UNIVERSIDAD DE GRANADA

ESTUDIOS DE INGENIERIA ELECTRÓNICA

PROYECTO FIN DE CARRERA

ARQUITECTURAS DE COMPUTACIÓN BASADAS EN DISPOSITIVOS DE PUNTOS CUÁNTICOS

CURSO : 2002/2003

ÁNGEL DÍAZ RODRÍGUEZ
ALBERTO PRIETO ESPINOSA
(Director)

UNIVERSIDAD DE GRANADA

ESTUDIOS DE INGENIERIA ELECTRÓNICA

ARQUITECTURAS DE COMPUTACIÓN BASADAS EN DISPOSITIVOS DE PUNTOS CUÁNTICOS

REALIZADO POR :

Ángel Díaz Rodríguez

DIRIGIDO POR :

Alberto Prieto Espinosa

DEPARTAMENTO :

Arquitectura y tecnología de Computadores

Palabras clave : Computación Cuántica

Resumen :

Se hace una introducción a la computación cuántica, así como un estudio de los puntos cuánticos como puertas lógicas.

Alberto Prieto Espinosa, Catedrático de la Universidad de Granada, como director del Trabajo Fin de Carrera de Ángel Díaz Rodríguez,

MANIFIESTA: que el presente trabajo titulado:

ARQUITECTURAS DE COMPUTACIÓN BASADAS EN DISPOSITIVOS DE PUNTOS CUÁNTICOS

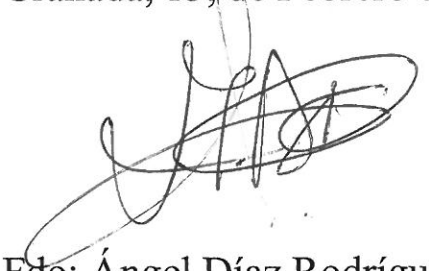
que ha sido redactado y presentado por dicho alumno, corresponde a la investigación realizada bajo su dirección, y con esta fecha autoriza su presentación:

Granada a 13 de Febrero de 2003

Fdo: Alberto Prieto Espinosa

Yo, Ángel Díaz Rodríguez, con DNI : 24278388, autorizo a que la presente copia de mi Proyecto Fin de Carrera se ubique en la biblioteca del centro para ser libremente consultada por las personas que lo deseen.

Granada, 13, de Febrero de 2003



Fdo: Ángel Díaz Rodríguez

UNIVERSIDAD DE GRANADA

ESTUDIOS DE INGENIERIA ELECTRÓNICA

PROYECTO FIN DE CARRERA

AUTOR : .....

TÍTULO : .....

TRIBUNAL :

D. Juan A. López Villanueva.....

D. Alberto Prieto Espinosa.....

D. Alberto J. Palma López.....

Presentado en Granada a 25 de Febrero de 2000

Evaluado en Granada a 26 de Febrero de 2000

El Presidente El Vocal El Secretario

INDICE

CAPITULO 1

1.INTRODUCCIÓN

En este capitulo se hace una introducción histórica sobre las computadoras, y como han ido evolucionando a lo largo de la historia. Se justifica el uso de la física cuántica en computación debido a que el nivel de integración de los componentes esta llegando a los limites físicos. Por lo que es necesaria una alternativa al silicio.

1.1.GENERALIDADES

Desde hace 20 años, los científicos intentan aprovechar la mecánica cuántica para propiciar un cambio radical en los fundamentos de la informática.

La evolución del ordenador desde su creación en 1.946 hasta nuestros días ha sido vertiginosa, especialmente desde los años setenta, donde la potencia de los computadores se ha ido duplicando, cada dos años, hasta nuestros días.

En 1965, Gordon Moore, cofundador de Intel, hizo su famosa predicción que posteriormente sería conocida como la Ley de Moore, según la cual el numero de transistores en un circuito integrado se duplica cada 18 meses. Esta ley se ha venido cumpliendo hasta la actualidad, ver figura 1.1, lo que significa que en menos de cuarenta años, los circuitos han pasado de tener unos pocos cientos de transistores a estar cercanos a los cien millones de transistores.

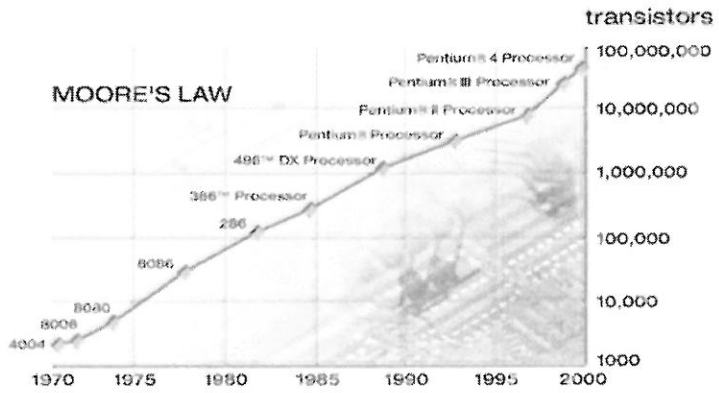


Figura 1.1. Ley de Moore

Debido a que cuantos mas transistores posea un circuito integrado mayor es la complejidad de los dispositivos que se pueden construir con los mismos y por tanto las operaciones que estos pueden realizar, la potencia de los mismos también se duplica cada 18 meses, así como cada vez es mas barato comprar un determinado circuito.

Todos los circuitos integrados están fabricados con transistores situados en una superficie plana y unidos mediante pistas. Hasta ahora la potencia de los circuitos integrados se ha venido aumentando mediante la miniaturización de los componentes que lo forman. Actualmente, los componentes que forman los circuitos tienen unos tamaño en torno a las 0.18 micras, y se considera que todavía pueden reducirse aun más. Por ejemplo, el AMD Athlon XP que usa esta tecnología de 0.18 micras permite integrar 37.5 millones de transistores en una superficie de 128 mm^2

El crecimiento exponencial en la miniaturización del transistor alcanzará su límite cuando un bit de información pueda codificarse en un solo átomo. Es entonces cuando surgirán efectos mecánico-cuánticos. Será posible usar estos efectos para crear una alternativa tecnológica al transistor.

Hay que hacer aquí un poco de historia. A principios del siglo pasado con el avance en el conocimiento de los mecanismos internos del átomo; físicos de la talla de Niels Bohr, Max Planck, Werner Heisenberg, etc. llegaron a la conclusión de que la física newtoniana (también llamada mecánica clásica) no podía aplicarse al mundo subatómico, en el cual las leyes del movimiento responden a principios diferentes que en ocasiones contradicen nuestro sentido común. Llevándolos a fundar una nueva rama de la física llamada mecánica cuántica.

Un ejemplo sencillo: la mecánica newtoniana es capaz de establecer con una gran precisión, la velocidad y la posición de objetos de mayor tamaño que el átomo; en este sentido se puede establecer con objetividad la trayectoria que por ejemplo siguen la Luna o el Sol, y ésta determinación es independiente del método de estudio empleado, es decir, la observación que hagamos no influye en forma significativa en las conclusiones que obtenemos sobre el movimiento estudiado (en este sentido, nuestra observación es objetiva).

Sin embargo, cuando se trata de observar y estudiar objetos del tamaño del átomo o menores, la mecánica cuántica sostiene que es imposible hacer observaciones objetivas, en el sentido de que no perturben de manera importante nuestro objeto de estudio. Se aplica aquí un principio denominado Principio de Incertidumbre de Heisenberg[1].

Las leyes de la mecánica cuántica implica una forma diferente de procesar la información respecto a la física clásica. El tratamiento de la información contenida en la función de onda de un sistema cuántico, es lo que trata de estudiar la teoría cuántica de la información.

La manipulación cuántica ofrece aplicaciones reales y potenciales como la criptografía cuántica (seguridad), tele transportación y codificación densa (transmisión), corrección de errores (fiabilidad) y diseño de algoritmos con una velocidad impensable hasta ahora; por ejemplo, un ordenador convencional tardaría millones de años en factorizar números muy grandes, mientras que un ordenador cuántico lo haría en una fracción de segundo.

Las ventajas más características de la computación cuántica frente a la clásica son: la superposición, que permite ejecutar múltiples operaciones matemáticas a la vez y el enredado cuántico, que proporciona una correlación entre las respuestas mucho mayor a la correlación clásica.

Sin embargo se pierde en estabilidad, ya que el ordenador cuántico es muy sensible al ruido, tanto más cuanto más grande sea. Otro problema que se presenta es la imposibilidad de poder amplificar una señal cuántica debido al teorema de la no-clonación (no existen dos estados coherentes exactamente iguales formados entre dos o más átomos) limitando así el alcance de la comunicación cuántica.

---

[1] Ver apéndice B

Aunque no parece cercana la construcción de un ordenador cuántico, algunas ideas sobre su diseño y funcionamiento ya tienen aplicaciones prácticas: la transmisión de información mediante canales cuánticos ha permitido crear un sistema de comunicación totalmente seguro (criptografía cuántica).

La computación cuántica tiene básicamente dos efectos en la tecnología de las computadoras: En términos de hardware, a medida que la información pase a ser representada por unas cuantas partículas subatómicas, (a diferencia de como se representa ahora mediante una gran cantidad de éstas a través de los diferenciales de voltaje en los componentes de la computadora), los dispositivos deberán de reconocer los fenómenos cuánticos, como por ejemplo: las partículas pueden tener varios estados atómicos a la vez (niveles de energía), pueden atravesar barreras aparentemente infranqueables, pueden seguir varias rutas a la vez, etc...

En relación a los algoritmos (procedimientos matemáticos para resolver problemas), la computación cuántica abre posibilidades antes no imaginadas: disminuciones exponenciales en el tiempo de procesamiento y realización de operaciones en paralelo sin la necesidad de agregar procesadores a la máquina.

1.2. DESCRIPCIÓN DEL PROBLEMA

El tamaño de los componentes de los circuitos electrónicos no podrá seguir disminuyendo eternamente, y la ley de Moore dejara de ser efectiva. Los motivos por los cuales los circuitos integrados basados en el silicio tendrán que alcanzar un límite son varios, y los expertos calculan que para dentro de 15 o 20 años se habrá alcanzado este límite, y habrá que usar otros sistemas para la fabricación de los circuitos integrados y por lo tanto de los computadores. Para entender cual es el problema al que se enfrenta la industria de fabricación de circuitos integrados hay que conocer en que se basa la fabricación de los mismos.

Un chip actual de solo unos centímetros cuadrados de superficie contiene varios millones de componentes, cada uno de estos componentes mide menos de media micra de diámetro lo que viene a ser mas o menos doscientas veces mas pequeño que el grosor de un pelo. Estos componentes están hechos básicamente de silicio que conduce la electricidad y de dióxido de silicio que es un aislante, estos dos compuestos se unen formando capas aislantes y conductoras. El procedimiento para

grabar los circuitos electrónicos en el silicio emplea una técnica llamada fotolitografía, mediante la cual sobre las capas de silicio y de dióxido de silicio se forma una capa de polímeros en la que posteriormente se graba el patrón del circuito exponiéndolo a la luz a través de una mascara. Después se utilizan substancias que corroen las partes del silicio no protegidas por el polímero de tal forma que se obtienen las zonas conductoras rodeadas de material no conductor.

Por lo tanto, la reducción del tamaño de los componentes del circuito vendrá determinado por la longitud de onda que se usa para grabar el patrón en la oblea. Así para obtener componentes mas pequeños los fabricantes utilizaran ondas luminosas de menor longitud, como la luz ultravioleta de menor longitud de onda, los rayos X, etc..

Considerando que se puedan seguir fabricando componentes de tamaño cada vez menor, utilizando para ello longitudes de onda también menores, se llegaran a obtener circuitos en los que las capas aislantes no tengan mas que unos pocos átomos de grosor, con lo cual cabe preguntarse si los materiales aislantes conservaran completamente sus capacidades aislantes, de no ser así, los circuitos fabricados de esta forma no funcionarían correctamente.

Además, al trabajar con dimensiones tan pequeñas las leyes de la física normal dejan de tener validez y hay que tener en cuenta la física cuántica. Ya que se trabajara con tamaños tan pequeños en los que se habrá un numero muy reducido de electrones, lo que quiere decir que como es normal en la física cuántica todo estará determinado por probabilidades. Es decir, podría llegar a ocurrir que ya no tuviésemos un 1 o un 0 sino una probabilidad de tener cada valor, con los problemas de indeterminación que esto supondría.

Otro problema de la miniaturización surge de las pistas que unen los distintos componentes del circuito integrado. Estas pistas se han ido reduciendo a la vez que se han reducido los transistores, sin embargo, al contrario que estos, su rendimiento cae cuanto mas pequeños se hacen debido a que aumenta la resistencia que tienen al paso de la corriente eléctrica. Este aumento de resistencia hace que cada vez los circuitos integrados se calienten más, que unido a que debido a la gran integración de componentes, hace que esta energía sobrante no se disipe correctamente, con lo que se produce un sobrecalentamiento, que puede producir el fallo total del circuito. Este problema se ha solucionado hasta ahora utilizando varias capas de interconexión con

lo que se consigue disminuir la resistencia, pero si se sigue reduciendo el tamaño habrá que buscar nuevas alternativas, como por ejemplo, la utilización de nuevos materiales que ofrezcan menos resistencia, o ninguna resistencia como es el caso de los superconductores. También se deberá solucionar el problema del suministro del voltaje eléctrico.

Cuando los componentes de los circuitos electrónicos se hacen mas pequeños se debe reducir el voltaje, ya que cada vez son mas delicados. El problema es que si se reduce demasiado el voltaje habrá que tener cuidado para mantener de alguna forma las fluctuaciones que pueda haber ya que al movernos en rangos de voltajes mas pequeños se podría llegar a confundir el voltaje correspondiente a un 0 con un 1, con las consecuencias que esto traería. Hasta ahora hemos tenido en cuenta los problemas físicos que la reducción del tamaño de los componentes acarrea, sin embargo, hay que tener en cuenta algo que la Ley de Moore ya contemplaba, y es que el precio de fabricación de un circuito electrónico también aumenta acorde con la miniaturización de los componentes, y por lo tanto, cada vez cuesta mas fabricar un microprocesador. Esto es debido a que se necesitan realizar grandes inversiones en tecnología para poder construir elementos tan pequeños. Por ejemplo, los impresores de las obleas de silicio que tiene Intel cuestan alrededor de 10 millones de dólares cada una. Por lo tanto, puede llegar el momento en el que construir un circuito electrónico sea sumamente caro y no compense económicamente. La ley de Moore se enfrentara a su final cuando llegue a los límites físicos o económicos cosa que sucederá antes o después, ya que esta ley no es una ley física si no una ley obtenida de la observación de la tendencia en la construcción de circuitos en el momento en el que se realizo. Hasta aquí se han expuesto distintos aspectos que provocaran el fin de la ley de Moore, aunque existen numerosos expertos que consideran que no se puede hablar todavía de un fin ya que siempre cabe la posibilidad de descubrir algo que no se había tenido todavía hasta ahora que posibilitaría aumentar todavía más aumentar la integración solucionando los problemas antes mencionados, por lo que la ley de Moore podría seguir cumpliéndose por varias décadas más.

Por si acaso, varias universidades y empresas sobre todo en EEUU, están llevando a cabo investigaciones para encontrar formas de computación viables y ajenas al silicio que abran nuevas vías para las ciencias de la computación[2].

[2] Para ver otros tipos alternativos de computación ver apéndice D

CAPITULO 2

2. FUNDAMENTO FÍSICO

En este capitulo se estudia el fundamento físico de la computación cuántica, se explica que es la información cuántica, y se estudia las propiedades que tiene este método de computación: enredo cuántico, tele transportación, paralelismo cuántico y criptografía cuántica.

2.1 TEORÍA CUÁNTICA

Según la física quántica cualquier partícula (electrón, neutrón, protón,...) esta en todos los estados a la vez. Esto es así porque la situación de una partícula viene determinada por una onda de probabilidad. En cierta medida se puede decir que una partícula se puede encontrar en cualquier lugar de esa onda con la probabilidad indicada por esta. Es en el momento de interactuar con la partícula cuando esta se encuentra en un punto concreto.

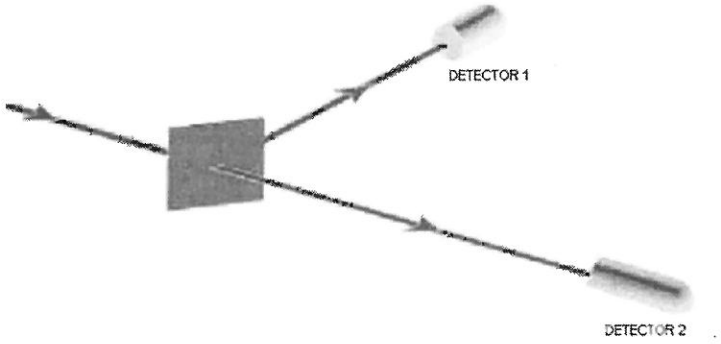


Figura 2.2. Experimento 1. Espejos semireflectantes al 50%

Para probar esto, se usa un experimento con espejos semireflectantes al 50%. Si lanzamos un fotón contra el espejo semireflectante que refleja la mitad de la luz mientras que la otra mitad la deja pasar, es de suponer que el fotón llegue a cada uno de los detectores en el 50% de los casos, es decir, el 50% de las veces pasara a través del espejo y el otro 50% se reflejara. La mecánica cuántica sostiene que este fotón no realizara un camino y luego otro, sino que recorrerá ambos caminos a la vez.

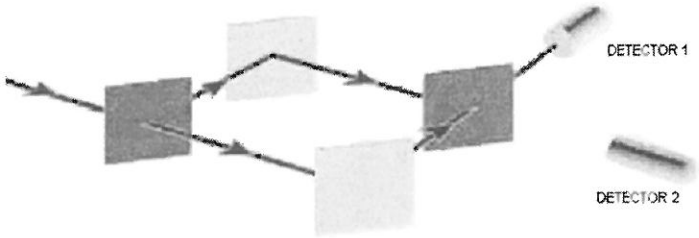


Figura 2.2. Experimento 2. Dos espejos reflectantes al 100% y otro semireflejante al 50%

Esto se demuestra añadiendo al ejemplo dos espejos reflectantes al 100% y otro semireflectante creando así dos caminos para llegar a los detectores. Si suponemos que el fotón recorrerá uno u otro camino con una probabilidad del 50% para cada camino y que el espejo final es semireflectante al 50% se asume que llegara a ambos detectores con igual probabilidad. En cambio, se observa lo que se denomina un fenómeno de interferencia cuántica, es decir, si ambos caminos son iguales en longitud, llegara a un detector el 100% de las veces y al otro el 0% de las veces.

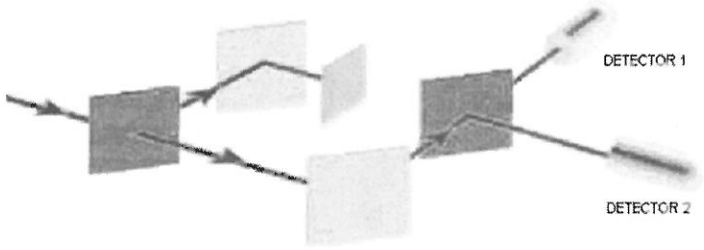


Figura 2.3. Experimento 3. Dos espejos reflectantes al 100%, uno de ellos bloqueado y otro semireflejante al 50%

Sin embargo, si bloqueamos uno de los dos caminos con una superficie que absorba el fotón, el fotón llega a ambos detectores con igual probabilidad. Viendo esto se asume que el fotón de alguna manera sabe que el camino al detector 2 esta bloqueado o no, y esto solo se puede explicar si el fotón recorre ambos caminos a la vez. Así pues, se dice que el fotón esta coherentemente superpuesto en el camino transmitido y en el camino reflejado.

2.2. INFORMACIÓN CUÁNTICA

En un ordenador cuántico, el almacenamiento de información básica se realiza en un sistema cuántico: el qubit[3] o bit cuántico, formado por dos estados independientes. La gran diferencia entre el bit clásico y el qubit estriba en que, mientras que los bit clásicos representan estados independientes unos de otros, los qubits cuánticos representan ambos estados simultáneamente, un "0" y un "1" lógico, dos estados ortogonales de una sub partícula atómica, como se ve en la figura 2.1. El estado de un qubit se puede escribir como $\{|0\rangle, |1\rangle\}$, describiendo su múltiple estado simultaneo, es decir, los estados pueden interferir entre sí formando estados colectivamente enredados que aumenta en gran medida la rapidez de las operaciones. La base ortonormal $|0\rangle$ y $|1\rangle$ puede corresponder a los dos estados de polarización $|\uparrow\rangle$ y $|\rightarrow\rangle$ de un fotón o a los dos estados del espín de un electrón $|\uparrow\rangle$ y $|\downarrow\rangle$.

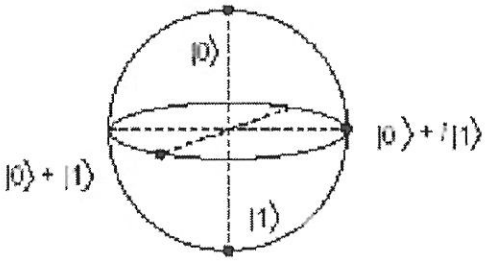


Figura 2.4. Representación de cuatro estados diferentes de un qubit

[3] Quantum bit, qubit por sus siglas en ingles, este termino fue acuñado por Schumacher en 1995

Supongamos que N qubits viven en un espacio de Hilbert de dimensión 2^N donde hay 2^N estados ortogonales. Los espacios de estados se combinan con el producto tensorial[4] mientras que en la computación clásica se usa el producto cartesiano. El crecimiento exponencial (2^N) viene dado por los estados que no son productos de estados individuales; por ejemplo:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle)$$

Veamos que ocurre para N=2 y para N=4:

Para N=2. Sean dos qubits cuyas bases ortonormales son $\{|0\rangle, |1\rangle\}$ el producto tensorial de estas dos bases nos va a dar la siguiente base $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$ está se puede escribir de forma más compacta $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, es decir, podemos definir un registro en una superposición con todos los números del 0 al 3 como :

$$|a\rangle = (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) = |00\rangle + |10\rangle + |01\rangle + |11\rangle = \sum_{x=0}^{3} |x\rangle$$

Los estados de espín podemos definirlos como $|\uparrow\rangle \equiv |1\rangle$ y $|\downarrow\rangle \equiv |0\rangle$.

Para N=4 qubits, la aplicación de la operación unitaria $U_{\otimes}$ que implementa el algoritmo suma módulo 2^2 entre el estado descrito en el ejemplo anterior para N=2 y un segundo estado con $|b\rangle = |x'\rangle$ (con x'=0, 1, 2, 3) da una salida de la forma:

$$|a\rangle|b\rangle = \sum_{x=0}^{3} |x\rangle|x'\rangle \xrightarrow{U_{\oplus}} \sum_{x=0}^{3} |x\rangle|x \oplus x'\rangle$$

Lo que se hace en este caso es calcular de manera simultánea la suma $x \oplus x'$ para cuatro valores distintos de x, mientras que para que esto ocurra con ordenadores clásicos debemos colocar cuatro que trabajen en paralelo. A este fenómeno descrito

[4] Ver apéndice A

en los ejemplos se le denomina *paralelismo cuántico* (se verá más adelante). Esto es lo que hace que un computador cuántico sea mucho más potente que uno clásico, pero sólo podemos medir una de las cuatro respuestas en la salida: a esto se le llama *interferencia cuántica*, en la que se produce un cambio de fases relativas de forma que se favorezca selectivamente el valor en el que estamos interesados. Pero lo que realmente hace potente a la computación cuántica es el llamado enredado cuántico o estados "entangled" (se verá más adelante).

En un ordenador, un bit de información viene representado por una cierta situación eléctrica. Por ejemplo, un condensador cargado puede representar un 1 y otro descargado un 0. El bit no es sino un cuanto de información. La pregunta es si podemos crear sistemas cuánticos que funcionen como un condensador. Uno de los dispositivos que se han propuesto para realizar computación cuántica, ensayado con éxito en el laboratorio, usa cadenas de iones atrapados por campos electromagnéticos. Con estos campos se consigue construir una "trampa" que mantiene esos iones confinados en un espacio donde previamente se ha hecho el vacío. Al hacer incidir sobre ellos un haz láser se consigue prácticamente congelarlos en su posición formando una cadena estable. El electrón más externo de cada ión puede encontrarse en dos estados distintos que, a su vez, pueden manipularse con láser. Esos dos estados del electrón constituyen el llamado qubit. En este dispositivo tendremos tantos qubits como iones podamos almacenar en la trampa electromagnética.

Otro prototipo se basa en construir qubits con los dos estados de rotación de los núcleos de átomos que se encuentran en determinadas moléculas. La manipulación de los estados puede realizarse mediante resonancia magnética nuclear.

Por último otra forma de almacenar qubits es la utilización de semiconductores o superconductores. En ella se aprovecha el comportamiento de los electrones en estos materiales para crear y manipular qubits.

2.3. ENREDO CUÁNTICO O ESTADOS ENTANGLED, (PARADOJA EPR[5])

La capacidad computacional del procesamiento paralelo de la computación cuántica, es enormemente incrementado por el procesamiento masivamente en

[5] La correlacion de "Einstein-Podolsky-Rosen" (EPR) o "entanglemend", ha sido al menos en parte conocido desde los años 30 cuando fue discutido por Albert Einstein, Boris Podolsky y Nathan Rosen.

paralelo, debido a una interacción que ocurre durante algunas millonésimas de segundo. Este fenómeno de la mecánica cuántica es llamado enredo cuántico o entanglement (del ingles entangled).

Debido al enredo cuántico, dos partículas subatómicas, permanecen indefectiblemente relacionadas entre si, si han sido generadas en un mismo proceso. Por ejemplo la desintegración de un positrón y un electrón. Estas partículas forman subsistemas que no pueden describirse separadamente. Cuando una de las dos partículas sufre un cambio de estado, repercute en la otra. Esta característica se produce cuando se realiza una medición sobre una de las partículas.

En resumen, una definición simple de lo que es el enredo cuántico seria la siguiente: partículas o sistemas se dice que están en enredo o en entangled (en ingles) si la medida en una tiene efecto en la otra.

¿Es posible para un observador en el punto A (observador A), que ignora en general cuál es el estado cuántico de una partícula p, transmitir a un segundo observador en el punto B (observador B) información que le permita copiar dicho estado?. Esta respuesta es afirmativa gracias a la existencia de los estados enredados, que mantienen las correlaciones cuánticas entre sus partes aunque estas se encuentren espacialmente separadas.

Supongamos que en otro punto C un tercer observador (observador C) prepara una pareja auxiliar de partículas, a, b, del mismo tipo que aquella cuyo estado se quiere copiar, por ejemplo dos fotones o dos partículas de espín 1/2.

El siguiente estado enredado es un estado muy particular: es un estado puro, es decir, está máximamente determinado

$$\left|\Psi^{-}\right\rangle_{ab} = \frac{1}{\sqrt{2}}\left\{\left|0\right\rangle_a\left|1\right\rangle_b - \left|1\right\rangle_a\left|0\right\rangle_b\right\}$$

Donde los subíndices indican la partícula a la que se refiere el estado. Cada pareja de partículas tiene la misma probabilidad de estar en el estado $|0\rangle$ o en el $|1\rangle$, sin poder predecir cual de los resultados se obtendrá. Pero si sobre la partícula *a* se hace una medición y se encuentra en el estado $|0\rangle_a$, al medir sobre la partícula *b* ésta

se encontrará siempre en el estado $|1\rangle_b$, y recíprocamente: existe una correlación perfecta entre ambas partes del sistema, aunque ambas se encuentren arbitrariamente separadas. Experimentalmente se ha conseguido mantener el enredo hasta distancias de 10 km, pero teóricamente la separación podría ser de años luz.

Los problemas que se presentan son de tipo conceptual cuando se confunde correlación con relación causal, y se dice que las mediciones sobre la primera partícula "influyen" sobre los resultados para la segunda, o que al efectuar una medición sobre una de las partes del sistema hay una "transmisión instantánea" de información a la otra parte.

Esta correlación parece sugerir una "acción a distancia", pero los estados enredados no entran en conflicto con ningún principio físico esencial (causalidad, relatividad,...) ni con ningún resultado de tipo experimental, es decir, aparentemente es como si se produjese una transmisión instantánea de información, algo que contradice un principio físico básico: nada puede propagarse a mayor velocidad que la luz. Ahora bien, el experimento EPR (Einstein-Pdosky-Rosen) nunca ha fallado. Al contrario, es uno de los mecanismos más utilizados para la transmisión segura de información y constituye, de hecho, uno de los pilares más firmes de las nuevas tecnologías de criptografía cuántica. Veamos a continuación algunas aplicaciones cuánticas reales y potenciales del enredo.

2.4. TELE TRANSPORTACIÓN

Una de las aplicaciones más espectaculares del enredo es la tele transportación de un sistema cuántico desconocido de un lugar a otro del espacio sin necesidad de acarrear materia, sólo información. La tele transportación cuántica es descrita por Stean como la posibilidad de *"transmitir qubits sin enviar qubits"*. En la computación cuántica no es posible clonar, tampoco copiar, y mucho menos enviar qubits de un lugar a otro como se hace con los bits.

Si enviamos un qubit con un estado desconocido, el receptor no podrá leer su estado con certidumbre, cualquier intento de medida podría modificar el estado del qubit, por lo tanto se perdería su estado, imposibilitando su recuperación. La tele transportación cuántica resuelve este problema, esta se basa en los estados enredados para poder transmitir un qubit sin necesidad de enviarlo. El emisor y el

receptor poseen un par de qubits "enredados" (entangled). Entonces el qubit es transmitido desde el emisor, desaparece del emisor y el receptor tiene el qubit tele transportado. Este fenómeno es posible debido al efecto EPR. Veamos este fenómeno con más detalle.

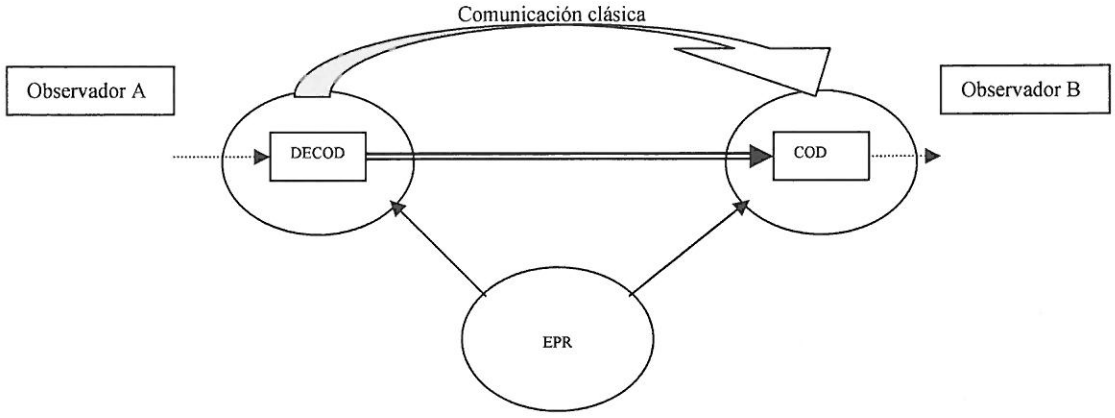


Figura 2.5. Tele transportación

Supongamos que *A* quiere enviar una partícula *p* al observador *B*. El observador *C* envía la partícula *a* y *b* a los observadores *A* y *B* respectivamente.

En el punto *A* el observador *A* reúne la partícula auxiliar *a* con su partícula de partida *p*. El conjunto de las tres partículas (*p*, *a* en el punto *A*, *b* en el punto *B*) está ahora en el estado

$$|\Phi\rangle_{pab} = |\Psi\rangle_p |\Psi^-\rangle_{ab} = \frac{\alpha}{\sqrt{2}}\left\{|0\rangle_p|0\rangle_a|1\rangle_b - |0\rangle_p|1\rangle_a|0\rangle_b\right\} + \frac{\beta}{\sqrt{2}}\left\{|1\rangle_p|0\rangle_a|1\rangle_b - |1\rangle_p|1\rangle_a|1\rangle_b\right\}$$

También puede escribirse como

$$|\Phi\rangle_{pab} = \frac{1}{2}|\Psi^-\rangle_{ab}\left\{-\alpha|0\rangle_b - \beta|1\rangle_b\right\} + \frac{1}{2}|\Psi^+\rangle_{pa}\left\{-\alpha|0\rangle_b + \beta|1\rangle_b\right\} +$$
$$+\frac{1}{2}|\Phi^-\rangle_{pa}\left\{\beta|0\rangle_b + \alpha|1\rangle_b\right\} + \frac{1}{2}|\Phi^+\rangle_{pa}\left\{-\beta|0\rangle_b + \alpha|1\rangle_b\right\}$$

Donde los $\left|\Psi^{\pm}\right\rangle_{pa}$, $\left|\Phi^{\pm}\right\rangle_{pa}$, son los llamados estados de Bell[6], mutuamente ortogonales:

$$\left|\Psi^{\pm}\right\rangle_{pa} = \frac{1}{\sqrt{2}}\left\{\left|0\right\rangle_p\left|1\right\rangle_a \pm \left|1\right\rangle_p\left|0\right\rangle_a\right\}$$

$$\left|\Phi^{\pm}\right\rangle_{pa} = \frac{1}{\sqrt{2}}\left\{\left|0\right\rangle_p\left|0\right\rangle_a \pm \left|1\right\rangle_p\left|1\right\rangle_a\right\}$$

En el estado $\left|\Phi\right\rangle_{pab}$ que estamos considerando, los cuatro posibles estados tienen la misma probabilidad, y cada uno tiene una correlación total con un estado concreto de la partícula *b*.

La correlación perfecta entre ambas partes enredadas del sistema siguen sin implicar ningún tipo de "influencia" de una sobre otra.

Tras efectuar su medición, *A* comunica a *B* (por teléfono, radio, correo electrónico...) el resultado obtenido. Al recibir esta información, que ha sido transmitida clásicamente a velocidad igual o menor que la de la luz, *B* procede en consecuencia. Por ejemplo, si *A* le informa que a obtenido como resultado el estado $\left|\Psi^{-}\right\rangle_{pa}$, no hace nada: ya tiene una copia del estado de partida. Si el observador *A* ha obtenido uno de los otros tres resultados posibles, *B* aplica a su partícula *b* la rotación correspondiente. En los cuatro casos, el estado de la partícula b acaba siendo $\left|\Psi\right\rangle_b$. Puesto que *B* no puede determinar cuál es el estado cuántico de su sistema individual, parece que se está pidiendo un acto de fe en la mecánica cuántica para aceptar el estado de partida, desconocido, ha sido correctamente copiado en otro estado también desconocido. Todo esto puede comprobarse experimentalmente, basta que un cuarto observador proporcione a *A* un gran número de sistemas individuales en el mismo estado puro $\left|\Psi\right\rangle_b$.

Debemos de tener en cuenta los siguientes puntos:

---

[6] El físico John S. Bell demostró que la paradoja EPR podía demostrarse científicamente. Más información en apéndice C

1.- Sin la comunicación clásica entre *A* y *B*, éste no sabría qué transformación aplicar a su partícula *b*, y por tanto no podría preparar el estado $|\Psi\rangle_b$. El proceso de tele transportación requiere un canal dual: un par de partículas con el tipo de correlación cuántica (*EPR*), y una comunicación clásica.

2.- El observador *A* informa a *B*, no del estado de la partícula *p*, que en general desconoce, sino del resultado de una medición sobre el par *pa*.

3.- No hace falta que *A* conozca la ubicación de *B* en el momento en que éste reproduce el estado inicial: *A* puede radiar el resultado de su medición sobre el par *pa*, de forma que *B* capte el mensaje allí donde se encuentre; sólo se necesita que *B* haya recibido la partícula *b* del par *EPR* y que no se haya roto la correlación entre los miembros del par. La tele transportación no es un proceso direccional.

4.- Lo que se tele transporta es el estado cuántico de la partícula *p*, no la propia partícula, que no abandona el punto *A*. Lo que se transmite (clásicamente) entre los puntos *A* y *B* es el resultado de la medición realizada por el observador *A* sobre el par *pa*.

5.- El observador *B* puede obtener una única copia del estado original, y que tras la medición de los estados de Bell sobre el par *pa* la partícula *p* no se encuentra ya en su estado de partida. No hay pues duplicación del estado cuántico, prohibida por el teorema de la no-clonación, sino sustitución del estado en un punto por el mismo estado en otro punto.

Esto abre el camino para las telecomunicaciones cuánticas, aunque tengamos el problema de la decoherencia. Al no poder clonar los qubits, no es posible amplificar la señal cuántica, la cual se degrada tras una decena de kilómetros, aun utilizando fibra óptica. Este problema se podría solucionar utilizando estaciones de tele transportación intermedias o bien manteniendo canales cuánticos por largo tiempo, entonces el observador *A* podría tele transportar estados al observador *B*, a distancias relativamente grandes, sin preocuparse de la atenuación de la señal.

2.5. PARALELISMO CUÁNTICO

La superposición cuántica permite un paralelismo exponencial o paralelismo cuántico en el cálculo, mediante el uso de las puertas lógicas de qubits. Los qubits, a diferencia de los bits, pueden existir en un estado de superposición, representado por $a|0\rangle + b|1\rangle$, donde a y b son números complejos que satisfacen la relación $|a|^2 + |b|^2 = 1$.

Dada una puerta lógica de un qubit *f*, esrta transforma el estado $|a|$ en el estado $|f(x)|$, cuando el qubit de entrada tiene en el estado

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

una superposición igual de $|0\rangle$ y $|1\rangle$.

Por linealidad de la mecánica cuántica, la compuerta lógica *f* transforma el estado del qubit a

$$\frac{1}{\sqrt{2}}|f(0)\rangle + \frac{1}{\sqrt{2}}|f(1)\rangle$$

El estado resultante el la superposición de los dos valores de salida, siendo *f* evaluado para los dos valores de salida, siendo *f* evaluado para los dos valores de entrada en paralelo.

Para una puerta lógica *g* de dos qubits, que tiene dos qubits de entrada en superposición de $|0\rangle$ y $|1\rangle$, tendríamos una superposición de cuatro estados

$$c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle$$

La puerta lógica *g* transforma el estado de entrada a

$$c_0|g(00)\rangle + c_1|g(01)\rangle + c_2|g(10)\rangle + c_3|g(11)\rangle$$

así *g* es evaluado en un solo paso para cuatro valores de entrada.

En una puerta lógica *h* de tres qubits, se tienen tres qubits de entrada en superposición de $|0\rangle$ y $|1\rangle$, juntos hacen una superposición de ocho estados, que son evaluados en paralelo. Por cada qubits adicional la cantidad de estados se duplica.

2.6. CRIPTOGRAFÍA CUÁNTICA

La criptografía es la ciencia matemática de las comunicaciones secretas, tiene tiene una larga y distinguida historia de uso militar y diplomático que se remonta a los antiguos griegos. Fue un elemento importante y decisivo durante la segunda guerra mundial. Hoy en día se su uso es muy común y necesario, para brindar seguridad en las transacciones comerciales, comunicaciones y privacidad; que se llevan a cabo mediante internet.

Para encriptar un mensaje secreto *M* necesitamos: una clave *K*, conocida por el remitente y el destinatario, y un algoritmo criptográfico *E* que asigna un criptograma $C=E_K(M)$ a *M* por medio de *K*. El proceso de desencriptación consiste en aplicar el algoritmo inverso.

Veamos que ocurre para pares enredados electrón-positrón: tanto el observador *A* como el observador *B* pueden elegir la dirección de sus espines ↑ ó →. Tras medir *n* pares, los observadores hacen publica la elección de dirección pero no el resultado de la medida que puede ser $|\uparrow\rangle = 1$ ó $|\downarrow\rangle = 0$. Habrán coincidido en la elección de dirección alrededor de *n*/2 veces para los cuales las respuestas están anti-correlacionadas. Lo que se hace es quedarse con las respuestas que son 0 y 1 de esas *n*/2 coincidencias y construir la clave en binario, pasándola luego a decimal.

Las respuestas estarán efectivamente anti-correlacionadas, si y sólo si, una tercera persona no ha intentado "tocar" el par electrón-positrón, esto puede verificarse sacrificando parte de la clave *K*. La seguridad de este algoritmo reside en que la observación de una tercera persona, destruiría el enredo mecánico-cuántico, es decir, las telecomunicaciones cuánticas detectan la presencia de "fisgones".

Pero por ahora, desgraciadamente, la criptografía por enredo sale cara, por lo que se usa el protocolo clásico RSA de la "clave publica" que consiste en lo siguiente: El observador *A* hace publica su clave, consistente en un par de números enteros grandes (*s, c*) para que cualquiera pueda mandarle mensajes *M* encriptados de la siguiente forma: *M* es una secuencia larga de números binarios, la versión encriptada será $C=M^S \bmod c$. Para desencriptar el mensaje, el observador *A* usa la formula $M = C^t \bmod c$, donde $t=t(s,p,q)$ se deduce fácilmente de s y los factores primos *p* y *q* de *c*. Estos factores primos , *p* y *q* se obtienen resolviendo las ecuaciones $st \equiv 1 \bmod (p-1)$ y $st \equiv 1 \bmod (q-1)$.

Cualquiera que intente desencriptar el mensaje *M* a partir de *C* tiene que calcular *t*, para lo cual tiene que descomponer antes *c* en producto de primos $c=pq$. Para hacernos una idea , si *c* es un número con 50 dígitos, un computador capaz de hacer 10^{10} divisiones por segundo tardaría del orden de 10^{15} segundos (¡más de 31.7 millones de años!) en encontrar *p* y *q* por lo que es prácticamente imposible descifrar el código, cosa que no ocurriría con un ordenador cuántico que efectuase $2^{(10^{10})}$ operaciones por segundo, terminando así con la criptografía clásica.

Esencialmente, el problema de factorizar un número *c* se reduce a encontrar el periodo *r* de la función $f(x)=a^x \bmod c$, donde *a* es un número cualquiera entre 0 y *c*. El *algoritmo de Shor* describe como determinar *r* eficazmente con un computador cuántico.

Una superposición entre todos los números *x* comprendidos entre 0 y c^2 se colocan en un registro de bits cuánticos. Todos los resultados de $f(x)=a^x \bmod c$ se calculan simultáneamente usando el paralelismo cuántico y son almacenados en un segundo registro.

Si el primer registro se mide y se encuentre el valor *x*, una medición en el segundo registro producirá $a^x \bmod c$, justo como en la paradoja EPR. Sin embargo, tales mediciones no pueden llevarse ha cabo, ya que esto destruiría la superposición. Para evitarlo se intenta conocer la frecuencia con que la secuencia de restos se repite, de esta forma se puede determinar *r*.

Realizando una transformada de Fourier cuántica sobre el primero de los registros (aunque la transformada de Fourier cuántica esta basada en la transformada de Fourier clásica esta es eficiente en la computación cuántica) el enredo entre los dos registros permite que la periodicidad del segundo se refleje en el primero.

En el primer registro se almacena una superposición que es esencialmente sólo múltiplos de c^2/r. Se puede medir el registro para obtener uno de estos múltiplos, del cual con alta probabilidad podremos deducir r.

CAPITULO 3

3. DISPOSITIVOS DE ELECTRÓN ÚNICO

Una vez vista la forma de almacenamiento de los qubits y sus principales características, veamos con que dispositivos podemos construir puertas lógicas para los qubits y que operaciones lógicas podemos realizar con estos dispositivos. Para ello se van ha tratar los dispositivos de electrón-único.(e-u)

3.1. PUERTAS CUÁNTICAS

El concepto básico del e-u es permitir a un pequeño conductor, llamado isla, adquirir un electrón externo adicional. Inicialmente, la isla es neutral, es decir, tiene exactamente el mismo número de electrones que de protones en su red cristalina. En este estado la isla no genera ningún campo eléctrico apreciable más allá de sus limites, y una débil fuerza externa puede atraer un electrón externo. En la mayoría de los dispositivos de e-u, el electrón es inyectado del exterior por un túnel a través de una barrera de energía creada por una delgada capa de aislante.

Ahora la carga Q de la isla es (-*e*), y el campo eléctrico resultante ε repele otros electrones que podrían ser añadidos (fig 3.1.)

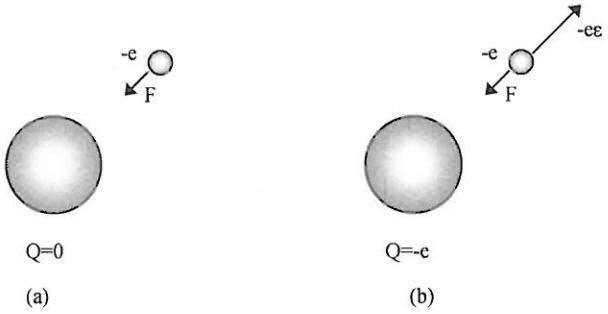


Figura 3.1. Concepto básico de electrón-único (a) antes de (b) después de la adición del electrón. Y la descompensación de carga de e-u creado por el campo ε, el cual dificulta la adición de siguientes electrones

Un dispositivo basado en e-u es la caja de e-u (fig. 3.2. (a)). El dispositivo consiste en una isla separada de un electrodo largo (fuente) por un túnel barrera. Un campo eléctrico externo puede ser amplificado por la isla usando otro electrodo (puerta) separado de la isla por un grueso aislante. La carga electrostática del sistema puede presentarse como:

$$W = \frac{Q^2}{2C_\Sigma} + \frac{C_0}{C_\Sigma} QU + const \qquad (1)$$

Donde Q=-ne es la carga de la isla (n es el numero de electrones no compensados), C_0 es la capacidad entre la isla y la puerta, mientras que C_Σ es la capacidad de la isla, incluida C_0. La expresión (6) se puede rescribir como:

$$W = \frac{(ne - Q_e)^2}{2C_\Sigma} + const$$

Donde el parámetro Q_e está definido como Q_e=UC_0 usualmente llamado carga externa. Cuando el campo eléctrico esta bien localizado entre la puerta y la isla; entonces (-Q_e) es justo la carga de polarización, la cual es atrapada por el campo de la puerta y así, saca del equilibrio al túnel de enlace (fig. 3.2. (b)).

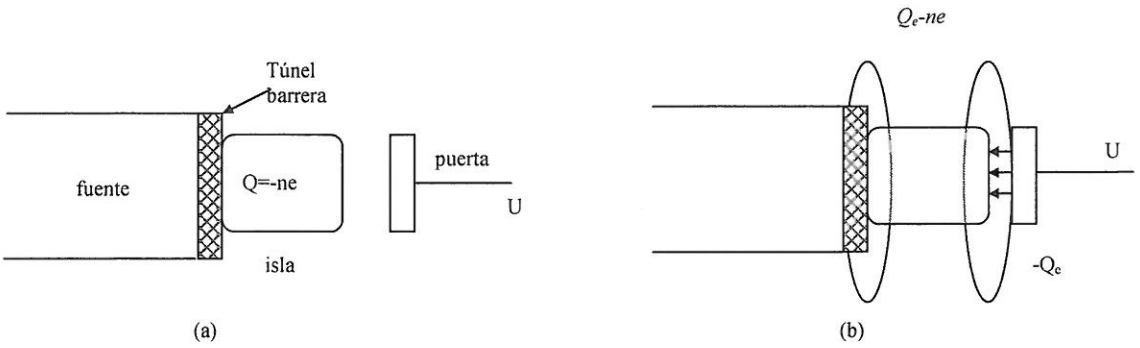


Figura 3.2.. Caja de electrón-único (a) esquema: (b) visualización de la carga externa

Veamos como podemos utilizar la caja de electrón-único para poder diseñar un transistor de electrón-único, y a partir de este, poder diseñar puertas lógicas.

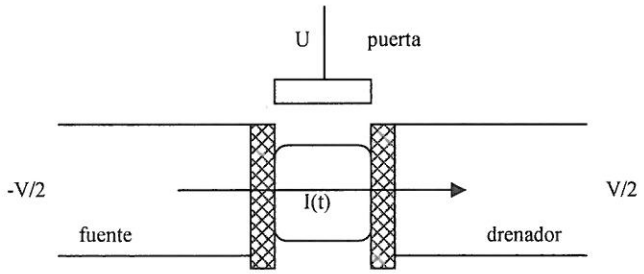


Figura 3.3. Esquema del transistor de electrón-único

Dividiendo el túnel de enlace de la caja de electrón-único (fig.3.2. (a)) y aplicando una corriente continua *V* entre las dos partes externas del electrodo (fig. 3.3.), obtenemos como resultado el transistor de electrón-único. El dispositivo recuerda a un MOSFET común pero con una isla conductora entre los dos túneles barrera, en vez de un canal de inversión.

La expresión para la energía electrostática *W* del sistema es :

$$W = \frac{(ne - Q_e^2)}{2C_\Sigma} - \frac{eV[n_1C_2 + n_2C_1]}{C_\Sigma} + const$$

Esta expresión es una generalización de (1). Donde n_1 y n_2 son el número de electrones que pasan a través de los túneles barrera uno y dos respectivamente, así $n=n_1-n_2$, mientras que ahora la capacidad total C_Σ es la suma de C_0, C_1, C_2 y cualquier

capacidad parásita que la isla pueda tener. La carga externa debe ser lo más parecida al voltaje *U* de la puerta.

Estos transistores, al igual que los transistores comunes, pueden funcionar como puertas lógicas. Un electrón se sitúa en una isla, y a través de la carga electrostática se determina el sentido que tiene el espín confinado en la isla. La versión más simple de puerta lógica es usando tres islas separadas por dos túneles barrera (fig.3.4. (a), (b)), complicándose aún más al usar más islas (fig.3.4. (c)).

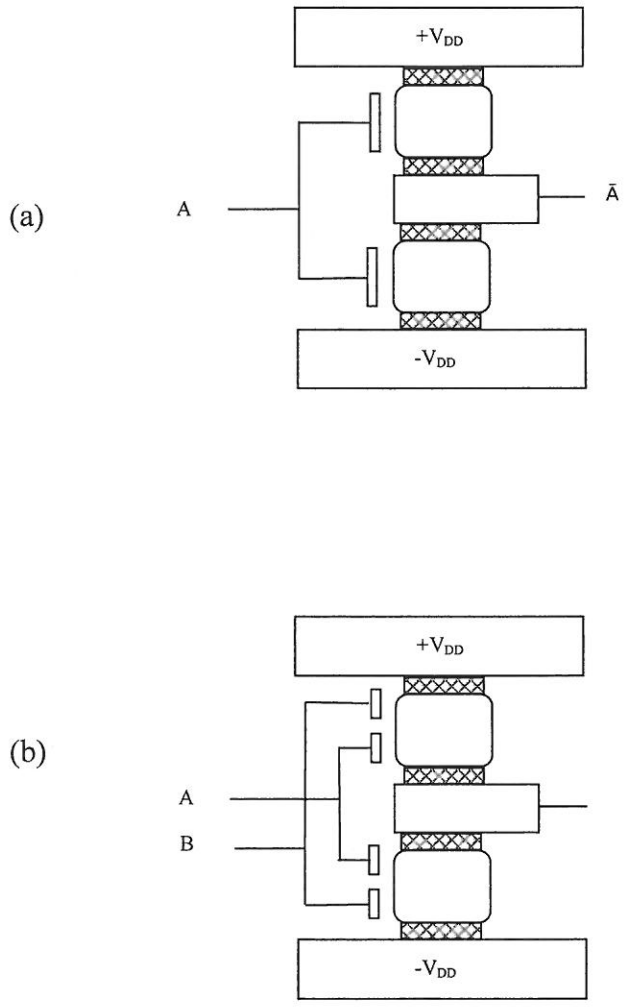


Figura 3.4.. Puertas lógicas usando transistores de electrón-único (a) inversor, (b) XOR.

(c)

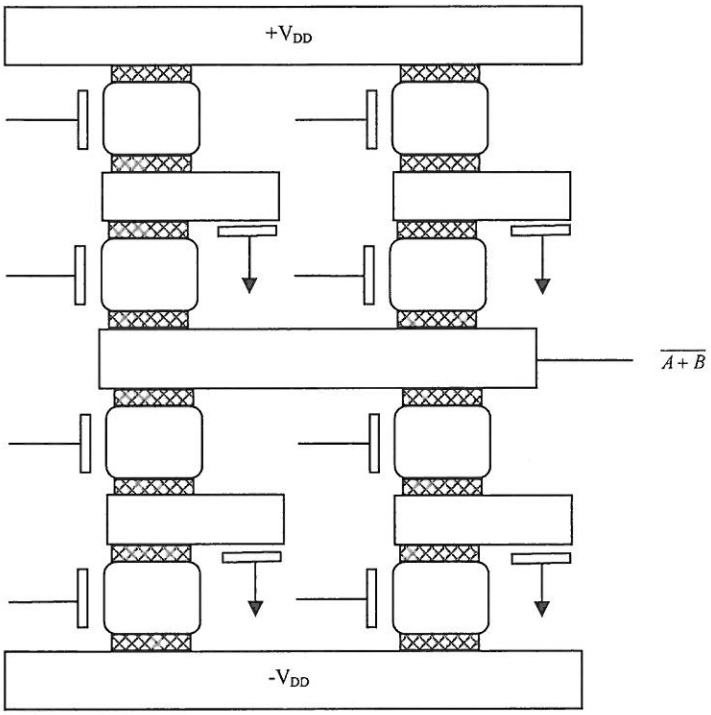
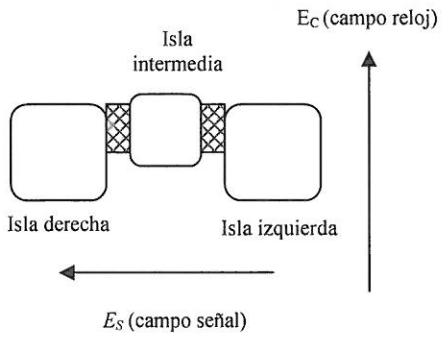
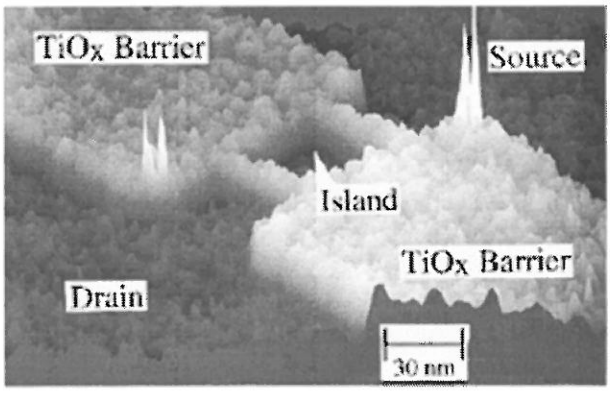


Figura 3.4.. Puertas lógicas usando transistores de electrón-único (c) NOR/NAND

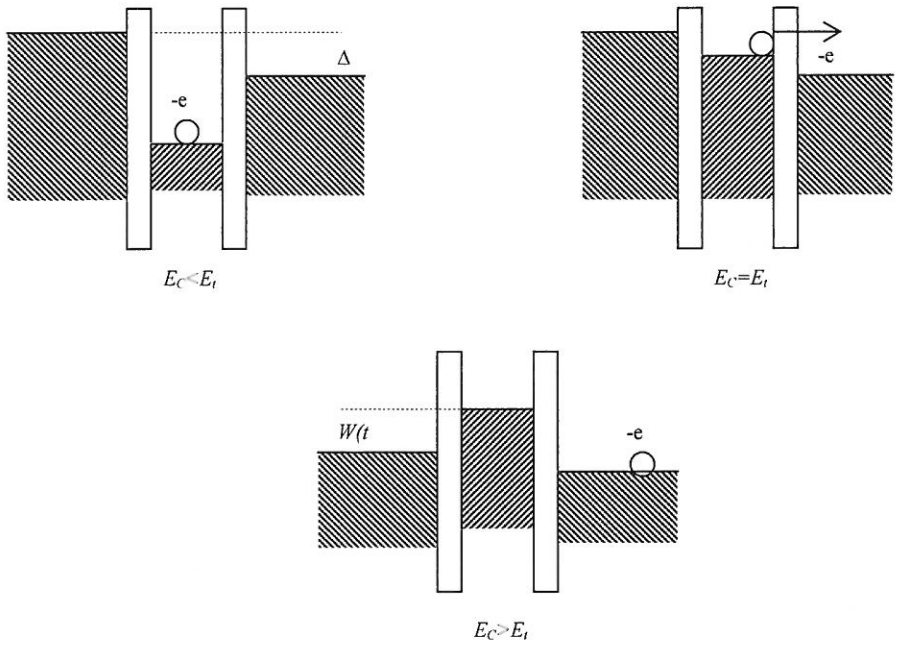
Un campo eléctrico periódico "reloj" guarda un electrón extra en la isla central en un periodo de reloj. En ese instante el electrón es transferido a una de las islas laterales cambiando el signo de ΔW de negativo a positivo. Si el sistema fuese completamente simétrico, el cambio entre las dos islas laterales seria aleatorio. No obstante un pequeño campo adicional E_S generado por un dispositivo cercano, puede determinar la dirección del electrón para pasar a una isla o a otra (fig.3.5. (a)).



(a)



(b)



(c)

Figura 3.5. Colocación de parámetros (a) esquema. (b) Transistor de electrón único, fotografiado con un microscopio de efecto túnel, (c) diagramas de energías para tres estados del campo reloj E_C *(t)*

Una vez creada la barrera *W(t)*, favorecida por el campo reloj, el electrón queda atrapado en una de las islas laterales y el campo E_S generado por un dispositivo cercano deja de actuar (fig.3.5. (b)). Ahora el circuito puede generar un campo E_S para circuitos adyacentes. El signo de este campo, es decir, el momento dipolar eléctrico del circuito, presenta un bit de información.

Visto el funcionamiento del transistor, veamos que operaciones podemos realizar con las puertas cuánticas elementales, hay que tener en cuenta que las transformaciones han de ser reversibles, ya que entonces no seria posible una realización física :

Para un qubits:

$\|0\rangle \rightarrow \|0\rangle$ $\|1\rangle \rightarrow \|1\rangle$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Identidad (I)
$\|0\rangle \rightarrow \|1\rangle$ $\|1\rangle \rightarrow \|0\rangle$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	NOT (σ_x) (X)
$\|0\rangle \rightarrow \|1\rangle$ $\|1\rangle \rightarrow -\|0\rangle$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	Combinación de corrimiento de fase y negación Y=ZX ($i\sigma_y$) (Y)
$\|0\rangle \rightarrow \|0\rangle$ $\|1\rangle \rightarrow -\|1\rangle$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	Desplazamiento de fase (σ_z) (Z)
$\|0\rangle \rightarrow \|+\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$ $\|1\rangle \rightarrow \|-\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$	$H \rightarrow \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Transformada Walsh-Hadamand (H)

Estas puertas forman uno de los más pequeños grupos de la computación cuántica. La tecnología física cuántica puede implementar estas puertas eficientemente. Todas excepto la CNOT operan en un simple qubit; la puerta CNOT opera en dos qubits.

Una compuerta de dos qubits en especial interesante, es la conocida como "U controlada",

$$|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$$

son operadores actuando sobre dos qubits, donde *I* es la operación de identidad sobre un qubit. El estado del qubit *U* es controlado mediante el estado del qubit *I*. Por ejemplo el NOT controlado (CNOT) es : $|00\rangle \rightarrow |00\rangle$; $|01\rangle \rightarrow |01\rangle$; $|10\rangle \rightarrow |11\rangle$; $|11\rangle \rightarrow |10\rangle$

Esta última puerta, Walsh-Hadamand, es importante, ya que genera una superposición de todos los 2^n posibles estados (con igual peso), es decir:

$$H|0\rangle = |+\rangle$$

$$H \otimes H \otimes \ldots \otimes H|00\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

Para dos qubits:

NOT controlado ó CNOT : cambia el segundo bit si y solamente sí el primer bit es un 1.

$$|00\rangle \rightarrow |00\rangle$$
$$|01\rangle \rightarrow |01\rangle$$
$$|10\rangle \rightarrow |11\rangle$$
$$|11\rangle \rightarrow |10\rangle$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

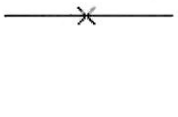
La transformación CNOT es unitaria CNOT*=CNOY y CNOT*CNOT=I. La puerta CNOT no puede ser descompuesta en el producto tensorial de dos operaciones sobre un solo qubit

$$CNOT = |0\rangle\langle 0| \otimes \Gamma + |1\rangle\langle 1| \otimes X$$

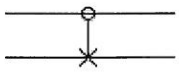
Se puede definir una " controlled υ" Cυ tal que:

$$C\upsilon = |0\rangle\langle 0| \otimes \Gamma + |1\rangle\langle 1| \otimes \upsilon$$

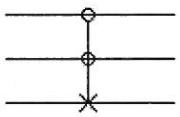
La puerta CNOT es normalmente representada como se muestra a continuación:



a	a´
0	1
1	0



a	b	a´	b´
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0



a	b	c	a´	b´	c´
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Figura 3.6. Representación esquemática de las puertas reversibles: NOT, CNOT, CCNOT o puerta Toffoli. La concatenación de la puerta CNOT con la puerta CCNOT da como resultado un "sumador cuántico", donde se almacena el acarreo.

Para tres qubits o puertas Toffoli:

CCNOT: Cambia el último bit a condición de que los dos primeros sean 1:

$$|a,b,c\rangle \xrightarrow{T} |a,b,c \oplus ab\rangle$$

$$T = |0\rangle\langle 0| \otimes \Gamma \otimes \Gamma + |1\rangle\langle 1| \otimes CNOT$$

Para una puerta NOT tenemos: *T(1,1,x)=(1,1,NOT(x)).*

Y para una puerta AND: *T(x,y,0)=(x,y,AND(x,y)).*

Para terminar se va ha ver una aplicación de la puerta CNOT, la codificación densa.

3.2. OPERACIONES CON PUERTAS LÓGICAS. CODIFICACIÓN DENSA

Básicamente, la codificación densa usa un qubit junto con estados enredados (que hacen de canal) para codificar dos bits clásicos.

El estado enredado se puede distribuir con antelación, por lo que un qubit tiene que ser físicamente transmitido. Un qubit parece valer por dos bits clásicos.

Supongamos que el observador *A* y el observador *B* quieren comunicarse, y comparten un estado enredado :

$$|\Psi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Siendo el primer bit del observador *A* y el segundo del observador *B*, cada observador sólo podrá operar con el suyo.

El observador *A* recibe dos bits clásicos, los cuales tienen un valor entre 0, 1, 2 y 3. Dependiendo del valor recibido se harán las siguientes operaciones I, X, Z, Y:

$$0 \quad I \otimes I|\Psi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |\Psi_0\rangle$$

$$1 \quad X \otimes I|\Psi_0\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) \qquad |\Psi_1\rangle$$

$$2 \quad Z \otimes I|\Psi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \qquad |\Psi_2\rangle$$

$$3 \quad Y \otimes I|\Psi_0\rangle = \frac{1}{\sqrt{2}}(-|00\rangle + |01\rangle) \qquad |\Psi_3\rangle$$

Hay que notar que *A* sólo opera en su qubit y se lo envía a *B,* este aplica un CNOT a cada estado obteniendo:

$$CNOT|\Psi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$$

$$CNOT|\Psi_1\rangle = \frac{1}{\sqrt{2}}(|11\rangle + |01\rangle) = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle)|1\rangle$$

$$CNOT|\Psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle$$

$$CNOT|\Psi_3\rangle = \frac{1}{\sqrt{2}}(-|11\rangle + |01\rangle) = \frac{1}{\sqrt{2}}(-|1\rangle + |0\rangle)|1\rangle$$

Así el observador *B*, puede medir el segundo bit sin perturbar el primero. Si el resultado es: $\begin{cases} |0\rangle \rightarrow 0, 2 \\ |1\rangle \rightarrow 1, 3 \end{cases}$ Nos podríamos preguntar: ¿Cómo distinguir cada caso? Para ello usamos la puerta lógica Walsh-Hadamand para un bit:

$$(H \otimes I)CNOT \begin{cases} \Psi_0|00\rangle & 0 \\ \Psi_1|01\rangle & 1 \\ \Psi_2|10\rangle & 2 \\ \Psi_3|11\rangle & 3 \end{cases}$$

CAPITULO 4

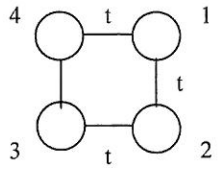
4.REALIZACIÓN DE SISTEMAS DIGITALES CON DISPOSITIVOS DE ELECTRÓN-ÚNICO

En este apartado vamos a abordar la computación cuántica usando el concepto de autómata celular de punto cuántico (quantum-dot cellular automata QCA). Las celdas de puntos cuánticos se cargan pudiendo así, codificar información. Las interacciones entre las celdas son culómbicas y las interconexiones entre estas se deben a la interacción física entre celda y celda.

4.1. QCA

Vamos a ver de forma cualitativa como trabajan las celdas QCA y como interaccionan para formar una arquitectura computacional.

Un celda de cuatros puntos QCA se muestra en la figura 4.1.



(a)

P=+1 P=-1

(b)

Figura 4.1. (a) Geometría de la celda. La energía por efecto túnel entre dos posiciones vecinas está designada por t, mientras que la distancia entre estas es de a. (b) Repulsión culombica entre electrones que ocupan posiciones opuestas en la celda dando dos estados de polarización P=+1 y P=-1

La celda se constituye por cuatro puntos cuánticos posicionados cada uno en las esquinas de un cuadrado fig. 4.1. (a). La celda contiene dos electrones móviles extra, los cuales pueden pasar de celda a celda por medio del efecto túnel. Si el potencial barrera entre las celdas es suficientemente alto, los electrones pueden ser localizados de forma individual en uno de los cuatro puntos cuánticos que tiene la celda. La repulsión culombiana entre los electrones servirá para que estos ocupen posiciones opuestas como se muestra en la fig.4.1. (b). Para una celda podemos tener dos posiciones energéticas equivalentes dependiendo de la posición de los electrones, estas dos posiciones se denotan por P=+1 y P=-1.

El termino de celda polarizada se refiere a estas dos posiciones de carga. No implica un momento dipolar en la celda. La polarización de la celda puede ser codificada en forma binaria, así pues P=+1 representa un uno y P=-1 representa un cero en binario.

Si tenemos dos celdas próximas entre si, los estados de polarización no son energéticamente equivalentes.

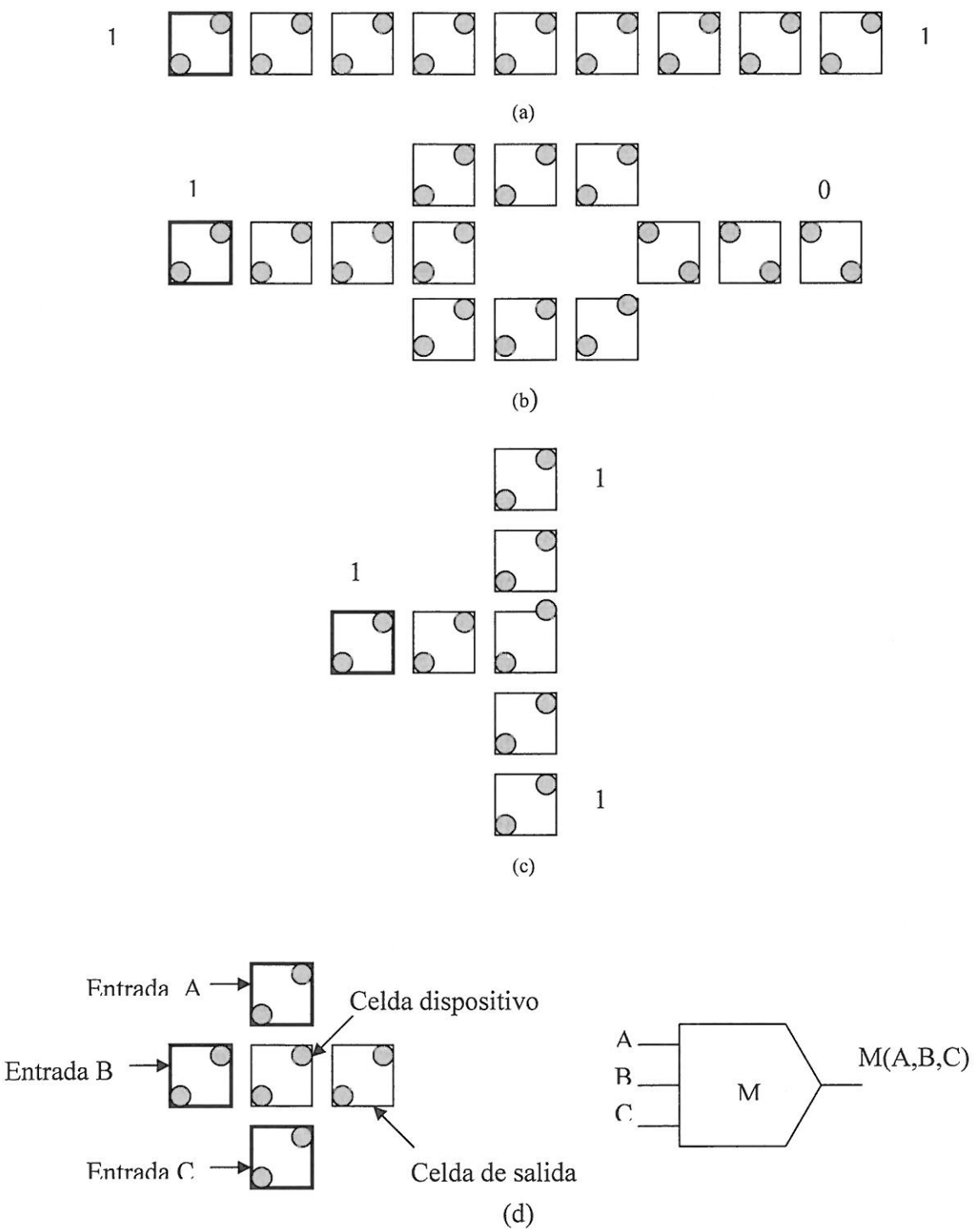


Figura 4.2. Dispositivos fundamentales QCA: (a) Cable binario que permite la transmisión de un punto a otro, (b) inversor de señal (c) Bifurcación para poder transmitir a dos o más puntos una señal y (d) elemento lógico fundamental de una cadena QCA y su símbolo.

En la fig. 4.2.(a) se muestra un cable de QCA. La celda más a la izquierda es fijada con una polarización determinada fijando la entrada. La configuración de la celda de entrada pasa por cada una de las celdas libres, quedando las celdas de la misma forma que la celda de entrada, esto se puede considerar como una transmisión.

Las celdas dispuestas en diagonal respecto a otras, tienden a ser antilineales. Esta propiedad es empleada para construir inversores como se muestra en la fig. 4.2. (b). La antilinealidad de las celdas puede también ser vista como una consecuencia de la mutua repulsión entre los electrones y la geometría de la celda. La separación de una señal se muestra en la fig. 4.2. (c).

En la fig.4.2. (d) se muestra el dispositivo lógico fundamental de una cadena QCA. Con tres entradas, a partir de este circuito se pueden construir circuitos más complejos. La celda de control, llamada celda dispositivo, tiene tres entradas: A, B y C. La celda dispositivo tiene el estado de energía más bajo, si esta adquiere la polarización de las tres celdas de entrada , la celda de salida podrá conectarse a otras celdas de salida.

La diferencia entre celdas de entrada y salida en este dispositivo y en cadenas de QCA en general es que las entradas están fijas mientras que las salidas pueden cambiar.

Es posible conseguir otras puertas lógicas fijando una de sus tres entradas a un uno o un cero. Si fijamos una de las entradas con un uno, la función OR se realiza en las otras dos entradas. Si se fija con un cero la función AND se realiza en las otras dos entradas. También podríamos implementar una puerta lógica programable AND/OR. Combinada con un inversor la puerta AND/OR genera circuitos lógicos más completos.

Un ejemplo más complejo de cadenas QCA puede ser la implementación de un sumador de un bit. El esquema de un sumador lógico formado por inversores y puertas se muestra en la fig. 4.3. (a).

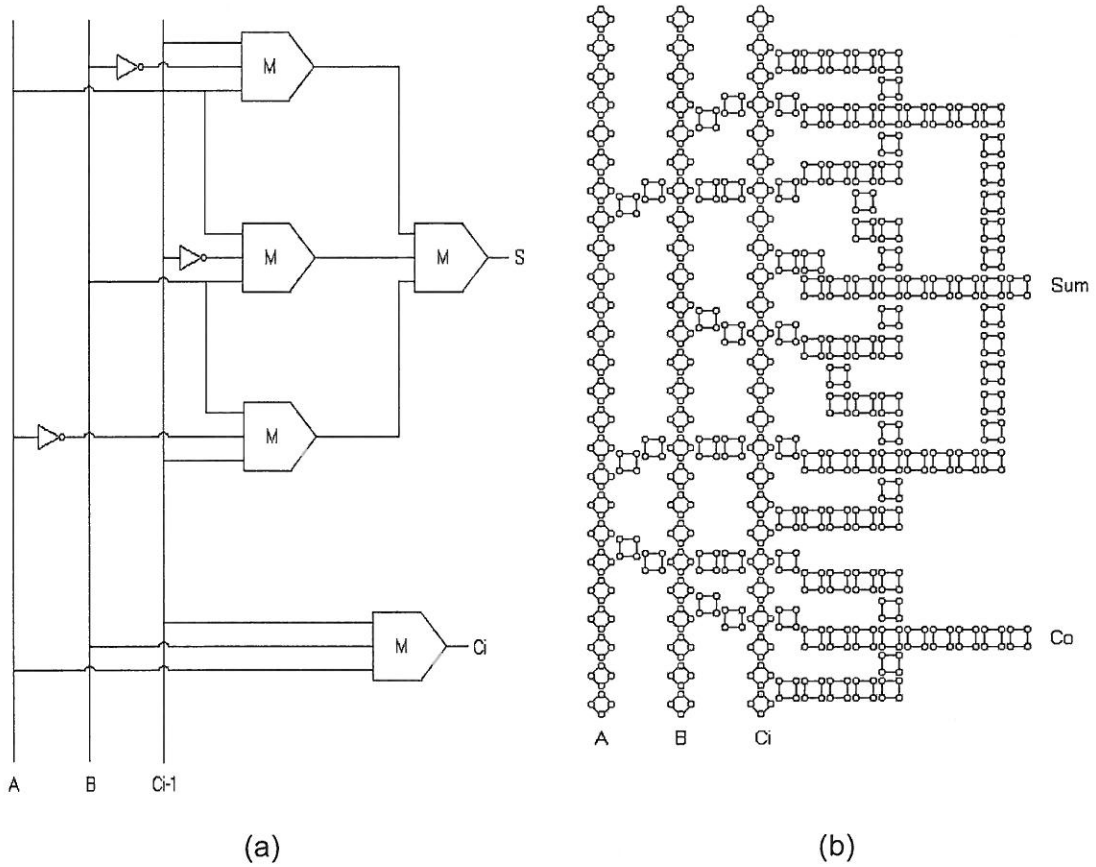


Figura 4.3. (a) Esquema con puertas lógicas para implementar un sumador QCA. Esta constituido por cinco puertas lógicas y tres inversores. (b) Esquema equivalente del sumador construido con celdas QCA.

En la fig. 4.3. (b) se muestra el diseño del sumador implementado con celdas QCA. Las puertas lógicas M de la fig. 4.3. (a) se implementan en la fig. 4.3. (b) como la unión de tres entradas y una salida.

Una utilidad importante que presentan las celdas QCA es la habilidad de atravesar cables en el plano. Mientras que para atravesar cables en el plano con la tecnología convencional se necesitan puentes, usando cables QCA podemos ser capaces de transmitir las señales sin necesidad de crear puentes, tan sólo rotando las celdas como se muestra en los cables A, B y Ci de la fig. 4.3. (b). Los cables diagonalmente orientados alternan su polarización hasta el siguiente cable (una inversión en cadena).

Esta característica se debe a la interacción culombiana y a la simetría presente en las celdas cargadas. La alternación de la polarización en los cables permite una fácil medida de la señal y su complemento.

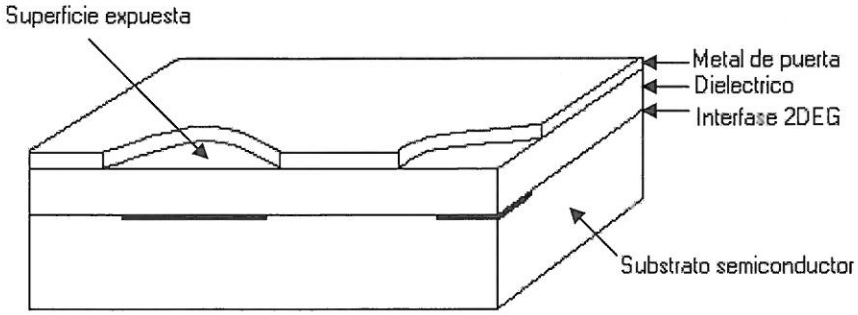
Veamos a continuación como son la entrada y la salida para una cadena QCA. Colocar un cable de entrada requiere introducir el estado de la primera celda en el cable. Esto puede hacerse de forma sencilla a trabes de dos cargas próximas que repelen electrones de un punto cuántico y lo atraen de otro. En la fabricación de puntos cuánticos en semiconductores, esto a llegado a ser una técnica estándar experimental llamada *plunger electrode* (émbolo de electrodos) para cambiar la posición de un electrón en un punto cuántico.

Leer un estado de salida resulta más complicado. Debemos de intuir el estado de carga de un punto cuántico sin considerar la alteración del estado de carga. En resumen, la entrada y salida sólo actúan como bordes de una cadena, no es necesario el flujo de información o energía en el interior de las celdas.

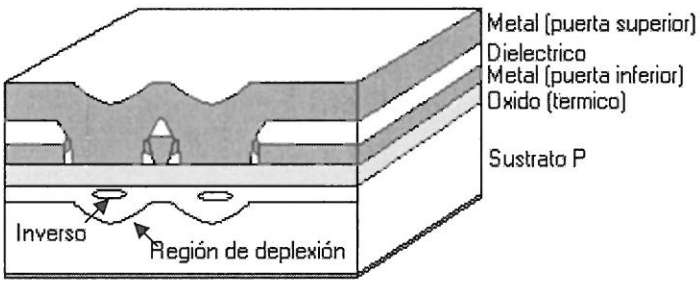
4.2. POSIBLE IMPLEMENTACIÓN QCA

La mayoría de los trabajos sobre puntos cuánticos son enfocados sobre puntos en semiconductores. Una técnica bien desarrollada usa metal de puerta para modelar la energía de los electrones confinados en un gas de dos dimensiones (2-D electrón gas (2DEG)) en una interfase. Esta interfase puede ser una heterofunción III-V o una interfase Si-SiO_2. Esta se muestra en la fig. 4.4. (a).

Un perfil más exacto es usar puertas dobles, esta configuración se muestra en la fig. 4.4. (b).



(a)



(b)

Figura 4.4. (a) Posible implementación física de una celda QCA. El metal de puerta introduce campos eléctricos y agota los electrones en la capa 2DEG que hay entre el dieléctrico y el sustrato. Los puntos cuánticos se forman en la superficie expuesta. Esta estructura puede fabricarse usando GaAs o una combinación de silicio y oxido de silicio. (b) Posible realización de celda QCA en un sistema de silicio. Se usan dos metales de puerta para controlar la ocupación de los electrones en el sustrato P. El metal de puerta inferior se usa para agotar los huecos cerca de la superficie del sustrato en todos los lugares donde los puntos cuánticos no son deseados. La puerta superior entonces invierte el sustrato P en lugares particulares guiando la creación de puntos cuánticos. Esta combinación de deplexión e inversión nos proporciona un excelente control sobre el tamaño y la posición de los puntos.

Usando estas aproximaciones podemos fabricar un punto cuántico lo bastante pequeño como para contener un único electrón. Si los puntos cuánticos tienen gran cantidad de electrones, estos pueden ser tratados como mayas metálicas cargadas electrónicamente, estas mayas se usan como capacidades de enlace.

Muchos trabajos actualmente consisten en crear patrones en materiales los cuales puedan soportar puntos cuánticos, pero todas las aproximaciones litográficas necesitan un ajuste debido al problema de la desviación de carga.

Un posible candidato para una buena generación de puntos cuánticos es la realización molecular con moléculas de carboxilato. La implementación molecular presenta grandes cambios, particularmente en la entrada y la salida. Pero tiene la ventaja de que son muy uniformes y tiene un gran número de ellas. Experimentos eléctricos y ópticos sobre el carboxilato son posibles gracias a las largas cadenas que se pueden sintetizar.

Actualmente IBM a sido la primera empresa en conseguir crear un microchip molecular, reduciendo considerablemente su tamaño y aumentando su velocidad en comparación con los microchips de silicio los cuales tienen limitaciones físicas cuando se les reduce de tamaño.

4.4. OTRAS POSIBLES IMPLEMENTACIONES

El qubit no puede ser construido a partir del transistor ya que este es un elemento que sólo funciona en las computadoras actuales; más bien se deben utilizar partículas o sistemas de partículas que manifiesten el fenómeno de la interferencia cuántica.

En este sentido, se han hecho diversos experimentos a parte del mencionado anteriormente sobre puntos cuánticos:

Moléculas Líquidas: En esta técnica se utilizan grupos de moléculas, en lugar de una partícula elemental. Al ser sometidos a un campo magnético, los núcleos de las moléculas giran en una determinada dirección que puede ser utilizada para describir su estado (giro hacia arriba = "uno", giro hacia abajo = "cero"). Mediante señales de radiofrecuencia, el giro puede modificarse.

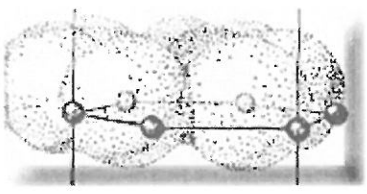


Figura 4.5. Grupo de moléculas

En este sentido, el computador cuántico vendría a estar representado por las moléculas, y los qubits por los núcleos. ¡Se piensa que la molécula de la cafeína sería un buen computador!

Átomos de Cesio: Recientemente (Marzo del 2000), se han hecho pruebas en las que en lugar de utilizar varios qubits, se utiliza un solo átomo capaz de adoptar varios estados de energía para guardar y recuperar información. También se utilizan aquí pulsos de láser para obtener la superposición.

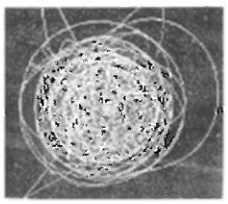


Figura 4.6. Nube electrónica del átomo de cesio

En abril del 2000 se ha propuesto un computador cuántico escalable que podría contener más de 10 qubits, utilizando iones (átomos con carga eléctrica).

CAPITULO 5

5. ARQUITECTURA DE UNA COMPUTADORA CUÁNTICA

Una vez vista la posible implementación de puertas cuánticas mediante celdas QCA, vamos a ver como seria la posible implementación de una computadora cuántica.

La arquitectura de una computadora cuántica es similar a las computadoras tradicionales, con ciertos elementos propios de la computación cuántica. La computadora cuántica estará formada por una ALU cuántica, memoria cuántica y un planificador dinámico, tal como se puede observar en la figura 5.1.

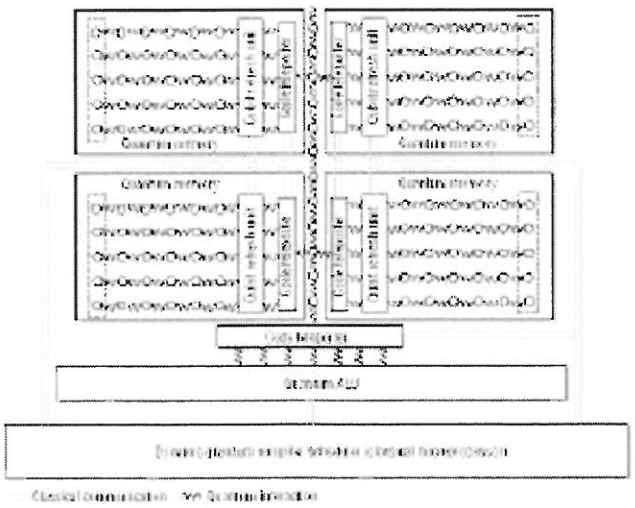


Figura 5.1. Computador cuántico

La corrección de error tomado muy en cuenta en el diseño de una arquitectura cuántica.

5.1. ALU CUÁNTICA

La ALU cuántica tiene como funciones fundamentales la ejecución de operaciones cuánticas y la corrección de errores.

La ALU prepara los datos cuánticos, antes de ejecutar cualquier puerta lógica, aplicando una secuencia de transformaciones cuánticas básicas, que incluyen:

- Hadamard (raíz cuadrada, transformada de Fourier de un qubit)
- I, Identidad (I, NOP cuántico)
- X, NOT cuántico
- Z, cambia los signos de las amplitudes
- Y=XZ
- Rotación por $\pi/4$ (S)
- Rotación por $\pi/8$ (T)
- NOT controlado (CNOT)

La ALU aplica esta secuencia de operaciones elementales para la corrección de errores, indispensable en la computación cuántica. Este procedimiento consume estados auxiliares adicionales, para la verificación de paridad. La ALU hace uso de hardware especializado estándar, que provee estados elementales estándares, para producir los estados auxiliares adicionales.

5.2. MEMORIA CUÁNTICA

Al igual que en las arquitecturas actuales en la arquitectura cuántica, la memoria cuántica es un elemento arquitectural muy importante. La memoria cuántica debe ser confiable, con el propósito de dotarla de esta característica se incluye una unidad especializada de "actualización" en cada banco de memoria, como se muestra en la figura 5.1.. Una unidad especializada actualiza periódicamente los qubits lógicos individuales, ejecutando algoritmos de detección y corrección de errores.

5.3. TELE TRANSPORTADORA DE CÓDIGO

La tele transportadora de código desde la memoria cuántica a la ALU, añade alguna funcionalidad adicional a la tele transportación cuántica convencional, proveyendo un mecanismo general para simultáneamente ejecutar operaciones mientras transporta los datos cuánticos.

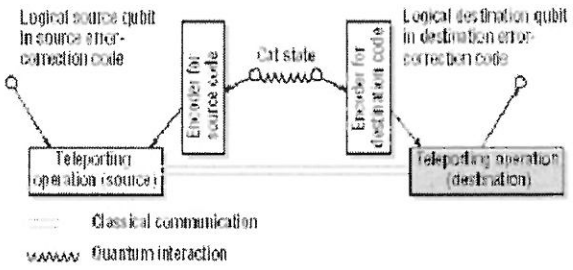


Figura 5.2. Tele transportadora de código

Este mecanismo se usa para la corrección de errores en el codificador de código origen y en el codificador de código destino, como se observa en la figura 5.2.. El emisor y el receptor entonces ejecutan qubits lógicos equivalentes en la operación de tele transportación en cada terminal del par enredado (entangled).

5.4. PLANIFICADOR DINÁMICO

Se puede proponer un procesador clásico de alto rendimiento como parte principal del planificador dinámico. Este procesador ejecuta un algoritmo de planificación dinámico que toma operaciones cuánticas lógicas, intercaladas con construcciones clásicas de control de flujo, dinámicamente las traduce en operaciones individuales de qubits físicos.

5.5. COMPUTADORA CUÁNTICA

Para hacerse una idea de cómo sería una computadora cuántica debemos imaginarla como un sistema de circuitos cuánticos, actuando en un espacio de estados, que es un espacio complejo 2n-dimensional de Hilbert. El circuito es una secuencia de transformaciones unitarias $U_t \in SU(2^n)$ seguido por una medición. Esas transformaciones, son llamadas puertas cuánticas y son controladas por una computadora clásica. El espacio de estados de una computadora cuántica tiene la estructura de un espacio de un vector Hermitian. Así esto permite la superposición simultanea de estados básicos ortogonales (correspondientes a estados clásicos 0 y 1) con la posibilidad de interferencia constructiva y destructiva entre las diferentes

rutas de computación. Este principio permite el uso de los estados enredados (entangled states).

Para implementar una computadora cuántica, se deben cumplir al menos cinco requisitos:

1. Se necesita un sistema de qubits.
2. Los qubits deben ser individualmente direccionables y deben interactuar con otros para conformar puertas lógicas de propósito general.
3. Debe ser posible la inicialización de las puertas.
4. Se debe tener la posibilidad de extraer los resultados computacionales.
5. Se necesita un tiempo de coherencia duradero.

CAPITULO 6

6. SIMULACIÓN DE PUERTAS LÓGICAS

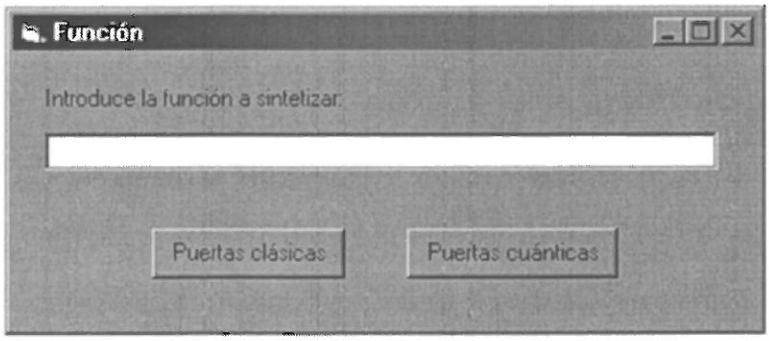
En este capitulo se explica el funcionamiento del sintetizador realizado para este proyecto, se muestran los organigramas usados para el sintetizador y se realiza una valoración económica.

6.1. FUNCIONAMIENTO DEL SINTETIZADOR

El sintetizador creado para este proyecto puede sintetizar tanto puertas lógicas clásicas como puertas cuánticas de dos entradas, las funciones de entrada se pueden ir modificando y así observar los cambios que se producen en los circuitos lógicos.

El funcionamiento del programa es el siguiente:

Una vez realizada la instalación, hacemos doble click en el ejecutable sintetizador del menú inicio, nos aparecerá la siguiente pantalla:

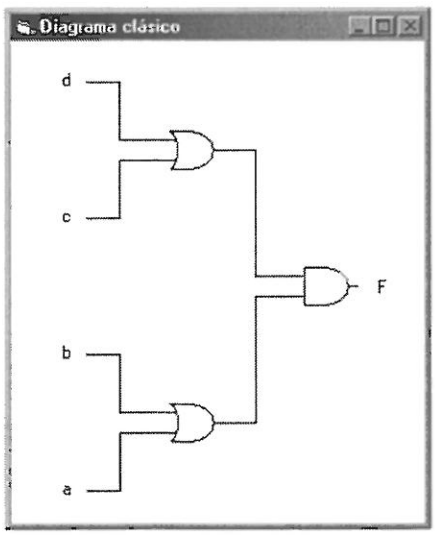


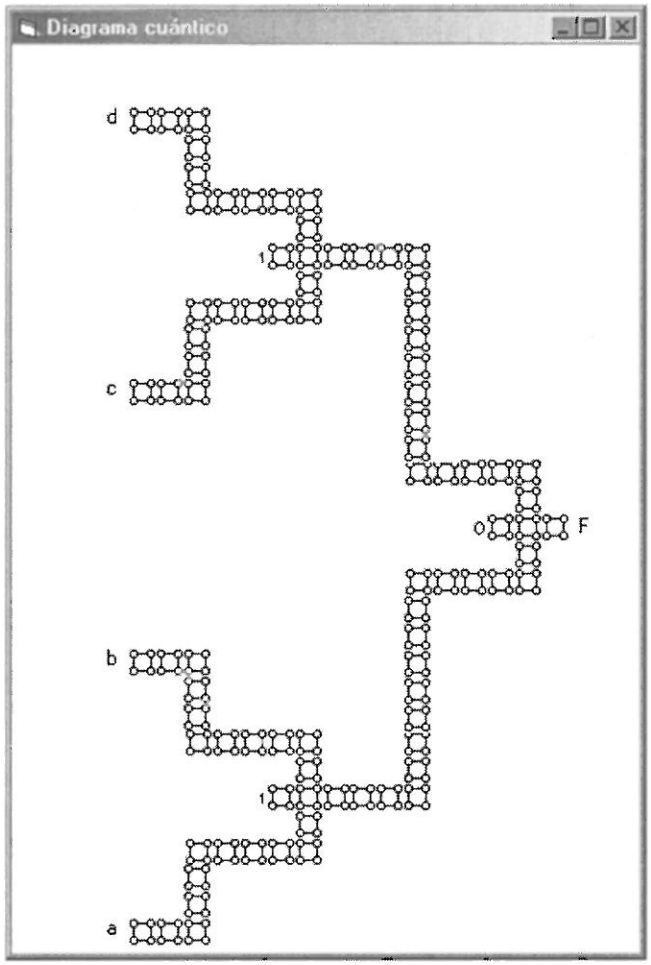
Introducimos la función que queremos sintetizar. Esta función puede estar constituida por letras y números, las operaciones que se pueden introducir son la operación and (*), la operación or (+) y la operación not, esta última operación se introduce con los corchetes, introduciendo entre estos la variable o función que queremos complementar, por ejemplo, para complementar una variable, sería [a] y para complementar una función [a+b].

Una vez introducida la función pulsamos en uno de las dos botones que muestran el la figura anterior, al hacerlo nos aparecerá el circuito lógico clásico o cuántico, una vez introducida la función y visualizado el circuito podemos modificar la función para observar las modificaciones que se producen en el circuito lógico.

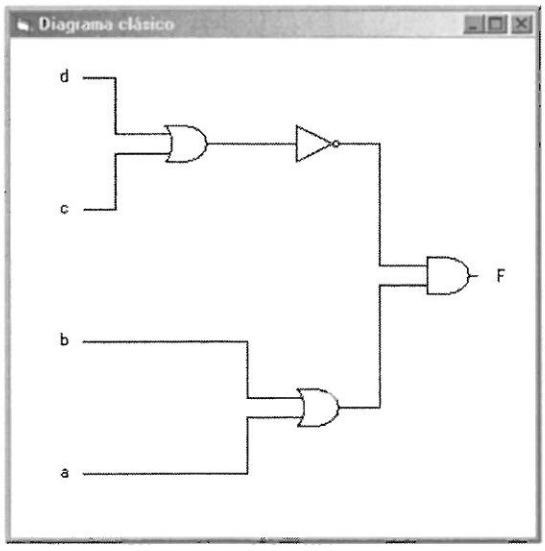
Veamos algunos ejemplos de como funciona el programa:

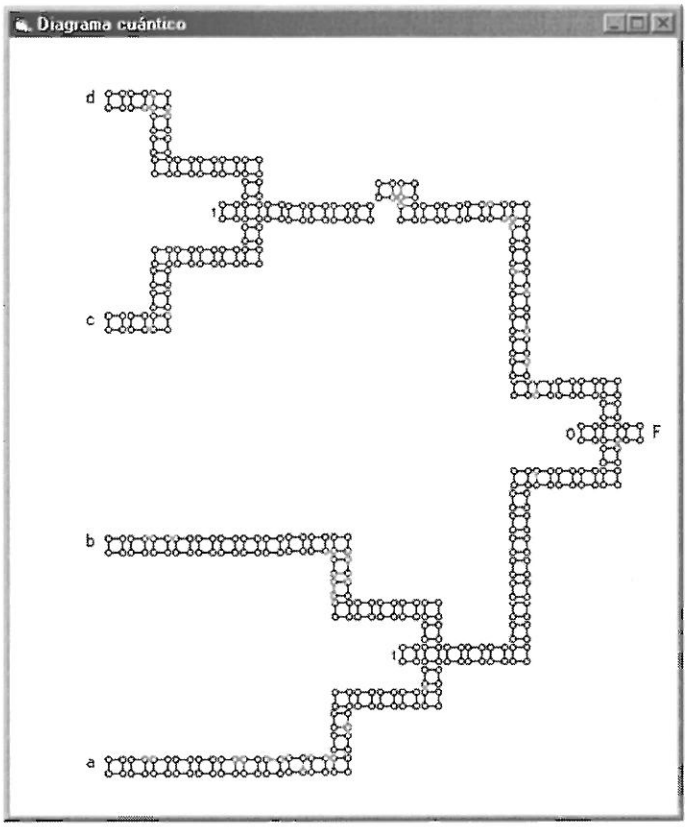
Introducimos la función *(a+b)*(c+d)* el resultado que obtenemos para puertas clásicas y cuánticas es el siguiente:





Introducimos ahora la función *(a+b)*[(c+d)]* el resultado que obtenemos para puertas clásicas y cuánticas es el siguiente:





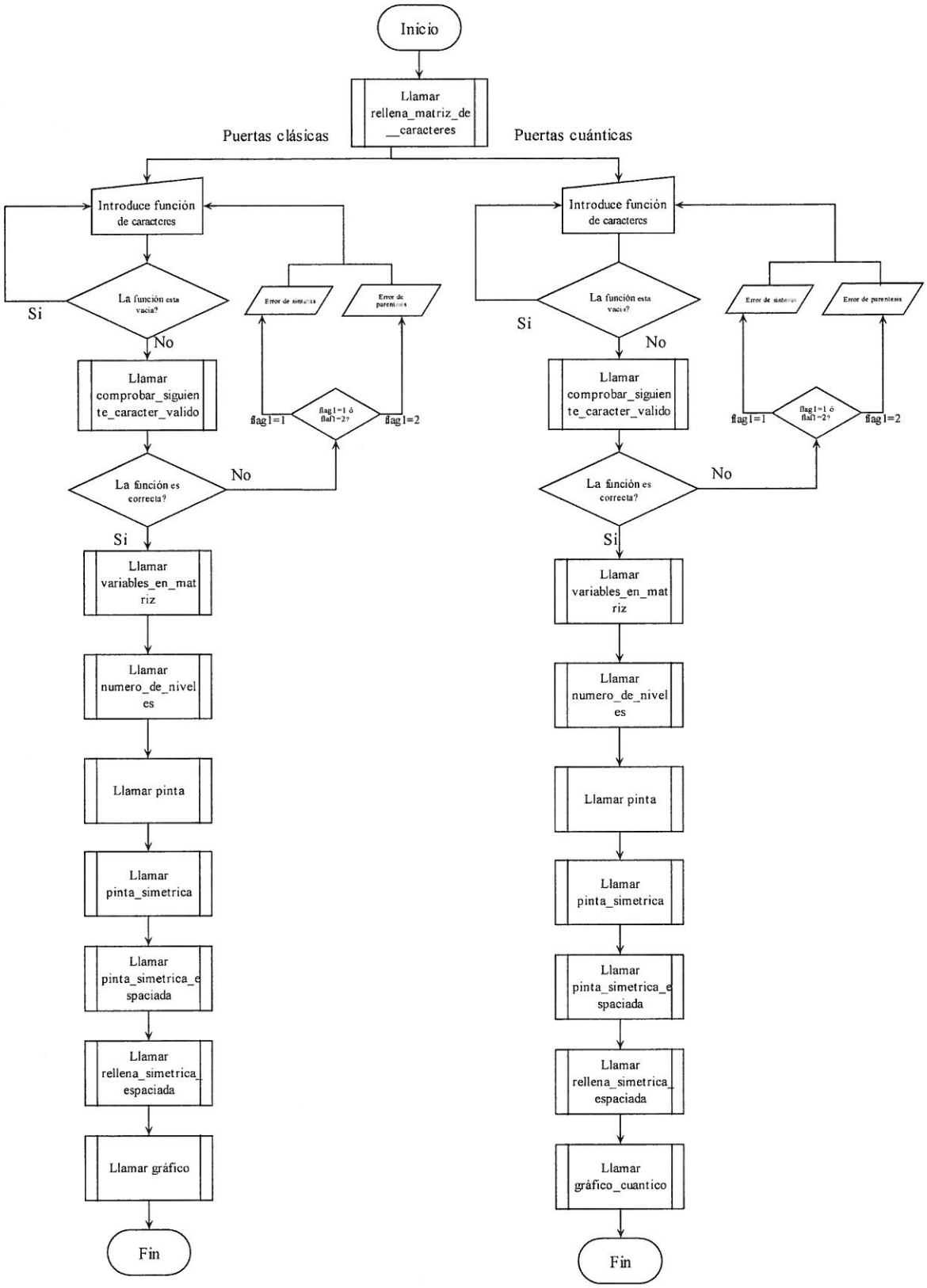
6.2 ORGANIGRAMAS

Los diagramas de flujo correspondientes a este programa se muestran a continuación:

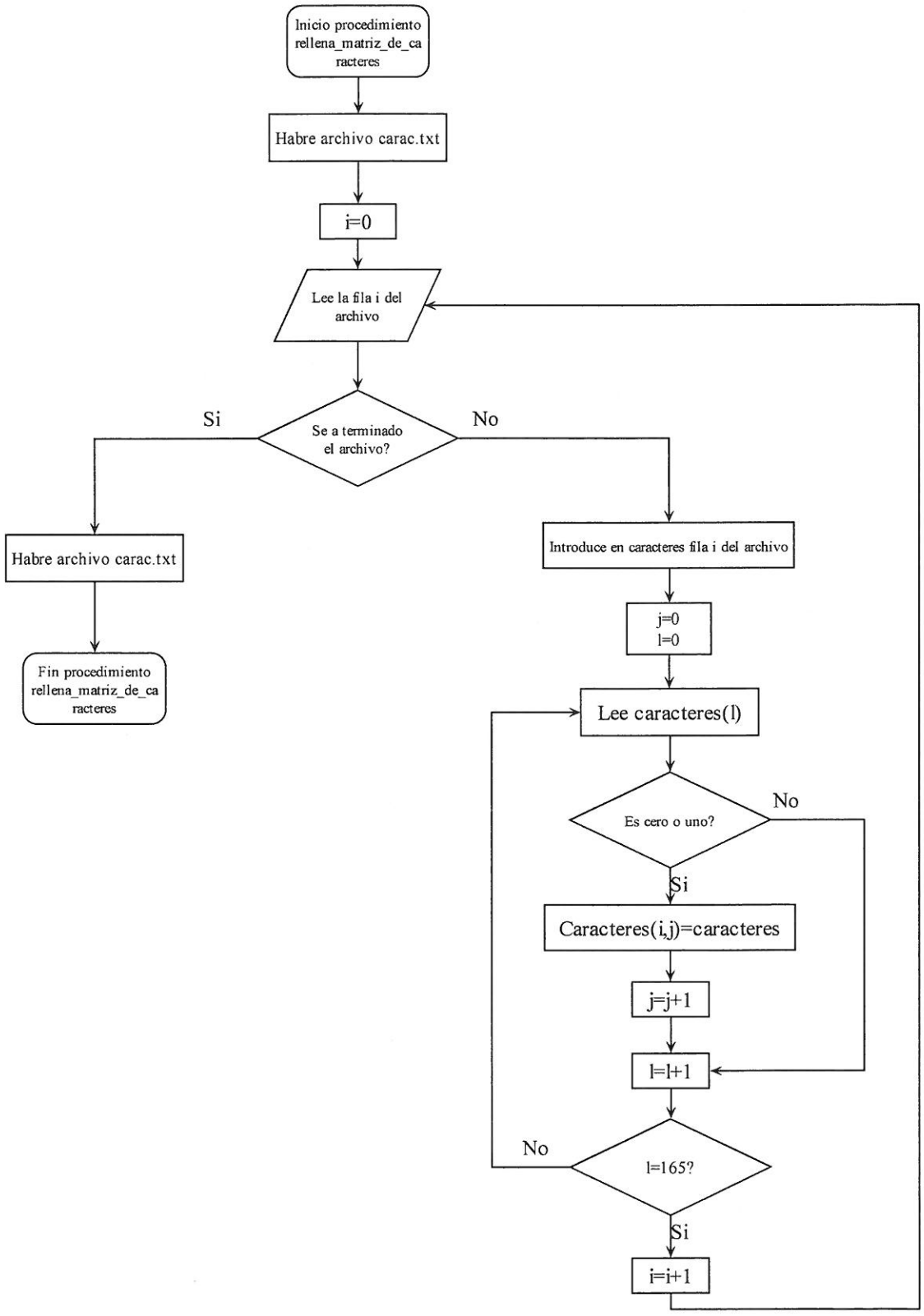
- Organigrama nº1: Diagrama de flujo principal donde se van haciendo llamadas a los procedimientos.
- Organigrama nº2: procedimiento donde se rellena una matriz con los caracteres validos, que posteriormente se introducirán para sintetizar la función.
- Organigrama nº3: procedimiento que comprueba si la función introducida tiene errores de sintaxis.
- Organigrama nº4: procedimiento que convierte la cadena función en números.
- Organigrama nº5: procedimiento que determina el número de niveles que tiene la función contando el número de paréntesis.

- Organigrama nº6: procedimiento que introduce en una matriz y de forma ordenada las variables y los operadores para su posterior visualización.
- Organigrama nº7: procedimiento que introduce filas alternas en blanco para que la función quede de forma simétrica.
- Organigrama nº8: procedimiento que introduce columnas alternas en blanco para introducir las conexiones.
- Organigrama nº9: procedimiento que rellena la matriz con caracteres que identifiquen el tipo de conexión que tiene cada operador y cada variable.
- Organigrama nº10 y11: procedimiento que coloca los dibujos de los operadores, cables y conexiones en una posición determinada.

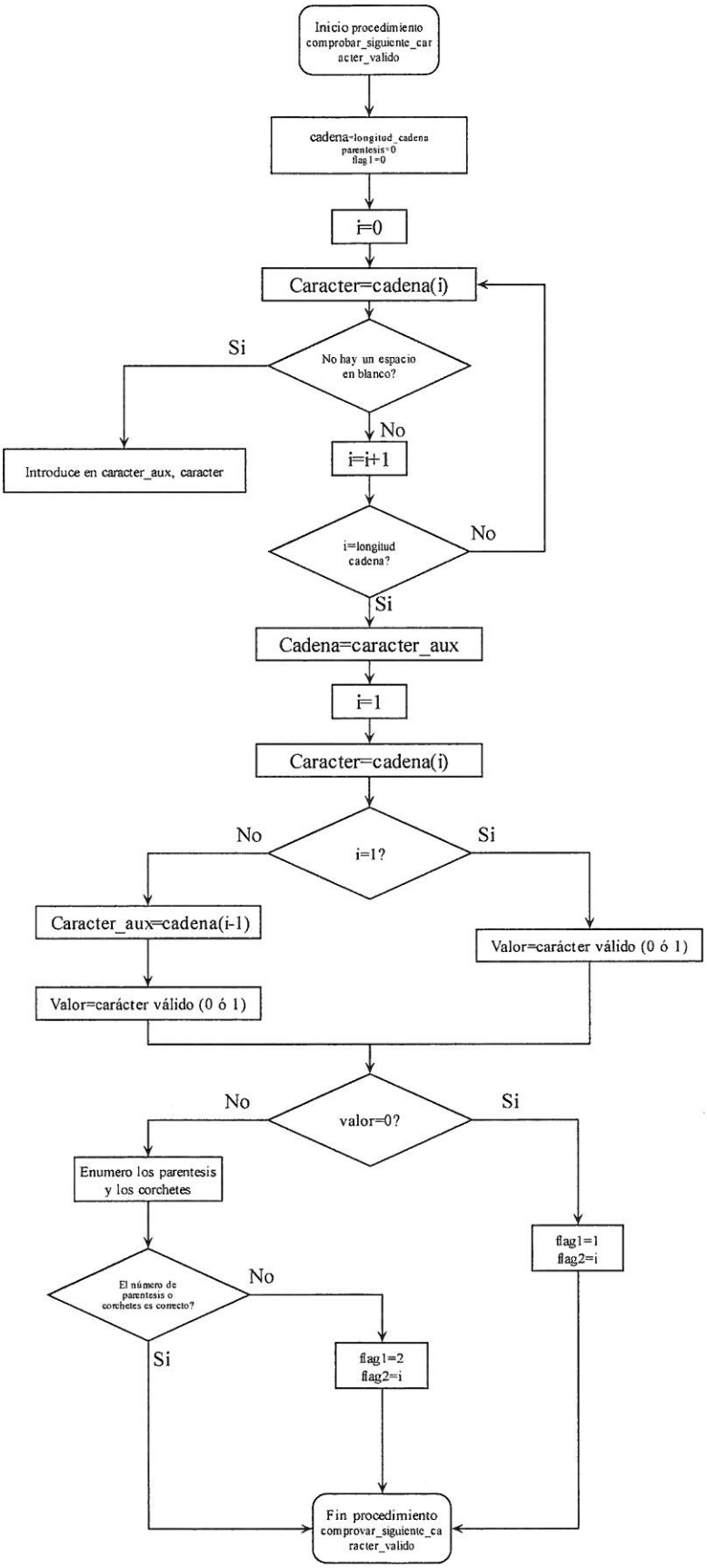
Organigrama nº1



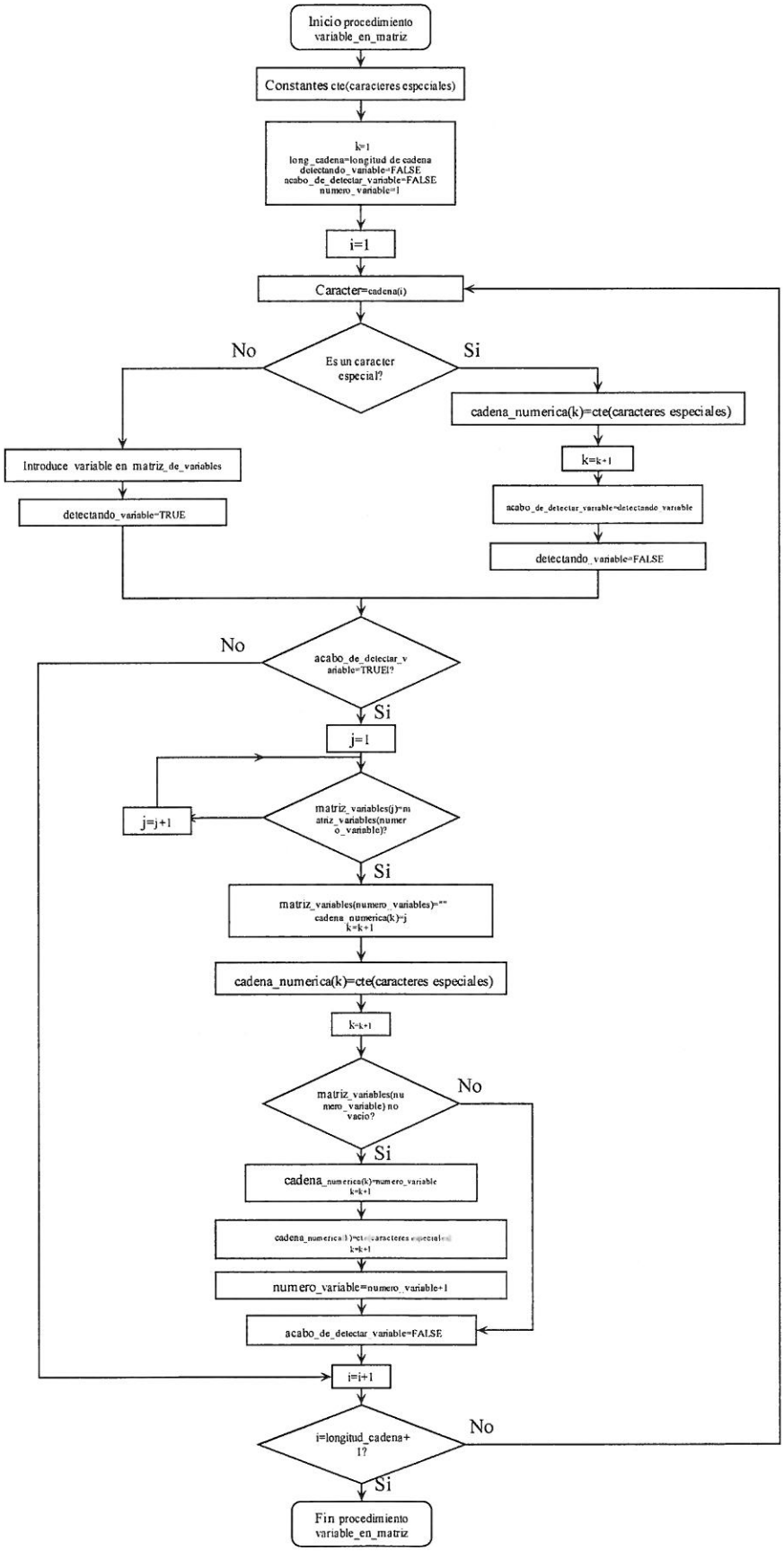
Organigrama nº2



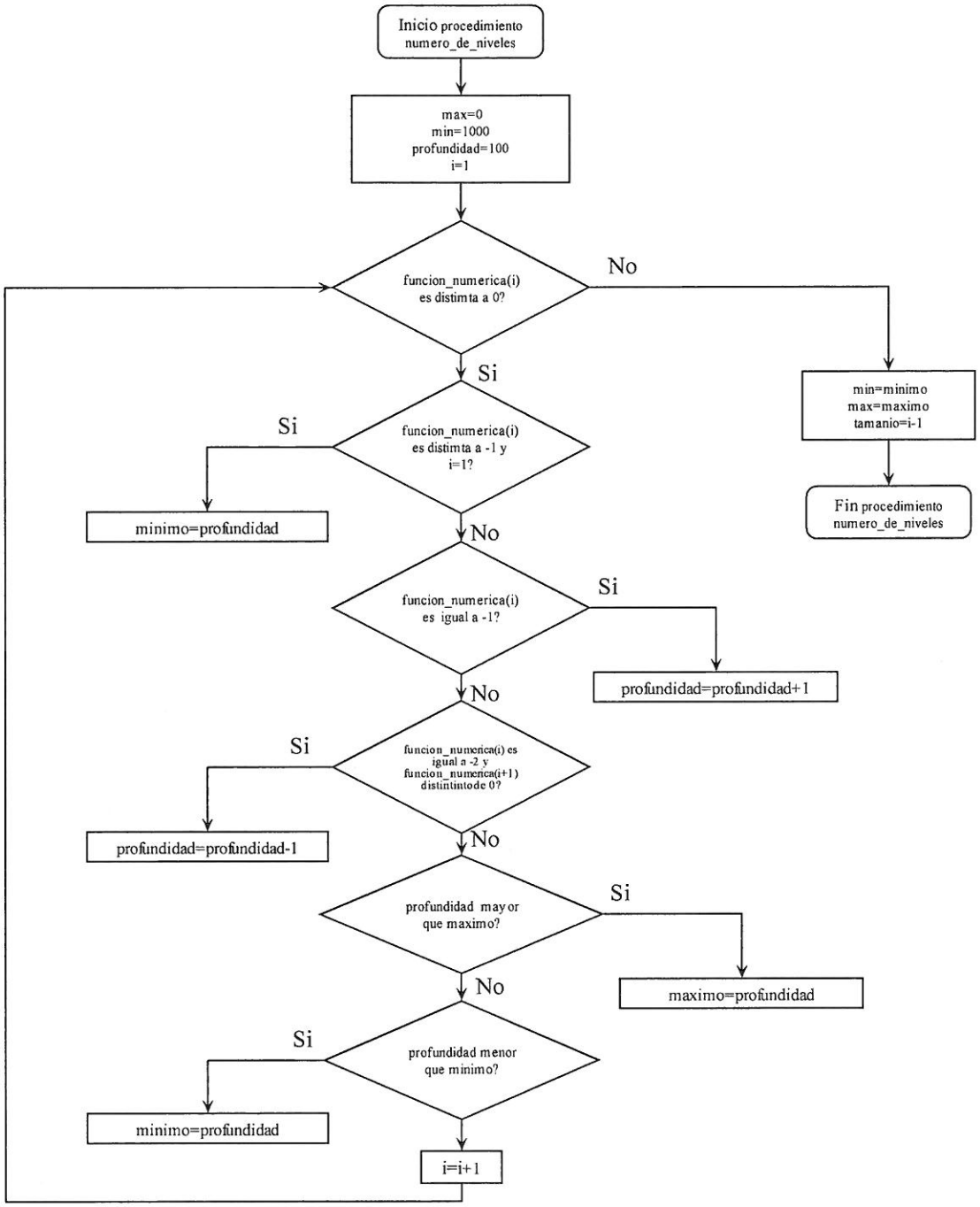
Organigrama nº3



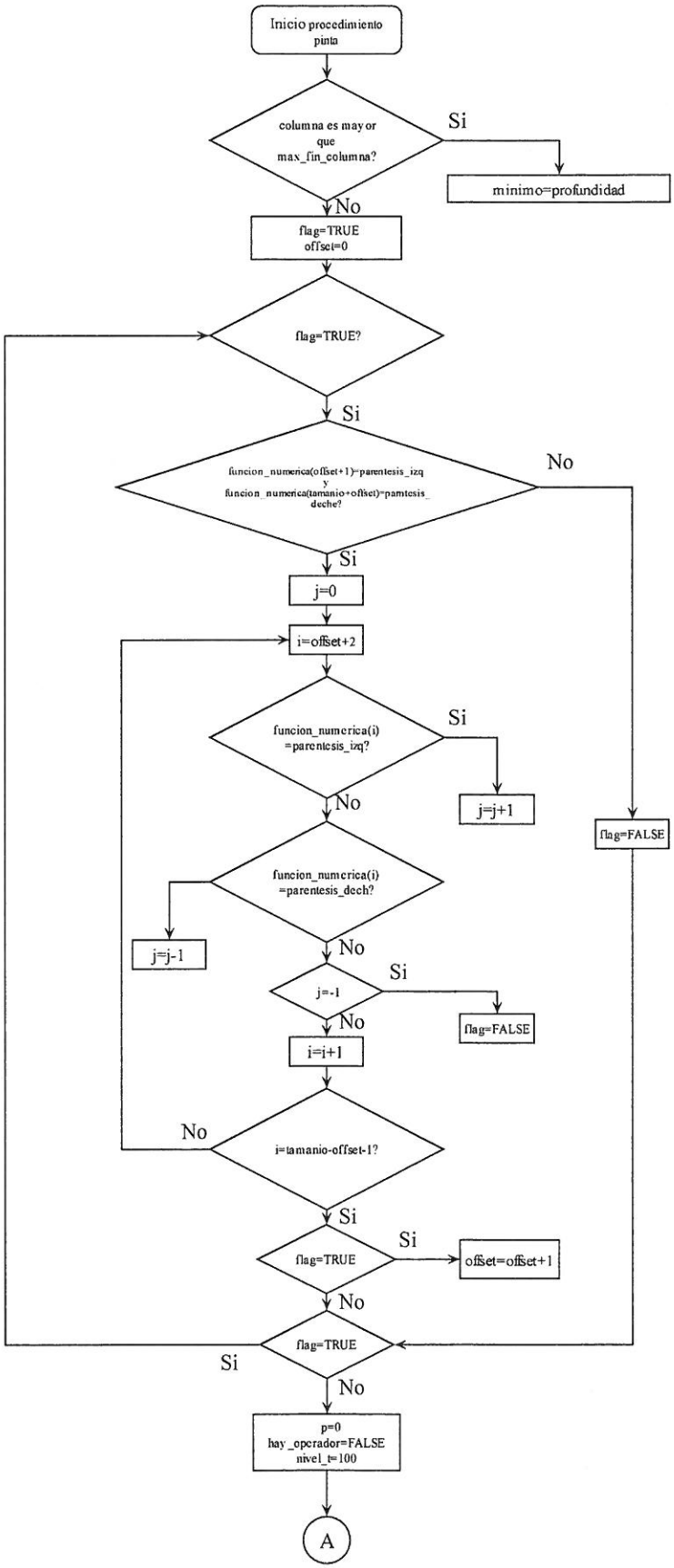
Organigrama nº4

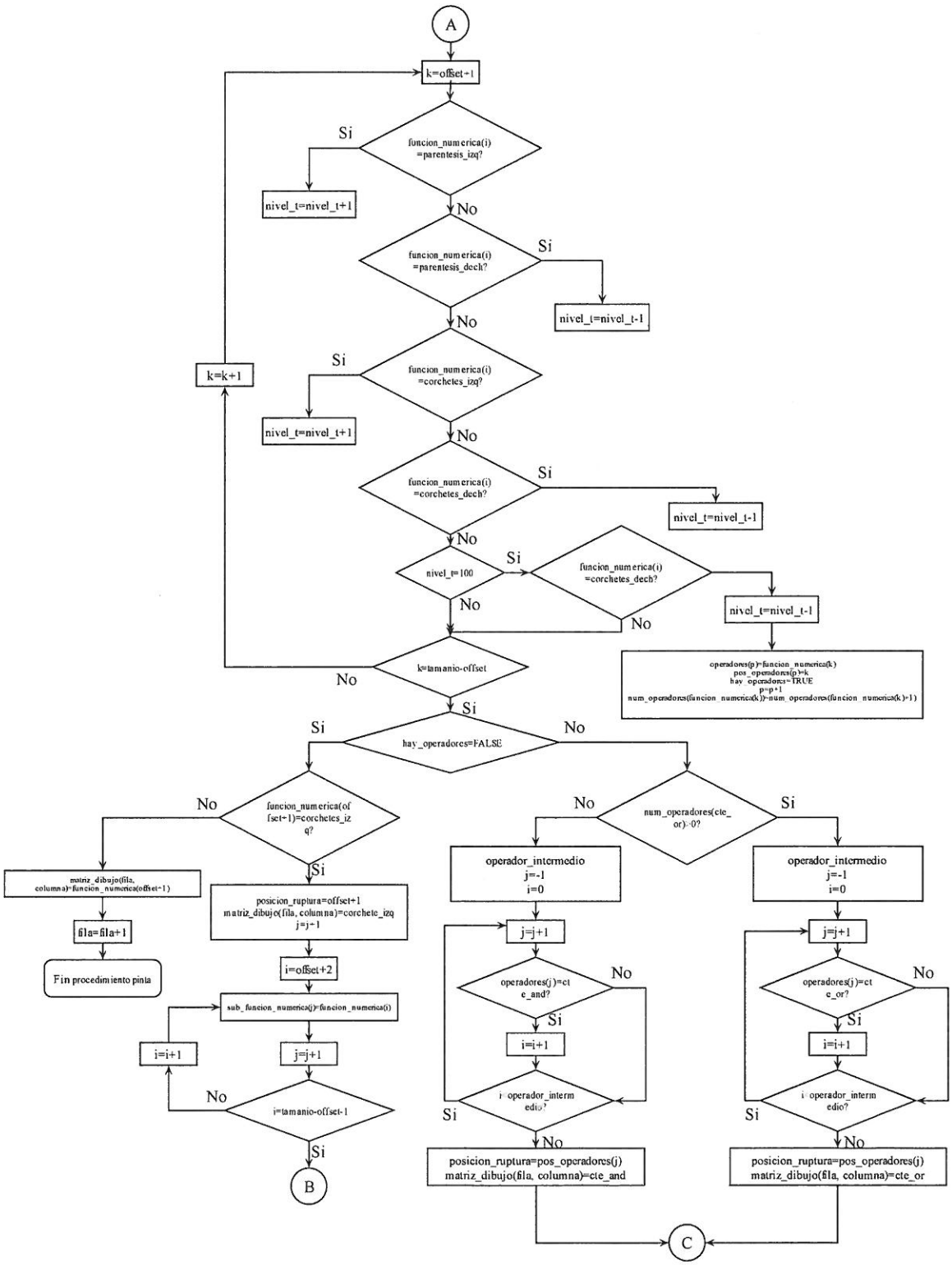


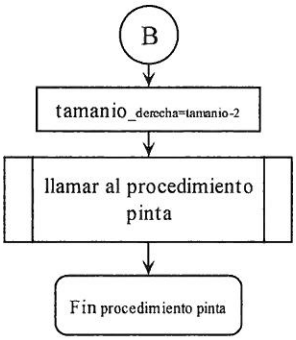
Organigrama nº5

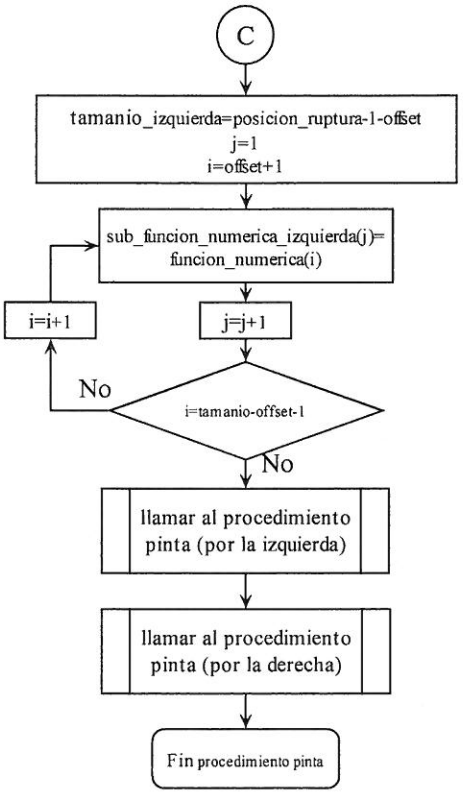


Organigrama nº6

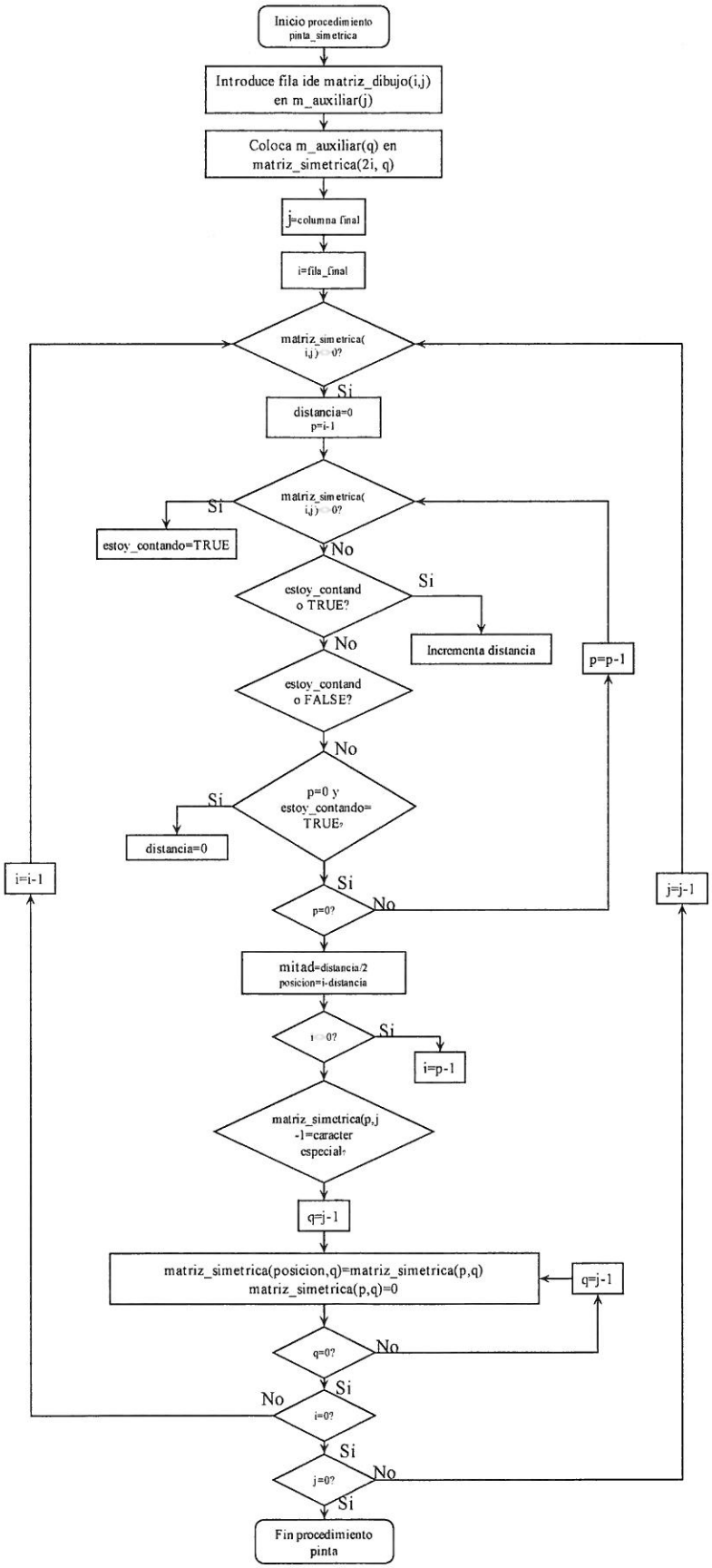



A
k=offset+1
funcion_numerica(i)=parentesis_izq?
Si
nivel_t=nivel_t+1
No
funcion_numerica(i)=parentesis_dech?
Si
nivel_t=nivel_t-1
No
funcion_numerica(i)=corchetes_izq?
Si
nivel_t=nivel_t+1
k=k+1
No
funcion_numerica(i)=corchetes_dech?
Si
nivel_t=nivel_t-1
No
nivel_t=100
Si
funcion_numerica(i)=corchetes_dech?
No
nivel_t=nivel_t-1
operadores(p)=funcion_numerica(k)
pos_operadores(p)=k
hay_operadores=TRUE
p=p+1
num_operadores(funcion_numerica(k))=num_operadores(funcion_numerica(k))+1
No
k=tamanio-offset
No
Si
hay_operadores=FALSE
Si
No
funcion_numerica(offset+1)=corchetes_izq?
No
Si
matriz_dibujo(fila, columna)=funcion_numerica(offset+1)
fila=fila+1
Fin procedimiento pinta
posicion_ruptura=offset+1
matriz_dibujo(fila, columna)=corchete_izq
j=j+1
i=offset+2
sub_funcion_numerica(j)=funcion_numerica(i)
j=j+1
i=i+1
No
i=tamanio-offset-1
Si
B
num_operadores(cte_or)=0?
No
Si
operador_intermedio
j=-1
i=0
j=j+1
operadores(j)=cte_and?
No
Si
i=i+1
i=operador_intermedio?
Si
No
posicion_ruptura=pos_operadores(j)
matriz_dibujo(fila, columna)=cte_and
operador_intermedio
j=-1
i=0
j=j+1
operadores(j)=cte_or?
No
Si
i=i+1
i=operador_intermedio?
Si
No
posicion_ruptura=pos_operadores(j)
matriz_dibujo(fila, columna)=cte_or
C

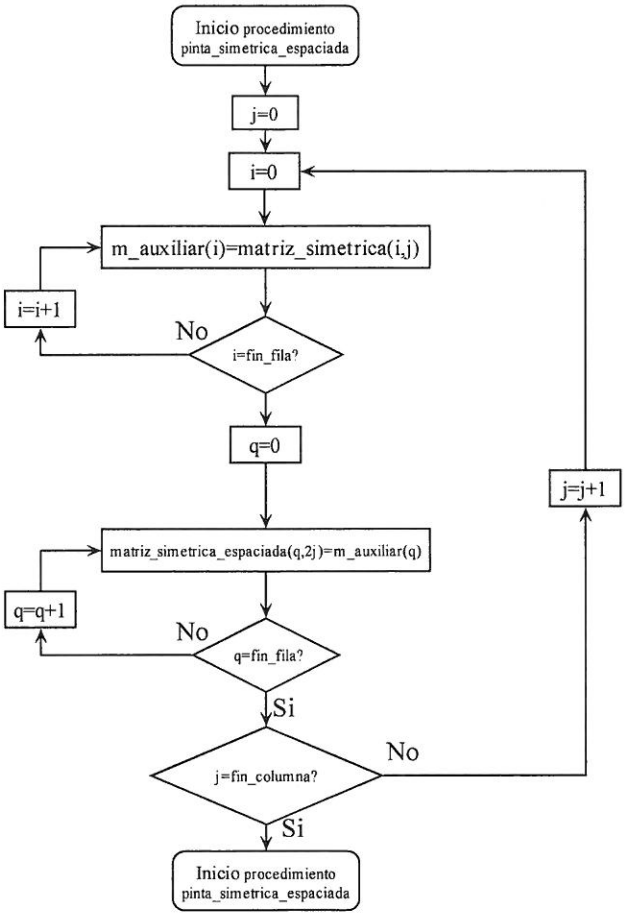

B
tamanio_derecha=tamanio-2
llamar al procedimiento pinta
Fin procedimiento pinta


C
tamanio_izquierda=posicion_ruptura-1-offset
j=1
i=offset+1
sub_funcion_numerica_izquierda(j)= funcion_numerica(i)
i=i+1
j=j+1
No
i=tamanio-offset-1
No
llamar al procedimiento pinta (por la izquierda)
llamar al procedimiento pinta (por la derecha)
Fin procedimiento pinta

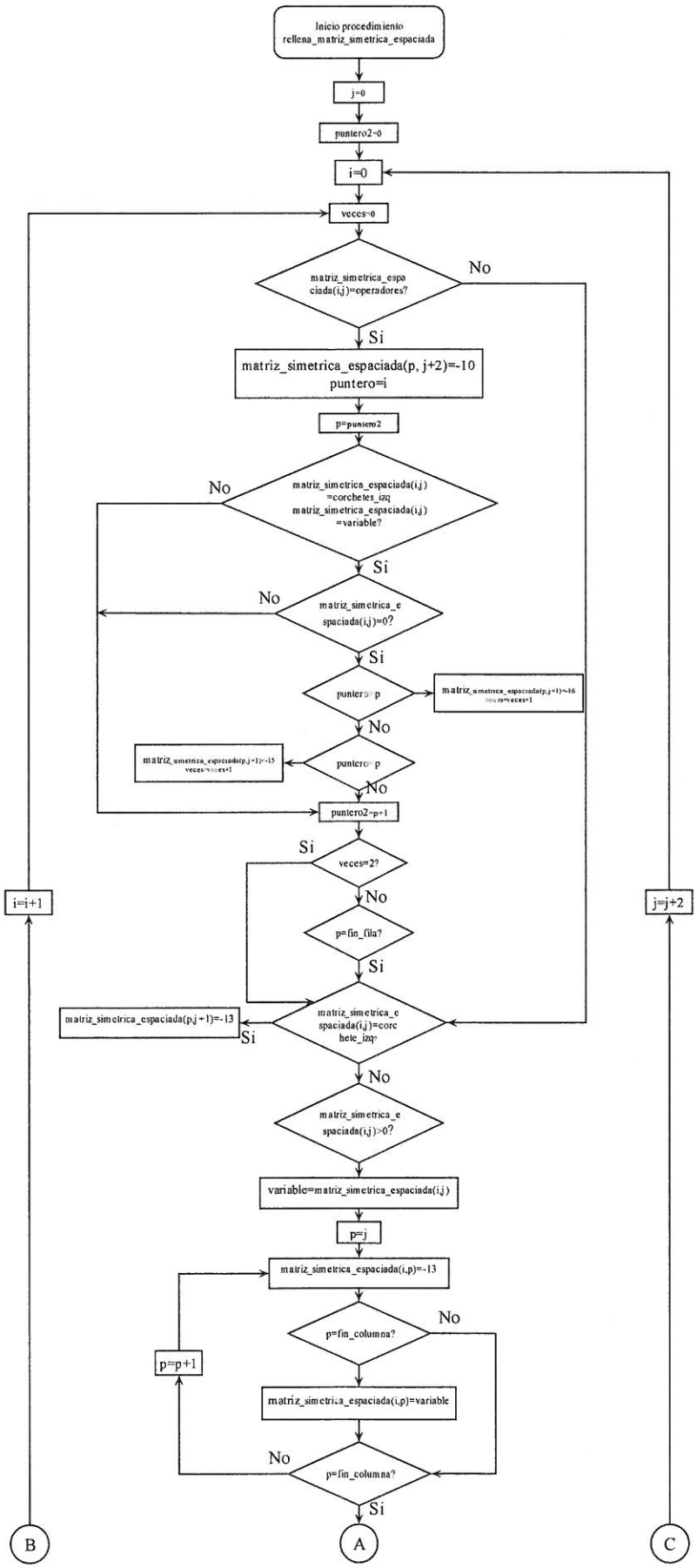
Organigrama nº7

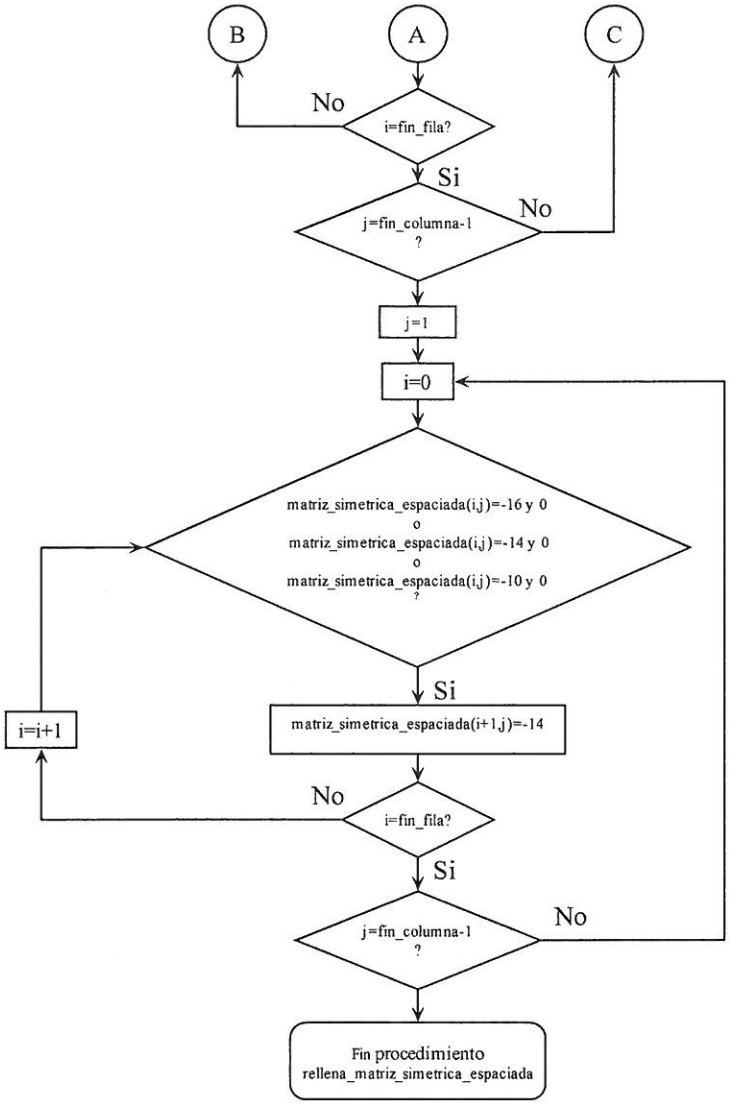


Organigrama nº8

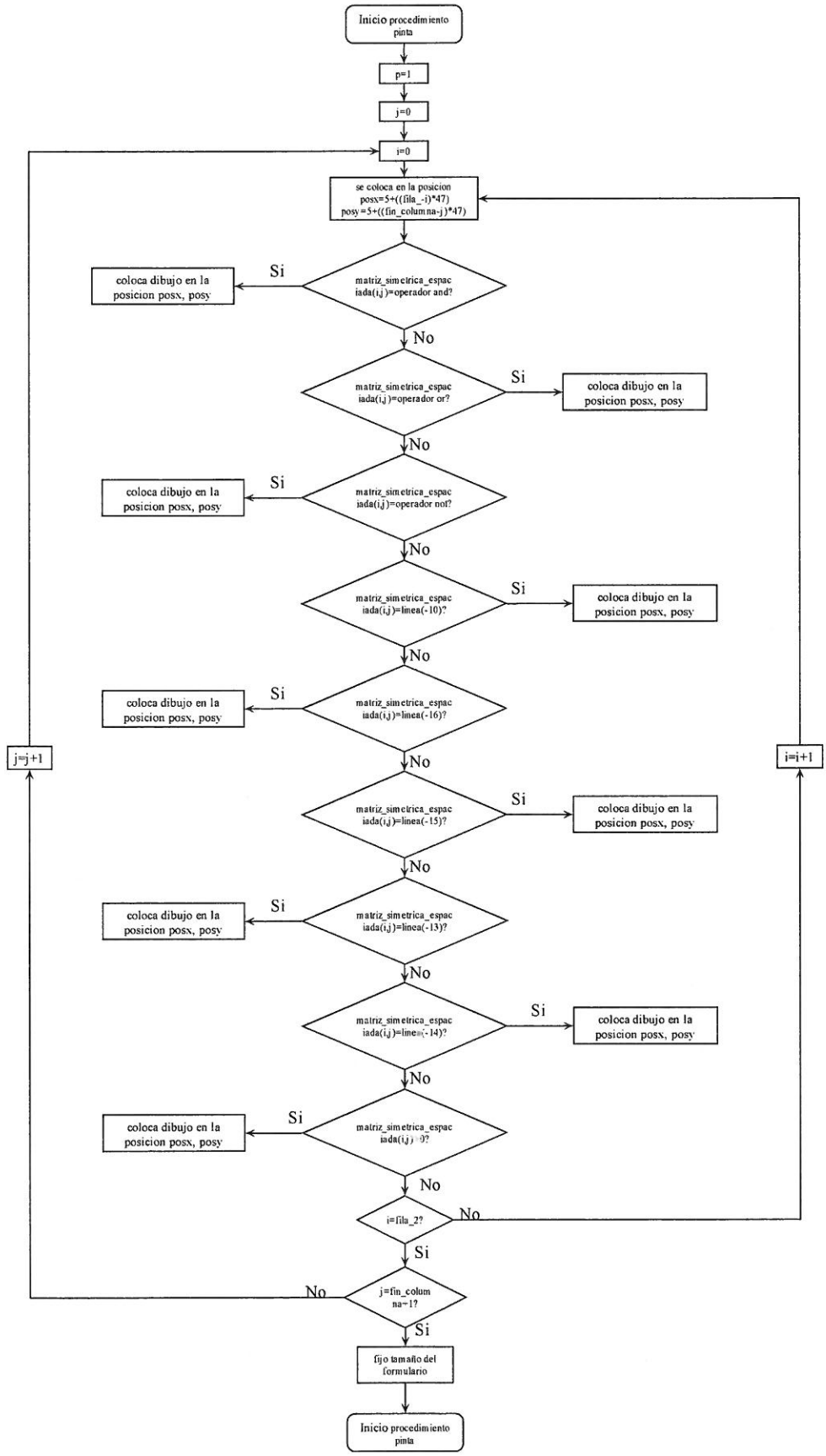


Organigrama nº 9




B
A
C
No
i=fin_fila?
Si
No
j=fin_columna-1
?
j=1
i=0
matriz_simetrica_espaciada(i,j)=-16 y 0
o
matriz_simetrica_espaciada(i,j)=-14 y 0
o
matriz_simetrica_espaciada(i,j)=-10 y 0
?
Si
i=i+1
matriz_simetrica_espaciada(i+1,j)=-14
No
i=fin_fila?
Si
No
j=fin_columna-1
?
Fin procedimiento
rellena_matriz_simetrica_espaciada

Organigrama nº 10 y 11



6.3. VALOR ECONÓMICO

Para la realización de este proyecto se han utilizado los siguientes materiales:

- Ordenador Pentium III a 933Hz
- Microsoft Visual Estudio 6.0
- Paquete Office 2000

El precio total de los componentes junto con la documentación es de aproximadamente *1200€.*

Respecto al software se ha desarrollado un programa con diversos módulos. El precio del desarrollo del programa va a venir en función del número de líneas que tenga el programa. Se estima que cada línea de código cuesta *1€.*

El número de líneas ha sido de 718. Si se hacen por media unas 40 líneas al día se necesitan 18 dias. El coste del desarrollo del software será de: *800€.*

El coste total será de: *2000€.*

7. CONCLUSIONES

Las computadoras actuales están llegando al límite de la miniaturización y la frecuencia de pulsaciones de los relojes de cuarzo, pronto no podrán ser más rápidos. La computación cuántica es una gran promesa que podría permitirnos seguir construyendo computadoras más veloces. La arquitectura cuántica es muy similar a las arquitecturas actuales, sin embargo la computación cuántica introduce elementos arquitecturales cuánticos que obedecen a los fenómenos causados por la interacción cuántica como la corrección de errores.

El avance de la computación cuántica esta limitada por sus principales ventajas. Con lo referente a la superposición cuántica, que permite el paralelismo masivo y mantener una gran cantidad de múltiples estados en un mismo instante, el mayor inconveniente esta en la imposibilidad de leer toda esa información sin desestabilizar el sistema.

Desde el punto de vista del hardware, en la parte física la meta es lograr diseñar dispositivos en sólidos, y no en gases como se da en la mayoría de los experimentos actualmente. En la parte lógica mantener la coherencia en un dispositivo cuántico es un desafío, principalmente debido a la gran cantidad de información adjunta que se necesita para garantizar la ausencia de errores, por lo que es necesario el desarrollo de mejores mecanismos de corrección de errores.

Prevenir la incoherencia y preservar los frágiles estados cuánticos. Esto es facil en pequeños sistemas pero mas complejo en grandes sistemas cuánticos.

En el futuro, se espera que las computadoras cuánticas, estén completamente desarrolladas aproximadamente el 2020. Sin embargo, la computación cuántica, ya esta siendo aplicada, es así que "Magiq" es la primera empresa que lanzará al mercado, el 2003, tecnología de encriptación cuántica. [Johnson02a]. Otro sistema de encriptación cuántica es el desarrollado por Prem Kumar y Horace Yuen, profesores de la universidad "Northwestern", [Johnson02b] capaz de codificar flujos de datos y enviarlos velocidades de las troncales de Internet.

8.- APÉNDICES

APÉNDICE A

PRODUCTO TENSORIAL

El producto tensorial ($\otimes$) de dos vectores con *n* y *k* dimensiones respectivamente es otro vector con dimensión *nk*. De igual forma, sean *A* y *B* dos transformaciones sobre dos vectores de dimensión *n* y *k* respectivamente, el producto tensorial $A \otimes B$ es una transformación sobre un vector de dimensión *nk*.

Para los propósitos de este proyecto podemos seguir las siguientes reglas algebraicas, que será suficientes para el cálculo del producto tensorial. Sea *A*, *B*, *C*, *D*, *U* matrices , *u*, *x*, *y* vectores y *a*, *b* escalares se tiene las siguientes reglas:

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

$$(A \otimes B)(x \otimes y) = Ax \otimes By$$

$$(x + y) \otimes u = x \otimes u + y \otimes u$$

$$u \otimes (x + y) = u \otimes x + u \otimes y$$

$$ax \otimes by = ab(x \otimes y)$$

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \otimes U = \begin{pmatrix} A \otimes U & B \otimes U \\ C \otimes U & D \otimes U \end{pmatrix}$$

Esta última regla puede extenderse a escalares. Siendo a, b, c, d escalares, se tiene:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes U = \begin{pmatrix} aU & bU \\ cU & dU \end{pmatrix}$$

Se define la transpuesta conjugada de un producto tensorial como:

$$(A \otimes B)^* = A^* \otimes B^*$$

Se dice que una matriz U es unitaria si su transpuesta conjugada es su inversa: $U^*U = I$.

El producto tensorial de varias matrices es unitario si y solo si cada una de la matrices es unitaria salvo una constante. Sea $U = A_1 \otimes A_2 \otimes \cdots \otimes A_n$. U es unitaria si $A_i^* A_i = k_i I$ y $\prod_i k_i = 1$.

$$U^*U = \left(A_1^* \otimes A_2^* \otimes \cdots \otimes A_n^*\right) \times \left(A_1 \otimes A_2 \otimes \cdots \otimes A_n\right) =$$
$$= A_1^* A_1 \otimes A_2^* A_2 \otimes \cdots \otimes A_n^* A_n = k_1 I \otimes \cdots k_n I = I$$

Donde I es la matriz identidad.

Por ejemplo la ley distributiva del producto tensorial permite para dos qubits lo siguiente:

$$\left(a_0|0\rangle + b_0|1\rangle\right) \otimes \left(a_1|0\rangle + b_1|1\rangle\right) =$$
$$= \left(a_0|0\rangle \otimes a_1|0\rangle\right) + \left(b_0|1\rangle \otimes a_1|0\rangle\right) + \left(a_0|0\rangle \otimes b_1|1\rangle\right) + \left(b_0|1\rangle \otimes b_1|1\rangle\right) =$$
$$= a_0 a_1\left(|0\rangle \otimes |0\rangle\right) + b_0 a_1\left(|1\rangle \otimes |0\rangle\right) + a_0 b_1\left(|0\rangle \otimes |1\rangle\right) + b_0 b_1\left(|1\rangle \otimes |1\rangle\right) =$$
$$a_0 a_1|00\rangle + b_0 a_1|10\rangle + a_0 b_1|01\rangle + b_0 b_1|11\rangle$$

APÉNDICE B

EL PRINCIPIO DE INCERTIDUMBRE DE HEISENBERG

El hecho de que cada partícula lleva asociada consigo una onda, impone restricciones en la capacidad para determinar al mismo tiempo su posición y su velocidad. Este principio fué enunciado por W. Heisenberg en 1927.

Es natural pensar que si una partícula esta localizada, debemos poder asociar con ésta un paquete de ondas mas o menos bien localizado. Un paquete de ondas se construye mediante la superposición de un número infinito de ondas armónicas de diferentes frecuencias. En un instante de tiempo dado, la función de onda asociada con un paquete de ondas esta dado por

$$\Psi(x) = \int_0^{+\infty} g(x) Sen(kx) dx$$

donde *k* representa el número de onda

$$k = \frac{2\pi}{\lambda} = \frac{2\pi\nu}{c}$$

y donde la integral representa la suma de ondas con frecuencias (o número de ondas) que varian desde cero a mas infinito ponderadas mediante el factor *g*(*k*).

El momento de la partícula y el número de ondas estan relacionados ya que

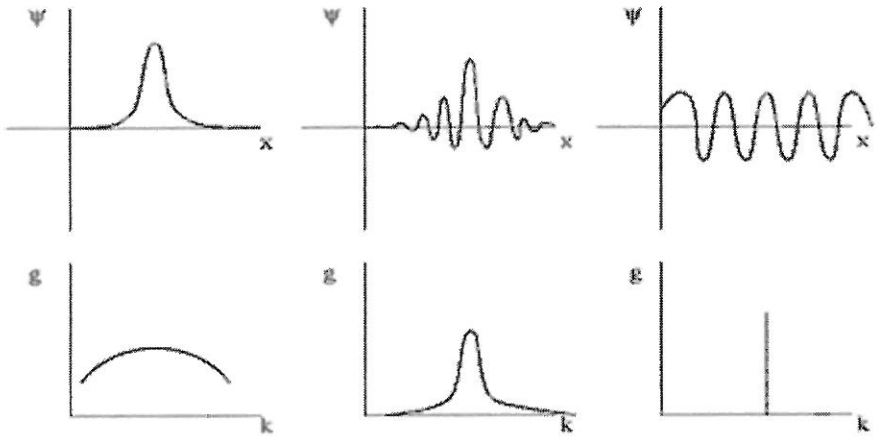
$$p = \frac{h\nu}{c}; \qquad k = \frac{2\pi\nu}{c}$$

de lo cual se deduce que

$$p = \frac{h}{2\pi} k = \hbar k$$

Queda claro que para localizar una partícula es necesario sumar todas las contribuciones de las ondas cuyo número de onda varia entre cero e infinito y por lo tanto el momento $p = \hbar k$ también varia entre cero e infinito. Es decir que esta completamente indeterminado.

Para ilustrar lo anterior hemos indicado en la siguiente figura diferentes tipos de paquetes de onda y su *transformada de Fourier* que nos dice como estan distribuidas las contribuciones de las ondas con número de ondas *k* dentro del paquete.



En el primer caso vemos que un paquete de ondas bien localizado en el espacio *x*, tiene contribuciones prácticamente iguales de todas las ondas con número de ondas *k*.

En el segundo caso vemos que si relajamos un poco la posición del paquete de ondas, también es posible definir el número de ondas (o el momento) de la partícula.

En el último caso vemos que para definir bien el momento $p = \hbar k$ de la partícula, entonces su posición queda completamente indefinida.

Es posible determinar el ancho, o la incertidumbre, del paquete de ondas tanto en el espacio normal Δx como en el espacio de momentos Δp.

El *principio de incertidumbre* nos dice que hay un límite en la precisión con el cual podemos determinar al mismo tiempo la posición y el momento de una partícula. La expresión matemática que describe el principio de incertidumbre de Heisenberg es:

$$\Delta x \Delta p \geq \hbar$$

Si queremos determinar con total precisión la posición:

$$\Delta x = 0$$

De la desigualdad para el principio de incertidumbre verificamos entonces que

$$\Delta p \geq \frac{\hbar}{\Delta x} \rightarrow \infty$$

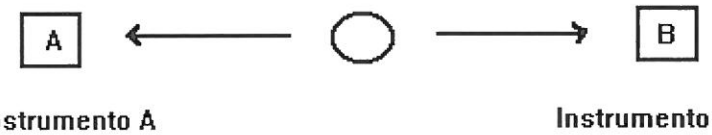
Es decir, que la incertidumbre en el momento es infinita.

APÉNDICE C

EL TEOREMA DE BELL

El Teorema de Bell prueba la conexión-correlación entre sistemas no relacionados causalmente. Bell aduce que mientras la separación en el tiempo o en el espacio son "reales" en ciertos contextos, dicha separación es "irreal" o carece de importancia en la mecánica cuántica.

Imagínese una fuente que emite dos corrientes de fotones (o rayos de luz, para entenderlo mejor), fotones que son interceptados por dos instrumentos: A y B



Instrumento A **Instrumento B**

Estos instrumentos pueden estar todo lo lejos que se quiera entre sí, incluso hallarse emplazados en puntos opuestos del universo. Por simple aplicación de leyes aceptadas de la mecánica cuántica, Bell demuestra que cualquier propiedad de las partículas que se mida en el instrumento A, provocará, *simultáneamente*, una medición matemáticamente complementaria en el instrumento B. Lo asombroso del caso viene cuando nos damos cuenta de que eso significa que cada fotón sabe la medición a la que está siendo sometido el otro fotón, y lo sabe *instantáneamente*.

Bell prueba que este tipo de relación no-local debe darse tanto en separaciones espaciales como en separaciones temporales. Todo parece indicar que "cierta energía" es la causante de esta correlación simultánea de conocimiento, pero en física no se conoce una energía que pueda moverse tan rápidamente. Einstein, ya en 1935, se topó con este efecto misterioso derivado de la mecánica cuántica, y lo tildó de "fantasmal" (spooky), ya que ampararía fenómenos paranormales hasta entonces desdeñados por la ciencia, como la telepatía. Einstein concluyó que debía haber algo radicalmente erróneo en la mecánica cuántica para permitir llegar a semejantes conclusiones.

APÉNDICE D

OTRAS TECNOLOGIAS

Computadoras moleculares.

Como solución al problema de la miniaturización, se plantea la posibilidad de usar moléculas en lugar de switches. La ventaja es clara puesto que las moléculas apenas tienen unos pocos nanometros y se pueden empaquetar a mas bajo costo.

El funcionamiento de un switch molecular usa el efecto de túnel cuántico. Hay unas moléculas especiales que se pueden contraer y relajar ante un cambio eléctrico en su entorno. Colocándola a modo de puente entre dos conductores, al contraerse acercara los conductores provocando el efecto túnel cuántico.

En la practica, se están construyendo dispositivos de este tipo. Constan de una capa de unas pocas moléculas de espesor, la cual se sumerge en una capa de silicio con los electrodos incorporados con lo que se la molécula se orienta adecuadamente. Se repite el proceso por el otro lado de la capa de moléculas.

También se trabaja en mejorar el cableado. En este aspecto se han desarrollado unos nanotubos de carbono que parecen ideales para conducir electrones por su interior, reduciendo el problema del sobrecalentamiento asociado a la miniaturización.

Computación óptica.

Se basa en reemplazar los electrones por fotones. Estos son mucho más rápidos, no requieren de condiciones especiales para su transmisión. La lógica binaria se obtiene orientando en una u otra posición el fotón. Así pues los chips deben trabajar con fotones y no con electrones.

Para esto se crean unos cristales fotónicos, similares en su comportamiento a los semiconductores. Permiten desviar y conducir el curso del fotón. El problema esta en el proceso de fabricación de estos, puesto que hay que producir pequeños defectos en el cristal para que refleje la luz en la dirección deseada. Hay dos técnicas: la litografía y el crecimiento espontáneo. La litografía permite mucha precisión y poner los defectos donde se quiera, pero solo a muy pequeña escala. En cambio el

crecimiento espontáneo se pueden conseguir cristales a gran escala con precisión, pero aun no han logrado controlar donde poner los defectos. Así pues, aun queda camino por andar, los que trabajan en litografía buscan medios para producir a gran escala y los que prefieren el crecimiento espontáneo buscan el modo de generar defectos de manera controlada.

Puertas lógicas reversibles.

Al aumentar la densidad se hace más difícil disipar el calor. Para solucionar esto, Xerox e IBM están evaluando la posibilidad de retornar a los capacitadores a su estado inicial al final de los cálculos. Por eso las puertas lógicas podrían retomar algo de la energía expulsada generando menos perdidas de calor.

APÉNDICE E

CODIGO DEL PROGRAMA

Código correspondiente al diagrama de flujo nº 1:

```
Dim flag1, flag2 As Integer
Dim marcar As Boolean
Dim matriz_de_caracteres(200, 200) As Integer
Option Explicit


Private Sub Command1_Click()
Dim funcion As String
   Dim funcion_numerica(1000) As Integer
   Dim vector_pila(1000) As Integer
   Dim vector_aux(1000) As Integer
   Dim longitud, num, i, j, total As Integer
   Dim nivel_min, nivel_max, tamanio, fila_2, columna, fin_fila, fin_columna, fin_columna_2 As Integer
   Dim comienzo, final, comienzo_drch, final_drch As Integer
   Dim matriz_de_variables(0 To 1000) As String
   Dim matriz_de_todas_las_variables(0 To 1000) As String
   Dim matriz_de_variables_numericas(0 To 1000) As Integer
   Dim matriz_dibujo(100, 100) As Integer
   Dim matriz_simetrica(100, 100) As Integer
   Dim matriz_simetrica_espaciada(100, 100) As Integer
   Dim frmForm2 As Form2
   'Entrada de la función a sintetizar
   funcion = Text1.Text
   LCase (funcion) 'pone la funcion en minuscula
   longitud = Len(funcion) 'numero de caracteres de la funcion
   num = Asc(funcion)
   'quito espacios en blanco
   If funcion = " " Then
      MsgBox "Debes introducir una funcion primero"
      Exit Sub
   End If
   'miro si la sintaxis es correcta
   'module 3
   Call comprobar_siguiente_caracter_valido(flag1, flag2, funcion, matriz_de_caracteres)
   Select Case flag1
      Case 1
         MsgBox "Error de sintaxis en la posicion " & flag2, vbExclamation, "Sintetizador de puertas lógicas"
         marcar = True
         Text1.SetFocus
         Exit Sub
      Case 2
         Text1.SelStart = flag2 - 1
         Text1.SelLength = 2
         DoEvents
         MsgBox "Error de parentesis en la posicion " & flag2, vbExclamation, "Sintetizador de puertas
lógicas"
         Text1.SetFocus
         Exit Sub
   End Select
   'introduzco las variables en una matriz
   'module 1
   Call variables_en_matriz(funcion, matriz_de_variables, matriz_de_todas_las_variables,
funcion_numerica)
```

```
    'module 4
    Call numero_de_niveles(funcion_numerica, nivel_min, nivel_max, tamanio)
    comienzo = 1
    final = tamanio
    comienzo_drch = 1
    final_drch = tamanio
    fila_2 = 0
    columna = 0
    fin_columna_2 = 0
    fin_columna = 0
    fin_fila = 0
    'module 6
    Call pinta(matriz_dibujo, funcion_numerica, fila_2, 0, tamanio, fin_columna)
    fila_2 = fila_2 - 1
    'module 7
    Call pinta_simetrica(matriz_dibujo, matriz_simetrica, fila_2, fin_columna)
    'module 7
    Call pinta_simetrica_espaciada(matriz_simetrica, matriz_simetrica_espaciada, fila_2, fin_columna)
    'module 7
    Call rellena_simetrica_espaciada(matriz_simetrica_espaciada, fila_2, fin_columna)
    'Centramos el formulario a la pantalla
    Move (Screen.Width - Width) \ 2, (Screen.Height - Height) \ 2
    Set frmForm2 = New Form2
    Call grafico(matriz_simetrica_espaciada, matriz_de_todas_las_variables, fila_2, fin_columna,
frmForm2)
    frmForm2.Show
End Sub


Private Sub Command2_Click()
Dim funcion As String
    Dim funcion_numerica(1000) As Integer
    Dim vector_pila(1000) As Integer
    Dim vector_aux(1000) As Integer
    Dim longitud, num, i, j, total, flag1, flag2 As Integer
    Dim nivel_min, nivel_max, tamanio, fila_2, columna, fin_fila, fin_columna, fin_columna_2 As Integer
    Dim comienzo, final, comienzo_drch, final_drch As Integer
    Dim matriz_de_variables(0 To 1000) As String
    Dim matriz_de_todas_las_variables(0 To 1000) As String
    Dim matriz_de_variables_numericas(0 To 1000) As Integer
    Dim matriz_dibujo(100, 100) As Integer
    Dim matriz_simetrica(100, 100) As Integer
    Dim matriz_simetrica_espaciada(100, 100) As Integer
    Dim frmForm1 As Form1

    'Entrada de la función a sintetizar
    funcion = Text1.Text
    LCase (funcion) 'pone la funcion en minuscula
    longitud = Len(funcion) 'numero de caracteres de la funcion
    num = Asc(funcion)
    'quito espacios en blanco
    If funcion = " " Then
        MsgBox "Debes introducir una funcion primero"
        Exit Sub
    End If
    'miro si la sintaxis es correcta
    'module 3
    Call comprobar_siguiente_caracter_valido(flag1, flag2, funcion, matriz_de_caracteres)
    Select Case flag1
        Case 1
            MsgBox "Error de sintaxis en la posicion " & flag2, vbExclamation, "Sintetizador de puertas lógicas"
            marcar = True
            Text1.SetFocus
            Exit Sub
        Case 2
            Text1.SelStart = flag2 - 1
            Text1.SelLength = 2
```

```
        DoEvents
        MsgBox "Error de parentesis en la posicion " & flag2, vbExclamation, "Sintetizador de puertas
lógicas"
        Text1.SetFocus
        Exit Sub
    End Select
    'introduzco las variables en una matriz
    'module 1
    Call variables_en_matriz(funcion, matriz_de_variables, matriz_de_todas_las_variables,
funcion_numerica)
    'module 4
    Call numero_de_niveles(funcion_numerica, nivel_min, nivel_max, tamanio)
    comienzo = 1
    final = tamanio
    comienzo_drch = 1
    final_drch = tamanio
    fila_2 = 0
    columna = 0
    fin_columna_2 = 0
    fin_columna = 0
    fin_fila = 0
    'module 6
    Call pinta(matriz_dibujo, funcion_numerica, fila_2, 0, tamanio, fin_columna)
    fila_2 = fila_2 - 1
    'module 7
    Call pinta_simetrica(matriz_dibujo, matriz_simetrica, fila_2, fin_columna)
    'module 7
    Call pinta_simetrica_espaciada(matriz_simetrica, matriz_simetrica_espaciada, fila_2, fin_columna)
    'module 7
    Call rellena_simetrica_espaciada(matriz_simetrica_espaciada, fila_2, fin_columna)
    'Centramos el formulario a la pantalla
    Move (Screen.Width - Width) \ 2, (Screen.Height - Height) \ 2
    Set frmForm1 = New Form1
    Call grafico_cuantico(matriz_simetrica_espaciada, matriz_de_todas_las_variables, fila_2, fin_columna,
frmForm1)
    frmForm1.Show
End Sub

Private Sub Form_Load()
    flag1 = 1
    flag2 = 1
    Call rellena_matriz_de_caracteres(matriz_de_caracteres)
End Sub

Private Sub Text1_GotFocus()
    If marcar Then
        Text1.SelStart = flag2 - 1
        Text1.SelLength = 1
    End If
    marcar = False
End Sub
```

Código correspondiente al diagrama de flujo nº 2:

```
Public Sub rellena_matriz_de_caracteres(caracteres_validos() As Integer)
Dim i, j, l As Integer
Dim caracteres, caracter  As String
Open ".\carac.txt" For Input As #1
  ''''''''''''''''''''''''''''''''''''''''''
  'relleno la matriz de caracteres'
  ''''''''''''''''''''''''''''''''''''''''''
  i = 0
  While Not EOF(1)
    Input #1, caracteres
    j = 0
    For l = 0 To 165
      caracter = Mid(caracteres, l + 1, 1)
      If (caracter = Chr(48) Or caracter = Chr(49)) Then
        caracteres_validos(i, j) = caracter
        j = j + 1
      End If
    Next l
    i = i + 1
  Wend
  Close #1
End Sub
```

Código correspondiente al diagrama de flujo nº 3:

```
Public Sub comprobar_siguiente_caracter_valido(flag1, flag2, cadena, caracteres_validos() As Integer)
   Dim caracter, caracter_aux, caracter_aux2, p As String
   Dim parentesis, i, valor, corchete As Integer
   'Voy a ver si hay errores de sintaxis
   cadena = LCase(cadena)
   parentesis = 0
   flag1 = 0
   'quito espacios en blanco
   For i = 1 To Len(cadena)
      caracter = Mid(cadena, i, 1)
      If (caracter <> " ") Then caracter_aux2 = caracter_aux2 + caracter
   Next i
   cadena = caracter_aux2
   For i = 1 To Len(cadena)
      'Con Mid separo en caracteres la cadena
      caracter = Mid(cadena, i, 1)
      If (i = 1) Then
         valor = caracteres_validos(0, Asc(caracter) - 40)
         If valor = 0 Then
            flag1 = 1
            flag2 = i
            Exit Sub
         End If
      Else
         caracter_aux = Mid(cadena, i - 1, 1)
         valor = caracteres_validos(Asc(caracter_aux) - 40, Asc(caracter) - 40)
         If valor = 0 Then
            flag1 = 1
            flag2 = i
            Exit Sub
         End If
      End If
      p = Asc(caracter)
      If (i = Len(cadena) And Asc(caracter) = 43 Or Asc(caracter) = 42) Then
         flag1 = 1
         flag2 = i
         Exit Sub
      End If
      'compruebo si el numero de parentesis es correcto
      If Asc(caracter) = 40 Then parentesis = parentesis + 1
      If Asc(caracter) = 41 Then parentesis = parentesis - 1
      If Asc(caracter) = 91 Then corchete = corchete + 1
      If Asc(caracter) = 93 Then corchete = corchete - 1
   Next i
   If (parentesis <> 0 Or corchete <> 0) Then
      flag1 = 2
      flag2 = i
   End If
End Sub
```

Código correspondiente al diagrama de flujo nº 4:

```
Public Sub variables_en_matriz(cadena, matriz_variables() As String, matriz_de_todas_las_variables() As
String, cadena_numerica() As Integer)
  Dim i, j, k, q, numero_variable, contador, long_cadena As Integer
  Dim detectando_variable, acabo_de_detectar_variable, variable_existe As Boolean
  Dim caracter As String
  ''''''''''''''''''''''''''''''''''''
  'combierto la cadena en numeros'
  ''''''''''''''''''''''''''''''''''''
  k = 1
  long_cadena = Len(cadena)
  detectando_variable = False
  acabo_de_detectar_variable = False
  numero_variable = 1
  q = 1
  For i = 1 To long_cadena + 1
    caracter = Mid(cadena, i, 1)
    If ((caracter >= Chr(40) And caracter <= Chr(43)) Or caracter = Chr(93) Or caracter = Chr(91) Or
caracter = "") Then
      'aquí he detectado un caracter especial
      If Not (detectando_variable) Then
        Select Case caracter
          Case Chr(40)
            cadena_numerica(k) = parentesis_izq
          Case Chr(41)
            cadena_numerica(k) = parentesis_dech
          Case Chr(42)
            cadena_numerica(k) = cte_and
          Case Chr(43)
            cadena_numerica(k) = cte_or
          Case Chr(91)
            cadena_numerica(k) = corchetes_izq
          Case Chr(93)
            cadena_numerica(k) = corchetes_derech
        End Select
        k = k + 1
      End If
      acabo_de_detectar_variable = detectando_variable
      detectando_variable = False
    Else
      matriz_de_todas_las_variables(q) = matriz_de_todas_las_variables(q) + caracter
      q = q + 1
      detectando_variable = True
      matriz_variables(numero_variable) = matriz_variables(numero_variable) + caracter
    End If
    If acabo_de_detectar_variable Then
      For j = 1 To numero_variable - 1
        'Si no es la primera variable miro si esta repetida
        If (matriz_variables(j) = matriz_variables(numero_variable)) Then
          matriz_variables(numero_variable) = ""
          cadena_numerica(k) = j
          k = k + 1
          Select Case caracter
            Case Chr(40)
              cadena_numerica(k) = parentesis_izq
            Case Chr(41)
              cadena_numerica(k) = parentesis_dech
            Case Chr(42)
              cadena_numerica(k) = cte_and
            Case Chr(43)
              cadena_numerica(k) = cte_or
            Case Chr(91)
              cadena_numerica(k) = corchetes_izq
            Case Chr(93)
```

```
                cadena_numerica(k) = corchetes_derech
            End Select
            k = k + 1
            Exit For
          End If
        Next j
        If matriz_variables(numero_variable) <> "" Then
          cadena_numerica(k) = numero_variable
          k = k + 1
          Select Case caracter
            Case Chr(40)
              cadena_numerica(k) = parentesis_izq
            Case Chr(41)
              cadena_numerica(k) = parentesis_dech
            Case Chr(42)
              cadena_numerica(k) = cte_and
            Case Chr(43)
              cadena_numerica(k) = cte_or
            Case Chr(91)
              cadena_numerica(k) = corchetes_izq
            Case Chr(93)
              cadena_numerica(k) = corchetes_derech
          End Select
          k = k + 1
          numero_variable = numero_variable + 1
        End If
        acabo_de_detectar_variable = False
      End If
    Next i
End Sub
```

Código correspondiente al diagrama de flujo nº 5:

```
Public Sub numero_de_niveles(funcion_numerica() As Integer, min, max, tamanio)
Dim i, profundidad, minimo, maximo As Integer
'''''''''''''''''
'numero de niveles'
'''''''''''''''''
   maximo = 0
   minimo = 1000
   profundidad = 100
   i = 1
   Do While funcion_numerica(i) <> 0
      If ((funcion_numerica(i) <> -1) And i = 1) Then minimo = profundidad
      If funcion_numerica(i) = -1 Then profundidad = profundidad + 1
      If (funcion_numerica(i) = -2 And funcion_numerica(i + 1) <> 0) Then profundidad = profundidad - 1
      If (profundidad > maximo) Then maximo = profundidad
      If (profundidad < minimo) Then minimo = profundidad
      i = i + 1
   Loop
   min = minimo
   max = maximo
   tamanio = i - 1 'cuento con el cero
End Sub
```

Código correspondiente al diagrama de flujo nº 6:

```
' k        puntero a la posicion que se esta explorando en la funcion numerica
' nivel_t   almacena el nivel temporal que se está explorando en la función
' p        puntero al vector que almacena las operaciones encontradas
' i        indica que tipo de operador se está explorando

Public Sub pinta(matriz_dibujo() As Integer, funcion_numerica() As Integer, fila, ByVal columna As Integer,
ByVal tamanio As Integer, max_fin_columna)
Dim k, nivel_t, p, i, j As Integer
Dim offset As Integer
Dim operadores(100) As Integer
Dim pos_operadores(100) As Integer
Dim hay_operadores, flag As Boolean
Dim sub_funcion_numerica_derecha(101) As Integer
Dim tamanio_izquierda As Integer
Dim tamanio_derecha As Integer
Dim sub_funcion_numerica_izquierda(101) As Integer
Dim num_operadores(-5 To 5)
Dim posicion_ruptura As Integer
Dim operador_intermedio As Integer

   If columna > max_fin_columna Then max_fin_columna = columna
   flag = True
   offset = 0
   Do While flag
      If funcion_numerica(offset + 1) = parentesis_izq And funcion_numerica(tamanio - offset) =
parentesis_dech Then
         j = 0
         For i = offset + 2 To tamanio - offset - 1
            If funcion_numerica(i) = parentesis_izq Then j = j + 1
            If funcion_numerica(i) = parentesis_dech Then j = j - 1
            If j = -1 Then
               flag = False
               Exit Do
            End If
         Next i
         If flag = True Then offset = offset + 1
      Else
         flag = False
      End If
   Loop
   p = 0
   hay_operadores = False
   'Eliminamos parentesis del principio y del final
   'miro el nivel
   nivel_t = 100
   For k = offset + 1 To tamanio - offset
      Select Case funcion_numerica(k)
         Case parentesis_izq
            nivel_t = nivel_t + 1
         Case parentesis_dech
            nivel_t = nivel_t - 1
         Case corchete_izq
            nivel_t = nivel_t + 1
         Case corchete_derech
            nivel_t = nivel_t - 1
      End Select
      'If k = fin And hay_operadores = False And funcion_numerica(k) < 0 Then nivel = nivel + 1
      'meto operadores en matriz
      If (nivel_t = 100) Then
         If ((funcion_numerica(k) = cte_and) Or (funcion_numerica(k) = cte_or) Or (funcion_numerica(k) =
corchete_izq)) Then
            operadores(p) = funcion_numerica(k)
            pos_operadores(p) = k
```

```
            hay_operadores = True
            p = p + 1
            num_operadores(funcion_numerica(k)) = num_operadores(funcion_numerica(k)) + 1
          End If
        End If
      Next k
      If Not hay_operadores Then
      'No hay operadores, entonces o hay una variable o hay corchetes
      'unicamente hay una variable entonces hay que actuar en consecuencia
        If funcion_numerica(offset + 1) = corchete_izq Then
        'hay que complementar
          posicion_ruptura = offset + 1
          matriz_dibujo(fila, columna) = corchete_izq
          j = 1
          For i = offset + 2 To tamanio - offset - 1
            sub_funcion_numerica_derecha(j) = funcion_numerica(i)
            j = j + 1
          Next i
          tamanio_derecha = tamanio - 2
          Call pinta(matriz_dibujo, sub_funcion_numerica_derecha, fila, columna + 1, tamanio_derecha,
max_fin_columna)
          Exit Sub
        Else
        'es un variable
          matriz_dibujo(fila, columna) = funcion_numerica(offset + 1)
          fila = fila + 1
          Exit Sub
        End If
      Else
        'hay que "partir" la funcion por el operador apropiado
        If num_operadores(cte_or) > 0 Then
        'Partimos por una operacion or
          operador_intermedio = Fix(CSng(num_operadores(cte_or)) / 2# + 0.5)
          j = -1
          i = 0
          Do While i < operador_intermedio
            j = j + 1
            If operadores(j) = cte_or Then i = i + 1
          Loop
          posicion_ruptura = pos_operadores(j)
          matriz_dibujo(fila, columna) = cte_or
        Else
        'partimos por una operacion and
          operador_intermedio = Fix(CSng(num_operadores(cte_and)) / 2# + 0.5)
          j = -1
          i = 0
          Do While i < operador_intermedio
            j = j + 1
            If operadores(j) = cte_and Then i = i + 1
          Loop
          posicion_ruptura = pos_operadores(j)
          matriz_dibujo(fila, columna) = cte_and
        End If
        'escribimos parte izquierda
        tamanio_izquierda = posicion_ruptura - 1 - offset
        j = 1
        For i = offset + 1 To posicion_ruptura - 1
          sub_funcion_numerica_izquierda(j) = funcion_numerica(i)
          j = j + 1
        Next i
        j = 1
        'escribimos parte derecha
        tamanio_derecha = tamanio - offset - posicion_ruptura
        For i = posicion_ruptura + 1 To tamanio - offset
          sub_funcion_numerica_derecha(j) = funcion_numerica(i)
          j = j + 1
        Next i
```

```
        'Hacemos la llamada por la izquierda
        Call pinta(matriz_dibujo, sub_funcion_numerica_izquierda, fila, columna + 1, tamanio_izquierda,
max_fin_columna)
        Call pinta(matriz_dibujo, sub_funcion_numerica_derecha, fila, columna + 1, tamanio_derecha,
max_fin_columna)
    'aquí hay que retornar
    End If 'Not hay_operadores
End Sub
```

Código correspondiente al diagrama de flujo nº 7:

```
Public Sub pinta_simetrica(matriz_dibujo() As Integer, matriz_simetrica() As Integer, fin_fila, fin_columna)
Const A = 100
Dim i, j, p, q, fila_final, columna_final, flag, distancia, mitad, posicion As Integer
Dim m_auxiliar(A) As Integer
Dim estoy_contando As Boolean

'meto filas alternas en blanco, asi coloco las variables
For i = 0 To fin_fila
   For j = 0 To fin_columna
      m_auxiliar(j) = matriz_dibujo(i, j)
   Next j
   'escribo en matriz simetrica

   For q = 0 To fin_columna
      matriz_simetrica((2 * i), q) = m_auxiliar(q)
   Next q
Next i

columna_final = fin_columna      'la ultima fila son todo vaiables por eso la quito
fila_final = 2 * (i - 1)         'por que he dejado un espacio en blanco
'obtengo la posicion media para las siguientes variables

 For j = columna_final To 1 Step -1
    For i = fila_final To 0 Step -1
      If matriz_simetrica(i, j) <> 0 Then '((matriz_simetrica(i, j) = cte_and) Or (matriz_simetrica(i, j) = cte_or)
Or (matriz_simetrica(i, j) = corchetes_izq)) Then
         estoy_contando = True
         distancia = 0
         For p = i - 1 To 0 Step -1
               If matriz_simetrica(p, j) <> 0 Then estoy_contando = False
               If estoy_contando = True Then distancia = distancia + 1
               If estoy_contando = False Then Exit For
               If ((p = 0) And (estoy_contando = True)) Then distancia = 0
         Next p
         If p = -1 Then p = 0
         mitad = Fix(CSng(distancia) / 2# + 0.5)
         posicion = i - mitad 'posicion del anterior
         If i <> 0 Then i = p + 1
         'bajo toda la fila
         If ((matriz_simetrica(p, j - 1) = cte_and) Or (matriz_simetrica(p, j - 1) = cte_or)) Then
            For q = j - 1 To 0 Step -1

               matriz_simetrica(posicion, q) = matriz_simetrica(p, q)
               matriz_simetrica(p, q) = 0
            Next q
         End If

      End If
   Next i
Next j
fin_fila = fila_final
fin_columna = columna_final + 1
End Sub
```

Código correspondiente al diagrama de flujo nº 8:

```
Public Sub pinta_simetrica_espaciada(matriz_simetrica() As Integer, matriz_simetrica_espaciada() As
Integer, fin_fila, fin_columna)
Const A = 100
Dim i, j, q As Integer
Dim m_auxiliar(A) As Integer

'meto filas alternas en blanco, asi coloco las variables
For j = 0 To fin_columna
   For i = 0 To fin_fila
      m_auxiliar(i) = matriz_simetrica(i, j)
   Next i
   'escribo en matriz simetrica

   For q = 0 To fin_fila
      matriz_simetrica_espaciada(q, (2 * j)) = m_auxiliar(q)
   Next q
Next j
fin_columna = (fin_columna * 2) - 2
End Sub
```

Código correspondiente al diagrama de flujo nº 9:

```
Public Sub rellena_simetrica_espaciada(matriz_simetrica_espaciada() As Integer, fin_fila, fin_columna)
Dim i, j, p, puntero, puntero2, veces, variable As Integer
'coloco el numero especifico para rellenar luego con los dibujos
   For j = 0 To fin_columna - 1 Step 2
      puntero2 = 0
      For i = 0 To fin_fila
         'If ((matriz_simetrica_espaciada(i, j) >= -5) And (matriz_simetrica_espaciada(i, j) < 0)) Then
puntero = i
         veces = 0
         Select Case matriz_simetrica_espaciada(i, j)
            Case -3, -4
               matriz_simetrica_espaciada(i, j + 1) = -10 'salida de la puerta
               puntero = i
               For p = puntero2 To fin_fila
                  If ((matriz_simetrica_espaciada(p, j + 2) >= -5) And (matriz_simetrica_espaciada(p, j + 2)
<> 0)) Then
                     If matriz_simetrica_espaciada(p, j + 1) = 0 Then
                        Select Case puntero
                           Case Is > p
                              matriz_simetrica_espaciada(p, j + 1) = -16 'curva izq arriba
                              veces = veces + 1
                           Case Is < p
                              matriz_simetrica_espaciada(p, j + 1) = -15 'curva izq a bajo
                              veces = veces + 1
                        End Select
                     End If 'matriz_simetrica_espaciada(p, j + 1) = 0
                  End If
                  puntero2 = p + 1
                  If veces = 2 Then Exit For
               Next p
            Case -5
               matriz_simetrica_espaciada(i, j + 1) = -13 ' linea horizontal
            Case Is > 0
               variable = matriz_simetrica_espaciada(i, j)
               For p = j To fin_columna
                  matriz_simetrica_espaciada(i, p) = -13 ' linea horizontal
                  If p = fin_columna Then matriz_simetrica_espaciada(i, p) = variable
               Next p
         End Select
      Next i
   Next j
   'coloco las lineas verticales
   For j = 1 To fin_columna - 1 Step 2 'comienzo en uno pq estas lineas estan en las columnas impares
      For i = 0 To fin_fila
         If (((matriz_simetrica_espaciada(i, j) = -16) And (matriz_simetrica_espaciada(i + 1, j) = 0)) Or
((matriz_simetrica_espaciada(i, j) = -14) And (matriz_simetrica_espaciada(i + 1, j) = 0)) Or
((matriz_simetrica_espaciada(i, j) = -10) And (matriz_simetrica_espaciada(i + 1, j) = 0))) Then
            matriz_simetrica_espaciada(i + 1, j) = -14
         End If
         'If matriz_simetrica_espaciada(i, j) = -15 Then Exit For
      Next i
   Next j
End Sub
```

Código correspondiente al diagrama de flujo nº 10:

```
Public Sub grafico(matriz_simetrica_espaciada() As Integer, matriz_de_todas_las_variables() As String,
fila_2, fin_columna, frmForm2 As Form2)
Dim i, j, pos_y, pos_x, posy, posx, p As Integer
Dim X As Picture 'declaro X como un objeto picture
   p = 1
   For j = 0 To fin_columna + 1
      For i = 0 To fila_2
         pos_y = 5 + ((fila_2 - i) * 47)
         pos_x = 5 + ((fin_columna - j) * 47)
         If ((j = 0) And (i <> 0) And (matriz_simetrica_espaciada(i, j) <> 0)) Then
            frmForm2.CurrentX = pos_x + 60
            frmForm2.CurrentY = pos_y + 17
            frmForm2.Print "F"
            posx = pos_x + 80
         End If
         Select Case matriz_simetrica_espaciada(i, j)
            Case -3
               Set X = LoadPicture(".\picture\and.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case -4
               Set X = LoadPicture(".\picture\or.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case -5
               Set X = LoadPicture(".\picture\not.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case -10
               Set X = LoadPicture(".\picture\dibujo0.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case -16
               Set X = LoadPicture(".\picture\dibujo2.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case -15
               Set X = LoadPicture(".\picture\dibujo1.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case -13
               Set X = LoadPicture(".\picture\dibujo3.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case -14
               Set X = LoadPicture(".\picture\dibujo4.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm2.PaintPicture X, pos_x, pos_y
            Case Is > 0
               frmForm2.CurrentX = pos_x + 30
               frmForm2.CurrentY = pos_y + 15
               frmForm2.Print matriz_de_todas_las_variables(p)
               p = p + 1
         End Select
      Next i
   Next j
   'establezco el tamaño del formulario
   frmForm2.Height = 47 * (posx / 2)
   frmForm2.Width = 47 * (posx / 3)
End Sub
```

Código correspondiente al diagrama de flujo nº 11:

```
Public Sub grafico_cuantico(matriz_simetrica_espaciada() As Integer, matriz_de_todas_las_variables() As
String, fila_2, fin_columna, frmForm1 As Form1)
Dim i, j, pos_y, pos_x, posx, p As Integer
Dim X As Picture 'declaro X como un objeto picture
   p = 1
   For j = 0 To fin_columna + 1
      For i = 0 To fila_2
         pos_y = 5 + ((fila_2 - i) * 79)
         pos_x = 5 + ((fin_columna - j) * 64)
         If ((j = 0) And (i <> 0) And (matriz_simetrica_espaciada(i, j) <> 0)) Then
            frmForm1.CurrentX = pos_x + 70
            frmForm1.CurrentY = pos_y + 32
            frmForm1.Print "F"
            posx = pos_x + 70
         End If
         Select Case matriz_simetrica_espaciada(i, j)
            Case -3
               Set X = LoadPicture(".\picture\c_and.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case -4
               Set X = LoadPicture(".\picture\c_or.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case -5
               Set X = LoadPicture(".\picture\c_inv2.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case -10
               Set X = LoadPicture(".\picture\c_dibujo0.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case -16
               Set X = LoadPicture(".\picture\c_dibujo5.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case -15
               Set X = LoadPicture(".\picture\c_dibujo4.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case -13
               Set X = LoadPicture(".\picture\c_dibujo2.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case -14
               Set X = LoadPicture(".\picture\c_dibujo1.bmp") 'asigno a X el objeto que esta en esa direccion
               frmForm1.PaintPicture X, pos_x, pos_y
            Case Is > 0
               frmForm1.CurrentX = pos_x + 50
               frmForm1.CurrentY = pos_y + 30
               frmForm1.Print matriz_de_todas_las_variables(p)
               p = p + 1
         End Select
      Next i
   Next j
   'establezco el tamaño del formulario
   frmForm1.Height = 64 * (posx / 2)  'para el ancho
   frmForm1.Width = 64 * (posx / 3)
End Sub
```

9. BIBLIOGRAFÍA

[1] E. Rieffel, W. Polak, "An introduction to Quantum Computing for Non-Physicists", ACM Computing Surveys, Vol 32, No. 3, September 2000, pp. 300-335.

[2] C. S. Lent, P. Douglas Tougaw, "A Device Architecture for Computing with Quantum Dots", Proceedings of the IEEE, Vol. 85, No. 4, April 1997.

[3] M. Calixto, "Computación Cuántica: un reto tecnológico".

[4] Andrew M. Steane, Eleanor G. Rieffel, "Beyond Bits: the Future of Quantum Information Processing",Computer IEEE, January 2000.

[5] Konstantin K. Likharev, "Single-Electron Devices and their Applications", Proceedings of the IEEE, Vol, 87, No. 4, April 1999.

[6] Ramón Muñoz Tapia, UAB, "Introducción a la Computación Cuántica", Apuntes de clase, Noviembre 1999,.

[7]Guillermo García Alcaine, "Teleportación: realidad y ficción", Revista Española de Física 12.

[8] Hillary Caituiro-Monge, "Arquitectura Cuántica", Electrical and Computer Engineering Departament, University of Puerto Rico, Mayagüez Campus, Mayagüez, Pueto Rico 00681-5215.

[9] Unai Agulera Irazabal, Ruben Gonzalez Lodeiro, "Al ternativas al Silício", http://www.int80h.net .

[10] Lamberto García del Cid, "La paradoja Einstein-Podolsky-Rosen y el teorema de Bell", http://www.elrincondelaciencia.com.

[11] S. Benjamin ,A. Ekert, "A short introduction to quantum-scale computing", http://www.qubit.org .

[12] A. Lloris, A. Prieto, "Diseño Lógico", McGraw Hill, 1996.

[13] Fco. J. Cevallos Sierra, "Visual Basic, Curso de Programación 2ª edición", Ra-ma, 1999.