Tema VI.

Rutado

6.1. Introducción

El último paso en la realización de un layout consiste en el rutado de las conexiones. Esta fase se puede definir de la siguiente forma:

> La fase de rutado consiste en conectar a través de diferentes capas de metal todos los terminales del sistema

Para ello, esta fase debe disponer de la siguiente información:

- Lista de nodos, indicando a qué nodos deben estar conectados todos los terminales de los diferentes módulos del sistema.
- La colocación óptima de los módulos del sistema.
- Las reglas tecnológicas que garantizarán la integración correcta de las diferentes conexiones.

Con esta información se pretende llegar a los siguientes objetivos:

- Conectar todos los nodos del sistema.
- Verificar las reglas tecnológicas.
- Maximizar las prestaciones del sistema.
- Minimizar el área ocupada.
- Invertir poco tiempo de CPU.

De tal forma que se obtenga el layout del sistema completo.

En la figura 6.1 se muestran las diferentes etapas por las que debe pasar la fase de rutado para obtener el layout final. Dichas etapas se muestran para el contador binario de cinco bits. En primer lugar se parte de una colocación buena, como puede ser una de las obtenidas en la técnica iterativa mostrada en el tema anterior. El siguiente paso será determinar los canales por los cuales irán las conexiones necesarias, para lo cual adecuaremos la orientación de las diferentes celdas. De esta forma determinaremos el canal por el cual irán los diferentes nodos (etiquetaremos los nodos por el número del bloque del que salen, excepto la señal de reloj clk). Así, el nodo clk irá por los canales C1, C4, C3 y C2, el nodo n1 irá por los canales C1, C2 y C3; el nodo n2 irá por los canales C1, C2 y C3; el nodo n3 irá por el canal C2; el nodo n4 irá por los canales C2 y C3; el nodo n5 irá por los canales C1, C2 y C4; el nodo n6 irá por el canal C3; y el nodo n7 irá por el canal C4. Una vez que hemos determinado los canales por los que pasarán cada uno de los nodos, obtendremos un rutado aproximado gracias al cual podremos determinar la ocupación (y por tanto el tamaño) de cada canal. Por último, dicho rutado aproximado se convierte en rutado detallado colocando líneas de metal (por lo general dos: una vertical y otra horizontal; para evitar las conexiones no deseadas en los cruces) de tal forma que se verifiquen las reglas tecnológicas de la tecnología (valga la redundancia) en la que se integrará el sistema. Existen una serie de nodos que se suelen tratar con especial importancia debido a la necesidad de evitar dispositivos parásitos en su camino o a la necesidad de equiparar dichos caminos; dichos nodos son los nodos de polarización, tierra y reloj (aunque en el ejemplo se haya tratado como un nodo más debido a la poca complejidad del sistema).

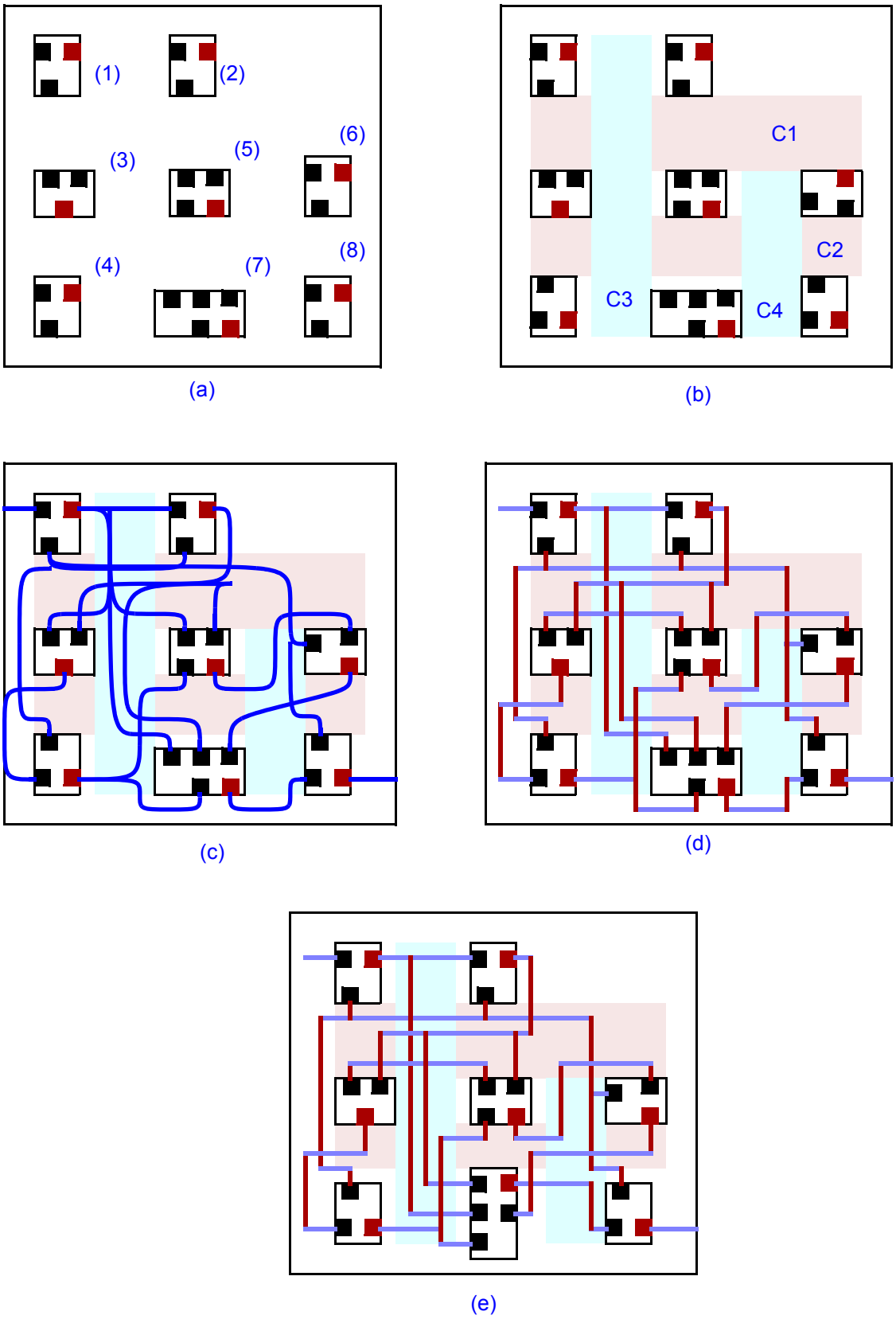

Figura 6.1.- Fases del rutado. (a) Colocación y posición de terminales. (b) Determinación de canales. (c) Rutado aproximado. (d) Rutado válido. (e) Rutado válido con otra orientación de celdas.

Todas estas etapas se suelen agrupar en dos fases de rutado: rutado global, que abarca desde la colocación hasta el rutado aproximado; y rutado detallado, que llegará hasta la obtención del layout final.

Seguidamente vamos a dar una serie de definiciones útiles en la fase de rutado de un circuito integrado.

> Un canal es la región existente entre dos módulos, por la cual irán las conexiones necesarias.

La característica que tienen los canales es que únicamente tienen conexiones en dos de sus lados (ambos opuestos), tal como se puede apreciar en la figura 6.2.

> Un switchbox (caja de conmutación) es la región de rutado que puede tener conexiones en sus cuatro lados.

Estas cajas se pueden definir por la interconexión de canales, como se muestra en la figura 6.2. La existencia de un switchbox indicará la presencia de otro tipo de región.

> Un L-canal es la intersección de dos canales perpendiculares.

Dicho L-canal puede ser descompuesto en dos canales y un switchbox (que correspondería a la zona de intersección de los canales).

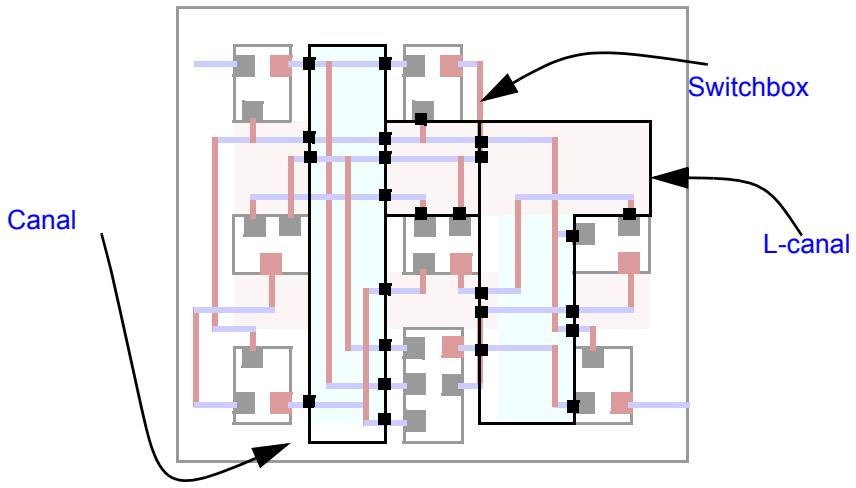

Figura 6.2.- Ejemplo de regiones en el layout del contador de cinco bits.

Por último, en el caso de layout basado en celdas estándares, las cuales están dispuestas en hileras, existe un tipo de celda especial (a la que ya se ha hecho alusión) dedicada al rutado de conexiones.

> Una celda de feedthrough es una celda vacía, por la cual se permite el paso de conexiones entre hileras de celdas no adyacentes.

La existencia de este tipo de celdas viene motivada por la existencia de más de dos filas; puesto que en esta situación se puede dar el caso de tener que conectar celdas en filas no adyacentes. En este caso es necesario atravesar una fila con el problema de realizar conexiones no deseadas con determinadas celdas. No obstante, para permitir la conexión intrínseca de los nodos de polarización y tierra, y mantener la uniformidad del nwell de toda una fila; estas cel-

das vacías también disponen de estos elementos, es decir, de nodos de polarización, tierra y de nwell. Otra utilidad que tienen estas celdas es la de mantener la misma longitud en todas las filas. En la figura 6.3 mostramos un ejemplo de cuatro celdas feedthrough utilizadas para permitir el paso de una conexión entre dos canales y unificar la longitud de todas las filas.

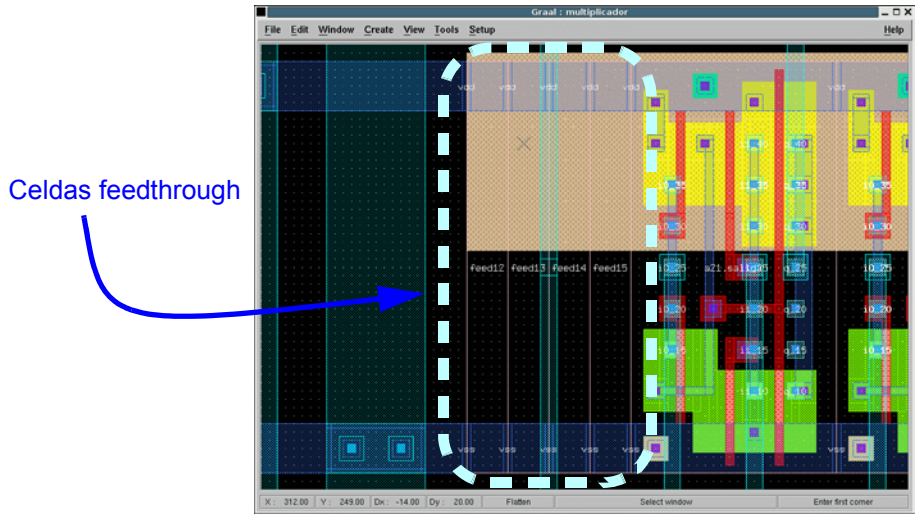

Figura 6.3.- Ejemplo de una celda feedthrough.

Otro elemento importante es la forma de representar el sistema para adecuar el paso de las diferentes conexiones. Existen dos modelos muy utilizados: modelo de grafo de rejilla (grid graph) y modelo de grafo de intersección de canal (channel intersection graph).

El grafo de rejilla divide toda el área del sistema en vértices, que indicarán el tipo de elemento al que corresponden: celdas, canales y terminales. Los canales son identificados con vértices claros, y por ellos podrán colocarse conexiones. Los terminales son identificados con vértices semioscuros (rojos), y por ellos sólo se permite la conexión de su propio nodo. Las celdas son identificadas con vértices oscuros, y por ellos no se permite el paso de conexiones. Un ejemplo de este modelo se muestra en la figura 6.4a. En ella podemos apreciar la colocación y orientación de celdas superpuesto con el modelo de rejilla. Para unificar el tamaño de todos los vértices de la rejilla, el tamaño de los módulos no tienen porque coincidir con los límites de los vértices.

El grafo de intersección de canal consiste en dividir el área en ejes (cada eje equivaldrá a un canal del sistema). Sobre estos ejes se colocarán los vértices, que indicarán los terminales de los módulos (vértices semisombreados en rojo) y cruces de canales (vértices claros). Como puede darse el caso de que dos terminales diferentes ocupen la misma posición en el eje, es decir, el mismo vértice, se define el peso del canal como la capacidad que debe tener. Una aproximación de este peso es el número de terminales que ocupan un mismo vértice. Un ejemplo de este modelo se muestra en la figura 6.4b.En ella podemos apreciar la colocación y orientación de celdas superpuesto con el modelo de intersección de canal. En ella podemos ver el peso de cada canal; en el caso de que el peso aproximado sea uno, dicho valor no aparecerá al lado del canal en cuestión.

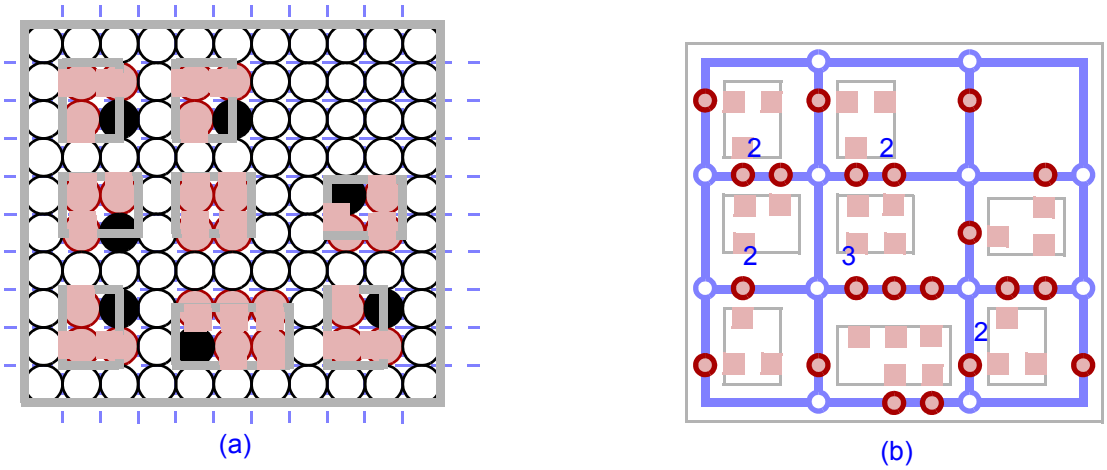

Figura 6.4.- Ejemplos de modelos de sistema para rutado. (a) Grafo de rejilla y (b) grafo de intersección de canales

6.2. Conexionado global

Entre las diferentes formas de definir el problema de conexionado o rutado global podemos utilizar la siguiente forma:

> Dado un netlist (lista de nodos) y el grafo de routing, el rutado global consiste en encontrar el árbol de Steiner que contenga todas las conexiones de tal forma que el número de conexiones de un canal sea inferior a su capacidad, y la longitud total de las conexiones sea mínima.

El algoritmo más utilizado en el rutado global es el algoritmo del laberinto (maze). En este punto vamos a realizar una distinción entre conexiones punto a punto y conexiones multipunto. En el caso de la conexión punto a punto, los pasos a seguir en el algoritmo del laberinto son las siguientes:

- Identificar en el grafo de routing los puntos inicial y final.
- Etiquetar los vértices accesibles con la distancia normalizada (según la distancia de Manhatan), hasta llegar al punto final. Obviamente el etiquetado suele ser secuencial (según la distancia) partiendo del punto inicial.
- Una vez que llegamos al punto final, obtenemos el camino eligiendo los vértices etiquetados en orden decreciente.

Para iniciar una nueva conexión, los vértices ocupados por la conexión permanecerán en blanco para permitir su uso en una nueva conexión. Cuando se han conectado todos los nodos, el número de veces que ha sido ocupado el vértice de un canal indicará la capacidad que deba tener. En el caso de superarse la capacidad máxima, los vértices de dicho canal serán oscurecidos para evitar el paso de nuevas conexiones.

En la figura 6.5 mostramos dos ejemplos de la aplicación del algoritmo del laberinto para el contador binario de cinco bits. En ambos casos, partimos identificando los puntos inicial (etiquetado como I) y final (etiquetado como F). El siguiente paso consiste en etiquetar con ‘1’ a todos los vértices adyacentes y utilizables al punto inicial. Seguimos etiquetando con ‘2’ a

todos los vértices adyacentes y utilizables a aquellos etiquetados con ‘1’. Generalizando, se etiquetarán con ‘n+1’ a todos los vértices adyacentes y utilizables a aquellos etiquetados con ‘n’. Repetiremos el paso anterior hasta llegar al punto final (‘F’). Una vez que hemos llegado al punto final, vamos uniendo los vértices adyacentes cuyas etiquetas estén siempre en orden decreciente hasta llegar al punto inicial. Esta unión se corresponderá con la conexión deseada.

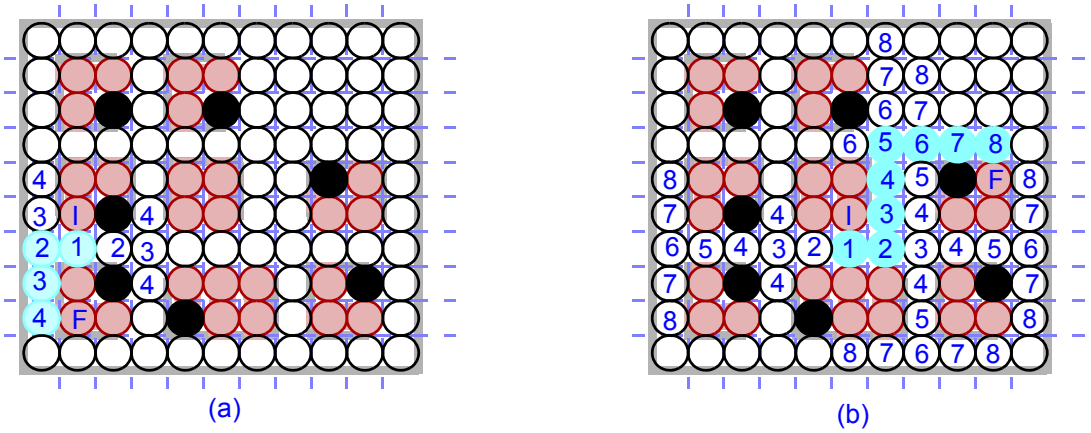

Figura 6.5.- Ejemplos de aplicación del algoritmo del laberinto con conexiones de punto a punto. (a) Conexión del nodo n3 y (b) conexión del nodo n5.

En el caso de la figura 6.5b, podemos elegir dos posibles caminos: uno al lado del biestable; y otro al lado de la puerta AND de tres entradas. Esta posibilidad viene determinada por la diferencia de tamaño de los módulos que ha provocado que parte de dicho canal sea más ancho. La elección entre un camino u otro dependerá de la capacidad del canal, ya que se puede aprovechar dicho aumento para introducir más señales si es necesario.

Para el caso de la conexión multipunto, vamos a basarnos en la conexión punto a punto. Los pasos que vamos a seguir son los siguientes:

- Se identifican todos los terminales que deben ser conectados.
- Se realiza una primera conexión entre dos de dichos terminales como si se tratase de una conexión punto a punto.
- Todos los vértices de la conexión pasan a tener una etiqueta de ‘0’, y se obtiene un nuevo camino desde uno de los puntos anteriores hasta uno nuevo.
- Se repite el paso anterior hasta que todos los puntos han sido conectados.

En la figura 6.6 mostramos un ejemplo de la aplicación del algoritmo del laberinto a una conexión multipunto. En este caso hemos elegido el rutado de la señal de reloj, la cual ha sido rutado como una señal más debido a que el número de conexiones es bajo. En primer lugar hemos obtenido la conexión de los terminales de los biestables 1 y 2 (que se encuentran en la fila superior). Una vez que hemos obtenido dicha conexión, hemos etiquetado todos los vértices de dicho conexión con distancia ‘0’, y hemos obtenido las distancias para llegar al terminal del biestable 4 (situado en la esquina inferior izquierda). Como dicha conexión se tiene que realizar por el lado superior, hemos elegido el vértice de ese lado en lugar del vértice del lado izquierdo. Se ha vuelto a repetir la operación con los terminales de los biestables 6 y 8 con unas consideraciones similares. Hay que tener en cuenta que la conexión no tiene porqué seguir el orden elegido, es decir, para conectar el terminal del biestable 6, se ha obtenido que se debe partir de la conexión del biestable 2 en lugar del biestable 4.

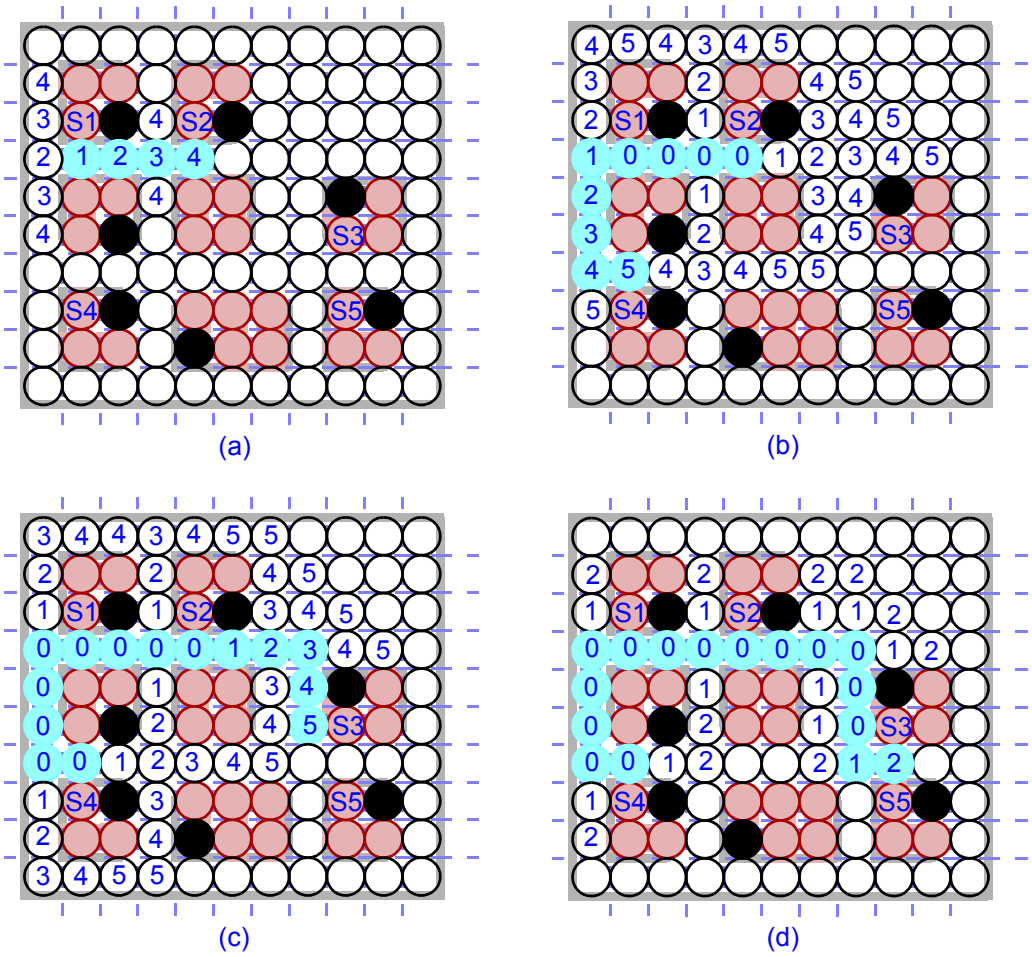

Figura 6.6.- Ejemplos de aplicación del algoritmo del laberinto con conexión multipunto (nodo reloj). (a) Conexión del módulo 1 y 2. (b) Conexión con el módulo 4. (c) Conexión con el módulo 6. (d) Conexión con el módulo 8.

En la figura 6.7 mostramos el ejemplo del rutado global del contador binario de cinco bits, indicando la capacidad que deben tener todos los canales. A los canales que indicamos anteriormente, le hemos añadido los canales laterales, es decir, los canales Cc1 y Cc2 para las columnas, y Cr1 y Cr2 para las filas. Por orden, mostramos el rutado global de las conexiones de clk, n1, n2, n3, n4, n5, n6 y n7. Con esta información podemos obtener el límite máximo de la capacidad de cada uno de los canales como el número de conexiones que se pasan por un canal. Dicho número se muestra en la tabla 6.1. No obstante, este número puede no ser demasiado significativo, ya que en el cálculo de la capacidad real hay que eliminar las conexiones que no se solapen, y por lo tanto puedan coexistir en el mismo nivel de conexión. Por ejemplo, en el canal 4, con un límite igual a 7, pueden coexistir las siguientes conexiones:

- Los nodos n3, n4 y n5 en un mismo nivel,
- los nodos clk y n1 en un mismo nivel. Si además es cierto que el canal 2 es más ancho en su primer tramo, también podría coexisitir el nodo n2 con los anteriores (situación que hemos considerado en la tabla).
- El nodo n6 en otro nivel.

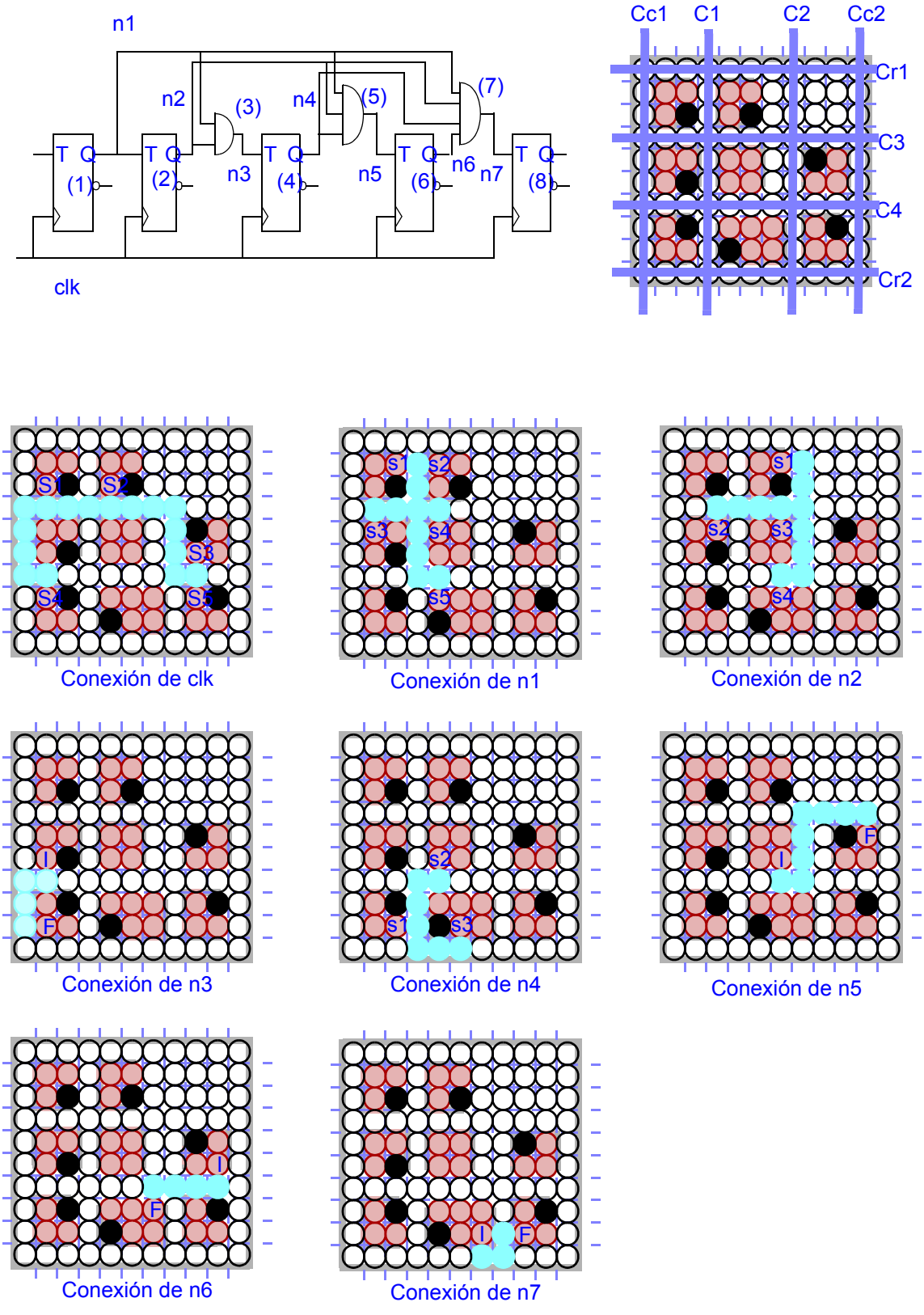

Figura 6.7.- Ejemplo de rutado global del contador binario de cinco bits

Canales	Nodos								Límite	Capacidad
	clk	n1	n2	n3	n4	n5	n6	n7		
Cc1	X			X					2	2
C1		X			X				2	2
C2	X		X			X		X	4	3
Cc2									0	0
Cr1									0	0
C3	X	X	X			X			4	3
C4	X	X	X	X	X	X	X		7	3
Cr2					X			X	2	1

Tabla 6.1. Obtención del límite máximo de la capacidad de cada canal.

6.3. Conexionado de canal

En el conexionado de canal o detallado, se considera las diferentes capas de las conexiones para poder identificarlas. En un canal podemos encontrarnos con los elementos mostrados en la figura 6.8.

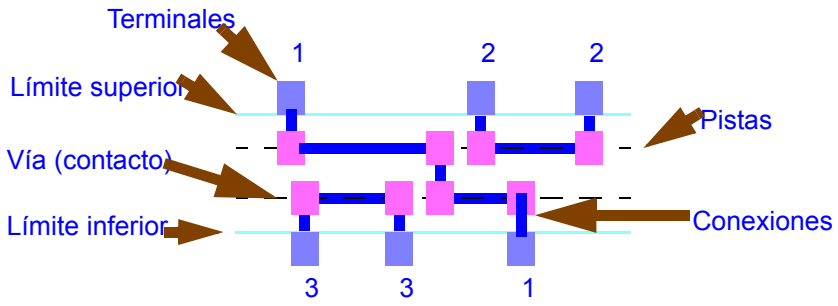

Figura 6.8.- Elementos de un canal

- Terminales. Son los puntos de conexión de los módulos, que estarán a uno de los lados del canal.
- Límites superior e inferior. Son las líneas que delimitan la altura de un canal
- Pistas. Son los diferentes niveles de un canal (paralelos a los límites) por las cuales pueden existir líneas de conexión.
- Conexiones. Son líneas que conectan una pista con un terminal o con otra pista.
- Son los elementos que conectan las líneas de las pistas con las conexiones.

El problema del rutado detallado consiste en asignar las pistas y conexiones de tal forma que la altura del canal sea minimizada. La anchura no hace falta ser minimizada porque viene determinada por los diferentes módulos y switchboxes que deben conectar.

Las vías son necesarias debido al hecho de que las líneas que van por las pistas y por las conexiones son capas de metal diferentes, para que los posibles cruces entre ellas no produzcan

conexiones no deseadas. Luego, las conexiones deseadas de dichas capas de metal diferentes son realizadas por las vías, como podemos apreciar en la figura 6.9. En ella podemos apreciar dos capas de metal, las cuales están a diferentes alturas para evitar una conexión no deseada. Realmente entre las capas de metal existe óxido; pero para poder apreciar las diferentes capas, dicho óxido no se muestra. Únicamente se produce una conexión cuando colocamos una vía, con la cual se perfora el óxido y se rellena de metal conectando los dos niveles (alturas) de metal.

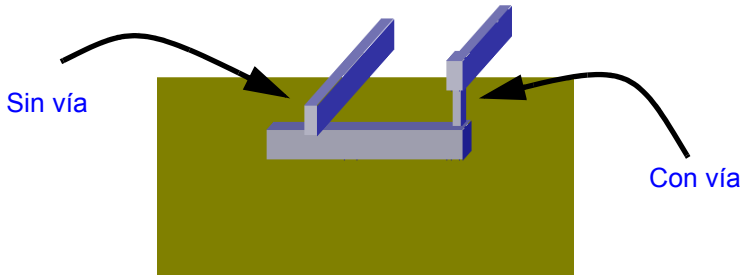

Figura 6.9.- Ejemplo de un cruce de dos capas de metal.

Atendiendo a esta situación, podemos encontrar dos tipos de modelos de rutado en función del número de capas de metal utilizadas: el modelo de dos capas y el modelo de tres capas, mostrados en la figura 6.10.

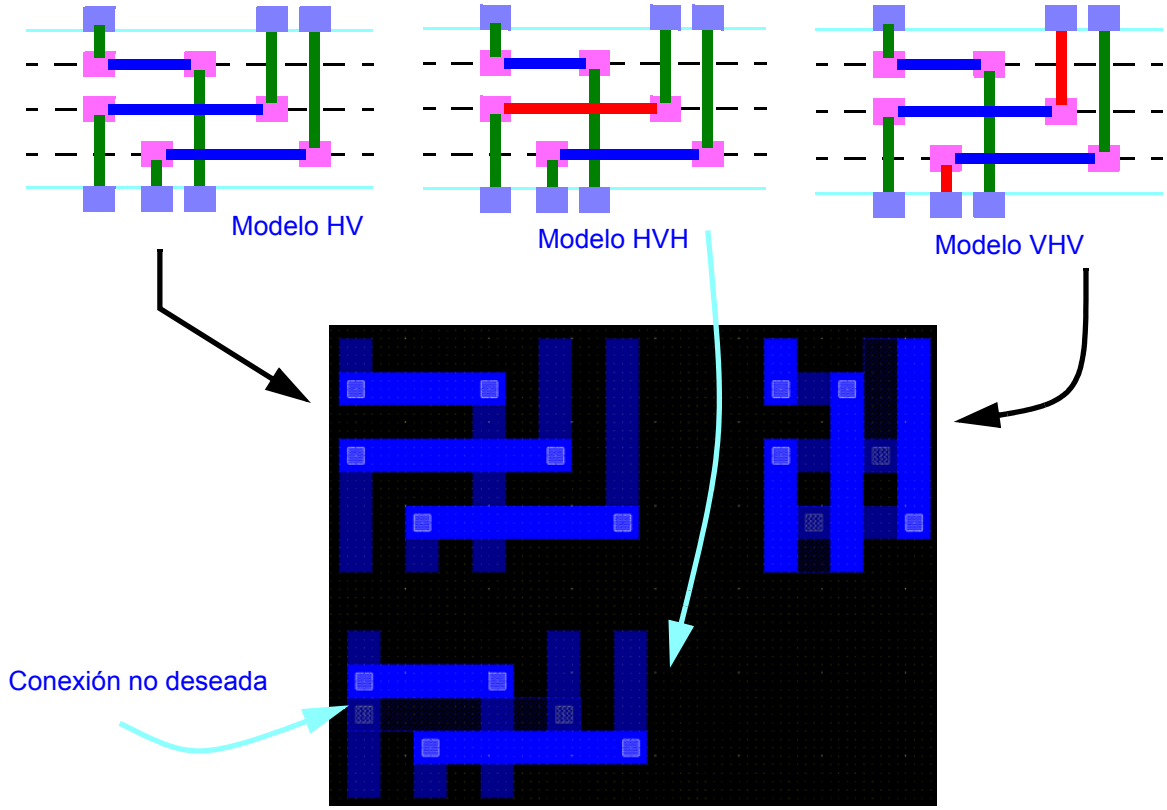

Figura 6.10.- Modelos de rutado.

El modelo de dos capas es el más utilizado y suele denominarse por HV o VH. En este modelo se utiliza una capa para las líneas horizontales y otra capa para las líneas verticales. En este caso la distancia entre pistas y conexiones es la distancia mínima que hay que dejar entre dos líneas de la misma capa para garantizar que no se produzca conexión.

El modelo de tres capas suele denominar HVH o VHV. En este modelo se utilizan dos capas de metal para una dirección: HVH (en la horizontal) o VHV (en la vertical); y una sola capa en la dirección complementaria. La ventaja de este modelo es que podemos reducir la distancia entre pistas; ya que en la dirección en la que están las dos capas, podemos introducir tres pistas (caso HVH) en el espacio en el que habría únicamente dos pistas en el modelo de dos capas. No obstante, dicha afirmación no es del todo cierta ya que si debe existir una vía a la misma altura en dos capas diferentes, se produce una conexión no deseada al juntar dos capas del mismo material (el metal de las conexiones) en la vía como se puede apreciar en la figura 6.10.

Para obtener la capacidad real de un canal, debemos considerar los segmentos (porción de una capa de metal, ya sea en las pistas o en las líneas de conexión) que pueden convivir en un mismo nivel. Una de las formas de obtener la convivencia de estos segmentos es creando el grafo de restricciones.

> El grafo de restricciones es un grafo en el que los vértices son los terminales presentes en el canal, los cuales se unen si no pueden convivir en un mismo nivel.

Existen dos versiones de dicho grafo: una para las líneas horizontales (denominado HCG), y otro para las líneas verticales (denominado VCG).

En la figura 6.11 se muestra un ejemplo de cómo obtener los grafos de restricciones de un canal del contador binario de cinco bits. Tendremos dos grafos de restricciones: HCG y VCG. En el caso de HCG, tenemos que considerar todos los segmentos (líneas que unen los terminales que deben ser conectados). Para ello, colocaremos todos los segmentos en niveles diferentes. Como los nodos 2 y 4 tienen sus terminales perpendicularmente, no contienen ningún segmento. El grafo tendrá cinco nodos correspondientes a cada uno de los nodos involucrados en el canal. Únicamente uniremos los nodos cuyos segmentos estén solapados en el hipotético caso de que hubiese un solo nivel.

Para el caso del VCG, prolongaremos líneas perpendiculares al canal a partir de cada uno de los terminales. De nuevo el grafo tendrá tantos nodos como nodos estén involucrados en el canal. Se unirán los nodos que tengan la misma perpendicular, es decir, aquellos nodos que tengan terminales en la misma posición pero en lados opuestos. En nuestro caso particular, el grafo no tendrá ninguna unión ya que no se cumple el requisito anterior para ninguno de los nodos.

Para obtener el rutado detallado podemos aplicar una serie de algoritmos. Entre estos algoritmos vamos a destacar el algoritmo LEA (Left- Edge Algorithm) y el algoritmo basado en cliques.

El algoritmo LEA impone una serie de limitaciones y acciones necesarias, las cuales son las siguientes:

- El modelo de rutado utilizado es un modelo de dos capas HV.

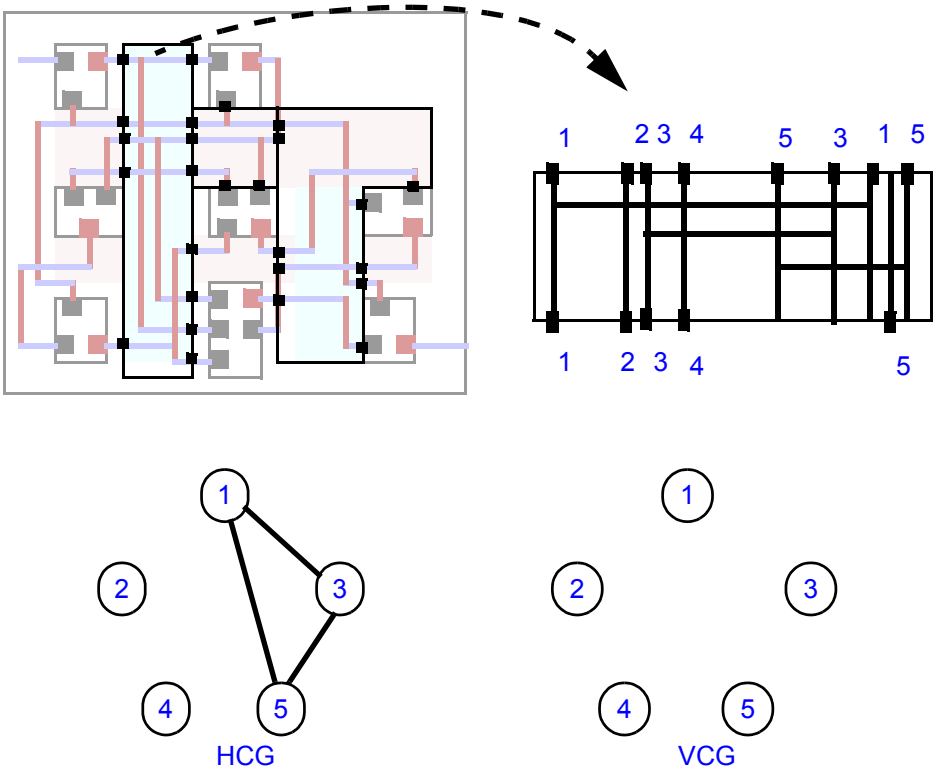

Figura 6.11.- Ejemplo del desarrollo de los grafos de restricciones.

- Los nodos serán clasificados por el eje X empezando po el terminal de la izquierda (de ahí el nombre del algoritmo).
- Se rutarán de forma secuencial todos los nodos siguiendo el orden anterior.
- Las pistas son buscadas desde la superior hasta la inferior, asignando la primera pista que pueda albergar el nodo.
- No se permiten nodos con multipista ni con VCG
- Produce soluciones con un número mínimo de pistas.

El algoritmo consta de los siguientes pasos:

- Se crea una primera pista.
- Se asigna el primer nodo a dicha pista.
- Se comprueba si el segmento del nuevo nodo se solapa con el ya colocado. En caso de que no exista solapamiento, se asigna dicha pista al nuevo nodo; en caso contrario, se crea una nueva pista y se la asigna al nuevo nodo.
- Se repite el paso anterior con todos los nodos, empezando a chequear solapamientos por la primera pista.
- Una vez que estén asignados todos los segmentos, se conecta el terminal izquierdo y derecho con la pista.

En la figura 6.12 mostramos un ejemplo de aplicación del algoritmo LEA. En primer lugar generamos los dos grafos de restricciones, y comprobamos que el VCG nos indica que no existen restricciones verticales, y por tanto se puede aplicar el algoritmo LEA. El primer paso consiste en etiquetar los terminales del límite superior empezando por la izquierda. Creamos una pista y colocamos sobre ella el segmento del nodo 1. Posteriormente comprobamos si existen incompatibilidades entre los segmentos de los nodos 1 y 2. El HCG me indica que no existen incompatibilidades, por lo que colocamos el segmento del nodo 2 en la misma pista que el nodo 1. No obstante, hay que indicar que el nodo 2 no tiene segmento por lo que no aparecerá en la gráfica. Realizamos la misma operación con el nodo 3, comprobando que existen incompatibilidades con el nodo 1. Como no existen más pistas, creamos una nueva pista y colocamos en ella el segmento del nodo 3. Seguimos con el nodo 4, y comprobamos que existe un comportamiento similar al del nodo 2, por lo que hipotéticamente el segmento del nodo 4 estaría en la pista del nodo 1, pero no existe tal segmento. En el caso del nodo 5, seguimos un procedimiento similar empezando por la pista 1 (comprobando la incompatibilidad con el nodo 1, que es el que está en la pista 1), que existe incompatibilidad; seguimos con la pista 2, que también existe incompatibilidad; y como no hay más pistas, creamos una nueva y colocamos en ella el segmento del nodo 5. Por último conectamos los extremos de cada segmento con los terminales correspondientes.

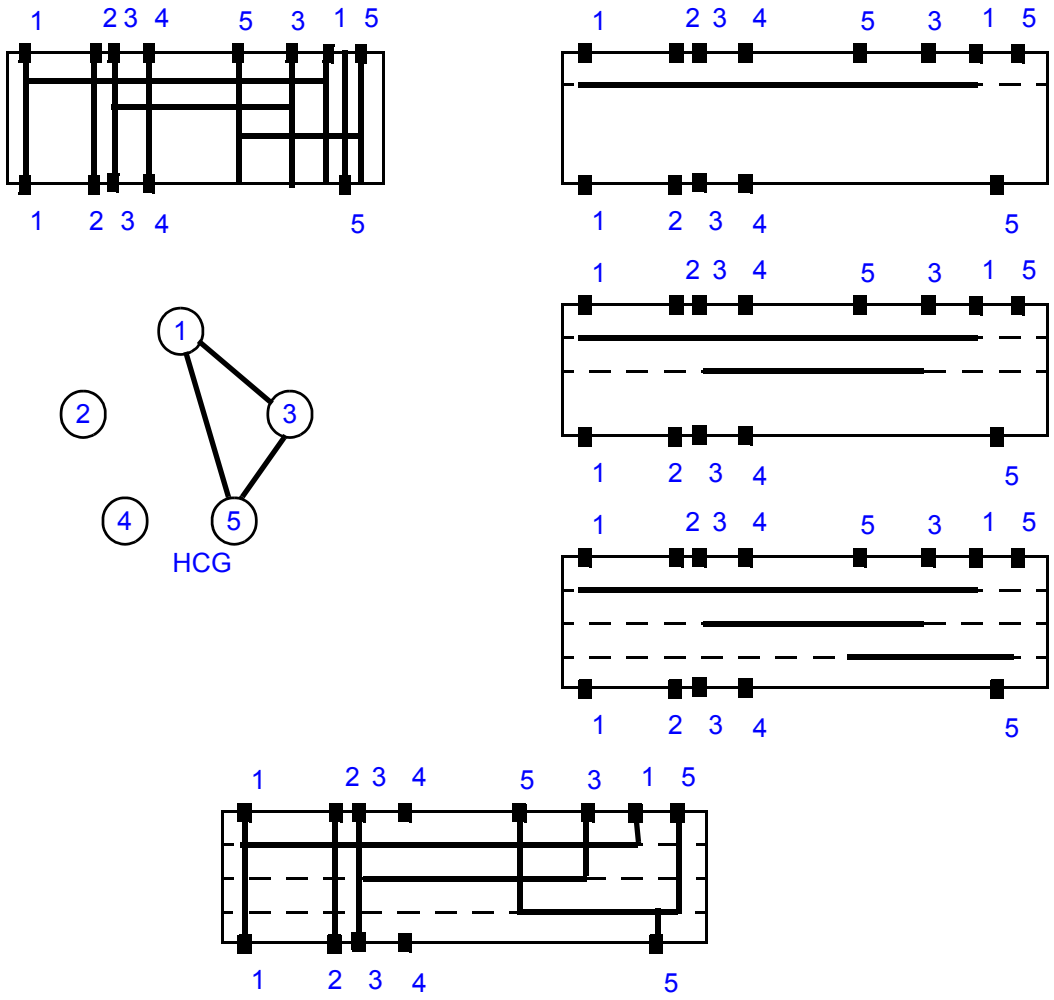

Figura 6.12.- Ejemplo de aplicación del algoritmo LEA.

En la figura 6.13 mostramos otra aplicación del algoritmo LEA, pero con la posibilidad de que una misma pista sea compartida por más de un nodo. si pasamos directamente a obtener la posición del nodo 3, podemos observar en el HCG que no existe compatibilidad con el nodo 1, por lo tanto se colocará en la misma pista que ocupa dicho nodo. De esta forma siempre vamos a obtener el número mínimo de pistas.

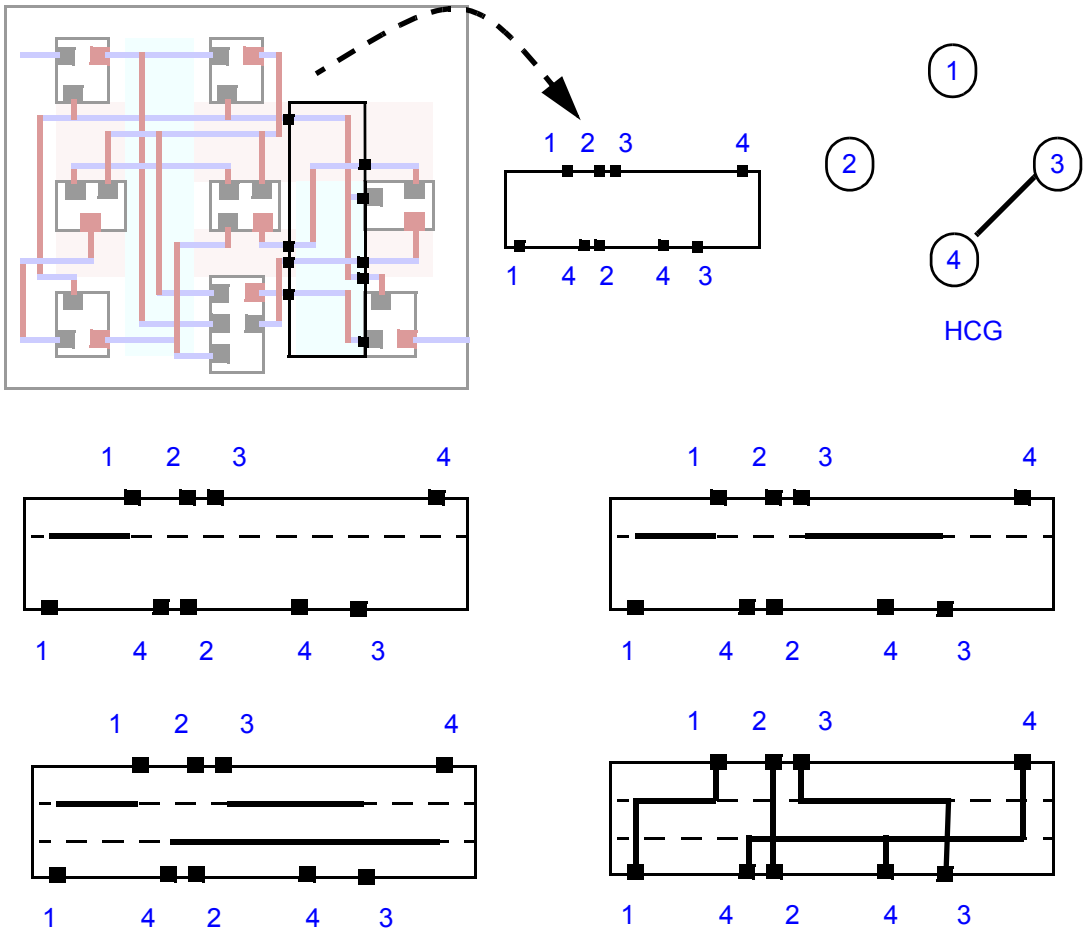

Figura 6.13.- Ejemplo del algoritmo LEA con posibilidad de compartir pistas.

El siguiente algoritmo que vamos a considerar es el algoritmo basado en cliques. Para ello vamos a dar unas definiciones previas.

> Un clique es una porción de grafo tal que de todos los nodos parte al menos un eje que irá al resto de nodos de la misma porción. El clique mínimo será un solo nodo del grafo.
>
> El grafo de compatibilidad es el grafo complementario a la unión de los dos grafos de restricciones (HCG y VCG)

El algoritmo consta de los siguientes pasos:

- Añadir la primera pista
- Construir el grafo de compatibilidad del canal

- Por cada clique máximo encontrado en dicho grafo, obtener el peso como la diferencia entre el número de terminales del clique en el límite superior y en el límite inferior.
- Mientras haya cliques,
  - Buscar el clique con más peso
  - Colocar el clique en la pista no utilizada
  - Eliminar los nodos del clique del grafo
  - Añadir una nueva pista.
- Eliminar la última pista (que nunca será utilizada).

En la figura 6.14 mostramos un ejemplo de aplicación del algoritmo basado en cliques. En primer lugar obtenemos los grafos de restricciones HCG y VCG. Estos dos grafos los uniremos para después complementarlos, y así obtener el grafo de compatibilidad. De este grafo de compatibilidad, obtenemos los cliques máximos los cuales estarán formados por los nodos {1, 2, 4}, {2, 3, 4}, {2, 4, 5}. Los nodos pertenecientes a un mismo clique me indican los nodos que pueden compartir una misma pista, por lo que el número de cliques será el número de pistas. Se obtienen el peso de cada clique, comprobándose que todos los cliques tienen el mismo peso igual a 1; por lo tanto, el orden de elección de cada clique es indiferente. En la primera pista se han colocado los segmentos y conexiones relativas a los nodos 1, 2 y 4. En la segunda pista se ha colocado el segmento y conexiones relativa al nodo 3 (puesto que los nodos 1 y 2 ya se colocaron en el clique anterior). En la tercera pista se ha colocado el segmento y conexiones relativas al nodo 5.

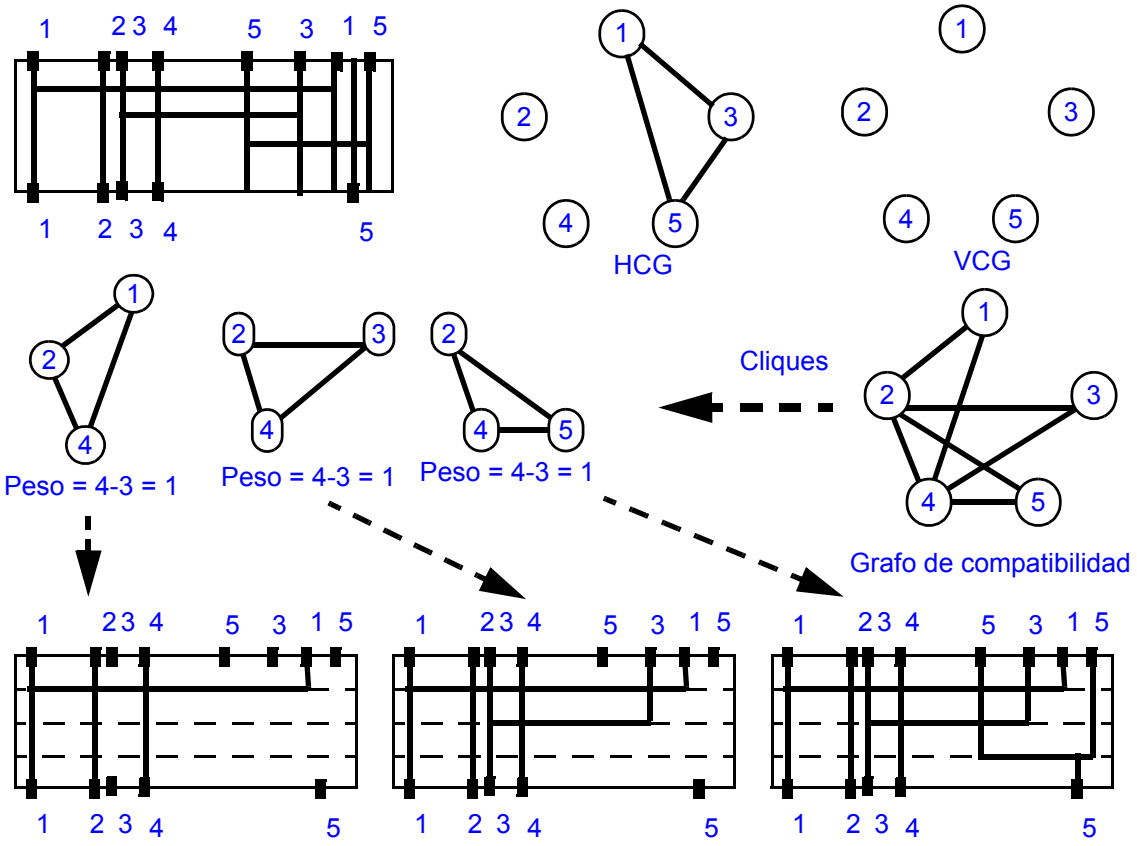

Figura 6.14.- Ejemplo del algoritmo basado en cliques.