

**Tema IV.**

**Partición**

## 4.1. Introducción

La tarea de particionado consiste en dividir un sistema complejo en varios subsistemas de complejidad menor, tanto externa como interna. Este hecho indica que la complejidad de las conexiones entre los diferentes subsistemas debe ser minimizada. De hecho cualquier sistema VLSI debe llevar a cabo varios procesos de particionado a lo largo de su proceso de diseño. Este proceso es la base del diseño jerárquico, que se basa en el principio de *divide y vencerás*. Una vez que tengamos dividido el sistema, cada uno de los subsistemas será tratado de forma independiente.

Una vez que todos los subsistemas están implementados, se debe garantizar que la fase final, la conexión de todos los subsistemas, no introducirá una complejidad demasiado elevada. Por lo tanto, la tarea de particionado deberá minimizar la conexión entre los diferentes subsistemas, básicamente debido a dos razones:

- La comunicación entre subsistemas es cara. El coste no se reduce al coste en el número físico de conexiones, que tiene una influencia directa en el área ocupada (y por tanto en el coste de integración), sino que también tiene un coste (reducción de prestaciones) en el resto de parámetros. Por ejemplo, la existencia de muchas conexiones implica la dependencia de muchos bloques por lo que será necesario esperar a que dichos bloques acaben su operación, con la consiguiente pérdida de velocidad.
- El diseño de interfaz de subsistemas. Si hay que conectar muchos bloques, el interfaz que los conecte alcanzará una complejidad mayor.

A modo de ejemplo de mediana complejidad, en la figura 4.1 se muestra el esquema a nivel de bloques de un circuito de control de un convertidor analógico-digital. Según dicho esquema, las particiones más óptimas son las formadas por cada una de las columnas (incluyendo un bloque de inicio, una matriz de registros, una celda `cod_gen`, una celda `CfR` y un bloque de biestables). De esta forma, todas las particiones tienen el mismo número de conexiones con el resto de partes. No obstante, el tamaño de cada partición es diferente; por lo que para tratar de obtener un circuito lo más homogéneo posible (un layout lo más cuadrado posible) se ha establecido que las diferentes particiones sean lo más proporcionadas posibles (utilizándose como una restricción). Luego, el floorplanning de dicho esquema se muestra también en la figura 4.1. En el particionado se ha eliminado los bloques biestables, los cuales han sido utilizados para rellenar espacios vacíos, y de esta forma obtener un layout lo más cuadrado posible, o en su defecto, rectangular.

Como ya se ha comentado, procesos de particionado son muy comunes en el diseño VLSI, y no solo en uno de los pasos del flujo de diseño, sino en varios de ellos. A continuación indicamos las principales situaciones en las que es muy común realizar dicho proceso, el particionado.

- Empaquetamiento de diseño en general. Los circuitos lógicos deben ser agrupados en grupos, los cuales estarán sujetos a limitaciones de área y/o límites de entrada/salida.
- Particionado a nivel de netlist en procesos de síntesis basados en lenguajes de descripción de hardware (HDL). Este proceso tiene el mismo objetivo que el anterior pero

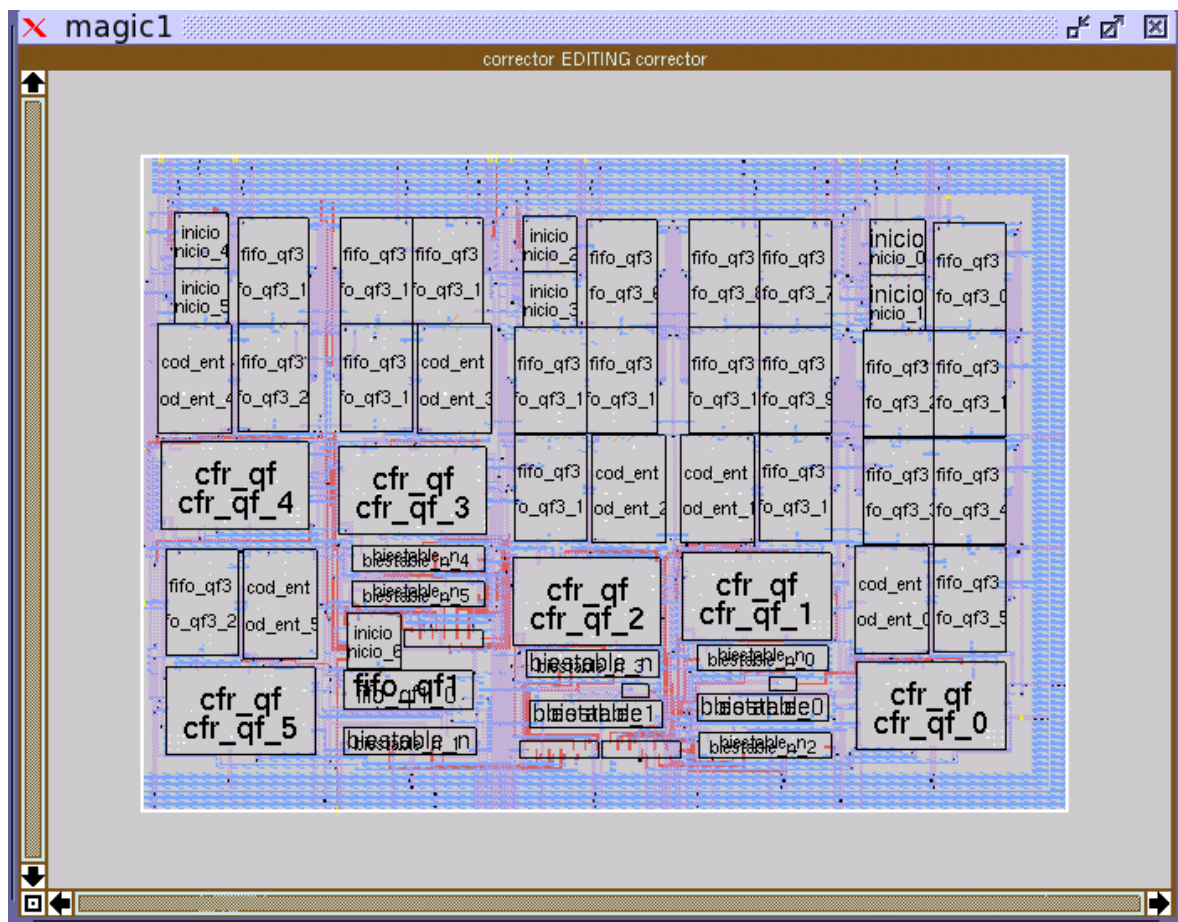
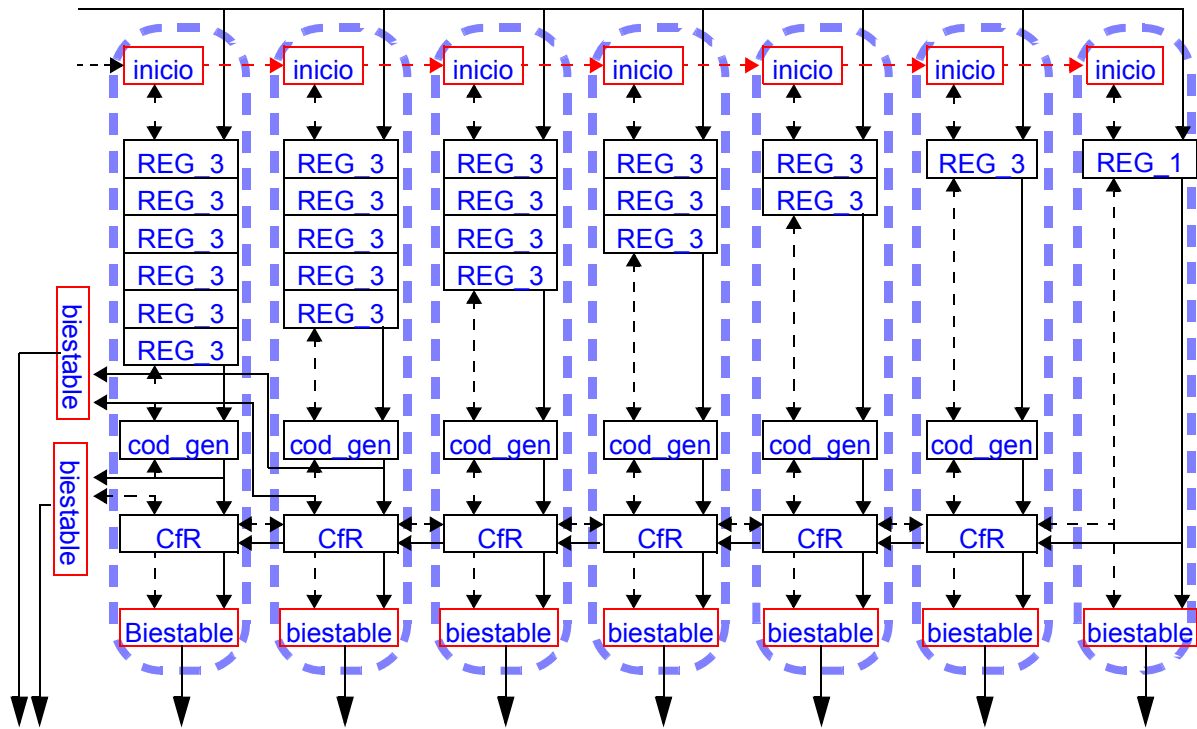


Figura 4.1. Esquema a nivel de bloques y floorplanning de un circuito de control de un ADC.

con una gran diferencia, no existencia una relación directa entre el número de líneas de código con el área ocupada por el circuito final. Por lo tanto, el tamaño de los diferentes grupos será diferente teniendo en cuenta el mapeado tecnológico.

- Estimación para optimización de diseño. El proceso de particionado, además de formar los diferentes grupos, nos indica su tamaño (área ocupada) y el número de conexiones entre los diferentes grupos. Por lo tanto, se puede realizar una optimización relativamente precisa acerca del área ocupada del circuito final.
- Emulación de sistemas y prototipado rápido. Este proceso es muy utilizado en los dispositivos de prototipado rápido como puede ser una FPGA. En este proceso, el particionado debe garantizar una serie de limitaciones tanto de tiempos como de áreas (puertas) típicos de dichos dispositivos.
- Caracterización y test hardware. Si el particionado de un circuito es óptimo, el número de conexiones entre las diferentes partes se reduce al mínimo. En este sentido, el test y la caracterización del circuito global puede reducirse a generar patrones de test de cada una de las partes del sistema ya que serán lo más independientes posible.

Existen una cantidad elevada de algoritmos de particionado, los cuales pueden dividirse en las siguientes categorías:

- Algoritmos basados en movimientos, o migración de grupos. Estos algoritmos tratan de obtener la partición óptima cambiando (moviendo) componentes de un grupo a otro, de tal forma que una función de coste sea minimizada.
- Algoritmos geométricos. Estos algoritmos tratan de construir una representación geométrica basada en el problema de particionado.
- Algoritmos matemáticos. Estos algoritmos tratan de traducir el problema de particionado en un problema típico matemático.
- Algoritmos de clustering (basados en grupos). Estos algoritmos tratan de obtener la partición óptima uniendo en diferentes grupos los módulos que componen el circuito global.

Debido al elevado número de algoritmos, vamos a ver unos pocos que pertenecen al primer grupo, es decir, algoritmos de migración de grupos.

Un algoritmo basado en la migración de grupos muestra el esquema general de la figura 4.2. Se empieza por generar una partición inicial. A partir de esta partición, se genera una nueva partición. Seguidamente se evalúa la nueva partición atendiendo a unos determinados criterios, los cuales nos indicarán si la partición es óptima o no. Si la partición es óptima se acaba el proceso; en caso contrario, se genera una nueva partición repitiendo todo el proceso.

En primer lugar hay que generar una partición inicial, la cual hay que representarla de alguna forma. El método más común de representar un conjunto es utilizar un grafo.

Un grafo se puede definir como un conjunto de módulos ( $V$ ) y nodos ( $E$ ) de tal forma que la lista de nodos está formada por el conjunto de conexiones de los diferentes módulos, teniendo al menos dos conexiones.

A modo de ejemplo vamos a representar una partición del esquema a nivel de bloques de la parte de control mencionada anteriormente, la cual puede verse en la figura 4.3a. Dicha repre-

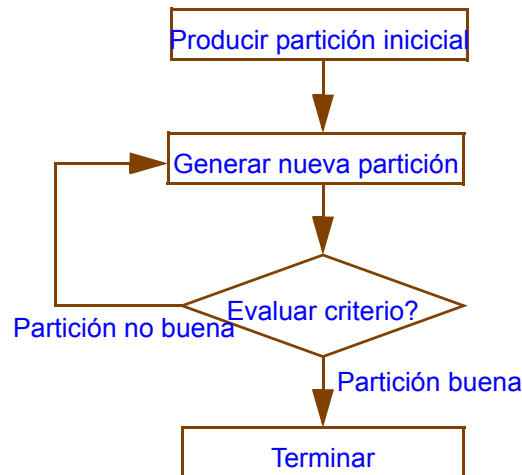


Figura 4.2. Esquema general de un algoritmo de migración de grupos.

sentación se puede ver en la figura 4.3b. Los módulos son representados por las etiquetas enmarcadas, mientras que los nodos son representados por los arcos. En este caso hemos considerado un grafo dirigido, indicando el sentido del flujo de señal; en el caso de la señal de entrada compartida por los módulos R1 y R2, el arco se ha utilizado con la doble dirección indicando que son entradas para ambos módulos. También hemos incluido un solo arco por cada dos conexiones (punteada y sólida), ya que en este caso particular (cuyo funcionamiento no influye en el tema abordado) ambos tipos de conexiones son dependientes y siempre aparecen por parejas (excepto en los bloques IX, en ñps que se ha incluido la conexión punteada).

No obstante, la situación del módulo R1 que comparte su salida con los módulos B1 y F2 no suele ser real en el layout en el sentido de que dicha señal tome dos caminos diferentes desde el módulo de origen. La situación más real es que tome un solo camino, el cual será bifurcado en el lugar preciso, tal como se muestra en la figura 4.3c. Dicha figura se corresponde con el denominado hipergrafo.

Un hipergrafo se puede definir como un grafo utilizado en el proceso de partición en el que habrá tantos cortes entre las dos partes como nodos comunes existan, y no como módulos relacionados tengan.

Si seguimos con el ejemplo de la figura 4.3, y hacemos una primera división con los siguientes módulos:  $P1 = \{I2, I1, R2, R1\}$  y  $P2 = \{G2, F2, B2, B1\}$ ; en el grafo de la figura 4.3b habrá tres cortes, correspondientes a las siguientes relaciones: R2-G2, R1-B1 y R1-F2. No obstante, las dos últimas relaciones se corresponden con un mismo nodo por lo que el grafo que tenemos no es un hipergrafo. En cambio, en la figura 4.3c únicamente habrá dos cortes, correspondientes a las siguientes relaciones: R2-G2 y R1- $\{B1, F2\}$ ; las cuales se corresponden a nodos diferentes por lo que esta vez sí estamos ante un hipergrafo. En el caso de los hipergrafos, las relaciones externas (que ya veremos tienen mucha influencia a la hora de determinar una partición óptima) se deben asignar únicamente a uno de los módulos de cada parte, mientras que el resto de conexiones a módulos de la misma parte se consideran internos. Por ejemplo, la conexión con el módulo B1 se puede considerar como externa y con F2 como interna, es decir, estaríamos ante la situación R1-B1-F2. Por lo tanto, si cambiamos la partición, el tipo de conexión de un módulo puede cambiar de externa a interna. Dicha situación se puede apreciar en la figura 4.3d.

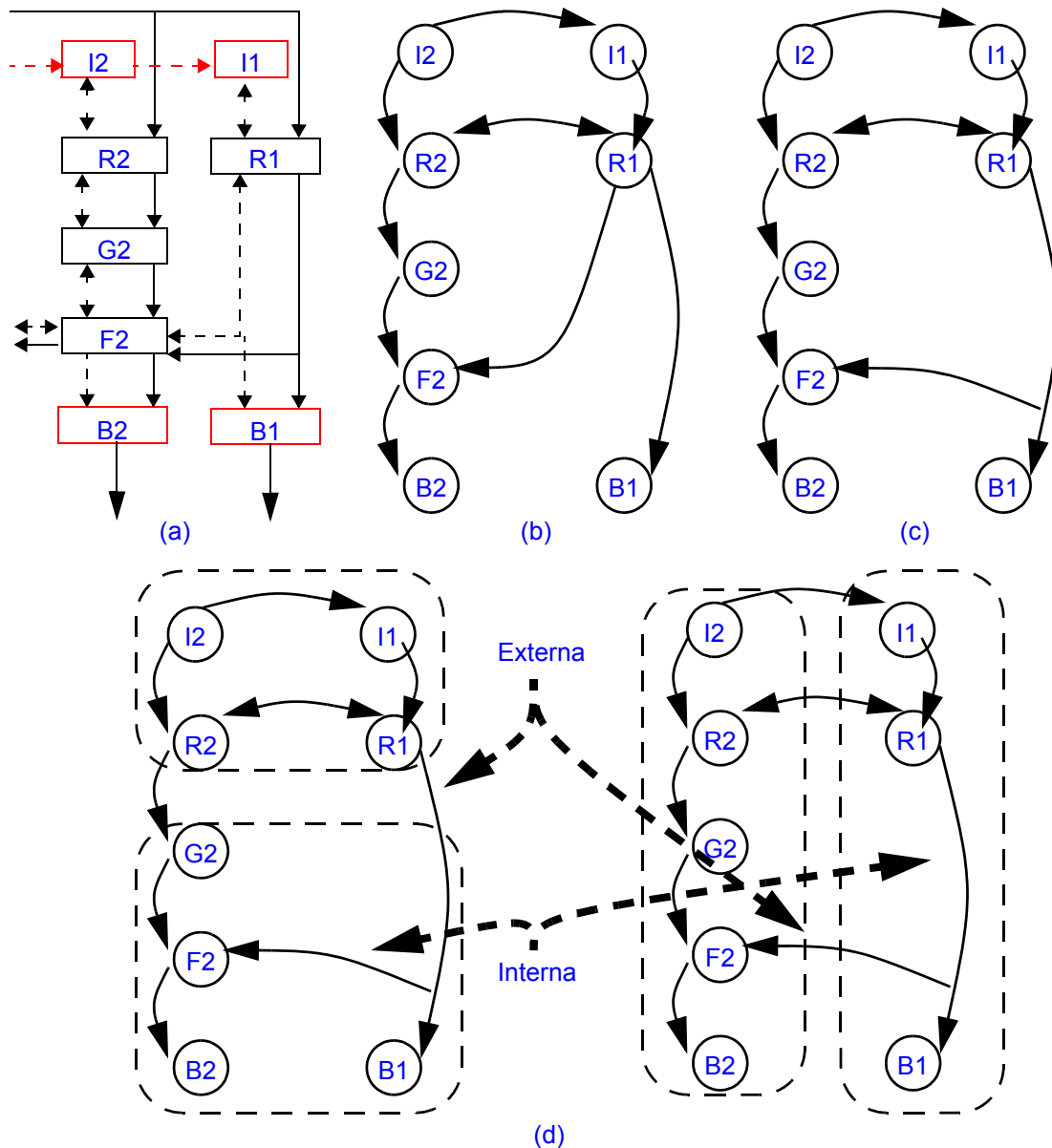


Figura 4.3. (a) Porción del ejemplo de la figura 4.1 que va a ser representado. (b) Representación en forma de grafo. (c) Representación en forma de hipergrafo. (d) Identificación de conexiones internas y externas de un hipergrafo.

A menudo, este tipo de grafos suelen venir etiquetados con unos pesos que indican una determinada preferencia al cambio de grupo. Suelen existir pesos tanto en los módulos como en las conexiones. En el caso de los módulos, su peso generalmente da idea del área que ocupa en el circuito; por lo tanto los módulos de inicio de la figura 4.1 tendrán aproximadamente la mitad del peso que los módulos registros. Con esta consideración podemos obtener una partición lo más equilibrada posible hablando en área ocupada.

En cambio, el peso de los nodos generalmente indica el carácter crítico de un camino, por lo que dicho camino debe ser lo más pequeño posible con el fin de que tenga el menor número posible de dispositivos parásitos.

## 4.2. Algoritmos clásicos

Como ya hemos comentado previamente, vamos a centrarnos en los algoritmos basados en la migración de grupos o movimientos. Esta elección viene motivado por varios motivos, los cuales también han supuesto una mayor expansión de estos algoritmos:

- Por lo general son más intuitivos. La búsqueda de una mejor solución suele ser encontrada a través de pequeños cambios, ya que de esta forma se explora mejor el espacio completo de soluciones.
- Los algoritmos iterativos son más simples de describir y de implementar.
- Los algoritmos basados en movimientos suelen contar con estrategias más sofisticadas para la búsqueda de soluciones.
- Los algoritmos basados en movimientos son independientes de la función objetivo utilizada para medir la calidad de la solución obtenida.

Los algoritmos que vamos a considerar son los siguientes: algoritmo de Kernighan & Lin (KL), algoritmo de Schweikert & Kernighan, algoritmo de Fiduccia-Mattheyses (FM) y una adaptación para el particionado en múltiples grupos.

### 4.2.1. Algoritmo de Kernighan & Lin (KL)

En 1970, Kernighan and Lin presentaron lo que a menudo se ha descrito como la primera bisección heurística *buena*.

La forma de representación del sistema que se utiliza en el algoritmo KL es el grafo con peso. Los pasos que se sigue en el esquema anterior son los siguientes:

- Crear una partición inicial, de tal forma que ambas partes tengan el mismo tamaño.
- El criterio que sigue para llegar a una nueva solución es tal que el número de relaciones existentes entre los dos grupos sea lo más pequeño posible. Para ello calcula las ganancias entre dos módulos de las diferentes partes (los cuales son susceptibles de cambiar de grupo). La ganancia es la suma total de las relaciones externas de ambos módulos menos la suma de relaciones internas de ambos módulos menos el doble de las relaciones existentes entre los dos módulos (esta última resta es debido a que se cuentan dos veces estas relaciones, y siempre estarán presentes), es decir, la fórmula:

$$\text{Ganancia}_{IJ} = \sum \text{relaciones externas}_{IJ} - \sum \text{relaciones internas}_{IJ} - 2 \cdot \text{relaciones}_{IJ}$$

- Si la ganancia es positiva implica que el peso y/o número de relaciones externas es mayor que el de las internas, por lo que dichos módulos deberían cambiarse de grupo. En cambio, si la ganancia es negativa o nula implica que el peso y/o número de relaciones externas es menor o igual que las internas por lo que el grupo en el que se encuentran es correcto.
- De todas las parejas de módulos cuya ganancia es positiva, se elige aquella pareja con mayor ganancia. Los módulos de esta pareja son intercambiados obteniendo una nueva partición.
- Volvemos a repetir todos los pasos exceptuando el primero, pero sin considerar los

módulos que han sido intercambiados previamente (porque se supone que ya están en su grupo correcto). La repetición se llevará a cabo hasta que hayan sido intercambiados todos los módulos o que todas las ganancias sean negativas o nulas.

A modo de ejemplo mostramos el cálculo de la ganancia del grafo mostrado en la figura 4.4. Se ha elegido que todos los módulos y conexiones tengan el mismo peso, por lo que se han excluido del cálculo. Las ganancias han de ser calculadas de todos los módulos de un grupo con todos los del otro. Las particiones que hemos elegido han sido las formadas por los módulos  $\{I2, R2, G2, F2\}$  y  $\{I1, R1, B1, B2\}$ . En el cálculo de las ganancias hemos separado las relaciones de cada módulo por separado, es decir, la fórmula

$$\text{Ganancia}_{I-J} = (\text{rel\_externa} - \text{rel\_interna})_I + (\text{rel\_externa} - \text{rel\_interna})_J - 2 \cdot \text{relación}_{IJ}$$

Vamos a detallar la ganancia de los módulos I1 e I2. Podemos ver que el módulo I1 tiene dos relaciones: una con el módulo R1 (y por tanto interna); otra con el módulo I2 (y por tanto externa). El módulo I2 tiene dos relaciones: una con el módulo R2 (y por tanto interna); otra con el módulo I1 (y por tanto externa). Por último, vemos que existe una relación entre ambos módulos (I1 e I2). Por lo tanto, la ganancia de dicha pareja de módulos es:

$$G_{I1-I2} = (1-1)_{I1} + (1-1)_{I2} - 2 \cdot 1 = -2$$

Se puede apreciar que la partición no es óptima ya que hay más de una pareja de módulos cuya ganancia es positiva, en particular, las parejas I1-F2, B2-I2 y B2-F2. Si todas ellas presentasen la misma ganancia, la elección de la pareja a intercambiar sería aleatoria ya que no tenemos ningún mecanismo para encontrar el mejor intercambio, en cambio, la mayor ganancia se obtiene para la pareja I2-F2, la cual ha de ser intercambiada.

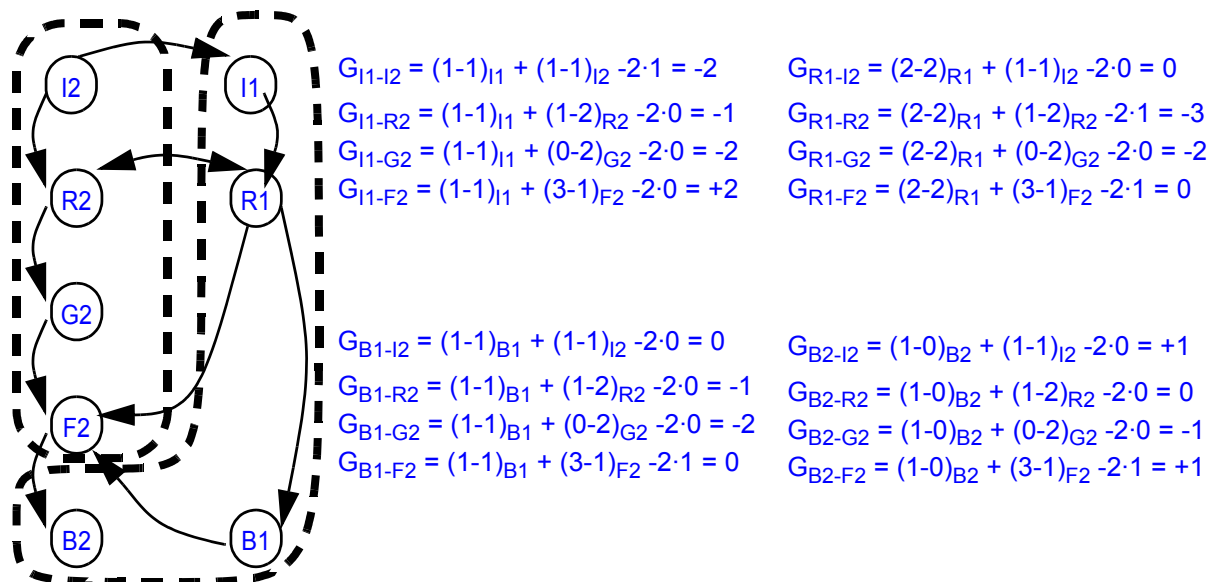


Figura 4.4. Cálculo de las ganancias, según el algoritmo KL, para una determinada partición de un grafo.

Las principales características de este algoritmo son las siguientes:

- Tiempos de ejecución bajos.
- Escapa de algunos mínimos locales.

- Muy dependiente de la partición inicial.
- No garantiza un mínimo global.

#### 4.2.2. Algoritmo de Schweikert & Kernighan

La utilización de grafos no es muy aconsejable cuando tratamos con sistemas electrónicos, ya que un nodo con dos conexiones diferentes tendrán parte del camino en común. Es más, cuanto mayor sea la parte común de ambos caminos, más optimizado estará el circuito en el sentido de que el área ocupada será menor. Por lo tanto, la utilización de un hipergrafo será más conveniente para estos sistemas.

Luego, Schweikert & Kernighan variaron el algoritmo anterior en el sentido de la utilización de un hipergrafo en lugar de un grafo. El resto del algoritmo permanece invariante, es decir, el cálculo de las ganancias, la elección de la pareja de módulos que se deberían intercambiar y el criterio de finalización del algoritmo.

En la figura 4.5 mostramos el cálculo de las ganancias según este algoritmo, es decir, utilizando hipergrafos en lugar de grafos. En el ejemplo utilizando el hipergrafo obtenido del grafo de la figura 4.4, no se obtiene ninguna diferencia con respecto al algoritmo KL, ya que el mismo nodo sólo tiene un módulo destino en su bloque y otro en el complementario. Para apreciar mejor la diferencia, vamos a utilizar los siguientes bloques: {I1, I2, R1, R2} y {G2, F2, B1, B2}, mostrados en la figura 4.5. Los módulos a los que afectará el cambio de la representación son R1, F2 y B1, por lo que mostraremos la comparación de las ganancias de una sola pareja que incluya a alguno de estos módulos; en particular, hemos elegido las parejas R1-I2, R2-F2, B1-G2. Según KL (la parte derecha), el módulo R1 tendría dos relaciones internas (I1 y R2) y dos relaciones externas (F2 y B2), mientras que según SK (la parte izquierda) las relaciones externas pasarían a ser solo una; por lo que la ganancia de la pareja R1-I2 pasaría de -2 a -3. Seguidamente hemos seleccionado el módulo B1, para que sea el distribuidor del nodo en cuestión, es decir, es el único que va a ver la relación externa; por lo tanto, según KL tendrá una única relación externa, mientras que según SK tendrá una relación externa y una relación interna (la distribución hacia F2). Por último, el módulo F2 no vería ninguna relación externa, ya que la señal le llega del módulo B1 en lugar del R1, por lo que únicamente tendrá tres relaciones internas.

Así mismo, las principales características también son comunes a ambos algoritmos.

#### 4.2.3. Algoritmo de Fiduccia-Mattheyses (FM)

Fiduccia y Mattheyses implementaron una variante del algoritmo KL en la cual la principal diferencia estriba en que el movimiento entre los dos grupos es realizado por un solo módulo, por lo que se permite una determinada descompensación en el tamaño de los bloques (así como que la forma de representación es a través de hipergrafo).

El cálculo de la ganancia debe cambiar ya que únicamente debe afectar a un solo módulo. No obstante, es muy similar al anterior conceptualmente. Así, la ganancia será la suma de relaciones externas menos la suma de relaciones internas, es decir,

$$\text{Ganancia}_1 = \sum \text{rel\_externas}_1 - \sum \text{rel\_internas}_1$$

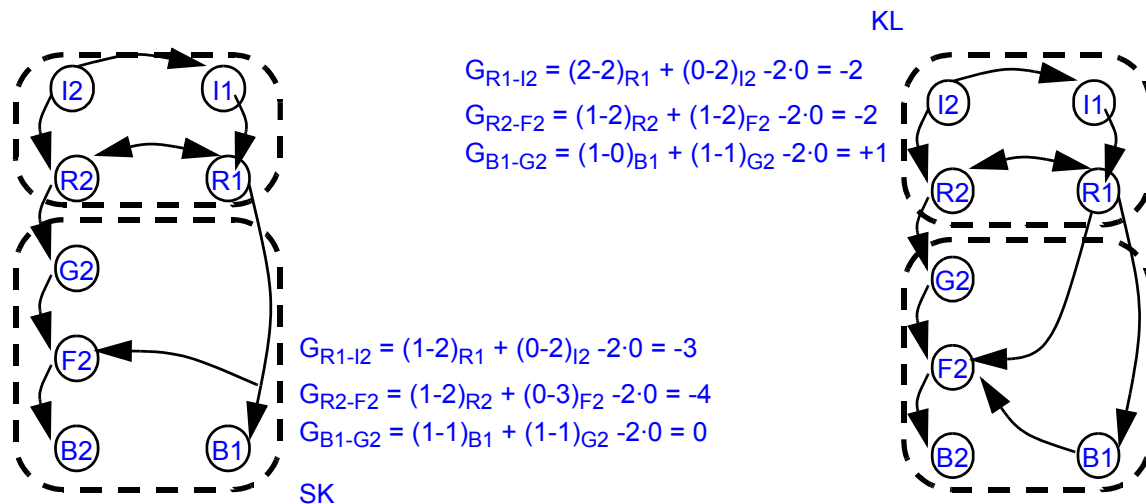


Figura 4.5. Cálculo de las ganancias, según el algoritmo SK, para una determinada partición de un grafo.

Luego, cuando dicho valor es negativo indica que el módulo en cuestión está más relacionado con módulos de su mismo bloque que con los del otro, y por lo tanto, no debe cambiar. Por el contrario, si dicha ganancia es positiva indica que su relación es mayor con el otro bloque, y debería ser cambiado para una mayor optimización.

Otra diferencia con respecto al algoritmo KL es el tratamiento del tamaño de los bloques de la partición. El hecho de mover un solo módulo implica que el tamaño de los bloques no se mantiene constante como sucede en el algoritmo KL. Para escapar de la solución trivial en la que todos los módulos se encuentran en el mismo bloque, se dará una determinada permisividad al tamaño de los bloques, pero no total; generalmente se considera una ratio de tamaño de las dos partes y una holgura, de tal forma que si el tamaño de cada bloque se corresponde con la ratio definida con un error menor a la holgura se considera una solución aceptable.

Los pasos que sigue en el esquema anterior son los siguientes:

- Crear una partición inicial, de tal forma que ambas partes cumplan con el criterio de balance seleccionado.
- El criterio que sigue para llegar a una nueva solución es tal que el número de relaciones existentes entre los dos grupos sea lo más pequeño posible, siempre y cuando la ratio no supere la holgura seleccionada. Para ello calcula las ganancias de cada módulo de las diferentes partes (los cuales son susceptibles de cambiar de grupo). La ganancia es la suma total de las relaciones externas menos la suma de relaciones internas de cada módulo, según la fórmula indicada anteriormente.
- El módulo seleccionado para su cambio es el que presente una mayor ganancia, y con su cambio, los tamaños de los dos bloques esté lo más cercana posible a la ratio seleccionada.
- Una vez que se ha seleccionado un módulo según el punto anterior, el módulo seleccionado se cambia de bloque.
- Volvemos a repetir todos los pasos exceptuando el primero, pero sin considerar el

módulo que ha sido movido previamente. La repetición se llevará a cabo hasta que no se encuentre ningún módulo que se pueda seleccionar según el tercer punto.

A modo de ejemplo mostramos el cálculo de las ganancias según el algoritmo FM en la figura 4.6. Según dichas ganancias, elegimos los módulos de cada bloque con la ganancia más alta, los cuales son B2 y F2. Ambos módulos son susceptibles de cambiar, la decisión de cual de ellos se va a elegir dependerá de la ratio que se obtendría con la nueva partición.

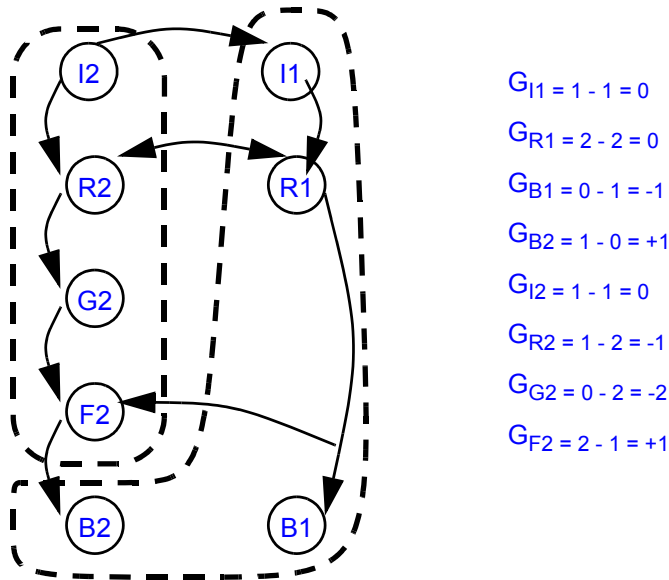


Figura 4.6. Cálculo de las ganancias, según el algoritmo FM, para una determinada partición de un grafo.

Las principales características de este algoritmo son las siguientes:

- El algoritmo es simple, eficiente y rápido
- La calidad de la solución para un elevado número de módulos es pobre
- El algoritmo es el estándar industrial

#### 4.2.4. Algoritmo de particionado de múltiples modos

Los tres algoritmos vistos anteriormente son utilizados para biparticiones, es decir, dividir el sistema completo en dos subsistemas conectados entre sí, de tal forma que dichas conexiones se reduzcan al mínimo. No obstante, el problema de la partición a la hora de implementar un circuito integrado puede requerir (por criterios de optimización) que la partición se lleve a cabo en más de dos bloques.

Una solución directa sería la reiteración de los algoritmos anteriores, centrándose cada vez en uno de los bloques obtenidos en la partición anterior, como se muestra en la figura 4.7. Esta solución presenta los siguientes problemas:

- Si la primera partición obtenida no es buena, dicha situación es heredada por las demás iteraciones.

- El algoritmo trata de minimizar las conexiones externas, es decir, maximizar las conexiones internas; por lo tanto, la reiteración del algoritmo puede ser difícil de llevar a cabo debido a la congestión de nodos en cada grupo.

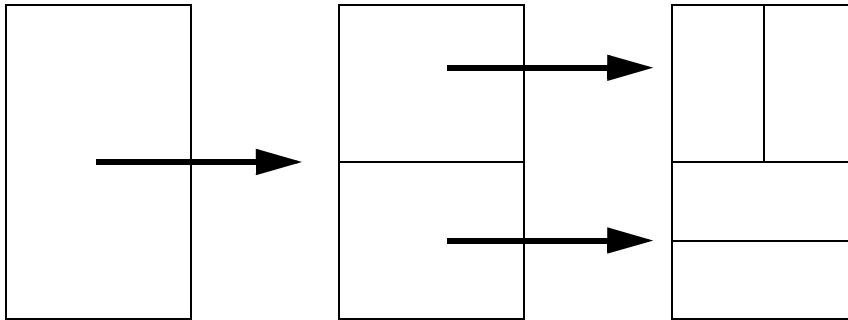


Figura 4.7. Ejemplo de reiteración de algoritmos de bipartición.

### 4.3. Utilización de algoritmos

Para comprobar la utilización de los algoritmos anteriores, vamos a realizar el particionado del esquema de la figura 4.1, cuyos módulos los vamos a etiquetar según la figura 4.8. En dicho esquema los bloques que se encuentran juntos (como el R70 y R71) indican que sólo tienen una conexión entre ambos, luego el bloque R71 tendrá una conexión con el bloque R70 y otra con el bloque R72. Otra consideración que vamos a realizar es que únicamente vamos a considerar las conexiones representadas por las líneas no punteadas, excepto los bloques IX en los que sí se consideraran estas conexiones por ser las únicas.

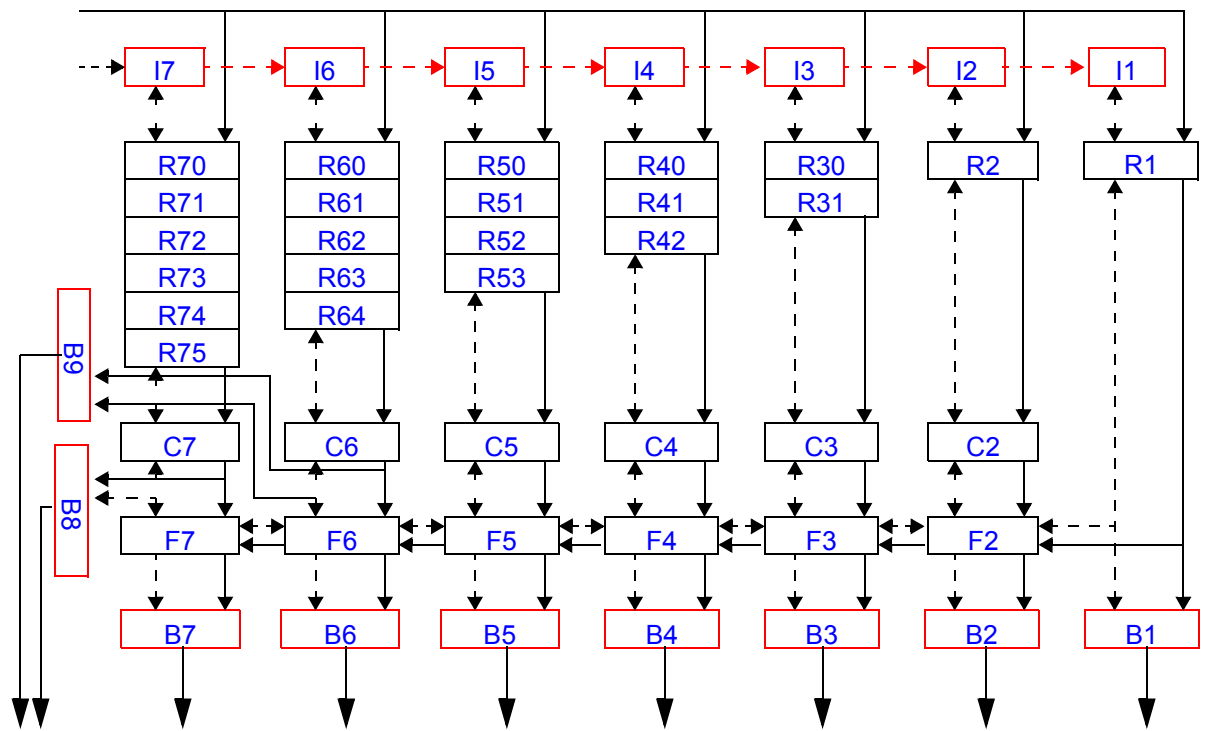


Figura 4.8. Ejemplo de la utilización de los algoritmos de particiones.

En primer lugar vamos a aplicar los tres algoritmos tomando como partición inicial los siguientes grupos:

I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50 R51 R52  
R53 C5 F5 B5 I6 R60 R61 R62 R63 R64 C6 F6 B6 I7 R70 R71 R72 R73 R74 R75 C7 F7 B7  
B8 B9

Dichos grupos se corresponderían a la división vertical del sistema, es decir, como si se hubiese trazado una línea vertical por la mitad del sistema para obtener los dos grupos que lo conformarían. Naturalmente, no se corresponde exactamente al razonamiento anterior para que el tamaño de los dos grupos sean lo más parecido posible.

Para el caso del algoritmo de Kernighan & Lin, calcularemos las ganancias de cada pareja de módulos (cada uno perteneciente a un grupo diferente) del grafo en el que estarían todos los módulos y sus conexiones. En la figura 4.9a mostramos el proceso de particionado de dicho algoritmo. Para mostrar el cálculo de las ganancias, se muestran las dos primeras parejas, que detallaremos a continuación:

- $G_{I1-R53} = (0-2)_{I1} + (1-1)_{R53} - 2 \cdot 0 = -2$
- $G_{I1-C5} = (0-2)_{I1} + (0-2)_{C5} - 2 \cdot 0 = -4$

El módulo I1 está conectado a los módulos I2 y R1, ambos de su mismo grupo. El módulo R53 está conectado a los módulos R52 y C5, cada uno a un grupo diferente. El módulo C5 está conectado a los módulos R53 y a F5, ambos de su mismo grupo.

También se muestran las parejas de módulos cuya ganancia es superior o igual a cero (aunque realmente sólo nos interesan las parejas con ganancia mayor que cero). Seleccionamos la pareja que tiene mayor ganancia (en este caso particular, ganancia 2), pero al haber más de una pareja con dicho valor, hemos seleccionado la primera de ellas, es decir, R52-R60. Dichos módulos son cambiados de grupo y son etiquetados como no cambiables ya que ahora se encuentran en sus grupos correctos. Una vez que se ha realizado el cambio anterior, se vuelven a calcular las ganancias. En este caso, la pareja con mayor ganancia está formada por los módulos R51 y R70, con una ganancia de 4. Volvemos a realizar el intercambio de grupos y el cálculo de las nuevas ganancias (sin considerar los módulos intercambiados: R52, R60, R51 y R70). En este nuevo caso, todas las ganancias son menores o iguales a cero por lo que ya tenemos una partición buena, la cual está formada por los siguientes grupos:

{R60 R70 I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50}  
{R52 R51 R53 C5 F5 B5 I6 R61 R62 R63 R64 C6 F6 B6 I7 R71 R72 R73 R74 R75 C7 F7 B7  
B8 B9}

El caso del algoritmo de Schweikert & Kernighan sólo se diferencia del algoritmo KL en la utilización del hipergrafo como medio de representación en lugar del grafo. Los cambios afectarán a los nodos conectados a más de dos módulos (y por tanto a dichos módulos). En particular son los módulos: R1, R2, R30, R40, R50, R60, R70, C6, C7, F2, F3, F4, F5, F6 y F7. Dicho algoritmo se muestra en la figura 4.9b, utilizando la misma partición inicial que en el caso anterior. En dicha figura mostramos las tres primeras ganancias a modo de ejemplo, en concreto la ganancia de la pareja de módulos I1 y R60. En este caso particular, la partición inicial es elegida como la buena ya que todas las ganancias son menores que cero (razón por la que no aparece en la figura).

Partición inicial:

{I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50 R51 R52}  
 {R53 C5 F5 B5 I6 R60 R61 R62 R63 R64 C6 F6 B6 I7 R70 R71 R72 R73 R74 R75 C7 F7 B7 B8 B9}

I1	R53	-2	R41	R60	0	Intercambiar R52 y	R42	R70	2		
I1	C5	-4	R41	R70	0	R60 (G=2)	C4	R70	2		
I1	R60	0	R42	R60	0	I1	R70	2	F4	R70	2
I1	R70	0	R42	R70	0	B1	R70	2	B4	I6	0
B1	R60	0	C4	R60	0	I2	R70	1	B4	R70	3
B1	R70	0	C4	R70	0	C2	R70	2	I5	R70	3
C2	R60	0	F4	R60	0	B2	I6	0	R51	I6	1
C2	R70	0	F4	R70	0	B2	R70	3	R51	R61	0
B2	R60	1	B4	R60	1	I3	R70	1	R51	R70	4
B2	R70	1	B4	R70	1	R31	R70	2	R60	R70	0
R31	R60	0	I5	R60	1	C3	R70	2	Intercambiar R51		
R31	R70	0	I5	R70	1	F3	R70	0	y R70 (G=4)		
C3	R60	0	R51	R60	0	B3	I6	0	B2	I6	0
C3	R70	0	R51	R70	0	B3	R70	3	B3	I6	0
B3	R60	1	R52	R60	2	I4	R70	1	B4	I6	0
B3	R70	1	R52	R70	2	R41	R70	2			

Partición "óptima":

{R60 R70 I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50}  
 {R52 R51 R53 C5 F5 B5 I6 R61 R62 R63 R64 C6 F6 B6 I7 R71 R72 R73 R74 R75 C7 F7 B7 B8 B9}

(a) KL

I1	R53	-2
I1	C5	-4
I1	R60	-4

Partición "óptima":

{I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50 R51 R52}  
 {R53 C5 F5 B5 I6 R60 R61 R62 R63 R64 C6 F6 B6 I7 R70 R71 R72 R73 R74 R75 C7 F7 B7 B8 B9}

(b) SK

I1	-2
R1	-6
B1	-2
R52	0
R53	0

Partición "óptima":

{I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50 R51 R52}  
 {R53 C5 F5 B5 I6 R60 R61 R62 R63 R64 C6 F6 B6 I7 R70 R71 R72 R73 R74 R75 C7 F7 B7 B8 B9}

(c) FM

Figura 4.9. Ganancias de los módulos según (a) el algoritmo de Kernighan & Lin, (b) Schweikeirt & Kernighan y (c) Fiduccia & Mattheyses.

El caso del algoritmo de Fiduccia & Mattheyses se muestra en la figura 4.9c. En este caso, como solamete se podrá cambiar un módulo por cada iteración, debemos plantear un criterio de balance que indique un mínimo y un máximo para el tamaño de los grupos. El criterio que hemos elegido será que el tamaño de un grupo no podrá ser inferior a la tercera parte de todos los módulos y (por tanto) no podrá ser superior a las dos terceras partes del tamaño total del sistema. Para implementar el algoritmo debemos calcular las ganancias de cada módulo; de tal forma que se selecciona para el cambio el módulo con mayor ganancia y cuyo cambio no incumpla el criterio de balance antes establecido. En este caso particular, la partición inicial se muestra como buena ya que todas las ganancias son menres o guales a cero. n la figura sólo mostramos las ganancias de los tres primeros módulos (a modo de ejemplo), y los módulos cuya ganancia es cero (puesto que no los hay con ganancia superior a cero).

En la figura 4.10 mostramos el mismo ejemplo anterior, pero partiendo de una partición inicial diferente. Dicha partición inicial se ha llevado a cabo mediante una división horizontal, es decir, como si el límite fuese una línea horizontal, de tal forma que se considere que los dos-bloques deben tener el mismo número de módulos. La partición inicial ha quedado compuesta por los siguientes módulos:

I1 I2 I3 I4 I5 I6 I7 R1 R2 R30 R40 R50 R60 R70 R31 R41 R51 R61 R71 R42 R52 R62 R72  
R53 R63

R73 R64 R74 R75 C2 C3 C4 C5 C6 C7 F2 F3 F4 F5 F6 F7 B1 B2 B3 B4 B5 B6 B7 B8 B9

En este caso particular, la aplicación de los tres algoritmos han dado como resultado que la particion inicial es buena, demostrándose la alta dependencia del resultado final con la partición de entrada.

Partición inicial:

{I1 I2 I3 I4 I5 I6 I7 R1 R2 R30 R40 R50 R60 R70 R31 R41 R51 R61 R71 R42 R52 R62 R72 R53 R63}  
 {R73 R64 R74 R75 C2 C3 C4 C5 C6 C7 F2 F3 F4 F5 F6 F7 B1 B2 B3 B4 B5 B6 B7 B8 B9}

I1	R73	-2		R42	R73	0		R72	C4	0		R63	R73	0
I1	R64	-2		R42	R64	0		R72	C5	0		R63	C2	0
I1	R74	-4		R42	C2	0		R72	B1	0		R63	C3	0
R31	R73	0		R42	C3	0		R53	R73	0		R63	C4	0
R31	R64	0		R42	C5	0		R53	R64	0		R63	C5	0
R31	C2	0		R42	B1	0		R53	C2	0		R63	B1	0
R31	C4	0		R72	R64	0		R53	C3	0				
R31	C5	0		R72	C2	0		R53	C4	0				
R31	B1	0		R72	C3	0		R53	B1	0				

Partición "óptima":

{I1 I2 I3 I4 I5 I6 I7 R1 R2 R30 R40 R50 R60 R70 R31 R41 R51 R61 R71 R42 R52 R62 R72 R53 R63}  
 {R73 R64 R74 R75 C2 C3 C4 C5 C6 C7 F2 F3 F4 F5 F6 F7 B1 B2 B3 B4 B5 B6 B7 B8 B9}

(a) KL

I1	R73	-2		R42	R64	0		R53	R73	0		R63	C5	0
I1	R64	-2		R42	C2	0		R53	R64	0				
I1	R74	-4		R42	C3	0		R53	C2	0				
R31	R73	0		R42	C5	0		R53	C3	0				
R31	R64	0		R72	R64	0		R53	C4	0				
R31	C2	0		R72	C2	0		R63	R73	0				
R31	C4	0		R72	C3	0		R63	C2	0				
R31	C5	0		R72	C4	0		R63	C3	0				
R42	R73	0		R72	C5	0		R63	C4	0				

Partición "óptima":

{I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50 R51 R52}  
 {R53 C5 F5 B5 I6 R60 R61 R62 R63 R64 C6 F6 B6 I7 R70 R71 R72 R73 R74 R75 C7 F7 B7 B8 B9}

(b) SK

I1	-2		R53	0		C4	0
I2	-3		R63	0		C5	0
I3	-3		R73	0			
R31	0		R64	0			
R42	0		C2	0			
R72	0		C3	0			

Partición "óptima":

{I1 R1 B1 I2 R2 C2 F2 B2 I3 R30 R31 C3 F3 B3 I4 R40 R41 R42 C4 F4 B4 I5 R50 R51 R52}  
 {R53 C5 F5 B5 I6 R60 R61 R62 R63 R64 C6 F6 B6 I7 R70 R71 R72 R73 R74 R75 C7 F7 B7 B8 B9}

(c) FM

Figura 4.10. Ganancias de los módulos según (a) el algoritmo de Kernighan & Lin, (b) Schweikeirt & Kernighan y (c) Fiduccia & Mattheyses.