Profundidad de Lógica y Determinismo de las Herramientas de Emplazamiento y Rutado. Algunos Experimentos en FPGAs

Eduardo Boemo y Sergio López-Buedo

ETS Informática, Universidad Autónoma de Madrid, Ctra. Colmenar Km.15, Madrid, España
http://www.ii.uam.es

Elías Todorovich y Nelson Acosta

INCA, Universidad Nacional del Centro, Tandil, Argentina
http://www.exa.unicen.edu.ar

Resumen

En este trabajo se muestra la conexión entre la profundidad de lógica y el determinismo de las herramientas de emplazamiento y rutado. La idea se verifica mediante una serie de experimentos utilizando herramientas y silicio del fabricante de FPGAs Xilinx.

Abstract

In this work, a relationship between logic depth and the determinism of the place and route results is evidenced. A set of experiments on Xilinx FPGAs to verify this idea are presented.

Introducción

Una característica distintiva de las FPGAs de Xilinx es la variabilidad de la frecuencia máxima de operación del circuito, cuando éste se implementa repetitivamente, utilizando las opciones por defecto de emplazamiento y rutado. Este fenómeno es intrínseco del tipo de algoritmos utilizados por la herramienta, derivado de la técnica de *Simulated Annealing* o SA, propuesta por Kirkpatrick *et al*. [1].

La idea principal de la técnica de SA, que en castellano se suele traducir como "recocido simulado", es realizar un número predeterminado de cambios (*swaps*) partiendo de una configuración inicial de emplazamiento. Después de ello, se modifica un parámetro central, llamado "temperatura" (T), y se vuelve a comenzar. Los cambios favorables, es decir, los que minimizan la función de costo (usualmente suma de longitudes o retardos de pista) son aceptados. Por el contrario, los cambios desfavorables tienen una probabilidad de ser aceptados que dependen del valor actual de T y de un número pseudoaleatorio que se obtiene después de cada cambio. El proceso se termina usualmente cuando no se supera un porcentaje estipulado de optimización.

Del párrafo anterior se deduce que cada ejecución del algoritmo SA conducirá a diferentes configuraciones finales de emplazamiento. Tras el posterior rutado se obtendrán circuitos con diferentes características (en particular, ancho de banda y consumo de potencia). La variabilidad del SA ha sido estudiada por Rutermbar [2] e incluso fue aprovechada en las versiones iniciales de las herramienta de Xilinx. En efecto, la opción llamada *aprloop (automatic placement and routing loop*) permitía al diseñador ejecutar la etapa de SA de manera repetida un número arbitrario de veces. De este modo, se podía obtener un circuito con mayor ancho de banda.

Los tecnólogos interesados en esta técnica pueden consultar los libros de Sechen y Wong [3], [4]. Adicionalmente, en la U.C. Berkeley se ha desarrollado un magnifico simulador de SA codificado en Java, que resulta muy interesante desde el punto de vista educativo. Este programa junto con su código fuente puede obtenerse en [5].

Camino Crítico y Profundidad de Lógica

En un circuito convencional, es decir, que no haya sido específicamente diseñado para operar como *Wave Pipeline* [6], la frecuencia de operación está determinada por el camino crítico (*critical path*) o el camino con mayor retardo entre registros.

El concepto de camino crítico incluye al de profundidad de lógica (*logic depth* o *LD*), definido como el máximo número de elementos combinacionales entre dos bancos sucesivos de registros. Para el caso de las FPGAs basadas en

RAM, la profundidad de lógica se expresa en LUTs (*look-up tables*).

Cuando se calcula el retardo de caminos, la relación entre la profundidad de lógica y el camino crítico no es directa, pues al retardo de las LUTs (un valor conocido) hay que agregarle los retardos de las pistas. Éstos últimos dependen de los resultados del emplazamiento y por lo tanto, no son determinísticos.

De acuerdo con el peso de los retardos de pistas y LUTs que componen el camino crítico, los circuitos mapeados en FPGAs se puede separar en dos tipos: grano fino y grano grueso [7]. Para los primeros se cumple que:

- La frecuencia de operación está determinada por un conjunto de caminos independientes (formados por pistas diferentes), de los cuales uno se transforma en el crítico después de un determinado emplazamiento y rutado.
- Cada camino está compuesto por pocas pistas (una o dos).
- Finalmente, las pistas tienen un peso importante (cerca del 50%) en la composición de período mínimo de operación.

Para la segunda categoría, los circuitos de grano grueso, se presenta la situación opuesta:

- El camino crítico surgirá de un conjunto de caminos dependientes.
- Cada camino está compuesto por muchas pistas (más de veinte).
- Las pistas tienen menor peso en la composición del período.

Rent, Distribución Pareto-Levy de Retardos de Pistas y Determinismo del Algoritmo SA

Un lector observador podría preguntarse porque, en los circuitos de grano grueso (circuitos con más LUTs entre registros sucesivos), el peso de las pistas en la composición del período es menor que en los circuitos de grano fino (una o dos LUTs entre registros). Es decir: ¿porqué un camino que tiene únicamente una pista (o dos) y una LUT es diferente de un camino que tiene 20 pistas y 20 LUTs en serie? La respuesta está en la Regla de Rent y la distribución que rige los retardos de pistas de un circuito integrado.

Las FPGAs basadas en RAM son estructuras regulares que ilustran perfectamente la ley empírica conocida como Regla de Rent. Ésta relaciona la cantidad de terminales de una determinada partición de un circuito con la cantidad de lógica que contiene, a través de la siguiente fórmula[1]:

$$t_k \approx AK^P \qquad \text{(Ec. 1)}$$

donde:

- t_k es el número promedio de terminales (*pines* en la terminología de Xilinx) del bloque *K*.
- *K* es el tamaño promedio del bloque (número de unidades que contiene: puertas, circuitos integrados, etc.).
- *A* es el número promedio de terminales de las unidades que componen el bloque.
- *P* constante dependiente de la topología, acotada entre 0 y 1.

Es decir, cuando un determinado circuito se particiona en bloques con *K* puertas, una fracción de las interconexiones quedará confinada dentro de cada bloque, pero existirán t_k terminales que estarán conectados con puertas pertenecientes a otros bloques. La Regla de Rent indica que la relación entre t_k y *K* no depende del tamaño de la partición; sólo es función de la cantidad *A* de terminales de las unidades elementales que agrupa *K* y de un valor *P*, conocido como constante de Rent, que es función de la topología del circuito.

β	t_k	*K*	*A*	*P*
1	3,86	3,63	2,23	0,43
2	4,1	3,68	2,39	0,41
4	4,49	3,96	2,57	0,41
8	4,89	4,28	2,71	0,41
15	5,22	4,46	2,81	0,41

Tabla 1: Parámetros de la Regla de Rent en una FPGA 3090PC84-100. Array Guild 8x8.

Todo circuito que se mapea en una FPGA es partido en dominios *K*, correspondientes a la ocupación media de las *LUT*s. Por ejemplo, en la Tabla 1 se muestran los valores de la ec.1 para un

[1] Este fenómeno fue apuntado por E.F. Rent en diversos informes internos del centro Tom Watson de IBM pero nunca se tradujo en un *paper*. El primer artículo que hace referencia a ella es [8], seguidos de [9, 10], [11] y [12]. Estos trabajos han sido luego utilizados como base por [13] para modelar la interconexión en FPGAs. Trabajos y aplicaciones más recientes de esta regla son [14], [15] y [16].

conjunto de multiplicadores Guild [17] de 8x8 bits y diferentes niveles β de segmentación (*pipelining*). Un valor β=1 indica segmentar en la salida de cada PE del *array*, mientras que β=15 corresponde al circuito combinacional con E/S registrada [18]. Estos datos han sido calculados mediante un programa que extrae el número de puertas, registros, *pins* y CLBs a partir de los resultados de mapeado de la herramienta Xilinx. Obsérvese que *P* resulta prácticamente constante.

En dos artículos clásicos sobre este tema, [9] y [11], se demuestra mediante diferentes modelos, que los circuitos que cumplen con la regla de Rent exhiben una distribución de longitudes Pareto-Levy, de la forma:

$$f_x = \frac{P_1}{x^{P_2}} \quad \text{(Ec. 2)}$$

Es decir, la fracción f_x de pistas de retardo x depende inversamente del valor del retardo. P_1 es una constante de normalización y P_2 está relacionada con la topología del circuito.

En efecto, los retardos de interconexión en una FPGA siguen esta distribución. Por ejemplo, en las Figs. 1 y 2 se puede observar la distribución de pistas para dos *arrays* combinacionales y FPGAs diferentes.

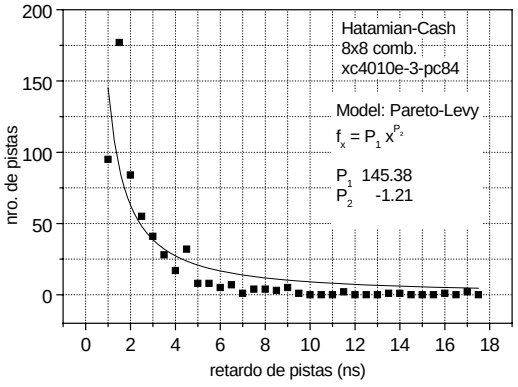

Fig. 1: Aproximación Pareto-Levy para retardos de pistas.Array Hatamian 8x8 [19], β=15. FPGA XC4010PC84.

Si las pistas tienen una distribución Pareto-Levy, se puede inferir que a medida que el grano de la segmentación aumente, el peso del retardo de pistas en la composición del camino crítico tenderá a ser menor que el de las LUTs. Por ejemplo, la suma del retardo de 20 pistas siempre será mucho menor que el retardo de la peor pista multiplicado por 20, pues la distribución de retardos es Pareto-Levy. Sin embargo, el retardo de 20 LUTs siempre es 20 veces el retardo de una LUT.

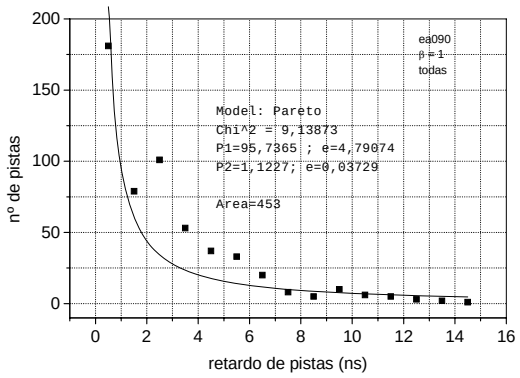

Fig. 2: Aproximación Pareto-Levy para retardos de pistas. Array Guid 8x8 [17], β=15. FPGA XC3090PC84.

Para grano fino no ocurre lo anterior. Cada camino (*path*) tendrá una (o dos pistas) y una LUT. Por lo tanto, la pista que corresponde al camino crítico forzosamente será una de las peores de la distribución (pues las LUTs tienen todas idéntico retardo). Por lo tanto, en este caso, el peso de las pista en la determinación del período de operación será más importante.

Si ahora consideramos que el retardo de pistas tiene un caracter aleatorio, los resultados del SA serán menos determinísticos cuando el peso de las pistas en la composición del período sea significativo. Así, en tecnologías semiautomáticas, la profundidad de lógica del circuito impactará sobre la metodología de implementación. En particular, en un circuito de grano fino, será posible mejorar sensiblemente la velocidad de operación simplemente repitiendo el proceso de implementación. Sin embargo, la misma estrategia resultará ineficaz para grano grueso, aún para un número grande (impracticable) de repeticiones.

Resultados Experimentales

Para verificar empíricamente el análisis anterior, se implementaron 3 multiplicadores de 8x8 bits con la herramienta PAR (*Xilinx Place and Route* M1.4.12). Se realizaron 100 repeticiones (que es el número máximo permitido por este programa) para el dispositivo XC4010E-3-PC84.

Los circuitos de prueba han sido los multiplicadores Hatamian-Cash para β=1 (Fig. 5) y β=16 (Fig. 6),

es decir, grano fino y grueso; y el McCanny-McWhirter [21] segmentado con β=1 (Fig. 7). Ambos circuitos son analizados en detalle en [20]. Finalmente, para mostrar que el efecto es general, se incluyen dos ejemplos basados en el multiplicador de Guild (β=15 y β=1) [22] y compilados con la antigua versión de la herramienta Xilinx XACT 6.1 (Figs. 3 y 4).

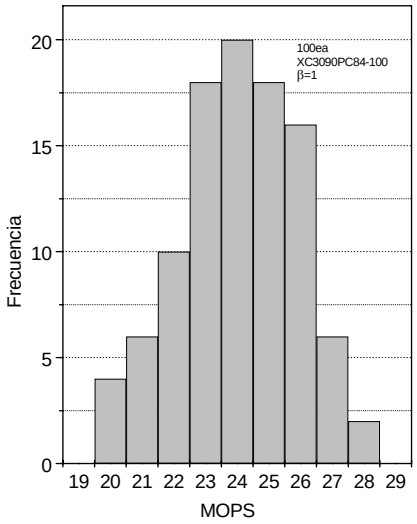

Fig. 3: Frecuencia máxima. Resultados de 100 realizaciones (PPR por defecto) del mult. Guild con granularidad 1, en el dispositivo xc4010e-3-pc84.

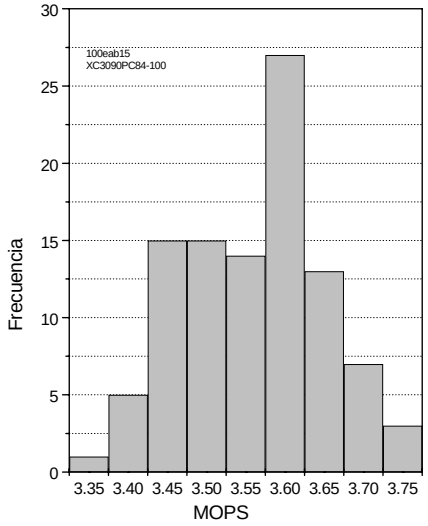

Fig. 4: Frecuencia máxima. Resultados de 100 realizaciones (PPR por defecto) del mult. Guild combinacional, en el dispositivo xc4010e-3-pc84.

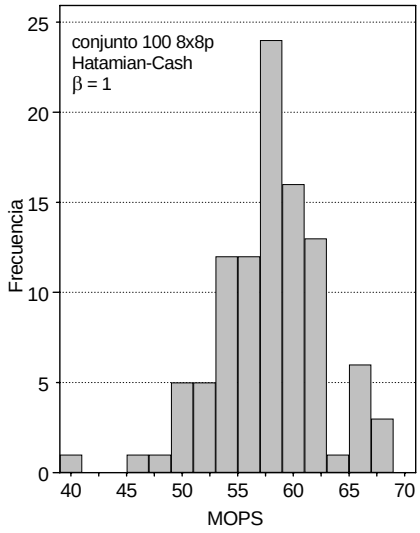

Fig. 5: Frecuencia máxima. Resultados de 100 realizaciones (PPR por defecto) del mult. Hatamian-Cash con granularidad 1, en el dispositivo xc4010e-3-pc84.

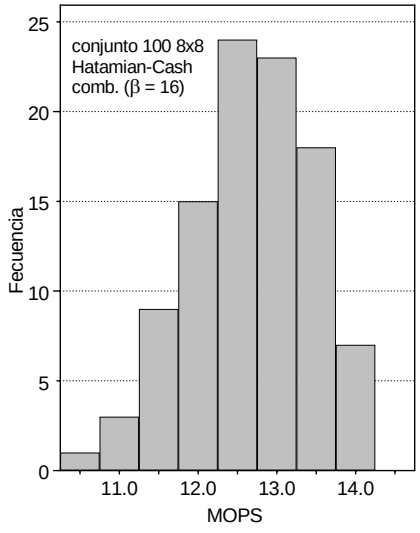

Fig. 6: Frecuencia máxima. Resultados de 100 realizaciones (PPR por defecto) del mult. Hatamian-Cash combinacional, en el dispositivo xc4010e-3-pc84.

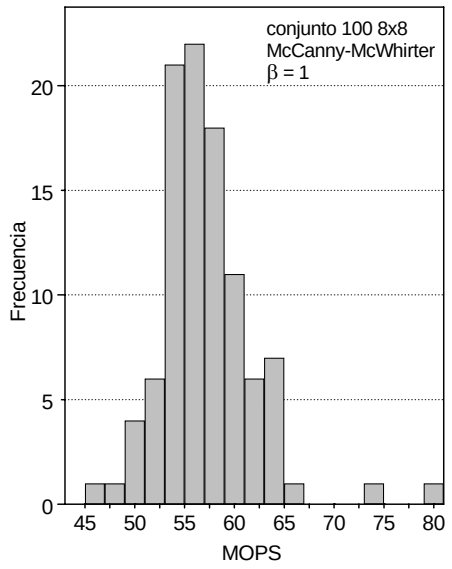

Fig. 7: Frecuencia máxima. Resultados de 100 realizaciones (PPR por defecto) del array McCanny-McWhirter segmentado con granularidad 1, en el dispositivo xc4010e-3-pc84.

Se puede observar que en todos los casos existe variación en los resultados, que es mayor para los circuitos de grano fino (β=1). El caso más notable corresponde al array McCanny-McWhirter β=1 donde el circuito más rápido (80 MHz) supera en un 76% al más lento. Para el caso del array Hatamian-Cash de grano fino ocurre ago similar: se obtiene una ganancia del 75%. Sin embargo, cuando se analizan los datos del mismo circuito pero en su versión combinacional, la ganacia se reduce a solo el 40%.

Finalmente, para el *array* de Guild (Figs.6 y 7), la frecuencia de operación del circuito segmentado con β=1 varía entre 20 MHz y 28 MHz. Es decir, el mejor valor es un 45% más rápido que el peor. Por el contrario, el circuito combinacional (β=15), tiene una frecuencia de operación que varía entre 3.35 MHz y 3.75 MHz. Así, la máxima frecuencia que se obtiene es solo un 11 % mayor que la correspondiente al peor caso

Conclusiones

En este artículo se ha demostrado que el grado de aleatoriedad de diferentes implementaciones de un mismo circuito en FPGAs depende de la profundidad de lógica del mismo. Teóricamente, este fenómeno se debe a que el retardo de interconexión tiene una distribución Pareto-Levy, típica de las estructuras en *array*. Cabe mencionar que en otras tecnologías, por ejemplo Altera, donde el retardo es casi uniforme, no se produce este efecto.

Agradecimientos

Los primeros experimentos sobre este trabajo fueron realizados dentro del proyecto CICYT TIC92-0083 (España) dirigido por el Prof. Juan Meneses. La participación de Nelson Acosta y Elías Todorovich ha sido posible gracias al Programa FOMEC del Banco Mundial. E. Todorovich es becario de CONICET.

Bibliografía

[1] S. Kirkpatrick, C. Gelatt y M. Vecci, "Optimization by Simulated Annealing", *Science*, Vol.220, No.4598, May 1983.

[2] R. Rutenbar, "Simulated Annealing Algorithms: An Overview", *IEEE Circuits an Devices Magazine*, pp.19-26, Enero 1989.

[3] C. Sechen, "*VLSI Placement and Global Routing using Simulated Annealing*", Kluwer Academic Pub. Boston:1989.

[4] D. Wong, H. Leong y C. Liu, "*Similated Annealing for VLSI Design*", Kluwer Academic Pub. Boston:1988

[5] S. Szollar y J. Young, "The incredible Anneal-O-Matic", disponible en http://www-cad.eecs.berkeley.edu/~jimy/classes/ee244/hw2/index.html

[6] E. Boemo, S. Lopez-Buedo, and J. Meneses, "Some Experiments about Wave Pipelining on FPGAs", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.6, No.2, Junio 1998.

[7] E. Boemo, "*Contribución al Diseño de Arrays VLSI con Paralelismo de Grano Fino*", Tesis Doctoral, ETSI Telecomunicación, Universidad Politécnica de Madrid, 1996.

[8] B. Landman y R. Russo, "On a Pin vs. Block Relationship fot Partitioning of Logic Graphs", *IEEE Trans. on Computers*, pp-1469-1479, Diciembre 1971.

[9] W. Donath, "Placement and Average Interconnection Lengths of Computer Logic", *IEEE Trans. on Circ. and Systems*, pp.272-277, Abril 1979.

[10] W. Donath, "Wire Length Distribution for Placemment of Computer Logic", *IBM J. of Res. Development*, vol.25, nº3, Mayo 1981.

[11] M. Feuer, "Connectivity of Random Logic", *IEEE*

Trans. on Computers, pp.29-33. Enero 1982.

[12] D. Ferry, "Interconection lengths and VLSI", *IEEE Circ. and Devices Magazine*, pp.39-42. Julio 1985.

[13] A. El Gammal, "Two-Dimensional Stochastic Model for Interconnections in Master Slice Integrated Circuits", *IEEE Trans. on Circ. and Systems*, Vol. C-28, nº2, pp-127-138. Febrero 1981.

[14] F. Kurdahi y A. Parker, "Techniques for Area Estimation of VLSI Layouts", *IEEE Transactions on Computer-Aided Design*, vol.8, No.1, Enero 1989.

[15] D. Liu y C. Svensson, "Power Consumption Estimation in CMOS VLSI Circuits". *IEEE Journal of Solid-State Circuits*, Vol.29, No.6, pp. 663-670, Jun.1994.

[16] G. Snider, P. Kuekes, W. Bruce y R. Carter, "The Teramac Configurable Computer Engine", *Proc. Fifth Int. Workshop on Field Programmable Logic and Applications*, pp.44-53, Oxford, Reino Unido. Editores: W. Moore & W.Luk. Berlín: Springer-Verlag 1995.

[17] Guild H., "Fully Iterative Fast Array for Binary Multplication and Addition", *Electronic Letters*, pp.263, Vol.5, Nº12, Jun. 1969.

[18] C. Hauck, C. Bamji and J. Allen, "The Systematic Exploration of Pipelined Array Multiplier Performance", *Proceeding ICASSP 85*, pp.1461-1464. New York: IEEE Press, 1985.

[19] Hatamian M. and G. Cash. "A 70-MHz 8-bit x 8 bit Parallel Pipelined Multiplier in 2.5-um CMOS". *IEEE Journal of Solid-State Circuits*. Aug. 1986.

[20] E. Boemo, S. Lopez-Buedo, N. Acosta, and E. Todorovich, "Local versus Global Interconnections in Pipelined Arrays: An Example of the Interaction between Architecture and Technology", *Proc. XIV DCIS Conference*, pp.181-186, Palma de Mallorca, Noviembre 1999.

[21] McCanny J. and McWhirter J., "Completely iterative, pipelined multiplier array suitable for VLSI". *IEE Proc*. pp.40-46. Vol.129, Part G, Nº2. April 1982.

[22] López-Buedo, S.“Web-based Parametrizable Module Generator”. Disponible en http://www.ii.uam.es/~eda.