

(12) SOLICITUD INTERNACIONAL PUBLICADA EN VIRTUD DEL TRATADO DE COOPERACIÓN EN MATERIA DE PATENTES (PCT)

(19) Organización Mundial de la Propiedad
Intelectual
Oficina internacional



(43) Fecha de publicación internacional
19 de Febrero de 2009 (19.02.2009)

PCT

(10) Número de Publicación Internacional
WO 2009/022024 A1

(51) Clasificación Internacional de Patentes:
G06N 3/12 (2006.01) G11C 8/16 (2006.01)

(71) Solicitantes e

(21) Número de la solicitud internacional:
PCT/ES2007/000478

(72) Inventores: LLOPIS LLOPIS, Jose, Daniel [ES/ES]; Avenida Cortes Valencianas nº35, Piso 5a, Puerta 29a, E-46015 Valencia (ES). LLOPIS LLOPIS, Carlos [ES/ES]; Avenida Cortes Valencianas nº35, Piso 5a, Puerta 29a, E-46015 Valencia (ES). LLOPIS LLOPIS, Silvia [ES/ES]; Avenida Cortes Valencianas nº35, Piso 5a, Puerta 29a, E-46015 Valencia (ES).

(22) Fecha de presentación internacional:
2 de Agosto de 2007 (02.08.2007)

(74) Mandatario: UNGRIA LÓPEZ, Javier; Avda. Ramón y Cajal, 78, E-28043 Madrid (ES).

(25) Idioma de presentación: español

(81) Estados designados (a menos que se indique otra cosa, para toda clase de protección nacional admisible): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

(26) Idioma de publicación: español

[Continúa en la página siguiente]

(54) Title: ELECTRONIC SYSTEM FOR EMULATING THE CHAIN OF THE DNA STRUCTURE OF A CHROMOSOME

(54) Título: SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA

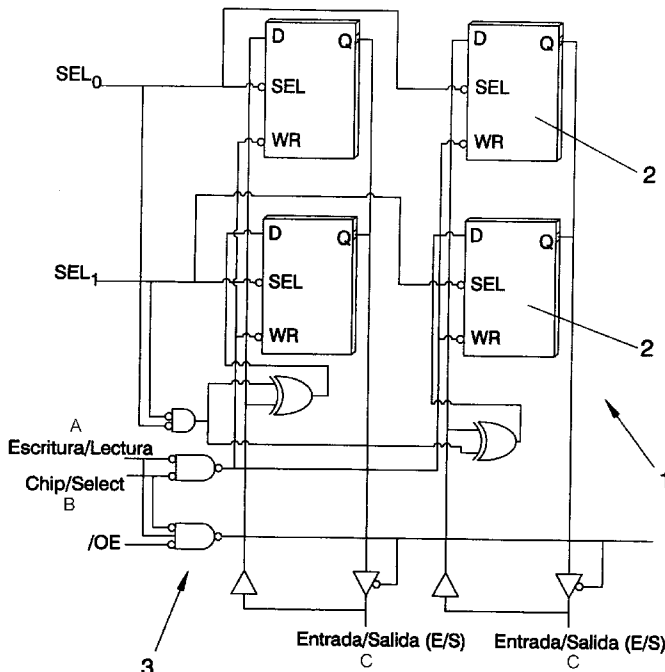


FIG. 3

A Write/Read
B Chip/Select
C Input/output (I/O)

(57) Abstract: The invention relates to an electronic system for emulating the chain of the DNA structure of a chromosome. The invention is characterised in that it includes means for the binary coding of the four types of nucleotides (A, G, C, T) that form the strands, such that the nucleotides that form complementary links are assigned complementary codes. The invention also includes units for storing (1) a code of a nucleotide and of the complementary thereof, said storage units being connected in series in order to form the chain of nucleotides that form each strand of a DNA chain and, in this way, to obtain the electronic structure thereof and allow selective access to a nucleotide, to a strand and to the two strands forming the DNA chain. The invention can be applied to any problem that can be represented with a graph which is coded in the storage units (1), in order to solve currently unsolvable problems relating to the calculation of paths in a parallel manner and in polynomial time.

(57) Resumen: Se caracteriza porque comprende medios de codificación binaria de los cuatro tipos de nucleótidos (A, G, C, T) que componen las hebras, de manera a los nucleótidos que forman enlaces complementarios se les asignan códigos complementarios; e incluye unidades de almacenamiento (1) de un código de un nucleótido y de su complementario, conectándose dichas unidades de almacenamiento en serie para formar la cadena - de nucleótidos que constituyen cada

[Continúa en la página siguiente]

WO 2009/022024 A1



HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

UG, ZM, ZW), euroasiática (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europea (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(84) Estados designados (*a menos que se indique otra cosa, para toda clase de protección regional admisible*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ,

Publicada:

— *con informe de búsqueda internacional*

hebra de una cadena de "ADN" y así obtener su estructura electrónica y permitir el acceso selectivo a un nucleótido, a una hebra y a las dos hebras que componen la cadena de "ADN". Se aplica en cualquier problema que puede ser representado mediante un grafo el cual se codifica en las unidades de almacenamiento (1), para permitir resolver problemas en el cálculo de trayectorias de forma paralela y en tiempo polinomial, que hoy en día son irresolubles.

- 1 -

SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA
ESTRUCTURA DE "ADN" DE UN CROMOSOMA

OBJETO DE LA INVENCION

La invención que nos ocupa, tal y como lo expresa el
5 enunciado de esta memoria descriptiva, tiene por objeto
proporcionar un sistema electrónico de emulación de la
cadena de la estructura de "ADN", para permitir, resolver
problemas de computación con "ADN", que en el caso de la
invención, es "ADN" inorgánico ya que se materializa
10 mediante circuitos electrónicos y no mediante "ADN"
orgánico, tal y como se realiza convencionalmente, y así
aprovechar las características de la naturaleza orgánica
del "ADN" a partir de un "ADN" inorgánico.

La invención consiste en un sistema hardware que
15 proporciona una estructura electrónica del "ADN" que
permite resolver problemas de trayectorias, redes y flujos,
los cuales son utilizados en los cálculos de trayectorias
de los sistemas "GPS" (Sistema de Posicionamiento Global),
cálculo de rutas en navegadores terrestres, marítimos o
20 aéreos, y también en problemas de control industrial en
procesos de producción u optimización de los flujos de
trabajo industriales.

En general la invención es aplicable en la resolución
de cualquier problema que pueda ser representado mediante
25 grafos que se asocian a la cadena de la estructura de
"ADN", tal y como será descrito más adelante.

ANTECEDENTES DE LA INVENCION

La matemáticas y la computación se unen de la mano, en
un área de la ciencia que es conocida como la teoría de
30 grafos. Los grafos son colecciones de elementos que se
denominan nodos o vértices, los cuales están unidos por
aristas o arcos, que a su vez, estas aristas pueden tener
dirección u orientación designada. Uno de los primeros
pioneros dentro de este campo, fue Leonard Euler, quien en
35 1736 se planteó un problema que es conocido como el

- 2 -

problema de los puentes de Königsberg, que era una ciudad prusiana que poseía siete puentes. Euler suscitó la cuestión, de si era posible recorrer toda la ciudad cruzando cada uno de los puentes, una y sólo una vez. En una representación esquemática de la ciudad se puede observar que los puentes unen cuatro porciones de tierra, por lo que la búsqueda de prueba y error, no conducen a ningún resultado, por ese motivo, si se puede representar este problema, mediante un grafo, se podrían aplicar técnicas, métodos y algoritmos de la teoría de grafos para encontrar la solución. A raíz de este problema se dio el primer paso hacia la teoría de grafos, la cual ha evolucionado a lo largo del tiempo y, donde existen numerosos grupos de investigación trabajando en áreas o campos de esta disciplina. Unos años más tarde, el matemático, físico, y astrónomo irlandés, William Rowan Hamilton planteó uno de los problemas más complejos que existen en la teoría de grafos, el problema de los caminos hamiltonianos o el problema del viajante que consiste en tener un camino o grafo en el que se visita cada vértice exactamente una vez. Este problema por su complejidad es considerado en computación como un problema NP-Completo. En este punto cabe señalar que los problemas matemáticos o de computación se clasifican por su complejidad en diferentes grupos:

- Complejidad P, donde los problemas pueden ser resueltos en una máquina determinista en tiempo polinómico.
- Complejidad NP, donde los problemas pueden ser resueltos por una máquina no determinista en tiempo polinómico.
- Complejidad NP-Completo, que son problemas de complejidad NP que tienen difícil solución en tiempos polinomiales, con una máquina no determinista.

También existen implementaciones paralelas para la solución del problema del camino hamiltoniano pero necesitan un gran volumen de recursos y comunicaciones entre los nodos conectados. Por ese motivo Adleman presentó una nueva alternativa para programar con "ADN" orgánico que presenta un masivo paralelismo en la realización de las operaciones y por tanto en la obtención de los resultados finales, pero con las limitaciones comentadas anteriormente. Diseñó un sistema para resolver el problema del camino hamiltoniano con programación "ADN" orgánico y, por lo tanto demostró que se podían resolver problemas NP-Completos en un tiempo computacional cercano al polinomial.

En este sentido puede citarse el documento de patente US7167847.

Ya en el siglo XX, el científico Edsger Dijkstra, planteó unos nuevos sistemas para el cálculo de rutas, de modo que no sólo se establecía cómo se debía de llegar de un punto a otro, sino que también se establecía que fuera en el menor tiempo posible. Dijkstra apoyó sus investigaciones en las ciencias de la computación, fruto de ello fue el algoritmo de Dijkstra. Este algoritmo se fundamenta en la determinación del camino más corto, dado un vértice origen en un grafo dirigido y con pesos positivos en las aristas. La idea es ir explorando todos los caminos más cortos que parten del vértice origen y, que llegan a todos los demás vértices. Una vez determinado el camino más corto desde el vértice origen, al resto de vértices que componen el grafo, el algoritmo se detiene. Este algoritmo se basa en una estrategia, conocida como algoritmo voraz, que no funciona con grafos con aristas de coste negativo. Más tarde Robert W. Floyd, junto con Wharshall, modificaron el algoritmo de Dijkstra, aunque, las modificaciones eran relevantes, aún no podían ser aplicados en redes industriales o en el cálculo de rutas. Viendo la problemática de una y las ventajas de la otra

- 4 -

ciencia, la invención establece una tecnología que une la electrónica y la genética, dos áreas hasta la fecha disjuntas. Esta tecnología está basada en circuitos integrados, electrónica y, "ADN" o genética, creando con
5 ello el primer "ADN" inorgánico con circuitos electrónicos, motivo por el cual la invención ha denominado a esta tecnología InChroSil (Inorganic CHROmosome based in SILicon).

El sistema de la invención tiene la gran ventaja de
10 que permite su reutilización, ya que el usuario puede utilizarlo tantas veces como se desee, ya que se materializa mediante circuitos electrónicos.

DESCRIPCIÓN DE LA INVENCIÓN

Para resolver los inconvenientes y conseguir los
15 objetivos anteriormente indicados, la invención ha desarrollado un nuevo sistema que consigue la obtención de un "ADN" inorgánico.

Para comprender la invención, es necesario realizar una introducción genética de los cromosomas, que están
20 formados por "ADN" y proteínas. A su vez el "ADN" está compuesto por dos fibras longitudinales unidas por centrómero. Estas fibras se llaman cromatidas y representan dos hebras idénticas del "ADN" duplicado.

Por consiguiente el "ADN" está compuesto por hebras o
25 cadenas de nucleótidos, los cuales se diferencian en cuatro tipos que se numeran a continuación:

- Adenina, se representa mediante la letra A.
- Guanina, se representa mediante la letra G.
- Citosina, se representa mediante la letra C.
- 30 - Timina, se representa mediante la letra T.

Por consiguiente el "ADN" está compuesto por dos hebras enlazadas y cada hebra contiene innumerables nucleótidos. En una misma hebra no existen nucleótidos complementarios, es decir, pueden unirse sin restricción
35 alguna. En cambio al enlazar las dos hebras, existen

- 5 -

restricciones de complementariedad, de tal modo, que una Adenina (A) de una hebra sólo se puede unir con las Timinas (T) de la hebra complementaria, de igual modo las Guaninas (G) sólo se pueden unir con las citosinas (C).

5 El sistema de la invención se caracteriza porque comprende medios de codificación binaria de los diferentes cuatro tipos de nucleótidos que componen las hebras, de manera que a los nucleótidos que forman enlaces complementarios se les asignan códigos complementarios
10 mediante la inversión de uno de ellos. Además la invención incluye unidades de almacenamiento de un código de un nucleótido y de un código seleccionado entre su código complementario o ningún código; todo ello para obtener un nucleótido y su complementario o únicamente un nucleótido,
15 conectándose dichas unidades de almacenamiento en serie para formar la cadena de nucleótidos que constituyen cada hebra de una cadena de "ADN". Por consiguiente mediante la codificación comentada, que se almacena en las unidades de almacenamiento, se obtiene una estructura electrónica a la
20 que se permite el acceso a un nucleótido y/o a su complementario, a una hebra y a las dos hebras de la cadena de "ADN" según se requiera, mediante circuitos electrónicos conocidos que realicen la selección de las diferentes unidades de almacenamiento en función de los requerimientos
25 deseados.

La invención prevé diferentes módulos que están constituidos por agrupaciones de unidades de almacenamiento, en las que en cada módulo se almacena una cadena de "ADN", estando los módulos conectados en filas y
30 columnas, de manera que se permita acceder a cada cadena de "ADN" mediante la selección de filas o columnas.

De la descripción realizada se deduce que la primera aplicación de la invención consiste en su empleo como banco de datos de "ADN".

35 En la realización preferente de la invención el código

- 6 -

asignado a cada nucleótido comprende dos bits y su almacenamiento depende de las entradas aplicadas a cada unidad de almacenamiento mediante medios electrónicos, de manera que cada unidad de almacenamiento puede ser
5 codificada según el nucleótido que se desee y su complementario.

La invención prevé la incorporación de medios de codificación de numeración en base 4 compuesta por cuatro nucleótidos de un cromosoma (A, G, C y T).

10 Además prevé que los dos bits correspondientes a cada nucleótido adquieren las cantidades decimales de 0 a 3 de manera que se permite conformar una pluralidad de presentaciones numéricas determinadas por la posición de los dos bits y por el valor que recibe en cada posición.
15 Esta codificación permite que cualquier sistema de codificación que tenga correspondencia con el decimal o binario, puede ser codificado con el sistema de numeración y codificación de la invención.

La codificación comentada permite que la invención
20 incorpore medios para efectuar operaciones como son sumas, restas, multiplicaciones y/o divisiones, así como codificar cualquier número en coma fija o flotante de simple y doble precisión utilizando codificación en complemento A1 y en complemento A2.

25 Al igual que las memorias del estado de la técnica, las unidades de almacenamiento de la invención pueden ser volátiles, pero en este caso la invención denomina InCroSil Access Memory (IAM), pudiendo ser también memorias no volátiles de sólo lectura que la invención denomina
30 InCrosil Read Only Memory (IROM). La invención incorpora medios de acceso estructurado a los módulos y unidades de almacenamiento mediante localización de una cadena de "ADN", posterior localización de una hebra y a continuación mediante la designación de un nucleótido o grupo de
35 nucleótidos, de manera que se pueda acceder a toda la

- 7 -

cadena o parte de ella. Por consiguiente, mediante los módulos y unidades de almacenamiento de la invención no se accede por filas o columnas como sucede en las memorias convencionales, sino que permite tres accesos diferentes, según ha sido comentado, por lo que se permite un acceso a partir de tres selecciones que puede denominarse acceso tridimensional.

A su vez, tal y como ya fue comentado con anterioridad una cadena de "ADN" puede ser considerada como una fila o como una columna, es decir, el sistema que administra la memoria IAM, puede estipular cómo estarían almacenadas las cadenas de "ADN" en la memoria, ya sea por filas o por columnas.

En base a toda la estructura descrita, se comprende que la invención puede incorporar medios para codificar en las unidades de almacenamiento, que forman hebras, representaciones de problemas mediante grafos que comprenden elementos constituidos por vértices y aristas y opcionalmente pueden incorporar el peso o dirección de la arista, de manera que el sistema de la invención codifica a una hebra en un primer conjunto que contiene vértices y aristas y no contiene el peso de éstas, correspondientes a elementos iniciales y los correspondientes a los elementos que no contienen elementos finales; habiéndose previsto que en la hebra complementaria se codifique un segundo conjunto que contiene elementos del grafo constituidos igualmente por vértices y aristas correspondientes a los elementos finales y los correspondientes a los elementos que no contienen elementos finales. Cada una de las unidades de almacenamiento de ambas hebras están unidas entre sí mediante comparadores que forman una matriz para que se realice el procesado de ambas hebras de forma paralela así como las comparaciones entre todos los elementos de ambos conjuntos lo que permite resolver problemas que pueden representarse mediante grafos.

Los comparadores forman una matriz y están conectados a un buffer al que se aplican medios de recuperación de caminos de orden polinomial para determinar la ruta más óptima entre dos puntos del grafo.

5 Como fue señalado las aristas de los grafos pueden tener peso o dirección, en cuyo caso la invención comprende medios adicionales de almacenamiento del peso de una arista, así como medios para sumar los pesos de cada arista y obtener los pesos totales de cada arista, habiéndose
10 previsto que además incorpore medios de ordenación de los pesos totales de cada arista según criterios previamente establecidos, lo que permite resolver problemas de algoritmos conocidos como son Dijkstra, Floyd-Warshall y Bellman-Ford, Ford-Fulkenson.

15 La estructura descrita permite su aplicación en problemas de flujo como pueden ser flotas de vehículos, optimización de procesos industriales, flujos de trabajo, programas de acoplamientos y de redes de comunicaciones, para lo que estos problemas se representan como un grafo.
20 Así por ejemplo la información e instrucción de la maquinaria de una planta industrial puede ser representada como un grafo para resolver el problema planteado.

 Igualmente la invención prevé codificar el problema del camino hamiltoniano en las unidades de memoria de
25 almacenamiento que constituyen las hebras, en una de las cuales se codifica el grafo completo y en las complementarias el grafo completo con restricciones conocidas. En este caso las hebras están conectadas a comparadores formando una matriz para comparar las aristas
30 de ambos grafos dos a dos de forma paralela para indicar si los bits del resultado de cada comparación son todos iguales y se refieren a una misma arista y en consecuencia ésta pertenece a ambos grafos.

 Al resultado de la comparación se la aplican medios
35 para crear niveles del árbol de wallace indicativo de si la

- 9 -

arista es camino o no lo es, de forma que se resuelve el problema del camino hamiltoniano.

En este caso los comparadores que comparan las aristas de ambos grafos dos a dos, están constituidos por semisumadores lógicos sin acarreo, cuya salida está conectada a una puerta lógica de detección de un solo bits del resultado de la comparación, para obtener un solo dato, tal y como fue señalado, y que indica si los bits del resultado de cada comparación pertenecen a ambos grafos de los conjuntos codificados.

Los medios para crear niveles del árbol de wallace comprenden unas primeras puertas lógicas por cada columna de la matriz de comparadores que representa el grafo completo y están conectadas a las salidas de los comparadores para confirmar la pertenencia de dicha arista al conjunto de aristas del grafo con restricciones de forma que indica si es o no camino.

Las salidas de las primeras puertas lógicas están conectadas a una segunda puerta lógica de detección de si todas las aristas son camino, de manera que la invención determina el camino hamiltoniano. Además incorpora medios de almacenamiento del resultado obtenido.

En este caso la invención también prevé la posibilidad de que se tenga en cuenta el peso de las aristas, para lo que comprende medios para recibir los caminos encontrados y establecer la correspondencia entre peso y arista y obtener los pesos de cada camino. En este caso también comprende medios de ordenación de los pesos de los caminos según criterios previamente establecidos para la elección del mejor camino.

En otra realización de la invención cuando se codifiquen los vértices y, por lo tanto se establezca el valor de las aristas que los unen, el sistema almacena en esa dirección de memoria, el valor del peso de esa arista en el grafo. En este caso se busca el valor de las aristas

-10-

en la memoria, ya que como se conoce su dirección, se conoce su valor, es decir se codifica una referencia a la memoria, como es su dirección y no el valor de la arista. Los pesos obtenidos se suman para obtener los totales de cada camino que se almacenan en memorias con direcciones consecutivas y mediante medios de ordenación de los pesos de los caminos según criterios previamente establecidos, se efectúa la elección del mejor camino.

Esta configuración también permite la aplicación de la invención a la solución de problemas de ciclos hamiltonianos mediante una codificación indicativa de que el vértice de salida del grafo es igual al vértice destino.

También cabe señalar que la invención incorpora medios para codificar un grafo de manera que los vértices representan las aristas y a la inversa y así obtener qué secuencia de vértices se debe seguir para pasar por todas las aristas y permitir resolver el problema del camino euleriano.

Mediante la incorporación de medios para codificar en las hebras que el vértice inicial es el mismo que el vértice final, se permite la aplicación de la invención para resolver ciclos eulerianos.

La invención también permite establecer codificación del peso de las aristas en la resolución de caminos eulerianos o ciclos eulerianos, de forma que permite plantear nuevos problemas de optimización de los caminos eulerianos.

A continuación para facilitar una mejor comprensión de esta memoria descriptiva y formando parte integrante de la misma, se acompañan una serie de figuras en las que con carácter ilustrativo y no limitativo se ha representado el objeto de la invención.

BREVE ENUNCIADO DE LAS FIGURAS

Figura 1.- Muestra una representación de la cadena de "ADN" orgánica.

Figura 2.- Muestra los diferentes tipos de estructuras de cadena "ADN" y las uniones entre nucleótidos.

Figura 3.- Muestra un posible ejemplo de realización de una implementación de una unidad de almacenamiento de un
5 nucleótido y de su complementario.

Figura 4.- Muestra una tabla de la correspondencia entre diferentes sistemas de numeración comparados con la codificación en base 4 de la invención.

Figura 5.- Muestra una tabla correspondiente a la
10 operación suma de dos nucleótidos según la codificación de la invención.

Figura 6.- Muestra una tabla equivalente a la figura anterior, pero para la resta.

Figura 7.- Muestra una representación de un posible
15 ejemplo de resta.

Figura 8.- Muestra una tabla equivalente a las figuras 5 y 6, pero para la multiplicación.

Figura 9.- Muestra un posible ejemplo de una multiplicación.

Figura 10.- Muestra una tabla de representación de
20 coma flotante en simple precisión según la codificación de la invención.

Figura 11.- Muestra una tabla de diferentes combinaciones del exponente (como fija, complemento A1, A2
25 y exceso 4^4).

Figura 12.- Muestra una tabla de representación de coma flotante de doble precisión.

Figura 13.- Muestra una representación esquemática de la unidad de almacenamiento básica representada en la
30 figura 3.

Figura 14.- Muestra una representación esquemática equivalente a la figura anterior, pero según una configuración más detallada.

Figura 15.- Muestra una representación de la

estructura de una posible agrupación de memorias volátiles y no volátiles representadas en las figuras 13 y 14.

Figura 16.- Muestra una representación de un posible ejemplo de un grafo.

5 **Figura 17.-** Muestra una posible implementación para el cálculo de caminos.

Figura 18.- Muestra una representación de un grafo hamiltoniano dirigido y no ponderado.

10 **Figura 19.-** Muestra una representación del grafo del camino hamiltoniano.

Figura 20.- Muestra una cadena de "ADN" inorgánico de la invención referente a una agrupación posible de aristas y vértices para resolver el problema del camino hamiltoniano.

15 **Figura 21.-** Muestra una codificación de los vértices y aristas según la figura anterior.

Figura 22.- Muestra una codificación de los vértices y aristas detallada de la figura anterior.

20 **Figura 23.-** Muestra el conjunto de cadenas de "ADN" inorgánicos de la invención con todas las posibles soluciones del camino.

Figura 24.- Muestra la representación de un semisumador lógico que se aplica en el sistema de la invención.

25 **Figura 25.-** Muestra una representación de una posible implementación para efectuar la comparación de aristas en aristas en paralelo para la resolución del problema hamiltoniano.

30 **Figura 26.-** Muestra un diagrama de bloques funcional de la implementación de la invención para la resolución de problemas hamiltonianos.

Figura 27.- Muestra el circuito completo para la solución del camino hamiltoniano.

35 **Figura 28.-** Muestra un diagrama de bloques funcional de una primera implementación para la resolución del

problema del camino hamiltoniano ponderado.

Figura 29.- Muestra un diagrama de bloques funcional de una segunda implementación para la solución del problema del camino hamiltoniano ponderado.

5 DESCRIPCIÓN DE LAS FORMAS DE REALIZACIÓN PREFERIDAS

A continuación se realiza una descripción de la invención basada en las figuras anteriormente comentadas.

Como ya ha sido comentado con anterioridad, un cromosoma está formado por "ADN" y proteínas. A su vez el
10 "ADN" está compuesto por hebras o cadenas de nucleótidos, los cuales se diferencian en cuatro tipos: Adenina (A), Guanina (G), Citosina (C) y Timina (T).

El "ADN" está compuesto por dos hebras enlazadas y cada hebra contiene innumerables nucleótidos. En una misma
15 hebra no existe nucleótidos complementarios, es decir, pueden unirse sin restricción alguna. En cambio al enlazar las dos hebras, existen restricciones de complementariedad, de tal modo, que un Adenina (A) de una hebra sólo se podrá unir con las Timinas (T) de la hebra complementaria, de
20 igual modo con las Guaninas (G) con las Citosinas (C), es otro enlace posible. Debido a esta organización de las hebras, la codificación del ADN no puede ser al azar y debe tener una estructura previamente definida (véase figura 1).

Para modelar el comportamiento orgánico del cromosoma se
25 diseño una codificación, que se describe de forma detallada en secciones posteriores, para cada uno de los nucleótidos, en la siguiente enumeración se presenta una parte de esta codificación:

- 30 • Adenina (A) se le asigna el valor de "00".
 • Guanina (G) se le asigna el valor de "01".
 • Citosina(C) se le asigna el valor de "10".
 • Timina (T) se le asigna el valor de "11".

Una vez codificados los nucleótidos, siguiendo la codificación implementada, el siguiente paso es la implementación a nivel físico mediante circuitos electrónicos (por ejemplo, biestables u otro dispositivo que almacene los nucleótidos o estados), y establecer la estructura física de los mismos dentro de las hebras y sus relaciones con la hebra complementaria. La idea que se planteó inicialmente era un circuito, el cual estaba dividido en cuatro partes iguales. Cada parte del circuito estaba separado de la otra mediante un material aislante, para no producir interferencias entre los distintos componentes que se almacenan en ellas. Por otro lado, en cada parte del circuito, se colocarían los componentes que codifican a un determinado nucleótido inorgánico. Esto permite la libre elección de un determinado nucleótido dentro de la posición de una hebra artificial mediante las conexiones permitidas entre los diferentes nucleótidos.

La longitud de la hebra dependerá del número de circuitos colocados en forma de pila, creando así un chip tridimensional, cosa que permitirá la emulación de la estructura orgánica (cromosoma). También se puede realizar a nivel planar, la elección de una tecnología u otra, dependería de los recursos existentes y las necesidades finales del proyecto.

Una vez codificado el circuito con los cuatro nucleótidos, se debe conectar cada circuito con su circuito superior, mediante conexiones en serie, de esta forma se obtiene una hebra o cadena de nucleótidos. Por otra parte se pensó en las conexiones en paralelo o enlaces con la hebra complementaria, hay que tener en cuenta las relaciones de complementariedad de los distintos nucleótidos, relación explicada anteriormente. Estos enlaces se codificaran dentro del propio circuito mediante componentes lógicos.

-15-

Con este sistema se definiría el nucleótido de una hebra y con el circuito combinacional o método de inversión se obtendría de forma inmediata el nucleótido complementario en la otra hebra. Si esto se aplica en cada circuito
5 apilado, se produciría una reacción en cadena, que nos permitiría definir la hebra completa. De esta forma se evitaría inconsistencias entre los enlaces paralelos, ya que no se pueden unir dos nucleótidos que no sean complementarios.

10 Por otra parte, los enlaces en serie o entre nucleótidos de una misma hebra se realizará por posición concreta, es decir, se realizará de una forma contigua, por lo se podrán realizar tanto accesos secuenciales como aleatorios a un determinado nucleótido dentro de una determinada hebra. Con
15 este sistema de acceso, y las conexiones paralelas, se podrá activar un determinado par de nucleótidos en cualquier momento y poder conseguir innumerables combinaciones y, cromosomas diferentes.

Este planteamiento inicial desaprovecha mucha
20 información en cada operación, ya que existen partes del circuito que no se utilizan, aprovechándose sólo la mitad del circuito, es decir, el planteamiento inicial era realizar un circuito con los cuatro nucleótidos de forma estática, de esta forma si se elegía un nucleótido
25 determinado, en el mismo circuito se activaba el nucleótido de la hebra complementaria. Indicar que en este planteamiento inicial los nucleótidos de un mismo nivel residen en el mismo circuito y, se activan por la activación de uno de los dos. Por ese motivo si se
30 configuraba un par de nucleótidos se desaprovecha la mitad de los recursos del circuito, por esa causa se pensó en un planteamiento más dinámico, donde los nucleótidos no estuvieran predefinidos en el circuito y que dependiendo de unas determinadas entradas obtuviéramos un determinado
35 nucleótido u otro.

Con este nuevo planteamiento los nucleótidos en un circuito pasan a ser dos y, además son dinámicos, es decir, dependiendo de unas determinadas entradas se obtiene un determinado tipo de nucleótido. Esta reducción hace que se
5 aprovechen mejor los recursos del circuito, ya que se pasa de un circuito de cuatro partes a un circuito de dos. Toda esta simplificación no afectará la estructura del nucleótido, ya que seguirá estando compuesto de dos biestables o componentes electrónicos equivalentes, aunque
10 el valor almacenado en ellos dependerá de las entradas aplicadas, ya que deben de ser dinámicos.

Con este nuevo diseño se pasa de una estructura de pila a una forma totalmente helicoidal como se ve en la figura 1. Por tanto, con este planteamiento se simplifica el
15 circuito y la utilización del mismo, además en fase de fabricación existe una reducción considerable en los costes de producción y la utilización de recursos digitales y electrónicos, al aprovechar mejor el espacio y los componentes

20 Las conexiones inorgánicas del cromosoma se dividen en dos partes muy importantes: la conexión en serie, donde un determinado nucleótido pueda conectarse con otro nucleótido del mismo tipo u otro y, por otra parte las conexiones paralelas donde un determinado nucleótido de una hebra sólo
25 se puede conectar con su complementario o ninguno. Por lo tanto, si la hebra tiene todas sus conexiones o enlaces se le dice que la hebra está completa, mientras que si uno de los nucleótidos no tiene correspondencia con su complementario, la hebra está incompleta. Esta
30 característica se tuvo en cuenta a la hora de diseñar las conexiones en paralelo entre nucleótidos de hebras complementarias, ya que en la activación de uno de los nucleótidos se indica al mismo si va producirse la activación del nucleótido complementario. Con este proceder
35 se emula cualquier estructura de cromosomas, ya que no

todos poseen las mismas características y todos los pares de nucleótidos.

Para explicar las posibles combinaciones de las cadenas de ADN, en la figura 2 se presentan las combinaciones que se podrían dar. Estas estructuras permiten poder realizar operaciones con las cadenas de ADN, de tal modo que se pueden realizar problemas complejos, tales como los NP o NP-Completos. Como se ha comentado la invención se basa completamente en la relación existente entre el par nucleótido y todos sus posibles enlaces entre nucleótidos de una misma hebra, como entre nucleótidos de hebras complementarias.

En la figura 3 se representa un ejemplo de implementación que se ajusta a la descripción anteriormente comentada y en la que una unidad de almacenamiento 1 se materializa mediante cuatro biestables 2 de tipo D, de forma que los dos superiores están previstos para la codificación de un nucleótido y los dos inferiores para la codificación de su complementario.

Por ejemplo, supongamos una implementación como la figura 3, si sólo se quiere utilizar el nucleótido de arriba (hebra incompleta) se activara la señal SEL0 de la figura 3, de este modo la señal de entrada/salida (E/S) nos permite establecer el valor que se desee que almacenen los biestables 2 del nucleótido de arriba. En cambio si sólo se quiere el nucleótido de abajo se activará la señal SEL1 de la figura 3 y, del mismo modo la señal (E/S) permitirá establecer el valor de los biestables del nucleótido de abajo.

En cambio, si se quiere que los dos nucleótidos se activen (hebra completa) se activaran las señales SEL0 y SEL1 y mediante las señales (E/S) se establece el valor de uno de los nucleótidos, mientras que el otro nucleótido se activara con la utilización de puertas XOR. Con este diseño de un par de nucleótidos se pueden conseguir todas las

combinaciones posibles entre nucleótidos de hebras complementarias y por lo tanto emular el comportamiento de las posibles combinaciones de un cromosoma orgánico.

5 Las puertas lógicas de la unidad de almacenamiento 1 permite codificar los enlaces dentro del propio circuito, así como establecer que se desea almacenar un nucleótido o obtener su valor mediante la señal de escritura/lectura (W/E).

10 La representación esquemática de esta unidad de almacenamiento 1 se muestra en las figuras 13 y 14 tal y como será descrito con posterioridad.

También prevé la codificación de un sistema de numeración en base 4, el cual está compuesto por dos nucleótidos que forman una hebra de "ADN". Este nuevo sistema es denominado en la invención Cod-InChroSil (CODification - Inorganic CHROMosome based in SILicon), de forma que se proporciona la posibilidad de aumentar la precisión de los sistemas numéricos actuales, además de poder utilizar en las operaciones, las características intrínsecas de la estructura de una hebra de ADN.

20 El sistema de numeración en base 4 de la invención utiliza como símbolo los nucleótidos de una hebra de "ADN" A, G, C y T. Convencionalmente es conocida la codificación del "ADN" orgánico de forma macroscópica, ya sea mediante la codificación de una hebra o porción, o también codificando las operaciones con sus operandos. Estas codificaciones no permiten tanta flexibilidad, porque limitan el número de combinaciones que se pueden realizar. Por ese motivo la invención consiste en un sistema de numeración donde se codifica el ADN de forma microscópica, es decir, a nivel de nucleótido que forma la hebra de ADN. Esta característica de atomicidad, permite realizar combinaciones para generar un sistema de numeración que tenga una potencia y escalabilidad, necesarias para la representación de cantidades numéricas. Se define la

25
30
35

codificación de un nucleótido como una secuencia de dos bits, los cuales adquieren las cantidades decimales de 0 a 3. Este sistema de numeración que presentamos, permite conformar numerosas representaciones numéricas, conformadas por la posición y el valor del símbolo dentro de la cadena numérica, resaltar que este sistema numérico es posicional y que el símbolo tiene dos significados, por una parte la posición dentro de la cadena numérica y, por otra parte el valor que recibe el símbolo. Con la combinación de ambas informaciones, se permite obtener parte del valor final de la cantidad numérica que se desee representar. En la figura 4, se observan la correspondencia entre nuestro sistema de numeración de la invención y el resto de sistema de numeración que se están utilizando en la actualidad.

Siempre un ejemplo es más aclaratorio que un conjunto de palabras, por ese motivo, si se considera el número 4567_{10} en base decimal, la representación correspondiente en binario es 0001000111010111_2 , en octal es 010727_8 , en hexadecimal es $11D7_{16}$ y finalmente en la codificación de Cod-InChroSil es $AGAGTGGT_4$. También se puede cambiar de un sistema numérico utilizando un desarrollo polinómico del número (teorema fundamental de la numeración), donde las cifras del número poseen dos valores, por una parte el intrínseco como cifra y luego el posicional, de este modo la ecuación de cambio viene determinada como la siguiente:

$$n^{\circ} (X_{n-1} \dots X_2 X_1 X_0)_b = X_{n-1} b^{n-1} + \dots + X_2 b^2 + X_1 b^1 + X_0 b^0$$

Por ejemplo, tenemos el siguiente número codificado en Cod-InChroSil $AGAGTGGT_4$, el número en decimal es el siguiente:

$$\begin{aligned} AGAGTGGT_4 &= A \cdot 4^7 + G \cdot 4^6 + A \cdot 4^5 + G \cdot 4^4 + T \cdot 4^3 + G \cdot 4^2 + G \cdot 4^1 + \\ T \cdot 4^0 &= 0 \cdot 4^7 + 1 \cdot 4^6 + 0 \cdot 4^5 + 1 \cdot 4^4 + 3 \cdot 4^3 + 1 \cdot 4^2 + 1 \cdot 4^1 + 3 \cdot 4^0 = \end{aligned}$$

35

4567₁₀

La operación inversa de un número decimal a una codificación *Cod-InChroSil*. Suponiendo el número en decimal 189_{10} , el número en *Cod-InChroSil* es el siguiente:

5

$$189 \text{ div } 4 = 47 \rightarrow \text{resto}(189,4) = 1 \rightarrow G$$

$$47 \text{ div } 4 = 11 \rightarrow \text{resto}(47,4) = 3 \rightarrow T$$

$$11 \text{ div } 4 = 2 \rightarrow \text{resto}(11,4) = 3 \rightarrow T$$

$$2 \text{ div } 4 = 0 \rightarrow \text{resto}(2,4) = 2 \rightarrow C$$

10

$$189_{10} = \text{CTTG}_4$$

Se muestran ejemplos, supongamos se tiene el número $873,875_{10}$ y se quiere pasar a *Cod-InChroSil*, la mecánica es la siguiente:

15

- La parte entera es calculada como el ejemplo anterior TGCCG.
- $0,875 \times 4 = 3,5 \rightarrow$ la parte entera es 3 entonces es T.
- $0,5 \times 4 = 2 \rightarrow$ la parte entera es 2 entonces es C.
- Finalmente, la parte decimal es cero, entonces el número final es TGCCG,TC₄

20

De forma general en un sistema de numeración se puede representar una cantidad numérica de la siguiente forma N_b
 25 = $[a_{n-1} a_{n-2} a_{n-3} \dots a_3 a_2 a_1 a_0, a_{-1} a_{-2} a_{-3} \dots a_k]_b$, donde $n+k$ indica la cantidad de dígitos de la cifra, a su vez n indica el número de dígitos en la parte entera, mientras que k indica la parte fraccionaria. Finalmente indicar, que con la utilización del sistema de numeración *Cod-InChroSil* lo
 30 se hace compatible con el resto de sistemas de numeración actuales, prueba de ello la encontramos en los párrafos anteriores, que se ha pasado de decimal a *Cod-InChroSil* y de *Cod-InChroSil* a decimal.

Todo sistema de numeración necesita de unas reglas y operaciones para poder realizar cálculos entre sus operandos. Por ese motivo en *Cod-InChroSil*, consideran las habituales, como son; la suma, resta, multiplicación, división, etc. En primer lugar se define la suma, debido a que es una operación altamente utilizada en numerosos cálculos matemáticos. La suma es una operación aritmética definida sobre un conjunto de números y sus estructuras (vectores, conjuntos, etc.). En la invención, la suma se define como una operación sobre números codificados en *Cod-InChroSil*. Como bien es sabido, la suma posee diferentes propiedades que la caracterizan y que serán aplicables en la invención, éstas son; la propiedad conmutativa, donde el orden de los operandos no altera el resultado, la propiedad asociativa, el elemento neutro y, finalmente el elemento opuesto. En la figura 5 se dan las directrices para la suma de dos nucleótidos. Resaltar, que como en la suma decimal que cuando se supera la base se dice la frase y *llevamos una*, en esta operación de suma también se tiene en cuenta el acarreo ocasionado por un desbordamiento, lo podemos observar en la figura 5 donde hay celdas de la tabla que posee dos símbolos, el primero de ellos representa el desbordamiento producido y, el segundo el resultado de la operación, es decir, suponiendo que se quieren sumar dos números ATCGATA y TAGCCAA, en este caso no se produce ningún acarreo.

	Cuenta o acarreo
	ATCGATA número 1 (3660 en decimal)
+	TAGCCAA número 2 (12704 en decimal)

	TTTTCTA resultado (16364 en decimal)

En este otro ejemplo tenemos los siguientes números ATCGATA y ATCTTTC, en este caso si que se producen acarreos.

G G G G G cuenta o acarreo
 A T C G A T A número 1 (3660 en decimal)
 + A T C T T T C número 2 (3838 en decimal)

 5 G↑T↑G↑G↑A↑CC resultado (7498 en decimal o GTGGACC)

La siguiente operación que se define es la resta en codificación Cod-InChroSil. La resta es conocida como una
 10 operación aritmética opuesta a la suma, en el caso particular de la invención la resta entre dos nucleótidos se define siguiendo las siguientes directrices de la figura 6.

Como se observa en la figura 6 se ha tenido en cuenta el
 15 acarreo ocasionado al restar dos nucleótidos, por ese motivo y al igual que se hizo con la operación de suma, en una misma celda de la tabla (véase figura 6), se escriben dos símbolos, el primero hace referencia al desbordamiento o acarreo y el segundo hace referencia al resultado. Para
 20 ver como funciona la resta mediante esta codificación se muestra un ejemplo aclaratorio de esta operación. Los siguientes números en Cod-InChroSil, TGAT₄ (en decimal es el 211) y GCCC₄ (en decimal es el 106), el resultado de la resta es 105 en decimal GCCG₄ en Cod-InChroSil,
 25 seguidamente se realiza el cálculo paso a paso con la resta que ha sido definida en la figura 7.

Como observamos en la figura 7, cuando se resta T - C = G no se produce acarreo, pero si cuando se resta A - C = GC (la siguiente cifra del numero), el resultado es C, pero el
 30 acarreo es G, esta cantidad tiene que sumarse a la siguiente cifra, en este caso es C. De esta operación de suma se obtiene T. A continuación se resta G del numero 1 al nuevo resultado (T), obteniendo el valor de GC, de nuevo se ha producido un acarreo con valor G. Se suma este
 35 acarreo con la siguiente cifra del numero 2 y el resultado

-23-

se resta con su homólogo posicional del número 1, el cual se obtiene G. Como se observa el mecanismo es el mismo que en la resta en decimal, cuando la cifra del número 1 es mayor a la cifra del número 2 se realiza la resta y no se provoca desbordamiento, pero cuando es al contrario, se calcula cuantos llevamos desde la cifra del número 2 hacia la cifra del número 1, siendo este el resultado. A continuación se suma una unidad a la siguiente cifra y se restan ambas cifras de la misma posición. La resta también se puede establecer como la suma de dos números, donde uno de ellos se utiliza el opuesto (en este caso complementario) más la suma del nucleótido G (más tarde se explicará en secciones posteriores este formato de representación). Como se comentó en *InChroSil* la cadena de ADN está formada por dos hebras que se rigen por propiedades de complementariedad, es decir, un nucleótido de una hebra no se puede unir con cualquier nucleótido de la hebra complementaria, sólo con su complementario establecido. Esto permite que cuando se codifique un número en *Cod-InChroSil* en una cadena de ADN, se esté codificando automáticamente su complementario en la hebra complementaria. Esta propiedad intrínseca de las cadenas de ADN, permite aplicar operaciones de suma muy fácilmente, debido a que ya se posee el opuesto o complementario del número, sólo faltaría que este número fuera sumado por el nucleótido G, formato de representación denominado complemento a 2 y, que en líneas posteriores a este documento se explica con detalle. Se recurre otra vez al ejemplo, para dar una visión más aclaratoria de la operación resta en este sistema de numeración. Si se tiene el número TGCG (217 en decimal) y se quiere restarlo con el número CCCT (171 en decimal), la resta decimal nos da el siguiente resultado; 46 (ACTC). En este sistema de numeración se aplican los siguientes pasos:

35

- El primer operando no se modifica (TGCG)
- Se toma el complementario del segundo operando, es decir, A se transforma en T y viceversa y, C se transforma en G y viceversa, en este caso en concreto es GGGA.
- Al número resultante del punto 2, se suma G (GGGG).
- Se suma el primer operando y el número del punto 3.
- Se elige, el número máximo de nucleótidos de los dos operandos y, este será el resultado de la operación, en este ejemplo es ACTC.

En forma de ecuación quedaría de la siguiente forma:

$$\text{Resol}_{\text{Resta}} = \text{Operando}_{1_4} + \text{Complemento}(\text{Operando}_{2_4}) + G_4,$$
donde $\text{Resol}_{\text{Resta}}$ es el resultado de la resta, Operando_{1_4} es el primer operando en codificación *Cod-InChroSil*, Operando_{2_4} es el segundo operando en la misma codificación que el Operando_{1_4} y, finalmente G_4 que representa el literal del nucleótido G de una hebra de ADN. Por otra parte se define la función *Complemento*, como una función inyectiva tal que $\forall x, y \in \{A, G, C, T\}, \exists$ una f , función de complemento, tal que $f: \{A, G, C, T\} \rightarrow \{A, G, C, T\}$, donde $f(x) = y, x \neq y$, y es el nucleótido complementario de x.

La siguiente operación que se define es la multiplicación. La multiplicación o producto se define como una operación aritmética donde se realizan sucesivas sumas. Al igual que en las anteriores operaciones se dan unas directrices para la multiplicación a nivel de nucleótidos, estos productos se observan en la figura 8.

Se tienen los siguientes números GTTG (125 en decimal) y GGA (20 en decimal) entonces el producto de ambos es el que se muestra en la figura 9. Como se observa se realiza un producto de un nucleótido por toda la cadena, generando así sumas sucesivas, al igual que pasa con la multiplicación decimal. Resaltar que en la operación anterior se han tenido en cuenta los acarreo en las sub-sumas, aunque no

se han representado, la causa es la no sobre-carga gráfica de información en la representación del producto anterior. Además se explicó en la definición de sumas, que debido a la suma de dos nucleótidos que provocan desbordamiento o acarreo, se suma la cantidad de acarreo al nucleótido siguiente.

Por último falta definir la operación aritmética de la división. La división se realiza igual (tiene el mismo mecanismo) que en el caso decimal, pero a diferencia de la de decimal, la restas y multiplicaciones se realizan mediante la operaciones de resta y multiplicación respectivamente, que se han definido en este documento. Por ejemplo, supongamos los siguiente números en *Cod-InChroSil*; dividendo es AGAACGA (en decimal es 1060), mientras que el divisor es TGA (en decimal es 52). Tras realizar la operación el cociente de la división es 20 en decimal (GGA en *Cod-InChroSil*) y, el resto, también es 20 en decimal (GGA *Cod-InChroSil*), de esta forma, $AGAACGA = TGA \cdot GCA + GCA$ (dividendo = divisor x cociente + resto).

Finalmente indicar, que con este sistema de numeración, *Cod-InChroSil*, se pueden realizar todo tipo de operaciones al igual que con los existentes sistemas (binario, octal, hexadecimal, etc.). Esta característica unida a la gran posibilidad de migración a otros sistemas numéricos, permite trabajar con la información de las hebras de ADN, con mayor facilidad y, realizar con ellas operaciones que anteriormente eran impensables. Por ese motivo, el sistema de numeración *Cod-InChroSil*, además de servir como apoyo a toda la tecnología relacionada a *InChroSil* y a esta invención, se presenta como una alternativa para la manipulación y utilización de la información contenida en las hebras de ADN.

Por otra parte, decir que una representación en coma flotante es un método de representación de números reales, donde la peculiaridad es la flotación o el movimiento de la

coma, elemento que separa la parte entera de la parte decimal (fraccionaria) de un número real. En oposición existe el sistema de representación en coma fija, donde de antemano se establece el número de dígitos que pertenecen a la parte decimal y cual a la parte entera. Estos sistemas de representación en coma flotante permiten a los sistemas de numeración representar cantidades numéricas grandes. Esta es la causa por la que se ha utilizado ampliamente este formato de coma flotante para la representación de números grandes. Un claro ejemplo, se encuentra en la área de la informática, que ha utilizado este tipo de representación, para el almacenamiento de representaciones numéricas muy grandes, que luego podían ser manipuladas matemáticamente. Por otra parte, todo número en coma flotante sigue un patrón establecido, el cual se puede resumir en la siguiente ecuación: $r = m \cdot b^e$, donde r es el número real, m es la mantisa, b es la base y finalmente e es el exponente, que permite elevar la base. Por ejemplo, el número 0.000123_{10} en coma fija (utilizamos el sistema decimal), en coma flotante podría tener las siguientes representaciones: $1.23 \cdot 10^{-4}$, $0.123 \cdot 10^{-3}$, $0.0123 \cdot 10^{-2}$, $0.00123 \cdot 10^{-1}$, etc. Como se observa en las anteriores representaciones numéricas, todas hacen referencia al mismo número pero con diferente representación en coma flotante. Por ese motivo y para evitar las ambigüedades en las representaciones de cantidades numéricas, se estableció un estándar que permite la representación homogénea de los números en coma flotante. Este estándar fue presentado al IEEE en el siglo pasado, más concretamente en la década de los ochenta, con el nombre de estándar IEEE 745. Este estándar permite definir un amplio rango de formatos de representación de cantidades numéricas, como son el cero y valores des-normalizados (el infinito y NAN). Por otra parte, también se definen un conjunto de operaciones que se pueden hacer en este formato, denominada *Aritmética de Coma*

Flotante. En este estándar, el número de bits que se puede utilizar viene relacionado con la precisión que se puede obtener con la representación de los números, es decir, si utilizan más bits la precisión y el rango de representación es mayor. Esto permite diferenciar dos grandes grupos, aunque existen en la literatura extensiones de ambos. El primero de ellos es el de simple precisión, el cual utiliza 32 bits para su representación. Estos 32 bits están repartidos en tres secciones que identifican al número. Por otra parte, tenemos otro formato del estándar, que presenta mayor capacidad en sus representación y, que está compuesto por 64 bits, el doble, de ahí su nombre doble precisión. Al igual que su hermano pequeño, los bits de este formato están repartidos entre 3 secciones o campos, que caracterizan e identifican al número. En las siguientes líneas explicaremos el sistema en coma flotante para el sistema de numeración Cod-InChroSil. Como se comento anteriormente se está tratando con un sistema de numeración que permite representar cantidad numéricas con sólo cuatro símbolos, los nucleótidos que componen una hebra de ADN. Pero al igual que los sistemas de numeración actuales, se necesitan muchos símbolos del sistema para representar cantidades numéricas grandes. Como se ha observado anteriormente la flexibilidad que posee los formatos en coma flotante es muy importante, por ese motivo, se ha considerado establecer un formato en coma flotante que permita representar números muy grandes mediante la codificación en Cod-InChroSil. Si se utiliza esta codificación en los sistemas InChroSil encontramos que por una parte se tiene la posibilidad de representar números muy grandes y, por otra parte, se aumenta la información almacenada del número, debido a que cuando se representa un número en este formato, también se está almacenando el número complementario u opuesto a este. Esto permite simplificar en gran medida las operaciones matemáticas que

-28-

necesiten ambos números. El siguiente paso es ¿Cómo establecer el formato para poder utilizar esta característica que ofrece los sistemas InChroSil y, en definitiva el ADN?, la contestación la encontramos en las siguientes líneas, donde se explica como se ha definido un formato de representación en coma flotante para números codificados en Cod-InChroSil. La invención puede definir dos grandes precisiones para las representaciones en coma flotante; por una parte está el formato de simple precisión, el cual se ha definido que posea 20 nucleótidos en su representación, haciendo un pequeño inciso en esta parte del documento, como bien ha percibido el lector en las líneas anteriores no se está hablando para cuantificar la longitud de la representación, de bits, sino, de nucleótidos. Continuando con la explicación del formato, se han utilizado 20 nucleótidos para establecer el formato de simple precisión. En este formato los 20 nucleótidos se reparten o distribuyen en 6 secciones o campos, que se describen a continuación; dos secciones de las seis están dedicadas a establecer cual de las dos hebras se utiliza para representar o definir el exponente y la mantisa respectivamente. Sendas secciones poseen una longitud de un nucleótido, información suficiente para poder decidir qué parte se debe elegir. Por ejemplo, si encontramos una representación donde en una sección anterior al exponente, aparece el nucleótido A, indica que el exponente que se necesita está en la hebra superior, por el contrario, si se encuentra un nucleótido T, indica que el exponente viene representado por la hebra inferior. Con este juego de posiciones se pueden realizar diversas combinaciones en la representación de un mismo número y simplificar en gran medida posibles operaciones matemáticas. Dentro del formato de representación en coma flotante existen otras dos secciones que tienen la función de especificar el tipo de codificación que están representados el exponente y la

mantisa. Se puede considerar para este formato cuatro tipos de codificación, que son los que se presentan en la siguiente enumeración.

- 5 • Signo Magnitud (toma el valor de A en la sección): El formato de representación de enteros con signo que se denomina Signo Magnitud, la cual imita nuestra escritura manual de enteros, por una parte el signo y luego la magnitud del numero. Por ejemplo, el número
- 10 34₁₀ el cual en Cod-InChroSil es GAAC₄, si se considera que el signo negativo se codifica como C y, por el contrario el signo positivo por A, por lo tanto el número de nuestro ejemplo es el siguiente CGAAC.
- 15 • Complemento a uno (toma el valor de G en la sección): Se define el complemento a uno con la siguiente ecuación $N_b^{Ca1} = b^n - N_b$, donde N_b^{Ca1} es el numero en complemento a uno del numero N_b . Sea $N_4 = CCTAAAG$ entonces su complemento a uno es $N_4^{Ca1} = 4^7 - CCTAAAG =$
- 20 GAAAAAAA - CCTAAAG = GGATTTC.
- Complemento a dos (toma el valor de C en la sección): Una vez calculado el complemento a 1, es fácil obtener el complemento a 2, por lo tanto el cálculo del
- 25 complemento a dos sería mediante la siguiente ecuación $N_b^{Ca2} = b^n - N_b + 1$, por lo tanto siguiendo el ejemplo anterior, el numero en complemento a dos es el siguiente $N_4^{Ca2} = 4^7 - N_4 + G = GAAAAAAA - CCTAAAG + G =$
- 30 GGATTTT.
- Exceso N (toma el valor de T): El exceso n se formula de la siguiente forma $N^{Exceso}_4 = N_4 + (N-1)$.

En definitiva, estas secciones especifican la

35 codificación en la que está el elemento (exponente o

-30-

mantisa respectivamente), es decir, si el anterior nucleótido al exponente o la mantisa posee por ejemplo el valor de G, esto indica que el exponente o mantisa está codificado en complemento A1. Finalmente, indicar que la longitud de estas secciones es de un nucleótido, al igual que como las secciones que determinaban la hebra a utilizar. Para poner punto final a la explicación de las secciones que conforman el formato de coma flotante de simple precisión, se refiere la sección de signo con una longitud de uno nucleótido y, que permite saber el signo del numero que se representa, es decir, si en esta sección se tiene el nucleótido A, quiere decir que el numero es positivo, por el contrario si se tiene el nucleótido T el numero es negativo. Por otra parte, está la sección del exponente, la cual posee una longitud de 4 nucleótidos. Finalmente esta la sección de la mantisa que con una longitud de 11 nucleótidos se completa los 20 nucleótidos que forman el formato de representación. En general y de forma gráfica en la figura 10 se muestran las diferentes secciones en que se divide el formato de coma flotante en simple precisión con 20 nucleótidos.

Con el anterior formato se puede conseguir exponentes con los siguientes rangos de valores, que se presentan en la siguiente enumeración.

- Coma fija con signo, Sea x el exponente entonces $x \in [-127,0] \cup [0,+127]$
- Complemento A1, Sea x el exponente entonces $x \in [-127,0] \cup [0,+127]$
- Complemento A2, Sea x el exponente entonces $x \in [-128,0] \cup [0,+128)$
- Exceso 4^4 , Sea x el exponente entonces $x \in [-128,0] \cup [0,+128)$

-31-

En la figura 11 se puede observar las diferentes codificaciones que puede adquirir el exponente. También se pueden observar las diferentes relaciones del mismo número en diferentes sistemas de codificación, además de la característica de la doble hebra que puede ser utilizada según las operaciones o el propósito convengan.

Por otra parte, se tiene que la mantisa que esta compuesta por 11 nucleótidos, de este modo los rangos que se pueden obtener con las diferentes codificaciones son los siguientes:

- Coma fija sin signo, Sea m la mantisa entonces $m \in [-4^{11}-1, 0] \cup [0, 4^{11}-1]$
- Complemento A1, Sea m la mantisa entonces $m \in [-4^{11}-1, 0] \cup [0, 4^{11}-1]$
- Complemento A2, Sea m la mantisa entonces $m \in [-4^{11}, 0] \cup [0, 4^{11})$
- Exceso 4^{11} , Sea m la mantisa entonces $m \in [-4^{11}, 0] \cup [0, 4^{11})$

De forma global, con este formato de representación en coma flotante con simple precisión se pueden conseguir números que estarían comprendidos en el siguiente intervalo numérico $[-17592186044416 \cdot 10^{127}, 0] \cup [0, 17592186044416 \cdot 10^{127}]$. Finalizaremos la explicación del formato de simple precisión con un ejemplo que ya ha sido utilizado para el formato IEEE 745. Recordando que el número era -118.625_{10} y su codificación en IEEE 745 de simple precisión es 11000010111011010100000000000000 0, un número binario de 32 bits. Por tanto utilizando nuestro sistema de coma flotante en simple precisión de la invención se tiene la siguiente codificación.

Hebra 1: T AT CAGG AA AGCCCAAAAAA

Hebra 2: A TA GTCC TT TCGGGIT T TTT

Como se puede observar el numero es negativo, por ese motivo se a codificado el primer nucleótido como T, los siguiente nucleótidos indican que el exponente esta representado en la hebra 1 y su codificación es exceso 128. Con respecto a la mantisa esta representada con coma fija sin signo y se corresponde con la hebra 1.

El siguiente formato de representación en coma flotante es la doble precisión, la cual utilizaba 40 nucleótidos o dos hebras de 20 nucleótidos. El formato de representación en doble precisión es el mismo que el de simple, bajo la salvedad que se aumenta el numero de nucleótidos en las secciones de exponente y mantisa. De forma gráfica en la figura 12 se muestra el formato de representación en doble precisión para 40 nucleótidos.

El rango de valores que puede admitir el exponente vendría definido por la siguiente enumeración.

- Coma fija con signo, Sea x el exponente entonces $x \in [-4^7-1,0] \cup [0,4^7-1]$
- Complemento A1, Sea x el exponente entonces $x \in [-4^7-1,0] \cup [0, 4^7-1]$
- Complemento A2, Sea x el exponente entonces $x \in [-4^7,0] \cup [0, 4^7)$
- Exceso 4^7 , Sea x el exponente entonces $x \in [-4^7,0] \cup [0, 4^7)$

Para el caso de la mantisa el rango de valores que se podrían representar en este formato de coma flotante de doble precisión son los que se muestran en la siguiente enumeración.

- Coma fija sin signo, Sea m la mantisa entonces $m \in [-4^{28}-1,0] \cup [0, 4^{28}-1]$

-33-

- Complemento A1, Sea m la mantisa entonces $m \in [-4^{28}-1, 0] \cup [0, 4^{28}-1]$
- Complemento A2, Sea m la mantisa entonces $m \in [-4^{28}, 0] \cup [0, 4^{28})$
- 5 • Exceso 4^{28} , Sea m la mantisa entonces $m \in [-4^{28}, 0] \cup [0, 4^{28})$

Realizando diferentes combinaciones entre el exponente y la mantisa se pueden obtener números con este formato de coma flotante de doble precisión que estarían comprendidos dentro de este intervalo numérico $\in [-72057594037927936 \cdot 10^{65536}, 0] \cup [0, 72057594037927936 \cdot 10^{65536}]$.

La unidad de almacenamiento 1 descrita, puede ser de dos tipos que la invención ha denominado *Mem-InChroSil* (Memory INorganic CHROsome based in SILicon). La invención puede utilizar memorias volátiles y no volátiles. La memoria volátil implementada, se ha denominado como *InChroSil Access Memory (IAM)*. Este tipo de memorias, además de tener la característica de ser volátil, tiene la propiedad de acceso estructurado, de este modo accedemos a un nucleótido, mediante la previa localización de la hebra, para posterior designar el nucleótido. La memoria volátil de la invención permite la escritura y lectura de la información almacenada en ella. Esta memoria es volátil, es decir, si se desconecta de la energía eléctrica, la información se pierde. Por este motivo, este tipo de memoria temporal se suele utilizar para almacenar resultados intermedios o, datos que no son permanentes. El acceso diseñado, para este tipo de memorias es estructurado, es decir, si se quieren obtener datos de esta memoria, la funcionalidad es la siguiente; se elige una cadena de ADN inorgánico (o también denominado *InChroSil*) de la memoria. Una vez determinada la cadena de ADN inorgánico se elige que hebra, que contiene la información que buscamos, indicar que las cadenas de ADN inorgánico que

se ha desarrollado para las memorias, están compuestas de dos hebras, al igual que sus homólogas orgánicas. Con estas dos premisas localizadas, el siguiente punto es determinar que nucleótido o grupo de nucleótidos se quiere acceder, aunque también se puede dar el caso, que el usuario quiera acceder a la vez, a parte o a toda la cadena de ADN inorgánico, el cual previamente se había localizado. Con este sistema tridimensional, que hemos implementado para un sistema de memoria volátil, se ha utilizado la tecnología básica explicada en este documento (InChroSil), ya que es fácilmente integrable en la tecnología existente, para la fabricación de memorias volátiles. Por otra parte, esta característica de estructuración hace que el sistema de memoria de la invención sea innovador y novedoso, ya que se ha considerado una dimensión más, la hebra, ya sea esta principal o la complementaria, ya que en los anteriores sistemas de memoria volátil, sólo se considera dos dimensiones; las filas y las columnas. Otra característica de la invención es que se puede considerar una cadena de ADN inorgánico como una fila o como una columna, es decir, el sistema que administra la memoria (IAM), puede estipular como estarían almacenadas las cadenas de ADN en la memoria, ya sean por filas o por columnas. Este sistema de memoria (IAM) se diferencia por el carácter genético que tiene, debido a que mediante materiales inorgánicos se pueden almacenar información del ADN orgánico. Por lo tanto, las memorias volátiles o RAM existentes, además de ser distintas por el tipo de acceso, también lo son por la unidad básica de almacenamiento, los bits o grupos de bits para el caso de las RAM's y, los nucleótidos o grupos de nucleótidos para el caso de nuestra memoria IAM. Estas memorias pueden integrarse en los sistemas o dispositivos actuales que demandan memoria, por la simple razón que se pueden fabricar con semiconductores e integrarse en un circuito mediante micro-tecnología o nano-tecnología.

Debido a esto estas memorias también se pueden agrupar en módulos 21 con diferentes conectores. Estos conectores varían respecto a los dispositivos o placas electrónicas donde estuvieran conectadas las memorias. De esta forma se
5 creara con esta estructura una memoria genética inorgánica basada en la estructura que posee la misma naturaleza, por esta razón, el acceso de memoria es mucho más interesante acceder a hebras completas que a un par de nucleótidos concretos, debido esencialmente por la forma de unión,
10 separación, etc, de la propia naturaleza del ADN orgánico. Como se puede observar en la figura 15, el sistema de memoria está formado por cadenas de ADN inorgánico 12, que su vez están formadas por dos hebras, una principal y otra complementaria. Estas hebras de la memoria se dividen en
15 nucleótidos, que representan las unidades básicas 1 de almacenamiento de estas memorias. Así de este modo, cuando queramos localizar un nucleótido o grupo de nucleótidos, primero debemos de localizar la cadena de ADN inorgánico, para luego determinar en qué hebra esta la información,
20 para más tarde establecer que nucleótido o nucleótidos dentro de la hebra queremos acceder. En la figura 13 se muestra un esquema de una de estas unidades InChroSil, la cual contiene dos nucleótidos, que son complementarios. Finalmente indicar, que debido a sus características
25 genéticas se podrían utilizar estas memorias (IAM), para el cálculo de operaciones intermedias o finales, pero siempre de carácter temporal, como es el caso del cálculo del trayectorias o resolución de grafos ponderados (camino hamiltonianos, disjktra, euler, etc.), aunque podría ser
30 ampliado a cualquier tipo de información que necesite de información mediante codificación genética, es decir, mediante la representación de cadenas de ADN. Una segunda memoria se ha implementado para la invención son las memorias no volátiles, la cuales se denominan en la
35 invención InChroSil Read Only Memory (IROM). Este tipo de

memorias no volátiles, la información que contiene es de sólo lectura y al igual que las memorias IAM, descritas anteriormente, su unidad básica de almacenamiento son los nucleótidos, por lo tanto, su acceso es estructurado como en las IAM.

Las memorias no volátiles tiene la peculiaridad de que son memorias que sólo son de lectura. Las memorias IROM o memorias de sólo lectura con InChroSil al igual que las memorias IAM explicadas anteriormente, poseen la misma estructuración que en la figura 15 y, también la misma estructura de acceso explicada antes para las memorias IAM, factor que la diferencia también de las memorias ROM o no volátiles existentes en la actualidad. Pero esta memoria tiene la propiedad de que su información es sólo de lectura, por lo tanto se puede implementar mediante fusibles u otro sistema que nos permitiera crear la información estática, de tal modo, que cuando se quemará o creará un nucleótido en este sistemas, el método inverso, que se ha descrito en la sección anterior, permite quemar o crear el nucleótido complementario, de este modo, seguiríamos un funcionamiento indicado en la figura 14. En esta figura la referencia 5 representa la información primaria, la referencia 6 la inversión de dicha información primaria, la referencia 7 la información secundaria que se obtiene y la referencia 8 los controles aplicados para conseguir el correcto funcionamiento.

Finalmente indicar, que este tipo de memorias tienen una gran utilidad, debido a que pueden almacenar información que quisiéramos que se mantuviera estática, como por ejemplo, la secuenciación de nuestro ADN orgánico, es decir, existen investigaciones donde intentan guardar el ADN orgánico en algún tubo o dispositivo, pero al ser un material orgánico es difícil de conservar, con la memoria no volátil de la invención se puede almacenar la información de forma permanente e incluso ser accedida

desde un ordenador o sistema informático. Otras aplicaciones, podría ser el conformar base de datos de ADN mediante memoria IROM, existen base de datos que almacenan de forma convencional el ADN, mediante software, la memoria
5 de la invención permite el almacenamiento directo de esta información genética. En definitiva, la utilidad o funcionalidad de esta memoria IROM, puede ser muy variada y para diversos fines, dependiendo de la utilidad en que se desee emplear.

10 La configuración descrita permite efectuar la resolución de problemas de rutas, flujos y redes abordándolos desde la perspectiva de la teoría de grafos.

Los problemas de encaminamiento, suelen ser resueltos mediante algoritmos deterministas (tiempo razonable) y,
15 utilizando software, en vez de hardware. Esto hace que los sistemas opten por dispositivos que realizan modelos predictivos, no reales, sobre una posible solución aproximada, es decir, no existe en el mercado, ningún dispositivo que pueda decirle a un distribuidor de bebidas,
20 por ejemplo, cual es el camino por el cual pasaría una única vez por cada bar o restaurante de una ciudad y, de forma óptima. Este problema es irresoluble con los dispositivos que en la actualidad existen, software de rutas, encaminamiento, etc. En cambio, si se pudiera
25 resolver, para la compañía de bebidas y el propio distribuidor, serían unas ganancias y ahorro de tiempo respectivamente. Otro ejemplo, son la detección de averías o anomalías, por parte de las compañías de gas, electricidad, agua y teléfono. Existen en el mercado
30 dispositivos de posicionamiento, mediante PLC, sensores, GIS, GPS, etc. Pero a nivel de localización y encaminamiento, se realiza mediante software, en algunos casos y, en otros es totalmente manual, donde el operario mediante una referencia posicional de la área geográfica,
35 tiene que encontrar la posible avería. Por ese motivo, la

utilización de un dispositivo en tiempo real, que determine la posición y, calcule el camino más corto posible hasta llegar al punto de averías, es crucial para los intereses de la compañía, ya que en menos tiempo, puede resolver el problema. También en la organización industrial, donde se podría optimizar, a nivel de producción, la maquinaria de unos procesos y máquinas industriales, ya que lo podríamos representar como en un grafo la información y distribución de la maquinaria de la planta industrial. En definitiva, todo sistema que pueda ser reducido o esquematizado a una representación en forma de grafo, pueden aplicarse técnicas y metodológicas para resolver problemas en este campo, además si se considera que con la codificación del grafo en ADN y con la utilización de la tecnología InChroSil, se pueden resolver de forma óptima, problemas que anteriormente no podían ser resueltos, además de poseer de un dispositivo que puede ser integrado en los sistemas actuales y, su reutilización es total, como se ha comentado en secciones anteriores. A continuación se tratan tres problemas que son la piedra angular en el cálculo de muchos de los problemas anteriormente mencionados.

- Encontrar la ruta más corta ó más óptima entre dos puntos.
- Resolver problemas de los llamados del viajante bajo.
- La obtención de caminos eulerianos.

Encontrar la ruta más corta ó más óptima entre dos puntos.

30

Un problema típico de teoría de grafo, es encontrar la ruta más corta u óptima desde un punto inicial hasta un punto final. El planteamiento que se presenta a este problema es considerar un grafo dirigido, ponderado (positivo o negativo) y, utilizar las características del

35

ADN, mediante tecnología InChroSil. De este modo, mediante un dispositivo basado en tecnología InChroSil, se puede encontrar el camino más corto u óptimo. Pongamos un ejemplo, supongamos que se desea resolver el problema de la figura 16 mediante las listas de adyacencias y la teoría de conjunto.

En el grafo de la figura 16 se puede representar como una lista de adyacencias, donde la nomenclatura sería la siguiente; (vértice origen, arista, vértice destino), por lo tanto, el grafo quedaría de la siguiente forma.

Nodo(s): $\{(s, x1, u), (s, x2, v)\}$
 Nodo(u): $\{(u, x3, w), (u, x7, v)\}$
 Nodo(v): $\{(v, x5, t)\}$
 Nodo(w): $\{(w, x4, v), (w, x6, t)\}$

Cabe resaltar que, en la teoría de grafos se define la estructura de datos, lista de adyacencia, como una estructura que permite asociar a cada vértice i del grafo, una lista que contenga todos aquellos vértices j , que sean adyacentes a él. De este modo, se ahorra en espacio en su representación y, además el grafo, se puede representar por medio de un vector de n componentes (si $|V|=n$), donde cada componente va a ser una lista de adyacencia correspondiente a cada uno de los vértices del grafo. Además, cada elemento de la lista consta de un campo indicando el vértice adyacente. Por otra parte, si el grafo fuera etiquetado, habrá que añadir un segundo campo para mostrar el valor de la etiqueta.

Utilizando la teoría de conjuntos, podríamos formar un conjunto con todos los elementos de las listas de adyacencia, es decir, los elementos (vértice, arista, vértice). Este conjunto que hemos formado tendría unas premisas que se enumeran a continuación.

- Si existen dos elementos pertenecientes al conjunto, tales como e_1 y e_2 , se observa que el componente final de e_1 es igual al componente inicial de e_2 , ambos elementos reaccionan, creando un nuevo elemento, el cual pertenecerá al conjunto.
- Si existen elementos que pertenecen al conjunto, que contienen componentes que son iniciales y finales, no reaccionan con el resto, debido a que serían ya solución.
- Dos elementos que reaccionen tienen una relación positiva o 1, en cambio dos elementos que no reaccionen tienen una relación negativa o 0.
- Los elementos con componentes iniciales y que pertenecen a las listas de adyacencia, una vez reaccionados no vuelve a reaccionar.

Si se sabe que E es el conjunto de elementos y, C es el conjunto de componentes, podemos decir que la reacción es la siguiente operación:

$$\forall x \in C \mid \exists yax, xbz \in E \text{ tal que } yax \wedge xbz \equiv yaxbz$$

siendo $y, a, x, b \in C$ y $yaxbz \in E$.

En este caso, tenemos un conjunto de elementos pertenecientes a las listas de adyacencias, los cuales los denotaremos como Elemento[i], por lo tanto, el conjunto inicialmente quedaría de la siguiente forma.

Elemento1: $\{(s, x1, u)\}$
 Elemento2: $\{(s, x2, v)\}$
 Elemento3: $\{(u, x3, w)\}$
 Elemento4: $\{(u, x7, v)\}$
 Elemento5: $\{(w, x4, v)\}$
 Elemento6: $\{(w, x6, t)\}$
 Elemento7: $\{(v, x5, t)\}$

Con este conjunto inicial, se puede analizar sus elementos y ver cómo podrían cumplir las premisas asociadas al conjunto y, crear nuevos elementos. En este caso, se ve que Elemento1 reaccionaría con Elemento3 y Elemento4, formando Elemento8 y Elemento9 respectivamente. También se ve que Elemento2 reacciona con Elemento7 para formar Elemento10, el cual es ya solución. Después de esto el conjunto quedaría de la siguiente forma.

10

Elemento1: $\{(s, x1, u)\}$ [no reacciona] premisa 4

Elemento2: $\{(s, x2, v)\}$ [no reacciona] premisa 4

Elemento3: $\{(u, x3, w)\}$

Elemento4: $\{(u, x7, v)\}$

15

Elemento5: $\{(w, x4, v)\}$

Elemento6: $\{(w, x6, t)\}$

Elemento7: $\{(v, x5, t)\}$

Elemento8: $\{(s, x1, u, x3, w)\}$

Elemento9: $\{(s, x1, u, x7, v)\}$

20

Elemento10: $\{(s, x2, v, x5, t)\}$

En una siguiente ronda, se observa que Elemento8 puede reaccionar con Elemento5 formando Elemento11 y, que también Elemento8 puede reaccionar con Elemento6 para formar Elemento12, finalmente, Elemento9 reacciona con Elemento7 formando Elemento13. En esta ronda el conjunto quedaría.

25

Elemento1: $\{(s, x1, u)\}$ [no reacciona] premisa 4

Elemento2: $\{(s, x2, v)\}$ [no reacciona] premisa 4

30

Elemento3: $\{(u, x3, w)\}$

Elemento4: $\{(u, x7, v)\}$

Elemento5: $\{(w, x4, v)\}$

Elemento6: $\{(w, x6, t)\}$

Elemento7: $\{(v, x5, t)\}$

35

Elemento8: $\{(s, x1, u, x3, w)\}$

-42-

Elemento9: $\{(s, x1, u, x7, v)\}$
 Elemento10: $\{(s, x2, v, x5, t)\}$
 Elemento11: $\{(s, x1, u, x3, w, x4, v)\}$
 Elemento12: $\{(s, x1, u, x3, w, x6, t)\}$
 5 Elemento13: $\{(s, x1, u, x7, v, x5, t)\}$

Observando el conjunto se ve que existe una única
 reacción, la del Elemento11 que reacciona con Elemento7
 para formar Elemento14. Con esta última reacción, el
 10 conjunto quedaría de la siguiente forma.

Elemento1: $\{(s, x1, u)\}$ [no reacciona] premisa 4
 Elemento2: $\{(s, x2, v)\}$ [no reacciona] premisa 4
 Elemento3: $\{(u, x3, w)\}$
 15 Elemento4: $\{(u, x7, v)\}$
 Elemento5: $\{(w, x4, v)\}$
 Elemento6: $\{(w, x6, t)\}$
 Elemento7: $\{(v, x5, t)\}$
 Elemento8: $\{(s, x1, u, x3, w)\}$
 20 Elemento9: $\{(s, x1, u, x7, v)\}$
 Elemento10: $\{(s, x2, v, x5, t)\}$
 Elemento11: $\{(s, x1, u, x3, w, x4, v)\}$
 Elemento12: $\{(s, x1, u, x3, w, x6, t)\}$
 Elemento13: $\{(s, x1, u, x7, v, x5, t)\}$
 25 Elemento14: $\{(s, x1, u, x3, w, x4, v, x5, t)\}$

Se puede ver que los resultados son Elemento10,
 Elemento12, Elemento13 y finalmente Elemento14, debido a
 que poseen componentes iniciales y finales. El
 30 funcionamiento del planteamiento anterior, para encontrar
 las soluciones o caminos, se realiza de forma secuencial.
 Pero en el circuito de la invención, se realiza de forma
 paralela, es decir, la invención ha considerado dos
 conjuntos; el primer conjunto C_1 está formado por los
 35 elementos que contienen componentes iniciales y los

elementos que no contienen componentes finales. El segundo conjunto C_2 está formado por los elementos que contienen componentes finales y los elementos que no contienen componentes iniciales. Ambos conjuntos son codificados mediante *Cod-InChroSil* y, almacenados en sendas hebras 10 respectivamente. Estas hebras unen sus componentes mediante una circuitería que contiene comparadores 9 formando una matriz como se observa en la figura 17 de esta forma reaccionan de forma paralela, obteniendo un conjunto de reacciones codificadas como positivas (1) o negativas (0). Estas reacciones forman una matriz de 0's y 1's. Por lo tanto, existen dos conjuntos C_1 y C_2 , que representan a los dos anteriores conjuntos mencionados y los cuales están codificados en hebras de *InChroSil*. Por otra parte, tenemos un conjunto C_r que esta formado por las reacciones entre los elementos de los conjunto C_1 y C_2 , de este modo se redefine la reacción como la operación donde si $e_2 \in C_2$ y $e_1 \in C_1$, entonces $e_1 \mathfrak{R} e_2 \in C_r$, siendo \mathfrak{R} la reacción y los valores que puede tomar son 0 o 1. Como se observa, en la reacción ya no se crea un nuevo elemento que deberá de reaccionar, sino que reaccionan todos a la vez y se indica en una matriz, con un 1 si ha habido reacción o la reacción es positiva y, por el contrario con un cero si no habido reacción o la reacción ha sido negativa.

Una vez obtenida la matriz con las reacciones de todos los elementos con todos, se pasa esta información a un buffer y, mediante software o circuitería se le aplica algoritmos de recuperación de caminos, los cuales no tienen un coste muy elevado y de orden polinomial. Si se consideran grafos ponderados, los pesos estarían almacenados en una memoria (*Mem-InChroSil* u otro tipo de memoria), por lo tanto, las aristas del grafo anterior contienen la dirección y no el valor del peso, el motivo es por que se pueden utilizar cantidades grandes para los pesos y las direcciones tienen una longitud determinada.

Una vez encontrados las aristas del camino, sólo hay que sumar los costes de cada arista, para obtener el coste total del camino. Luego con los costes totales, se pueden ordenar los caminos según un criterio de ordenación, todo esto se puede realizar mediante circuitería o módulos software, que se conectarían con el dispositivo. Con este circuito se resuelven los problemas planteados en los algoritmos de Dijkstra, Floyd-Warshall y Bellman-Ford, Ford-Fulkenson, al poder considerar costes negativos, etc.

La ventaja que presentan estos sistemas es la fácil incorporación a los sistemas hardware existentes, ya que ambos se fundamentan en la electrónica y los circuitos integrados. Por otra parte, resulta una alternativa hardware para los sistemas de cálculo que existen, debido a que con los sistemas software, necesitan de otro entorno (sistema operativo, máquinas virtuales, etc.) para funcionar y, en cambio este dispositivo se conectaría directamente a la circuitería del dispositivo anfitrión.

20 Resolver el problema del camino Hamiltoniano.

Los sistemas de computación actuales se basan en una tecnología secuencial establecida por *John Von Neumann* a mediados del siglo pasado. Estos computadores secuenciales son buenos en la solución de problemas matemáticos, pero tienen dificultades en problemas denominados *de llave en mano*, donde se tiene que buscar todas las posibles soluciones hasta encontrar la solución adecuada. Esto hace que el tiempo de computación de los problemas pase a orden exponencial o logarítmico.

Se han intentado paliar estos problemas con la construcción de computadores paralelos que puedan reducir el tiempo de computación. Estas soluciones consumen muchos recursos en comunicaciones y numero de nodos conectados entre sí. Por ese motivo ha mediado de 1994, Adleman presentó una nueva alternativa para programar con ADN

orgánico, este sistema orgánico presento un masivo paralelismo en la realización de las operaciones y por tanto en la obtención de los resultados finales, pero con las limitaciones comentadas anteriormente. Diseñó un sistema para resolver el problema del camino Hamiltoniano con programación con ADN orgánico y, por lo tanto demostrar que se podían resolver problemas NP-Complejos en un tiempo computacional cercano al polinomial. A partir de este momento nació toda una nueva filosofía de programación con este material orgánico, la cual se denomino computación con ADN o programación molecular. Pero a lo largo de los años y debido a sus características del material, que principalmente es perecedero, los diferentes experimentos para traspasar la tecnología orgánica al mundo comercial fueron infructuosos. Para solucionar estos problemas, en este documento se presenta una solución con tecnología electrónica, en concreto con la tecnología InChroSil, una propuesta artificial que emula el comportamiento de los sistemas orgánicos.

Existen implementaciones paralelas para la solución del problema del camino Hamiltoniano pero necesitan un gran volumen de recursos y comunicaciones entre los nodos conectados. La ventaja que presenta el ADN inorgánico frente a las tecnologías secuenciales es el gran paralelismo en la realización de las operaciones, por ese motivo la invención resuelve de forma electrónica el problema del camino Hamiltoniano. Basándose en las características anteriormente expuestas, se codificó cada ciudad, los caminos y unas secuencias complementarias que servían para unir los caminos con hebras de ADN mediante la codificación expuesta en este documento véase figura 19 donde podemos ver el grafo 11 que representa el problema de Hamilton, donde los vértices están numerados del 0 al 6.

Este problema es de gran dificultad para problemas deterministas con orden polinomial, ya que es un problema

-46-

NP-Completo, pero se puede formular un algoritmo no determinista para la solución de este problema. El algoritmo no determinista se presenta en las siguientes líneas:

5

Entradas: Grafo, V_{in} y V_{out}

- Se generan de forma aleatoria todos los caminos existentes en el Grafo.
- 10 • Se eliminan todos los caminos que cumpla con las entradas.
- También se eliminan todos aquellos caminos que no posean los vértices requeridos.
- Para cada vértice, se rechazan los caminos que no
15 incluyan al propio vértice.

Salidas: SI (Existen caminos) NO (en caso contrario)

Por lo tanto en un ordenador convencional el tiempo de computación no puede ser polinomial, ya que se tiene que
20 comparar todas las ciudades una por una para encontrar el camino desde una ciudad inicial V_{in} hasta una ciudad final V_{out} .

El dispositivo electrónico que se explica a
25 continuación, se planteó a partir del grafo de 7 vértices 11 de la figura 19. En ella se representa un grafo dirigido que poseerá un número determinado de aristas. El primer paso es la codificación de grafo de la figura 19 para ello se utilizó la codificación Cod-InChroSil
30 (explicado en este documento) y se estableció que cada ciudad (vértice) tuviera un determinado número de nucleótidos, por ejemplo, 20 nucleótidos, y que las aristas estén compuestas por el mismo número, 20 nucleótidos, los cuales se corresponden con los 10 nucleótido últimos de un
35 vértice o ciudad origen y los 10 nucleótidos iniciales del

-47-

vértice o ciudad destino. Por lo tanto, utilizando la tecnología InChroSil se creará una hebra en la cual se albergarán todas las aristas representadas en dicho grafo. Por otro lado, la hebra debe de poder codificar todas las aristas de un grafo completo de 7 vértices 11, por lo tanto, para dar versatilidad al experimento, dicha hebra ofrecería la posibilidad, que en el peor de los casos comparar un grafo completo (un Grafo completo es en Teoría de grafos, en un grafo cuyo conjunto de aristas contiene todas las aristas posibles, $G=(V,VxV)$, siendo V el conjunto que contiene todos los vértices del grafo) y, en el mejor de los casos, por ejemplo, el grafo expuesto en la figura 19. Esto permite al programador del circuito electrónico que realice la codificación, elegir tantas aristas como desease dentro del rango establecido por el propio grafo, es decir, para el grafo del ejemplo, existirá una hebra de 5040 pares nucleótidos (factorial de 7 si se consideran 7 ciudades), de tal modo que, el programador solo exigirá la activación de la parte complementaria de la hebra en las aristas necesarias para el grafo a analizar. En este caso, solo se exigirá complementariedad con las partes de la hebra que correspondiesen a los estados que poseen arista del grafo de la figura 19. Por otra parte, a la codificación del grafo de la figura 19, se la denomina 'G' 17 (Grafo) y que puede ver en la figura 20, en la que A0, A1.....representan las aristas del grafo que une a los estados (Ei), y en la que E0, E1.....son los estados o las ciudades del problema planteado.

De forma gráfica se puede ver en la figura 21 como están conectados los estados mediante aristas y en la figura 22 se detalla esta codificación utilizando cod-inchrosil. En la figura 21 un estado está formado por 20 nucleótidos y una arista se compone de los 10 nucleótidos finales de un estado y los 10 nucleótidos iniciales del estado siguiente.

El siguiente paso, consiste en poder disponer de todos los posibles caminos hamiltonianos existentes en un grafo de 7 vértices completo, por lo que se creó un conjunto de 5040 cadenas de ADN, por supuesto con la tecnología InChroSil, estas hebras estarán compuestas por 140 pares de nucleótidos (puestos que existen 7 vértices y cada uno de ellos debe estar representado por 20 pares de nucleótidos, se exigiría una longitud de hebra de la multiplicación de 7 por 20). Estas hebras se dispusieron de forma matricial porque resultaba más cómodo para poder plantear un chip tridimensional, y seguir en la línea de poder emular la naturaleza orgánica. A esta codificación, de igual modo que con el grafo ejemplo, se denominó 'CG' 18 (Conjunto de Grafos).

La teoría de conjuntos se utiliza especialmente para delimitar el alcance de una proposición, lo que fuerza al objeto de la proposición a quedar en cierta medida concretizado. Esto es deseable pues permite operar con la proposición formada, al principio como mínimo a nivel de cierto o falso y en un nivel máximo, de acuerdo a como corresponda la parametrización de las proposiciones establecidas, es decir un conjunto S está definido si, dado un objeto cualquiera α , se sabe con seguridad si pertenece o no al conjunto. Por lo que, esta teoría aplicada a este proyecto, permite comparar cada arista de 'CG' de la figura 23 de un conjunto de hebras; que representan a todas las combinaciones posibles de los estados; con cada arista especificada en el grafo 'G' de la figura 20, de forma totalmente paralela, y determinando de este modo si dicha arista del grafo 'CG' pertenece o no al conjunto de aristas codificadas en el grafo 'G', por tanto las aristas del grafo 'G' se convierten en conjuntos y a su vez las aristas del grafo 'CG' se convierten en objetos de los cuales se debe saber con seguridad si pertenecían o no al conjunto expuesto por el programador, es decir, al grafo 'G', ya que

'CG' son plantillas de posibles soluciones. Para poder realizar estas comparaciones, se aprovechó la filosofía de los semisumadores lógicos, sumadores sin propagación de acarreo (CSA) y árboles de Wallace. Estas tecnologías
5 ampliamente reconocidas en el campo de la estructura de computadores, ofrecieron la posibilidad de crear modificaciones en los diseños para obtener el paralelismo exigido por el problema y de este modo obtener los resultados deseados. Por otra parte, como se observa en la
10 figura 24, se puede apreciar la estructura interna de un semisumador lógico estándar.

Una modificación que se presenta en este documento es la eliminación del acarreo quedando solo la suma directa de las dos entradas, si se tiene en consideración la tabla de
15 verdad de una puerta XOR, se puede apreciar que si los dos bits son iguales el resultado lógico es 0, por lo que en este proyecto se optó por negar dicha salida de este modo se trabajaría con lógica directa y no inversa. Debido a este planteamiento se desarrollo comparadores 12 mediante
20 puertas XNOR 12a para la solución del problema. Por otro lado, el hecho de juntar 40 puertas XNOR 12a (puesto que, las aristas son de 20 pares nucleótidos, esto supone 40 bits por cada arista y por tanto 80 entradas, ya que se comparará dos a dos), surgió de la idea de poder crear CSA
25 (sumador sin programación de acarreo, el cual permite realizar sumas obteniendo como resultado la suma de los datos y el acarreo), es decir la unión de puertas XNOR 12a dando un resultado único de 40 bits.

30 Pero estas puertas ofrecían un resultado de 40 bits, lo cual no era óptimo, por lo que se plantearon las siguientes preguntas ¿son todos los bits iguales? ¿Son la misma arista?, para ello se utilizo una puerta AND 12b de 40 entradas y 1 solo bit de respuesta, obteniendo en un
35 solo dato la contestación a nuestras preguntas. Ya que si

-50-

esta puerta AND 12b, da como resultado lógico un '1', significa que son iguales las aristas y, por tanto dicha arista perteneciente al grafo 'CG', y por lo tanto pertenecía al conjunto de aristas perteneciente al grafo 'G'. En caso contrario, si el valor es '0', significa que las aristas mencionadas no son iguales, pero no garantizan que no pertenecieran al conjunto programado, lo cual era un dato a resolver. Por tanto, se creó al igual que los CSA, un encapsulamiento del circuito, creando de este modo el comparador 12 mostrado en la figura 25.

Para poder resolver este problema, los comparadores 12 se establecen creando una matriz y comparando las aristas 13 y 14, dos a dos, de ambos grafos ('G' y 'CG') de forma paralela. Pero surge una nueva cuestión, ¿Cómo establecer si es camino o no?, para poder dar respuesta a esta pregunta se plantearon los ya conocidos árboles de Wallace, aunque no se utilizó su estructura si su filosofía de funcionamiento, la cual es crear diferentes niveles de exigencia para poder encontrar de forma totalmente paralela el resultado requerido.

Por lo tanto, se planteó que si se exigía que cada comparador en columnas de dicha matriz de comparadores, su resultado 12 de igualdad se dirigiera hacia una puerta OR 15, de tal modo, si cualquiera de los comparadores respondiese un '1' lógico, dicha puerta lógica 15 correspondiente a la columna, confirmaría de forma paralela la pertenencia de dicha arista al conjunto de las aristas programadas. De este modo se crean los mencionados niveles de los árboles de Wallace.

Por último, todas estas puertas lógicas OR 15, que en este caso son tantas como columnas o aristas existentes en las hebras de ADN InChroSil, pertenecientes al grafo 'CG', es decir, 6 puertas lógicas OR 15. De este modo si todas las aristas confirman la pertenencias al conjunto representan un camino Hamiltoniano, esto se consigue

dirigiendo todas las salidas hacia una puerta lógica AND 16, de esta forma se obtiene definitivamente la respuesta ansiada, se ha creado así el último nivel de exigencias.

Finalmente, el resultado de la puerta AND 16 se almacena en un vector de Resultados no representado, de tal modo que este vector tendrán 5040, es decir, tantas como hebras de ADN InChroSil, han sido codificadas, por lo que las posiciones en las que se vea reflejado un '1' lógico, reflejaría que la hebra codificada en dicha posición del vector es camino Hamiltoniano del grafo presentado, en la figura 26 en la que se muestra un esquema simplificado del circuito, y en la figura 27 podemos ver el circuito completo.

15 Resolver problemas de los llamados del viajante bajo.

El problema del viajante (traveling salesman problem TSP), es uno de los problemas más famosos y mejor estudiados en el campo de la optimización computacional. A pesar de la aparente sencillez de su planteamiento, el problema es uno de los más difíciles de resolver, debido a que es considerado de complejidad NP-Completo. La invención resuelve el problema del viajante con un sistema electrónico (sección anterior), con un coste temporal del orden polinomial. En este sistema se tiene un grafo ponderado y dirigido, donde se obtienen todos los caminos hamiltonianos existentes en el grafo. Por lo tanto, en el caso más favorable, sólo existía un camino, pero en el caso contrario o polo opuesto, en el caso desfavorable, es decir, cuando se tiene un grafo completo, podemos tener todos los caminos hamiltonianos. Como ya se comentó anteriormente, el cálculo de todos los caminos se realizan de forma paralela, pero lo que no contemplaba el circuito de la sección anterior era el peso de las aristas, para resolver los problemas de Hamilton ponderado. La idea que

se plantea en este documento para dar solución a los problemas de Hamilton ponderado, es sencilla. Utilizando la misma infraestructura electrónica prevista para el cálculo de los caminos hamiltonianos, pero además teniendo en cuenta dos implementaciones; la primera implementación en la figura 28 es transferir esta información (caminos encontrados) a un sistema software 23, 26 que determina la correspondencia entre peso-arista. Este sistema, una vez obtenido el coste individual, calcula el coste total de cada camino. Para finalmente, cuando estén calculados todos los costes de todos los caminos hamiltonianos, ordenarlos según un determinado criterio (de menor a mayor, de mayor a menor, etc.). Todas estas operaciones se tiene que tener en cuenta que la información esta codificada en *Cod-InChroSil* y, que la asignación de pesos a las aristas y posterior cálculo de costes, no es un trabajo costoso a nivel computacional, ya que tiene tiempos polinomiales. Por último, el usuario o un sistema tendrá una colección de caminos ordenados, y entonces dependerá de esté, para la elección del mejor camino. En esta figura 28, la referencia 21 representa el circuito de obtención del camino Hamiltoniano, la 22 el interface de conexión con interface 25 de un módulo de tratamiento de datos 23, la referencia 24 es una base de datos y la 26 un módulo de cálculo de costes y ordenación. La referencia 27 representa un dispositivo externo de demanda de datos.

Por otra parte en la base de datos de la invención están almacenados los datos intermedios, es decir los pesos de las aristas, etc. Por lo que sirve de apoyo a los cálculos que se van a realizar o realizados.

Otra implementación se muestra en la figura 29, es la utilización de memorias 28 (*Mem-InChroSil* u otro tipo de memoria), la idea es que cuando se codifiquen los vértices *InChroSil* y, por lo tanto se establezca el valor de las aristas que los unen, el sistema almacenara en esa

-53-

dirección de memoria (valor de la arista), el valor del peso de esa arista en el grafo, es decir, se almacena en la arista no su valor o peso, sino la dirección de memoria donde está su valor, debido a que este valor puede ser un número muy grande y las direcciones tienen un tamaño establecido. Cuando se determinen los caminos hamiltonianos, mediante el circuito 21 anterior, se busca el valor de las aristas en la memoria 28, ya que como sabemos su dirección, sabremos su valor. Estos costes se sumaran y buscaran por mediación de circuitería, es decir un bus 32 y sumadores 30, para obtener el coste total de cada camino que puede ser almacenado en la memoria 28 con direcciones consecutivas. Al igual que en la anterior implementación, el usuario o el sistema que recibe de entrada esta información elige con que camino se queda, mediante criterios de elección, ajenos a los dispositivos descritos en este documento. En este ejemplo la referencia 22 representa el interface de conexión con un interface 25a del módulo de tratamiento de datos 23a, la referencia 29 el módulo de ordenación, la 31 un interface de conexión del módulo 23a con un dispositivo externo de demanda de datos 27.

Finalmente, indicar que también se pueden resolver problemas de ciclos hamiltonianos, para lo que habría que considerar la premisa que el vértice de salida es igual al vértice destino.

La Obtención de caminos eulerianos

Los caminos eulerianos son aquellos caminos que recorren todos los vértices de un grafo pasando una sola vez por cada arista del grafo, siendo una de las condiciones fundamentales que regrese al vértice inicial o de salida, por lo tanto, un camino euleriano es un ciclo que contiene todas las aristas de un grafo una sola vez.

Este problema fue planteado y resuelto por el propio Leonhard Euler en 1736 en un problema que tiene el nombre de los siete puentes de la ciudad de Königsberg. El problema se enuncia de la siguiente forma: Dos islas en el río Pregel, en Königsberg se unen entre ellas y con la tierra firme mediante siete puentes. ¿Es posible dar un paseo empezando por una cualquiera de las cuatro partes de tierra firme, cruzando cada uno de los puentes una sola vez?.

10 Euler enfocó el problema representando cada parte de tierra por un punto y cada puente, por una línea, uniendo los puntos que se corresponden. Entonces, el problema anterior se puede trasladar a la siguiente pregunta: ¿se puede recorrer la representación sin repetir las líneas?,
15 euler demostró que no era posible puesto que el número de líneas que inciden en cada punto no es par (condición necesaria para entrar y salir de cada punto). Por lo tanto, este problema plantea preguntas como las siguientes; ¿Cómo se puede recorrer el cable de esta red eléctrica sin
20 repetir tramos de red?, ¿Cómo se puede realizar este recorrido, pasando por unas calles concretas?, etc.

Como se ha visto, existe una complejidad en la resolución de estos problemas, por ese motivo se podría plantear el problema de forma inversa, es decir, lo que
25 eran vértices se convierten en aristas y, las aristas se convierten en vértices. Con esta conversión, podemos aplicar o calcular los caminos hamiltonianos, puesto que ahora los vértices representan las aristas y, se quiere saber que secuencia de vértices debemos de seguir para
30 poder pasar por todas las aristas. Supongamos el grafo dirigido y no ponderado de la figura 18, el cual sería un multigrafo, es decir, un autómatas no determinista.

También se puede representar el grafo de la figura 18 de la siguiente forma, donde V_1 es el conjunto de vértices
35 y, E_1 es el conjunto de aristas.

$G_1 = (V_1, E_1)$ siendo $V_1 = (v_1, v_2, v_3)$ y $E_1 = (a_1, a_2, a_3, a_4, a_5, a_6)$

Además, tenemos las siguientes uniones de vértices,
5 por medio de aristas;

$$U = \{ (v_1, v_2, a_1), (v_1, v_2, a_2), (v_1, v_2, a_3), (v_2, v_3, a_4), (v_2, v_3, a_5), (v_2, v_3, a_6), (v_3, v_1, a_6) \}$$

10 Donde la nomenclatura utilizada para estas uniones es (vértice origen, vértice destino, arista). Si se realiza la inversión comentada y, se consideran las aristas como vértices y, los vértices como aristas, el grafo que se obtendría es el siguiente.

15

$$G_2 = (V_2, E_2) \text{ siendo } V_2 = (a_1, a_2, a_3, a_4, a_5, a_6) \text{ y } E_2 = ((v_1, v_2), (v_2, v_3), (v_3, v_1))$$

Además, se tienen las siguientes uniones de vértices, por
20 medio de aristas;

$$W = \{ (a_1, v_1, v_2, v_2, v_3, a_4) \dots \}$$

Por lo tanto el grafo tendrá la forma que se muestra
25 en la figura 18. Por consiguiente si se obtiene el camino hamiltoniano del anterior grafo G_2 , se puede afirmar que existe una solución o camino euleriano G_1 . También si se considera que el grafo es ponderado, se puede calcular el coste total de cada uno de los caminos y, por lo tanto
30 ordenarlos por coste, aunque los caminos eulerianos no poseen peso, podrían plantearse nuevos problemas de optimización en los caminos eulerianos. En conclusión, si existe camino hamiltoniano en el grafo G_2 , se puede afirmar que existe un camino euleriano en el grafo G_1 . Por último,
35 se puede afirmar que se pueden resolver ciclos eulerianos,

-56-

para lo cual, en la codificación de la base de hebras o bancos de hebras, se establecía que el nodo inicial fuese el mismo que el final. De este modo, mediante la invención se pueden resolver problemas de ciclos.

REIVINDICACIONES

1.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, cuya cadena comprende dos hebras, cada una de ellas constituidas por una cadena de nucleótidos, que comprenden Adenina (A), Guanina (G), Citosina (C) y Timina (T); de manera que entre dichas hebras se producen enlaces complementarios que se realizan mediante la unión de Adenina (A) con la Timina (T) y de la Guanina (G) con la Citosina (C); se caracteriza porque comprende medios de codificación binaria de los diferentes cuatro tipos de nucleótidos que componen las hebras, de manera que a los nucleótidos que forman enlaces complementarios se les asignan códigos complementarios mediante la inversión de uno de ellos; e incluyendo unidades de almacenamiento de un código de un nucleótido y selectivamente de un código seleccionado entre su código complementario y ningún código; para obtener un nucleótido y selectivamente su complementario; conectándose dichas unidades de almacenamiento en serie, para formar la cadena de nucleótidos que constituyen cada hebra de una cadena de "ADN", y así obtener su estructura electrónica y permitir el acceso selectivo a un nucleótido y selectivamente a su complementario, a una hebra y a las dos hebras de la cadena "ADN".

2.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 1, caracterizado porque se prevén diferentes módulos, cada uno constituido por agrupaciones de unidades de almacenamiento, en las que, en cada módulo se almacena una cadena de "ADN", estando los módulos conectados en filas y columnas para acceder a cada cadena de "ADN" mediante selección de filas o columnas.

3.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicaciones 1 ó 2, caracterizado porque el código

binario asignado a cada nucleótido comprende dos bits y su almacenamiento depende de las entradas aplicadas a cada unidad de almacenamiento.

4.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicaciones 1, 2 ó 3, caracterizado porque comprende medios de codificación de numeración en base 4 compuestos por los cuatro nucleótidos del "ADN".

5.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 4, caracterizado porque los dos bits correspondientes a cada nucleótido adquieren las cantidades decimales de 0 a 3, para conformar una pluralidad de representaciones numéricas determinadas por la posición de los dos bits y por el valor que reciben.

6.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 5, caracterizado porque comprende medios para realizar operaciones seleccionadas entre sumas, restas, multiplicaciones y divisiones; y comprende codificar cualquier número en coma fija o flotante de simple y doble precisión.

7.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 1, caracterizado porque las unidades de almacenamiento están seleccionadas entre memorias volátiles (IAM) y no volátiles (IROM).

8.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicaciones 2 y 7, caracterizado porque comprende medios de acceso estructurado mediante la localización de una cadena de "ADN", posterior localización de una hebra y posterior designación selectiva de un nucleótido o grupo de nucleótidos.

9.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE

LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicaciones anteriores, caracterizado porque comprende medios para codificar en las unidades de almacenamiento que forman hebras, representaciones de problemas mediante grafos que comprenden elementos constituidos por vértices y aristas y selectivamente peso (dirección); para codificar en una hebra un primer conjunto que contiene elementos del grafo constituidos por vértices y aristas correspondientes a elementos iniciales y los correspondientes a los elementos que no contienen elementos finales; y en la hebra complementaria se codifica un segundo conjunto que contiene elementos del grafo constituidos por elementos finales y los correspondientes a los elementos que no contienen elementos finales; estando cada unidad de almacenamiento de ambas hebras unidas entre sí mediante comparadores que forman una matriz para realizar paralelamente las comparación entre todos los elementos de ambos conjuntos y resolver problemas que puedan representarse mediante grafos.

10.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 9, caracterizado porque los comparadores que forman una matriz están conectados a un buffer al que se aplican medios de recuperación de caminos de orden polinomial para determinar la ruta más óptima entre dos puntos del grafo.

11.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 10, caracterizado porque comprende medios adicionales de almacenamiento del peso de una arista, medios para sumar los pesos de cada arista y obtener los pesos totales de cada arista, y medios de ordenación de los pesos totales de cada arista, según criterios previamente establecidos para resolver problemas de algoritmos convencionales seleccionados entre Dijkstra Floyd-Warshall,

Bellman-Ford y Ford-Fulkenson.

12.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 9, caracterizado porque se aplica en problemas de flujo seleccionados entre flotas de vehículos, optimización de procesos industriales, flujos de trabajo, problemas de acoplamientos y de redes de comunicaciones.

13.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 9, caracterizado porque comprende medios para codificar un problema del camino hamiltoniano en unidades de almacenamiento que constituyen dos hebras, en una de las cuales se codifica el grafo completo y en las complementarias el grafo completo con restricciones conocidas; estando las hebras conectadas a comparadores formando una matriz para comparar las aristas de ambos grafos dos a dos de forma paralela para indicar si los dos bits del resultado de cada comparación son todos iguales y se refieren a una misma arista y en consecuencia ésta pertenece a ambos grafos; aplicándose al resultado de la comparación a medios para crear niveles del árbol de Wallace indicativo de si la arista es camino o no lo es.

14.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 13, caracterizado porque los comparadores que comparan las aristas dos a dos, comprenden semisumadores lógicos sin acarreo, cuya salida está conectada a una puerta lógica de detección de un solo bits del resultado de la comparación para obtener un solo dato.

15.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 14, caracterizado porque los medios para crear niveles del árbol de Wallace comprenden unas primeras puertas lógicas por cada columna de la unidad de comparadores, que representa el grafo completo, y conectada

-61-

a sus salidas, para confirmar la pertenencia de dicha arista del grafo con restricciones, indicando si es o no camino.

16.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE
5 LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 14, caracterizado porque las salidas de las primeras puertas lógicas están conectadas a una segunda puerta lógica de detección de si todas las aristas son camino, determinando el camino hamiltoniano, e incluyendo
10 medios de almacenamiento del resultado obtenido.

17.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE
LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicaciones 3 y 16, caracterizado porque comprende medios para recibir los caminos encontrados y establecer en
15 correspondencia entre peso y arista y obtener los pesos de cada camino, e incluyendo medios de ordenación de los pesos de los caminos según criterios previamente establecidos, para la elección del mejor camino.

18.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE
20 LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 15, caracterizado porque comprende memorias adicionales de almacenamiento del peso de las aristas para establecer la correspondencia entre peso y arista, mediante el almacenamiento de la dirección de memoria referente a la
25 codificación de los vértices, sumándose los pesos para obtener los totales de cada camino que se almacena en las memorias adicionales con direcciones consecutivas, y mediante medios de ordenación de los pesos de los caminos, según criterios previamente establecidos, establece la
30 elección del mejor camino.

19.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE
LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 18, caracterizado porque se aplica a la resolución de problemas de ciclos hamiltonianos mediante
35 codificación indicativa de que el vértice de salida del

grafo es igual al vértice destino.

20.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 9, caracterizado porque comprende medios para codificar un grafo de manera que los vértices representan las aristas y las aristas a los vértices, y así obtener la secuencia de vértices que se deben seguir para pasar por todas las aristas y permitir resolver el problema del camino euleriano.

21.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicación 20, caracterizado porque comprenden medios para establecer codificación del peso de las aristas y plantear nuevos problemas de optimización en los caminos eulerianos.

22.- SISTEMA ELECTRÓNICO DE EMULACIÓN DE LA CADENA DE LA ESTRUCTURA DE "ADN" DE UN CROMOSOMA, según reivindicaciones 20 ó 21, caracterizado porque comprende medios para codificar en las hebras que el vértice inicial es el mismo que el vértice final para resolver ciclos eulerianos.

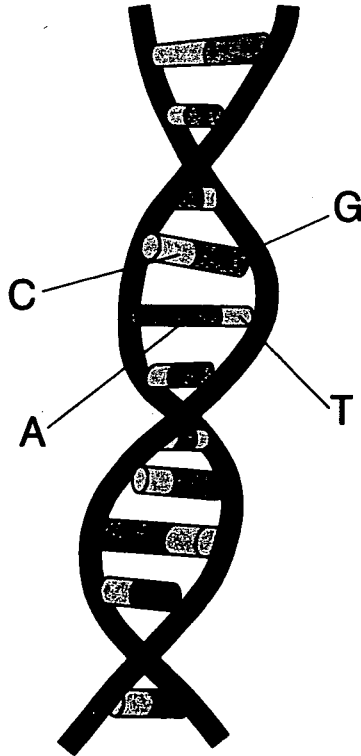


FIG. 1



FIG. 2

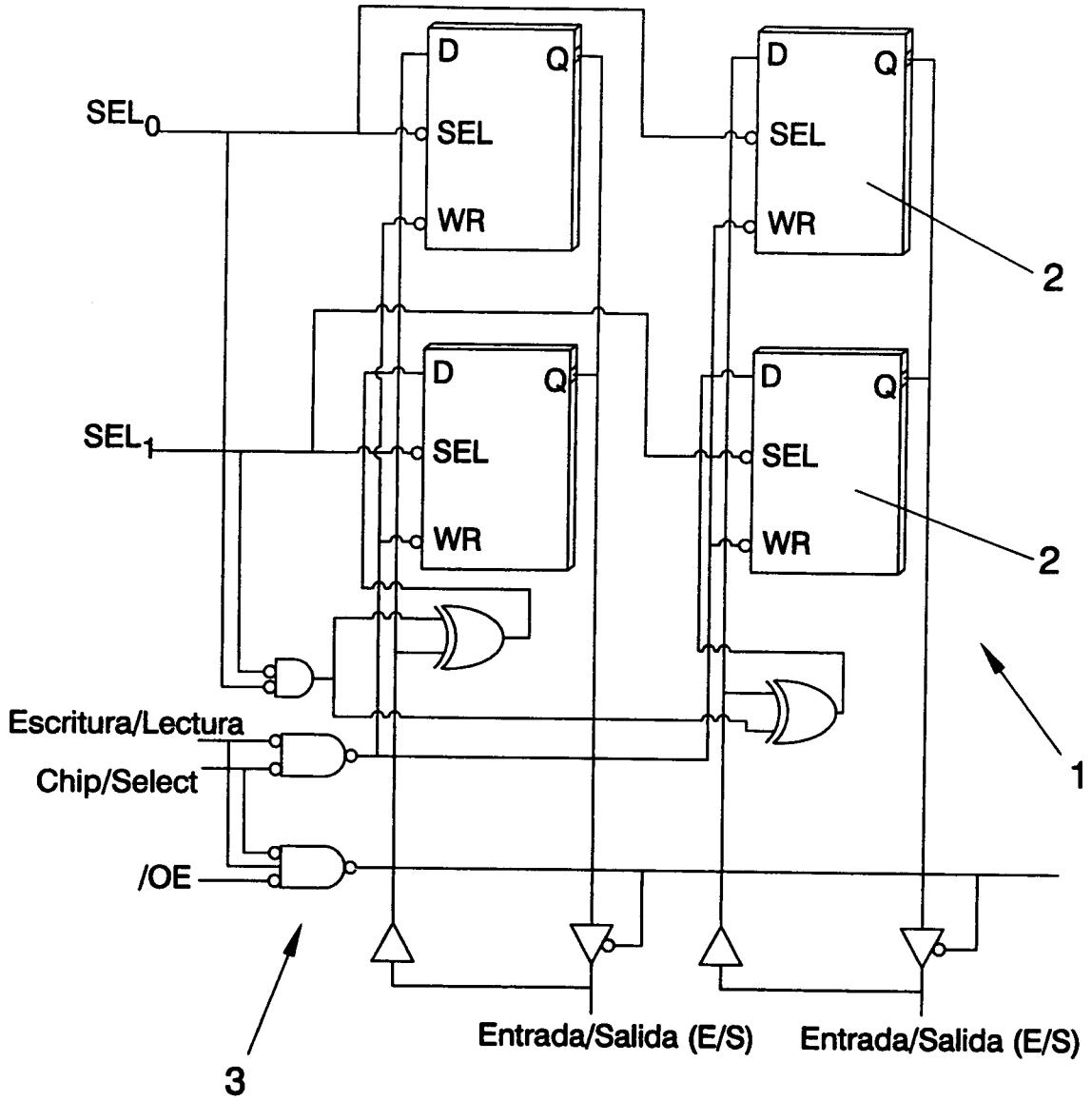


FIG. 3

Decimal	Binario	Octal	Hexadecimal	In Chro Sil
0	00	0	0	A
1	01	1	1	G
2	10	2	2	C
3	11	3	3	T
4	100	4	4	GA
5	101	5	5	GG
6	110	6	6	GC
7	111	7	7	GT
8	1000	10	A	CA
9	1001	11	B	CG
10	1010	12	G	CC
11	1011	13	D	CT
...

FIG. 4

4/17

	<i>A</i>	<i>G</i>	<i>C</i>	<i>T</i>
<i>A</i>	<i>A</i>	<i>G</i>	<i>C</i>	<i>T</i>
<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>GA</i>
<i>C</i>	<i>C</i>	<i>T</i>	<i>GA</i>	<i>GG</i>
<i>T</i>	<i>T</i>	<i>GA</i>	<i>GG</i>	<i>GC</i>

FIG. 5

	<i>A</i>	<i>G</i>	<i>C</i>	<i>T</i>
<i>A</i>	<i>A</i>	<i>GT</i>	<i>GC</i>	<i>GG</i>
<i>G</i>	<i>G</i>	<i>A</i>	<i>GT</i>	<i>GC</i>
<i>C</i>	<i>C</i>	<i>G</i>	<i>A</i>	<i>GT</i>
<i>T</i>	<i>T</i>	<i>C</i>	<i>G</i>	<i>A</i>

FIG. 6

$$\begin{array}{rcccccl}
 & T & G & A & T & \text{numero 1} \\
 - & G+G=C & C+G=T & C & C & \text{numero 2} \\
 \hline
 & G & \diagdown C & \diagdown C & G & \text{resultado}
 \end{array}$$

FIG. 7

	A	G	C	T
A	AA	AA	AA	AA
G	AA	AG	AC	AT
C	AA	AC	GA	GC
T	AA	AT	GC	CG

FIG. 8

	GTTG	numero 1
X	GGA	numero 2
	AA	producto G X A
	AAA	producto T X A(*)
	AAAA	producto T X A(*)
+	AAAAA	producto G X A(*)
	AAAAA	suma parcial
	AGA	producto G X G
	ATAA	producto T X G(*)
	ATAAA	producto T X G(*)
+	AGAAAA	producto G X G(*)
	AGTTGA	Suma parcial
	AGAA	producto G X G
	ATAAA	producto T X G(*)
	ATAAAA	producto T X G(*)
+	AGAAAAA	producto G X G(*)
	ACGTAGA	Resultado final (2500 en decimal)

(*) se añaden A's por la derecha.

FIG. 9

Signo	Hebra	Cod	Exp	Hebra	Cod	Mantisa
1	1	1	4	1	1	11

FIG. 10

Hebra	Coma fija + signo	Complemento A1	Complemento A2	Exceso 4 ⁴
Hebra1	GTTT (+127 ₁₀)	GTTT (+127 ₁₀)	GTTT (+127 ₁₀)	TTTT (+127 ₁₀)
Hebra2	CAAA (+0 ₁₀)	CAAA (-127 ₁₀)	CAAA (-128 ₁₀)	AAAA (-128 ₁₀)
Hebra1
Hebra2
Hebra1	AAAA (+0 ₁₀)	AAAA (+0 ₁₀)	AAAA (+0 ₁₀)	CAAA (+0 ₁₀)
Hebra2	TTTT (-127 ₁₀)	TTTT (-0 ₁₀)	TTTT (-1 ₁₀)	GTTT (-1 ₁₀)
Hebra1	CAAA (-0 ₁₀)	TTTT (-0 ₁₀)	AAAA (-0 ₁₀)	CAAA (+0 ₁₀)
Hebra2	GTTT (+127 ₁₀)	AAAA (+0 ₁₀)	TTTT (-1 ₁₀)	GTTT (-1 ₁₀)
Hebra1
Hebra2
Hebra1	TTTT (-127 ₁₀)	CAAA (-127 ₁₀)	CAAG (-127 ₁₀)	AAAG (-127 ₁₀)
Hebra2	AAAA (+0 ₁₀)	GTTT (+127 ₁₀)	GTTT (+126 ₁₀)	TTTC (+126 ₁₀)
Hebra1			CAAA (-128 ₁₀)	AAAA (-128 ₁₀)
Hebra2			GTTT (+127 ₁₀)	TTTT (+127 ₁₀)

FIG. 11

Signo	Hebra	Cod	Exp	Hebra	Cod	Mantisa
1	1	1	7	1	1	28

FIG. 12

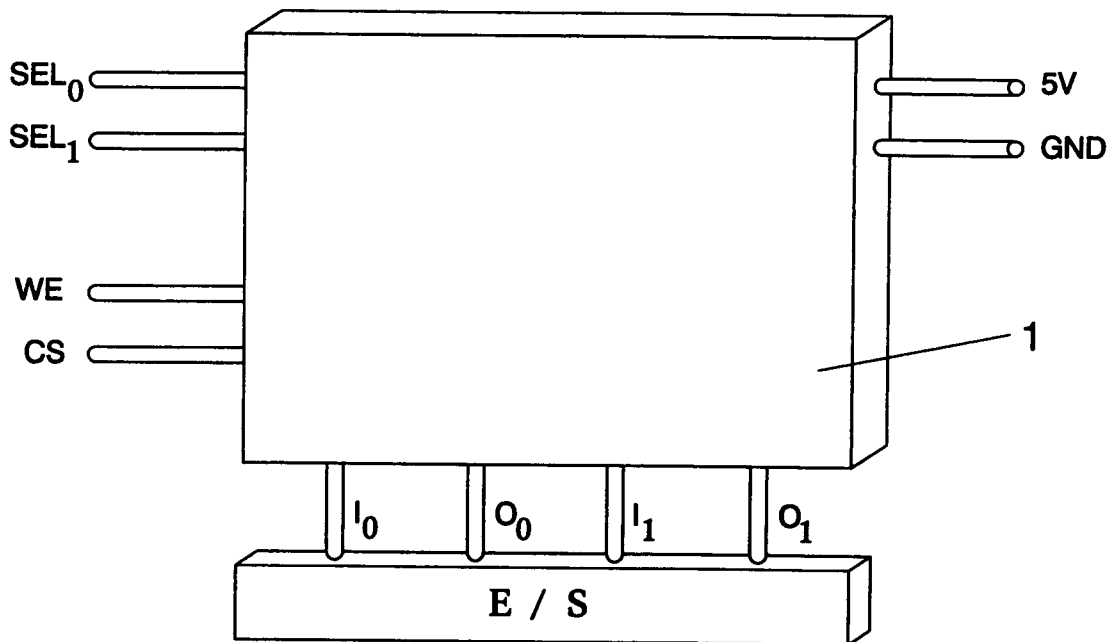


FIG. 13

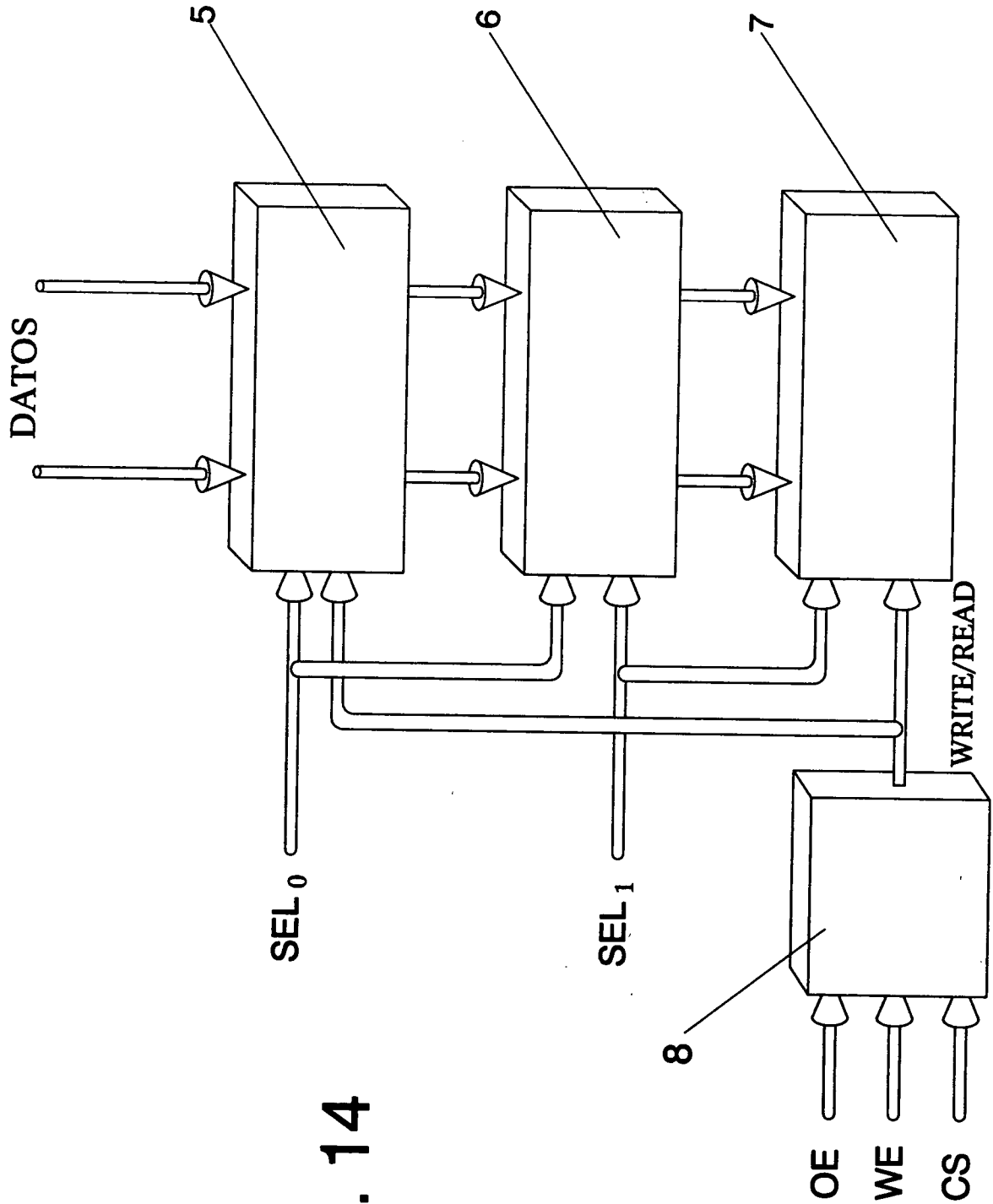


FIG. 14

9/17

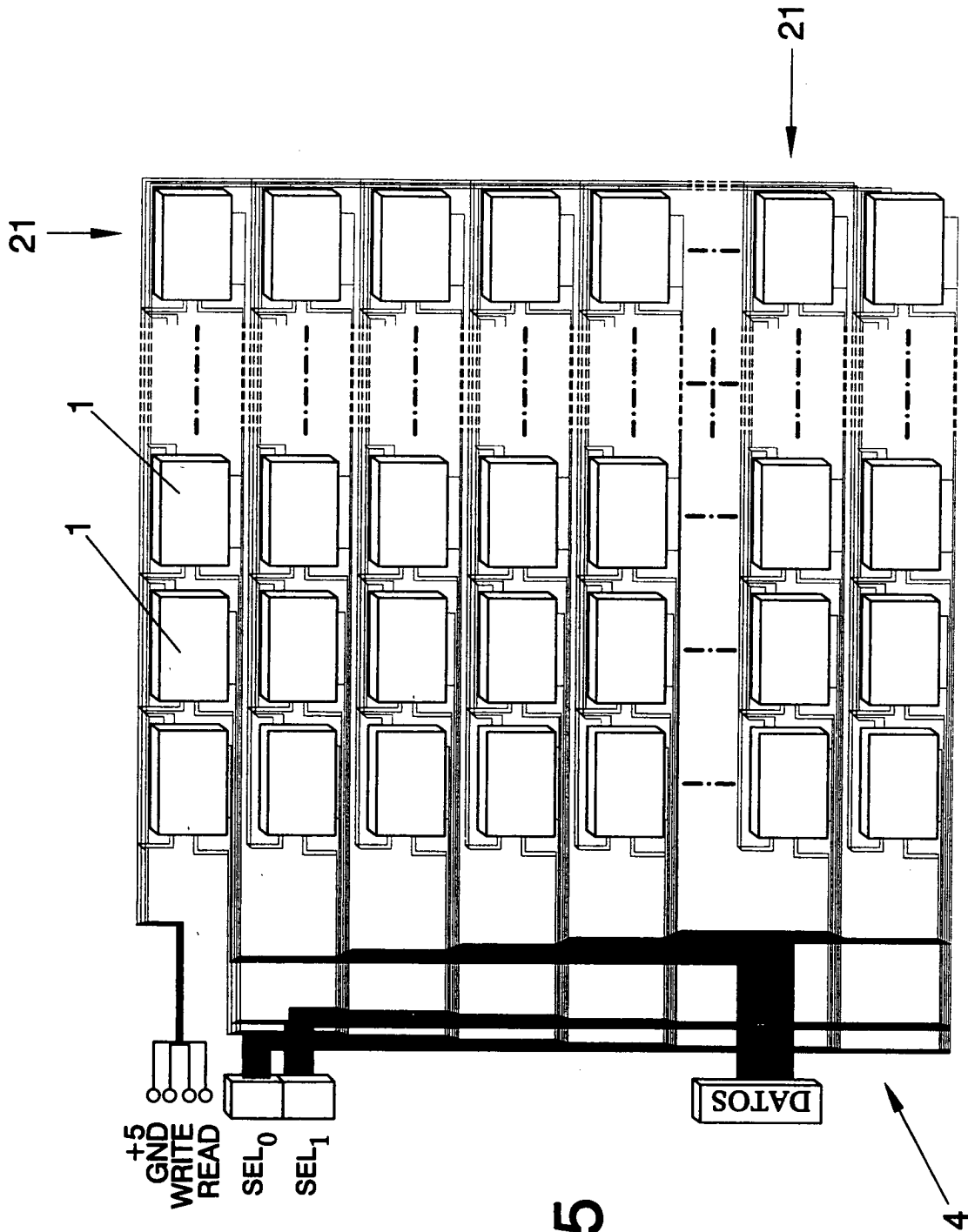


FIG. 15

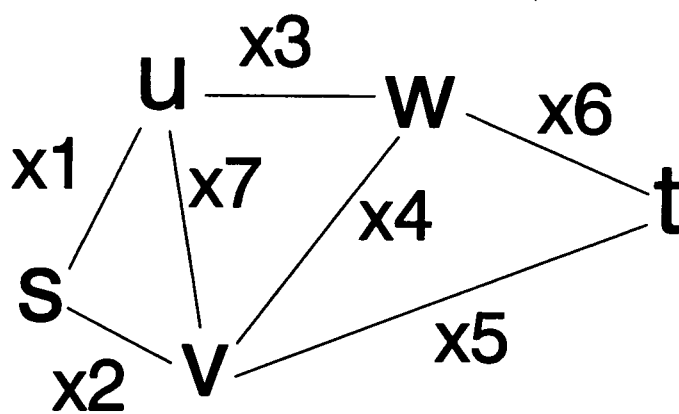


FIG. 16

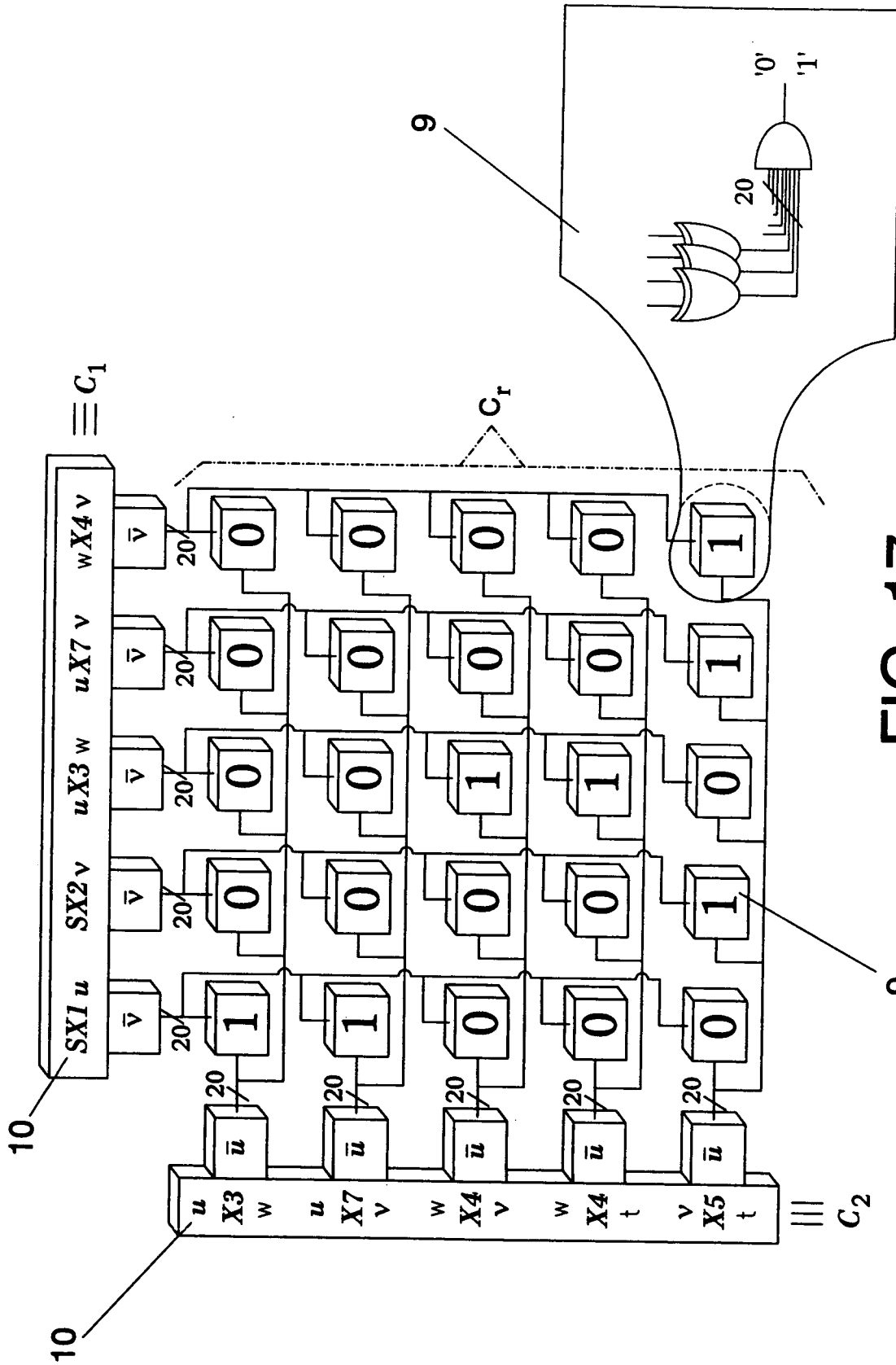


FIG. 17

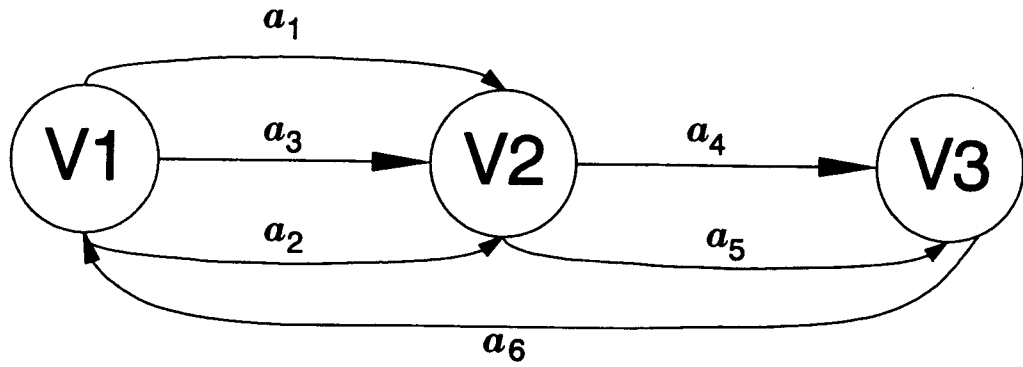


FIG. 18

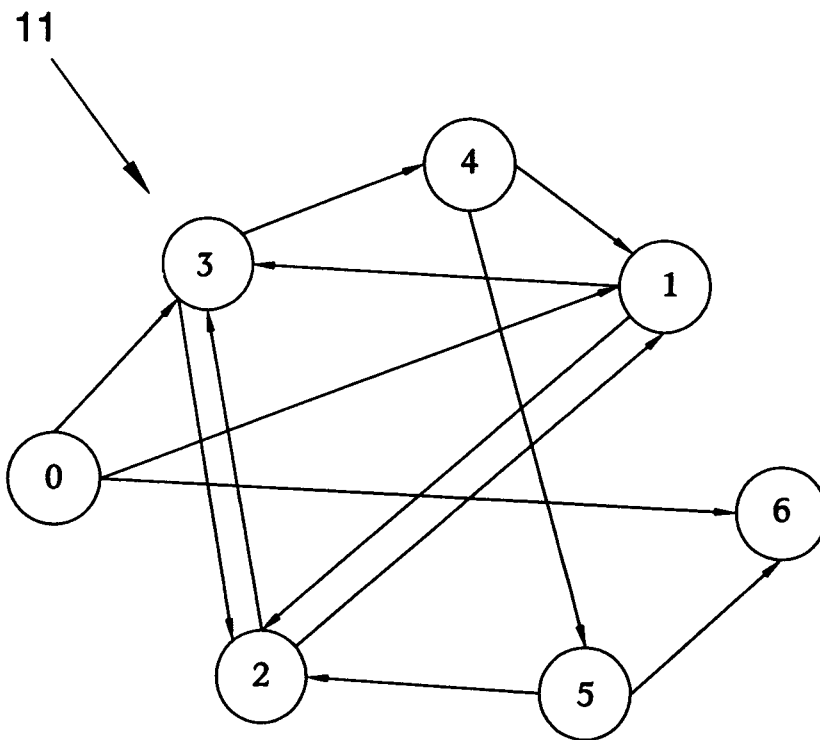


FIG. 19

13/17

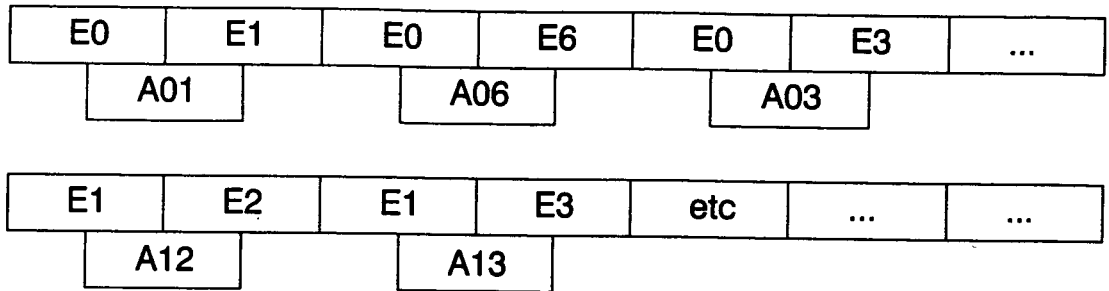


FIG. 20

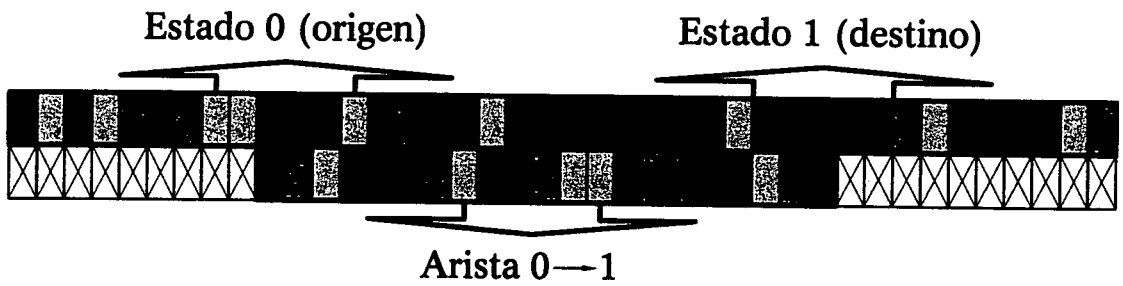


FIG. 21

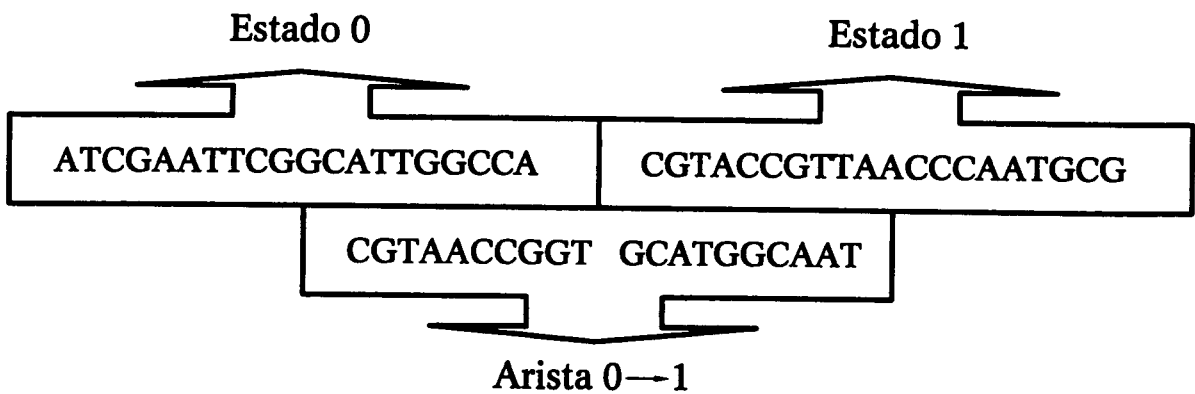


FIG. 22

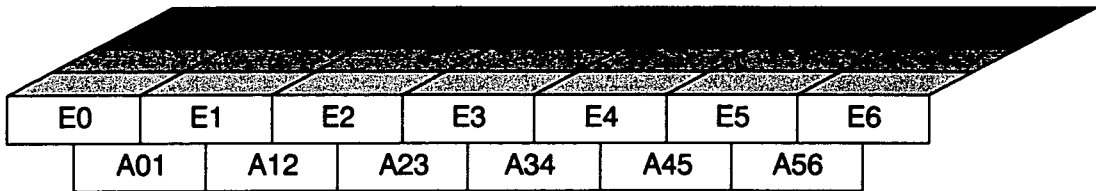


FIG. 23

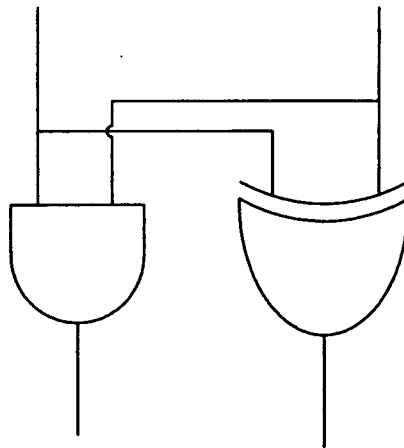


FIG. 24

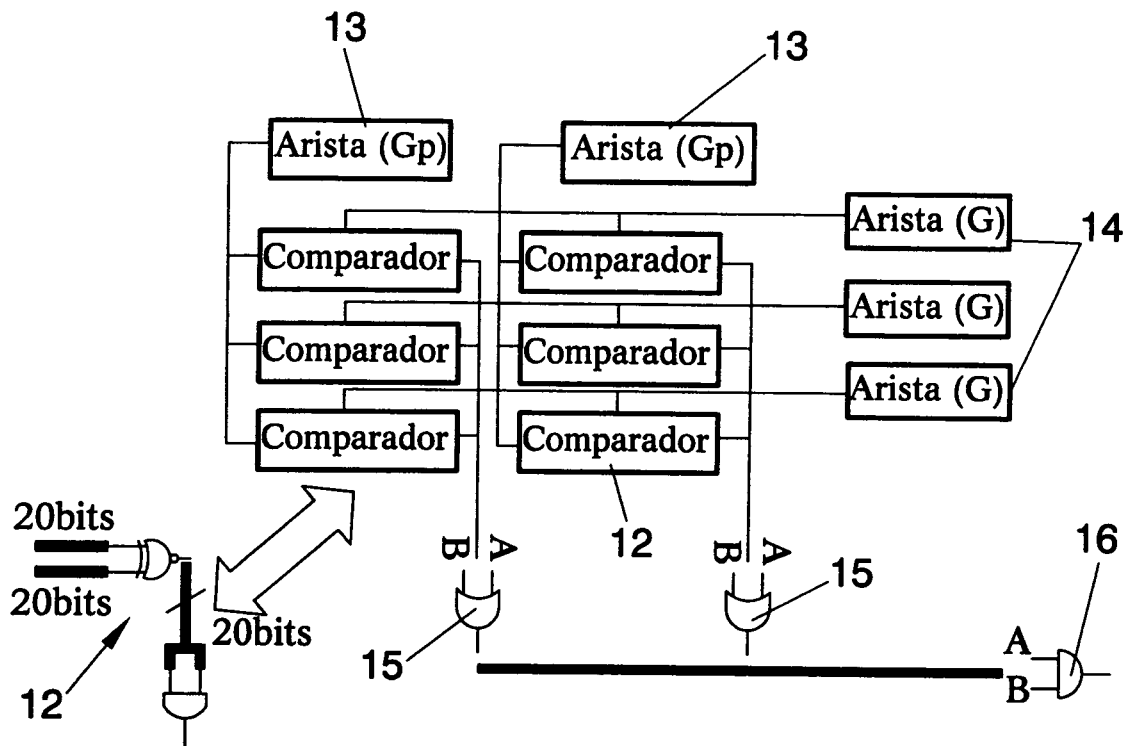


FIG. 25

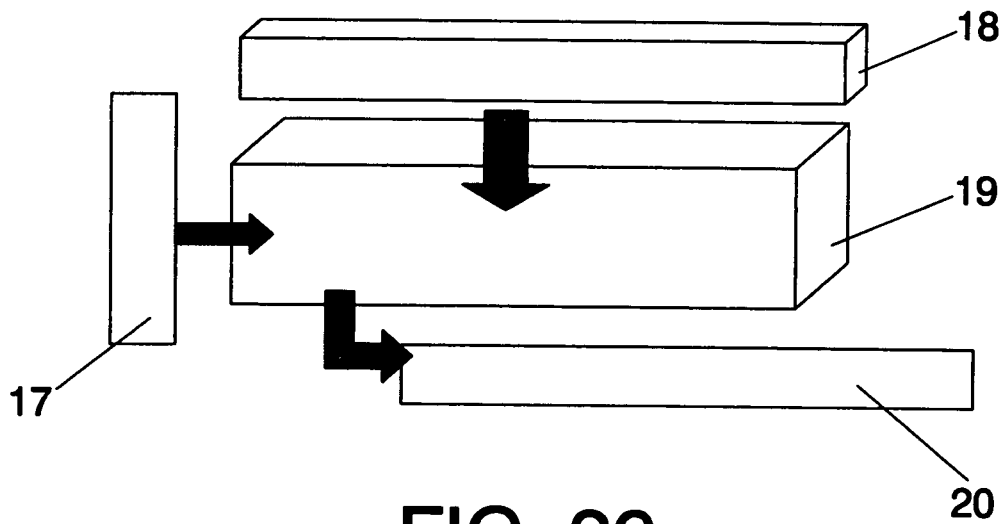


FIG. 26

16/17

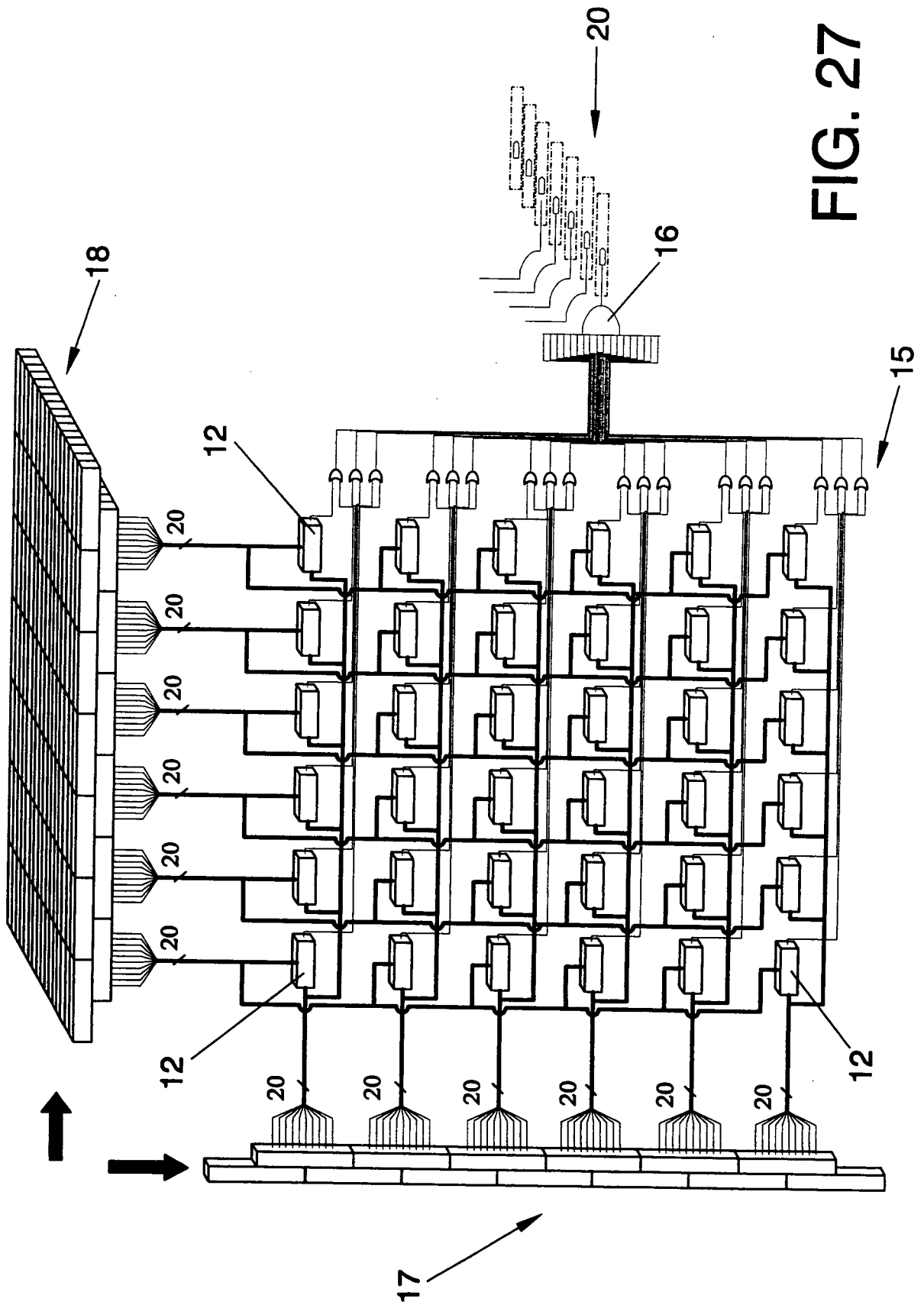


FIG. 27

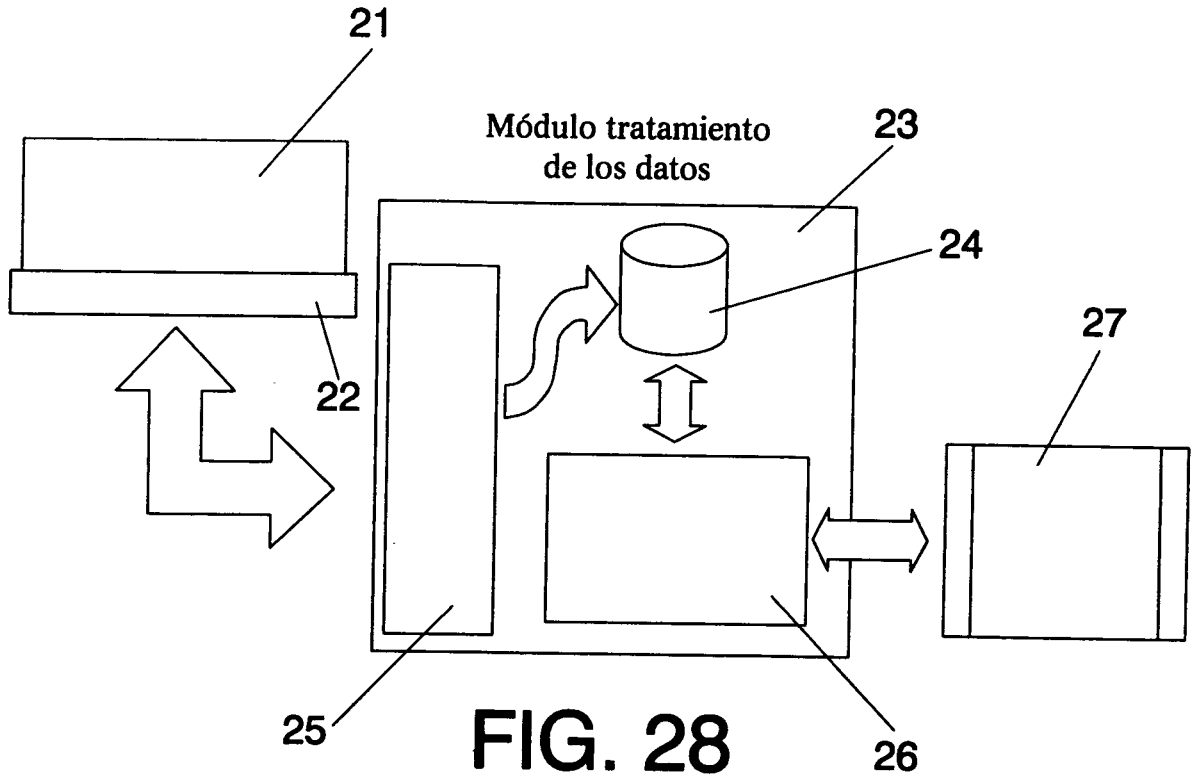


FIG. 28

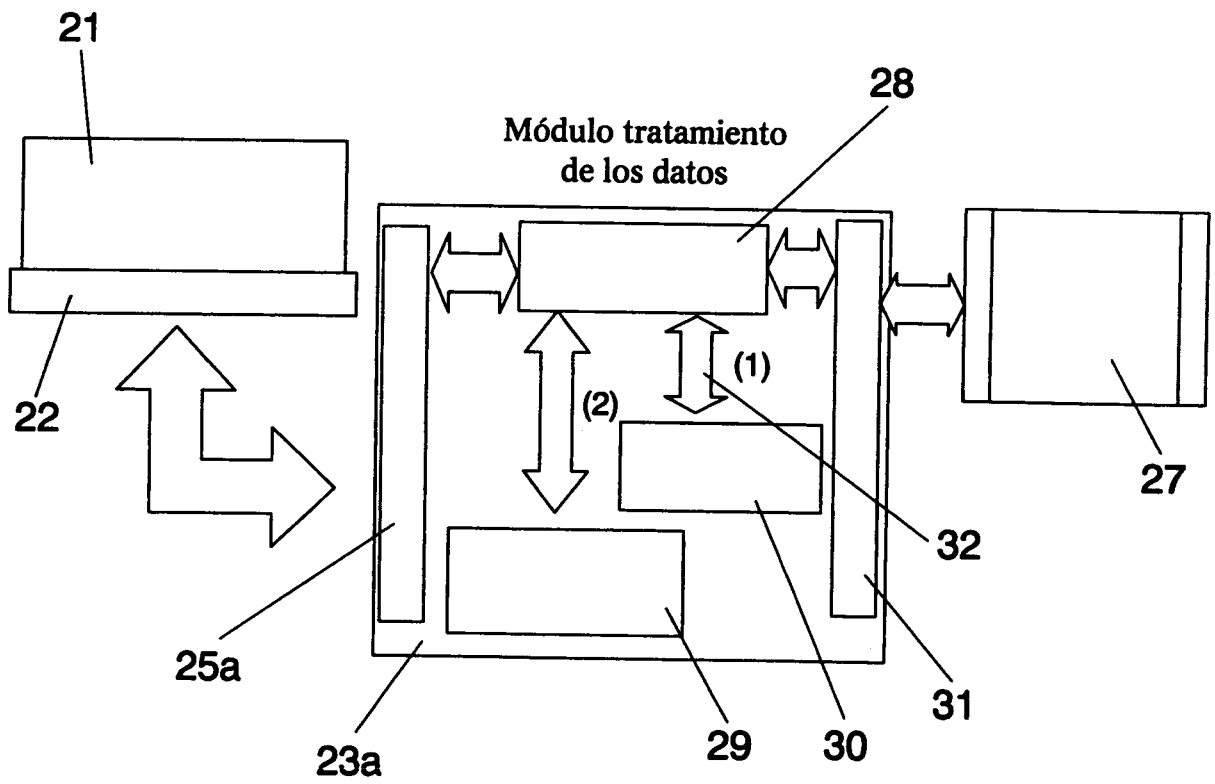


FIG. 29