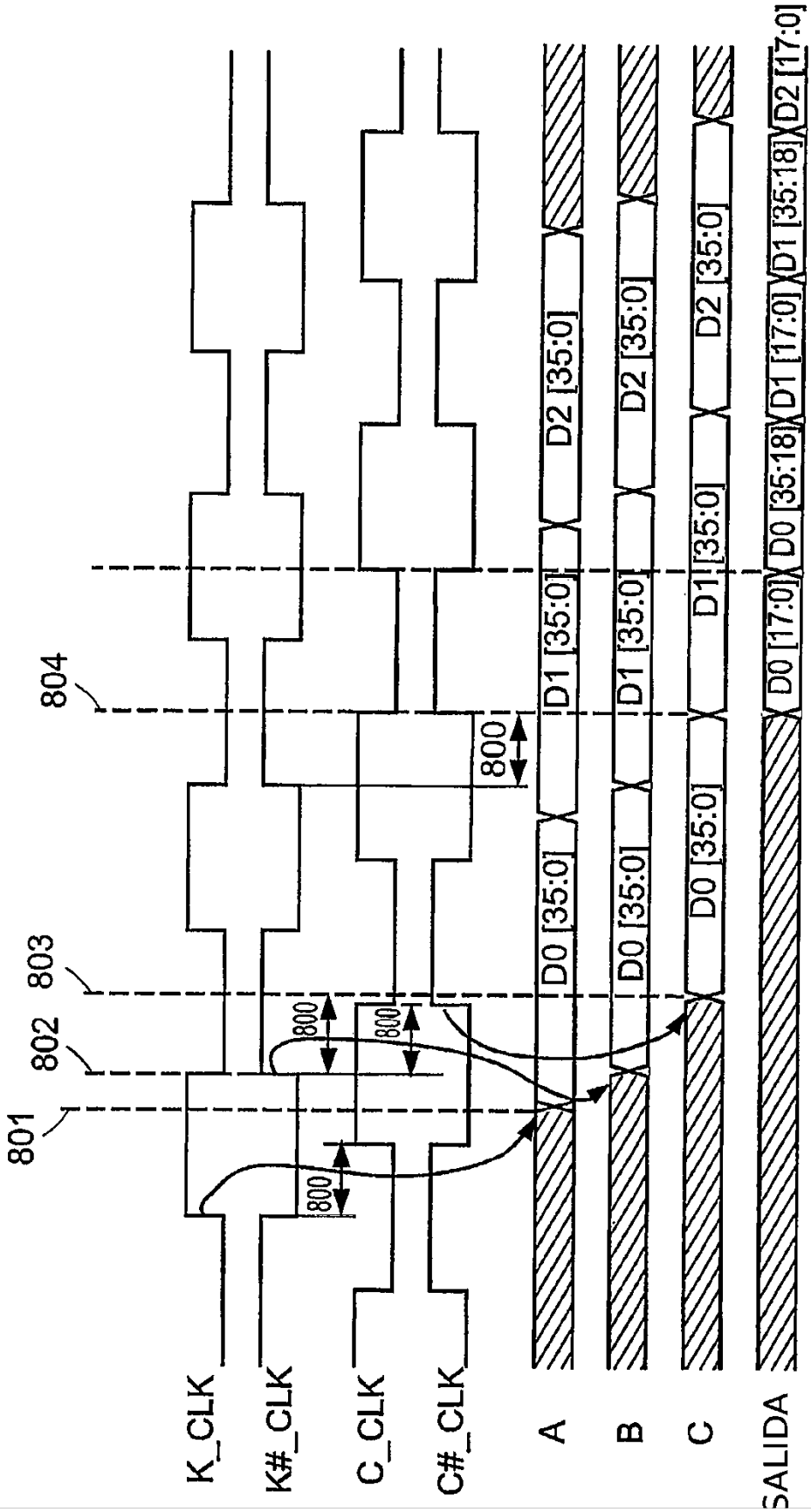
K_CLK
K#_CLK
C_CLK
C#_CLK
A
B
C
SALIDA
801
802
803
804
800
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [17:0]
D0 [35:18]
D1 [17:0]
D1 [35:18]
D2 [17:0]

FIG. 4

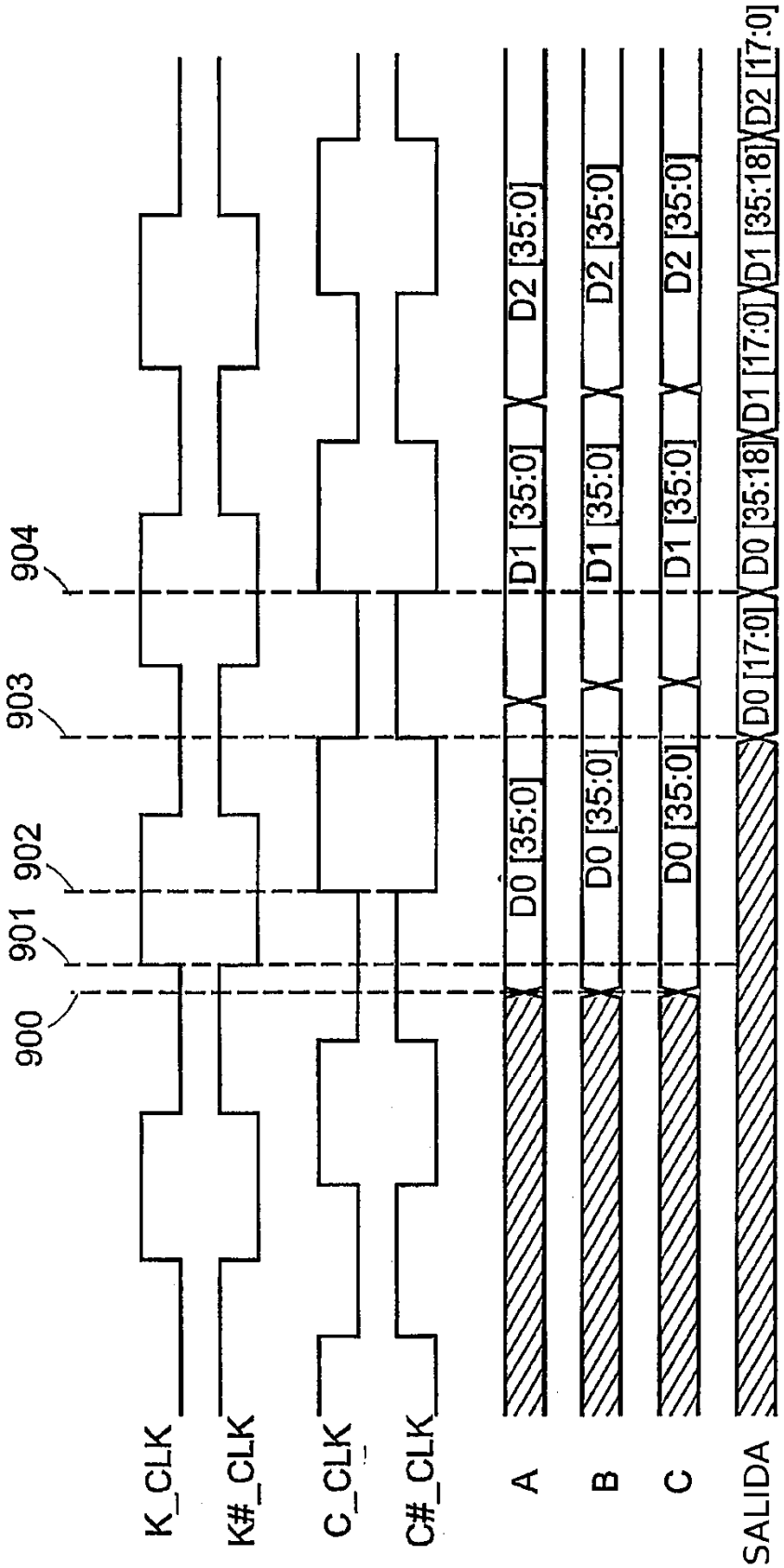
K_CLK
K#_CLK
C_CLK
C#_CLK
A
B
C
SALIDA
900
901
902
903
904
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [17:0]
D0 [35:18]
D1 [17:0]
D1 [35:18]
D2 [17:0]

FIG. 5

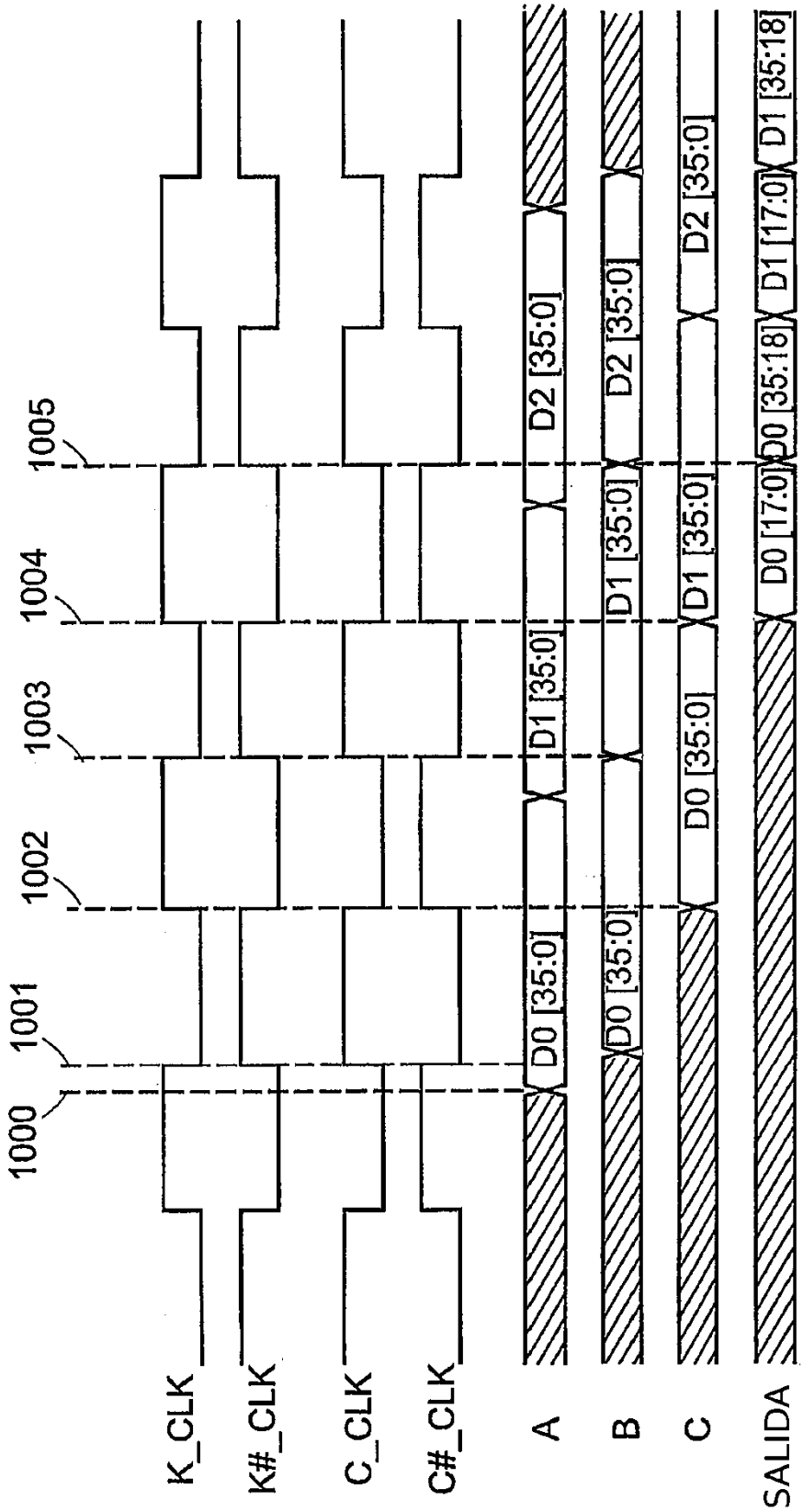
1000
1001
1002
1003
1004
1005
K_CLK
K#_CLK
C_CLK
C#_CLK
A
B
C
SALIDA
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [17:0]
D0 [35:18]
D1 [17:0]
D1 [35:18]

FIG. 6

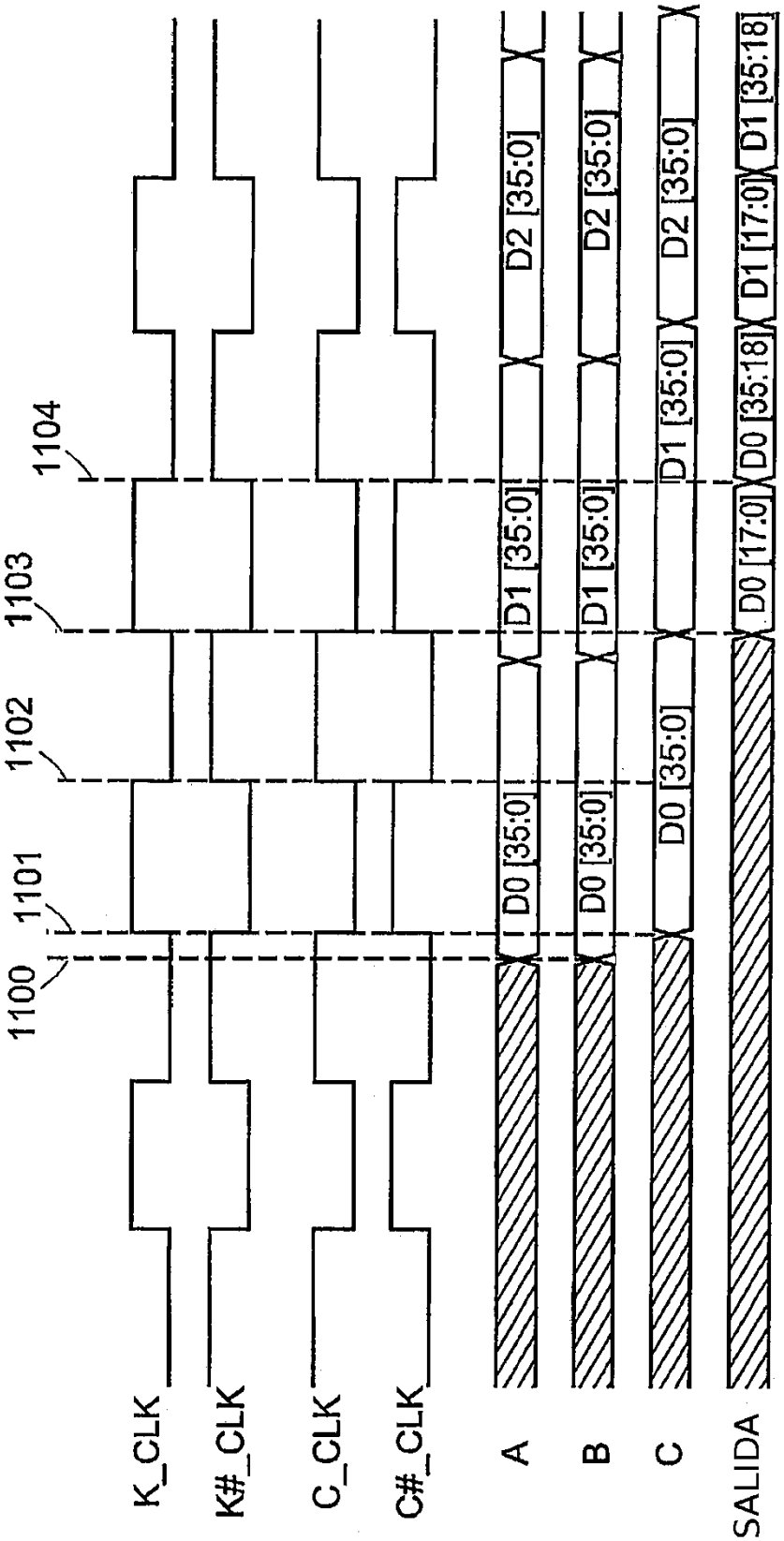
1100
1101
1102
1103
1104
K_CLK
K#_CLK
C_CLK
C#_CLK
A
B
C
SALIDA
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [17:0]
D0 [35:18]
D1 [17:0]
D1 [35:18]

FIG. 7

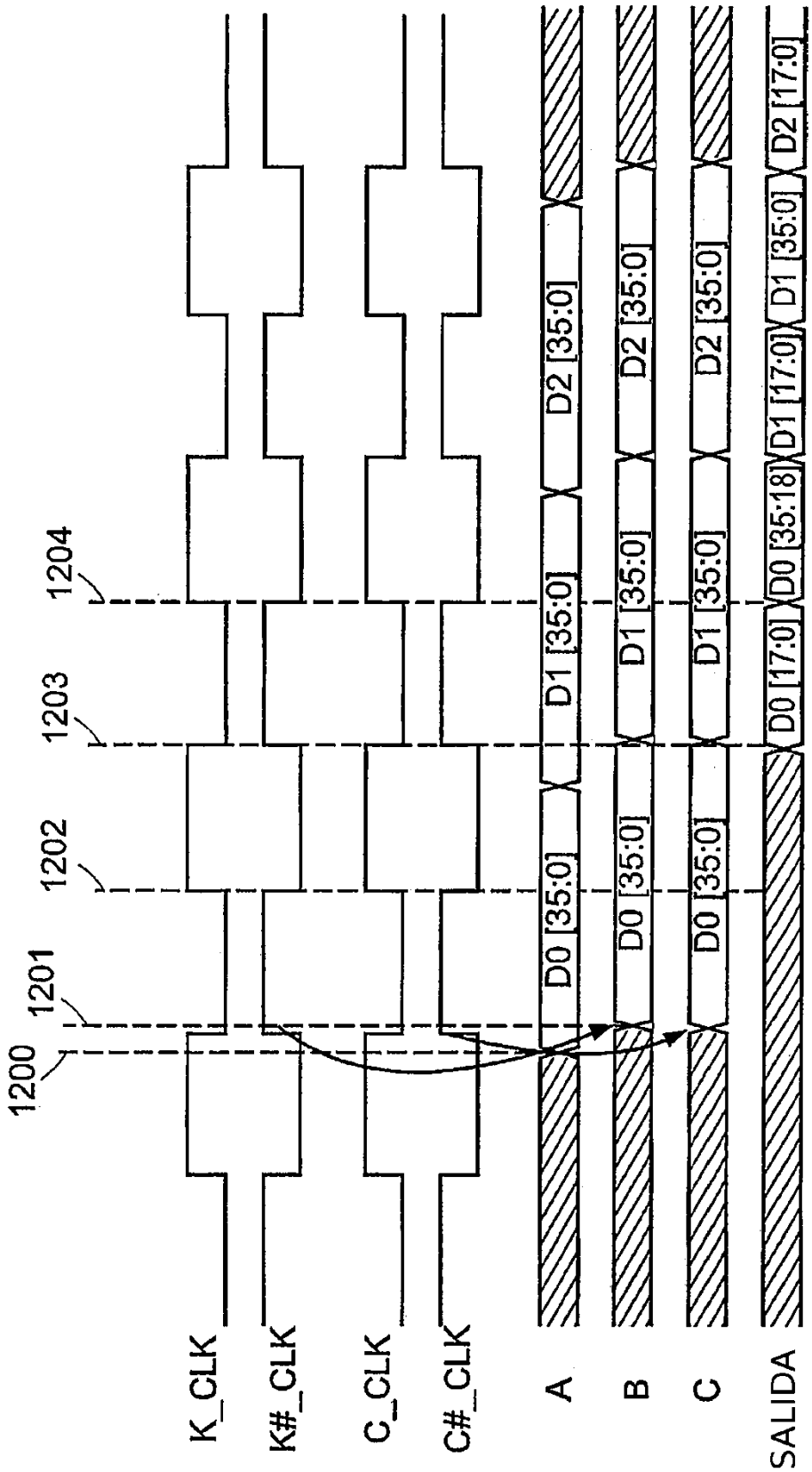
1200
1201
1202
1203
1204
K_CLK
K#_CLK
C_CLK
C#_CLK
A
B
C
SALIDA
D0 [35:0]
D1 [35:0]
D2 [35:0]
D0 [17:0]
D0 [35:18]
D1 [17:0]
D1 [35:0]
D2 [17:0]

FIG. 8

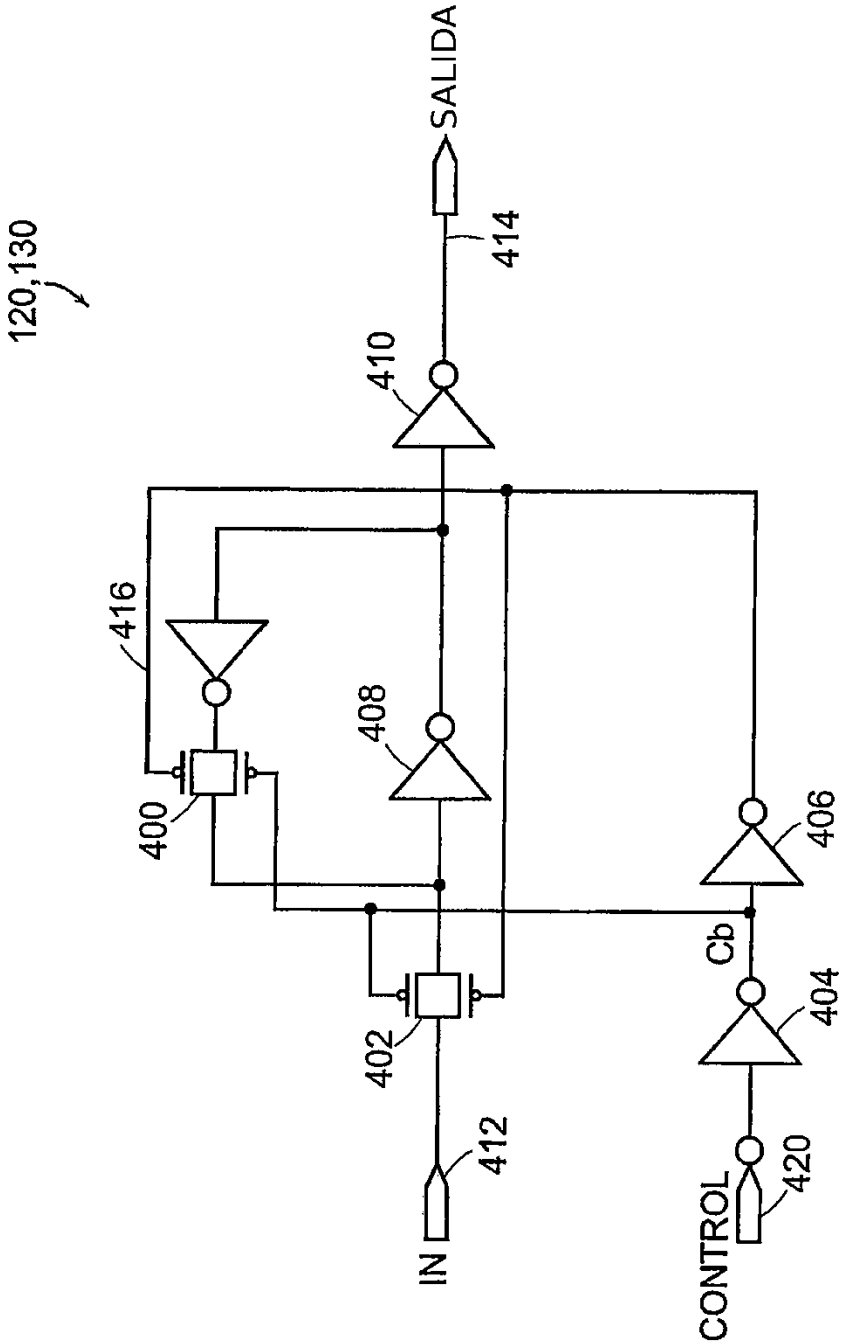
120,130
SALIDA
414
410
416
408
400
406
Cb
404
402
412
IN
CONTROL
420

FIG. 9

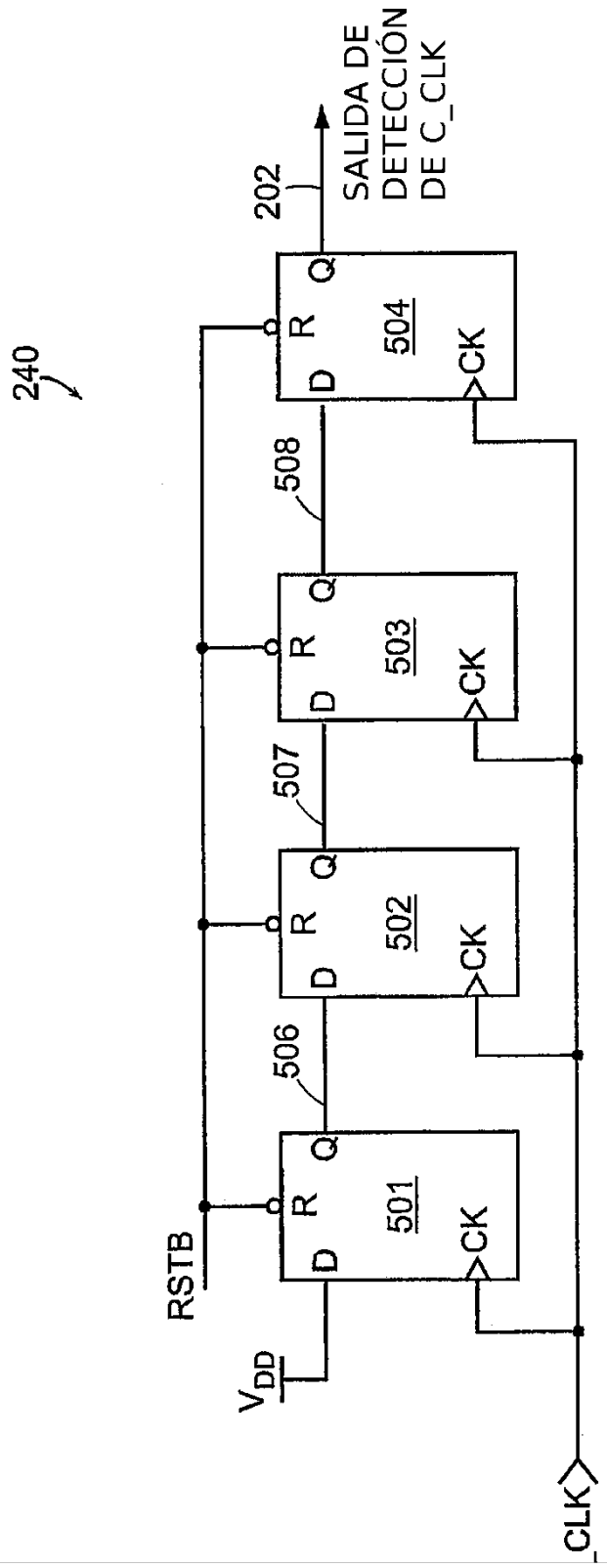
240
RSTB
$V_{DD}$
D
R
Q
CK
501
502
503
504
506
507
508
202
SALIDA DE DETECCIÓN DE C_CLK
_CLK

FIG. 10

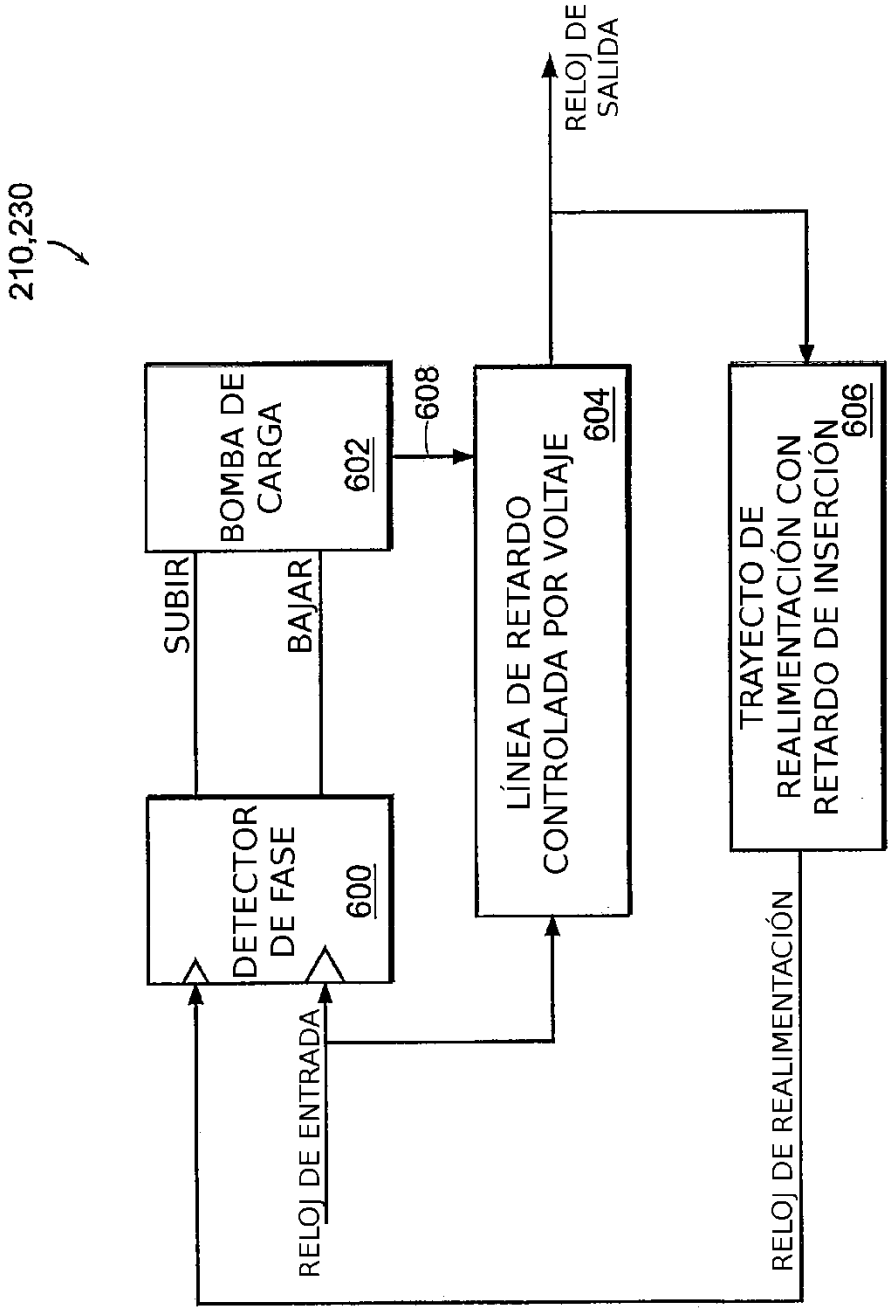
210,230
DETECTOR DE FASE
600
SUBIR
BAJAR
BOMBA DE CARGA
602
608
LÍNEA DE RETARDO CONTROLADA POR VOLTAJE
604
RELOJ DE SALIDA
TRAYECTO DE REALIMENTACIÓN CON RETARDO DE INSERCIÓN
606
RELOJ DE ENTRADA
RELOJ DE REALIMENTACIÓN

FIG. 11

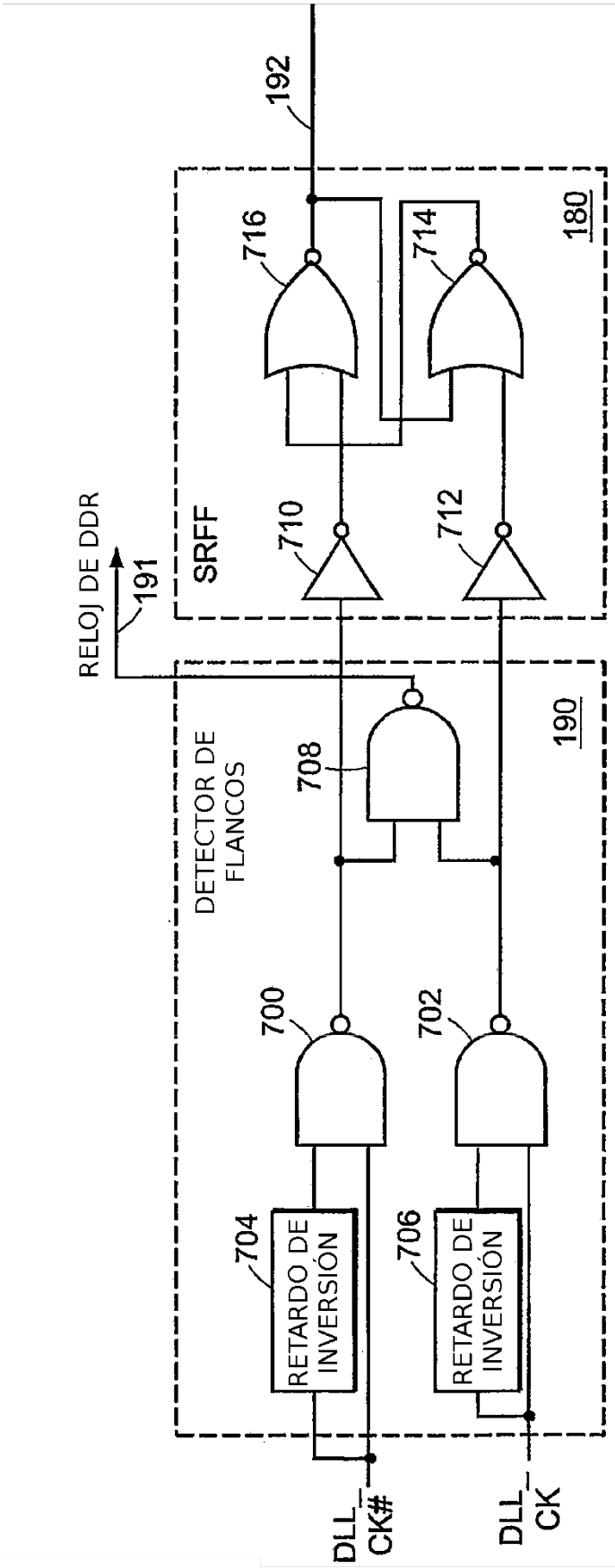
192
716
714
180
SRFF
710
712
RELOJ DE DDR
191
DETECTOR DE FLANCOS
708
190
700
702
704
RETARDO DE INVERSIÓN
706
RETARDO DE INVERSIÓN
DLL_ CK#
DLL_ CK

FIG. 12