

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 880 268**

51 Int. Cl.:

G06F 13/10 (2006.01)

G06F 13/362 (2006.01)

G06F 13/40 (2006.01)

G06F 21/62 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **27.06.2013 PCT/US2013/048110**

87 Fecha y número de publicación internacional: **10.04.2014 WO14055145**

96 Fecha de presentación y número de la solicitud europea: **27.06.2013 E 13737064 (9)**

97 Fecha y número de publicación de la concesión europea: **12.05.2021 EP 2904515**

54 Título: **Colas de mensajes en soporte físico para una comunicación dentro de un grupo de cómputo**

30 Prioridad:

02.10.2012 US 201261709142 P
27.02.2013 US 201313778307

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
24.11.2021

73 Titular/es:

ORACLE INTERNATIONAL CORPORATION
(100.0%)

500 Oracle Parkway, Mailstop 5OP7
Redwood Shores, CA 94065, US

72 Inventor/es:

AINGARAN, KATHIRGAMAR;
BRIDGE, WILLIAM H. JR.;
SWART, GARRET F.;
JAIRATH, SUMTI y
JOHNSON, JOHN G.

74 Agente/Representante:

TORNER LASALLE, Elisabet

ES 2 880 268 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

Colas de mensajes en soporte físico para una comunicación dentro de un grupo de cómputo

La presente solicitud reivindica prioridad con respecto a la solicitud de patente estadounidense nº 13/783.307, presentada el 27 de febrero de 2013, que reivindica prioridad con respecto a la solicitud provisional estadounidense nº 61/709.142, presentada el 2 de octubre de 2012.

Campo de la invención

La presente invención versa sobre el uso de circuitos especializados para enviar y recibir mensajes de uno o más nodos en un grupo de cómputo.

Antecedentes

Los sistemas de bases de datos distribuyen porciones de una base de datos entre varios nodos en un grupo para mejorar el rendimiento y aportar escalabilidad. Los grupos se configuran en una de dos maneras: como grupos de memoria coherente o como grupos de cómputo.

Los nodos de un grupo de memoria coherente comparten memoria física. Compartir memoria física permite a cada nodo del grupo comunicarse muy rápidamente. Enviando y recibiendo mensajes entre dos nodos en un grupo de memoria compartida, un nodo escribirá datos a la memoria compartida y el otro nodo leerá los datos de la memoria compartida. Desgraciadamente, los grupos de memoria coherente son muy costosos de construir y hay un límite de tamaño de la memoria compartida que puede estar almacenada en un grupo de memoria coherente.

Los nodos de un grupo de cómputo no comparten memoria física. La comunicación entre los nodos de un grupo de cómputo puede llevarse a cabo a través de mensajes habitualmente distribuidos en un subsistema de E/S. Un subsistema de E/S puede incluir un enlace de E/S, un mecanismo de sincronización y un hilo servidor en el nodo receptor para procesar los mensajes entrantes. Además, puede ser necesario que los nodos de cómputo reensamblen los mensajes entrantes y almacenen los mensajes reensamblados en la memoria principal de un nodo. Tal subsistema de E/S incurre en una penalización de rendimiento que aumenta la latencia de los mensajes y limita el caudal de procesamiento de mensajes.

Brewer et al.: "Remote Queues: Exposing Message Queues for Optimization and Atomicity", 7th annual ACM Symposium on Parallel Algorithms and Architectures, Santa Bárbara, 7 de julio de 1995, vol. Simp. 6, páginas 42-53, da a conocer una pluralidad de nodos en una red, teniendo cada nodo un procesador de cómputo que está soportado por un procesador de mensajes. Cuando ha de despacharse un mensaje, el procesador de cómputo de un nodo de despacho solicita al procesador de mensajes del nodo de despacho que transmita el mensaje al procesador de mensajes de un nodo receptor que pone en cola el mensaje recibido. En respuesta a una solicitud de interrogación del procesador de cómputo del nodo receptor, el procesador de mensajes del nodo receptor extrae de la cola el mensaje recibido y proporciona el mensaje al procesador de cómputo.

El documento EP 2 423 843 A1 versa sobre la configuración o programación de una FPGA. El documento menciona técnicas convencionales para configurar o programar una FPGA que incluyen cifrar los datos de configuración y almacenar los datos cifrados en una unidad de memoria dentro de la FPGA o en un dispositivo de almacenamiento externo conectado operativamente con la FPGA. El documento divulga, además, el uso de un protocolo complejo de Menezes-Qu-Vanstone de curvas elípticas (protocolo EC-MQV, por sus siglas en inglés) como protocolo de acuerdo de claves para calcular una clave de sesión.

Los planteamientos descritos en esta sección son planteamientos que podrían seguirse, pero no necesariamente planteamientos que se hayan concebido o seguido con anterioridad. Por lo tanto, a no ser que se indique algo distinto, no debería suponerse que cualquiera de los planteamientos descritos en esta sección pueda ser considerado técnica anterior meramente en virtud de su inclusión en esta sección.

Sumario de la invención

La invención está definida por las reivindicaciones independientes, teniendo debidamente en cuenta cualquier elemento que sea equivalente a un elemento especificado en las reivindicaciones. Las reivindicaciones dependientes se refieren a características opcionales de algunos elementos de la invención.

Breve descripción de los dibujos

La presente invención está ilustrada a título de ejemplo, y no de limitación, en las figuras de los dibujos adjuntos, y en las cuales números de referencia semejantes se refieren a elementos similares y en las cuales:

la Figura 1 es un diagrama de bloques que ilustra un nodo en un sistema de múltiples HMQ que emplea un coprocesador de HMQ para recibir mensajes de otros nodos del grupo de cómputo, según una realización de la invención;

la Figura 2 es un diagrama de bloques de una RKEY de HMQ, según una realización;

la Figura 3 es un diagrama de bloques de una cola de mensajería en soporte físico, según una realización de la invención;

la Figura 4 es un diagrama de flujo que ilustra la inicialización de una HMQ, según una realización;

la Figura 5 es un diagrama de flujo que ilustra la autenticación de una instrucción, según una realización;

la Figura 6 es un diagrama de flujo que ilustra la puesta en cola de forma atómica de un mensaje en una HMQ, según una realización;

la Figura 7 es un diagrama de flujo que ilustra la lectura de un mensaje almacenado en una entrada particular y la extracción de la entrada de la cola, según una realización;

la Figura 8 ilustra un sistema informático en el que se puede implementar una realización.

Descripción detallada

En la siguiente descripción, con fines explicativos, se presentan numerosos detalles específicos para proporcionar una comprensión cabal de la presente invención. Sin embargo, será evidente que la presente invención puede ser puesta en práctica sin estos detalles específicos. En otros casos, se presentan estructuras y dispositivos muy conocidos en forma de diagrama de bloques para evitar ofuscar innecesariamente la presente invención.

Visión general

La comunicación entre nodos en un grupo de cómputo que no comparten memoria física se lleva a cabo pasando mensajes a través de un subsistema de E/S. La tara para cada nodo que usa el subsistema de E/S es elevada. Cada nodo puede incluir un mecanismo de sincronización, un hilo en el nodo receptor preparado para recibir conexiones y otros hilos para procesar y reensamblar mensajes. Frecuentemente, un nodo receptor también mantiene una cola separada en memoria para cada nodo en el subsistema de E/S que envía mensajes al nodo receptor. Tal tara conlleva una penalización del rendimiento, que aumenta la latencia y limita el caudal de procesamiento de mensajes.

Se presenta un sistema eficiente para el envío y la recepción de mensajes entre nodos en un grupo de cómputo al que se denomina en el presente documento sistema de múltiples HMQ. El sistema de múltiples HMQ proporciona funcionalidad para las copias de memoria y el paso de mensajes. Cada nodo en el sistema de múltiples HMQ incluye una cola de mensajes en soporte físico ("HMQ", por sus siglas en inglés), un coprocesador de colas de mensajes en soporte físico ("coprocesador de HMQ") y un enlace de comunicaciones que conecta cada coprocesador de HMQ a los otros coprocesadores de HMQ en el sistema de múltiples HMQ.

Una HMQ comprende una cabecera y una carga útil. La cabecera en una HMQ comprende metadatos relativos al estado de la HMQ y punteros en entradas o direcciones particulares de la carga útil. El coprocesador de HMQ almacena entradas de la carga útil en función de las instrucciones que recibe el coprocesador de HMQ. Cada entrada de la carga útil también puede contener indicadores relativos al estado de esa entrada particular.

Los coprocesadores de HMQ envían instrucciones a otros coprocesadores de HMQ y reciben instrucciones procedentes de los mismos. El coprocesador receptor de HMQ asigna una entrada en la HMQ diana incrementando atómicamente un puntero de asignación en la cabecera de la HMQ. La entrada asignada se llena con un mensaje entrante y se pone un indicador especial de finalización en la entrada. El coprocesador de HMQ es capaz de enviar, recibir, autenticar, pone en cola, sincronizar y reensamblar mensajes entrantes sin intervención adicional de soporte lógico.

Gracias al coprocesador de HMQ, los mensajes en el sistema de múltiples HMQ se intercambian a una velocidad mayor y con menor latencia. Una mensajería más rápida permite un mayor rendimiento del sistema de múltiples HMQ y del grupo de cómputo. Además, la HMQ puede reducir el consumo de memoria en comparación con las aplicaciones genéricas de mensajería al almacenar mensajes de múltiples procedencias en la misma HMQ, eliminando la necesidad de colas por procedencia.

El coprocesador de HMQ

Los coprocesadores de HMQ son circuitos especializados. La expresión "circuitos especializados" se refiere a circuitos digitales que realizan un conjunto de funciones y que bien están cableados para realizar el conjunto de funciones o bien están programados de forma persistente para realizar el conjunto de funciones. Los circuitos programados de forma persistente incluyen circuitos integrados para aplicaciones específicas (ASIC, por sus siglas en inglés) o matrices de puertas programables *in situ* (FPGA, por sus siglas en inglés) electrónicos digitales. Los circuitos especializados no incluyen un procesador de uso general que esté configurado para realizar funciones ejecutando un conjunto de instrucciones o programa de soporte lógico que pueda residir en la memoria.

En cambio, la expresión “soporte lógico” se refiere a un conjunto de instrucciones o a un programa ejecutado por circuitos digitales que llevan a cabo un conjunto expansivo de funciones en un procesador de uso general. Un soporte lógico no está ni cableado ni programado de forma persistente para realizar funciones específicas. Aunque puede usarse un soporte lógico para implementar una funcionalidad expansiva, el soporte lógico normalmente se ejecuta con más lentitud y requiere más tara que las funciones realizadas por circuitos especializados. Por conveniencia de la expresión, cuando se dice que un soporte lógico realiza una acción, se están ejecutando instrucciones en un procesador de uso general para llevar a cabo la acción.

Los coprocesadores de HMQ procesan un conjunto limitado de instrucciones. Por ejemplo, un coprocesador de HMQ puede procesar, sin limitación de ningún tipo, las siguientes instrucciones sin ejecutar un soporte lógico: Copiar, Copiar y señalar, Llenar, Almacenar, Suma atómica, O atómica, Envío atómico de mensajes sin solicitud, Interrumpir, Sincronizar y Ausencia de operación.

Los coprocesadores de HMQ envían instrucciones a otros coprocesadores de HMQ, y reciben instrucciones de los mismos, sin la ayuda de soporte lógico. Por ejemplo, un primer coprocesador de HMQ puede ordenar a un segundo coprocesador de HMQ que almacene un mensaje. El segundo coprocesador de HMQ autentica la orden, asigna una entrada en la HMQ diana, y almacena la carga útil en la HMQ diana. La entrada asignada se llena con el mensaje entrante asociado con una instrucción y se pone un indicador especial de finalización en la entrada. El coprocesador de HMQ lleva a cabo estas operaciones sin ayuda de soporte lógico.

Como se describe en el presente documento, el coprocesador de HMQ puede, sin limitación de ningún tipo, enviar instrucciones a otros coprocesadores de HMQ, recibir instrucciones de otros coprocesadores de HMQ, autenticar las instrucciones recibidas, poner entradas en cola en una HMQ, sincronizar los mensajes recibidos, reensamblar los mensajes entrantes y notificar errores, sin intervención del soporte lógico.

El sistema de múltiples HMQ

La Figura 1 es un diagrama de bloques que ilustra un sistema 100 de múltiples HMQ, según una realización. El sistema de múltiples HMQ mueve datos de una ubicación de memoria a otra ubicación de memoria. Por ejemplo, el coprocesador 115-1 de HMQ puede ordenar al coprocesador 115-M que almacene un mensaje en memoria 120-M en el nodo 105-M. En la realización ilustrada en la Figura 1, el sistema 100 de múltiples HMQ incluye, sin limitación de ningún tipo, los nodos 105-1 a 105-M (colectivamente, “nodos 105”), incluyendo los nodos 105 los coprocesadores 115-1 a 115-M de HMQ (colectivamente, “coprocesadores 115 de HMQ”), respectivamente.

En la realización ilustrada en la Figura 1, los coprocesadores 115 de HMQ están situados en los propios chips de los procesadores 110-1 a 110-M (colectivamente, “procesadores 110”), respectivamente. En otra realización, los coprocesadores 115 de HMQ son núcleos situados en los procesadores 110, respectivamente. El coprocesador 115 de HMQ puede estar situado en la misma placa —pero no en el mismo chip— que los procesadores 115, respectivamente. Alternativamente, los coprocesadores 115 de HMQ pueden estar situados en otro lugar y estar meramente acoplados comunicativamente con los procesadores 110, respectivamente.

El coprocesador 115-1 de HMQ está acoplado comunicativamente con: el bus 150-1; la memoria 120-1; el soporte lógico 130-1; la HMQ 140-1; y otros coprocesadores 115 de HMQ. Cada una de las HMQ 140-1 a 140-M (colectivamente, “HMQ 140”) es una instancia de la HMQ 300. Además, los coprocesadores 115 de HMQ están conmutativamente acoplados a través de los enlaces 160-1 a 160-M (denominados colectivamente “enlaces 160”), respectivamente, y del conmutador 170. Sin embargo, en otras realizaciones, los coprocesadores de HMQ pueden estar directamente conectados o acoplados comunicativamente a través de otros medios, conocidos por los expertos en la técnica.

Como se ha expuesto anteriormente, el soporte lógico 130-1 a 130-M, colectivamente 130, se refiere a un conjunto de instrucciones o programa almacenado en nodos 105, respectivamente, y ejecutado en procesadores 110 de uso general, respectivamente. El soporte lógico 130-1 a 130-M (colectivamente, “soporte lógico 130”), no está cableado o programado de forma persistente para llevar a cabo funciones específicas. Por conveniencia de la expresión, cuando se dice que el soporte lógico 130 realiza una acción, se están ejecutando instrucciones que comprenden el soporte lógico 130 en un procesador 110, respectivamente, para realizar la acción.

Claves de acceso remoto a HMQ (“RKEY de HMQ”)

El coprocesador de HMQ puede no estar limitado por un sistema operativo y puede tener un acceso ilimitado a la memoria. Puede enviarse una RKEY de HMQ con cada instrucción y designar a qué HMQ debería acceder un coprocesador de HMQ. Además, una RKEY de HMQ puede incluir la ubicación y el tamaño de la memoria asignada para la correspondiente HMQ, y restringir que el coprocesador de HMQ acceda a memoria asignada para la HMQ. Por tanto, es importante garantizar que una RKEY de HMQ sea válida y que no ha sido falsificada o alterada. Para garantizar que el coprocesador de HMQ escribe en un espacio autorizado en memoria, el coprocesador de HMQ valida la RKEY de HMQ enviada con cada instrucción. Por ejemplo, puede crearse una RKEY de HMQ para cada HMQ en cada nodo del sistema de múltiples HMQ. Cada una de las múltiples HMQ asignadas en el mismo nodo puede tener una RKEY única de HMQ, que designa a la HMQ particular a la que debería acceder el coprocesador de HMQ para una instrucción dada.

La Figura 2 es un diagrama de bloques de la RKEY 200 de HMQ, según una realización. La RKEY 200 puede incluir, sin limitación de ningún tipo, la clave irreversible 210, el tamaño 220, la dirección 230 y permisos 240.

La RKEY 200 de HMQ comprende un certificado otorgado por el propietario de un bloque de memoria física a un usuario remoto. Por ejemplo, usando la realización ilustrada en la Figura 1, el soporte lógico 130-M puede autorizar que el nodo 105-1 acceda a un bloque específico en la memoria 120-M enviando al nodo 105-1 una instancia de la RKEY 200 de HMQ, generada por el soporte lógico 130-M. Cuando el coprocesador 115-1 de HMQ envía una instrucción al coprocesador 115-M de HMQ, el coprocesador 115-1 de HMQ también puede enviar una copia de la anterior instancia recibida de la RKEY 200 de HMQ. El coprocesador 115-M de HMQ puede validar la RKEY de HMQ para garantizar que la instrucción no fue enviada desde un coprocesador autorizado de HMQ y que está siendo almacenada en una memoria debidamente asignada. Alternativamente, el soporte lógico 130-1 puede enviar una instrucción al coprocesador 115-M de HMQ directamente, o a través del coprocesador 115-1 de HMQ. El tamaño de la RKEY de HMQ puede tener una longitud de 16 bytes; sin embargo, no se requiere que la RKEY de HMQ tenga una longitud de 16 bytes. Puede ser preferible una RKEY de HMQ menor para reducir el espacio ocupado en memoria y la congestión de la red; sin embargo, una RKEY de HMQ mayor puede permitir una clave irreversible mayor y más segura.

La clave irreversible 210 es una clave irreversible cifrada derivada de un secreto básico. La clave irreversible 210 también puede derivarse, sin limitación de ningún tipo, de los parámetros: tamaño 220, dirección 230 y permisos 240. Cada uno de los coprocesadores 115 de HMQ contiene el secreto básico, generado por el soporte lógico 130-1 a 130-M, respectivamente, para descifrar la irreversible 210 tras la recepción de una instancia de la RKEY 200 de HMQ. También pueden implementarse otros modelos de seguridad que usen cifrado por clave pública/clave privada, claves irreversibles, semillas, secretos o códigos.

El tamaño 220 es el tamaño de la región a la que una RKEY de HMQ está autorizada a acceder. La dirección 230 contiene la dirección física de la región a la que el generador de RKEY de HMQ (por ejemplo, el soporte lógico 140-M) está autorizado a acceder. Alternativamente, la dirección 230 puede contener solamente un número que identifica una HMQ específica en memoria, sin que sea la dirección de memoria de la HMQ. Además, la RKEY 200 de HMQ puede identificar bloques de control del coprocesador ("CCB", por sus siglas en inglés) de HMQ en lugar de una dirección de destino.

Los permisos 240 son los permisos que se dan al presentador (por ejemplo, el coprocesador 115-1 de HMQ). Por ejemplo, la RKEY de HMQ puede dar permiso para escribir un mensaje en una HMQ específica, pero no para leer de la HMQ. En una realización, los permisos 240 pueden incluir, sin limitación de ningún tipo, la capacidad de lectura y escritura y la actualización de la memoria intermedia.

La cola de mensajes en soporte físico ("HMQ")

La Figura 3 es un diagrama de bloques de una HMQ 300, según una realización. La HMQ 300 comprende una cabecera 310 y una carga útil 330. Aunque la Figura 3 ilustra una realización, otras realizaciones pueden omitir, reordenar, modificar cualquiera de los elementos mostrados y/o añadir a los mismos. La cabecera 310 comprende metadatos relativos al estado de la HMQ 300 y apunta a direcciones particulares dentro de la carga útil 330. La cabecera 310 puede incluir, sin limitación de ningún tipo, el tamaño de la carga útil 330, el número de entradas 332-1 a 332-N (denominadas colectivamente entradas 332) y el tamaño de cada entrada 332-1 a 332-N. La cabecera 310 también puede incluir, sin limitación de ningún tipo, un indicador 311 de habilitación, un indicador 312 de falta de preparación, un indicador 313 de error, un campo reservado 314, un área 315 de destinatario y un indicador 316 de interrupción. La cabecera 310 también puede incluir, sin limitación de ningún tipo, un puntero principal 320 y un puntero 322 de asignación. Por ejemplo, cada una de las HMQ 140 es una instancia de la HMQ 300. Cada uno de los coprocesadores 115 de HMQ y del soporte lógico 130-1 a 130-M, que asignan instancias de la HMQ 300, tiene acceso, respectivamente, a las HMQ 140.

El tamaño de las entradas es el tamaño de cada una de las entradas 332. En una realización, tamaño de la entrada es 64 bytes. Alternativamente, el tamaño de la entrada puede ser una potencia de dos; sin embargo, también pueden ser adecuados otros tamaños. Además, el tamaño de la carga útil y el tamaño de la entrada, almacenados en la cabecera 310, es de cuatro bytes cada uno de ambos. En otra realización, el tamaño de la entrada puede ser cero para indicar que las entradas 332 pueden ser de cualquier tamaño.

El indicador 311 de habilitación designa si una instancia de HMQ 300 está habilitada o inhabilitada. En una realización, el indicador 311 de habilitación es de un byte. Por ejemplo, si el indicador 311 de habilitación está puesto a cero, entonces la HMQ 300 está inhabilitada, pero, si no, la HMQ 300 está habilitada.

El indicador 312 de falta de preparación se copia al primer byte de una entrada particular para indicar que la entrada particular no es válida. En una realización, el indicador 312 de falta de preparación es de un byte. Por ejemplo, si se recibe una instrucción de almacenar un mensaje que supera la longitud de la entrada 332-1, entonces el valor almacenado en el indicador 312 de falta de preparación se copia al primer byte de la entrada 332-1, indicando que la entrada no es válida.

El indicador 313 de error se copia al primer byte de la entrada para indicar que hubo un error en el procesamiento de una instrucción. En una realización, el indicador 313 de error es de un byte. Por ejemplo, si la HMQ 140-1 está llena, el coprocesador 115-1 de HMQ puede poner el indicador 313 de error a 1 para indicar que no puede poner en cola ningún mensaje nuevo.

5 El campo reservado 314 almacena el desplazamiento entre la cabecera 310 y la carga útil 330. En una realización, el campo reservado 314 es de cuatro bytes y es cero si no hay desplazamiento alguno entre la cabecera 310 y la carga útil 330. Alternativamente, el campo reservado 314 es un puntero a la carga útil 330. Por ejemplo, si la cabecera 310 y la carga útil 330 no fueran contiguas, el campo reservado 314 puede comprender la dirección de la carga útil 330 en la memoria 120-1.

10 El área 315 de destinatario está disponible para ser usada por un procesador de HMQ en el mismo nodo. En una realización, el área 315 de destinatario es de 32 bytes. Por ejemplo, el coprocesador 115-1 de HMQ puede usar el área 315 de destinatario para reensamblar fragmentos de 16 bytes de un mensaje (presentado con mayor detalle *infra*).

15 El indicador 316 de interrupción indica si debería generar una interrupción tras un envío de una entrada sin solicitud realizado con éxito. En una realización, el indicador 316 de interrupción es un solo bit y está puesto a cero para indicar que debería generarse una interrupción tras envío de una entrada sin solicitud realizado con éxito. Por ejemplo, cuando el coprocesador 115-1 de HMQ escribe con éxito la entrada 332-1 en la carga útil 330, el coprocesador 115-1 de HMQ envía una interrupción al procesador 110-1.

20 El puntero principal 320 es un puntero actualizado atómicamente y apunta a la siguiente entrada de la carga útil 330 que está lista para ser leída. En una realización, el puntero principal 320 es un múltiplo de 64 bytes y está desplazado con respecto al final de la cabecera 310. Por ejemplo, si el puntero principal 320 tiene un valor de cero, entonces el puntero principal 320 está apuntando efectivamente a la entrada 332-1 de la carga útil 330.

25 El puntero 322 de asignación es un puntero actualizado atómicamente. En una realización, el puntero 322 de asignación es un múltiplo de 64 bytes y está desplazado con respecto al final de la cabecera 310. Por ejemplo, si el puntero 322 de asignación tiene un valor de cero, entonces el puntero 322 de asignación está apuntando efectivamente a la entrada 332-1 de la carga útil 330. Si el puntero principal 320 es igual al puntero 322 de asignación, la carga útil 330 se considera vacía. Sin embargo, si el puntero principal 320 menos el puntero 322 de asignación es igual a uno, entonces la carga útil 330 se considera llena.

30 Además, en otras realizaciones, la cabecera 310 puede incluir otros indicadores o valores necesarios para realizar eficientemente las funciones descritas en el presente documento. Los indicadores y los punteros pueden almacenarse en uno o más bits, campos de bits o bytes.

35 Según la realización de la Figura 3, la carga útil 330 incluye, sin limitación de ningún tipo, las entradas 332, cada una de las cuales contiene indicadores 334-1 a 334-N de señal (denominados colectivamente "indicadores 334 de señal"), respectivamente. Cada entrada de las entradas 332 puede ser de un tamaño fijo, y la suma del tamaño de las entradas 332 puede ser igual al tamaño de la carga útil almacenado en la cabecera 310. Por ejemplo, en la realización de la Figura 3, cada una de las entradas 332 tiene una longitud de 64 bytes, y el tamaño de la carga útil 330 es 64 veces N bytes.

40 Los indicadores 334 de señal pueden usarse para indicar que no se han leído entradas particulares 332; que hubo un error escribiendo la entrada; o que la entrada no está lista. Por ejemplo, la entrada 332-1 en la HMQ 140-1 puede ser el último byte de la entrada 332-1 y puede estar puesta a uno para indicar que hay un mensaje válido almacenado en la entrada 332-1, pero que no ha sido leído. En una realización, los indicadores 334 de señal se almacenan en el último byte en cada una de las entradas 332, respectivamente. En otra realización, los indicadores 334 de señal son el primer byte en cada una de las entradas 332, respectivamente.

Inicialización de la HMQ

45 La Figura 4 es un diagrama de flujo que ilustra la inicialización de una HMQ, según una realización. Aunque la Figura 4 ilustra etapas ejemplares según una realización, otras realizaciones pueden omitir, reordenar, modificar cualquiera de las etapas mostradas y/o añadir a las mismas. Aunque esta realización describe la inicialización de una HMQ por el soporte lógico para recibir atómicamente mensajes de múltiples nodos en un sistema de múltiples HMQ, otras realizaciones pueden incluir muchas otras etapas en función de la aplicación y del dispositivo.

50 Como se muestra en la Figura 4, usando la realización ilustrada en la Figura 1 como ejemplo, en la etapa 410, el soporte lógico 130-1 asigna la HMQ 140-1, que es una instancia de la HMQ 300. Por ejemplo, la asignación de la HMQ 140-1 comprende asignar la cabecera 310 y la carga útil 330 de manera contigua. Sin embargo, en otras realizaciones, la cabecera 310 y la carga útil 330 pueden no ser contiguas.

55 En la etapa 420, el soporte lógico 130-1 inicializa la cabecera 310 de la HMQ 140-1. Por ejemplo, la inicialización de la cabecera 310 puede incluir, sin limitación de ningún tipo, poner el indicador 311 de habilitación a uno, poner el indicador 312 de falta de preparación a uno, poner el indicador 313 de error a cero, poner el campo reservado 314 a

uno, asignar 32 bytes para el área 315 de destinatario, poner el indicador 316 de interrupción a cero, poner el puntero principal 320 a cero y configurar el puntero 322 de asignación.

En la etapa 430, el soporte lógico 130-1 crea una instancia de la RKEY 200 de HMQ. La creación de una instancia de la RKEY 200 de HMQ puede incluir, sin limitación de ningún tipo, el soporte lógico 130-1; configurar el tamaño 220 para que sea el tamaño de la carga útil 330, asignado en la etapa 410; designar que la dirección 230 sea la dirección de la carga útil 330. La etapa 430 puede incluir además, sin limitación de ningún tipo, generar un secreto básico; almacenar el secreto básico en el coprocesador 115-1 de HMQ; y generar una clave irreversible 210 en función del secreto básico, del tamaño 220, de la dirección 230 y de los permisos 240.

En la etapa 440, el soporte lógico 130-1 envía una instancia de la RKEY 200 de HMQ, creada a partir de la etapa 430, a todos los nodos 105 del sistema 100 de múltiples HMQ, que autoriza a los nodos 105 a enviar instrucciones al coprocesador 115-1 de HMQ y a almacenar datos en la HMQ 140-1.

En la etapa 450, los indicadores 334 de señal pueden ser configurados para indicar que el coprocesador de HMQ puede escribir en cada entrada respectiva. Por ejemplo, la puesta de los indicadores 334 de señal a cero puede indicar al coprocesador 115-1 de HMQ que el coprocesador de HMQ es libre de empezar a escribir en las respectivas entradas 332 en la HMQ 140-1.

Autenticación de instrucciones en un sistema de múltiples HMQ

La Figura 5 es un diagrama de flujo que ilustra la autenticación de una instrucción, según una realización, usando el sistema 100 de múltiples HMQ como ejemplo.

En la etapa 510 de la Figura 5, el coprocesador 115-1 de HMQ envía al coprocesador 115-M de HMQ una instrucción y una copia de una RKEY de HMQ que el nodo 105-1 recibió del soporte lógico 130-M durante la inicialización de la HMQ 140-M. En la etapa 520, el coprocesador 115-M de HMQ recibe la instrucción y la RKEY de HMQ procedentes de la etapa 510. El coprocesador 115-1 de HMQ puede iniciar el envío de una instrucción. Además, el coprocesador 115-M de HMQ puede ser el coprocesador de HMQ tanto emisor como receptor.

En la etapa 530 de la Figura 5, el coprocesador 115-M de HMQ valida la RKEY de HMQ enviada por el coprocesador 115-1 de HMQ. Dado que la RKEY de HMQ está cifrada en función de la dirección 230, una RKEY de HMQ puede ser válida si la dirección 230 no ha sido alterada. Por ejemplo, como se ha expuesto anteriormente, el coprocesador 115-M de HMQ tiene una copia del secreto básico, que fue generado por el soporte lógico 130-M durante la inicialización de la HMQ 140-M (etapa 430). El coprocesador 115-M de HMQ descifra la clave irreversible 210 de la RKEY de HMQ. Si la clave irreversible 210 es válida, entonces la dirección 230 es válida, y el coprocesador 115-M de HMQ prosigue a la etapa 540; sin no, el coprocesador 115-M de HMQ prosigue a la etapa 595.

En la etapa 540 de la Figura 5, para impedir que se escriba un mensaje que comience dentro de los límites de la HMQ 140-M asignada, pero termina fuera de los límites de la HMQ 140-M asignada, el coprocesador 115-M de HMQ realiza una comprobación para garantizar que el tamaño del mensaje no supere el tamaño 220 de la RKEY de HMQ. Si el tamaño del mensaje que ha de escribirse supera el tamaño 220 de la RKEY de HMQ, entonces el coprocesador 115-M de HMQ aborta, y prosigue a la etapa 595; si no, el coprocesador de HMQ prosigue a la etapa 590. Como se ha descrito anteriormente, el coprocesador 115-M de HMQ no está limitado por un sistema operativo y no hay restricción en cuanto adónde

puede tener un acceso ilimitado a la memoria en la memoria 120-M puede escribir el coprocesador 115-M de HMQ. Por ejemplo, los coprocesadores 115 de HMQ pueden abortar tal instrucción para impedir una escritura no autorizada en la memoria 120-1.

En la etapa 590 de la Figura 5, el coprocesador 115-M de HMQ pone en cola la entrada 332-1 en función de la instrucción recibida por el procesador 115-M de HMQ. Por ejemplo, el coprocesador 115-M de HMQ puede usar el procedimiento de puesta en cola ilustrado en la Figura 6.

En la etapa 595 de la Figura 5, el coprocesador 115-M de HMQ rechaza la instrucción y devuelve un error al coprocesador 115-1 de HMQ. En otra realización, el coprocesador 115-M de HMQ puede simplemente rechazar la instrucción y no devolver ningún error al coprocesador 115-1 de HMQ.

Puesta de mensajes en cola en la HMQ

Cuando el coprocesador 115-M de HMQ recibe una instrucción para poner en cola de manera atómica un mensaje en la HMQ 140-M, el coprocesador 115-M de HMQ introduce atómicamente el mensaje en la HMQ 140-M, que está designada en la RKEY de HMQ enviada con la instrucción. El mensaje es almacenado atómicamente manteniendo el puntero 322 de asignación atómica.

La Figura 6 es un diagrama de flujo que ilustra una puesta en cola de manera atómica de un mensaje en la HMQ 140-M, según una realización. Aunque la Figura 6 ilustra etapas ejemplares según una realización, otras realizaciones pueden omitir, reordenar, modificar cualquiera de las etapas mostradas y/o añadir a las mismas. Aunque esta

realización describe la puesta en cola de manera atómica de un mensaje en una HMQ, otras realizaciones pueden incluir muchas otras etapas en función de la aplicación y del dispositivo.

En la etapa 600 de la Figura 6, continuando con el ejemplo anterior en lo que antecede el coprocesador 115-M de HMQ recibe y autentica una instrucción para poner en cola de manera atómica un mensaje en la HMQ 140-M.

5 En la etapa 610, el coprocesador 115-M de HMQ determina si la HMQ 140-M está llena. El coprocesador 115-M de HMQ determina que la HMQ 140-M designada está llena en función del puntero principal 320, del puntero 322 de asignación y del tamaño de una entrada en la HMQ 140-M. Si la HMQ 140-M está llena, el coprocesador 115-M de HMQ prosigue a la rama afirmativa, etapa 695, y, si no, el coprocesador 115-M de HMQ prosigue a la rama negativa, etapa 620. En una realización, si el puntero 322 de asignación es igual a la cantidad del puntero principal 320 menos el tamaño de una entrada, entonces la HMQ 140-M está llena. En otra realización, el puntero principal 320 y el puntero 322 de asignación son múltiplos del tamaño de la entrada; así, si el puntero principal 320 menos el puntero 322 de asignación es igual a uno, entonces la HMQ 140-M está llena. Obsérvese que, en la realización descrita en la Figura 3, las entradas 332 son del mismo tamaño fijo; sin embargo, en otras realizaciones, el tamaño de las entradas 332 puede no ser igual o de tamaño fijo.

15 En la etapa 620, el coprocesador 115-M de HMQ actualiza el puntero 322 de asignación atómica para que apunte a la siguiente entrada de la carga útil 330. Por ejemplo, si el puntero 322 de asignación almacena un múltiplo del tamaño de la entrada, entonces el puntero 322 de asignación simplemente se incrementa en uno, módulo el número de entradas 332 en la carga útil 330.

20 En la etapa 630, el coprocesador 115-M de HMQ ejecuta la instrucción para poner en cola de manera atómica un mensaje almacenando el mensaje en la entrada 332-1, a la que apunta el puntero 322 de asignación atómica. Por ejemplo, el coprocesador 115-M de HMQ puede inicializar la entrada 332-1 configurando el indicador 334-1 de señal para que indique que la entrada 332-1 no está lista para ser leída. Después de escribir el mensaje en la entrada 332-1, el coprocesador 115-M de HMQ puede poner a cero el indicador 224-1 de señal, para indicar que la entrada 332-1 está completa y que el coprocesador 115-M de HMQ ya no escribe en ella. El coprocesador 115-M de HMQ, prosigue entonces a la etapa 640.

25 En la etapa 640, el coprocesador 115-M de HMQ determina si está puesto el indicador 316 de interrupción. Entonces, si lo está, el coprocesador 115-M de HMQ prosigue a la etapa 650. En la etapa 650, el coprocesador 115-M de HMQ envía una interrupción al sistema operativo en el nodo 105-M que indica que se ha puesto en cola un nuevo mensaje. Por ejemplo, el coprocesador 115-M de HMQ envía al procesador 110-M una interrupción con el valor almacenado en el indicador 316 de interrupción. Además, el coprocesador 115-M de HMQ puede escribir en la memoria intermedia L3 en el procesador 110-M, indicando que se ha puesto en cola un mensaje.

30 El coprocesador 115-M de HMQ prosigue a la etapa 690 si, en la etapa 640, la interrupción no está puesta, o después de la etapa 650. En la etapa 690, el coprocesador de HMQ da aviso al remitente —por ejemplo, el coprocesador 115-1 de HMQ— de que la instrucción recibida ha sido ejecutada. En otra realización, en la etapa 690, el coprocesador 115-M de HMQ no envía ningún aviso al remitente —por ejemplo, el coprocesador 115-1 de HMQ—, sino que, en vez de ello, el coprocesador 115-M de HMQ meramente aguarda a recibir otra instrucción.

35 En la etapa 695, el coprocesador 115-M de HMQ no ejecuta una instrucción para almacenar un mensaje en la HMQ 140-M y devuelve un error al remitente —por ejemplo, el coprocesador 115-1 de HMQ—. En otra realización, el coprocesador 115-M de HMQ no devuelve un error al coprocesador 115-1 de HMQ, sino que, en vez de ello, meramente rechaza la instrucción recibida y aguarda a recibir una nueva instrucción. En otra realización adicional, el coprocesador 115-M de HMQ aguarda una cantidad de tiempo particular y vuelve a intentar almacenar la instrucción recibida del coprocesador 115-1 de HMQ, volviendo a la etapa 610.

Extracción de una entrada de una HMQ

45 La Figura 7 es un diagrama de flujo que ilustra la lectura de un mensaje almacenado en una entrada particular y la extracción de la entrada de la cola, según una realización. Aunque la Figura 7 ilustra etapas ejemplares según una realización, otras realizaciones pueden omitir, reordenar, modificar cualquiera de las etapas mostradas y/o añadir a las mismas. Aunque esta realización describe la lectura de un mensaje almacenado en una entrada particular y la extracción de la entrada de la cola, otras realizaciones pueden incluir muchas otras etapas en función de la aplicación y del dispositivo.

50 En la etapa 700 de la Figura 7, continuando a partir del ejemplo anterior, el soporte lógico 130-M recibe una interrupción que indica que ha sido puesta en cola una entrada 332-1. En otra realización, el soporte lógico 130-M interroga al coprocesador 115-M de HMQ para determinar si se ha puesto en cola una nueva entrada. Si el coprocesador 115-M de HMQ está vacío, el soporte lógico 130-M puede no intentar leer una entrada.

55 En la etapa 710 de la Figura 7, el soporte lógico 130-M verifica el indicador 334-1 de señal de la entrada 332-1 a la que apunta el puntero principal 320, para garantizar que el mensaje almacenado en la entrada 332-1 está listo para ser leído. Si la entrada 332-1 está lista para ser leída, entonces el soporte lógico 130-M prosigue a la rama afirmativa,

etapa 720; si no, el soporte lógico 130-M prosigue a la rama negativa, etapa 795. En otra realización, la memoria intermedia L3 en el procesador 110-M puede contener la dirección de la entrada lista para ser leída.

En la etapa 720 de la Figura 7, el mensaje en la entrada 332-1 es leído por el soporte lógico 130. El soporte lógico prosigue a continuación a la etapa 790. Por ejemplo, el soporte lógico 130 puede copiar la entrada 332-1 a otro lugar en la memoria, o en un fichero, o en una interfaz de conexión de fichero.

En la etapa 790 de la Figura 7, el soporte lógico 130-M actualiza el puntero principal 320 para que apunte a la siguiente entrada de la carga útil 330. Por ejemplo, si el puntero principal 320 es un múltiplo del tamaño de la entrada, entonces el puntero principal 320 se incrementa en uno, módulo el número de entradas 332 en la carga útil 330. En otra realización, el soporte lógico 130-M actualiza atómicamente el puntero principal 320 a la dirección de la siguiente entrada de la carga útil 330. El soporte lógico 130-M también puede actualizar el indicador 334-1 de señal para que indique que la entrada 332-1 fue leída y está libre.

En la etapa 795, el soporte lógico 130-M puede detener su intento de lectura de la entrada 332-1 a la que apunta el puntero principal 320. En otra realización, el soporte lógico 130-M puede volver a la etapa 710. Alternativamente, el soporte lógico 130-M puede aguardar una cantidad de tiempo particular y volver a la etapa 710.

Sincronización de los mensajes recibidos

Las instrucciones enviadas a un coprocesador de HMQ pueden ser sincronizadas. Por ejemplo, el coprocesador 115-M de HMQ puede notificar al coprocesador 115-1 de HMQ que la instrucción que el coprocesador 115-1 de HMQ envió se ejecutó con éxito, y puede terminar la conexión entre los coprocesadores 115-1 y 115-M de HMQ una vez que la instrucción ha sido ejecutada.

Las instrucciones pueden ser procesadas en serie por el coprocesador 115-M de HMQ. Por ejemplo, las instrucciones que se reciben mientras el coprocesador 115-M de HMQ está ocupado pueden ser puestas en cola para su ejecución subsiguiente. Por ejemplo, el coprocesador de HMQ puede tener un indicador de serie, que, si está puesto a uno, ejecute las instrucciones en serie.

Las instrucciones también pueden ser obligadas a ser ejecutadas de forma condicional tras la finalización de una instrucción previa con éxito. En consecuencia, cada instrucción puede tener un indicador condicional. Por ejemplo, si una segunda instrucción es condicional con respecto a una primera instrucción, el indicador condicional de la segunda instrucción puede ser configurado de modo que la segunda instrucción pueda ser ejecutada si la primera instrucción se ejecuta sin errores.

Reensamblaje de mensajes

Para evitar la congestión del sistema de múltiples HMQ, se puede segmentar un mensaje asociado con una instrucción. El coprocesador receptor de HMQ puede reensamblar el mensaje antes de ejecutar la instrucción. Obsérvese que el receptor puede recibir los fragmentos en un orden diferente de aquel en el que se enviaron los fragmentos. Así, el coprocesador de HMQ puede no meramente reensamblar los fragmentos para una instrucción y un mensaje particulares, sino que también puede ordenar los fragmentos antes del reensamblaje. En una realización, la porción del mensaje puede ser segmentada en fragmentos. Alternativamente, tanto la instrucción como el mensaje pueden ser segmentados en fragmentos.

Por ejemplo, usando la realización ilustrada en la Figura 1, el coprocesador 115-1 de HMQ puede enviar una instrucción al coprocesador 115-M de HMQ con un mensaje que tiene 128 bytes. En vez de enviar todo el mensaje de 128 bytes de una sola vez, el coprocesador 115-1 de HMQ segmenta el mensaje en fragmentos de 16 bytes, asigna a cada fragmento un número, y envía cada fragmento numerado al coprocesador 115-M de HMQ. El coprocesador 115-M de HMQ reensambla los fragmentos numerados en orden en función del número de los fragmentos, y luego ejecuta la instrucción recibida y almacena el mensaje reensamblado acompañante.

Notificación de errores

Un coprocesador receptor de HMQ puede estar configurado para generar errores particulares, o devolver errores particulares a un coprocesador remitente de HMQ. Por ejemplo, si se produce un error mientras el coprocesador 115-M de HMQ está procesando una instrucción enviada desde el coprocesador 115-1 de HMQ, entonces el coprocesador 115-M de HMQ puede generar uno o más errores particulares, y también puede devolver los uno o más errores al coprocesador 115-1 de HMQ.

Un coprocesador de HMQ puede generar errores incluyendo, sin limitación alguna, la validación de la RKEY de HMQ; una cola que cruza un límite de página o una región de la RKEY de HMQ; el tamaño de un mensaje no es igual al tamaño de la entrada especificado en la cabecera 310; el indicador 311 de habilitación está puesto a cero o no coincide con un valor previsto; el tamaño del mensaje no es una potencia de dos; el tamaño de la carga útil 330 no es un múltiplo del tamaño del mensaje; el puntero 322 de asignación no se encuentra en el intervalo de cero a número de entradas 332; o la HMQ 300, o específicamente la carga útil 330, está llena. Además, si se genera un error mientras

se intenta escribir en la entrada 332-1, puede escribir en el indicador 334-1 de señal un valor que represente un error particular. Además, también se puede poner el indicador 313 de error si se genera un error.

Descripción general del soporte físico

Según una realización, las técnicas descritas en el presente documento son implementadas por uno o más dispositivos informáticos de uso especial. Los dispositivos informáticos de uso especial pueden estar cableados para llevar a cabo las técnicas, o pueden incluir digital dispositivos electrónicos tales como uno o más circuitos integrados para aplicaciones específicas (ASIC) o matrices de puertas programables *in situ* (FPGA) que están programados de forma persistente para llevar a cabo las técnicas, o pueden incluir uno o más procesadores de soporte físico de uso general programados para llevar a cabo las técnicas de acuerdo con instrucciones de programa en soporte lógico inalterable, memoria, otro almacenamiento o una combinación de los mismos. Tales dispositivos informáticos de uso especial también pueden combinar lógica cableada hecha a medida, ASIC o FPGA con programación a medida para lograr las técnicas. Los dispositivos informáticos de uso especial pueden ser sistemas de ordenadores de sobremesa, sistemas de ordenadores portátiles, dispositivos de mano, dispositivos de red o cualquier otro dispositivo que incorpore lógica cableada y/o programada para implementar las técnicas.

Por ejemplo, la Figura 8 es un diagrama de bloques que ilustra un sistema informático 800 en el que puede implementarse una realización de la invención. El sistema informático 800 incluye un *bus* 802 u otro mecanismo de comunicación para comunicar información, y un procesador 804 de soporte físico de acoplado con el *bus* 802 para procesar información. El procesador 804 de soporte físico puede ser, por ejemplo, un microprocesador de uso general.

El sistema informático 800 también incluye una memoria principal 806, tal como una memoria de acceso aleatorio (RAM, por sus siglas en inglés) u otro dispositivo de almacenamiento dinámico, acoplado al *bus* 802 para almacenar información e instrucciones que ha de ejecutar el procesador 804. La memoria principal 806 también puede ser usada para almacenar variables temporales u otra información intermedia durante la ejecución de las instrucciones que han de ser ejecutadas por el procesador 804. Tales instrucciones, cuando están almacenadas en un soporte de almacenamiento no transitorio accesible al procesador 804, convierten el sistema informático 800 en una máquina de uso especial que está hecha a medida para llevar a cabo las operaciones especificadas en las instrucciones.

El sistema informático 800 incluye, además, una memoria 808 de solo lectura (ROM, por sus siglas en inglés) u otro dispositivo de almacenamiento estático acoplado al *bus* 802 para almacenar información estática e instrucciones para el procesador 804. Se proporciona un dispositivo 810 de almacenamiento, tal como un disco magnético, un disco óptico o una unidad de estado sólido y se lo acopla al *bus* 802 para almacenar información e instrucciones.

El sistema informático 800 puede estar acoplado a través del *bus* 802 a un dispositivo 812 de visualización, tal como un tubo de rayos catódicos (CRT, por sus siglas en inglés), para mostrar información un usuario del ordenador. Hay un dispositivo 814 de entrada, que incluye claves alfanuméricas y otras, acoplado al *bus* 802 para comunicar información e indicar selecciones al procesador 804. Otro tipo de dispositivo de entrada de usuario es un control 816 del cursor, tal como un ratón, una bola de mando o teclas de dirección del cursor para comunicar información de dirección e indicar selecciones al procesador 804 y para controlar el movimiento del cursor en el dispositivo 812 de visualización. Este dispositivo de entrada normalmente tiene dos grados de libertad en dos ejes: un primer eje (por ejemplo, x) y un segundo eje (por ejemplo, y), que permite al dispositivo especificar posiciones en un plano.

El sistema informático 800 puede implementar las técnicas descritas en el presente documento usando lógica cableada hecha a medida, uno o más ASIC o FPGA, soporte lógico inalterable y/o lógica de programa que, en combinación con el sistema informático, hacen o programan el sistema informático 800 sea una máquina de uso especial. Según una realización, las técnicas del presente documento son efectuadas por el sistema informático 800 en respuesta a la ejecución por parte del procesador 804 de una o más secuencias de una o más instrucciones contenidas en la memoria principal 806. Tales instrucciones pueden ser leídas y puestas en la memoria principal 806 a partir de otro soporte de almacenamiento, tal como el dispositivo 810 de almacenamiento. La ejecución de las secuencias de instrucciones contenidas en la memoria principal 806 hace que el procesador 804 realice las etapas de proceso descritas en el presente documento. En realizaciones alternativas, pueden usarse circuitos cableados en lugar de instrucciones de soporte lógico o en combinación con las mismas.

La expresión "soportes de almacenamiento" usada en el presente documento se refiere a cualquier soporte no transitorio que almacene datos y/o instrucciones que hagan que una máquina opere de una forma específica. Tales soportes de almacenamiento comprenden soportes no volátiles y/o soportes volátiles. Los soportes no volátiles incluyen, por ejemplo, discos ópticos, discos magnéticos o unidades de estado sólido, como el dispositivo 810 de almacenamiento. Los soportes volátiles incluyen memoria dinámica, como la memoria principal 806. Formas comunes de soportes de almacenamiento incluyen, por ejemplo, un disquete, un disco flexible, un disco duro, una unidad de estado sólido, una cinta magnética o cualquier otro soporte magnético de almacenamiento de datos, un CD-ROM, cualquier otro soporte óptico de almacenamiento de datos, cualquier soporte físico con patrones de agujeros, una RAM, una PROM y una EPROM, una FLASH-EPROM, NVRAM y cualquier otro chip o cartucho de memoria.

Los soportes de almacenamiento son distintos de los soportes de transmisión, pero pueden ser usados en conjunción con los mismos. Los soportes de transmisión participan en la transferencia de información entre soportes de

almacenamiento. Por ejemplo, los soportes de transmisión incluyen cables coaxiales, hilo de cobre y fibra óptica, incluyendo los hilos que comprenden el *bus* 802. Los soportes de transmisión también pueden adoptar la forma de ondas acústicas o lumínicas, tales como las generadas durante las comunicaciones de datos por ondas de radio e infrarrojos.

- 5 Puede haber implicadas formas diversas de soportes para el transporte de una o más secuencias de una o más instrucciones al procesador 804 para su ejecución. Por ejemplo, las instrucciones pueden estar inicialmente transportadas en un disco magnético o una unidad de estado sólido de un ordenador remoto. El ordenador remoto puede cargar las instrucciones en su memoria dinámica y enviar las instrucciones por una línea telefónica usando un módem. Un módem local al sistema informático 800 puede recibir los datos por la línea telefónica y usar un transmisor
- 10 infrarrojo para convertir los datos en una señal infrarroja. Un detector infrarrojo puede recibir los datos transportados en la señal infrarroja y circuitos apropiados pueden poner los datos en el *bus* 802. El *bus* 802 transporta los datos a la memoria principal 806, desde la cual el procesador 804 recupera y ejecuta las instrucciones. Las instrucciones recibidas por la memoria principal 806 pueden almacenarse opcionalmente en el dispositivo 810 de almacenamiento ya sea antes o después de su ejecución por el procesador 804.
- 15 El sistema informático 800 también incluya una interfaz 818 de comunicaciones acoplada al *bus* 802. La interfaz 818 de comunicaciones proporciona un acoplamiento bidireccional de comunicaciones de datos con un enlace 820 de red que está conectado a una red local 822. Por ejemplo, la interfaz 818 de comunicaciones puede ser una tarjeta de red digital de servicios integrados (RDSI), un módem de cable, un módem de satélite o un módem para proporcionar una conexión de comunicación de datos a un correspondiente tipo de línea telefónica. Como ejemplo adicional, la interfaz
- 20 818 de comunicaciones puede ser una tarjeta de red de área local (LAN, por sus siglas en inglés) para proporcionar una conexión de comunicación de datos a una LAN compatible. También pueden implementarse enlaces inalámbricos. En cualquier implementación de ese tipo, la interfaz 818 de comunicaciones envía y recibe señales eléctricas, electromagnéticas u ópticas que transportan flujos de datos digitales que representan diversos tipos de información.
- 25 El enlace 820 de red normalmente proporciona a otros dispositivos de datos una comunicación de datos a través de una o más redes. Por ejemplo, el enlace 820 de red puede proporcionar una conexión a través de la red local 822 a un ordenador central 824 o a equipos de red operados por un proveedor 826 de servicios de Internet (ISP, por sus siglas en inglés). El ISP 826, a su vez, proporciona servicios de comunicaciones de datos a través de la red mundial de comunicaciones de paquetes de datos ahora comúnmente denominada "Internet" 828. Tanto la red local 822 como Internet 828 usan señales eléctricas, electromagnéticas u ópticas que transportan flujos de datos digitales. Las señales
- 30 a través de las diversas redes y las señales en el enlace 820 de red y a través de la interfaz 818 de comunicaciones, que transportan los datos digitales hacia y desde el sistema informático 800, son ejemplos de soportes de transmisión.
- El sistema informático 800 puede enviar mensajes y recibir datos, incluyendo código de programa, a través de la red o las redes, del enlace 820 de red y de la interfaz 818 de comunicaciones. En el ejemplo de Internet, un servidor 830 podría transmitir un código solicitado para un programa de aplicación a través de Internet 828, de un ISP 826, de la
- 35 red local 822 y de la interfaz 818 de comunicaciones.
- El código recibido puede ser ejecutado por el procesador 804 cuando es recibido, y/o ser almacenado en el dispositivo 810 de almacenamiento y otro almacenamiento no volátil para su ejecución posterior.
- En la memoria descriptiva precedente, se han descrito realizaciones de la invención con referencia a numerosos detalles específicos que pueden variar de una implementación a otra. En consecuencia, la memoria descriptiva y los
- 40 dibujos han de ser considerados en un sentido ilustrativo, no restrictivo. El único indicador exclusivo del alcance de la invención, y que los solicitantes pretenden que sea el alcance de la invención, es el alcance literal y equivalente del pliego de reivindicaciones que emanan de esta solicitud, en la forma específica en la que se presentan tales reivindicaciones, incluyendo cualquier corrección subsiguiente.

REIVINDICACIONES

1. Un dispositivo para facilitar la comunicación entre una pluralidad de nodos (105-1, 105-M) de un grupo de cómputo, comprendiendo el dispositivo un coprocesador (115-M) de un nodo particular (105-M) de dicha pluralidad de nodos (105-1, 105-M), comprendiendo el coprocesador (115-M):
 - 5 un circuito de puesta en cola;

estando acoplado dicho circuito de puesta en cola a una memoria (120-M) de dicho nodo particular (105-M);

siendo dicho circuito de puesta en cola un circuito cableado o un ASIC programado de manera persistente o una FPGA programada de manera persistente configurado para:

recibir (520), desde otro nodo (105-1) de dicha pluralidad de nodos (105-1, 105-M), una instrucción de puesta en cola que comprende una clave con un código (210) de clave irreversible cifrada derivado de un secreto básico, en donde:

la clave identifica una cola particular (300, 140-M) en dicha memoria (120-M), habiendo sido creada la clave previamente en dicho nodo particular (105-M) y enviada desde dicho nodo particular (105-M) a dicho otro nodo (105-1), y

la instrucción de puesta en cola incluye una carga útil que ha de almacenarse en la cola particular (300, 140-M);

autenticar (530) la instrucción de puesta en cola, comprendiendo el desciframiento del código (210) de clave irreversible cifrada el uso de dicho secreto básico almacenado en dicho coprocesador (115-M) y la determinación de si el código (210) de clave irreversible es válido; y

poner en cola (590) una entrada (332-1, 332-2) de cola, en la que se almacena dicha carga útil, en la cola particular (300, 140-M) identificada por la instrucción recibida de puesta en cola en respuesta a la recepción (520) de dicha instrucción de puesta en cola si la instrucción de puesta en cola fue autenticada con éxito.
 - 25 2. El dispositivo de la Reivindicación 1, incluyendo dicho dispositivo un chip de circuitos integrados que incluye, separado de dicho circuito de puesta en cola, un procesador (110-M), estando acoplado dicho procesador (110-M) a dicho circuito de puesta en cola.
 3. El dispositivo de la Reivindicación 1 o la Reivindicación 2 en donde dicho circuito de puesta en cola está configurado, además, para ejecutar en serie instrucciones de puesta en cola.
 - 30 4. El dispositivo de cualquiera de las Reivindicaciones 1-3 en donde dicha carga útil incluida en dicha instrucción de puesta en cola se recibe en una pluralidad de fragmentos, y estando configurado además dicho circuito de puesta en cola para reensamblar dicha pluralidad de fragmentos en respuesta a la recepción de dicha instrucción de puesta en cola y dicha pluralidad de fragmentos.
 - 35 5. El dispositivo de la Reivindicación 4 en donde dicha pluralidad de fragmentos se recibió en un orden diferente en el que se envió dicha pluralidad de fragmentos.
 6. El dispositivo de cualquiera de las Reivindicaciones 1-5 en donde un código válido (210) de clave irreversible indica que la identificación de la cola particular (300, 140-M) por la clave es válida.
 7. Un procedimiento para facilitar la comunicación entre una pluralidad de nodos (105-1, 105-M) de un grupo de cómputo, en donde un nodo particular (105-M) de dicha pluralidad de nodos (105-1, 105-M) comprende un coprocesador (115-M), incluyendo el procedimiento:
 - 40 transmitir, desde otro nodo (105-1) de dicha pluralidad de nodos (105-1, 105-M), una instrucción particular de puesta en cola a un circuito de puesta en cola de dicho coprocesador (115-M) de dicho nodo particular (105-M), la instrucción particular de puesta en cola, que es una instrucción de puesta en cola de una entrada (332-1, 332-N) de cola en una cola particular (300, 140-M) almacenada en una memoria (120-M) de dicho nodo particular (105-M), estando acoplada dicha memoria (120-M) a dicho circuito de puesta en cola, identificando dicha instrucción particular de puesta en cola dicha cola particular (300, 140-M);
 - 45 en donde dicho circuito de puesta en cola es un circuito cableado o un ASIC programado de manera persistente o una FPGA programada de manera persistente configurado para:
 - 50 recibir (520), desde dicho otro nodo (105-1) de dicha pluralidad de nodos (105-1, 105-M), dicha instrucción particular de puesta en cola que comprende una clave con un código (210) de clave irreversible cifrada derivado de un secreto básico, en donde:

la clave identifica una cola particular (300, 140-M) en dicha memoria (120-M), habiendo sido creada la clave previamente en dicho nodo particular (105-M) y enviada desde dicho nodo particular (105-M) a dicho otro nodo (105-1), y

- 5 la instrucción de puesta en cola incluye una carga útil que ha de almacenarse en la cola particular (300, 140-M);
autenticar (530) la instrucción de puesta en cola, comprendiendo el desciframiento del código (210) de clave irreversible cifrada el uso de dicho secreto básico almacenado en dicho coprocesador (115-M) y la determinación de si el código (210) de clave irreversible es válido; y
- 10 en respuesta a la recepción de la instrucción de puesta en cola, poner en cola (590) una entrada de cola, en la que se almacena dicha carga útil, en la cola particular (300, 140-M) identificada por la instrucción recibida de puesta en cola si la instrucción de puesta en cola fue autenticada con éxito.
8. El procedimiento de la Reivindicación 7 en donde dicho circuito de puesta en cola está situado en un chip de circuitos integrados que incluye, separado de dicho circuito de puesta en cola, un procesador (110-M), estando
15 acoplado dicho procesador (110-M) a dicho circuito de puesta en cola.
9. El procedimiento de la Reivindicación 7 o la Reivindicación 8 que, además, comprende la recepción (700) de una interrupción en respuesta a la puesta en cola de dicha entrada (332-1, 332-N) de cola.
10. El procedimiento de cualquiera de las Reivindicaciones 7-9 que, además, comprende extraer una entrada (332-1, 332-N) de cola de dicha cola particular (300, 140-M).
- 20 11. El procedimiento de la Reivindicación 10 en donde extraer una entrada (332-1, 332-N) de cola de dicha cola particular (300, 140-M) comprende establecer un valor (334-1, 334-N) en dicha entrada (332-1, 332-N) de cola para indicar que la entrada (332-1, 332-N) de cola ha sido leída.
12. El procedimiento de la Reivindicación 10 en donde extraer una entrada (332-1, 332-N) de cola de dicha cola (300, 140-M) comprende establecer atómicamente (790) un puntero para que apunte a una entrada (332-1, 332-N) de cola
25 que no ha sido leída.
13. El procedimiento de la Reivindicación 10 en donde extraer una entrada (332-1, 332-N) de cola de dicha cola particular (300, 140-M) comprende copiar dicha entrada (332-1, 332-N) de cola en memoria (120-M).
14. El procedimiento de cualquiera de las Reivindicaciones 7-13 en donde un código válido (210) de clave irreversible indica que la identificación de la cola particular (300, 140-M) por la clave es válida.
- 30 15. Un sistema que comprende medios para llevar a cabo el procedimiento mencionado en una cualquiera de las Reivindicaciones 7-14.

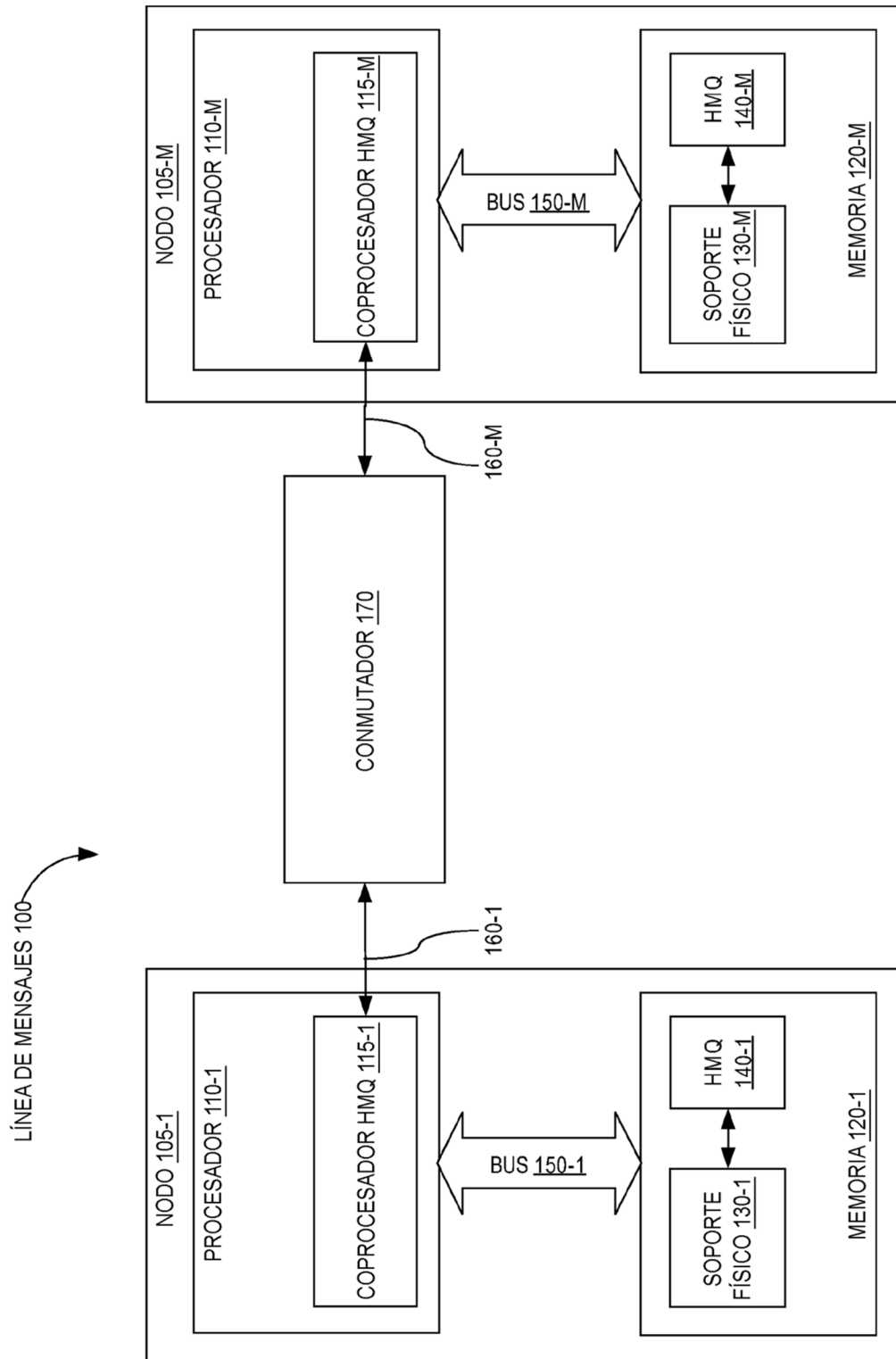


FIG. 1

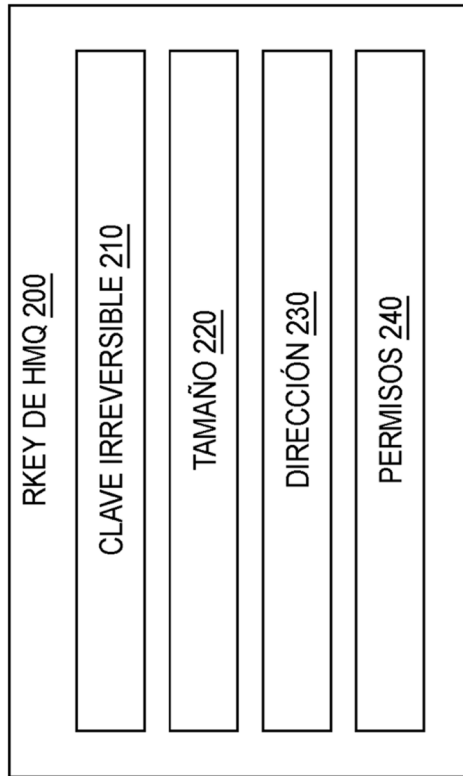


FIG. 2

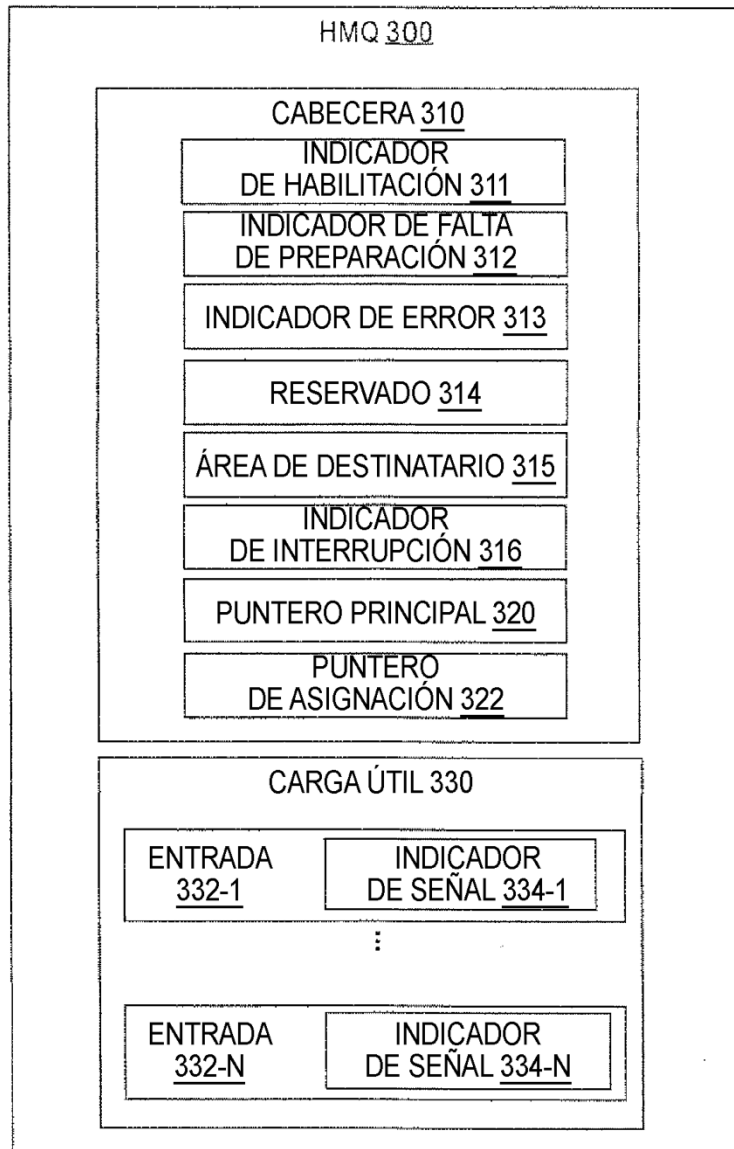


FIG. 3

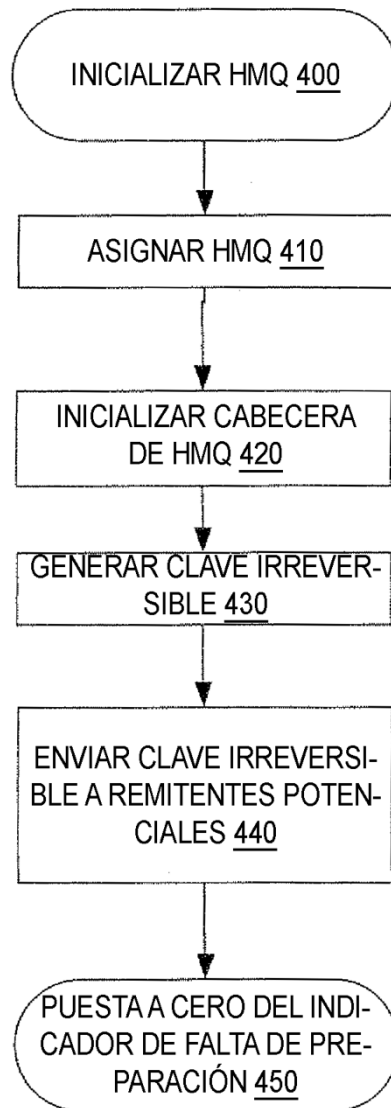


FIG. 4

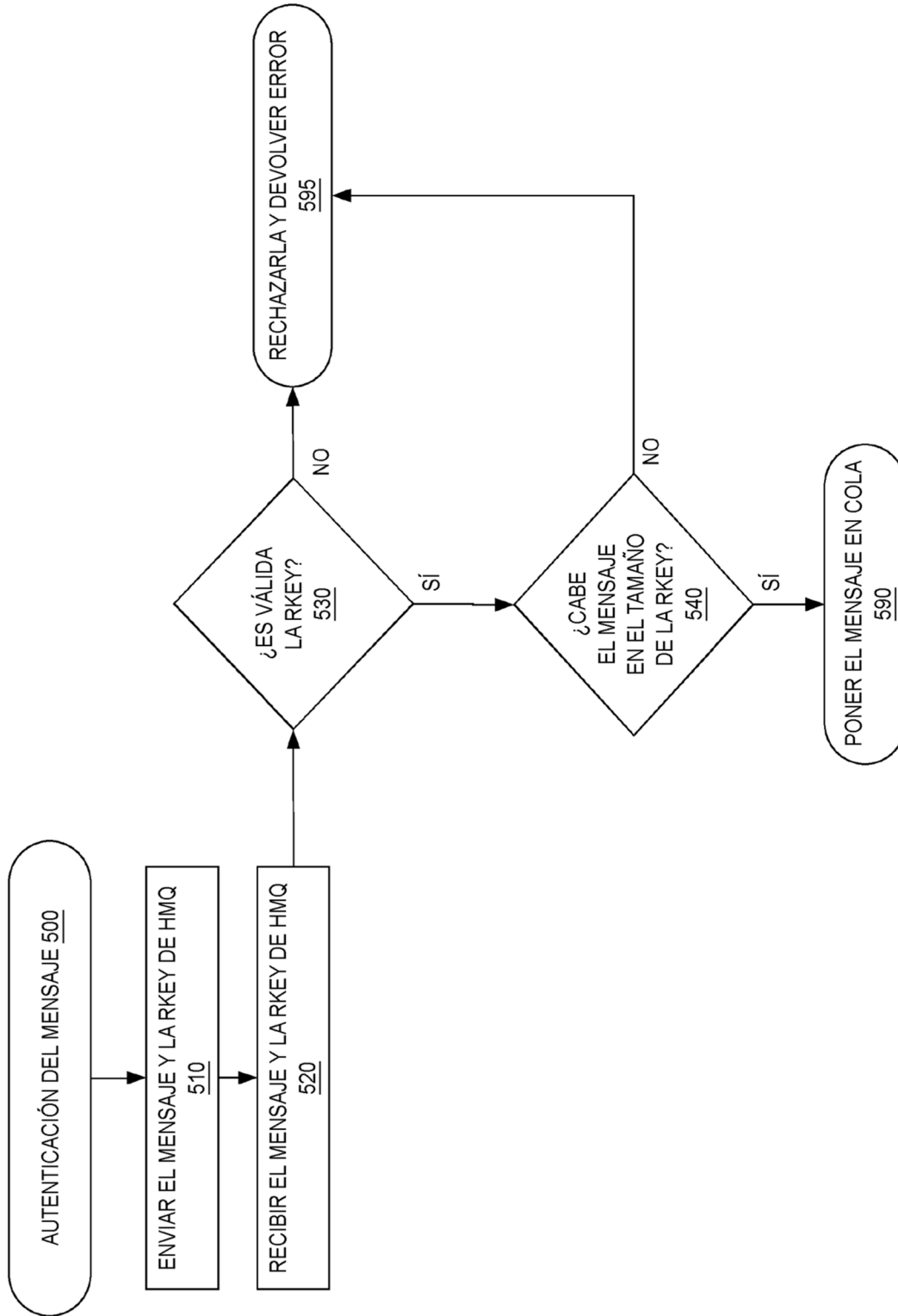


FIG. 5

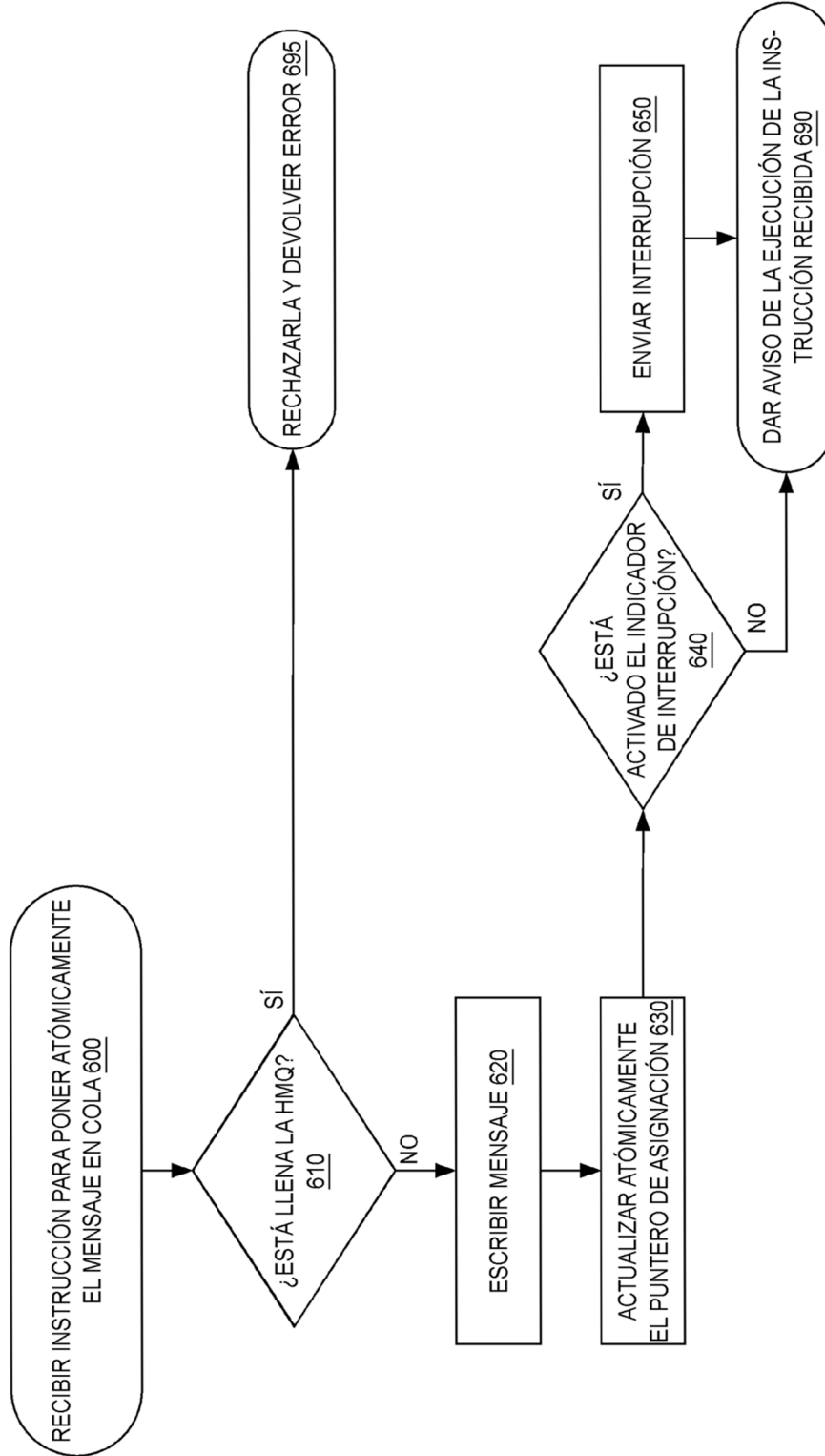


FIG. 6

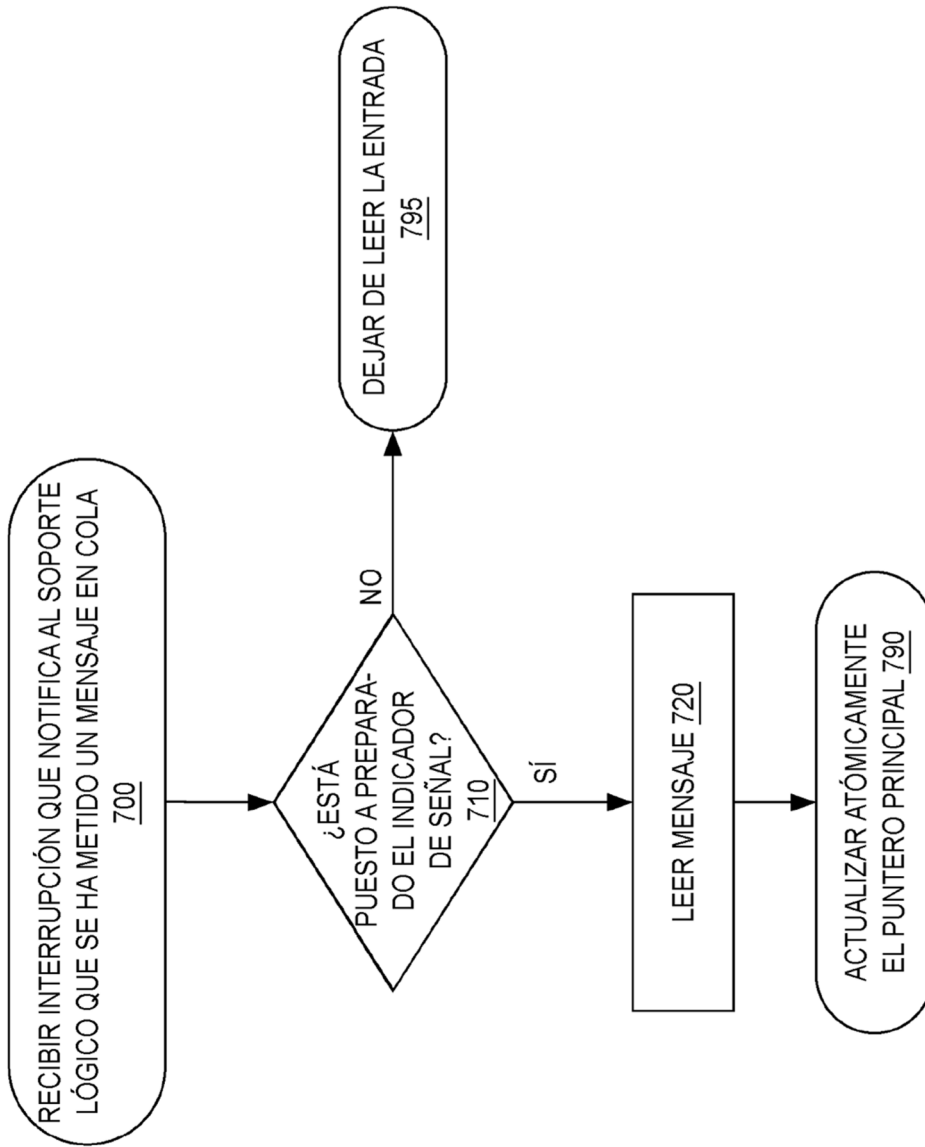


FIG. 7

FIG. 8

