Metodología orientada a la elección de FPGAs con prioridad en el consumo de potencia

Mora Gutierrez, J.M. Sassaw Teshome, G.
Instituto de Microelectrónica de Sevilla
Centro Nacional de Microelectrónica(CSIC)
Sevilla,España
jmiguel@imse-cnm.csic.es

Jiménez Fernández, C.J., Valencia Barrero, M.
Departamento de Tecnología Electrónica
Instituto de Microelectrónica /Universidad de Sevilla
Sevilla,España
cjesus@imse-cnm.csic.es

***Abstract*— En este trabajo se presenta una metodología de diseño orientada a explorar el cada vez más amplio conjunto de FPGAs con el fin de seleccionar la mejor opción. Los parámetros que se utilizan para realizar la exploración son los recursos consumidos, la frecuencia de operación y el consumo de potencia. Sobre este último parámetro, el más difícil de medir, se hace un especial énfasis. Se exploran dos fabricantes (Altera y Xilinx), dos familias diferentes de cada fabricante y dos subfamilias dentro de cada familia, una de la gama alta y otra de la gama baja. Esta exploración se ha realizado implementando dos circuitos que realizan la operación división de números de 64 bits usando dos algoritmos con plena vigencia.**

Keywords- Caracterización, Herramientas, Metodología, Consumo, Diseño, Divisor NonRestor, SRT, FPGA.

I. INTRODUCCIÓN

En los últimos años hemos asistido a una auténtica explosión de dispositivos programables FPGAs. Los fabricantes han ido incrementando las prestaciones de sus dispositivos con nuevas familias desarrolladas en tecnologías CMOS nanométricas buscando rapidez, mayor densidad de integración y mayor capacidad de computación. Pero estas ventajas, además de abrir un amplio abanico de costes, ofrecen soluciones con muy diversas prestaciones (velocidad, consumo de potencia, etc.), por lo que la elección del dispositivo correcto sobre el que implementar un diseño no es una tarea sencilla. Además, aunque los fabricantes cataloguen sus familias con criterios de costes y prestaciones, las implementaciones concretas pueden no responder a lo esperado.

Para clarificar algo más la influencia de estos parámetros, en esta comunicación se presenta una metodología para la elección de la familia de FPGA teniendo también en cuenta la potencia consumida. Para ello se han implementado dos alternativas de un sistema medianamente complejo, actualmente de gran importancia en multitud de aplicaciones, en concreto, dos algoritmos de división. Estas dos implementaciones se han caracterizado en distintas familias de FPGA y se han medido los recursos consumidos (área), la frecuencia máxima de operación (tiempo) y el consumo energético (potencia).

Con el incremento de la densidad de integración el consumo de potencia se ha convertido en uno de los problemas más relevantes, cuya estimación debe ser realizada con precisión [1][2].

Las herramientas de diseño de dispositivos programables FPGAs también muestran una cada vez mayor diversificación. Por una parte han surgido herramientas independientes de los fabricantes que permiten volcar los diseños en diferentes FPGAs, como es el caso de *Synplify* [3]. Por otra parte, los fabricantes de FPGAs han ido incorporando en su flujo de diseño programas de estimación y cálculo de consumo de potencia cada vez más avanzados y precisos. Esto, aunque aporta evidentes ventajas para el diseñador a la hora de validar sus diseños en dispositivos programables, evitando sorpresas de última hora en el prototipado y diseño arquitectural, también nos ofrece un panorama de cierta confusión sobre la metodología a usar en cada caso concreto: qué herramientas y en qué orden de uso, en qué resultados confiar, o qué tipos de FPGAs seleccionar para el producto final.

El objeto del trabajo que se presenta es establecer y probar una metodología orientada a la elección de FPGAs, en la que debe considerarse el consumo de potencia como uno de los principales parámetros de diseño.

II. ALGORITMOS DE DIVISIÓN

En la literatura se han presentado múltiples algoritmos para la realización de la división en hardware [4][5], entre los cuales los de recurrencia de dígitos, pese a ser de los primeros que se presentaron, siguen estando entre los más utilizados. Estos algoritmos generan un número fijo de bits en cada iteración, y sus implementaciones son, en general, de baja complejidad y consumen pocos recursos, aunque con un tiempo de latencia relativamente alto. Algoritmos clásicos como el de resta sin restauración (*NonRestor*) coexisten con otros más recientes como el *SRT* (Sweeney, Roberttson, y Tocher).

Algunas publicaciones recientes han analizado el comportamiento de estos divisores en tecnologías CMOS

nanométricas [6] y también implementaciones en FPGA [1][2][7].

III. METODOLOGÍA DE DISEÑO

Como ya se ha comentado anteriormente, la metodología de diseño que hemos utilizado emplea tanto herramientas propias de los fabricantes como herramientas externas. Se han seleccionado los dos fabricantes más importantes de dispositivos programables, *Xilinx* y *Altera*. De cada uno de los fabricantes se han utilizado familias diferentes de dispositivos *FPGAs* para poder estudiar la influencia de la tecnología de fabricación, tipo de dispositivo y su coste a la hora de implementar estas arquitecturas de divisores. Con este objetivo se han elegido unas familias optimizadas para bajo coste y otras de altas prestaciones, así como diferentes tecnologías de fabricación de estos dispositivos, 90nm, 65nm y 40/45nm.

A. *Herramientas de diseño, simulación y estimación de potencia.*

El punto de partida es un código VHDL con la descripción a nivel RT independiente de la tecnología. El primer paso es utilizar una herramienta de síntesis y de emplazamiento-conexionado con la elección del dispositivo programable a utilizar.

Se han seleccionado para esta tarea dos herramientas diferentes, la propia ofrecida por el fabricante de dispositivos programables en cuestión y otra independiente como es *Synplify Premier*. Esta herramienta de síntesis física, no está limitada a un solo fabricante, y se caracteriza por la eficacia de sus algoritmos a la hora de sintetizar y su facilidad de uso para la integración y comparación de diferentes familias de dispositivos programables con el mismo software.

En cuanto al cálculo de potencia, la metodología a seguir implica el uso de programas específicos de cada fabricante que están incorporados en sus flujos de diseño. En el caso que nos ocupa las herramientas incorporadas en el software por los fabricante de FPGAs elegidos son, *Altera´s Quartus II Power Play*, [8] para los dispositivos de *Altera* y *Xilinx´s Xpower Analyzer*, [9] para los circuitos programables de *Xilinx*.

Con el propósito de conseguir mayores precisiones en el cálculo del consumo de potencia es necesario incorporar información del índice de actividad en todos los nodos del circuito. Esta información se genera mediante la realización de simulaciones temporales, y es dependiente de la frecuencia de reloj utilizada. El formato utilizado para esta información es *vcd* (*value change dump*) o *saif* (*switching activity interchange*) y, en el caso de Altera es generado por *Modelsim-Altera* y para Xilinx es generado por su propio simulador *ISim*.

B. *Tipos de consumo estimado en FPGAs*

El continuo avance en las tecnologías de integración experimentado en los últimos años ha llevado a un aumento significativo en la densidad y cantidad de transistores de los dispositivos programables FPGAs con el consiguiente aumento de consumo de potencia estático. Las demandas actuales de mayor velocidad y complejidad en los diseños electrónicos han llevado también a un aumento de la potencia dinámica.

El consumo de potencia en FPGA [10][11] es dependiente de la familia y del dispositivo elegido, ya que en cada familia de dispositivos programables se tienen diferentes procesos tecnológicos, tensiones de polarización, arquitectura interna del dispositivo, etc.

C. *Familias de FPGAs utilizadas*

En los dos fabricantes se han escogido familias de bajo coste y familias de altas prestaciones.

En los dispositivos de *Xilinx*, se ha usado dos familias de bajo coste con procesos tecnológicos diferentes: *Spartan3* con tecnología de 90nm y *Spartan6* de 45nm. Para altas prestaciones se han utilizado también dos familias: *Virtex4* de 90nm y *Virtex6* de 40nm.

En *Altera*, se han elegido también dos familias con similares prestaciones a las de *Xilinx*. Como opción de bajo coste se han utilizado las familias *CycloneII* de 90nm y *CycloneIII* de 65nm mientras que para altas prestaciones se han usado *StratixII* de 90nm y *StratixIII* de 65nm.

IV. ANÁLISIS A ALTO NIVEL

Partiendo del código a nivel RT de los circuitos se ha utilizado la herramienta de síntesis física *Synplify Premier* C-2009.06-SP1 de *Synplicity* para conocer qué familias de dispositivos programables son las óptimas para la implementación de los algoritmos.

El esquema del flujo utilizado se muestra en la Figura1.

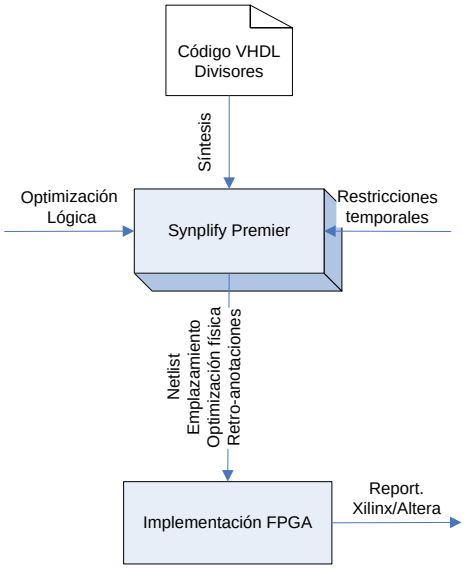

Fig. 1. Proceso de Síntesis con *Synplify*

Synplify Premier nos proporciona como dato de área el consumo en LUTS (*look-up-table*), elemento básico de implementación de lógica combinacional para cualquier

función que requiera 4 o menos entradas. En la Tabla 1 se muestran los resultados de la síntesis física para los diseños realizados.

TABLA 1: SÍNTESIS FÍSICA CON SYNPLIFY PREMIER.

XILINX	*Spartan3*		*Spartan6*	
	f(MHz)	**Area LUT**	**f(MHz)**	**Area LUT**
NonRestor	98	333	101	219
SRT	91	405	118	395
XILINX	*Virtex4*		*Virtex6LP*	
	f(MHz)	**Area LUT**	**f(MHz)**	**Area LUTs**
NonRestor	124	344	119	306
SRT	113	408	150	356
ALTERA	*CycloneII*		*CycloneIII*	
	f(MHz)	**Area LUT**	**f(MHz)**	**Area LUTs**
NonRestor	93	289	125	280
SRT	108	225	135	221
ALTERA	*StratixII*		*StratixIII*	
	f(MHz)	**Area LUT**	**f(MHz)**	**Area LUT**
NonRestor	145	273	177	272
SRT	162	155	200	153

Tanto la arquitectura *NonRestor* como *RST* consiguen las mejores prestaciones de área y velocidad con *Altera* utilizando una familia optimizada en prestaciones y más actual como es el caso de *StratixIII*. En el caso de *Xilinx* estas estimaciones son también ciertas salvo en algún caso y se observa una mejora en las prestaciones de frecuencia y área si se eligen familias con procesos tecnológicos más avanzados, *Spartan6* y *Virtex6*.

V. ESTIMACIÓN DE POTENCIA

En el caso de *Xilinx* hay tres opciones para el estudio del consumo. La primera es utilizar la herramienta de cálculo de potencia por separado, usando los ficheros de descripción del circuito que proporciona *Synplify* (fichero .ncd) junto con el fichero de restricciones y el de simulación. La segunda consiste en abrir el proyecto generado por *Synplify* con la herramienta de *ISE* y ejecutar la herramienta de *XPower*. La tercera es partir del código a nivel RT y seguir todo el flujo de diseño de Xilinx, ejecutando la herramienta de *XPower*.

En el caso de Altera se debe utilizar la herramienta *Quartus* II y abrir el proyecto generado por *Synplify*, para posteriormente ejecutar la herramienta *PowerPlay*.

Todos los diseños mapeados en sus diferentes dispositivos han sido simulados a una frecuencia de 50 MHz de forma que se garantice el correcto funcionamiento de todos ellos. En el caso de Xilinx la herramienta de simulación es la propia ofrecida por el fabricante *ISim* y para Altera la elección es *ModelSim*. Con estas simulaciones se obtiene un fichero de índice de actividad (*saif* o *vcd*) para su uso en la herramienta de estimación de potencia. Las medidas son realizadas a temperatura ambiente de 25ºC y condiciones típica de medida siguiendo el proceso indicado en la figura2.

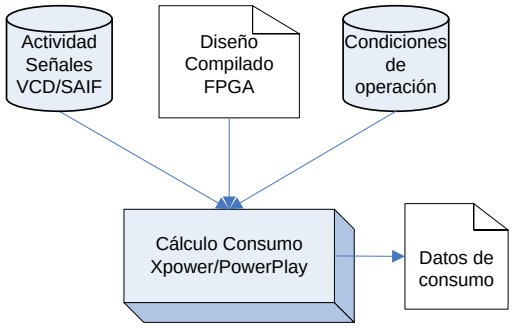

Fig. 2. Proceso de cálculo del consumo con Xilinx y Altera.

Las herramientas ofrecen un desglose de estos resultados finales y así es posible obtener el consumo estimado de los relojes a una frecuencia dada, lógica interna, señales internas y entradas/salidas.

Xilinx ofrece como datos finales la potencia dinámica y estática total. Sin embargo, *Altera* resume los resultados finales como disipación estática, disipación dinámica del núcleo (*core*) y disipación de las entradas/salidas. Para obtener una comparación de los resultados se han sumado estas dos últimas dando como dato final del consumo dinámico la suma del núcleo y de las entradas-salidas para referenciarlo en las tablas.

Las siguientes Tablas 4, 5, 6 y 7 muestran un resumen de la potencia estática y dinámica consumida para cada familia de *FPGAs* de *Xilinx* y de *Altera*, siendo el consumo de las I/O el de mayor peso para el cálculo final de la potencia dinámica total.

En el caso de Xilinx la reducción en el consumo dinámico llega a valores próximos al 4% en familias de bajo coste (*Spartan6*) y al 18% en el caso de familias de altas prestaciones (*Virtex6*). En estos resultados se observan discrepancias en los datos de reducción de potencia para las familias de bajo coste *Spartan* con la arquitectura *SRT* y que no tienen causa justificada.

TABLA 4: REDUCCIÓN EN EL CONSUMO DE POTENCIA DINÁMICA EN *FPGAS DE XILINX*.

XILINX	**Consumo dinámico total (mW)**		*Reducción*
	Spartan3	**Spartan6**	**%**
NonRestor	112	107	-4%
SRT	304	327	7%
	Virtex4	**Virtex6**	
NonRestor	125	102	-18%
SRT	346	285	-18%

En el caso *Altera*, las familias de bajo coste presentan mejores resultados de reducción de potencia que las familias de altas prestaciones, alcanzando valores del 49% en *Cyclone III*, existiendo también un caso anómalo, en esta ocasión con el algoritmo Nonrestor cuando se implementa en la familia *Stratix*.

TABLA 5: REDUCCIÓN EN EL CONSUMO DE POTENCIA DINÁMICA EN *FPGAS* DE *ALTERA*

ALTERA	Consumo dinámico total (mW)		*Reducción*
	CycloneII	**CycloneIII**	**%**
NonRestor	101	63	-38%
SRT	399	204	-49%
	StratixII	**StratixIII**	
NonRestor	74	79	+7%
SRT	305	242	-20%

En cuanto a la potencia estática, los resultados se resumen en las Tablas 6 y 7. Los dispositivos de *Xilinx* muestran dos tendencias en la disminución del consumo estático según la familia elegida. En el caso de dispositivos de bajo coste de la familia *Spartan*, al mejorar el proceso tecnológico de la familia elegida se disminuye el consumo estático, hasta un máximo del 19%. Sin embargo cuando se eligen familias de altas prestaciones, familia *Virtex*, el consumo estático aumenta considerablemente, valores máximos del 225%. Véanse los resultados en la tabla6.

TABLA 6: REDUCCIÓN EN EL CONSUMO DE POTENCIA ESTÁTICA EN *FPGAS* DE *XILINX*

XILINX	Consumo estático total (mW)		*Reducción*
	Spartan3	***Spartan6***	**%**
NonRestor	37	30	-19%
SRT	38	32	-15%
	Virtex4	***Virtex6***	
NonRestor	138	448	225%
SRT	139	450	224%

Para el caso de elegir dispositivos de *Altera*, los resultados muestran que en general se produce un aumento del consumo estático en familias con procesos tecnológicos más avanzados. En el caso de *CycloneIII* los aumentos son del 19% y para *Stratix*III, del 22%.

TABLA 7: REDUCCIÓN EN EL CONSUMO DE POTENCIA ESTÁTICA EN *FPGAS* DE *ALTERA*

ALTERA	Consumo estático total (mW)		*Reducción*
	CycloneII	**CycloneIII**	**%**
NonRestor	47.5	51.8	+9%
SRT	48.1	52.1	+8%
	StratixII	**StratixIII**	
NonRestor	303.5	371	+22%
SRT	305.5	373	+22%

No existe una correspondencia en los resultados entre Xilinx y Altera a la hora de establecer una reducción o aumento del consumo estático al utilizar dentro de la misma familia dispositivos de distinto proceso tecnológico, aunque siempre se produce un aumento del consumo de potencia cuando se utilizan dispositivos de altas prestaciones, *StratixIII* o *Virtex6*.

VI. CONCLUSIONES

En este artículo se ha expuesto una metodología para la elección adecuada del modelo de FPGA basado en una buena estimación de consumo de potencia, área y frecuencia de funcionamiento implementando dos algoritmos de divisores en FPGAs de los fabricantes *Xilinx* y *Altera*.

En los dispositivos de *Altera* se consiguen mejores prestaciones en cuanto a velocidad y área consumida al implementar los algoritmos de divisores seleccionados.

En conclusión, en cuanto a metodología, el uso de *Synplify* permite explorar el espacio de soluciones de fabricantes y familias de FPGAs proporcionando buenos resultados en cuanto a área y frecuencia de operación. Pero la caracterización en potencia debe realizarse con herramientas propias del fabricante.

La elección de las diferentes familias de dispositivos programables impacta principalmente en el consumo estático del dispositivo, aumentando siempre en el caso de elegir dispositivos de altas prestaciones (*Virtex-Stratix*) frente a los de bajo coste (*Spartan-Cyclone*).

Del análisis de los resultados obtenidos también es posible deducir que el consumo de potencia dinámico disminuye en la mayoría de los casos al elegir una familia de mayores prestaciones en los casos de *Virtex6* y *CycloneIII*.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto TEC2007-65105/MIC del Ministerio de Educación y Ciencia y el proyecto TIC-3604 de la Junta de Andalucía.

REFERENCIAS

[1] J. H. Anderson, F. N. Najim "Power Estimation Techniques for FPGAs".IEEE Transactions on VLSI systems, vol12, no. 10, October 2004.

[2] K. Arshak, E. Jaffer, C. Ibala "Power Testing of an FPGA based System Using Modelsim Code Coverage capability". " Design and Diagnostics of Electronic Circuits and Systems, 2007. DDECS '07. IEEE.

[3] Synplify Premier C-2009.06 Software Manuals, available at *www.synplicity.com.*

[4] S.F. Oberman and M.J. Flynn, "Division Algorithms and Implementations", *IEEE Trans. on Computers*, vol. 48, no. 8, pp. 833-854, aug. 1997.

[5] Israel Coren, "Computer Arithmetic Algorithms", Ed. AK Peters Ltd. 2nd edition, 2001.

[6] T.N. Pham, E.E. Swartzlander, Jr., "Design of Radix-4 SRT Dividers in 65 Nanometer CMOS Technology", *Proc. of Application-specific Systems, Architectures and Processors* (ASAP 2006).

[7] G. Sutter, J-P. Deschamps, G. Bioul and E. Boemo, "Power Aware Dividers in FPGA", *PATMOS 2004*, LNCS 3254, pp. 574-584, 2004.

[8] Altera Inc, Altera Quartus II Software Manuals, available at *www.altera.com*, 2009.

[9] "Xilinx Inc, Xilinx ISE 11 Software Manuals", available at *www.xilinx.com, 2009*.

[10] Mouzam Khan, Power Estimation in FPGAs Designs. Altera white paper.

[11] L. Shang. A. S. Kaviani and K. Bathala. "Dynamic Power consumption in Virtex-II FPGA Family". Proceedings of the 2002 ACM/SIGDA International Symposium on FPGA. 2002.