

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 569 129**

21 Número de solicitud: 201400907

51 Int. Cl.:

G06F 7/02

(2006.01)

12

PATENTE DE INVENCION CON EXAMEN PREVIO

B2

22 Fecha de presentación:

05.11.2014

43 Fecha de publicación de la solicitud:

06.05.2016

Fecha de la concesión:

19.08.2016

45 Fecha de publicación de la concesión:

26.08.2016

73 Titular/es:

**UNIVERSIDAD DE SEVILLA (100.0%)
Vicerrectorado de Transferencia Tecnológica.
Paseo de las Delicias s/n. Pabellón de Brasil
41013 Sevilla (Sevilla) ES**

72 Inventor/es:

**SENHADJI NAVARRO , Raouf y
GARCÍA VARGAS , Ignacio**

54 Título: **Reconocedor reconfigurable de patrones de bits basado en jerarquía de memoria**

57 Resumen:

Reconocedor reconfigurable de patrones de bits basado en jerarquía de memoria que comprende una pluralidad de circuitos digitales organizados en dos niveles de memoria: principal y secundaria. El reconocedor de la invención permite identificar un conjunto determinado de patrones de bits a partir de una secuencia de bits de entrada así como modificar dinámicamente los patrones a reconocer. El circuito que constituye la memoria principal implementa una máquina de estados genérica que puede reconocer cualquier secuencia de dos bits (subFSM). El circuito que constituye la memoria secundaria almacena todas las subFSM necesarias para reconocer un conjunto de patrones de bits dado. Estas subFSM son transferidas a memoria principal cuando son requeridas por la secuencia de bits de entrada al reconocedor.

ES 2 569 129 B2

DESCRIPCIÓN

Reconocedor reconfigurable de patrones de bits basado en jerarquía de memoria.

5 Objeto de la invención

La presente invención se refiere a un reconocedor reconfigurable de patrones de bits basado en jerarquía de memoria que comprende una pluralidad de circuitos digitales organizados en dos niveles de memoria. Los reconocedores de patrones de bits son circuitos digitales que permiten procesar y reconocer secuencias de bits. Su campo de aplicación es muy extenso, pues el procesamiento y reconocimiento de secuencias de bits es esencial en numerosas aplicaciones. Por ejemplo, en el ámbito de las redes de computadores, los sistemas de enrutamiento IP o los sistemas de detección de intrusos son dos ejemplos característicos.

15 Estado de la técnica

El uso de máquinas de estados finitos (FSM) para modelar reconocedores de patrones de bits esta ampliamente extendido. A diferencia de los reconocedores de patrones de bits existentes, el reconocedor objeto de la invención es una FSM cuyo diseño se enmarca dentro de un nuevo modelo de FSM basado en jerarquía de memoria. Dicho modelo, propuesto recientemente en [R. Senhadji-Navarro and I. García-Vargas. "Finite Virtual State Machines", *IEICE Transactions on Information and Systems*, 2012], establece una arquitectura general (a la que nos referiremos como arquitectura de referencia) constituida por una jerarquía de memoria de dos niveles: memoria principal y memoria secundaria. Mientras que la arquitectura de referencia establece un marco general para la implementación de FSM, la arquitectura del reconocedor de la presente invención está especialmente diseñada para reconocer patrones de bits, permitiendo implementar reconocedores de patrones bits de altas prestaciones que no son posibles con la arquitectura de referencia. De hecho, la solución mostrada en el ejemplo de realización de la invención no puede ser implementada en la arquitectura de referencia.

Las diferencias principales entre la invención propuesta y la arquitectura de referencia son:

- 35 • Mientras que el componente de memoria principal de la arquitectura de referencia es una RAM (del inglés, "Random Access Memory") convencional, el componente de memoria principal del reconocedor de la invención comprende un conjunto de registros y multiplexores que implementa una RAM con características especiales. El componente de memoria principal del reconocedor de la invención permite actualizar en un único ciclo de reloj todo su contenido mientras que el componente de memoria principal de la arquitectura de referencia requiere tantos ciclos de reloj como palabras almacene.
- 40 • El número de palabras que almacena el componente de memoria principal del reconocedor de la invención es independiente del número de patrones a reconocer y sus longitudes. En la arquitectura de referencia este valor depende de las características de la FSM que se implemente.
- 45 • El componente de memoria secundaria del reconocedor de la invención comprende 4 bloques de memoria independientes que permiten acceso simultáneo. La

operación de escritura en el componente de memoria principal se realiza utilizando un multiplexor 4:1. En la arquitectura de referencia sólo existe un único bloque de memoria secundaria. El reconocedor de la invención permite, a diferencia de la arquitectura de referencia, leer 4 datos distintos del componente de memoria secundaria y seleccionar a continuación cuál debe ser escrito en el componente de memoria principal.

- En el reconocedor de la invención, la frecuencia de la señal de reloj que controla el componente de memoria secundaria es la mitad de la frecuencia de la señal de reloj que controla el componente de memoria principal. En el caso de la arquitectura de referencia, existe una única señal de reloj que controla ambos componentes de memoria. Esta característica del reconocedor de la invención permite mejorar el rendimiento respecto a los reconocedores convencionales.

Las FSM que modelan reconocedores de patrones de bits se caracterizan por tener un elevado número de estados, pudiendo alcanzar hasta millones de estados en determinadas aplicaciones. La implementación electrónica de estas FSM puede llevarse a cabo mediante una memoria que almacena las transiciones de la FSM (es decir, el identificador del patrón reconocido en ese momento y el siguiente estado) y es accedida por una dirección compuesta por el estado actual y la señal por la que se introduce la secuencia de bits. Las arquitecturas basadas en memoria ofrecen como ventaja la posibilidad de modificar en tiempo de ejecución los patrones que reconoce el sistema. simplemente realizando escrituras sobre la memoria que contiene las transiciones. En estos casos. se habla de reconocedores reconfigurables.

El problema que presentan las arquitecturas basadas en memoria convencionales es que el rendimiento se reduce de manera importante conforme aumenta el número de estados de la FSM. Para mejorar las prestaciones. el reconocedor de la invención está basado en una jerarquía de memoria de dos niveles. El circuito que constituye la memoria principal implementa una máquina de estados genérica que puede reconocer cualquier secuencia de dos bits (subFSM). El circuito que constituye la memoria secundaria almacena todas las subFSM necesarias para reconocer un conjunto de patrones de bits dado. Estas subFSM son transferidas a memoria principal cuando son requeridas por la secuencia de bits de entrada al reconocedor. Como la memoria que impone mayor retraso (es decir. la secundaria) funciona a la mitad de la frecuencia con la que opera el reconocedor, el rendimiento mejora con respecto a los reconocedores convencionales.

Descripción de las figuras

Figura 1 muestra el diagrama de bloques del reconocedor de la invención.

Figura 2 muestra la arquitectura del componente de memoria principal de la invención.

Figura 3 muestra la arquitectura del componente de memoria secundaria de la invención.

Figura 4 muestra el Diagrama de Transición de Estados de la FSM del ejemplo de realización de la invención.

Figura 5 muestra con detalle la parte superior izquierda del Diagrama de Transición de Estados de la FSM del ejemplo de realización de la invención.

Figura 6 muestra con detalle la parte superior derecha del Diagrama de Transición de Estados de la FSM del ejemplo de realización de la invención.

5 Figura 7 muestra con detalle la parte inferior izquierda del Diagrama de Transición de Estados de la FSM del ejemplo de realización de la invención.

Figura 8 muestra con detalle la parte inferior central del Diagrama de Transición de Estados de la FSM del ejemplo de realización de la invención.

10 Figura 9 muestra con detalle la parte inferior derecha del Diagrama de Transición de Estados de la FSM del ejemplo de realización de la invención.

15 Figura 10 muestra la asignación de las subFSM a los bloques RAM previa al proceso de fusión correspondiente al ejemplo de realización de la invención.

Figura 11 muestra la asignación de las subFSM a los bloques RAM después del proceso de fusión correspondiente al ejemplo de realización de la invención.

20 Figura 12 muestra el contenido de los bloques RAM para el ejemplo de realización de la invención (subFSM1, subFSM2 y subFSM3).

Figura 13 muestra el contenido de los bloques RAM para el ejemplo de realización de la invención (subFSM4, subFSM5 y subFSM6).

25 Figura 14 muestra el valor de las señales IMM_i para el ejemplo de realización de la invención.

Descripción de la invención

30 Un reconocedor de patrones de bits es un circuito secuencial que permite identificar un conjunto determinado de patrones de bits a partir de una secuencia de bits de entrada. Para cada patrón de bits reconocido, el circuito genera un patrón de bits de salida que identifica a dicho patrón. En el caso de que ningún patrón sea reconocido, el circuito
35 genera un identificador especial al que llamaremos *identificador de patrón no reconocido*.

Un reconocedor de patrones de bits puede modelarse de manera sencilla mediante una máquina de estados finitos (FSM). En este caso, el Diagrama de Transición de Estados (DTE) de la FSM tiene forma de árbol binario. Cada estado del nivel k procesa el k -ésimo bit y representa una combinación de valores distinta de los $(k-1)$ -ésimos bits procesados previamente. Cada estado tiene dos transiciones: una asociada al valor 0 del bit procesado y otra asociada al valor 1. Si en un determinado estado, el patrón procesado hasta el momento es uno de los patrones buscados, la salida de la FSM contiene el
40 identificador de dicho patrón; en caso contrario, la salida contiene el identificador de patrón no reconocido.

Una FSM reconfigurable que reconozca patrones de bits puede ser implementada mediante una memoria RAM (del inglés, "Random Access Memory"). La dirección que ataca a la memoria está compuesta por los bits de codificación de estados y la señal de
50 entrada de la FSM. El contenido de la memoria esta formado por todas las transiciones de la FSM. Cada transición está compuesta por los bits de codificación del estado

siguiente y los bits de salida (correspondientes al identificador de patrón). La reconfiguración de la FSM se lleva a cabo mediante operaciones de escritura en la memoria RAM. Cuando el número de patrones es alto y la longitud de los mismos es grande, la FSM correspondiente puede llegar a tener un número muy elevado de estados. El número de estados afecta de manera importante a las prestaciones obtenidas con una implementación mediante RAM.

El reconocedor de patrones de bits de la invención está basado en una jerarquía de memoria de dos niveles: la memoria principal y la memoria secundaria. La memoria principal implementa una FSM genérica de 6 estados, lo que permite reconocer cualquier patrón de 2 bits. Cada uno de estos 6 estados genéricos se denomina marco. La FSM original se descompone en distintas subFSM que incluyen a lo sumo 6 estados, lo que permite que la memoria principal pueda contener cualquiera de ellas. Nos referiremos a la subFSM almacenada en memoria principal en un momento dado como *subFSM activa*.

La memoria secundaria contiene todas las subFSM, las cuales son transferidas dinámicamente a la memoria principal según sean requeridas por la evolución de la FSM. El contenido completo de la memoria principal puede ser modificado en una única operación de escritura. Esto permite transferir una subFSM completa de la memoria secundaria a la principal en un sólo ciclo de reloj. El control de la transferencia es llevado a cabo por la subFSM activa. Denominamos subFSM siguiente a la subFSM que será almacenada en la siguiente operación de escritura.

La memoria secundaria está constituida por 4 bloques de memoria independientes a los que se accede con la misma dirección de lectura. Cada palabra de estos bloques almacena una subFSM distinta, lo que permite la lectura simultánea de 4 subFSM que tengan asignadas la misma posición de memoria. En el momento de comenzar el proceso de carga de la subFSM siguiente, se inicia la lectura de las 4 subFSM candidatas para ser escritas en memoria principal. La evolución de la FSM determinará finalmente qué subFSM será transferida a memoria principal, convirtiéndose en la subFSM activa. Denominamos grupo al conjunto de estas subFSM candidatas. Cada una de las subFSM de un grupo se almacena en un bloque distinto pero en la misma posición de memoria de cada bloque.

Supongamos una FSM para reconocer patrones que tiene identificadores de M bits y que puede descomponerse en W grupos. Se requieren, por tanto, $R = \lceil \log_2 W \rceil$ bits para identificar a cada uno de estos grupos. Cada transición de una subFSM está constituida por: R bits para seleccionar el grupo de subFSM a leer de la memoria secundaria (*próximo-grupo*); 3 bits para seleccionar el siguiente marco de la memoria principal (*siguiente-marco*), M bits para el identificador del patrón (*identificador*); 2 bits para seleccionar la subFSM a cargar en la memoria principal de entre las 4 subFSM previamente leídas de la memoria secundaria (*próxima-subFSM*); y 1 bit para controlar la escritura de la memoria principal (*he*). Por tanto, cada transición requiere $T = R + M + 6$ bits.

Las Figuras 1, 2 y 3 muestran la arquitectura del reconocedor de patrones de la invención. La Figura 1 muestra el diagrama de bloques de más alto nivel, el cual incluye el componente de memoria principal (100) y el componente de memoria secundaria (200). La memoria principal (100), véase la Figura 2, está constituida por los 12 registros de T bits (301, ..., 312), los 12 multiplexores 2:1 de T bits (501, ..., 512), el multiplexor 12:1 de T bits (120) y el conjunto de registros (131), (133), (134) y (135) que almacenan la transición actual. El registro (131) es de 3 bits y almacena siguiente-marco; (133) es de

R bits y almacena próximo-grupo; (134) es de 2 bits y almacena *próxima-subFSM*; y (135) es de 1 bit y almacena *he*.

Cada uno de los registros (301, ..., 312) almacena una de las transiciones de la subFSM activa. La carga de estos registros está gobernada por la señal de reloj CLK (143) y controlada por la salida del registro (135). La señal de entrada de los registros (301, ..., 312) está conectada a la salida de los multiplexores (501, ..., 512), lo que permite inicializar los registros con los valores de las señales IMM_i (901, ..., 912) en el caso de que la señal de RESET (148) esté activa o cargar los registros con un valor procedente de la memoria secundaria. Las señales IMM_i (901, ..., 912) contienen las transiciones de la subFSM inicial, es decir, la que procesa los primeros bits cuando el reconocedor se *resetea*.

Los registros (131), (134) y (135) están gobernados por la señal de reloj CLK (143). El registro (133) está gobernado por la señal de reloj CLK2 (144) cuyo periodo es el doble que el de la señal CLK (143). Cuando el RESET (148) se activa, la salida del registro (135) se pone a 1 para permitir la carga de los registros (301, ..., 312) con los valores (901, ..., 912).

El multiplexor 120 permite seleccionar la transición actual. Los campos próximo-marco, *próximo-grupo*, *próxima-subFSM* y *he* de esta transición se almacenan en los registros (131), (133), (134) y (135), respectivamente. El campo *identificador* de la transición actual está conectado a la serial de salida del reconocedor OUTPUT (142).

El control del multiplexor (120) se lleva a cabo mediante una palabra formada por la salida del registro (131) y la señal de INPUT (141). De esta forma, se determina el siguiente marco y el identificador actual a partir del marco actual y el bit de la secuencia de entrada que está siendo procesado.

La memoria secundaria (200), véase la Figura 3, está constituida por los 4 bloques de RAM síncronos (601, ..., 604) y por el multiplexor 4:1 de V bits (210). Estos bloques RAM son de doble puerto: un puerto A de lectura con direcciones de R bits y datos de $V = 12 \cdot T$ bits, y un puerto B de escritura con direcciones de P bits y datos de Q bits. Tanto las operaciones de lectura como de escritura de los bloques RAM son síncronas, estando gobernadas por la señal de reloj CLK2 (144). Si la señal de RESET (148) no está activa, la salida del puerto A contiene la subFSM indicada por la señal BADDR (180), la cual contiene *próximo-grupo*. Cuando el RESET (148) está activo, la salida del puerto A toma el valor indicado por las señales externas ISM_i (701, ..., 704). Esto permite que la memoria principal se cargue con la subFSM necesaria cuando se *resetea* el reconocedor.

Las salidas de los bloques RAM (601, ..., 604) están conectadas al multiplexor (210), el cual permite seleccionar la subFSM que se cargará en memoria principal a partir de las 4 subFSM leídas. Los V bits de salida del multiplexor (210), véase la señal SFSM (182), se descompone en 12 señales de T bits que atacan a cada uno de los registros (301, ..., 312) a través de los multiplexores (501, ..., 512). El multiplexor (210) está controlado por la señal BSEL (181), la cual contiene *próxima-subFSM*.

El contenido de las subFSM puede ser reconfigurado o dinámicamente realizando operaciones de escritura en el puerto B de los bloques RAM (601, ..., 604). Las señales de habilitación de escritura de estos bloques están conectadas a las señales externas WE_i

(801, ..., 804); las señales de dirección, a las señales externas ADDR_i (811, ..., 814); y las señales de datos, a las señales externas DIN_i (821, ..., 824).

A continuación se describe el proceso para generar los grupos de subFSM a partir de la FSM original. Supongamos una FSM para reconocer patrones de bits genérica. Denotamos S₀ el nodo raíz (nivel 0). En cada nivel k = 1, 2, 3, 4, ... existen a lo sumo 2^k nodos a los que denotamos S_{2^{k-1}}, S_{2^k}, ..., S_{2^{k+1-3}} y S_{2^{k+1-2}} (en este apartado, mantendremos la numeración incluso cuando el DTE de la FSM no sea completo). La subFSM inicial (es decir, aquella que contiene la memoria principal cuando se resetea la FSM) está constituida por los estados S₀, S₁ y S₂. Por cada estado S_j del nivel k = 1, 3, 5, 7, ... se genera una subFSM con los estados S_{2j+1} (hijo izquierdo de S_j), S_{2j+2} (hijo derecho de S_j), S_{4j+3} (hijo izquierdo de S_{2j+1}), S_{4j+4} (hijo derecho de S_{2j+1}), S_{4j+5} (hijo izquierdo de S_{2j+2} y S_{4j+6} (hijo derecho de S_{2j+2}), es decir todos los hijos y nietos del estado de S_j. Denotaremos esta subFSM como F_j. Las cuatro subFSM (si existen) que se corresponden con los nietos de un mismo estado S_j del nivel k = 1, 3, 5, 7, ... forman parte siempre de un mismo grupo.

Las transiciones de las subFSM resultantes incluyen la siguiente información: *próximo-grupo*, *siguiente-marco*, *identificador*, *he* y *próxima-subFSM*. Cada transición de las subFSM contiene el mismo identificador de patrón que la transición equivalente de la FSM original. El campo *próximo-marco* de una transición contiene la dirección del marco donde se almacena el estado de la FSM original correspondiente al estado siguiente de la transición. El campo *próximo-grupo* de la transición que lleva a la FSM al estado S_j del nivel k = 1, 3, 5, 7, ... contiene la dirección en memoria secundaria del grupo que contiene las 4 subFSM correspondientes a los nietos de S_j (denotadas por F_{4j+3}, F_{4j+4}, F_{4j+5} y F_{4j+6}). Las transiciones que van desde los estados hijos a los estados nietos de S_j contienen como campo *próxima-subFSM* el valor i = 0, 1, 2 ó 3 que representa el bloque de memoria en la que está almacenada F_{4j+3+i}. Estas transiciones contienen como *he* el valor 1. Los valores de *próximo-grupo* y *próxima-subFSM* no especificados anteriormente contienen indeterminaciones. Los valores no especificados de *he* contienen 0.

El proceso mediante el cual se modifica la subFSM activa puede resumirse de la siguiente forma. La subFSM activa establece el valor de *próximo-grupo* en un determinado ciclo de la señal de reloj CLK (143). comenzando la lectura del grupo correspondiente en la memoria secundaria. Debido a que la memoria secundaria está gobernada por la señal de reloj CLK2 (144), cuya frecuencia es la mitad que la de CLK (143), las subFSM se leen de los bloques RAM (601, ..., 604) dos ciclos más tarde. Entonces, la subFSM indicada por *próxima-subFSM* es seleccionada para ser cargada en la memoria principal y *he* se activa. En el siguiente ciclo de la señal de reloj CLK (143). la carga de la subFSM se hace efectiva. permitiendo que en ese mismo ciclo la transición actual sea leída.

Gracias a los registros de salida de ambas memorias. las operaciones para cargar distintas subFSM pueden solaparse en el tiempo. Las señales *próxima-subFSM* y *he* que determinan la carga de F_j en memoria principal se activan en el mismo ciclo de reloj que la señal *próximo-grupo* que selecciona el grupo {F_{4j+3}, F_{4j+4}, F_{4j+5}, F_{4j+6}} (todas estas señales se activan en la transición al estado S_j). La señal *próximo-grupo* es activada por una subFSM distinta a la que activa la señal *próxima-subFSM* en la carga de una subFSM determinada.

La característica fundamental del reconocedor de patrones de la invención es que los estados de la FSM se encuentran almacenados en una memoria que funciona a la mitad de frecuencia que la frecuencia de operación de la máquina de estados. Esto permite mejorar el rendimiento respecto a los reconocedores convencionales cuando el número de patrones y sus longitudes son elevados.

Modo de realización de la invención

Como ejemplo de realización de la invención, se muestra el diseño de un reconocedor para los patrones bits siguientes: 01001, 00001, 100, 0000011, 0110, 10111, 11101, 111001, 1011 y 1011101. El identificador de patrón representará la posición que ocupa éste en la lista anterior, siendo el *identificador de patrón no reconocido* el 0000. La Figura 4 muestra el DTE de la FSM generada a partir de este conjunto de patrones (los detalles se muestran en la Figuras 5, 6, 7, 8 y 9). La descomposición en subFSM se ha realizado siguiendo el procedimiento descrito. Por claridad, el estado final (830) ha sido replicado cuando ha sido necesario. El estado final es el único estado al que le llega mas de una transición (para indicar esta excepcionalidad, se ha sombreado el nodo correspondiente). Para cada subFSM, se muestra su identificador así como el grupo al que pertenece.

La subFSM inicial está compuesta por los estados {S1, S2, S11} (véase la Figura 5). Esta subFSM se almacenará en memoria principal cuando se active la serial de RESET. El resto de subFSM se almacenaran en los bloques de RAM de la memoria secundaria, tal y como muestra la Figura 10. Cada subFSM está asignada a uno de los bloques de memoria de forma que se garantice que todas las subFSM que pertenezcan al mismo grupo ocupen la misma posición de la memoria secundaria. Por ejemplo, F4, F17 y F8 ocupan la misma posición en memoria secundaria, ya que todas pertenecen al grupo 1 (véase la Figura 10 y las Figuras 8 y 9). En el ejemplo, el identificador de grupo se ha utilizado como dirección en la que se almacena el grupo.

Debido a que, habitualmente, existe un número elevado de subFSM con menos de 6 estados, es posible fusionar varias subFSM en una misma, siempre que se cumpla que todas las subFSM que ocupen una misma posición de memoria en la asignación original estén incluidas en subFSM que ocupen la misma posición en la asignación final. Tras un procedimiento de fusión, las subFSMs se almacenan en la memoria secundaria tal y como muestra la Figura 11.

El reconocedor de patrones del ejemplo puede ser implementado en la arquitectura mostrada en las Figuras 1, 2 y 3 con los valores de para metros siguientes: $R = \lceil \log_2 2 \rceil = 1$, $M = \lceil \log_2 (10+1) \rceil = 4$, $T = R+M+6 = 11$ y $V = 12*T = 132$. Para cada estado de una subFSM se almacena el valor de las señales *siguiente-marco*, *identificador*, *próximo-grupo*, *próxima-subFSM* y he para el valor de entrada '0' y '1' (en ese orden). El orden utilizado para almacenar los distintos estados de una subFSM viene dado por la posición en que aparecen cuando se especifica la subFSM. Por ejemplo, la posición 1 de la RAM2 incluirá (en ese orden): la transición correspondiente a '0' de S5, la transición correspondiente a '1' de S5, la transición correspondiente a '0' de S6, la transición correspondiente a '1' de S6, la transición correspondiente a '0' de S30 y la transición correspondiente a '1' de S30. El contenido de las cuatro bloques RAM se muestran en la Figura 12 y 13. Los bloques RAM0, RAM1, RAM2 y RAM3 corresponden con los bloques (601), (602), (603) y (604), respectivamente. La Figura 12 muestra la parte alta de la palabra (correspondiente a la subFSM1, subFSM2 y subFSM3). La Figura 13 muestra la parte baja de la palabra (correspondiente a la subFSM4, subFSM5 y subFSM6).

- 5 Los valores de las señales IMM_i (901, ..., 912) se obtienen siguiendo la ordenación en que aparecen los estados en la subFSM inicial ($\{S1, S2, S11\}$). La Figura 14 muestra el valor de las señales $IMM1$ (901), $IMM2$ (902), ... y $IMM6$ (906). Nótese que el valor del resto de señales IMM_i es irrelevante, pues la subFSM inicial sólo contiene 3 estados. Las señales ISM_i (701, 704) contienen los valores almacenados en cada uno de los bloques RAM en la dirección 0.

REIVINDICACIONES

1. Reconocedor reconfigurable de patrones de bits, que identifica una secuencia de bits de entrada a través de la señal INPUT (141) y genera como salida en la serial OUTPUT (142) el identificador del patrón correspondiente, **caracterizado** porque esta basado en una jerarquía de memoria de dos niveles cuya memoria principal (100) permite implementar una subFSM genérica de 6 estados que reconoce 2 bits de la secuencia de entrada y cuya memoria secundaria (200) almacena todas las subFSM en la que se divide la máquina de estados finitos (FSM) que implementa el reconocedor. La memoria principal (100) está gobernada por una señal de reloj CLK (143) cuya frecuencia es el doble que la de la serial de reloj CLK2 (144) que gobierna la memoria secundaria (200). En cada ciclo de la señal de reloj CLK (143), la subFSM almacenada en la memoria principal (100), llamada subFSM activa, genera en OUTPUT (142) el identificador de patrón reconocido, establece el próximo estado de la subFSM activa y gestiona la transferencia desde la memoria secundaria (200) a la memoria principal (100) de la subFSM siguiente. En cada ciclo de la señal de reloj CLK2 (144), se realiza la lectura en paralelo de las 4 subFSM candidatas a ser escritas en la memoria principal (100) y en el siguiente ciclo de reloj se realiza la escritura correspondiente. El conjunto de patrones que identifica el reconocedor de la invención puede modificarse dinámicamente realizando operaciones de escritura en la memoria secundaria (200). La memoria secundaria (200) comprende:

- 4 bloques de memoria de acceso aleatorio (RAM) síncronos (601, ..., 604). gobernados por la señal de reloj CLK2 (144), que almacenan entre todos ellos el conjunto completo de subFSM, que disponen de puertos distintos para lectura y escritura, y que tienen como entrada las señales externas (701, ..., 704) cuyos valores definen las 4 primeras subFSM candidatas, respectivamente, las cuales aparecen en la salida del puerto de lectura cuando se activa la señal de RESET (148);
- y un multiplexor 4:1 (210), que selecciona una de las 4 subFSM candidatas, las cuales son leídas en paralelo de los bloques RAM (601, ..., 604).

La memoria principal (100) comprende:

- 12 registros (301, ..., 312). gobernados por la señal de reloj CLK (143), que almacenan las transiciones correspondientes a los 2 posibles valores de la señal de entrada INPUT (141) de cada uno de los 6 estados de la subFSM activa;
- un registro de 3 bits (131), gobernado por la señal de reloj CLK (143), que contiene el estado actual de la subFSM activa;
- un registro (133), gobernado por la señal de reloj CLK2 (144), que contiene la dirección que ataca a los 4 bloques RAM (601, ..., 604) de la memoria secundaria (200);
- un registro de 2 bits (134), gobernado por la señal de reloj CLK (143), cuya salida está conectada a las señales de control del multiplexor 4:1 (210), lo que permite seleccionar cuál de las 4 subFSM candidatas se almacena en los registros (301, ..., 312), es decir. cual se convierte en subFSM activa;
- un multiplexor 12:1 (120), controlado por la señal INPUT (141) y el registro (131), que selecciona la transición siguiente de entre las 12 transiciones de la subFSM activa

almacenadas en los registros (301, ..., 312), la cual se descompone en la señal OUTPUT (142) y en las señales que se conectan a los registros (131), (133), (134) y (135);

- 5 - un registro de 1 bit (135), gobernado por la señal de reloj CLK (143), cuya salida está conectada a la señal de habilitación de carga de los registros (301, ..., 312), lo que permite almacenar la subFSM activa en la memoria principal (100);
- 10 - y 12 multiplexores 2:1 (501, ..., 512), controlados por la señal RESET (148), que permiten seleccionar, como valor de entrada a los registros (301, ..., 312), la próxima subFSM activa o la subFSM inicial definida en la señales externas (901, ..., 912), respectivamente;

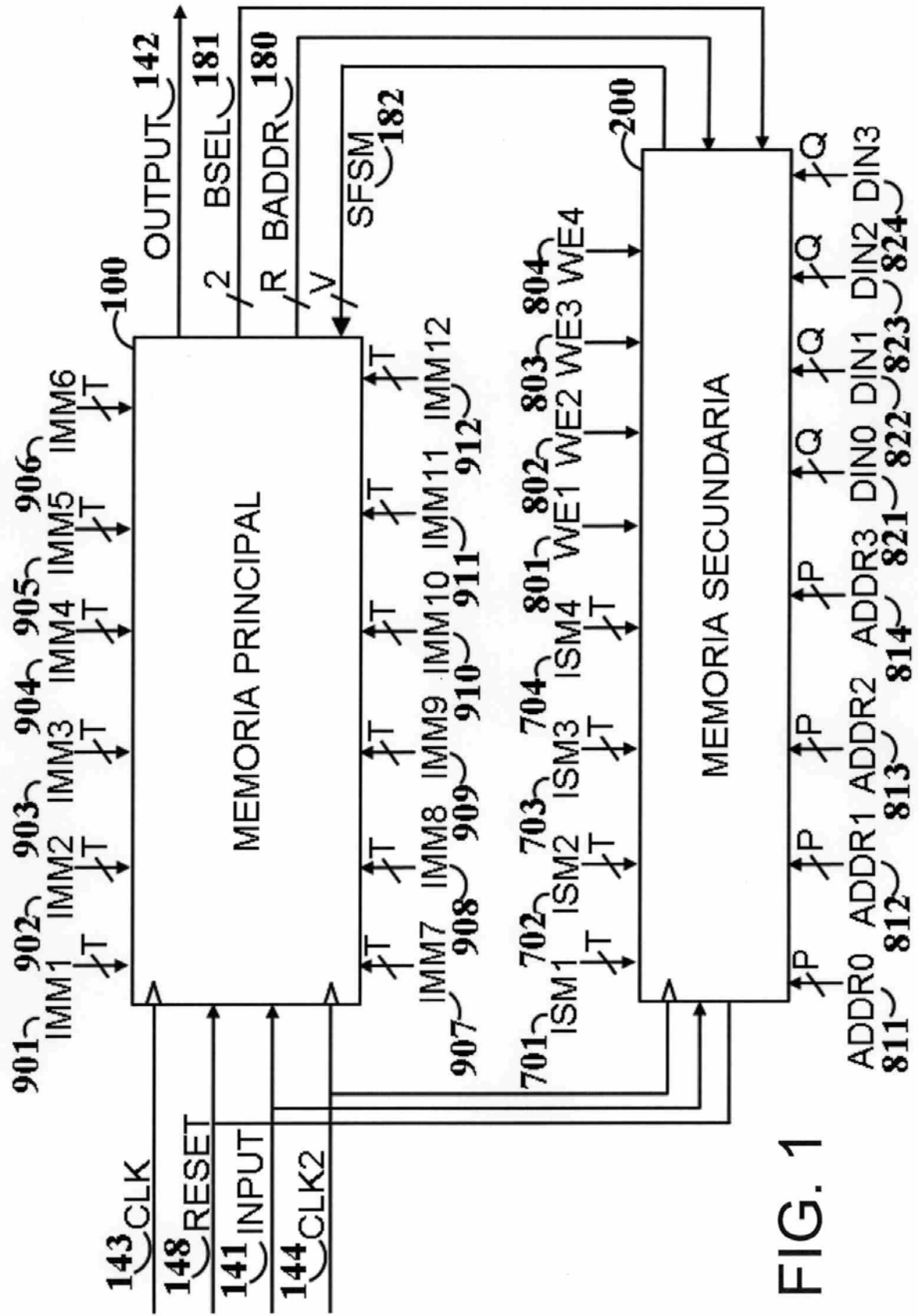


FIG. 1

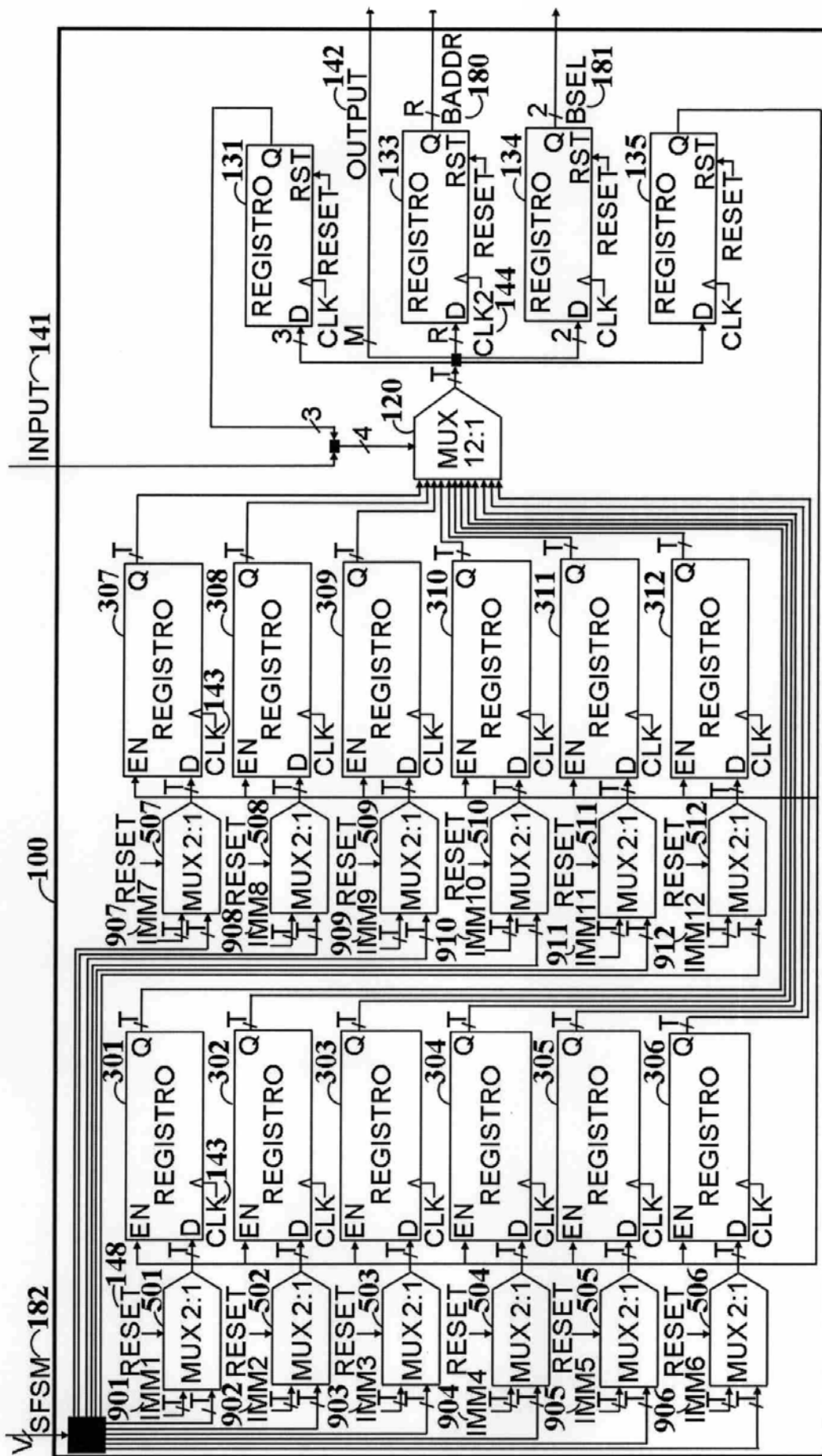


FIG. 2

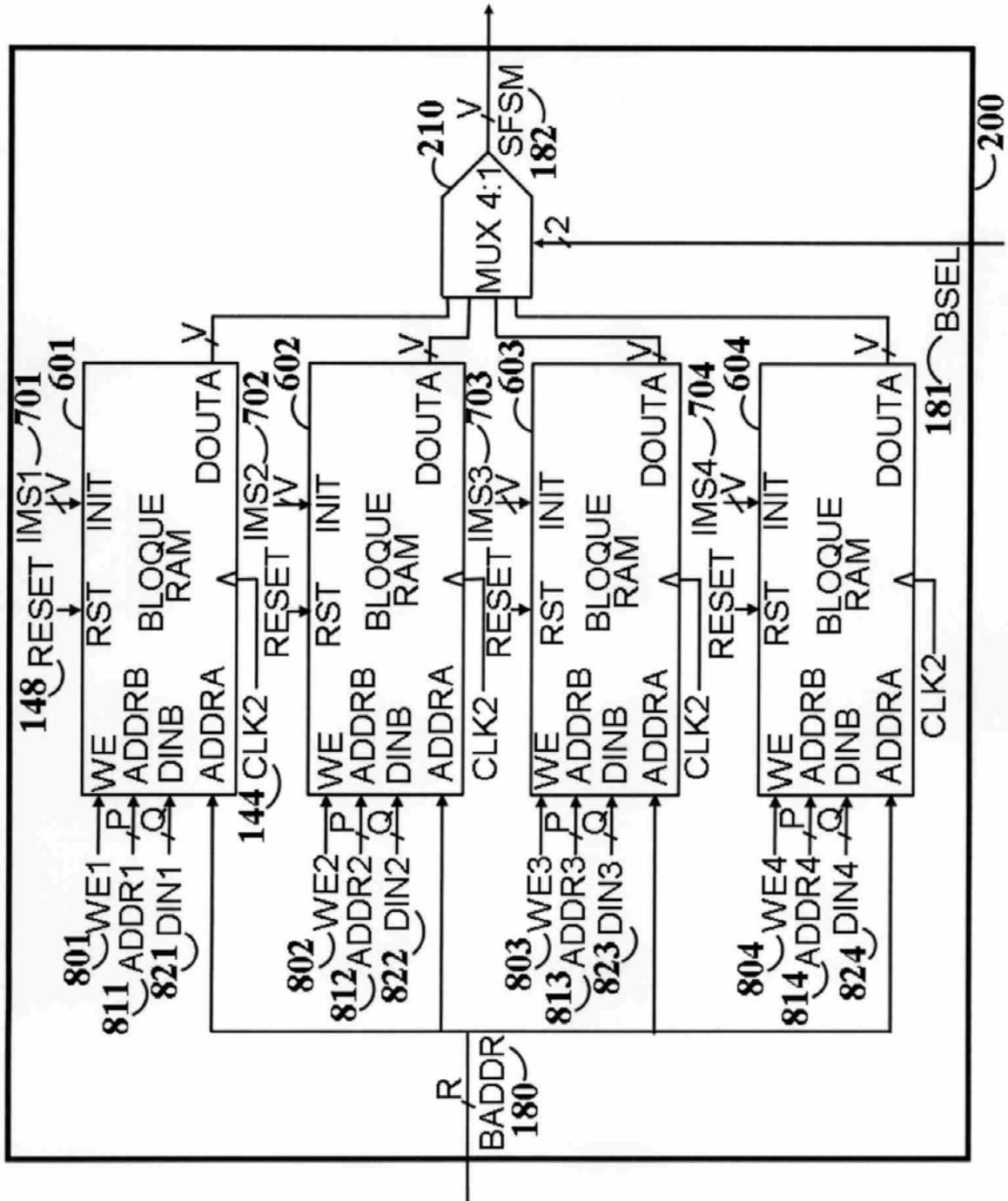


FIG. 3

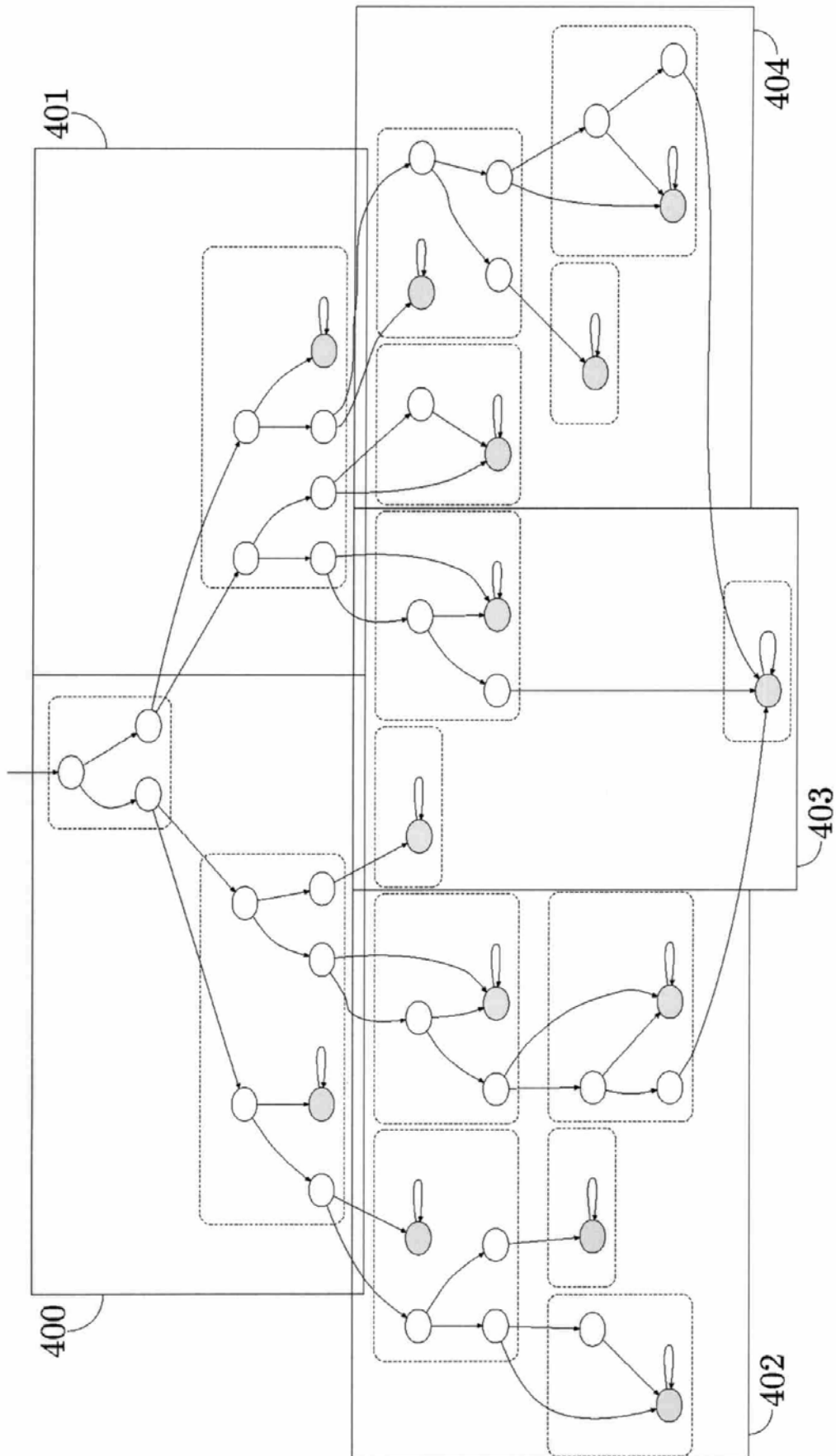


FIG. 4

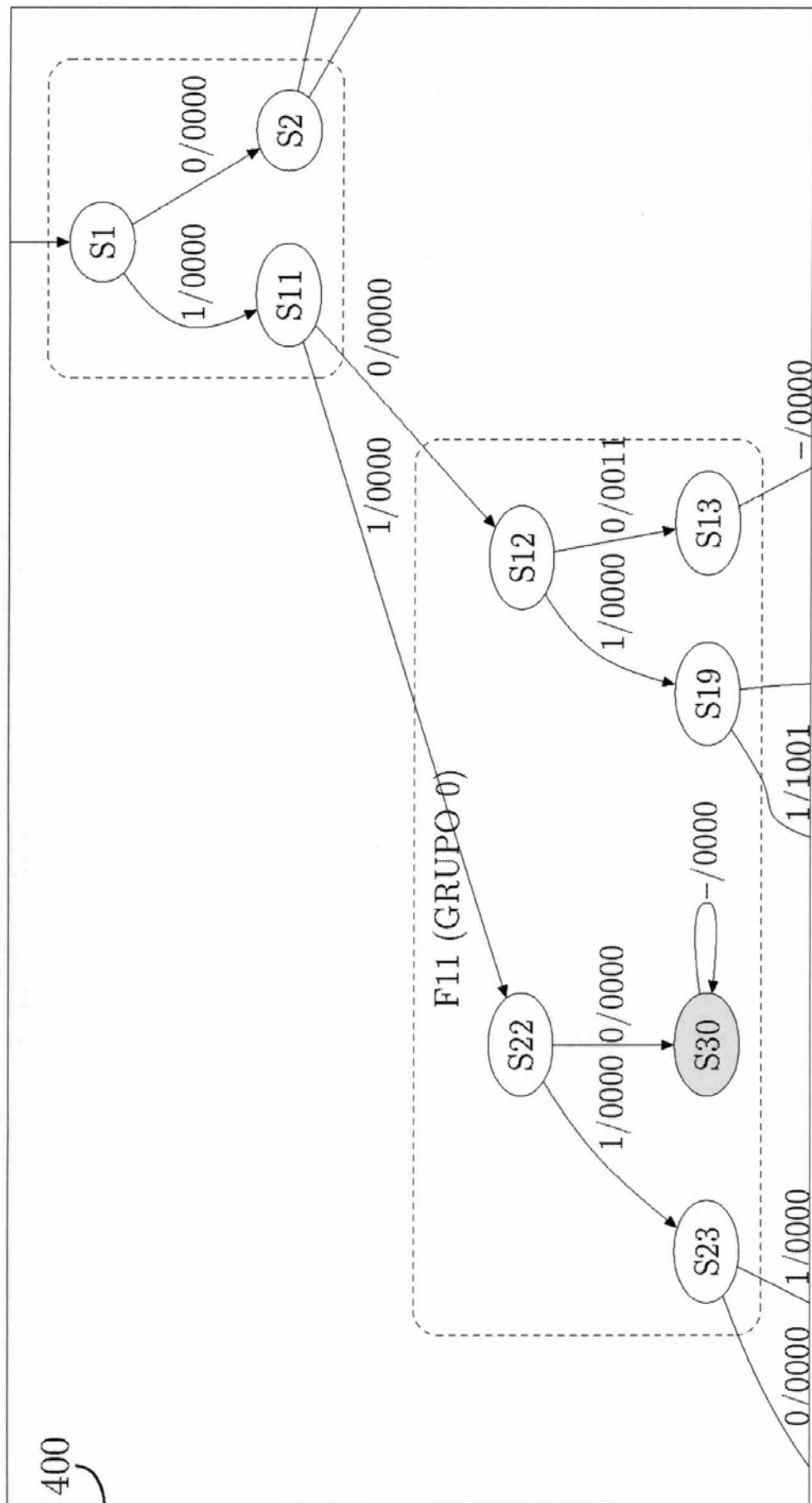


FIG. 5

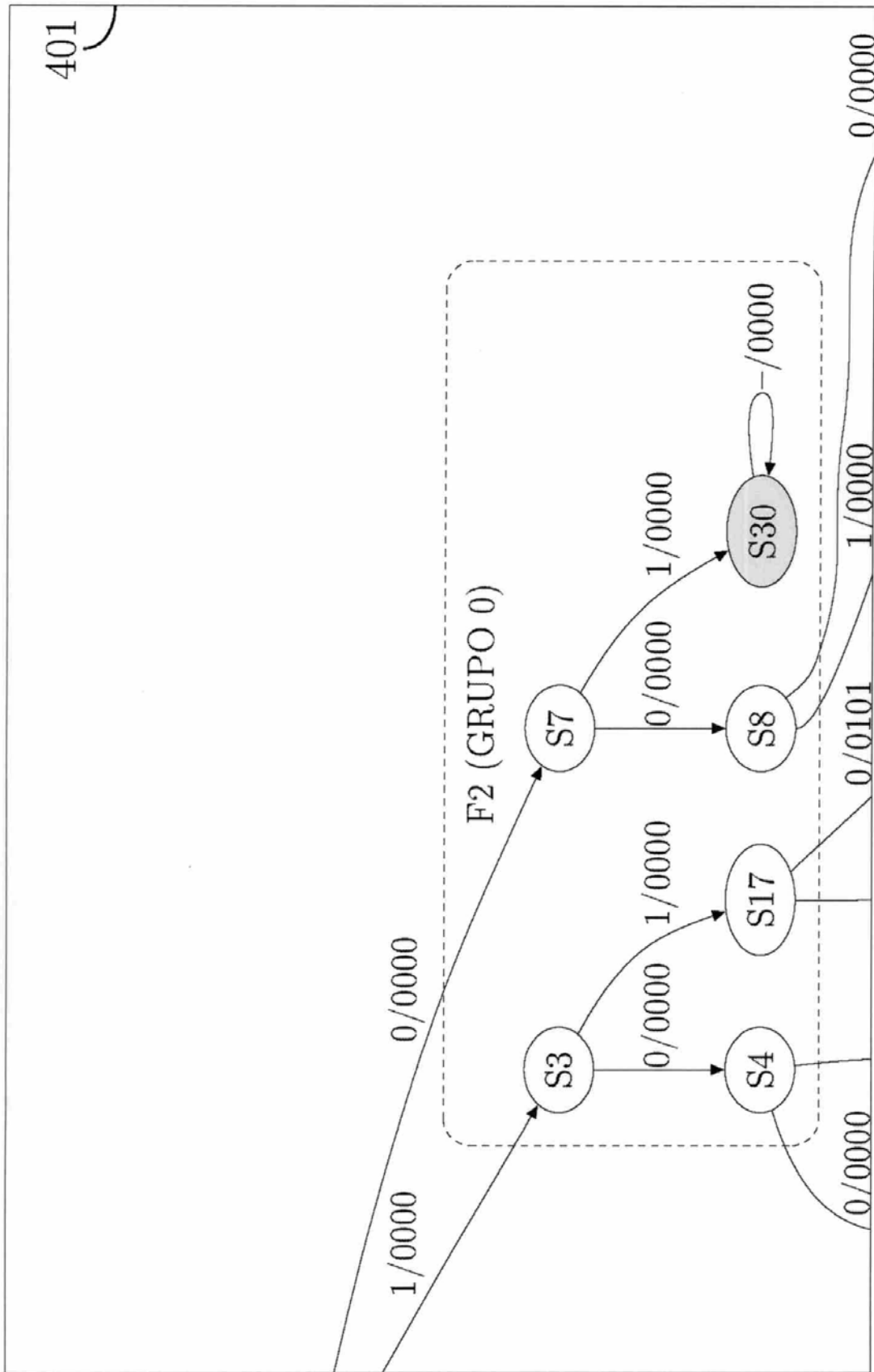


FIG. 6

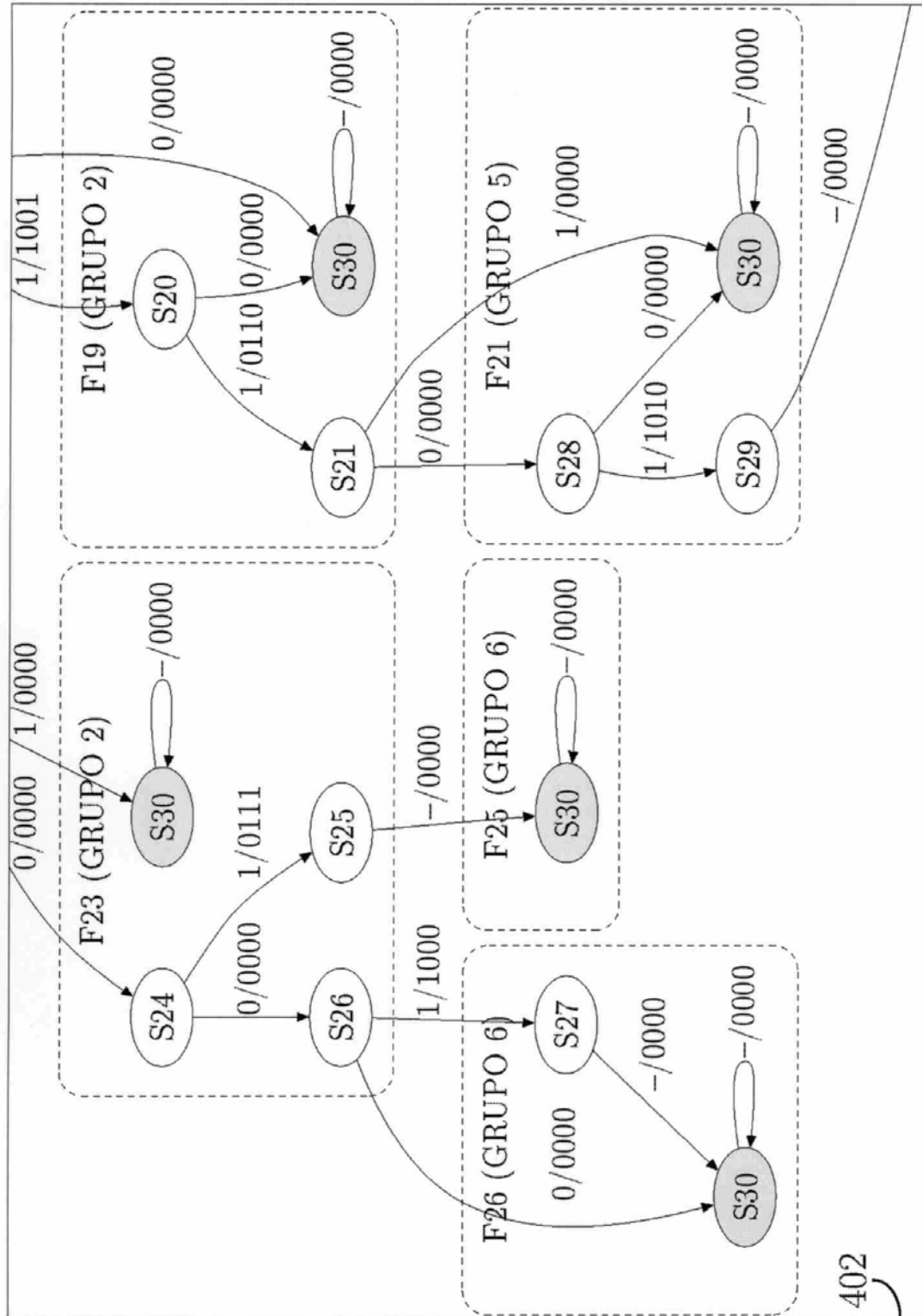


FIG. 7

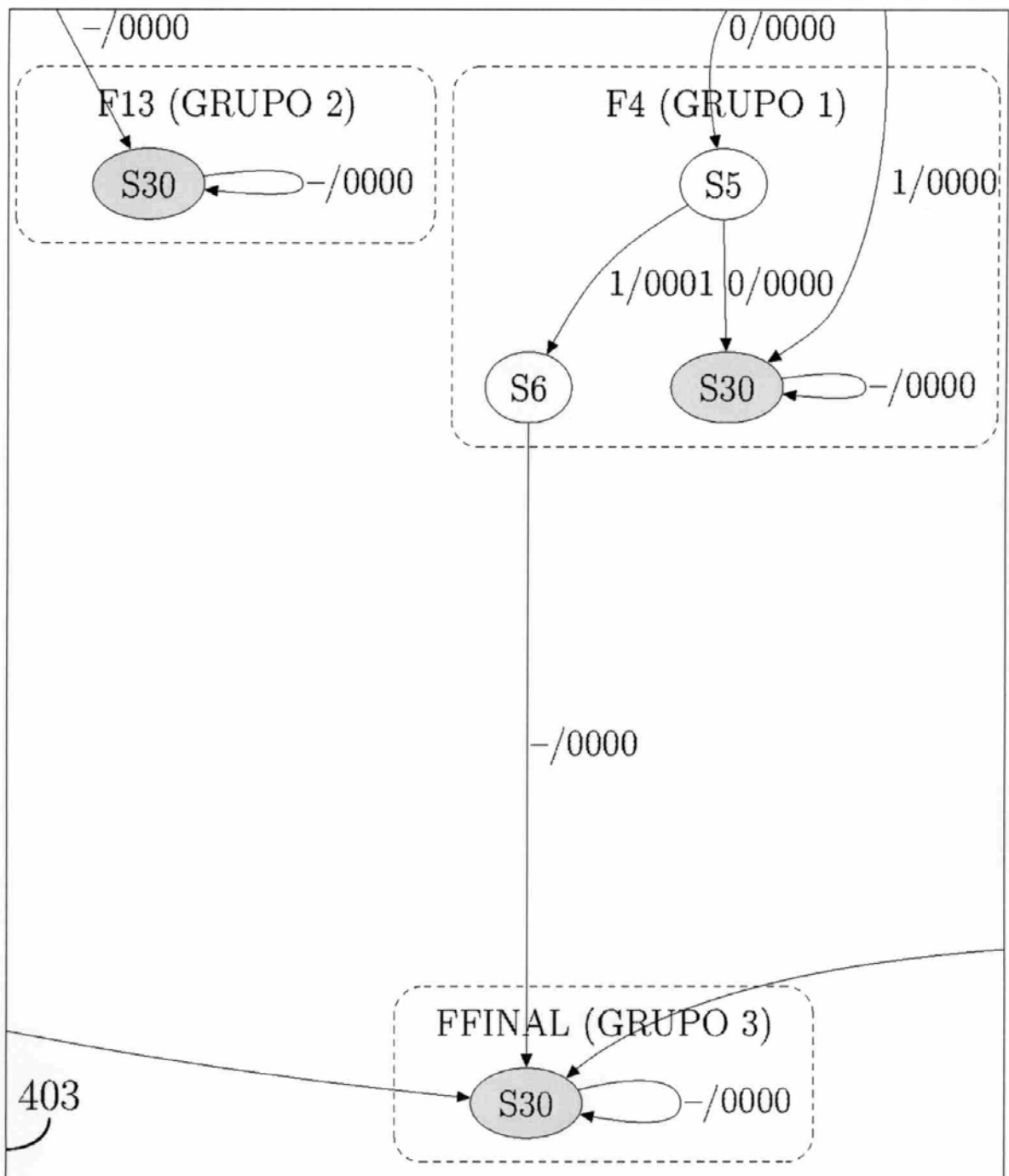


Fig. 8

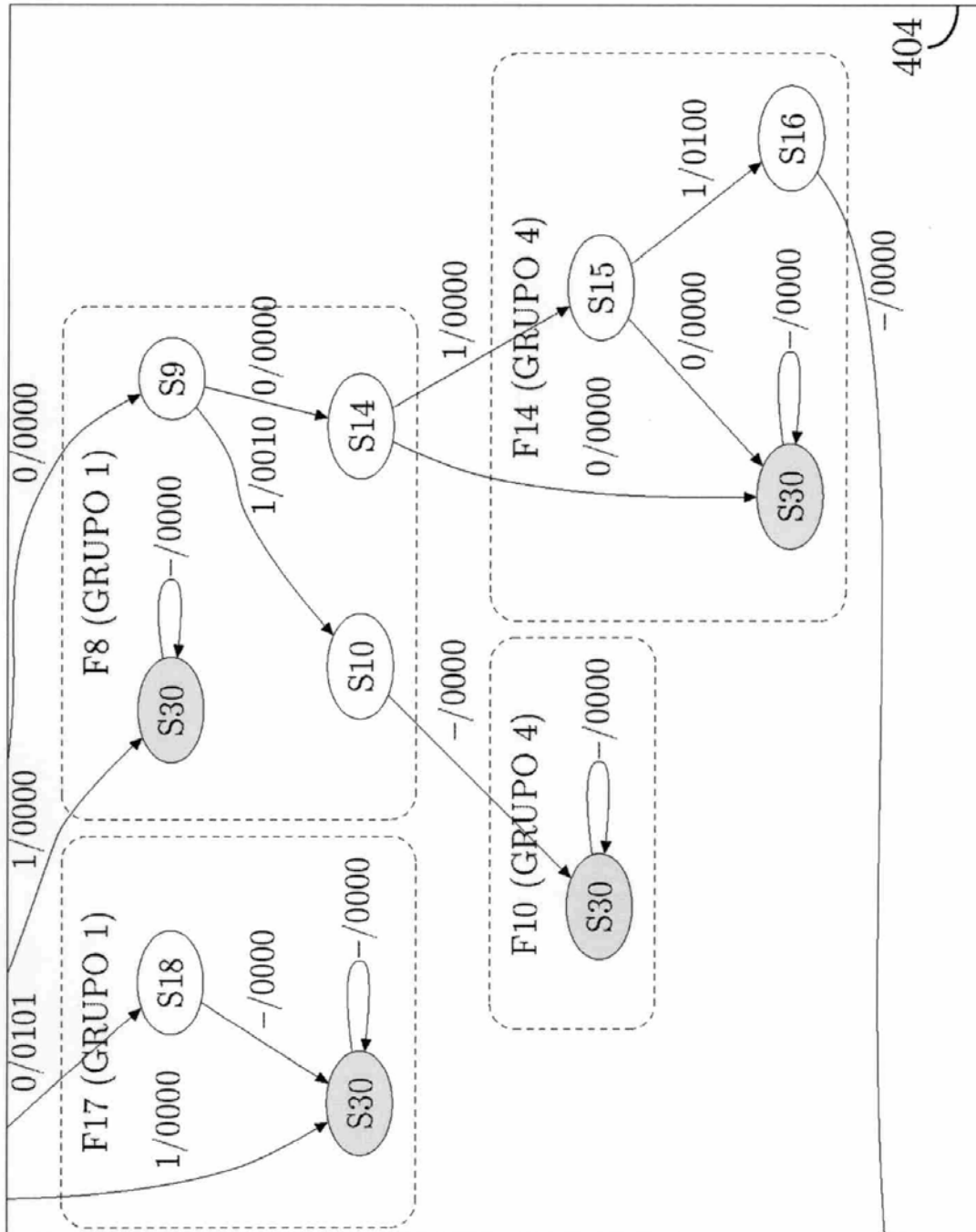


FIG. 9

DIRECCIÓN	CONTENIDO RAM0	DIRECCIÓN	CONTENIDO RAM1
0	{S3, S4, S7, S8, S17, S30}	0	{S12, S13, S19, S22, S23, S30}
1	{S9, S10, S14, S30 }	1	{ }
2	{S30}	2	{S20, S21, S30}
3	{ }	3	{S30}
4	{S15, S16, S30}	4	{S30}
5	{ }	5	{S28, S29, S30}
6	{S27, S30}	6	{S30}
7	{ }	7	{ }
8	{ }	8	{S30}

DIRECCIÓN	CONTENIDO RAM2	DIRECCIÓN	CONTENIDO RAM3
0	{ }	0	{ }
1	{S5, S6, S30}	1	{S18, S30}
2	{ }	2	{S24, S25, S26, S30}
3	{ }	3	{ }
4	{ }	4	{ }
5	{ }	5	{ }
6	{ }	6	{ }
7	{ }	7	{S30}
8	{ }	8	{ }

FIG. 10

DIRECCIÓN	CONTENIDO RAM0	DIRECCIÓN	CONTENIDO RAM1
0	{S3, S4, S7, S8, S17, S30}	0	{S12, S13, S19, S22, S23, S30}
1	{S9, S10, S14, S30}	1	{ }

DIRECCIÓN	CONTENIDO RAM2	DIRECCIÓN	CONTENIDO RAM3
0	{S15, S16, S20, S21, S27, S30}	0	{S24, S25, S26, S28, S29, S30}
1	{S5, S6, S30}	1	{S18, S30}

FIG. 11

	subFSM1		subFSM2		subFSM3	
	trans0	trans1	trans0	trans1	trans0	trans1
RAM0	00100000101	10000000111	00000000000	01000000000	01100000001	10100000000
	01000000101	00100100001	10100000000	10100000000	10100000000	00000000000
RAM1	00100110001	01000000101	10100000000	10100000000	10100000000	01010010000
	00000000000	00000000000	00000000000	00000000000	00000000000	00000000000
RAM2	10100000000	00101000011	10100000000	10100000000	10100000000	01101100111
	01000000000	00100010011	10100000000	10100000000	01000000000	01000000000
RAM3	01000000101	00101110001	10100000000	10100000000	10100000000	10010000000
	<u>001000000000</u>	00100000000	00100000000	00100000000	00000000000	00000000000

siguiente-marco (3 bits)	identificador (4 bits)	siguiente-grupo (1 bit)	siguiente-subFSM (2 bits)	he (1 bit)
-----------------------------	---------------------------	----------------------------	------------------------------	---------------

FIG. 12

	subFSM4		subFSM5		subFSM6	
	trans0	trans1	trans0	trans1	trans0	trans1
RAM0	0000000000	011000000000	00001010000	001000000000	101000000000	101000000000
	011000000000	011000000000	000000000000	000000000000	000000000000	000000000000
RAM1	101000000000	100000000111	000000000000	101000000000	101000000000	101000000000
	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
RAM2	011000000000	101000000000	101000000000	101000000000	101000000000	101000000000
	000000000000	000000000000	000000000000	000000000000	000000000000	000000000000
RAM3	101000000000	10010100011	101000000000	101000000000	101000000000	101000000000
	<u>000000000000</u>	000000000000	000000000000	000000000000	000000000000	000000000000

siguiente-marco (3 bits)	identificador (4 bits)	siguiente-grupo (1 bit)	siguiente-subFSM (2 bits)	he (1 bit)
-----------------------------	---------------------------	----------------------------	------------------------------	---------------

FIG. 13

Señal	Valor
IMM1	00100001001
IMM2	01000000011
IMM3	01000000000
IMM4	00000000000
IMM5	00000000000
IMM6	01100000000

Fig. 14



OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

②① N.º solicitud: 201400907

②② Fecha de presentación de la solicitud: 05.11.2014

③② Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TÉCNICA

⑤① Int. Cl.: **G06F7/02** (2006.01)

DOCUMENTOS RELEVANTES

Categoría	⑤⑥ Documentos citados	Reivindicaciones afectadas
A	SENHADJI-NAVARRO R et al. Finite Virtual State Machines. IEICE Transactions on Information and Systems Oct. 2012 IEICE Japan (10.2012) VOL: E95-D No: 10 Págs: 2544-2547 ISSN 0916-8532 (print) Doi: doi:10.1587/transinf.E95.D.2544, todo el documento.	1
A	VESPA L et al. Optimized memory based accelerator for scalable pattern matching. MICROPROCESSORS AND MICROSYSTEMS, 2 0091001 I PC BU SINESS PR ESS L TD. LONDON, GB 01.10.2009 VOL: 33 No: 7-8 Págs: 469-482 ISSN 0141-9331 Doi: doi:10.1016/j.micpro.2009.09.003, todo el documento.	1
A	EP 1691273 A1 (AGILENT TECHNOLOGIES INC) 16.08.2006, todo el documento.	1
A	US 2010100691 A1 (NOYES HAROLD B et al.) 22.04.2010, todo el documento.	1
A	US 2007233628 A1 (SHERWOOD TIMOTHY P et al.) 04.10.2007, todo el documento.	1

Categoría de los documentos citados

X: de particular relevancia

Y: de particular relevancia combinado con otro/s de la misma categoría

A: refleja el estado de la técnica

O: referido a divulgación no escrita

P: publicado entre la fecha de prioridad y la de presentación de la solicitud

E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

El presente informe ha sido realizado

☒ para todas las reivindicaciones

☐ para las reivindicaciones nº:

Fecha de realización del informe
22.03.2016

Examinador
M. L. Álvarez Moreno

Página
1/4

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

G06F

Bases de datos electrónicas consultadas durante la búsqueda (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

INVENES, EPODOC, WPI, Inspec

Fecha de Realización de la Opinión Escrita: 22.03.2016

Declaración**Novedad (Art. 6.1 LP 11/1986)**

Reivindicaciones 1
Reivindicaciones

SI
NO

Actividad inventiva (Art. 8.1 LP11/1986)

Reivindicaciones 1
Reivindicaciones

SI
NO

Se considera que la solicitud cumple con el requisito de aplicación industrial. Este requisito fue evaluado durante la fase de examen formal y técnico de la solicitud (Artículo 31.2 Ley 11/1986).

Base de la Opinión.-

La presente opinión se ha realizado sobre la base de la solicitud de patente tal y como se publica.

1. Documentos considerados.-

A continuación se relacionan los documentos pertenecientes al estado de la técnica tomados en consideración para la realización de esta opinión.

Documento	Número Publicación o Identificación	Fecha Publicación
D01	SENHADJI-NAVARRO R e t a l. Finite Virtual State Machines. IEICE Transactions on Information and Systems Oct. 2012 IEICE Japan (10.2012) VOL: E95-D No: 10 Págs: 2544-2547 ISSN 0916-8532 (print) Doi: doi:10.1587/transinf.E95.D.2544, todo el documento.	30.09.2012
D02	VESPA L et al. Optimized memory based accelerator for scalable pattern matching. MICROPROCESSORS AND MICROSYSTEMS, 20091001 IPC BUSINESS PRESS LTD. LONDON, GB 01.10.2009 VOL: 33 No: 7-8 Págs: 469 - 482 ISSN 0141-9331 Doi: doi:10.1016/j.micpro.2009.09.003, todo el documento.	01.10.2009
D03	EP 1691273 A1 (AGILENT TECHNOLOGIES INC)	16.08.2006
D04	US 2010100691 A1 (NOYES HAROLD B et al.)	22.04.2010
D05	US 2007233628 A1 (SHERWOOD TIMOTHY P et al.)	04.10.2007

2. Declaración motivada según los artículos 29.6 y 29.7 del Reglamento de ejecución de la Ley 11/1986, de 20 de marzo, de Patentes sobre la novedad y la actividad inventiva; citas y explicaciones en apoyo de esta declaración

El documento D01 se considera el más cercano a la solicitud. D01 muestra un reconocedor de patrones de bit basado en Máquinas de Estado Finitas que utiliza jerarquía de memoria de dos niveles (memoria principal y secundaria). La memoria principal sirve para almacenar los estados (subFSMs) activos en un momento dado y la secundaria almacena todos los estados. El sistema se encarga de transferir de forma dinámica la información los estados necesarios en cada momento desde la memoria secundaria hacia la principal. La figura 2 muestra la arquitectura general propuesta y puede verse que ambas memorias se encuentran controladas por la misma señal de reloj. Existen múltiples diferencias entre D01 y la solicitud. Por ejemplo la solicitud la frecuencia de la señal de reloj que controla la memoria secundaria es la mitad de la que controla la principal. En D01 la arquitectura de la memoria principal está constituida por una memoria convencional de doble puerto mientras que en la solicitud la arquitectura de dicha memoria principal está formada por un conjunto de registros y multiplexores con configuraciones particulares definidas en la reivindicación 1. No es posible derivar las características mostradas en la reivindicación 1 de la lectura del documento D01.

EL documento D02 muestra un reconocedor de patrones (aunque no de bits). En su apartado de introducción describe de forma muy clara el problema al que actualmente se enfrentan los reconocedores basados en memoria y es la necesidad de unos requisitos de memoria inmensos. Su propuesta particular es un reconocedor reconfigurable en el que existe un decodificador previo que reconoce un estado específico y en función del mismo genera todos los posibles subsiguientes estados. Al igual que en la solicitud evita una Máquina de Estados que contemple en una primera fase todos los estados posibles permitiendo reducir sus requisitos de memoria pero su propuesta es completamente distinta a la mostrada en la solicitud. No es posible derivar las características mostradas en la reivindicación 1 de la lectura del documento D02.

Los documentos de patentes D03 a D05 muestran otras alternativas también diferentes para realizar el reconocimiento de patrones. En todos los casos se trabaja con Máquinas de Estado Finitas y alguno de ellos utiliza multiplexores y/o registros pero de ninguno de ellos es posible derivar la utilización de una jerarquía de memorias (principal y secundaria). Tampoco se proporciona en ninguno de los documentos una gestión dinámica como la mostrada por el reconocedor la solicitud y no es posible derivar de su lectura ni el funcionamiento ni la arquitectura propuesta en la reivindicación 1.

La reivindicación 1 cumple los requisitos de novedad y actividad inventiva según los artículos 6 y 8 de la Ley de Patentes.