Tema V.

Colocación

5.1. Introducción

Una vez que tenemos dividido el sistema en sus diferentes partes, viene la tarea de colocación. Dicha tarea se puede definir del siguiente modo:

> La tarea de colocación (o *placement*) de un sistema electrónico consiste en construir un layout indicando las posiciones de los diferentes módulos.

Los criterios de calidad de dicho layout son el área total y el rutado, de tal forma que cuanto menor sea el área ocupada por el sistema, y las líneas de conexión de los diferentes módulos, mayor será la calidad de la colocación. Estos dos parámetros se encuentran íntimamente relacionados con las prestaciones ya que, por ejemplo, cuanto menores sean las líneas de conexión, menores serán los dispositivos parásitos aumentando las prestaciones del sistema. Otros criterios utilizados para medir la calidad son los siguientes: no se debe permitir el solapamientos de módulos; los módulos deben estar dentro de los límites del sistema; los módulos deben estar colocados en un lugar apropiado (como una fila para las celdas estándares o en el grid para los dispositivos configurables).

En la figura 5.1 podemos apreciar dos colocaciones diferentes de un mismo sistema implementado en una FPGA (la línea roja es una línea de conexión que estaba seleccionada en el momento de la captura de la ventana). En las dos colocaciones podemos apreciar las diferencias que conducen a unas líneas de conexión (rutado) de diferente tamaño. Una consecuencia directa en las prestaciones consiste en que las líneas de mayor longitud provocan un mayor retraso en la señal que pasa por ellas por lo que las dos colocaciones producirán sistemas con diferente velocidad.

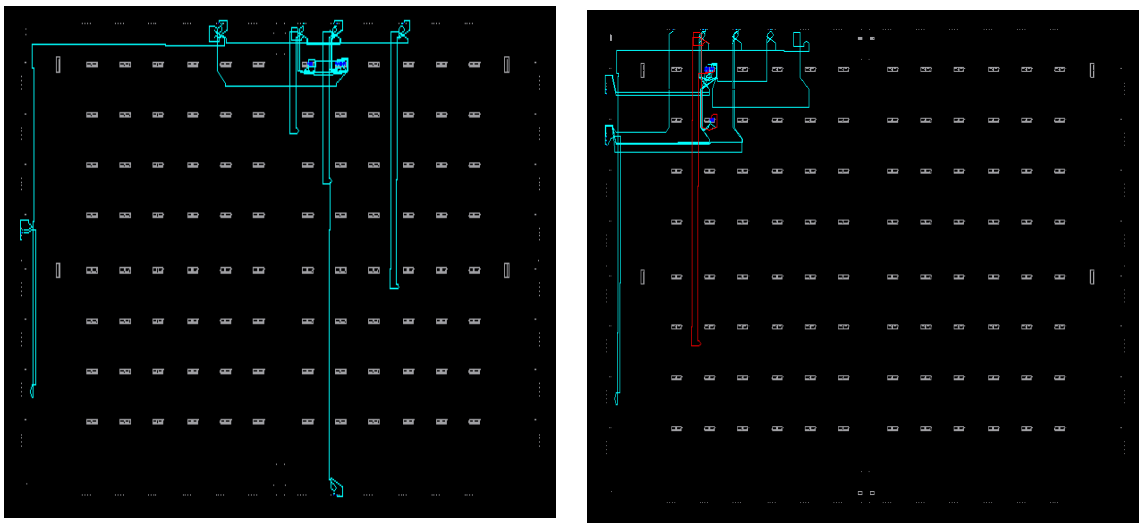


Figura 5.1.- Ejemplos de colcoaciones diferentes para un mismo sistema.

Una información necesaria que se debe conocer antes de la colocación es la forma de imlpementar los módulos que componen nuestro sistema, ya que será dicha implementación la que realmente estará en el layout. Aunque en un estado inicial podamos tratar todos los módu-

los como puntos sin dimensión, siempre debemos llegar a modelos más exactos en los que los módulos sí tendrán unas determiandas dimensiones, es más también habrá que saber cuál es la posición de cada una de las señales en dicho módulo.

Un primer estilo de layout consiste en la utilización de macrobloques, los cuales implementan los diferentes módulos del sistema. Este estilo es utilizado cuando llevamos a cabo un layout full-custom (al menos en la implementación de los módulos). La característica de estos macrobloques es que son de forma y tamaño irregulares, por lo que no pueden rellenar de forma homogénea una superficie. Los espacios que quedan entre los diferentes macrobloques son aprovechados para las conexiones entre los diferentes nodos.

Con la llegada de las herramientas de diseño automático, y para facilitar su tarea, llegamos a la utilización de celdas estándares. Una celda estándar son módulos lógicos con un layout interno prediseñado, el cual no puede ser alterado. La principal característica es que todas las celdas disponen de la misma altura (y por lo tanto, diferente anchura) y las conexiones de polarización y tierra se encuentran en la misma posición. Luego si unimos físicamente dos celdas, ya se está procudiendo la conexión de polarizaciones y tierra. Además, el resto de terminales suelen tener una conexión en los dos lados de la celda para permitir su conexión por arriab o por abajo, utilizando el espacio existente entre las diferentes filas de celdas.

Con la llegada de los dispositivos reconfigurables, también podemos optar por ellos. En este caso, el circuito estará compuesto de módulos (que suelen implementar funciones con un número determinado de entradas y salidas), los cuales no están prediseñados sino que están prefabricados en el chip como una matriz rectangular. Por lo tanto, el proceso de diseño se limita a seleccionar los diferentes módulos para las distintas funciones y realizar las conexiones necesarias.

Estas implementaciones nos proporciona una clasificación de los tipos de colocación utilizados, en función de las dimensiones a tratar: colocación 0-dim, 1-dim y 2-dim. En el caso de la colocación 0-dim, los diferentes módulos son tratados como puntos adimensionales, de tal forma que se colocan sobre los puntos de una cuadrícula.

En el caso de la colocación 1-dim, los módulos son colocados en filas con una altura fija y los terminales siempre se encuentran en, por lo que son los más utilizados cuando tratamos un circuito implementado con celdas estándares. En este caso, las conexiones se realizan en el espacio que queda entre las diferentes filas, es decir, únicamente se utiliza una dimensión (la altura) para realizar las conexiones.

Por último, la colocación 2-dim trata los módulos como cajas bidimensionales en los que los terminales pueden estar en los cuatro lados. Por lo tanto, es útil para la utilización de macrobloques y de dispositivos configurables. Luego, las conexiones deben ir a través de las dos dimensiones (altura y anchura).

Dichos tipos de colocación se muestran en la figura 5.2. Mostramos un ejemplo de cada uno de los tipos de colocación. En la figura 5.2a mostramos una etapa previa de la colocación de un dispositivo configurable (en concreto una FPGA), en la cual se ha llevado a cabo el tipo de colocación 0-dim. En la figura 5.2b mostramos la implementación de multiplicador utilizando celdas estándares; podemos apreciar que todas las celdas tienen la misma altura y están dispuestas en filas por lo que las conexiones son llevadas a cabo en el espacio que queda entre las diferentews celdas. También podemos ver unas conexiones que pasan a través de espacios

huecos entre las filas, dichas conexiones son las de polarización y tierra que deben ser lo más ideales posibles (con el menor número de dispositivos parásitos). Cuando se deben conectar dos nodos que no se encuentran en filas adyacentes, se utilizan unas celdas especiales denominadas feed-through, las cuales no tienen funcionalidad y su misión es dejar pasar las conexiones a través de ellas, pero no se utilizan para las conexiones de polarización y tierra. En la figura 5.2c mostramos la implementación de la circuitería de control de un convertidor analógico-digital implementado con macrobloques. Podemos apreciar los diferentes bloques (cajas) con diferentes tamaños, y las líneas de conexiónse encuentran en cualquiera de las dos dimensiones. Sólo se muestran una serie de canales, los cuales son los más anchos debido a la necesidad de que pasen un mayor número de señales por ellos; no obstante, entre todos los bloques existen conexiones aunque no se aprecien debido a que el número de señales sea pequeño. En el caso de las conexiones de polarización y tierra, se ha elegido que circunden el sistema para que sean fácilmente accesibles por todos los macrobloques.

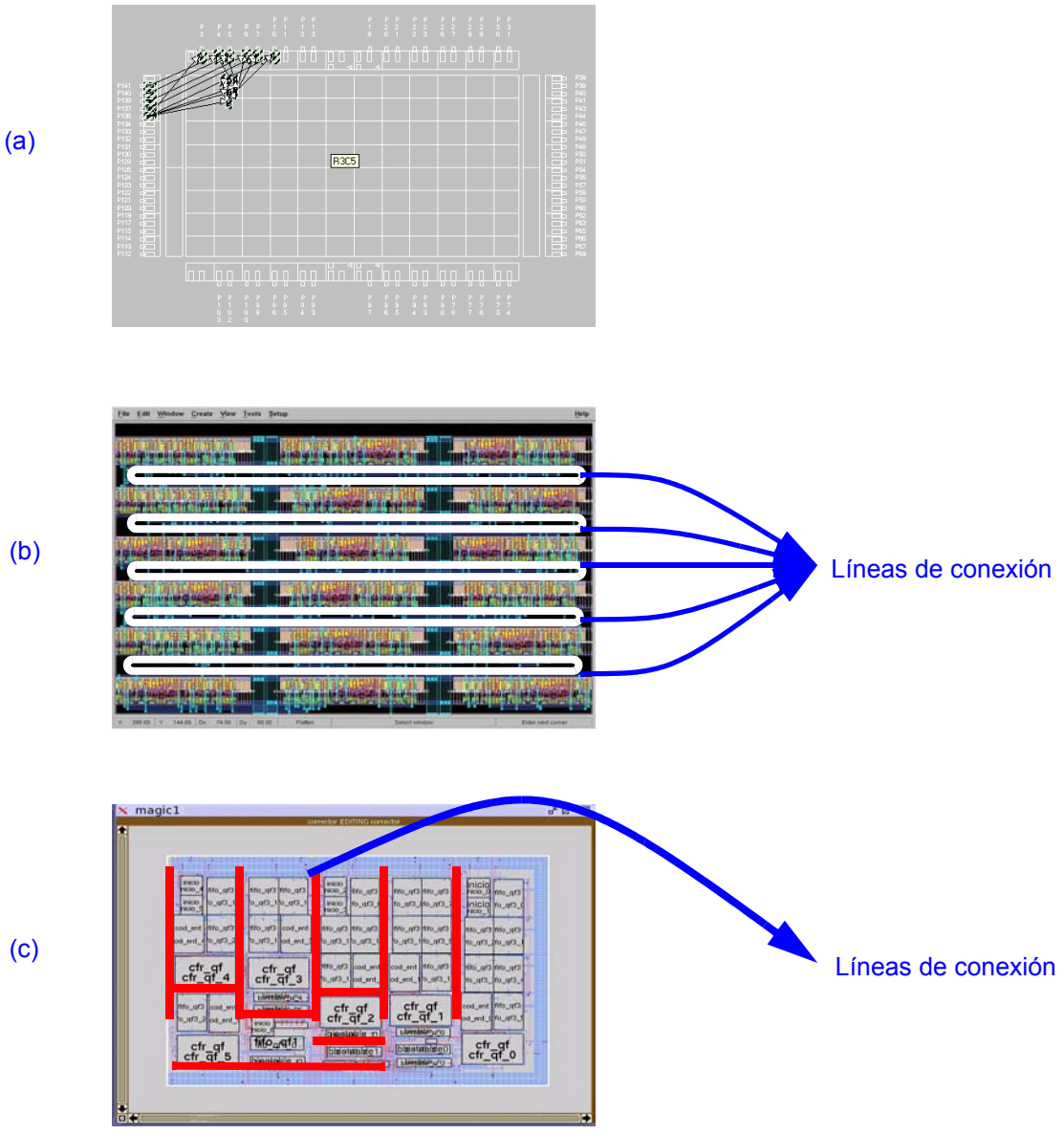


Figura 5.2.- Tipos de colocación de un circuito integrado. (a) Colocación 0-dim. (b) Colocación 1-dim. (c) Colocación 2-dim.

Otro de los objetivos de una buena colocación es minimizar las líneas de conexión, por lo que se debe saber cómo se realizan las conexiones para poder dar una estimación. Por lo tanto, a continuación vamos a tratar una serie de redes.

> Una red se define como el modo en que se conectan los diferentes componentes del sistema

Básicamente existen dos tipos de redes: redes punto a punto, y redes multipunto.

Las redes punto a punto únicamente considera dos puntos que han de conectarse. No obstante, cuando tenemos un nodo que tiene más de dos conexiones, tratamos las conexiones dos a dos de forma independiente de tal forma que los dos puntos que se conesideran en cada conexión son los más cercanos.

Para realizar una estimación del tamaño (la longitud) de las líneas de conexión debemos tener una forma de medir dichas líneas, y por lo tanto, como unir los diferentes terminales. En la figura 5.3 mostramos las dos formas más utilizadas. En la figura 5.3a mostramos la conexión Manhantan, la cual se basa en que los únicos ángulos permitidos son los ángulos rectos. Por lo tanto, una conexión típica estará compuesta por dos ramas: una horizontal y otra vertical, de tal forma que la longitud de la conexión se corresponderá con la siguiente fórmula

$$L = |X_1 - X_0| + |Y_1 - Y_0|$$

Luego, en el ejemplo de la figura 5.3a, la longitud de la conexión es igual a:

$$L = |1-0| + |3-1| + |1-1| + |0-3| + |2-1| + |2-0| + |3-2| + |0-2| = 12$$

Por contra, en la figura 5.3b mostramos la conexión euclídea, la cual se basa en que se permiten cualquier tipo de ángulos. Por lo tanto, cualquier conexión estará compuesta por la recta que une los nodos, de tal forma que la longitud de la conexión se corresponderá con la siguiente fórmula

$$L = \sqrt{(X_1 - X_0)^2 + (Y_1 - Y_0)^2}$$

Luego, en el ejemplo de la figura 5.3b, la longitud de la conexión es igual a:

$$L = \sqrt{|0-0|^2 + |3-1|^2 + |1-0|^2 + |3-3|^2 + |1-1|^2 + |0-3|^2 + |2-1|^2 + |2-0|^2 + |0-2|^2} \cong 4.8$$

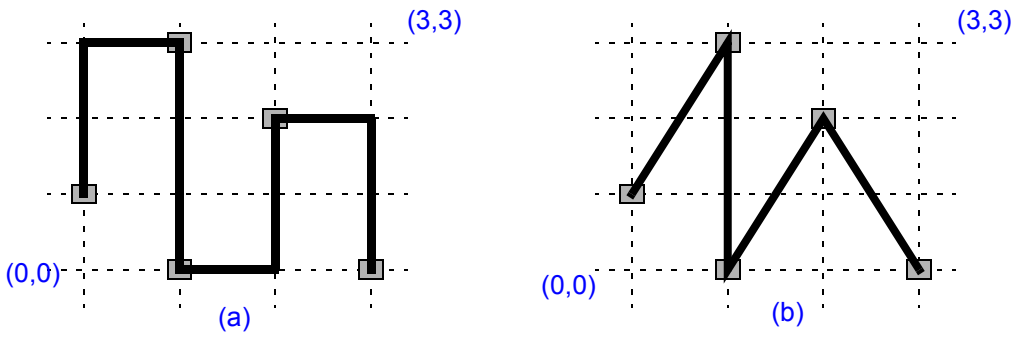


Figura 5.3.- Tipos de conexión: (a) Conexión Manhatan, y (b) conexión euclídea.

De estos dos tipos de conexiones, el más utilizado es la conexión Manhatan debido a dos razones básicas. En primer lugar, es más fácil la computación de sumas y restas que la del cuadrado y raíces cuadradas. En segundo lungar, la implementación física de una conexión en un circuito integrado está limitada a líneas rectas y codos (ángulos rectos), por lo que la conexión Manhatan es la más cercana a la conexión real y nos dará una estimación más aproximada.

En el caso de las redes multipunto, se estudian todos los nodos correspondientes a una misma señal a la vez para que la longitud de la conexión sea mínima. Básicamente existen dos alternativas: cadena margarita (daisy chain) y árbol de Steiner, mostrados en la figura 5.4. En el caso de la cadena margarita, se utilizan varias conexiones Manhatan punto a punto. Para asegurar que la longitud sea mínima se utiliza una solución del problema del viajante.

> El problema del viajante es aquel que trata de determinar la mejor ruta que debe elegir un viajante que debe pasar por una serie de pueblos.

Extrapolando este problema: los pueblos serán los nodos que se deben conectar y la ruta será la conexión que se debe obtener. En el caso de la figura 5.4a, la longitud de la conexión será la siguiente:

$$L = |0-1| + |1-0| + |1-0| + |3-1| + |2-1| + |2-3| + |3-2| + |0-2| = 10$$

Por contra, el árbol de Steiner está basado en la conexión Manhantan pero no se obliga a partir de uno de los nodos que se deben conectar (como en la cadena margarita), sino que se puede utilizar cualquier punto de la línea de conexión trazada. En el caso de la figura 5.4b la longitud de la conexión será la siguiente:

$$L = |3-0| + |0-1| + |1-1| + |3-1| + |1-1| + |0-1| + |2-2| + |1-2| = 8$$

que se corresponde con los puntos (0,1) (nodo), (3,0) (nodo), (1,3) (nodo), (1,1) (interna), (1,0) (nodo), (1,1) (interna), (2,1) (interna) y (2,2) (nodo). Como podemos apreciar tres nodos han sido conectados a puntos internos de la línea de conexión.

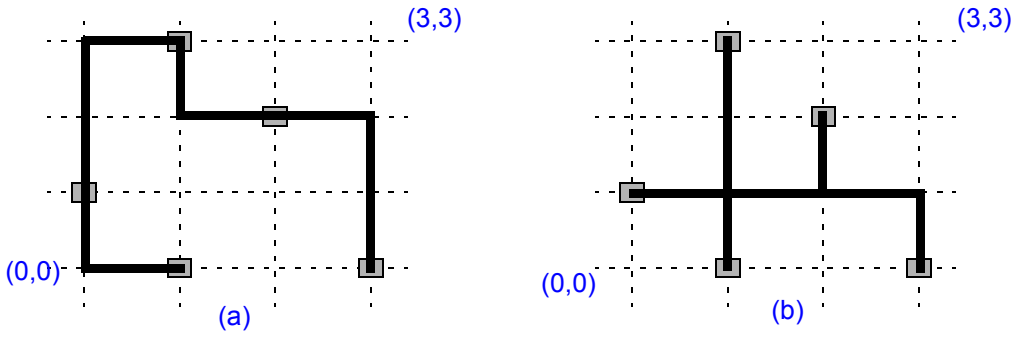


Figura 5.4.- Tipos de conexiones multipunto: (a) cadena margarita (daisy chain), y (b) árbol de Steiner.

A la hora de llevar a cabo una estimación de la longitud de la conexión se suele utilizar la cadena margarita debido a su facilidad de obtención, a pesar de que el resultado del árbol de Steiner será más exacto por ser utilizado en la implementación final puesto que nos dará una longitud menor. Para implementar la cadena margarita, sólo debemos obtener el siguiente nodo más cercano al actual, es decir, obtener una serie de sumas y restas. En cambio, para el árbol de Steiner debemos obtener los diferentes caminos que conectan dos nodos y determinar cuál sería más *aprovechable*, lo cual es mucho más complicado que el cálculo de la cadena margarita.

Otra forma de obtener una aproximación de la longitud de la cadena es el método semi-perímetro. Este método consiste en calcular el perímetro del rectangulo mínimo que circunscribe a todos los nodos que deben ser conectados con la misma línea. Esta es una cota mínima de la longitud de la línea de conexión, pero suele ser muy utilizada para realizar estimaciones de longitudes de conexión.

Las técnicas de colocación se pueden agrupar en dos grandes grupos: técnicas constructivas y técnicas iterativas. En las técnicas constructivas se selecciona un módulo (como semilla) el cual es colocado en el layout. Entonces otros módulos son seleccionados para colocarlos en una posición adecuada (y vacante) en función del módulo anterior, de tal forma que la longitud de las conexiones sean minimizadas. Este procedimiento es repetido hasta tener colocados todos los módulos. Los algoritmos englobados dentro de esta categoría suelen obtener resultados rápidamente pero de baja calidad; de hecho se suelen utilizar para obtener la colocación inical para los algoritmos iterativos. Algunos ejemplos de estos algoritmos son algoritmos de clustering y *min-cut* (corte mínimo).

Las técnicas iterativas parten de una colocación la cual se va cambiando para minimizar las longitudes de las conexiones. Luego, el procedimiento que se suele seguir es el siguiente. Se selecciona una colocación inicial, a la cual se aplica una perturbación para obtener una nueva colocación. Entonces se comprueba si la nueva colocación presenta mejoras sobre la colocación de partida. En caso de obtener una mejora, se selecciona la nueva colocación como la de partida y se vuelve a repetir (iterar) el procediemiento. El algoritmo se dará por terminado cuando no se produzca ninguna mejora en un número determinado de nuevas colocaciones (intentos). Algunos ejemplos de estos algoritmos son algoritmos de recocido simulado, algoritmos genéticos.

5.2. Colocación 0-dim

En el caso de la colocación 0-dim, los módulos son tratados como puntos adimensionales que deben ser colocados en los nodos de una rejilla, de tal forma que minimice la longitud de las conexiones y la congestión, es decir, la densidad de las conexiones. Este problema podría ser definido de la siguiente forma:

> El problema de colocación 0-dim es el siguiente. Dado un netlist y una rejilla, se pretende colocar los diferentes módulos en los nodos de la rejilla tal que la longitd de las conexiones y la congestión sean minimizadas.

Dentro de este tipo de colocación vamos a tratar dos tipos de algoritmos: técnicas *min-cut* (corte mínimo) y técnicas de clustering. Esta situación viene motivada porque estos algoritmos no tienen en cuanta (al menos de forma directa) el tamaño y forma de los módulos, así como la posición de los terminales.

Las técnica min-cut son llevadas a cabo con los siguientes pasos:

- Partimos de una red sin colocar
- Se realiza una primera bipartición
- Cada una de los bloques vuelve a ser bipartido
- Se repite el último paso hasta aislar cada módulo

Los algoritmos de partición utilizados suelen ser los algoritmos de Schweikert & Kernighan o de Fiduccia & Mattheyses. Las funciones objetivos generalmente pueden cambiar con respecto a la partición del tema anterior puesto que hay que considerar que los módulos deben estar conectados. Uno de los objetivos más utilizados es la minimización del corte total (como en el caso de la partición). Este objetivo considera el número total de conexiones que unen los dos bloques en los que se dividirá el sistema. Como trata de agrupar todos los módulos, también agrupará todos los terminales por lo que también minimizará el semiperímetro de las conexiones. Para considerar las conexiones con los bloques que han sido obtenidos de particiones previas, se sustitutyen dichos bloques por módulos vacios para tener información de cómo se deben colocar los bloques que se obtendrán de la nueva partición.

En la figura 5.5 mostramos la aplicación del algoritmo de Schweikert & Kernighan al contador de cinco bits para obtener una buena colocación. Partimos del hipergrafo sin particionar, al cual asignamos una primera partición aleatoria. En nuestro caso particular (figura 5.5a), las particiones asignadas están compuestas por los módulos {1, 2, 3, 4} y {5, 6, 7, 8}. Dicha partición ha resultado ser *buena* según el algoritmo considerado, ya que todas las ganancias son menores o iguales que cero. El siguiente paso consiste en volover a particionar los bloques obtenidos en la partición anterior, pero con la información del bloque complementario (etiquetados como Bi en la figura 5.5b). En el caso del bloque B1, los módulos que se encuentran conectados con el bloque B2 son los módulos 1, 2 y 4 por partida doble (a través de la señal de reloj y de la señal de dato); mientras que los módulos del bloque B2 conectados con el B1 son todos los módulos (en el caso de los módulos 6 y 8 por la señal de reloj, y los módulos 5 y 7 por las tres señales de dato de los módulos 1, 2 y 4). Una vez que se han realizado todas las particiones, pasamos a colocar todos los módulos en una rejilla siguiendo las pautas obtenidas por las particiones del paso anterior. En la figura 5.5c mostramos dos posibles colocaciones con una estimación de la longitud de las conexiones considerando una unidad de medida la distancia entre dos módulos adyacentes.

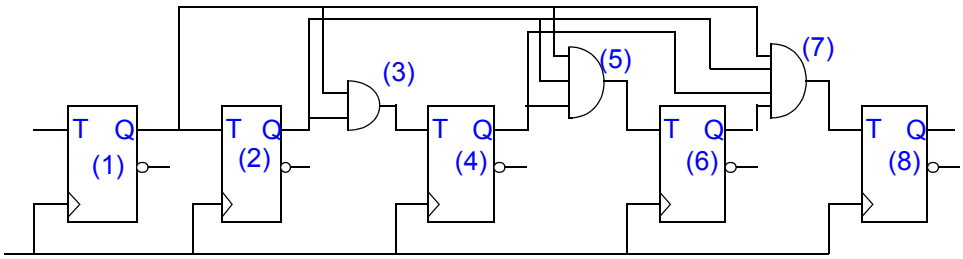
En la figura 5.6 mostramos la aplicación del algoritmo de Fiduccia & Mattheyses a un sumador de dos bits con pipeline a nivel de bit. La secuencia de pasos es la misma que la del ejemplo anterior. Aunque en los hipergrafos de las particiones no aparecen los bloques complementarios de las particiones de pasos anteriores, sí han sido tenidas en cuenta para obtener las particiones *buenas*.

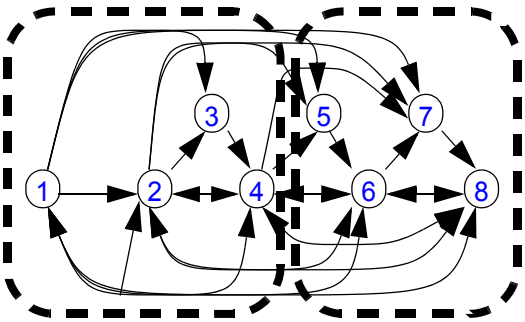
Las técnicas de clustering consiste en ir colocando en la rejilla los bloques de uno en uno en función de su relación con los demás. Podemos diferenciar dos tipos de clustering:

- Clustering uniforme, en el cual todos los módulos tienen el mismo peso (importancia)
- Clustering basado en fuerzas, en la cual cada módulo tendrá un peso (determinado por el número de conexiones, importancia del nodo, etc...), de tal forma que la colocación de un módulo procurará que la suma de todas las fuerzas (peso x distancia) se anulen.

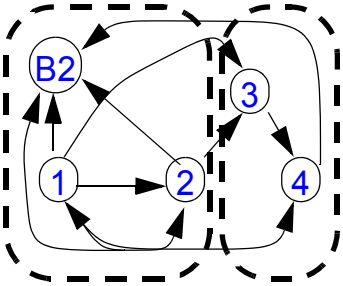
Los pasos que se deben seguir para tener una colocación utilizando un algoritmo de clustering son los siguientes:

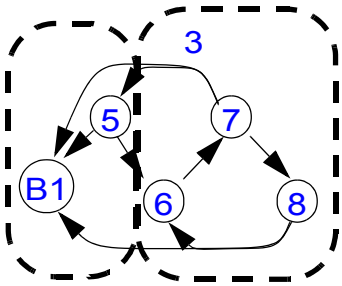
- Se obtiene el número de conexiones de cada módulo para obtener el orden en el que se irán colocacndo los diferentes módulos
- Se coloca en el centro del área una semilla (que será el módulo que presente un mayor número de conexiones).





(a)

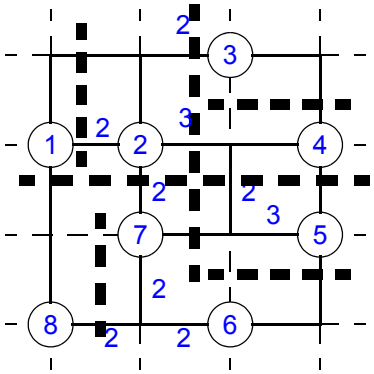




$G_{1-3}=(2+2)-(3+1)-2\cdot1=-2$
$G_{1-4}=(2+2)-(3+1)-2\cdot1=-2$
$G_{2-3}=(1+2)-(3+1)-2\cdot1=-3$
$G_{2-4}=(1+2)-(3+1)-2\cdot0=-1$
$G_{B2-3}=(1+2)-(4+2)-2\cdot0=-3$
$G_{B2-4}=(1+2)-(4+1)-2\cdot1=-4$

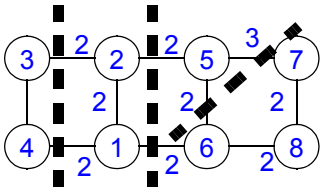
$G_{5-6}=(4+2)-(4+2)-2\cdot1=-2$
$G_{5-7}=(4+3)-(4+3)-2\cdot3=-6$
$G_{5-8}=(4+0)-(4+2)-2\cdot0=-2$
$G_{B1-6}=(1+2)-(4+2)-2\cdot1=-5$
$G_{B1-7}=(1+3)-(4+3)-2\cdot0=-3$
$G_{B1-8}=(1+0)-(4+2)-2\cdot0=-5$

(b)



$L = (3+5+2+3)_{col} + (4+6+4+5)_{fil} = 32$

$L = (1+2+2+2)_{col} + (7+6)_{fil} = 20$



(c)

Figura 5.5.- (a) Esquema a nivel de bloques de un contador de cinco bits. (b) Aplicación del algoritmo de Schweikert & Kernighan. (c) Posibles colocaciones.

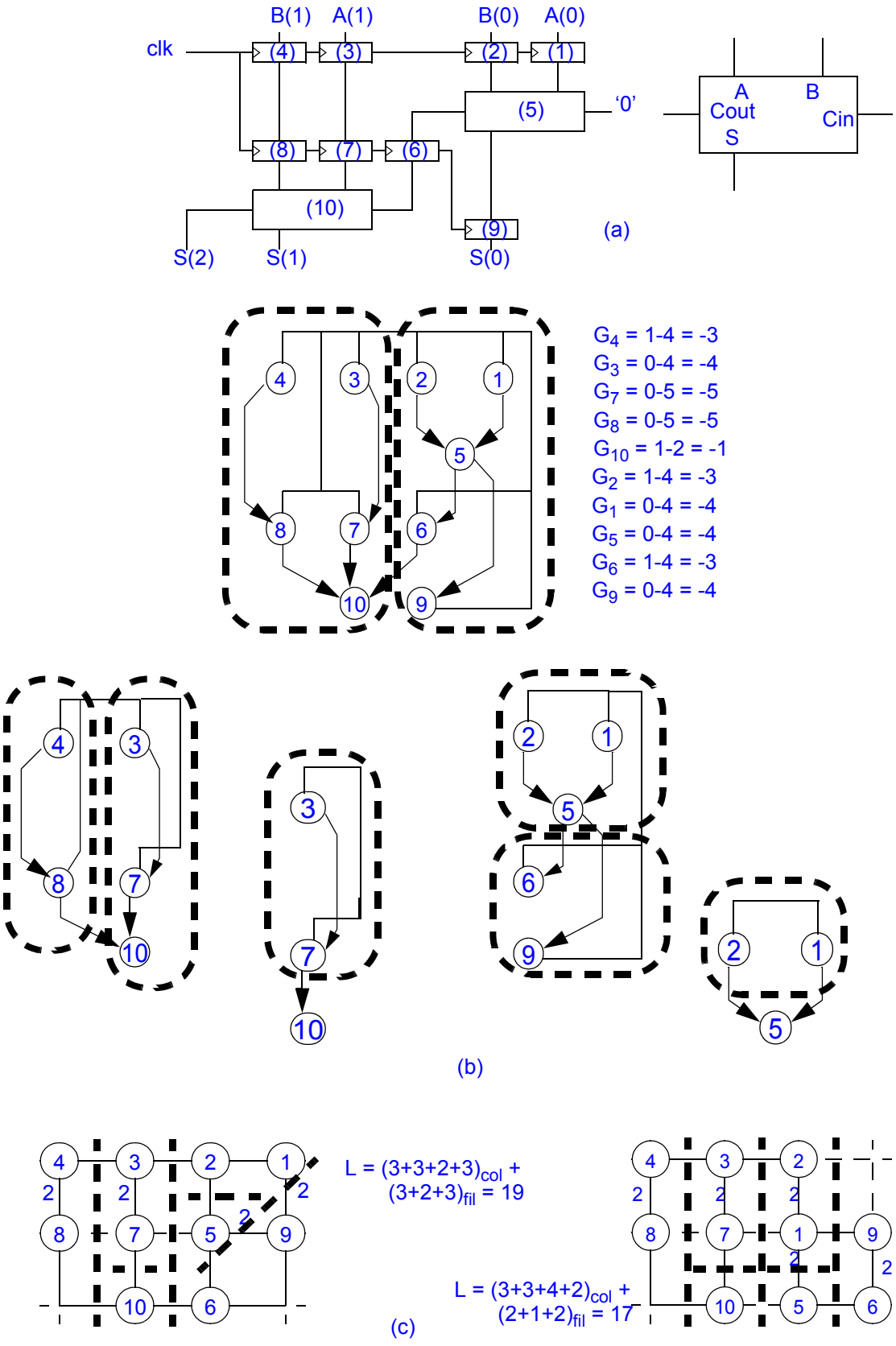
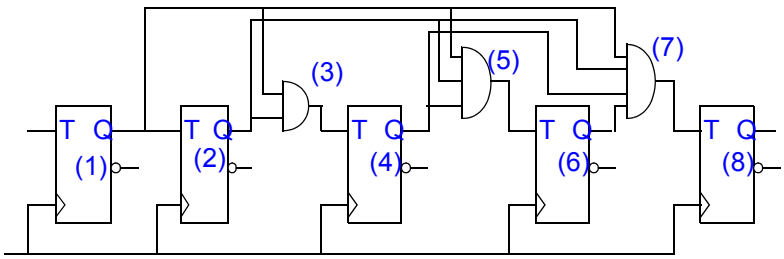


Figura 5.6.- (a) Esquema a nivel de bloques de un sumador de dos bits con pipeline a nivel de bits. (b) Aplicación del algoritmo de Fiduccia & Mattheyses. (c) Posibles colocaciones.

- Se colocarán a su alrededor los módulos que estén muy conectados con dicha semilla.
- Se colocarán el resto de módulos alrededor de los módulos ya colocados con muchas conexiones entre sí.

La diferencia entre la colocación uniforme y basado en fuerzas consiste en la colocación de estos módulos. Mientras que en el clustering uniforme la colocación es uniforme (como su propio nombre indica); en el clustering basado en fuerzas, la colocación de los módulos debe ser calculada para anular las fuerzas. Obviamente, en el clustering basado en fuerzas, los dos o tres primeros módulos deben ser colocados de forma uniforme para poder calcular las fuerzas.

En la figura 5.7 mostramos como ejemplo la aplicación de la ténica de clustering uniforme a l contador binario de cinco bits. Se muestran el número de conexiones como si tuviésemos un hipergrafo, que coincidirá con el número de conexiones de cada módulo. La semilla será el módulo 7 por tener cinco conexiones (el número máximo). Después se han seleccionado los módulos 3 y 5 por ser los más conectados con el 7 (3 y 2 conexiones respectivamente). Seguidamente se han elegido el resto de módulos excepto el 8 que ha sido el último en colocarse.



Conexiones de 1 = 3
Conexiones de 2 = 4
Conexiones de 3 = 3
Conexiones de 4 = 4
Conexiones de 5 = 4
Conexiones de 6 = 3
Conexiones de 7 = 5
Conexiones de 8 = 2

Conexiones de 1 con 7 = 1
Conexiones de 2 con 7 = 1
Conexiones de 3 con 7 = 2
Conexiones de 4 con 7 = 1
Conexiones de 5 con 7 = 3
Conexiones de 6 con 7 = 1
Conexiones de 8 con 7 = 1

Conexiones de 1 con 3 = 1
Conexiones de 2 con 3 = 1
Conexiones de 4 con 3 = 1
Conexiones de 6 con 3 = 0
Conexiones de 8 con 7 = 0

Conexiones de 1 con 5 = 1
Conexiones de 2 con 5 = 1
Conexiones de 4 con 5 = 1
Conexiones de 6 con 5 = 1
Conexiones de 8 con 5 = 0

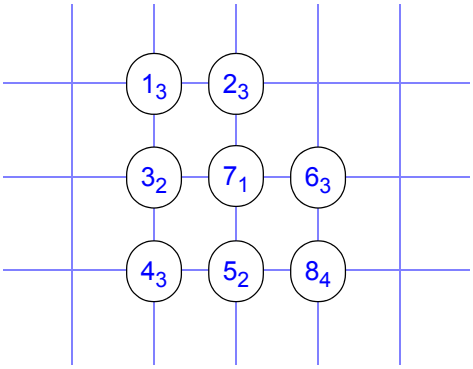


Figura 5.7.- Ejemplo de la técnica de clustering uniforme para la colocación del contador binario de cinco bits.

En la figura 5.8 se muestra como ejemplo la aplicación del clustering uniforme al sumador de dos bits con pipeine a nivel de bit. La semilla en este caso será el módulo 5, tras el cual se colocarán los módulos 1, 2, 6 y 9. Seguidamente se colocarán el resto de módulos.

También se han obtenido las posiciones ideales de los módulos 3, 4, 5, 7, 8 y 10 basadas en fuerzas donde los pesos se han igualado al número de conexiones de los diferentes módulos. En primer lugar se han obtenido los vectores peso de todos los módulos. Por ejemplo, el vector peso del módulo 7 es (1, 1, 2, 1, 0, 1, -, 1, 1, 1) colocados por orden, es decir, (módulo1, módulo 2, módulo 3, ..., módulo 9, módulo 10). Este vector peso es debido a que:

- con el módulo 1 tiene una conexión (la señal de reloj)
- con el módulo 2 tiene una conexión (la señal de reloj)
- con el módulo 3 tiene dos conexiones (la señal de reloj y la señal de dato)
- con el módulo 4 tiene una conexión (la señal de reloj)
- con el módulo 5 no tiene ninguna conexión
- con el módulo 6 tiene una conexión (la señal de reloj)
- no tiene sentido el peso del módulo 7 consigo mismo
- con el módulo 8 tiene una conexión (la señal de reloj)
- con el módulo 9 tiene una conexión (la señal de reloj)
- con el módulo 10 tiene una conexión (la señal de dato)

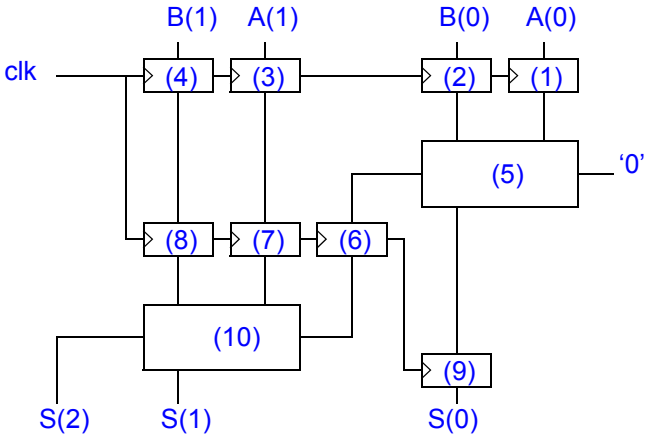
Una vez que tenemos los vectores peso, aplicamos la ecuación de cancelación de fuerzas con los módulos que ya se encuentran colocados. Dicha ecuación es la siguiente

$$(x,y) = \frac{\sum (x_i,y_i) w_i}{\sum w_i}$$

Dicha ecuación ha sido aplicada a dichos módulos obteniendo su colocación óptima. En el caso de que dicho lugar ya esté ocupada por otro módulo, se debe elegir un nuevo lugar (a ser posible cercano al óptimo). Esta elección se compensará con la compensación de fuerzas de los restantes módulos. En este caso se ha elegido la misma colocación que la obtenida a través del clustering uniforme.

También podemos aplicar algoritmos iterativos a diferencia de algoritmos constructivos, como el caso de los dos anteriores (técnicas de min-cut y de clustering). Estos algoritmos iterativos se basan principalmente en tres pasos:

- Obtener una partición inicial. Dicha partición puede ser obtenida de forma aleatoria, o bien, obtenida a partir de la aplicación de algunos de los algoritmos constructivos (situación que suele ser más común).
- A dicha partición se le aplica una perturbación para obtener una nueva partición. Básicamente las perturbaciones que se suelen considerar son las siguientes:
  - Intercambio de dos módulos. En este caso la posición de dos módulos son intercambiadas. La elección de los módulos que se intercambiarán puede ser aleatoria o exhaustiva. En este último caso se considerarán el intercambio de un módulo con todos los demás y quedarse con la colocación más ventajosa.



Conexiones de 1 = 2
Conexiones de 2 = 2
Conexiones de 3 = 2
Conexiones de 4 = 2
Conexiones de 5 = 4
Conexiones de 6 = 3
Conexiones de 7 = 3
Conexiones de 8 = 3
Conexiones de 9 = 2
Conexiones de 10 = 3

Conexiones de 1 con 5 = 1
Conexiones de 2 con 5 = 1
Conexiones de 3 con 5 = 0
Conexiones de 4 con 5 = 0
Conexiones de 6 con 5 = 1
Conexiones de 7con 5 = 0
Conexiones de 8 con 5 = 0
Conexiones de 9 con 5 = 1
Conexiones de 10 con 5 = 0

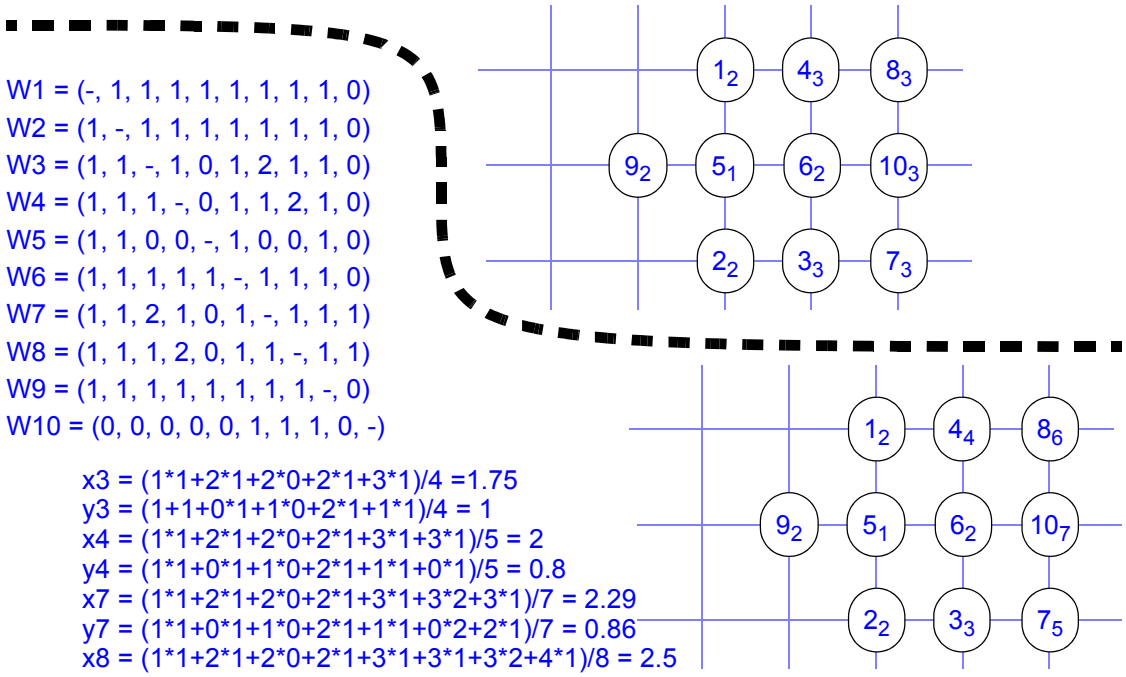
Conexiones de 3 con 1= 1
Conexiones de 4 con 1 = 1
Conexiones de 7 con 1 = 1
Conexiones de 8 con 1 = 1
Conexiones de 10 con 1 = 0

Conexiones de 3 con 2= 1
Conexiones de 4 con 2 = 1
Conexiones de 7 con 2 = 1
Conexiones de 8 con 2 = 1
Conexiones de 10 con 2 = 0

Conexiones de 3 con 6= 1
Conexiones de 4 con 6 = 1
Conexiones de 7 con 6 = 1
Conexiones de 8 con 6 = 1
Conexiones de 10 con 6 = 1

Conexiones de 3 con 9 = 1
Conexiones de 4 con 9 = 1
Conexiones de 7 con 9 = 1
Conexiones de 8 con 9 = 1
Conexiones de 10 con 9 = 0

W1 = (-, 1, 1, 1, 1, 1, 1, 1, 1, 0)
W2 = (1, -, 1, 1, 1, 1, 1, 1, 1, 0)
W3 = (1, 1, -, 1, 0, 1, 2, 1, 1, 0)
W4 = (1, 1, 1, -, 0, 1, 1, 2, 1, 0)
W5 = (1, 1, 0, 0, -, 1, 0, 0, 1, 0)
W6 = (1, 1, 1, 1, 1, -, 1, 1, 1, 0)
W7 = (1, 1, 2, 1, 0, 1, -, 1, 1, 1)
W8 = (1, 1, 1, 2, 0, 1, 1, -, 1, 1)
W9 = (1, 1, 1, 1, 1, 1, 1, 1, -, 0)
W10 = (0, 0, 0, 0, 0, 1, 1, 1, 0, -)



x3 = (1*1+2*1+2*0+2*1+3*1)/4 =1.75
y3 = (1+1+0*1+1*0+2*1+1*1)/4 = 1
x4 = (1*1+2*1+2*0+2*1+3*1+3*1)/5 = 2
y4 = (1*1+0*1+1*0+2*1+1*1+0*1)/5 = 0.8
x7 = (1*1+2*1+2*0+2*1+3*1+3*2+3*1)/7 = 2.29
y7 = (1*1+0*1+1*0+2*1+1*1+0*2+2*1)/7 = 0.86
x8 = (1*1+2*1+2*0+2*1+3*1+3*1+3*2+4*1)/8 = 2.5
y8 = (1*1+0*1+1*0+2*1+1*1+0*1+2*2+0*1)/8 = 1
x10 = (1*0+2*0+2*0+2*0+3*1+3*0+3*0+4*1+4*1)/3 = 3.67
y10 = (1*0+0*0+1*0+2*0+1*1+0*0+2*0+0*1+2*1)/3 = 1

Figura 5.8.- Ejemplo de la técnica de clustering uniforme y basado en fuerzas para la colocación del sumador binario de dos bits con pipeline a nivel de bit.

  - Cambio de un módulo. En este caso, únicamente cambiará de posición un módulo; en el caso de que la nueva posición esté ocupada podemos optar por dos situaciones: no ocupar la nueva posición; o sustituir el módulo que ocupaba dicha posición, el cual será colocado en una nueva posición en un paso posterior. La forma de elección del módulo, así como su posición, puede ser de forma aleatoria o bien basada en fuerzas (como la técnica de clustering).

- Una vez que tengamos la nueva partición, chequeamos si existe alguna mejora con respecto de la partición de partida, es decir, si se disminuye una determinada función de coste. Esta función de coste suele depender de la longitud de las conexiones que deben existir (para realizar todas las conexiones del sistema) y de la congestión de las conexiones (el número de cruces y de conexiones que deben ir por el mismo lugar). En el caso de que exista mejora, la nueva partición se toma como partición inicial y se vuelve a aplicar todo el proceso. En el caso de que no exista mejora, se aplica una nueva perturbación a la partición de partida y se chequea la mejora; cuando se aplican un cierto número de pertubaciones sin que exista mejora, se considera que la partición de partida es la partiación óptima.

En la figura 5.9 se muestra un ejemplo de aplicación del algoritmo iterativo con intercambio de módulos aleatorio, y utilizando como función de coste la longitud total de conexiones y el número máximo de nodos que coinciden en un mismo canal. Partimos de la colocación obtenida siguiendo el clustering uniforme, la cual tiene una longitud de 21 unidades (considerando una unidad como la distancia entre dos módulos adyacentes) y una congestión máxima de 3 nodos. Se ha elegido aleatoriamente el intercambio de los módulos 5 y 7; con lo cual se ha reducido la longitud de las conexiones en una unidad manteniendo la congestión (realmente también se ha reducido ya que ahora sólo hay un canal con tres conexiones por dos que había antes). Seguidamente se ha elegido el intercambio de los módulos 3 y 6; con lo cual se ha aumentado la longitud de las conexiones a veintidos unidades, y se ha aumentado el número de canales con máxima congestión a tres. Por lo tanto, la nueva colocación no es buena y no será seleccionada como nueva colocación de partida. Ahora se ha elegido el intercambio de los módulos 7 y 6 consiguiéndose una función de coste similar a la colocación de partida; en este caso habría que elegir una de las dos colocaciones (de forma aleatoria porque se están realizando todas las elecciones de forma aleatoria) desde la que seguir. En nuestro caso se ha seleccionado la última colocación. Se ha realizado un nuevo intercambio, en este caso de los módulos 2 y 3. Con esta nueva colocación, la función de coste ha aumentado al igual que el número de canales con la máxima congestión. Con estas cinco iteraciones se dará el algoritmo por concluido, por lo tanto se han obtenido dos colocaciones óptimas (las cuales están enmarcadas en la figura).

5.3. Colocación 1-dim

La colocación 1-dim está muy dirigida hacia el diseño con celdas estándares, en el cual todas las conexiones se ha de realizar en una dimensión (altura) existente entre las celdas estándares. En la figura 5.10 mostramos el layout de un multiplicador basado en sumas y acumulaciones utilizando celdas estándares. En él podemos distinguir las filas de celdas estándares que tienen una altura común y los terminales de tierra y polarización en las mismas posiciones para que colocando las diferentes celdas unas al lado de otras se conecten dichos terminales (sin necesidad de añadir líneas externas en una misma fila). El espacio existente entre las diferentes filas se utiliza para albergar las conexiones entre las diferentes celdas;

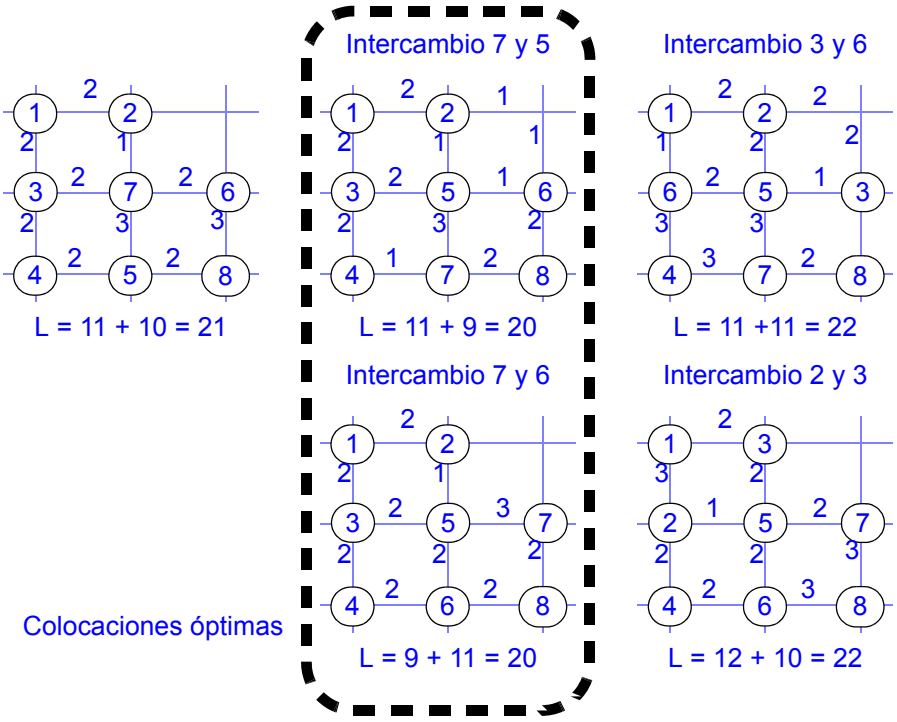


Figura 5.9.- Aplicación del algoritmo iterativo al contador de cinco bits.

dichos espacios se suelen conocer como canales. Cuando se deban conectar celdas que se encuentren en filas no adyacentes, la conexión debe atravesar (al menos) una fila. Para ello se hace uso de las denominadas celdas feedthrough, las cuales son celdas que no tienen ningún transistor implementado y se utilizan como un tunel. En realidad dichas celdas únicamente tienen nwell y líneas de polarización y tierra, para mantener su conexión. Las únicas conexiones que no se realizarán utilizando celdas feedthrough son las de polarización y tierra para evitar en la mayor medida de lo posible dispositivos parásitos que puedan provocar algún malfuncionamiento en dichas líneas.

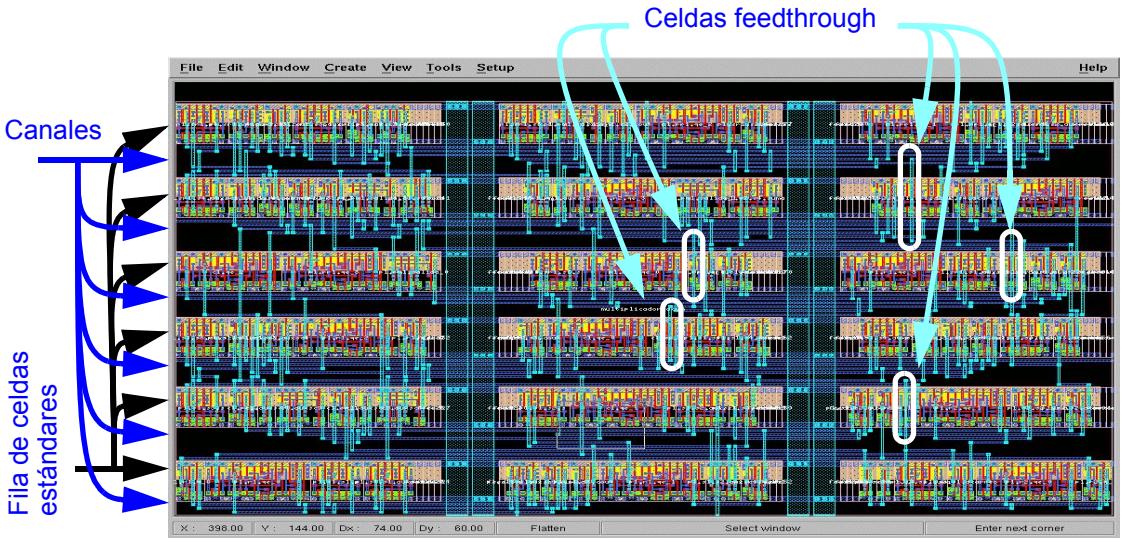


Figura 5.10.- Layout de un multiplicador implementado con celdas estándares.

Realmente la colocación 1-dim obtuvo su denominación cuando todas las celdas estándares se colocarán en una misma fila, como el caso de la figura 5.11. En este caso, habría que obtener el número de líneas que deberían existir, el cual depende de la colocación como se demuestra de las dos colocaciones posibles: en el primer caso se necesitarían ocho líneas; en cambio en el segundo caso sólo se necesitan cinco líneas ya que en un mismo nivel pueden ir más de una línea al no solaparse. Por ejemplo las salidas de las puertas AND puede solaparse en el mismo nivel reduciéndose en tres niveles los ocho niveles primarios (un nivel por cada nodo).

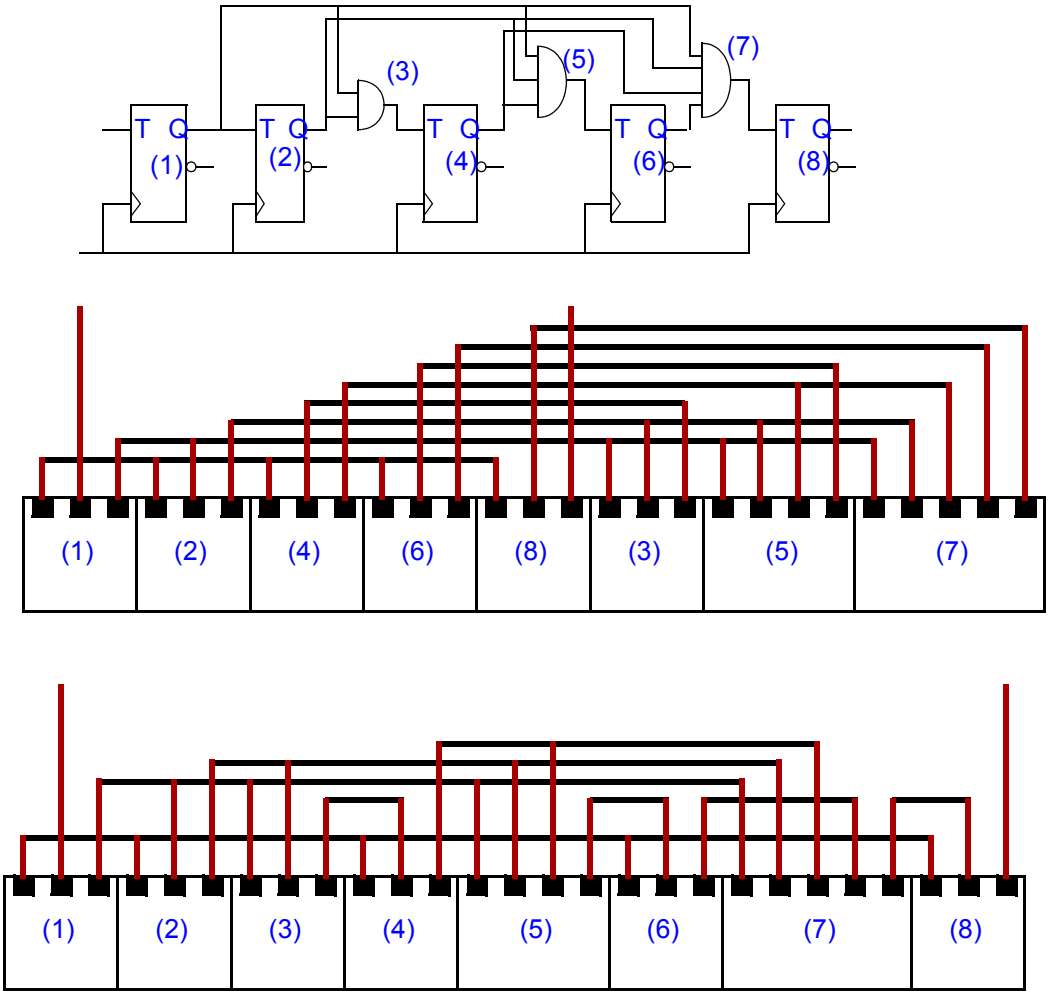


Figura 5.11.- Ejemplo de colocación del contador de cinco en una sola fila de celdas estándares.

Por lo tanto, la colocación a elegir debe ser tal que permita el máximo solapamiento de las líneas. No obstante, cuando el número de celdas crece en gran medida, no es viable utilizar una sola fila de celdas estándares ya que la longitud de la fila sería excesivamente larga. Por ello, actualmente todos los diseños basados en celdas estándares tienen más de una fila, y la colocación deberá tener en cuenta que las filas adyacentes tendrán que estar formadas por las celdas más conectadas.

Uno de los algoritmos más utilizados en la colocación de celdas estándares es el algoritmo de Timberwolf, basado en el recocido simulado. El algoritmo de recocido simulado es una aproximación de la cristalización de materiales. La ejecución del algoritmo requiere la

consecución de una serie de pasos (los cuales van a ser indicados con el paralelismo del método de cristalización):

- Obtener una colocación inicial, aplicando una técnica de colocación constructiva o de forma aleatoria. Dicha colocación sería el material que se desea cristalizar.
- Inicializar las iteraciones del algoritmo. Dichas iteraciones serían los valores de la temperatura por las que pasarían hasta llegar al nivel de cristalización deseado.
- Generar una colocación de forma aleatoria a traves de algún moviemiento partiendo de la actual. Esta generación sería igual a aplicar una perturbación al material para producirle un cambio. Los movimientos utilizados por el algoritmo de Timeberwolf son los siguientes, el cual será elegido de forma aleatoria según unas determinadas probabilidades:
  - M1: Mover un módulo a una nueva posición
  - M2: Intercambiar la posición de dos módulos. Estos dos movimientos son los que tienen una mayor probabilidad.
  - M3: Cambiar la orientación (espejar horizontalmente o verticalmente) de un módulo.
- Calcular el cambio de la función de coste debido al cambio de la colocación. Este paso sería equivalente a calcular el cambio de energia del material. La función de coste estarán incluidos los siguientes parámetros:
  - La longitud de las conexiones.
  - La ocupación de las filas.
  - El solapamiento de las celdas.
- Si dicho cambio es negativo (la función de coste de la nueva colocación es menor que la actual), se toma la nueva colocación como colocación actual y se vuelve a ejecutar desde el paso número tres.
- Si dicho cambio es positivo, se determina de forma aleatoria si se cambia la colocación actual o permanece la anterior. En cualquier caso, se vuelve a ejecutar desde el paso número tres si no se ha realizado el número máximo de iteraciones.

5.4. Colocación 2-dim

En la colocación 2-dim consideraremos macrobloques, en los cuales los terminales pueden estar en cualquiera de los cuatro lados. Lo que se suele realizar es adaptar los algortimos de colocación 0-dim para tener en cuenta la posición de los terminales.

La colocación 2-dim suele realizarse siguiendo dos pasos bien diferenciados:

- Obtener una colocación óptima utilizando algoritmos para la colocación 0-dim.
- Una vez que se conoce la mejor colocación de cada módulo, cada uno de los módulos se colocan de la forma más adecuada para obtener unas dimensiones mínimas.

En el segundo paso hay que considerar dos tipos diferentes de macrobloques:

- Hard macros. Son macrobloques que tienen una forma y posición de los terminales fijas, que no pueden ser variadas. Por lo tanto, la mejor colocación de estos macrobloques se reducirá a operaciones de rotar o espejar en diferentes direcciones. Un ejem-

plo de este tipo de macrobloques son los layouts full-custom que ha realizado el diseñador de niveles inferiores de la jerarquía.

- Soft macros. Son macrobloques que tienen una forma y posición de los terminales variables. Por lo tanto, la mejor colocación de estos macrobloques no se limita a rotar y espejar el bloque (como sucede en el caso anterior), sino que se pueden generar nuevos layouts con formas diferentes para que se adapten mejor a su posición. Un ejemplo de este tipo de macrobloques son porciones de código VHDL (por ejemplo), que el sintetizador generará de forma automática adecuándose a su posición final en el layout.