

# Aceleración de un algoritmo de enfriamiento simulado mediante particionamiento de redes. Aplicación a “placement” de circuitos VLSI

M.A.Aguirre, A.Torralba y L.G.Franquelo  
Dpto. de Ingeniería Electrónica. Escuela Superior de Ingenieros

Avda. Reina Mercedes s/n, Sevilla-41012 (SPAIN)  
e-mail: aguirre@gte.esi.us.es

## Resumen

Se propone un nuevo método de mejora de los resultados del “placement” de un circuito VLSI. El método propuesto utiliza un particionamiento recursivo para obtener una solución de partida para el posterior proceso de enfriamiento simulado. Para preservar los beneficios de esta solución de partida, la temperatura inicial del algoritmo de enfriamiento es seleccionada del espacio intermedio de las temperaturas. Se presentan resultados experimentales sobre diversos circuitos de prueba, demostrando que el método propuesto mejora el coste final de un 5% respecto de los resultados del programa TimberWolfSC Ver. 6.1. Es más, el tiempo de computación requerido viene a ser alrededor de un 40% menor que el empleado por TimberWolfSC Ver. 6.1.

## 1 Introducción

Para responder a la creciente complejidad de los circuitos VLSI, se necesitan nuevas y más eficientes herramientas de CAD para su diseño. A medida que el número de celdas del circuito crece, también crece la importancia relativa del área de conexionado, haciéndose necesario el desarrollo de herramientas eficientes de *colocación* (placement) y *conexión* (routing).

Una de las técnicas más populares utilizadas en la *disposición* (layout) de las celdas en circuitos VLSI es el conocido Enfriamiento Simulado (*Simulated Annealing*, SA). Desde los primeros trabajos debidos a Kirpatrick, Gelatt y Vecchi [1], las técnicas de enfriamiento simulado han sido sustancialmente mejoradas. SA garantiza buenos resultados si se mantienen ciertas condiciones referidas a la sucesión de estados de equilibrio durante el proceso. En la práctica, se han de realizar un gran número de simplificaciones, y por ello se al-

canzan soluciones casi-óptimas en tiempos razonables. Un buen ejemplo de estas simplificaciones se pueden encontrar en las diferentes versiones del paquete de programas TimberWolf [2]-[3].

Las técnicas de particionamiento y *agrupamiento* (clustering) se han aplicado con éxito a la *disposición* en circuitos VLSI, aportando soluciones cuya calidad es comparable con SA [4]-[5].

El principal inconveniente de la técnica SA es el excesivo tiempo de computación, por ello se están proponiendo nuevas técnicas heurísticas que aceleren el proceso de convergencia. Mallela y Glover [6] propusieron una técnica de SA en dos etapas, utilizando una estrategia de agrupamiento de “abajo a arriba”. Donath et al. [7] proponen una técnica que integra el particionamiento de circuitos y el enfriamiento simulado. Hamada et al. [8] proponen una estrategia de particionamiento jerárquico que habilita un SA multinivel. Posteriormente una ventana móvil se utiliza para compensar el efecto de la calidad del particionamiento.

En este artículo se propone un nuevo método que permite acortar el tiempo de convergencia del SA. A este propósito, el circuito a colocar se agrupa en bloques, que son colocados a lo largo del área de trabajo, siguiendo un método basado en el propuesto en [5]. Las celdas dentro de cada bloque se colocan aleatoriamente. Esta solución puede considerarse como inicial para un proceso de SA, cuya temperatura inicial se fuerza a ser inicialmente baja. Destacamos que esta aproximación utiliza un esquema de agrupamiento “arriba a abajo”, para obtener una buena solución a partir de un número reducido de bloques. Los resultados experimentales obtenidos con el método propuesto y sobre diferentes circuitos de prueba se han comparado con las soluciones obtenidas con TimberWolfSC Ver. 6.1. El coste de las soluciones encontradas a sido menor que el coste obtenido directamente con TimberWolf, con una mejora muy importante en tiempos de computación.

## 2 "Placement" mediante SA

En un proceso de *colocación* mediante SA, se pueden distinguir las siguientes fases:

1. Fase I. *Altas Temperaturas*. En esta fase, un gran porcentaje de movimientos se aceptan.
2. Fase II. *Temperaturas Medias*. Las celdas se agrupan en bloques, que son colocados a través del área de trabajo. Las conexiones más largas se acortan.
3. Fase III. *Temperaturas Bajas*. Las celdas dentro de cada bloque son reordenadas y ubicadas en posiciones cercanas a la final. Se mejoran las conexiones de longitud media y corta.
4. Fase IV. *Temperaturas de Congelación*. Sólo una pequeña fracción de movimientos se aceptan. El coste final no cambia significativamente, en esta fase.

La figura 1 presenta una curva de la evolución de la función de costes en un proceso típico de enfriamiento. Se puede observar que la principal reducción del coste tiene lugar en las fases II y III. Las fases I y IV son necesarias para asegurar la convergencia a un mínimo global de la función de costes. Las fases II y III pueden ser identificadas por la presencia de un valle en la curva conocida como el *calor específico*, y definida como:

$$C(T) = \frac{d\langle E(T) \rangle}{dT}$$

(ver [1]). En la *colocación* de circuitos prácticos, aparecen varios valles, indicando los diferentes niveles de agrupamientos.

## 3 Método Propuesto

Las soluciones obtenidas de SA pueden ser aceleradas y mejoradas reemplazando las primeras dos fases del proceso de enfriamiento por una técnica de agrupamiento y colocación, como la cuadrisección recursiva descrita en [5].

El método propuesto aquí sigue los siguientes pasos:

1. Una técnica de colocación global basada en cuadrisección recursiva, utilizada para agrupar el circuito en bloques y colocarlos en el área de trabajo. La técnica descrita en [5], que incluye la propagación de terminales ha sido la seleccionada para llevar a cabo esta etapa. La técnica de partición utilizada es la conocida como Ratio-Cut, propuesta por Wei y Chen en [9]. En este artículo, la técnica original es adaptada a cuadrisección.

El número óptimo de niveles de cuadrisecciones,  $L$  ha sido empíricamente determinado. Para pequeños circuitos (menores de 500 celdas),  $L = 2$  (es decir, 16 bloques) es suficiente para proporcionar buenos resultados. Para mayores circuitos  $L = 3$  (es decir, 16 bloques) es más aconsejable, aunque una división mayor de bloques ya no varía significativamente los resultados finales.

2. Las celdas, dentro de cada bloque son colocadas aleatoriamente.
3. La solución obtenida de los pasos 1 y 2 será considerada como solución inicial del algoritmo SA. Para poder realizar el proceso de enfriamiento, se ha utilizado TimberWolfSC Ver. 6.1 sin modificaciones, salvo para el caso de la temperatura inicial, que es seleccionable,  $T_{ini}$ . Se puede entender que un inicio a temperaturas elevadas las celdas serían aleatoriamente reordenadas a través de todo el área de trabajo, eliminando los beneficios de la solución inicial. Por otra parte, si esta temperatura es excesivamente baja nos conduciría irremisiblemente a un mínimo local. Por ello se ha seleccionado una temperatura, fija, intermedia. Los resultados experimentales muestran que, los mejores valores finales de la función objetivo se obtienen partiendo de temperaturas en el rango de entre 500 y 1000, por ello hemos fijado la temperatura inicial en  $T_{ini} = 500$  para todos los experimentos mostrados en la siguiente sección.

## 4 Resultados Experimentales

Se comparan los resultados de TimberWolfSC frente a una versión del mismo modificada con las rutinas de compactación y particionamientos. Las condiciones de evaluación de la función objetivo han quedado intactas, de tal manera que sea la comparación válida. En los tiempos de computación no se incluye el tiempo dedicado a los algoritmos de compactación y particionamiento, aunque hay que decir que nunca exceden de un 1 % del tiempo dedicado al enfriamiento simulado. Los resultados experimentales han sido obtenidos utilizando los circuitos de prueba disponibles en el Microelectronic Center de North Carolina (MCNC) [10]. Estos resultados se han comparado con los obtenidos con TimberWolfSC Ver. 6.1 con sus parámetros por defecto.

Los resultados comparativos aparecen en la Tabla I. En esta tabla, el coste de la solución final es

el dado por TimberWolfSC Ver. 6.1, el final del *routing global*. En el paquete no aparece ninguna herramienta de *routing* detallado. Se puede observar que el coste final obtenido con el método propuesto es menor que el obtenido con la versión original de TimberWolfSC Ver. 6.1. Es más, la principal ventaja del método propuesto es la reducción en el tiempo de convergencia, que llega a ser en algún caso del 60% (ver tabla 1).

El estudio del efecto de la temperatura inicial  $T_{ini}$  en los resultados finales se presenta en las figuras 1 y 2, que muestran la curva de enfriamiento de los circuitos *biomed* y *primary2*, con diferentes temperaturas iniciales. La figura 3 se ha obtenido con el método propuesto aquí, mientras que las figuras 1 y 2 se han obtenido con TimberWolfSC Ver. 6.1 con temperaturas iniciales en el rango de 200-100000. A medida que la temperatura inicial es más baja las soluciones tienden a quedar atrapadas en un mínimo local, que está lejos del mejor resultado.

La comparación entre ambos métodos se expone en la figura 4. Las dos curvas discurren paralelas, y puede comprobarse cómo la solución inicial confiere al proceso de una gran robustez.

La Tabla II presenta los resultados para la selección del número de niveles de la cuadrisección  $L$ . Dado que el tiempo requerido en el primer paso del método propuesto es despreciable frente al tiempo total, no existen diferencias significativas en los tiempos totales de convergencia. Para circuitos pequeños, con  $L = 2$  se obtienen los mejores resultados. Para estos circuitos,  $L > 2$  lleva a un óptimo local en la solución inicial. Un razonamiento equivalente se puede realizar para circuitos mayores y con  $L = 3$ .

## 5 Conclusiones

Un nuevo método de aceleración del proceso de convergencia de un algoritmo SA aplicado al *placement* de circuitos VLSI. Para ello se obtiene una solución inicial por medio de un método eficaz de agrupamiento. Los resultados experimentales muestran el método mejora las buenas soluciones características de SA en un tiempo notablemente más corto.

### Agradecimientos

Los autores desean agradecer al Dr. Sechen su colaboración suministrándonos el paquete TimberWolfSC.

## Bibliografía

- [1] S.Kirpatrick, C.D.Gelatt Jr. and M.P.Vecchi. "Optimization by simulated annealing". *Science*, vol. 220, pp. 671-680, May 1983.
- [2] C.Sechen. *VLSI placement and global routing using simulated annealing*. Kluwer Academic Publishers, 1988.
- [3] C.Sechen et al., *TimberWolfSC Version 6.1*, Yale University, 1992.
- [4] A.E.Dunlop and B.W.Kernighan, "A procedure for layout of standard-cell VLSI circuits". *IEEE Trans. Computer-Aided Design*, vol. 4, pp. 92-98, Jan. 1985.
- [5] P.R.Suaris and G.Kedem. "A quadrisection-based combined place and route scheme for standard cells". *IEEE Trans. on Computer-Aided Design*, vol. 8, no. 3, pp. 234-244, Mar 1989.
- [6] S.Mallela and L.Glover. "Clustering based simulated annealing for standard cell placement". *Proc. 25th Design automation Conf.*, pp. 312-317, June 1988.
- [7] W.E.Donath et al. "Timing driven placement using complete path delays". *Proc. 27th Design Automation Conf.*, pp. 84-89, June 1990.
- [8] T.Hamada, C.K.Cheng, and P.M.Chau. "An efficient multilevel placement technique using hierarchical partitioning". *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, vol 39, no. 6, pp. 432-439, June 1992.
- [9] Y.C.Wei and C.K.Cheng. "Ratio-cut partitioning for hierarchical designs". *IEEE Trans. on Computer-Aided Design*, vol. 10, no. 7, pp. 911-921, Jul. 1991.
- [10] M.A.Aguirre, *Nuevas metodologías heurísticas aplicadas al diseño de circuitos integrados*. Tesis Doctoral, May 1994.

Tabla 1: Comparación entre los costes finales y los tiempos de CPU para algunos circuitos de prueba (Benchmarks)

Circuito	Benchmark		TimberWolfSC6.1		Método Propuesto	
	Celdas	Redes	Coste Fin	Tiempo	Coste Fin	Tiempo
fract	149	163	61012	322.1	58410	199.4
primary1	989	1266	871785	754.9	870734	511.4
primary2	3240	3817	3281137	5614.3	2896567	3516.4
biomed	6514	7052	3400182	16659.3	3147166	9061.3

Tabla 2: Comparación de los costes finales con diferentes niveles de particionamiento

Benchmark	16 bloques	64 bloques
fract	58410	-
primary1	873805	870734
primary2	3034370	2896567
biomed	-	3147166

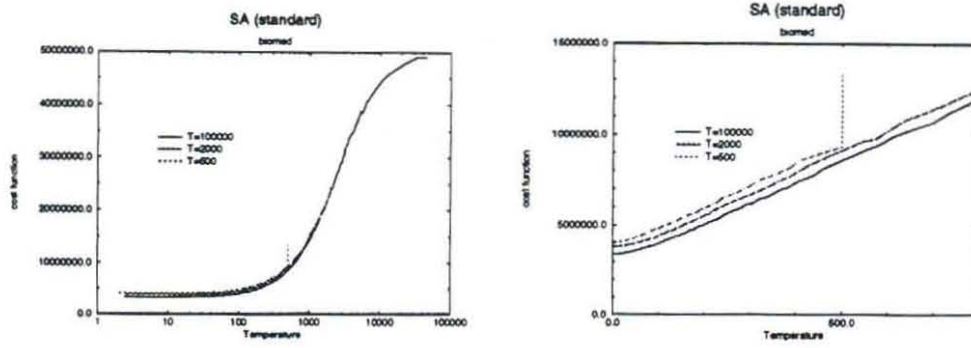


Figura 1: Curvas de enfriamiento para diferentes puntos de partida para un SA estándar. Circuito *biomed*. Curva completa y detalle para bajas temperaturas

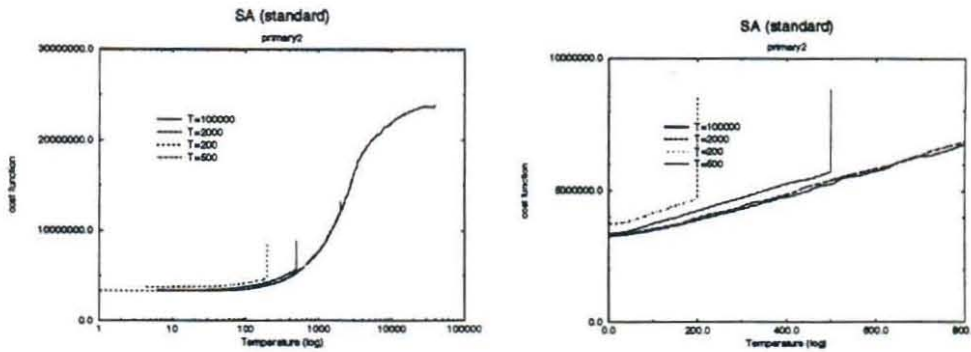


Figura 2: Curvas de enfriamiento para diferentes puntos de partida para un SA estándar. Circuito *primary2*. Curva completa y detalle para bajas temperaturas

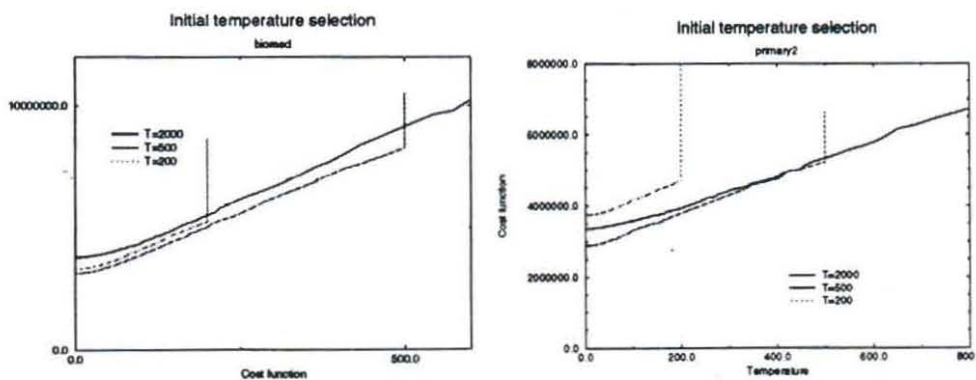


Figura 3: Curva de enfriamiento para diferentes puntos de partida mediante el método propuesto. Circuitos *biomed* y *primary2*. Coste final alcanzado.

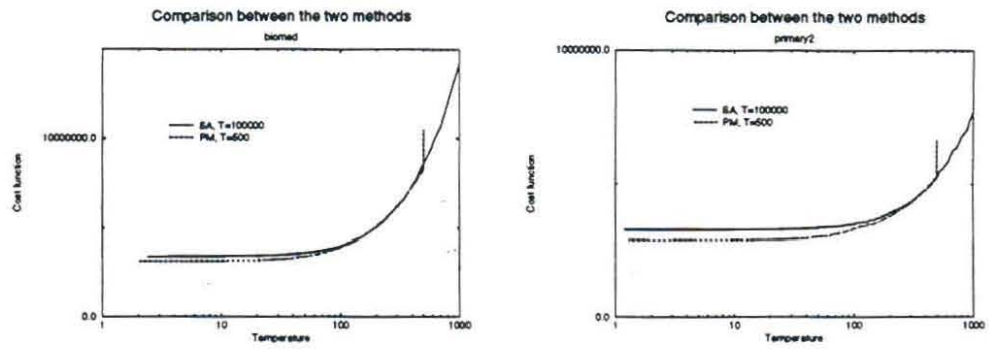


Figura 4: Comparación entre la curva de enfriamiento del SA estándar y del método propuesto para ambos circuitos.