(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 822 919**

(51) Int. CI.:

G06N 3/02 (2006.01)
G06N 3/04 (2006.01)
G06N 3/063 (2006.01)
H01L 25/04 (2014.01)
G06F 15/78 (2006.01)
H04L 12/70 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **27.04.2016 PCT/EP2016/059446**

(87) Fecha y número de publicación internacional: **03.11.2016 WO16174113**

(96) Fecha de presentación y número de la solicitud europea: **27.04.2016 E 16724294 (0)**

(97) Fecha y número de publicación de la concesión europea: **08.07.2020 EP 3289526**

(54) Título: **Red y método para sistemas informáticos escalables accionados por eventos**

(30) Prioridad:

27.04.2015 EP 15165272

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
05.05.2021

(73) Titular/es:

UNIVERSITÄT ZÜRICH (100.0%)
Prorektorat MNW Rämistrasse 71
8006 Zürich, CH

(72) Inventor/es:

INDIVERI, GIACOMO;
MORADI, SABER;
QIAO, NING y
STEFANINI, FABIO

(74) Agente/Representante:

INGENIAS CREACIONES, SIGNOS E INVENCIONES, SLP

DESCRIPCIÓN

Red y método para sistemas informáticos escalables accionados por eventos

La invención se refiere a redes, particularmente redes neurales, estructuras de encaminamiento y correspondientes métodos.

El documento US2014/0032465A1 describe una red neural que comprende múltiples circuitos de núcleos neurales funcionales, y una interconexión de conmutación dinámicamente reconfigurable entre los circuitos de núcleos neurales funcionales.

Adicionalmente, el documento US2008/0120260 divulga sistemas y métodos para formar redes neurales reconfigurables con FPGA interconectadas teniendo cada una un encaminador de paquetes.

El problema que subyace de la presente invención es proporcionar redes mejoradas, particularmente redes neurales, y correspondientes métodos.

Este problema se resuelve por una red de acuerdo con la reivindicación 1. La red comprende una pluralidad de circuitos de núcleo interconectados (por ejemplo, dispuestos en varios chips o unidades o piezas), en la que cada circuito de núcleo comprende:

- una matriz electrónica que comprende una pluralidad de nodos de cálculo y una pluralidad de circuitos de memoria, cuya matriz electrónica está configurada para recibir eventos entrantes (y particularmente usando memoria local para discriminar eventos de entrada o eventos entrantes), en la que cada nodo de cálculo está configurado para generar un evento que comprende un paquete de datos si los eventos entrantes recibidos por el respectivo nodo de cálculo satisfacen un criterio predefinido, y
- un circuito que está configurado para adjuntar una dirección de destino e información de origen adicional (por ejemplo, un ID de núcleo de origen, particularmente, ID de núcleo de origen virtual, véase, por ejemplo, a continuación) al respectivo paquete de datos, y
- un primer encaminador local (R1) para proporcionar conectividad intra núcleo rápida y/o entregar eventos a un segundo encaminador de nivel intermedio (R2) para inter núcleo y tercer encaminador de nivel superior (R3) para conectividad inter chip (o inter unidad o inter pieza), y
- un controlador de difusión para difundir eventos entrantes a todos los circuitos de memoria en el núcleo en paralelo.

Además, de acuerdo con la invención, la red comprende además un sistema de encaminamiento de eventos que interconecta los circuitos de núcleo, en la que el sistema de encaminamiento de eventos comprende dichos primeros encaminadores locales, y también segundos y terceros encaminadores, en la que la totalidad de los encaminadores forman una estructura jerárquica.

A continuación, los circuitos de núcleo se indican también como núcleos.

Adicionalmente, particularmente, el respectivo nodo de cálculo puede recibir dichos eventos entrantes a través de circuitos de memoria de la respectiva matriz electrónica.

Particularmente, se proporciona una estructura de encaminamiento basada en eventos que combina estructuras de encaminamiento jerárquicas con arquitecturas de memoria heterogéneas. Esta estructura de encaminamiento puede aplicarse a arquitecturas que comprenden nodos de cálculo asíncronos distribuidos a través de múltiples núcleos de cálculo. La estructura consta de una combinación de encaminamiento basada en origen y destino en la que se procesan paquetes de datos en diferentes niveles de la jerarquía antes de encaminarse. Esto permite la construcción de estructuras de red heterogéneas que habilitan la optimización de memoria y ancho de banda. Demostramos la invención con la realización de una pila de estructuras de encaminamiento que usan tres estrategias diferentes: encaminamiento de difusión, árbol y malla. La memoria usada para almacenar la conectividad entre nodos de cálculo usa diferentes estructuras distribuidas dentro de nodos, núcleos y encaminadores. Los nodos de cálculo operan en paralelo, independiente y asíncronamente, procesando una entrada asíncrona y produciendo un evento asíncrono cuando se cumplen las condiciones en los datos de entrada. En el caso de redes neurales de impulsos, los nodos de cálculo son neuronas artificiales que reciben eventos de entrada (por ejemplo, eventos entrantes) desde múltiples orígenes y producen un impulso de salida cuando la suma de los eventos de entrada cruza un umbral establecido. Demostramos la invención dentro del contexto de una red neural de impulsos escalable con conectividad programable implementada en un microchip totalmente personalizado.

Además, particularmente, la presente invención se refiere a sistemas basados en eventos asíncronos y circuitos para procesamiento de información y cálculo, particularmente redes basadas en eventos y, particularmente, redes neurales de impulsos. Sistemas basados en eventos son sistemas informáticos electrónicos en los que elementos efectúan cálculo y comunicación a través de eventos asíncronos, producidos por sus nodos de cálculo cuando se cumplen conjuntos de condiciones dados en sus señales de entrada. La invención se refiere a una estructura de encaminamiento jerárquica con estructuras de memoria distribuidas y heterogéneas que optimiza el uso de memoria

para programar la conectividad de red. Como la conectividad de red determina sus propiedades de cálculo, la invención puede usarse para construir implementaciones de hardware de "redes profundas", incluyendo redes convolucionales y de creencia profunda, redes neurales recurrentes, incluyendo redes de computación de contenedores, así como modelos gráficos probabilísticos, incluyendo gráficos de factores. La conectividad de red se realiza programando Tablas de Consulta (LUT) de encaminamiento distribuidas en la red y Memorias de Contenido Direccionable (CAM) asociadas a nodos de cálculo. Los tipos de redes que pueden programarse en el sistema dependen de la cantidad de memoria asignada en los elementos, núcleos y encaminadores.

Una instancia particular de tales sistemas es una red neural de impulsos. Los elementos de cálculo se modelan en las neuronas dinámicas o biológicas y como tal generan eventos, a menudo denominados como impulsos, en respuesta a los impulsos de entrada integrados que exceden un umbral establecido. Este evento de impulso se codifica como un paquete y se entrega a su destino por una red física que comprende unidades de encaminadores y unidades sinápticas. La conectividad de la red neuronal se realiza a través del encaminamiento apropiado de eventos de impulsos desde los orígenes a sus destinos usando estructuras de memoria locales y distribuidas. Esta arquitectura se denomina como accionada por eventos e implementa la red neural artificial.

Adicionalmente, particularmente, realizaciones de la invención proporcionan arquitecturas accionadas por eventos asíncronas para sistemas informáticos paralelos. El grado de conectividad entre los nodos de cálculo de sistema y los tipos de cálculos que pueden programarse en el sistema dependen de la cantidad de memoria asignada a los nodos individuales, núcleos y encaminadores.

Particularmente, la presente invención se describe por medio de una realización en la que los nodos de cálculo son neuronas de integrar y activar con sinapsis dinámicas sin limitar el alcance de la presente invención. De acuerdo con esta realización, una red neural comprende una pluralidad de nodos interconectados dentro de y a través de múltiples núcleos, distribuidos en uno o más chips electrónicos. Cada núcleo comprende una multitud de bloques con elementos neurales y sinápticos que almacenan la conectividad de la red y realizan el cálculo neural. Los elementos sinápticos de una neurona en el núcleo tienen estructuras de memoria que almacenan la identidad de las neuronas presinápticas desde las que aceptan entradas. Cuando la neurona presináptica correcta estimula un elemento sináptico válido, este elemento genera una corriente analógica que se integra por la neurona postsináptica a la que se conecta la sinapsis. Cuando las corrientes de entrada integradas enviadas por todas las sinapsis conectadas a una neurona exceden un umbral, la neura activa un mecanismo de generación de impulsos. Cuando una neurona produce un impulso de salida, este se codifica por su dirección de origen y esta dirección se encamina como un paquete de datos asíncrono a otros nodos que siguen un esquema jerárquico. En el nivel inferior, un encaminador de núcleo distribuye impulsos cuyos origen y destinos se ubican en el mismo núcleo. En niveles intermedios uno o más conjuntos de encaminadores de árbol distribuyen impulsos que o bien se generan por o dirigen a núcleos dentro del mismo chip. Encaminadores de árbol se organizan en un nivel jerárquico y múltiples encaminadores pueden expandir múltiples niveles de la jerarquía.

En el nivel superior, un encaminador de malla distribuye impulsos a través de múltiples chips distribuidos en una red bidimensional.

Además, de acuerdo con una realización de la red de acuerdo con la invención, cada encaminador (R1, R2, R3) comprende al menos un circuito de control que está configurado para encaminar eventos (o señales) de acuerdo con la carga útil de ruta, particularmente comprendida por el respectivo paquete de eventos/datos, y en la que cada primer encaminador (R1) comprende además:

- una memoria programable (por ejemplo, una matriz de memorias locales digitales) configurada para almacenar carga útil de ruta y, particularmente, ID de núcleo de origen virtual para dichos paquetes de datos,

- al menos un circuito configurado para adjuntar carga útil de ruta y, particularmente, ID de núcleo de origen virtual al respectivo paquete de datos, dependiendo de asignaciones de ruta programadas almacenadas en dicha memoria.

El ID de núcleo de origen virtual es un código adicional adjuntado a la dirección de origen independientemente en cada neurona para aumentar el espacio de direcciones total, particularmente para aumentar la discriminación de eventos y, por lo tanto, para reducir la ambigüedad de direcciones, sobre una base por neurona en lugar de una base por núcleo.

Además, de acuerdo con una realización de la red de acuerdo con la invención, los circuitos de núcleo se disponen en unidades, particularmente en forma de piezas modulares o chips, en la que particularmente cada unidad comprende varios de los circuitos de núcleo, y en la que cada unidad comprende uno de los primeros encaminadores, en la que particularmente cada uno de dichos primeros encaminadores se disponen en una estructura jerárquica formada por la totalidad de los encaminadores.

Además, de acuerdo con una realización de la red de acuerdo con la invención, dicha estructura jerárquica comprende un nivel inferior que comprende los primeros encaminadores, en la que particularmente los primeros encaminadores están configurados para distribuir eventos cuyos origen y destinos se ubican en el mismo circuito de núcleo.

Además, de acuerdo con una realización de la red de acuerdo con la invención, dicha estructura jerárquica comprende al menos un nivel intermedio que comprende los segundos encaminadores, en la que particularmente los segundos encaminadores están configurados para distribuir eventos generados por o dirigidos a circuitos de núcleo dentro de la misma unidad.

Además, de acuerdo con una realización de la red de acuerdo con la invención, dicha estructura jerárquica comprende un nivel superior que comprende los terceros encaminadores (por ejemplo, de malla) que están configurados para distribuir eventos entre diferentes unidades, en la que particularmente los terceros encaminadores se disponen en una malla bidimensional.

Además, de acuerdo con una realización de la red de acuerdo con la invención, los encaminadores se disponen en una estructura jerárquica formada por los encaminadores, en la que diferentes esquemas de encaminamiento asíncronos coexisten en diferentes niveles de la estructura jerárquica correlacionando direcciones de origen y dichos paquetes de datos generados por los nodos de cálculo en los circuitos de núcleo para hacer coincidir los diferentes esquemas de encaminamiento en los diferentes niveles.

Además, de acuerdo con una realización de la red de acuerdo con la invención, la red es una red neural sintética, en la que cada nodo de cálculo forma una neurona, y en la que cada uno de dichos circuitos de memoria forma una sinapsis, en la que particularmente los nodos de cálculo se diseñan para integrar (por ejemplo, sumar) eventos entrantes y para generar un evento cuando la señal formada por los eventos integrados cruza un umbral de activación.

También se divulga en este punto como una realización una red (por ejemplo, masivamente) paralela de procesadores de múltiples núcleos (por ejemplo, de alta) interacción, comprendiendo cada uno una pluralidad de circuitos de núcleo que están configurados para comunicar eventos en forma de paquetes de datos dentro de un circuito de núcleo y/o entre diferentes circuitos de núcleo, en la que la red está configurada para regular dicha comunicación por una jerarquía de encaminadores asíncronos que están configurados para actuar en trayectorias de comunicación independientes.

Además, de acuerdo con una realización de la red paralela de acuerdo con la invención, cada paquete de datos consta de: una dirección de origen codificada de un nodo de cálculo de un circuito de núcleo que genera esa dirección, códigos digitales adicionales que especifican una parte o todas las rutas del respectivo evento a lo largo de la red.

De acuerdo con la reivindicación 9, la presente invención se refiere adicionalmente al aspecto de un método para encaminar eventos en una red, usando una red de acuerdo con una de las reivindicaciones 1 a 8. El método comprende

- generar un evento que comprende un paquete de datos por medio de un nodo de cálculo si los eventos entrantes recibidos por el respectivo nodo de cálculo satisfacen un criterio predefinido,
- para cada unidad de núcleo,
- distribuir el evento generado dentro de su circuito de núcleo por medio de un primer encaminador local comprendido por cada circuito de núcleo, en el que una dirección de destino y, particularmente, ID de núcleo de origen virtual adicional se adjuntan por el respectivo primer encaminador al respectivo paquete de datos dependiendo de una asignación de ruta programada almacenada en dicha memoria, y
- difundir eventos entrantes a todos los circuitos de memoria en el respectivo circuito de núcleo en paralelo por el respectivo primer encaminador,
- distribuir eventos generados por circuitos de núcleo o dirigidos a circuitos de núcleo dentro de la misma unidad por el respectivo segundo encaminador de nivel intermedio de acuerdo con la dirección de destino adjuntada al respectivo paquete de datos, y
- distribuir eventos entre diferentes unidades por medio del respectivo tercer encaminador de nivel superior de acuerdo con la dirección de destino adjuntada al respectivo paquete de datos.

De acuerdo con una realización, en este punto se define una estructura de encaminamiento para encaminar eventos en una red que comprende una pluralidad de circuitos de núcleo interconectados, comprendiendo cada circuito de núcleo una matriz electrónica que comprende una pluralidad de nodos de cálculo y una pluralidad de circuitos de memoria (por ejemplo, usando memoria local para discriminar eventos de entrada o eventos entrantes) cuya matriz está configurada para recibir eventos entrantes, en la que cada nodo de cálculo está configurado para generar un evento que comprende un paquete de datos si los eventos entrantes recibidos por el respectivo nodo de cálculo satisfacen un criterio predefinido, comprendiendo la estructura de encaminamiento:

- una pluralidad de primeros encaminadores locales (R1) para proporcionar conectividad intra circuito de núcleo rápida, en la que un primer encaminador local (R1) está configurado para asignarse a cada circuito de núcleo, en la que el respectivo primer encaminador (R1) está configurado para
- distribuir paquetes de datos de eventos cuyos origen y destinos se ubican en el mismo circuito de núcleo al que se asigna el respectivo primer encaminador,
- una pluralidad de controladores de difusión, en la que cada controlador de difusión está configurado para asignarse a uno de los circuitos de núcleo y para entregar eventos entrantes a todos los circuitos de memoria en su circuito de núcleo asociado en paralelo,

- una pluralidad de segundos encaminadores (R2) configurados para proporcionar conectividad inter núcleo, en la que particularmente los segundos encaminadores están configurados para distribuir eventos de acuerdo con la carga útil de ruta comprendida por el respectivo paquete de datos, y
- una pluralidad de terceros encaminadores (R3) configurados para proporcionar conectividad inter chip (o inter unidad o inter pieza), en la que particularmente los terceros encaminadores están configurados para distribuir eventos de acuerdo con la carga útil de ruta comprendida por el respectivo paquete de datos.

Las características anteriormente descritas y otras características, aspectos y ventajas de la presente invención se entenderán con referencia a la siguiente descripción, reivindicaciones y figuras adjuntas.

La Figura 1 muestra un diagrama de visión general que ilustra la estructura de una red de múltiples núcleos de ejemplo, de acuerdo con una realización de la invención.

La Figura 2 muestra un diagrama de los procesos instanciados en el encaminador de núcleo R1 por un evento generado dentro del correspondiente núcleo, de acuerdo con una realización de la invención.

La Figura 3 muestra un diagrama de los procesos instanciados en el encaminador de chip R2 por un evento generado dentro de uno de los núcleos de chip, de acuerdo con una realización de la invención.

La Figura 4 muestra un diagrama de los procesos instanciados en el encaminador de chip R2 por un evento generado por una neurona dentro de cualquiera de los chips interconectados, de acuerdo con una realización de la invención.

La Figura 5 muestra un diagrama de los procesos instanciados en el encaminador de malla R3 por un evento generado por una neurona dentro del correspondiente chip, de acuerdo con una realización de la invención.

La Figura 6 muestra un diagrama de los procesos instanciados en el encaminador de malla R3 por un evento generado por una neurona dentro de cualquiera de los chips interconectados que llegan al puerto norte o sur de R3, de acuerdo con una realización de la invención.

La Figura 7 muestra un diagrama de los procesos instanciados en el encaminador de malla R3 por un evento generado por una neurona dentro de cualquiera de los chips interconectados que llegan al puerto este u oeste de R3, de acuerdo con una realización de la invención.

La Figura 8 muestra un diagrama que ilustra la arquitectura de una red neural, de acuerdo con una realización de la invención.

La Figura 9 muestra un diagrama que ilustra una multitud de píxeles de neurona y el flujo de señales difundidas a los mismos desde fueran del núcleo, de acuerdo con una realización de la invención.

La Figura 10 muestra el diagrama de bloques de un núcleo y el flujo de señales cuando se generan impulsos dentro de ese núcleo, de acuerdo con una realización de la invención.

La Figura 11 muestra el diagrama de bloques de un núcleo y el flujo de señales cuando se reciben impulsos por ese núcleo, de acuerdo con una realización de la invención.

La Figura 12 muestra el diagrama de bloques de un núcleo y el flujo de señales usadas para programar la memoria de núcleo y para configurar las neuronas, de acuerdo con una realización de la invención.

La Figura 13 muestra una malla de estructuras jerárquicas con tamaño de tres niveles de jerarquía y ramificaciones (árboles), de acuerdo con una realización de la invención.

La Figura 14 muestra el diagrama de bloques de un ejemplo de chip con 64 núcleos organizados en una estructura jerárquica y el flujo de comunicación, de acuerdo con una realización de la invención.

La Figura 15 muestra el flujo de comunicación entre núcleos y encaminadores y detalles de la estructura de paquete, de acuerdo con una realización de la invención.

La Figura 16 muestra un ejemplo de encaminamiento de un evento desde la neurona de origen a neuronas de destino, de acuerdo con una realización de la invención.

La Figura 17 muestra un dibujo que ilustra una pluralidad de nodos de cálculo dispuestos en módulos.

Realizaciones de la invención proporcionan arquitectura neural accionada por eventos implementable en VLSI con estructuras de memoria distribuida y memoria heterogénea para redes neurales escalables. La arquitectura de

encaminadores jerárquica proporciona una estrategia de potencia y tiempo eficiente para la interconexión de nodos dentro de y entre múltiples núcleos distribuidos en chips de múltiples núcleos. La memoria distribuida en núcleos y los eventos que se difunden en cada núcleo proporcionan una gran distribución de salida para implementar grandes redes neurales con restricciones estructurales típicas de modelos biológicamente plausibles. Los encaminadores totalmente asíncronos y la estructura de programación permiten operaciones rápidas de computación sináptica para aprendizaje fuera de línea inmediato.

El término neurona y sinapsis como se usan en este documento representan circuitos para estimular neuronas biológicas y sinapsis. La neurona electrónica suma contribuciones de sinapsis relativas para producir eventos de impulsos. Un sistema neuromórfico que comprende neuronas electrónicas y sinapsis de acuerdo con realizaciones de la invención puede incluir diversos elementos de procesamiento que se modelan en neuronas biológicas. Ciertas realizaciones ilustrativas de la invención se describen en este documento usando neuronas analógicas y módulos de CAM para almacenar conectividad sináptica. La presente invención no se limita a elementos de cálculo de neurona y sinapsis. El sistema informático accionado por eventos de acuerdo con realizaciones de la invención puede usar nodos de cálculo asíncronos arbitrarios que procesan múltiples eventos de entrada para producir un único evento de salida. Adicionalmente, la presente invención soporta cualquier tipo de cálculo basado en eventos de señal mixta masivamente paralela que requiere una gran distribución de salida para compartición de información.

De acuerdo con realizaciones de la invención, las implementaciones de circuito totalmente asíncrono se usan para encaminadores, pero la presente invención no se limita a tales implementaciones.

De acuerdo con una realización de la invención (consúltese, por ejemplo, la Figura 1), una red neural comprende una pluralidad de chips de múltiples núcleos 6. Cada chip 6 comprende una multitud de núcleos 10 con elementos neurales y sinápticos que almacenan la conectividad de la red y, por lo tanto, realizan una forma particular de cálculo neural.

Cada núcleo 10 comprende una matriz 9 de neuronas, una matriz de sinapsis 8 (matrices 8, 9 las cuales pueden formar parte de una matriz electrónica 8, 9), con múltiples sinapsis 80 (o circuitos de memoria 80) por neurona 90 (o por nodo de cálculo 90), una memoria SRAM 2 para almacenar una LUT de destino 3, y un (primer) encaminador de núcleo R1. Adicionalmente, cada chip 6 también comprende un (segundo) encaminador de chip R2, y un (tercer) encaminador de malla, R3. Cada neurona 90 integra múltiples eventos recibidos y aceptados por las correspondientes sinapsis 80, y genera eventos de impulso cuando la señal integrada cruza un umbral de activación. El impulso producido por una neurona 90 se codifica como un evento de dirección digital, que representa la identidad del origen, por el codificador del núcleo y se transmite a R1. De acuerdo con la información de destino almacenada en su LUT local 3, R1 decide si procesar y entregar el evento adicionalmente a R2 o de vuelta al núcleo. Adicionalmente, R1 puede generar una distribución de salida desde ese evento, es decir, pueden generarse hasta 4 eventos y asignarse diferentes destinos según se programa en la SRAM de LUT 2. Cuando un encaminador R2 recibe un evento de impulso desde cualquiera de los encaminadores de núcleo R1, comprueba las direcciones de destino y decide si entregar los eventos de vuelta a los correspondientes encaminadores de núcleo o adicionalmente a R3 en consecuencia. Cada sinapsis 80 tiene una palabra de CAM de n bits para almacenar la dirección de la neurona de origen 90 a la que se conecta, el tipo de sinapsis y su eficacia sináptica. Una vez que se envía un evento al núcleo 10 específico, la dirección se difunde a todas las sinapsis 80 dentro del núcleo 10. Las sinapsis 90 cuya dirección almacenada coincide con la dirección difundida generan una PSC con los parámetros datos de tipo de sinapsis y eficacia sináptica a la correspondiente neurona postsináptica 90. Obsérvese que la distribución de salida se genera (1) cuando un evento sale de un núcleo 10 y alcanza R1, dependiendo de cómo está programada la memoria de R1 para ese evento, y (2) cuando un evento alcanza el núcleo 10 desde R1. El esquema descrito soporta redes altamente interconectadas que requieren compartición de información en distancias cortas y largas.

La Figura 2 muestra detalles de proceso adicionales del encaminador R1 de acuerdo con una realización de la invención. Cuando una neurona 90 genera un impulso, la dirección de este impulso se envía a R1. El destino para este evento se almacena en una LUT 3 a la que R1 puede acceder. De acuerdo con el destino programado para el evento, R1 puede enviar el evento de vuelta al núcleo 10 o adjuntar al paquete de evento un código digital que representa el destino para este evento. En el ejemplo dado, un impulso desde el núcleo 10 se representa como un evento de dirección de 8 bits. La dirección de destino adjuntada consta de un código de 12 bits, incluyendo 6 bits para desplazamiento de chips (dx, dy), 2 bits para ID de núcleo de origen virtual y 4 bits para destinos dentro de chip. El ID de núcleo de origen virtual es un código adicional adjuntado a la dirección de origen independientemente en cada neurona para aumentar el espacio de direcciones total, para aumentar la discriminación de eventos y, por lo tanto, para reducir la ambigüedad de direcciones, sobre una base por neurona en lugar de una base por núcleo. Por ejemplo, a la neurona 243 del núcleo 1 puede asignarse ID virtual = 1, a la neurona 121 del mismo núcleo puede asignarse un ID virtual diferente, por ejemplo, 2. Un evento de impulso también puede generar distribución de salida según se programa en la memoria SRAM 2, asignándose a cada evento un destino diferente, pero transportado la misma dirección de origen.

Cuando R2 recibe un evento desde R1, comprueba si los núcleos 10 objetivo para este evento se ubican dentro del mismo chip 6, como se muestra en la Figura 3. Si este es el caso, R2 entregará el evento de acuerdo con el destino núcleo 10 leído desde el paquete de evento. De lo contrario, este evento se entregará a R3. R2 también recibe eventos desde diferentes chips 6 a través de R3 como se muestra en la Figura 4. Cuando sucede esto, enviará el evento a las

ramas de acuerdo con el código de destino adjuntado a la dirección de origen.

La Figura 5 muestra los detalles del procesamiento de encaminador R3 cuando recibe un evento desde R2, de acuerdo con una realización de la invención. En este ejemplo, R3 comprobará primero el número de desplazamiento de dirección x (este-oeste). Si el desplazamiento x no es 0, R3 comprobará el signo de dx para decidir la dirección de entrega, este para dx>0 y oeste para dx<0. A continuación, dx se disminuye en 1 y el evento se entrega a la dirección correcta. Si el desplazamiento x es 0, R3 comprobará el signo de dy para decidir la dirección para la entrega, norte para dy>0 y sur para dy<0. A continuación, dy se disminuye en 1 y el evento se entrega a la dirección correcta. Por lo tanto, en este ejemplo se establece una regla de prioridad de tal forma que una vez que un evento se entrega a R3, se desplazará primero a lo largo de la dirección este-oeste y a continuación a lo largo de la dirección norte-sur. El evento viajará a lo largo de la malla hasta que tanto dx como dy sean 0. Como se muestra en la Figura 6, una vez que R3 recibe un evento desde sur/norte (dirección y) se puede suponer que el evento no necesita desplazarse adicionalmente a lo largo de la dirección x, como sostiene la regla de prioridad anterior. Por lo tanto, una vez que el valor de dy es 0, el evento se entrega a R2 en el correspondiente chip.

Se usan palabras de CAM como las sinapsis 80 para almacenar conexiones de neurona y eficacias sinápticas. En un ejemplo, un núcleo 10 tiene 256 neuronas con 64 sinapsis basadas en CAM por neurona. Cada palabra de CAM se compone de 12 bits: 10 bits para dirección de origen y 2 bits para tipo sináptico. Los eventos de impulsos que llegan a un núcleo se difunden a todo el núcleo por el controlador de difusión. Cada CAM compara el evento en el bus de difusión con el contenido almacenado. Aquellos para los que el contenido coincide con el evento difundido establecerán un "estado coincidente" y generarán la Corriente Postsináptica (PSC) apropiada. En la Figura 9 la sinapsis 4 de la neurona 1, la sinapsis 1 de la neurona 16 y la sinapsis 5 de la neurona 5 almacenan una dirección que coincide con la entregada por el controlador de difusión y, por tanto, establecen una respuesta que genera corrientes apropiadas en las correspondientes neuronas.

La Figura 10 muestra el proceso de emisión de un evento para impulsos generado dentro de la matriz de neuronas 9. Los impulsos generados por las neuronas 90 se codificarán como las direcciones de neuronas. Por ejemplo, para una matriz de neuronas de 16x16, los impulsos pueden codificarse como 8 bits con 4 bits para columna y 4 bits para fila por un codificador de columnas 5 y codificador de filas 4. Este evento se enviará al encaminador de núcleo R1 primero para conseguir direcciones de destino e ID de núcleo de origen adicional leyendo la LUT de SRAM de destino como se ha explicado anteriormente.

La Figura 11 muestra el proceso de difusión de un evento una vez que un evento se envía a un núcleo 10 particular. El evento recibido se recibe primero por R1 y a continuación se difunde a la matriz de sinapsis 8 por controlador de difusión 7. El controlador de difusión 7 entregará los impulsos a todas las CAM en el núcleo 10, que, a continuación, discriminará la dirección de origen del evento y generará una PSC en consecuencia.

La Figura 12 muestra un ejemplo de la programación de la memoria 3 de un núcleo enviando la dirección y los datos al decodificador de filas 40 y decodificador de columnas 50. Ya que en cada núcleo 10 se usan una estructura de memoria heterogénea basada en CAM y una LUT de SRAM de destino 3 distribuida, los contenidos de CAM/SRAM pueden programarse fácilmente por el decodificador de filas 40 y el decodificador de columnas 50 usando operaciones de escritura estándar para CAM/SRAM.

La Figura 13 muestra otro ejemplo de una red jerárquica o estructura de encaminamiento con estructuras de memoria heterogéneas. La estructura de encaminamiento combina una malla bidimensional de estructuras de árbol 6, cuyas hojas son núcleos de múltiples neuronas 10. De acuerdo con una realización de la invención, cada núcleo 10 tiene un (primer) encaminador de núcleo R1 para entrada/salida de impulsos (por ejemplo, conectividad intra núcleo). Los eventos entregados desde un núcleo 10 al otro núcleo 10 en la misma rama 60 se enviarán primero a un encaminador de rama de nivel bajo (o segundo encaminador) R2 y, a continuación, se enviarán a uno o más núcleos 10 objetivo de acuerdo con el código de destino transportado por el evento de dirección. Los eventos entregados desde un núcleo 10 a uno o más otros núcleos 10 en diferentes ramas, pero dentro del mismo chip 6, se enviarán primero a (segundos) encaminadores R2 superiores, a continuación a encaminadores R2 de ramas inferiores según se codifica en el código de destino transportado por el evento. La profundidad del árbol y el número de diferentes núcleos dentro del mismo chip que pueden dirigirse dependiendo del número de bits transportados por el evento de dirección como código de dirección. En un ejemplo, cada chip 6 tiene un encaminador de pieza (o tercer encaminador) R3 para interconectar tales estructuras de árbol en una malla bidimensional.

La Figura 14 muestra un ejemplo de una estructura de chip que consta de 64 núcleos 10 usando el esquema de encaminamiento jerárquico descrito en la Figura 13. En este ejemplo, cada núcleo 10 tiene un (primer) encaminador R1 especializado para conexiones de núcleo locales. Un grupo de 4 núcleos se define como la ramificación de nivel inferior de la estructura de árbol. Grupos de cuatro de estos módulos se definen como rama de dos niveles, incluyendo, por lo tanto, cada uno 16 núcleos 10. El chip 6 de ejemplo consta de un grupo de cuatro de estos módulos (tercer nivel en el árbol), constando, por lo tanto, de un total de 64 núcleos 10. Cada nivel incluye un (segundo) encaminador R2 especializado para comunicaciones de núcleos dentro de este nivel y para enviar/recibir eventos a/desde otros niveles en el árbol 6.

De acuerdo con las direcciones de destino asignadas a cada evento generado dentro de los núcleos 10, eventos que se dirigen a destinos dentro del mismo núcleo 10 se encaminan por el encaminador de núcleo R1, implementando, por lo tanto, conectividad local. Eventos que se dirigen a otros núcleos 10 dentro de la misma rama de un nivel se enviarán al (segundo) encaminador R2 y, a continuación, se procesarán y entregarán a correspondientes núcleos 10. En general, las memorias de encaminador se programan de tal forma que los eventos escalan el árbol a través de encaminadores R2 en diferentes niveles tanto como sea necesario para alcanzar cualquier núcleo 10 de destino dentro del chip 6. Si el objetivo de un evento reside en un núcleo 10 de un chip 6 diferente, el evento se envía a través de todas las capas hasta el (tercer) encaminador de chip R3, que procesará adicionalmente y entregará el mismo a lo largo de las direcciones apropiadas en la malla.

La Figura 15 muestra detalles de la estructura de un núcleo 10 en la red. En este ejemplo, un chip de múltiples núcleos 6 tiene 4 núcleos 10 con 256 neuronas 90 en cada núcleo 10 y 64 sinapsis 80 para cada neurona 90. Cada núcleo 10 incluye: sinapsis/matriz de neuronas 8, 9, codificador de columnas/filas 5, 4, controlador de difusión 7, LUT de destino 3 y (primer) encaminador de núcleo local R1. Los eventos generados por la matriz de neuronas 9 se codificarán y presentarán como direcciones de 8 bits (4 bits para columna y 4 bits para dirección de fila). Los eventos generados por la matriz de neuronas 9 se asignarán a dirección de destino de 10 bits (3 bits para la distancia dx y signo de dirección x, 3 bits para dy y signo de dirección y y 4 bits para núcleo 10 objetivo en el chip 6 de destino, es decir, a qué núcleos 10 dirigirse una vez que se ha alcanzado el chip dx-dy) y 2 bits para ID de núcleo de origen adicional que aumenta la discriminación de dirección de origen. Cada evento puede replicarse varias veces y a cada replica puede asignarse diferentes direcciones de destino por el encaminador de núcleo R1. En una realización de la invención, se adjuntan internamente 2 bits a la dirección de origen por la dirección de núcleo y se usan para leer la LUT (3) cuatro veces. De esta manera, diferentes direcciones de destino de 12 bits e ID de núcleo se adjuntan a cada réplica. La dirección de evento de origen de 8 bits con los datos de 12 bits leídos de la LUT 3 se envían como un único paquete de un evento de 20 bits antes de su entrega al (segundo) encaminador R2. Un evento que alcanza a un encaminador R2 desde el (tercer) encaminador R3 se difunde a los núcleos de destino como se describe en la Figura 9.

La Figura 16 muestra un ejemplo de encaminamiento de un evento desde el nodo de cálculo de origen a los nodos de destino. Siguiendo el esquema de múltiples niveles jerárquicos como se describe en la Figura 13, se asigna un destino específico a un evento generado por un nodo 90 en un chip 6 cualquiera. El evento también se replica múltiples veces y a cada réplica se asigna un destino diferente. En el ejemplo, al evento generado por el nodo 90 indicado con un círculo abierto se asignan cuatro destinos diferentes y se envía a cuatro múltiples chips 6. Uno de los eventos de réplica se entrega a otro núcleo 10 en el mismo chip 6 (por lo tanto, su ruta es R1-R2-R1). Los otros tres eventos se entregan a diferentes chips 6 y, a continuación, se distribuyen localmente en múltiples núcleos 10 (por lo tanto, sus rutas son R1-R2-R3-...-R3-R2-R1).

La Figura 17 muestra otro ejemplo de una estructura de encaminamiento jerárquica o red con estructuras de memoria heterogéneas. En este ejemplo, una arquitectura informática autosimilar consta de una pluralidad de nodos de cálculo 90 basados en eventos que acceden a una memoria local 80 para implementar el cálculo requerido e interconectados por una jerarquía de sistemas de encaminamiento (R1, R2, R3) de naturaleza heterogénea. Los nodos 98 se disponen en núcleos 10 que constan de dicha pluralidad de nodos 90, un (primer) encaminador R1 local y una memoria de encaminador 3 accedida por dicho encaminador R1. Grupos de núcleos 10 se disponen en piezas (o chips o unidades) 6 que constan de dicho grupo de núcleos 10, un (segundo) encaminador R2 local y una memoria de encaminador 32 accedida por dicho encaminador R2. Dichos módulos de pieza 6 se combinan arbitrariamente para formar una estructura autosimilar en la que los encaminadores R2, R3 usan direcciones de origen como apuntadores a entradas de memoria local 3, 32, 33 usadas para adjuntar datos de ruta en el paquete para transmitirse adicionalmente. Adicionalmente, dichos encaminadores R2, R3 se programan para procesar eventos entregados desde niveles superior y para encaminar los mismos a destinos de nivel inferior objetivo de acuerdo con los datos de ruta contenidos en el paquete.

REIVINDICACIONES

1. Una red que comprende una pluralidad de circuitos de núcleo interconectados (10), en la que cada circuito de núcleo (10) comprende:

- una matriz electrónica (8, 9) que comprende una pluralidad de nodos de cálculo (90) y una pluralidad de circuitos de memoria (80), cuya matriz electrónica (8, 9) está configurada para recibir eventos entrantes, en la que cada nodo de cálculo (90) de dicha pluralidad de nodos de cálculo (90) están configurados para generar un evento que comprende un paquete de datos cuando los eventos entrantes recibidos por el respectivo nodo de cálculo (90) satisfacen un criterio predefinido, y
- un circuito que está configurado para adjuntar una dirección de destino e información de origen adicional al respectivo paquete de datos, y
- un primer encaminador local (R1) para proporcionar conectividad intra núcleo y/o entregar eventos a un segundo encaminador de nivel intermedio (R2) para conectividad inter núcleo y a un tercer encaminador de nivel superior (R3) para conectividad inter unidad, y
- un controlador de difusión (7) para difundir eventos entrantes a todos los circuitos de memoria (80) en el circuito de núcleo (10) en paralelo,

caracterizado por que
la red comprende además un sistema de encaminamiento de eventos (R1, R2, R3) que interconecta los circuitos de núcleo (10), en la que el sistema de encaminamiento de eventos comprende dichos primeros encaminadores locales (R1) y el segundo encaminador de nivel intermedio y tercer encaminador de nivel superior (R2, R3), en la que la totalidad de los encaminadores (R1, R2, R3) forman una estructura jerárquica.

2. La red de acuerdo con la reivindicación 1, en la que cada uno de dichos primeros encaminadores locales, segundos encaminadores de nivel intermedio y terceros encaminadores de nivel superior (R1, R2, R3) comprende al menos un circuito de control que está configurado para encaminar eventos de acuerdo con la carga útil de ruta que es parte del respectivo paquete de datos, y en la que cada uno de dichos primeros encaminadores locales (R1) comprende además: una memoria programable configurada para almacenar carga útil de ruta e ID de núcleo de origen virtual para dichos paquetes de datos, y al menos un circuito configurado para adjuntar carga útil de ruta e ID de núcleo de origen virtual al respectivo paquete de datos, dependiendo de asignaciones de ruta programadas almacenadas en dicha memoria.

3. La red de acuerdo con una de las reivindicaciones anteriores, comprendiendo adicionalmente unidades, en la que los circuitos de núcleo (10) se disponen en las unidades (6), en la que cada una de las unidades (6) comprende varios de los circuitos de núcleo (10), y en la que cada una de las unidades (6) comprende uno de los primeros encaminadores locales (R1), en la que cada uno de los dichos primeros encaminadores (R1) está dispuesto en la estructura jerárquica formada por la totalidad de los encaminadores (R1, R2, R3).

4. La red de acuerdo con la reivindicación 1 o 3, en la que dicha estructura jerárquica comprende un nivel inferior que comprende los primeros encaminadores locales (R1), en la que los primeros encaminadores locales (R1) están configurados para distribuir eventos cuyos origen y destinos se ubican en el mismo circuito de núcleo (10).

5. La red de acuerdo con una de las reivindicaciones 1, 3 y 4, en la que dicha estructura jerárquica comprende al menos un nivel intermedio que comprende los segundos encaminadores de nivel intermedio (R2), en la que los segundos encaminadores (R2) están configurados para distribuir eventos generados por o dirigidos a circuitos de núcleo (10) dentro de la misma unidad.

6. La red de acuerdo con una de las reivindicaciones 1, 3, 4 y 5, en la que dicha estructura jerárquica comprende un nivel superior que comprende los terceros encaminadores de nivel superior (R3) que están configurados para distribuir eventos entre diferentes unidades (6), en la que los terceros encaminadores de nivel superior (R3) se disponen en una malla bidimensional.

7. La red de acuerdo con una de las reivindicaciones anteriores, estando la red adaptada para hacer que diferentes esquemas de encaminamiento asíncronos coexistan en diferentes niveles de la estructura jerárquica correlacionando direcciones de origen y dichos paquetes de datos generados por los nodos de cálculo (90) en los circuitos de núcleo (10) para hacer coincidir los diferentes esquemas de encaminamiento en los diferentes niveles.

8. La red de acuerdo con una de las reivindicaciones anteriores, en la que la red es una red neural sintética, en la que cada nodo de cálculo (90) forma una neurona, y en la que cada uno de dichos circuitos de memoria (80) forma una sinapsis, en la que los nodos de cálculo (90) se diseñan para integrar eventos entrantes y para generar un evento cuando la señal formada por los eventos integrados cruza un umbral de activación.

9. Un método para encaminar eventos en una red usando una red de acuerdo con una de las reivindicaciones 1 a 8, en el que el método comprende

- generar un evento que comprende un paquete de datos por medio de un nodo de cálculo (90) cuando los eventos entrantes recibidos por el respectivo nodo de cálculo (90) satisfacen un criterio predefinido, para cada uno de los circuitos de núcleo:

- distribuir el evento generado dentro del respectivo circuito de núcleo (10) por medio del primer encaminador local (R1) del respectivo circuito de núcleo (10), en el que el respectivo primer encaminador local (R1) adjunta una dirección de destino al respectivo paquete de datos dependiendo de una asignación de ruta programada almacenada en una memoria (2, 3), y
- difundir eventos entrantes a todos los circuitos de memoria (80) en el respectivo circuito de núcleo (10) en paralelo por el respectivo primer encaminador local (R1),

- distribuir eventos generados por circuitos de núcleo (10) o dirigir circuitos de núcleo dentro de la misma unidad (6) por el respectivo segundo encaminador de nivel intermedio (R2) de acuerdo con la dirección de destino adjuntada al respectivo paquete de datos, y
- distribuir eventos entre diferentes unidades (6) por el respectivo tercer encaminador de nivel superior (R3) de acuerdo con la dirección de destino adjuntada al respectivo paquete de datos.

Fig. 1

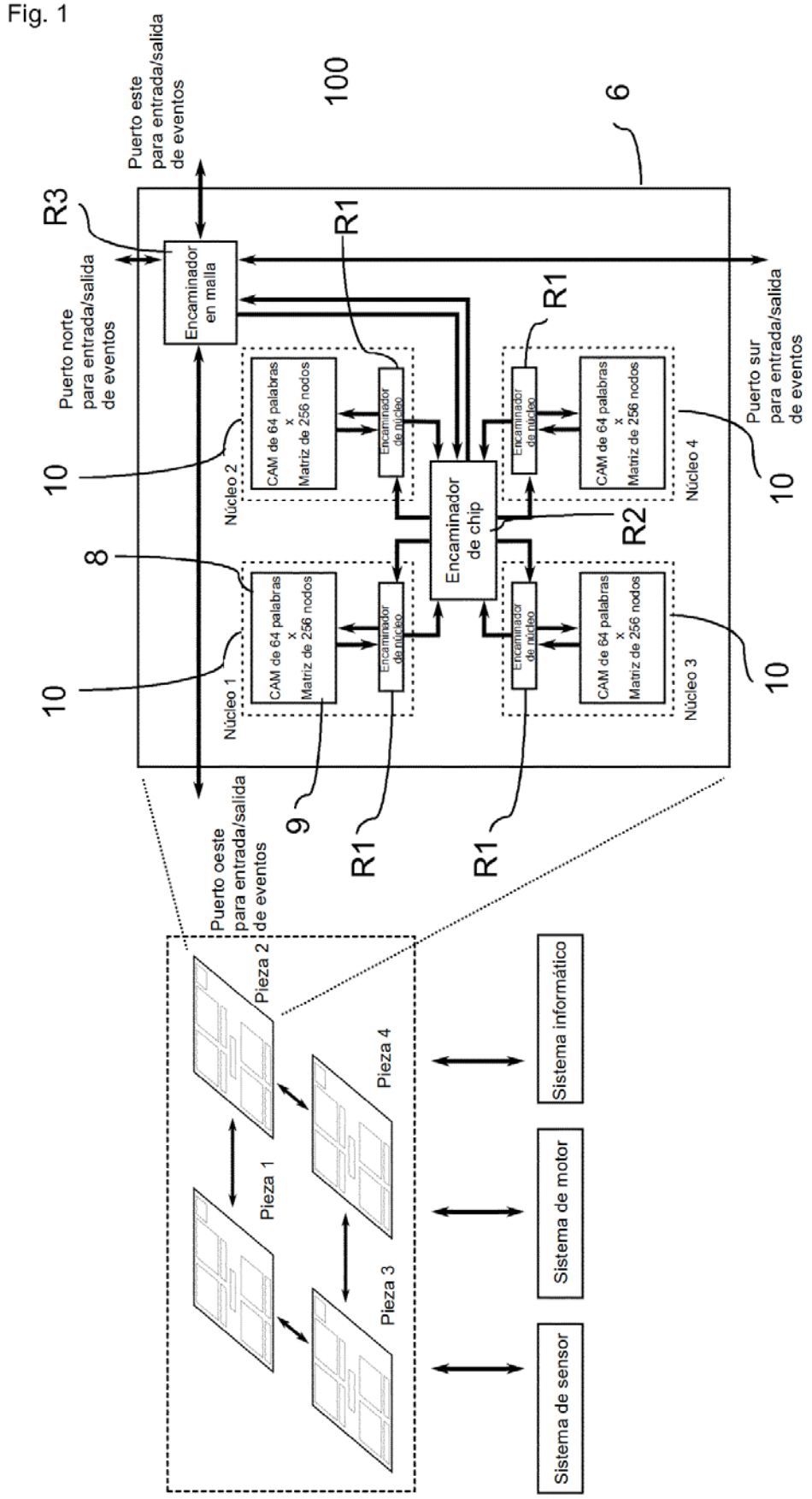
100
6
Puerto este para entrada/salida de eventos
R3
Puerto norte para entrada/salida de eventos
Encaminador en malla
R1
R1
10
8
10
Núcleo 2
CAM de 64 palabras x Matriz de 256 nodos
Encaminador de núcleo
Encaminador de chip
R2
Encaminador de núcleo
CAM de 64 palabras x Matriz de 256 nodos
Núcleo 4
Puerto sur para entrada/salida de eventos
10
Núcleo 1
CAM de 64 palabras x Matriz de 256 nodos
Encaminador de núcleo
Encaminador de núcleo
CAM de 64 palabras x Matriz de 256 nodos
Núcleo 3
10
9
R1
R1
Puerto oeste para entrada/salida de eventos
Pieza 2
Pieza 4
Pieza 1
Pieza 3
Sistema de sensor
Sistema de motor
Sistema informático

Fig. 2

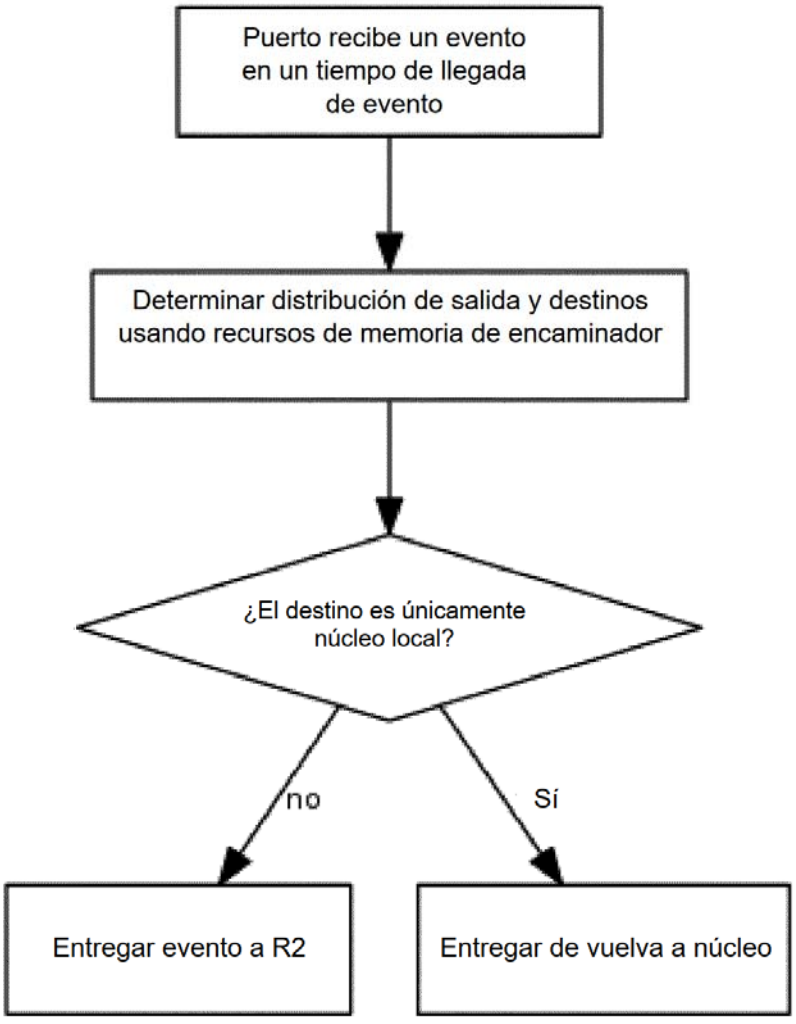
Puerto recibe un evento en un tiempo de llegada de evento
Determinar distribución de salida y destinos usando recursos de memoria de encaminador
¿El destino es únicamente núcleo local?
no
Sí
Entregar evento a R2
Entregar de vuelva a núcleo

Fig. 3

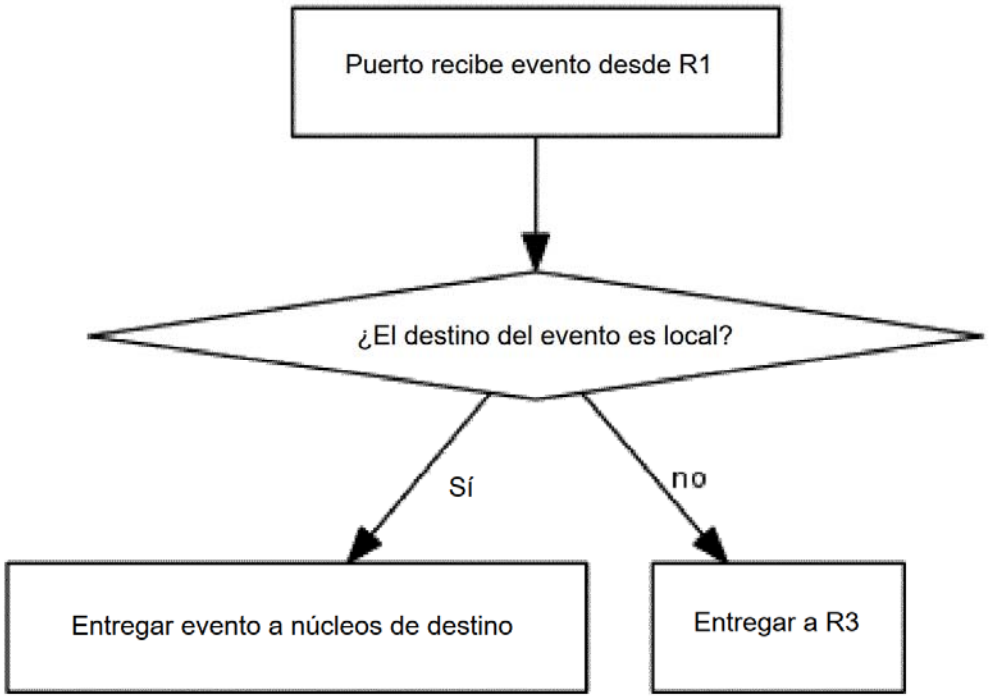
Puerto recibe evento desde R1
¿El destino del evento es local?
Sí
no
Entregar evento a núcleos de destino
Entregar a R3

Fig. 4

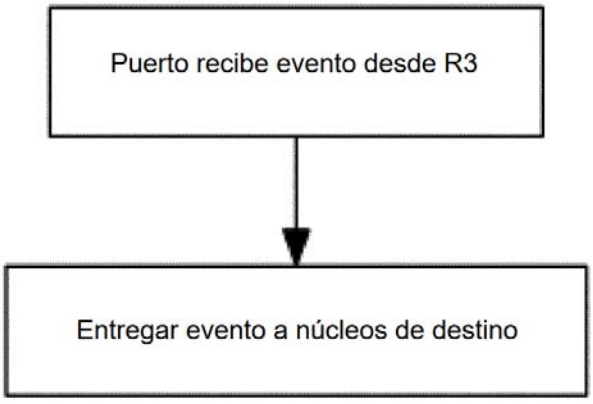
Puerto recibe evento desde R3
Entregar evento a núcleos de destino

Fig. 5

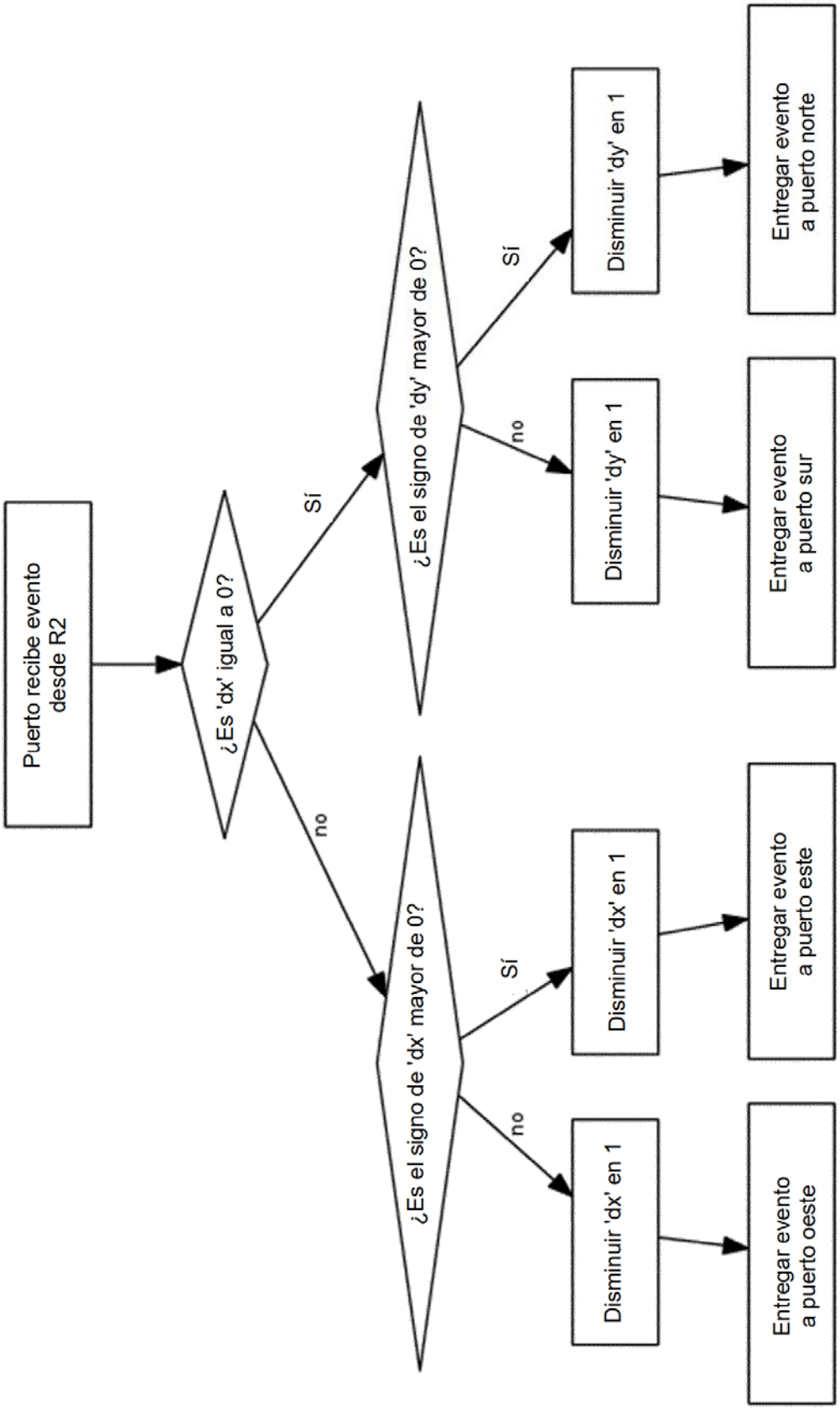
Puerto recibe evento desde R2
¿Es 'dx' igual a 0?
Sí
¿Es el signo de 'dy' mayor de 0?
Sí
Disminuir 'dy' en 1
Entregar evento a puerto norte
no
Disminuir 'dy' en 1
Entregar evento a puerto sur
no
¿Es el signo de 'dx' mayor de 0?
Sí
Disminuir 'dx' en 1
Entregar evento a puerto este
no
Disminuir 'dx' en 1
Entregar evento a puerto oeste

Fig. 6

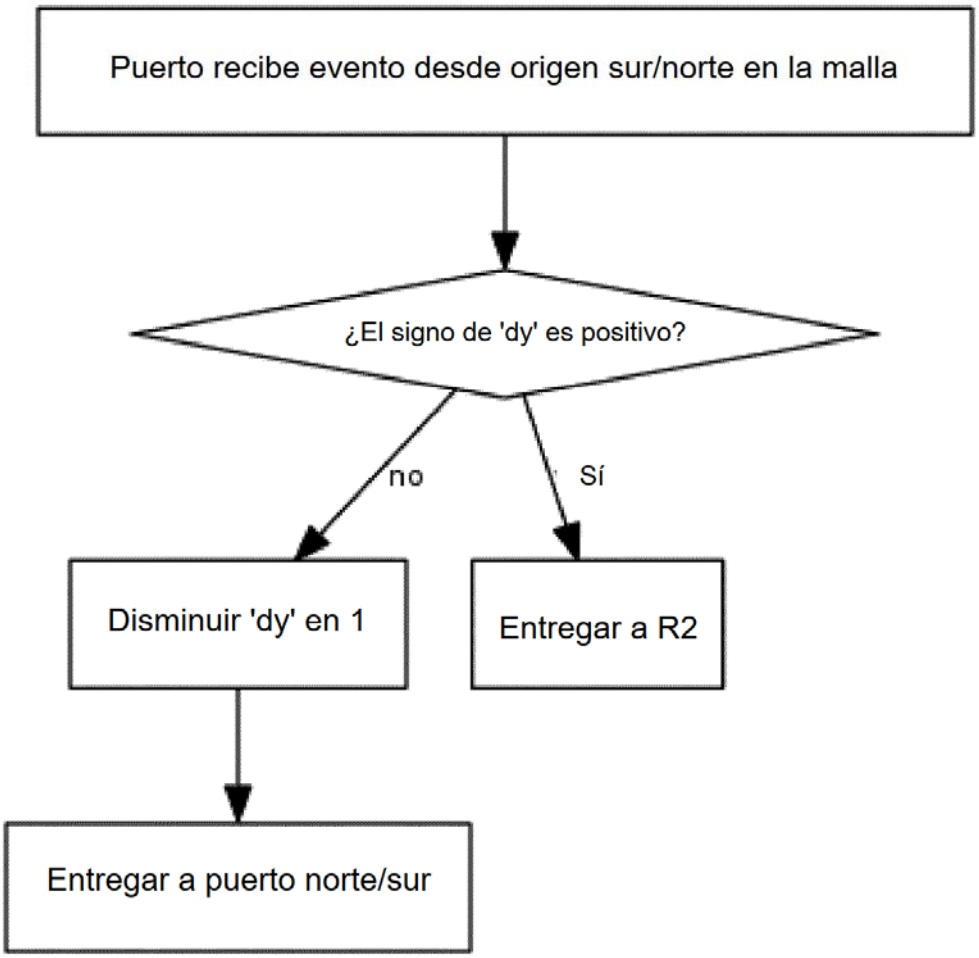
Puerto recibe evento desde origen sur/norte en la malla
¿El signo de 'dy' es positivo?
no
Sí
Disminuir 'dy' en 1
Entregar a R2
Entregar a puerto norte/sur

Fig. 7

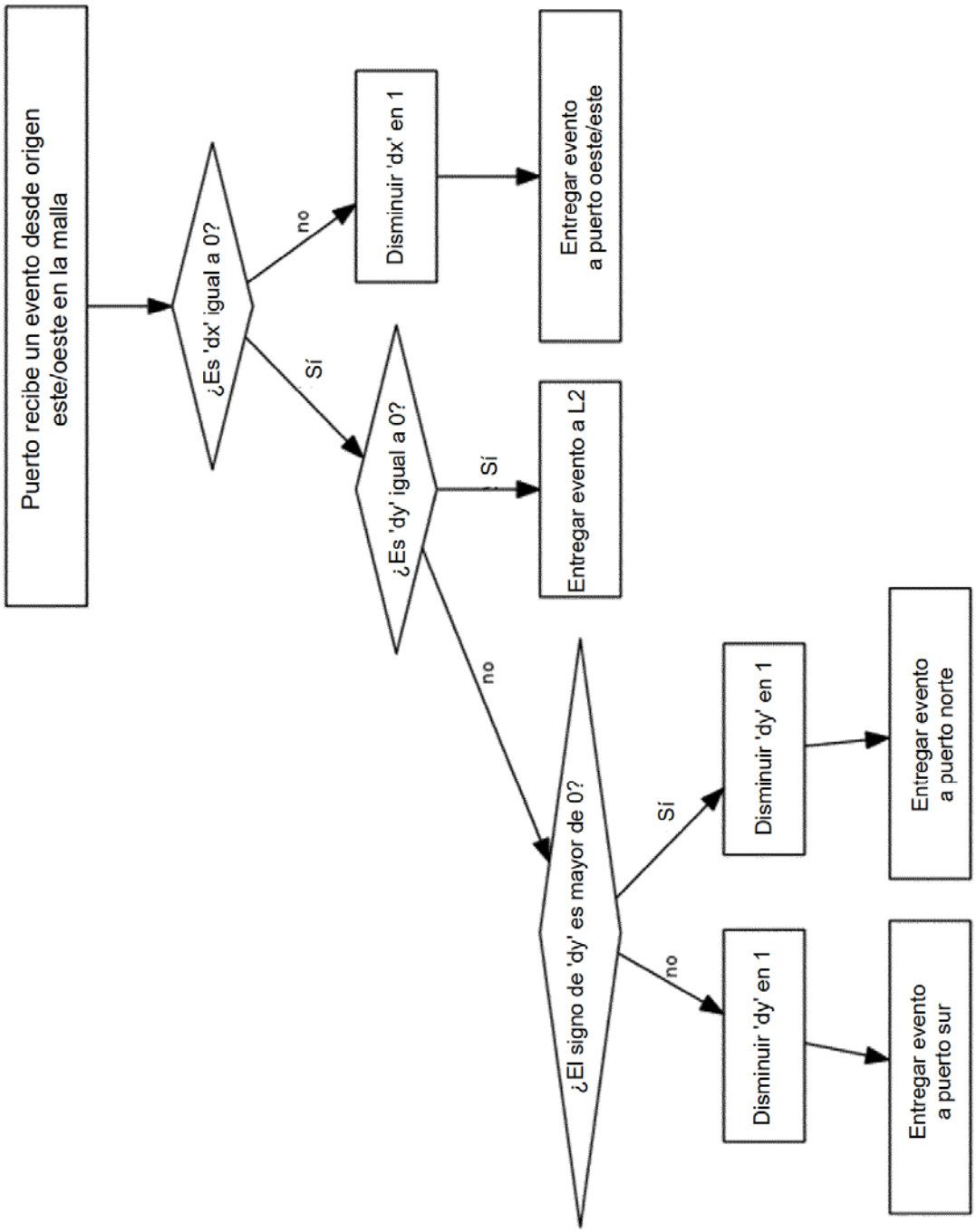
Puerto recibe un evento desde origen este/oeste en la malla
¿Es 'dx' igual a 0?
no
Disminuir 'dx' en 1
Entregar evento a puerto oeste/este
Sí
¿Es 'dy' igual a 0?
Sí
Entregar evento a L2
no
¿El signo de 'dy' es mayor de 0?
Sí
Disminuir 'dy' en 1
Entregar evento a puerto norte
no
Disminuir 'dy' en 1
Entregar evento a puerto sur

Fig. 8

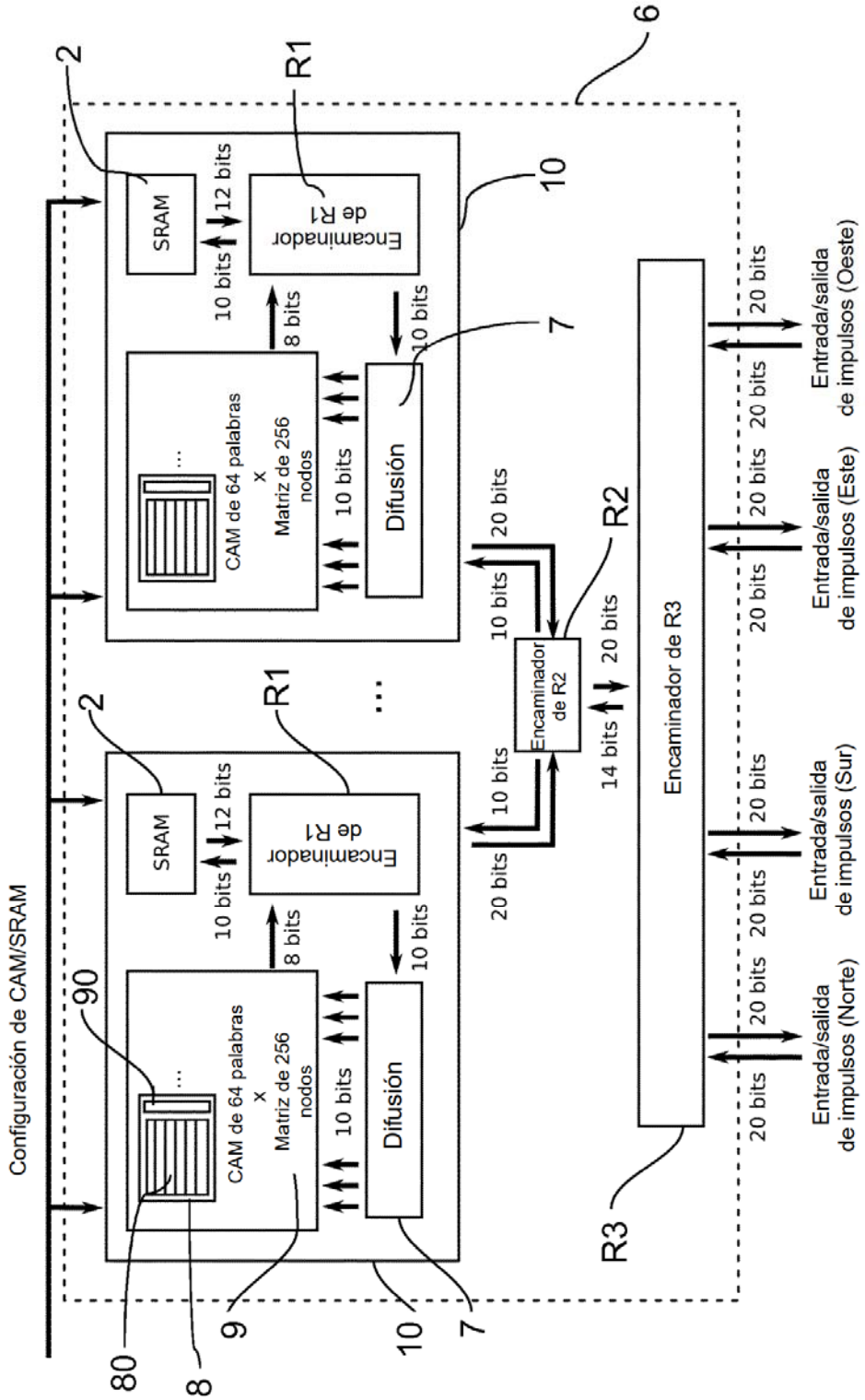
Configuración de CAM/SRAM
SRAM
Encaminador de R1
CAM de 64 palabras x Matriz de 256 nodos
Difusión
Encaminador de R2
Encaminador de R3
Entrada/salida de impulsos (Norte)
Entrada/salida de impulsos (Sur)
Entrada/salida de impulsos (Este)
Entrada/salida de impulsos (Oeste)
10 bits
12 bits
8 bits
20 bits
14 bits
2
6
7
8
9
10
80
90
R1
R2
R3

Fig. 9

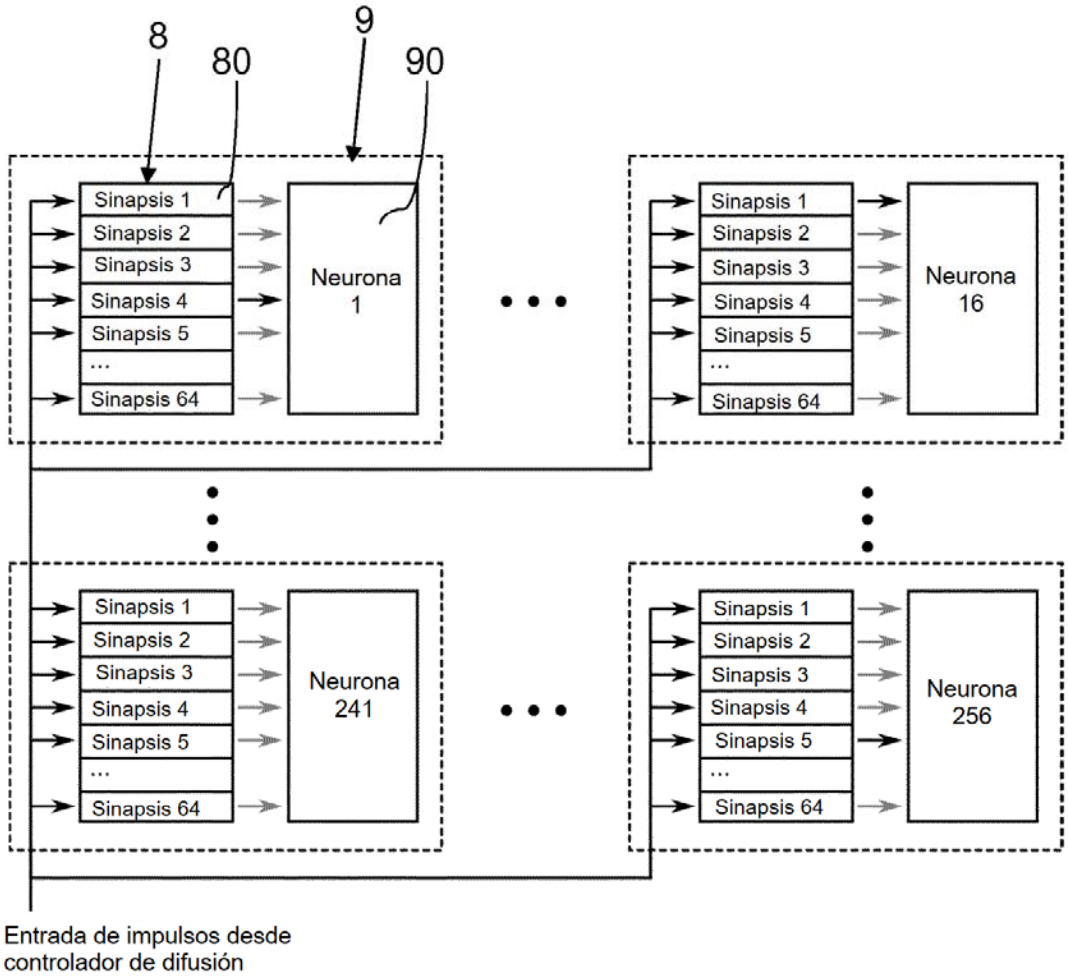
8
80
9
90
Sinapsis 1
Sinapsis 2
Sinapsis 3
Sinapsis 4
Sinapsis 5
...
Sinapsis 64
Neurona 1
Neurona 16
Neurona 241
Neurona 256
Entrada de impulsos desde controlador de difusión

Fig. 10

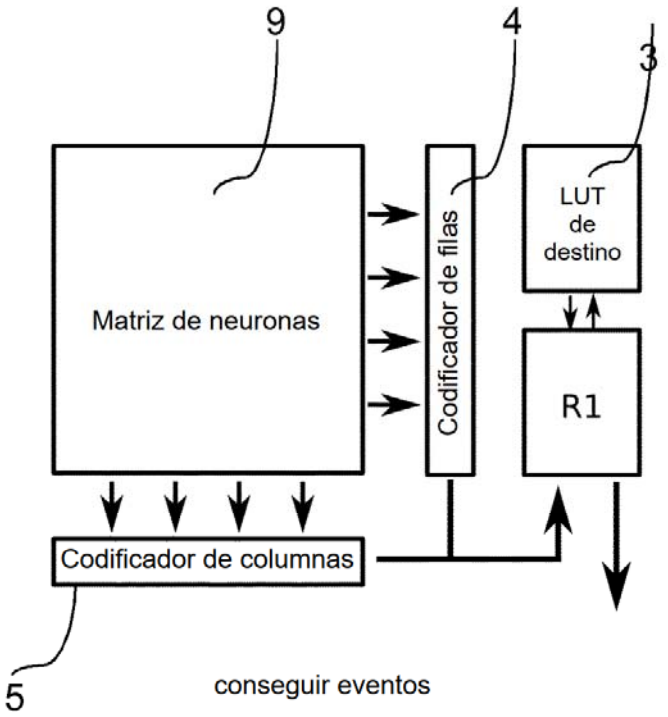
9
4
3
Matriz de neuronas
Codificador de filas
LUT de destino
R1
Codificador de columnas
5
conseguir eventos

Fig. 11

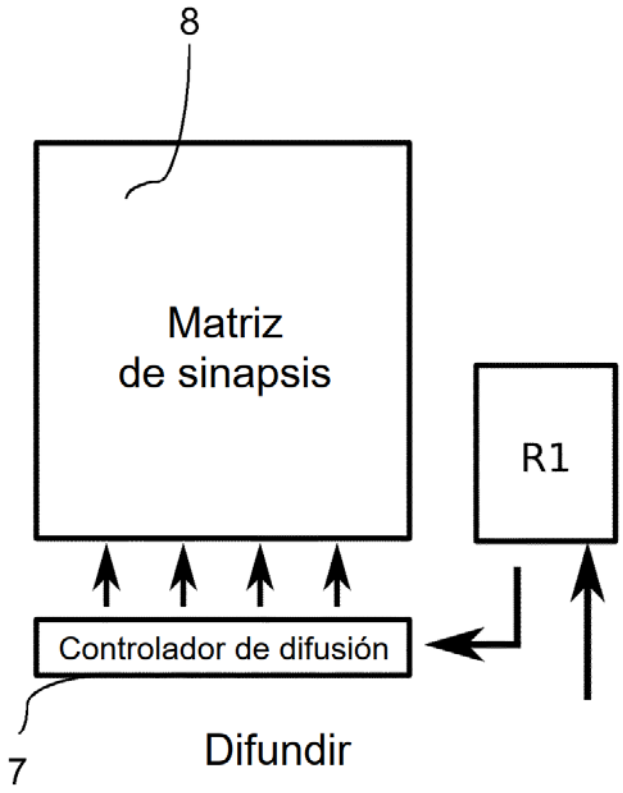
8
Matriz de sinapsis
R1
Controlador de difusión
7
Difundir

Fig. 12

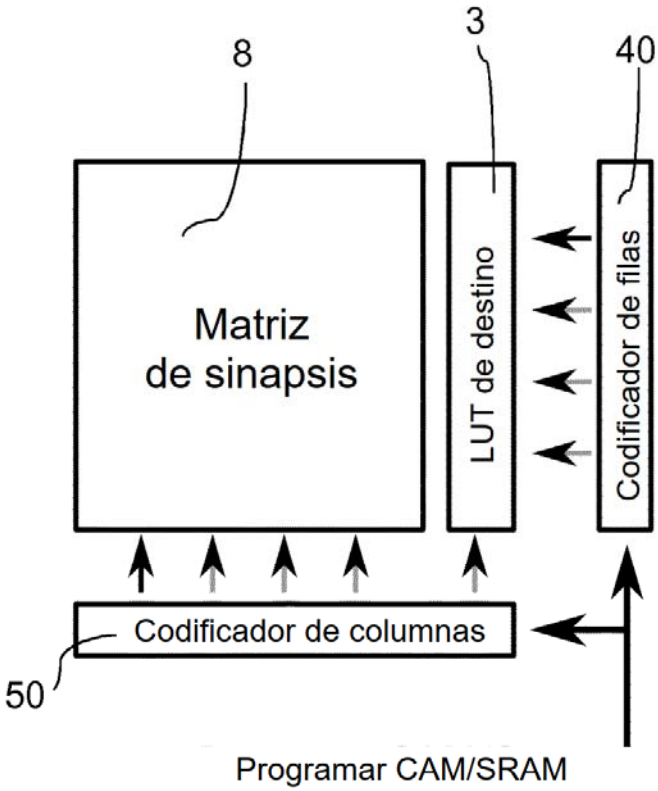
8
3
40
Matriz
de sinapsis
LUT de destino
Codificador de filas
Codificador de columnas
50
Programar CAM/SRAM

Fig. 13

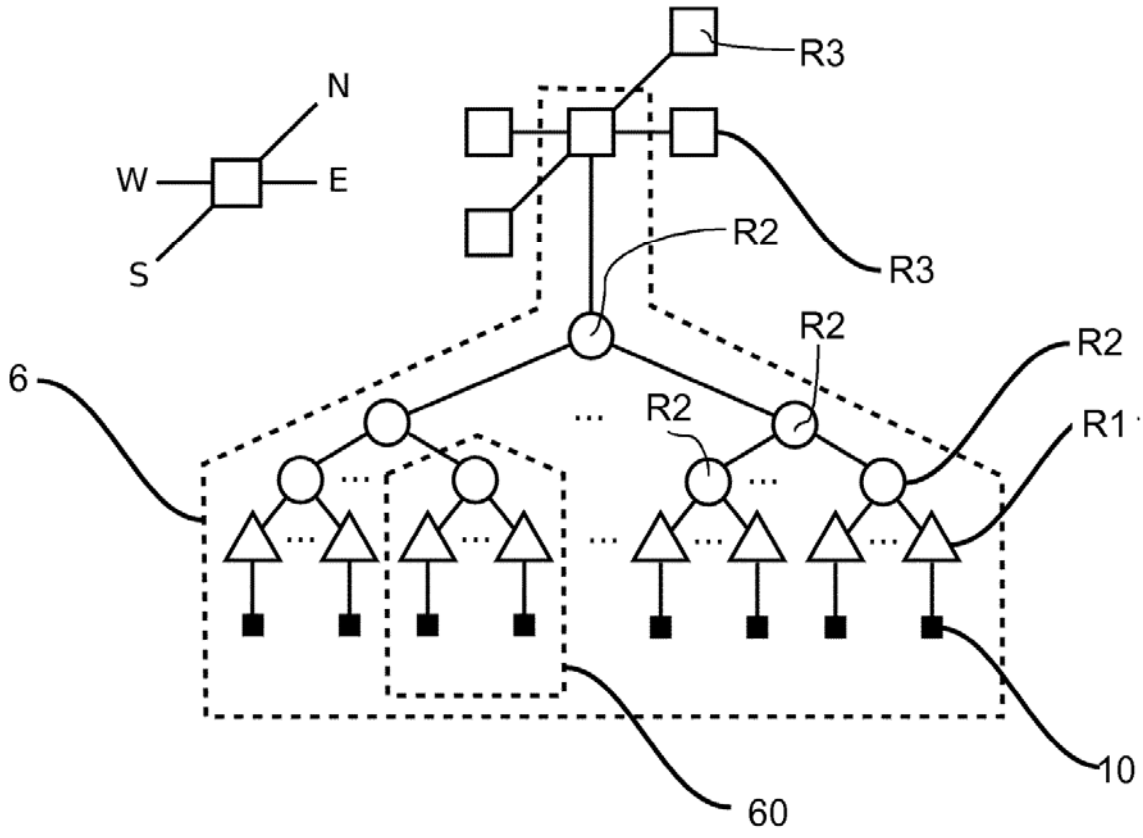
N
W
E
S
R3
R2
R3
R2
R2
R2
R1
6
10
60

Fig. 14

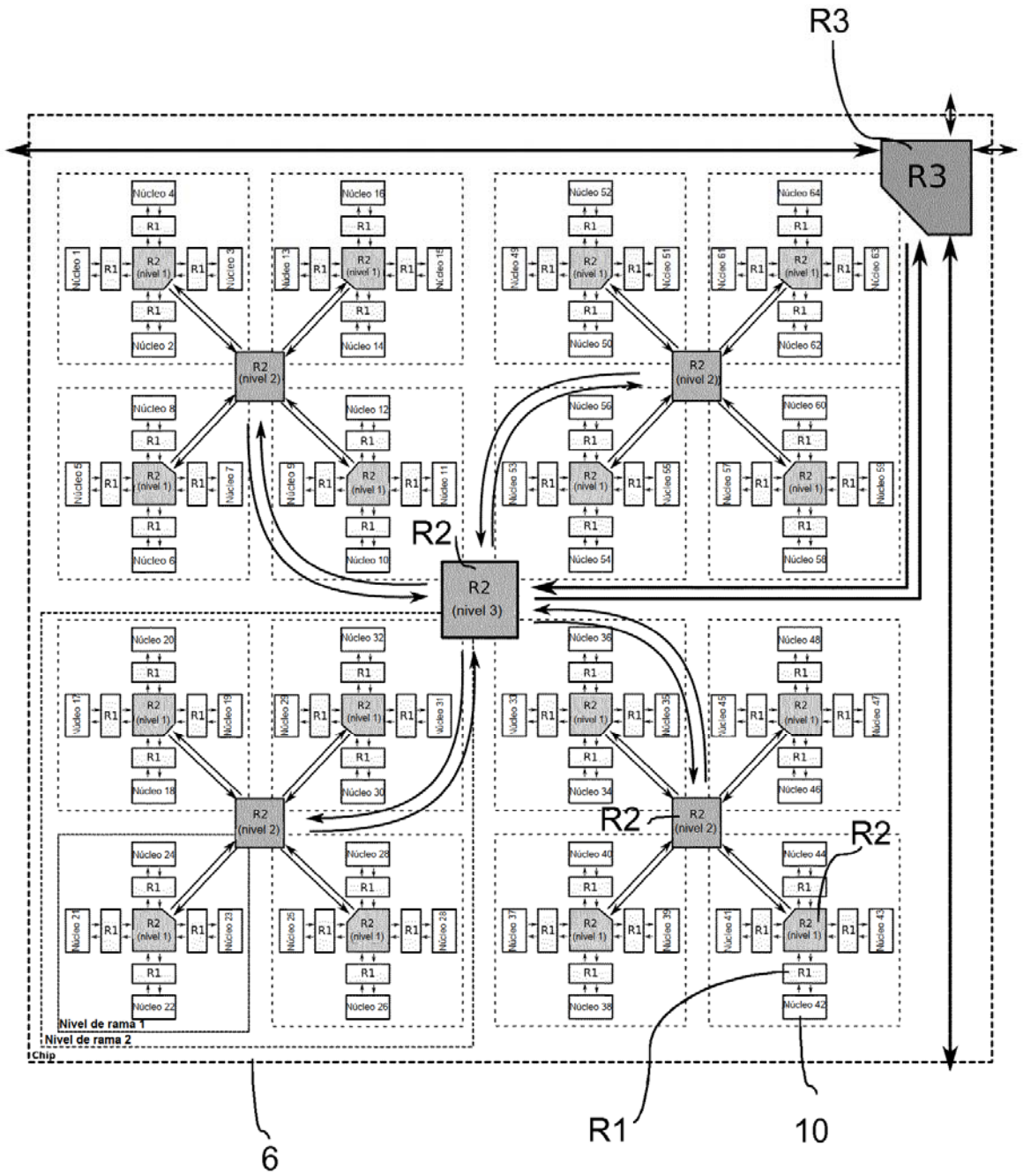
R3
R3
R2
(nivel 3)
R2
(nivel 2)
R2
(nivel 1)
R1
Núcleo 1
Núcleo 2
Núcleo 3
Núcleo 4
Núcleo 6
Núcleo 7
Núcleo 8
Núcleo 10
Núcleo 11
Núcleo 12
Núcleo 14
Núcleo 15
Núcleo 16
Núcleo 18
Núcleo 19
Núcleo 20
Núcleo 22
Núcleo 23
Núcleo 24
Núcleo 26
Núcleo 28
Núcleo 30
Núcleo 31
Núcleo 32
Núcleo 34
Núcleo 35
Núcleo 36
Núcleo 38
Núcleo 39
Núcleo 40
Núcleo 42
Núcleo 43
Núcleo 44
Núcleo 46
Núcleo 47
Núcleo 48
Núcleo 50
Núcleo 51
Núcleo 52
Núcleo 54
Núcleo 55
Núcleo 56
Núcleo 58
Núcleo 59
Núcleo 60
Núcleo 62
Núcleo 63
Núcleo 64
Nivel de rama 1
Nivel de rama 2
Chip
6
R1
10

Fig. 15

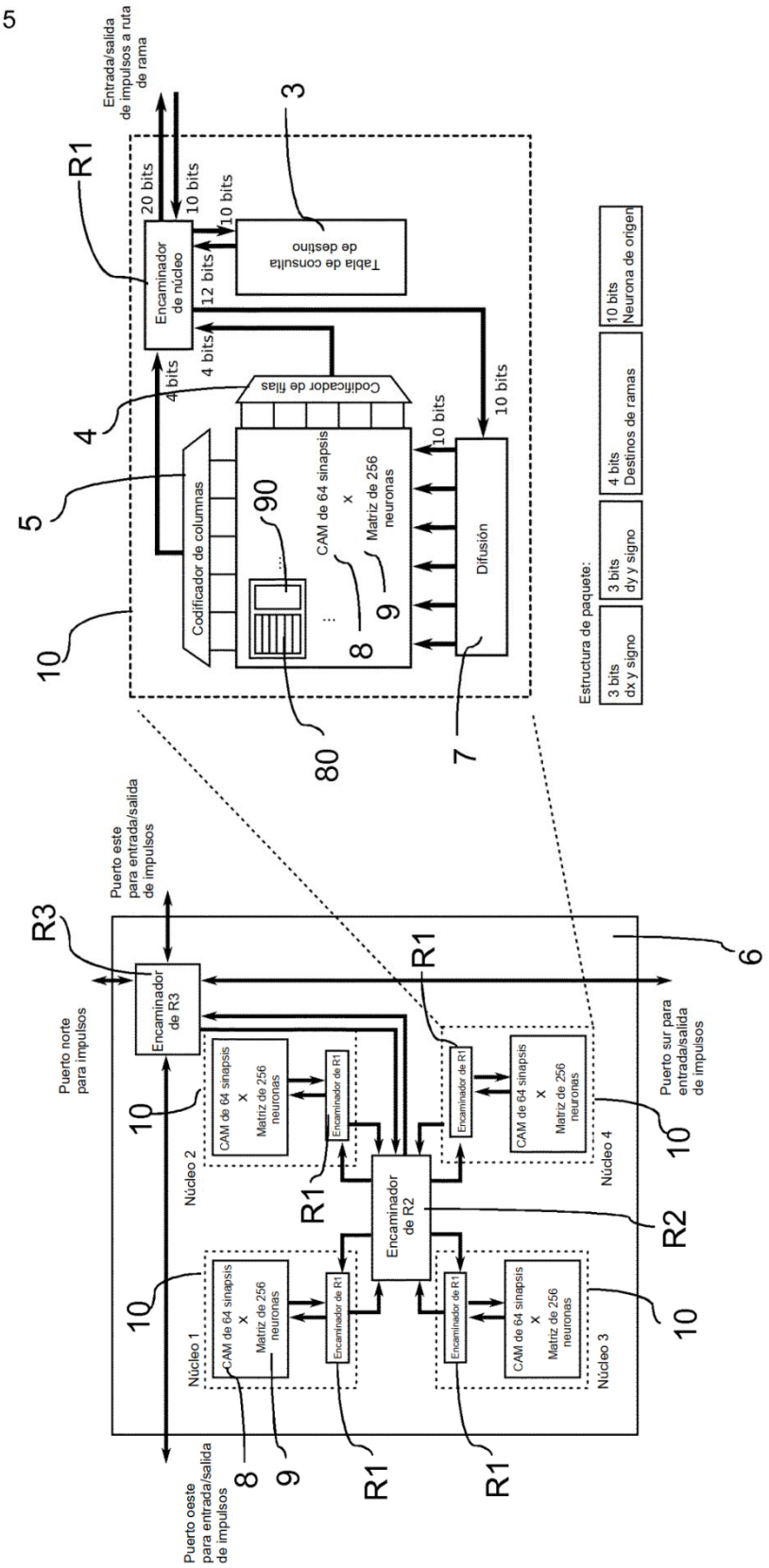
Entrada/salida de impulsos a ruta de rama
R1
3
20 bits
10 bits
10 bits
Encaminador de núcleo
12 bits
Tabla de consulta de destino
4 bits
4 bits
Codificador de filas
4
5
Codificador de columnas
90
CAM de 64 sinapsis x Matriz de 256 neuronas
10 bits
10 bits
Difusión
10
8
9
80
7
Estructura de paquete:
3 bits dx y signo
3 bits dy y signo
4 bits Destinos de ramas
10 bits Neurona de origen
Puerto este para entrada/salida de impulsos
R3
Encaminador de R3
R1
6
Puerto norte para impulsos
Puerto sur para entrada/salida de impulsos
10
Núcleo 2
CAM de 64 sinapsis x Matriz de 256 neuronas
Encaminador de R1
Núcleo 4
R1
Encaminador de R2
R2
Núcleo 1
Núcleo 3
8
9
R1
R1
Puerto oeste para entrada/salida de impulsos

Fig. 16

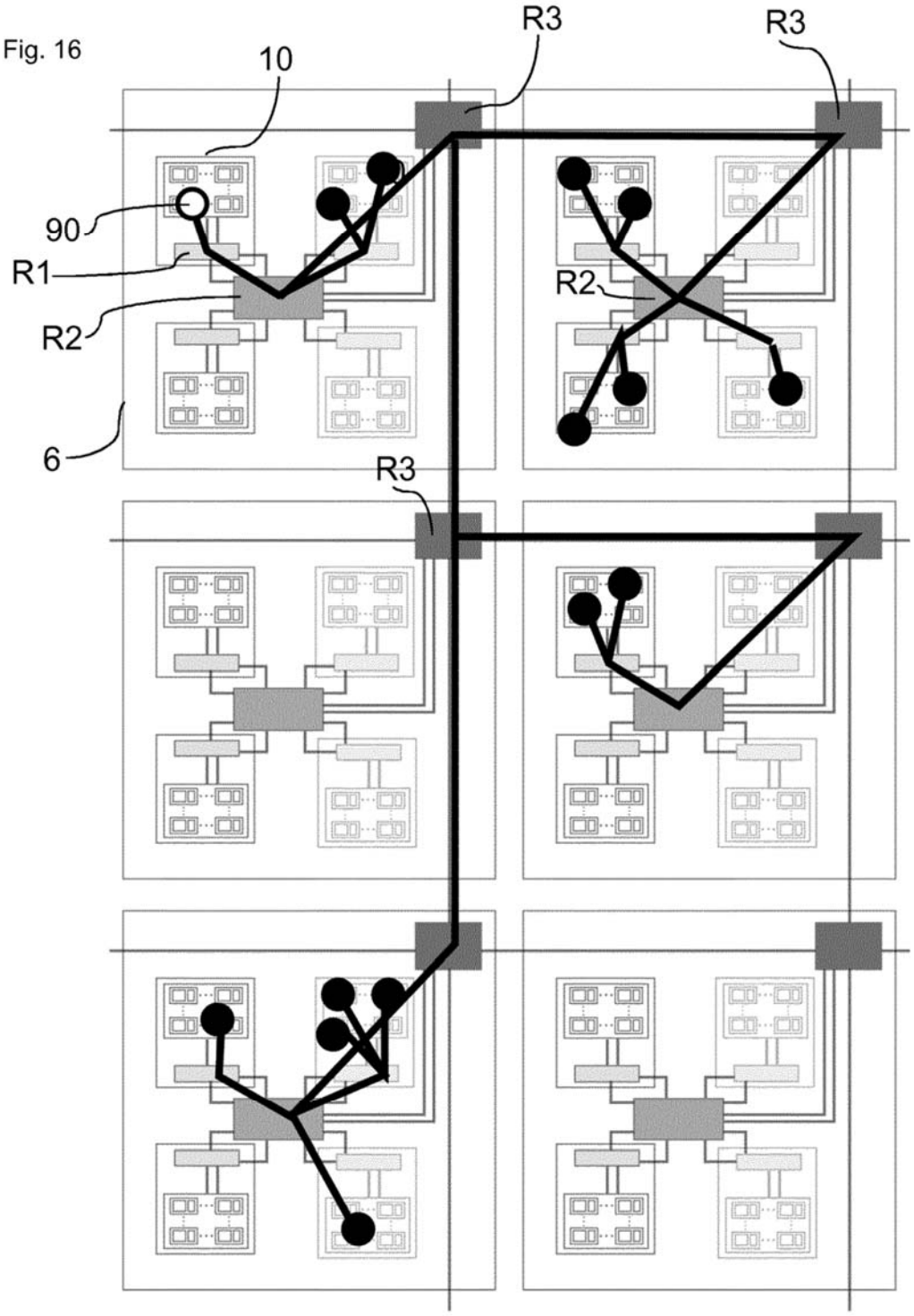

Fig. 17