

Implementación eficiente de sistemas GALS sobre FPGAs

Javier D. García Lasheras
Dept. de Ingeniería Eléctrica y Electrónica
ETS de Ingenieros Industriales y de Telecomunicación
Universidad Pública de Navarra
31006 Pamplona
garcialasheras@na2.es

Resumen

Este artículo presenta un nuevo modelo para la implementación de sistemas Globalmente Asíncronos – Localmente Síncronos (GALS) sobre Matrices de Puertas Programables por Campo (FPGA) comerciales. Esta nueva visión está orientada a disminuir la cantidad de lógica extra que trabajos previos en este campo muestran al aplicar técnicas GALS a dispositivos lógicos programables. La aproximación propuesta se fundamenta en un nuevo protocolo de handshaking basado en transiciones de paridad para las tareas de transferencia de datos y generación de reloj. La habilidad de la metodología presentada para seleccionar retardos de forma inteligente permite la implementación de una variedad de nuevas técnicas para la mitigación de interferencia electromagnética y la adaptación a los cambios del entorno del dispositivo.

1. Motivación

Desde la aparición del transistor integrado en la industria electrónica, el rendimiento de los sistemas digitales ha avanzado espectacularmente año tras año. La progresiva miniaturización de los procesos CMOS mantiene este crecimiento debido a que tamaños menores de transistor permiten mayores frecuencias de reloj y la integración de funciones lógicas más complejas en la misma área de silicio. Desafortunadamente, actualmente este paradigma está cambiando. Los transistores son ahora tan diminutos que sus características en términos de consumo de potencia y retardo son comparables con las asociadas a las conexiones establecidas entre ellos. Esta situación fuerza a los diseñadores de circuitos integrados a tener especial cuidado a la hora de implementar interconexiones metálicas. El problema es particularmente crítico cuando consideramos las

enormes redes de distribución de reloj implementadas en diseños hardware complejos. Estas redes no sólo consumen una gran cantidad de potencia, sino que también producen una dañina interferencia electromagnética (EMI) que puede afectar al correcto funcionamiento del propio dispositivo y de los sistemas que lo rodean. Para afrontar esta situación, recientemente se está planteando el uso de sistemas Globalmente Asíncronos – Localmente Síncronos (GALS). En estos sistemas especialmente diseñados, la red de reloj se divide en distintos dominios síncronos mientras que circuitería asíncrona controla la correcta coordinación entre ellos.

En los últimos años, conforme las Matrices de Puertas Programables por Campo (FPGA) son implementadas en procesos situados en el submicrón profundo, podemos extrapolar los problemas anteriormente mencionados a esta clase especial de dispositivos electrónicos. Contrariamente a lo que ocurre con el diseño full-custom, los ingenieros que tratan de implementar sistemas GALS en FPGAs disponibles en el mercado deben enfrentarse con una arquitectura fija que condiciona la forma en que la circuitería asíncrona puede ser implementada eficientemente. Esta situación conduce irremediablemente a un exceso de recursos lógicos usados que limita el uso de la aproximación GALS sobre estos dispositivos para aplicaciones comerciales [3].

En este artículo, intentaremos superar este problema mediante la introducción de una nueva aproximación GALS para dispositivos lógicos programables fuertemente inspirada en el protocolo de control GasP [7] para microtuberías asíncronas [8]. De esta forma, hemos diseñado un Módulo de Rendezvous de Propósito General (GPRM) optimizado para su uso sobre FPGAs con arquitecturas basadas en tablas de look-up (LUT-based) [9]. Este módulo tiene funcionalidades adicionales sobre el control asíncrono estándar como la generación de pulsos de reloj para la

operación autotemporizada (self-timed) de bloques síncronos independientes y la selección en tiempo real de la duración del tiempo de ciclo para implementar técnicas de ensanchado del espectro de EMI y control del throughput.

2. Implementación del protocolo

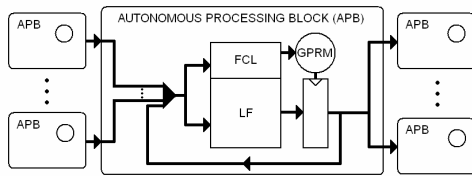


Figura 1. Arquitectura propuesta

2.1. Descripción de la arquitectura

Como es el caso de cualquier aproximación GALS, nuestra metodología está basada en la descomposición del sistema completo en Bloques de Procesado Autónomo (APB). En la Figura 1 podemos ver el esquema general de la arquitectura propuesta centrándonos sobre un APB individual. La parte principal del bloque está compuesta por una función lógica registrada (LF). El valor contenido en el propio registro y aquellos originados por APB externos actúan como entrada para cada función lógica. Debemos remarcar que la función lógica registrada será siempre un diseño síncrono convencional. La forma más común de implementar la LF es usando una descripción HDL sintetizada mediante una herramienta adecuada, pero en la nueva generación de platform-FPGAs la función lógica puede incluir bloques hardware (hard-IP) como motores DSP, bancos de memoria o incluso procesadores completos [9].

Cada bloque de procesado autónomo tiene asociado un GPRM, teniendo el conjunto de ellos la tarea de garantizar el correcto flujo de datos a través del sistema completo. Con este fin, establecemos un enlace entre los módulos de rendezvous de aquellos APBs que pueden intercambiar datos durante la operación del sistema. Adicionalmente, debemos situar un enlace aislado en la forma de lazo cerrado en los GPRMs situados en aquellos bloques de procesado en los que la función lógica tiene su

valor registrado como argumento para su funcionamiento secuencial. A través de estas dos clases de enlaces, el flujo de una serie de permisos proporciona toda la información necesaria para la señalización de la comunicación de datos entre APBs y para la generación de señales de reloj respectivamente.

La Figura 2 enseña el funcionamiento de los enlaces. Si fijamos nuestra atención en aquellos enlaces asociados a la comunicación entre dos bloques de procesado, podemos ver que los respectivos GPRMs comparten un único permiso. Estos enlaces tienen una pluralidad de canales en las dos direcciones de la comunicación, forzando por diseño que cada canal tenga un retardo de propagación distinto. Si miramos ahora al enlace de lazo cerrado, de nuevo existe un único permiso así como un conjunto de canales con distintas características de retardo. Debemos notar que el número de canales en cada dirección de un enlace de comunicación o en un enlace de lazo cerrado no está fijado para presentar el caso de implementación más general.

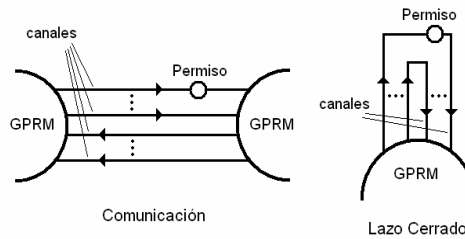


Figura 2. Enlaces usados en el protocolo

La existencia de múltiples canales con diferentes retardos está orientada a permitir la selección determinista del tiempo consumido en cada transferencia de datos en enlaces de comunicación y la duración del ciclo de reloj en enlaces de lazo cerrado. Hay que destacar que para un diseño eficiente usaremos siempre enlaces con un único canal y sólo usaremos enlaces multicanal cuando estemos interesados en usar la habilidad de selección de retardo para propósitos específicos. De esta forma, mediante el uso inteligente de la selección de retardo, podemos controlar la velocidad de trabajo de nuestro sistema o incluso implementar técnicas de mitigación de EMI sin introducir una sobrecarga apreciable de recursos lógicos consumidos.

2.2. Señalización

Dependiendo de la posición de un permiso sobre el enlace, el módulo de rendezvous asociado a un APB interpretará distintos mensajes. Si el enlace controlado está asociado a una salida hacia otro bloque de procesado que debe usar el valor almacenado en el registro, la presencia del permiso indica que el APB de destino ya ha usado el valor almacenado y ha solicitado su actualización. Si el permiso no está presente, el módulo de rendezvous interpreta que el valor almacenado todavía está siendo usado por el bloque de procesado de destino, por lo que el registro no debe ser actualizado.

Si el GPRM está controlando un enlace asociado a una entrada de su bloque de procesado asociado, la presencia del permiso indica que el dato originado por el APB externo ha sido recibido y procesado correctamente y está listo para la actualización del registro. Por otro lado, la ausencia del permiso indica que el dato todavía no está preparado.

Por último, la presencia del permiso en el punto controlado por el GPRM en un enlace de lazo cerrado indica que la información almacenada en el registro ha completado el camino de datos realimentado situado dentro del propio APB y está listo para la actualización del registro. Si el permiso no está presente, el módulo de rendezvous interpreta que el dato está siendo procesado todavía por la función lógica.

Cuando el GPRM detecta la presencia de un permiso en cada uno de los enlaces que controla, interpreta que todos los datos necesarios para evaluar el siguiente valor del registro han sido recibidos y procesados y que todos los bloques de procesado que utilizan el valor almacenado han solicitado su actualización. Bajo estas condiciones, el módulo de rendezvous genera un pulso de reloj que actualiza el dato contenido en el registro.

Antes de este instante, el GPRM debe haber decidido cuáles de los permisos van a ser enviados a través de sus respectivos enlaces y, si es necesario, que canal utilizarán dentro de éste. Para esta labor, el bloque de procesado contiene lógica dedicada para el control de flujo (FCL) que indica al módulo de rendezvous el próximo movimiento de los permisos. Como se muestra en la Figura 1, podemos considerar esta lógica de control como una función diferenciada paralela a la función

lógica principal, pero a efectos prácticos ambas pueden ser fusionadas en una única función global.

Ahora, estamos listos para tomar en consideración el comportamiento dinámico de los permisos. Si consideramos el caso de un enlace asociado a una salida, cuando el módulo de rendezvous decide enviar el permiso al bloque de procesado de destino significa que está enviando también el dato contenido en el propio registro. Para una correcta recepción del dato en el APB de destino, el retardo de propagación del canal implicado en la comunicación debe ser mayor que el tiempo que el dato tarda en viajar desde la salida del registro de origen a la entrada del de destino. Esta clase de señalización es conocida como protocolo asíncrono *bundled-data* [6]. Si el GPRM decide conservar el permiso, esto significa que el dato que acaba de ser actualizado en el registro no es el valor que el bloque de procesado de destino necesita.

Cuando un GPRM decide devolver un permiso asociado a una entrada, el significado de esa acción es una petición de actualización del dato de entrada necesario para el cálculo del siguiente valor del propio registro. En este caso, no es precisa ninguna consideración acerca del retardo del canal escogido. Si el módulo de rendezvous decide retener el permiso hasta que una nueva actualización del registro ocurra, ello implica que el dato presente en la entrada asociada va a ser usado de nuevo por la función lógica.

Finalmente, cuando la información contenida por el registro cambia y cruza el camino lógico realimentado dentro del propio APB, el módulo de rendezvous debe enviar el permiso a través del enlace de lazo cerrado. Para garantizar que la próxima actualización del registro será correcta, el retardo del canal escogido debe ser mayor que el retardo del camino más lento en el bucle lógico de la función. Cabe destacar que si diseñamos el circuito cuidadosamente, en muchos casos podremos eliminar el enlace de lazo cerrado del GPRM. Sin embargo, éste es estrictamente necesario en aquellos bloques de procesado que necesitan trabajar en modo secuencial mientras retienen los permisos asociados a los enlaces de comunicación. De hecho, el enlace de lazo cerrado es la base de la habilidad del módulo de rendezvous para generar ráfagas de reloj controladas.

Como hemos dicho anteriormente, cuando el GPRM detecta la presencia de un permiso en cada uno de los enlaces que esta controlando, éste genera un pulso de reloj. Antes de generar el próximo pulso de reloj, el módulo de rendezvous debe perder al menos uno de los permisos; después, cuando el GPRM recupera el permiso de nuevo, una nueva señal de sincronización es producida. El problema aparece cuando después de generar un pulso de reloj el bloque de procesado no envía ningún dato a algún APB de destino ni ninguna petición de actualización a algún APB de origen, por lo que todos los permisos asociados a enlaces de comunicación permanecen en el GPRM. En esta situación, el enlace de lazo cerrado garantiza la correcta generación de pulsos de reloj para el funcionamiento secuencial del bloque de procesado hasta que el módulo de rendezvous use un enlace de comunicación de nuevo. Para ello, el GPRM envía por el enlace de lazo cerrado el permiso asociado; tras cruzar a través del enlace, el módulo de rendezvous lo recupera de nuevo. De este modo, se produce una pérdida momentánea del permiso que permite la generación de ráfagas de reloj continuas.

2.3. Implementación lógica

A continuación, presentaremos la implementación de los enlaces y el GPRM optimizada para LUT-based FPGAs. Para afrontar la especial arquitectura de estos dispositivos, hemos adoptado un protocolo asíncrono de handshaking de dos fases [6, 8] basado en la propagación de cambios de paridad.

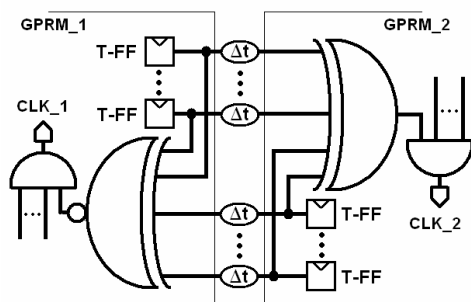


Figura 3. Implementación lógica del enlace de comunicación

En la Figura 3, mostramos el diseño de un enlace de comunicación entre dos módulos de rendezvous. Este está implementado básicamente por un conjunto de de flip-flops de tipo T en cada lado del enlace. En el caso general, tendremos tantos flip-flops T como canales situados en cada dirección del enlace. Inicialmente, todos los flip-flops contienen el valor lógico "0", por lo que la paridad de todos los canales juntos es par en ambos lados del enlace. Para garantizar que sólo haya un permiso en el enlace, situamos una función XOR en un lado del enlace y una XNOR en el otro. Asumiremos que cuando la salida de una de esas dos funciones indica el valor "1" lógico, significa que el permiso está presente en ese extremo del enlace. De esta forma, el permiso estará situado inicialmente en el módulo de rendezvous asociado al lado del enlace de la función XNOR.

Del mismo modo, en la Figura 4 podemos ver la implementación lógica del enlace de lazo cerrado, que es muy similar a la del de comunicación. La principal diferencia es que en el enlace de lazo cerrado sólo está presente la función XNOR, por lo que el permiso estará siempre situado inicialmente en el GPRM asociado.

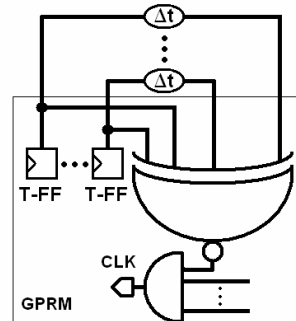


Figura 4. Implementación lógica del enlace de lazo cerrado

Esta clase de enlaces reduce la tarea de diseñar un módulo de rendezvous concreto a añadir una función AND a las salidas de cada función XOR/XNOR asociadas a dicho GPRM. De hecho, el módulo de rendezvous en sí mismo es implementado sobre la FPGA por la función resultante mapeada en tablas de look-up más el conjunto de flip-flops T que están siendo controlados por él.

Cuando todos los permisos asociados a un GPRM están presentes, la función AND adquiere el valor “1” y produce un flanco de reloj ascendente. Esta señal de sincronización se propaga a través de un dominio local de reloj y provoca la actualización de la función lógica registrada y de los flip-flops T. Si el fan-out de la función AND es lo suficientemente alto como para afectar a la integridad de la señal, será necesaria la utilización de un buffer de reloj. Una vez que los flip-flops T han sido actualizados, caminos de realimentación internos del GPRM fuerzan la salida de la AND de nuevo a “0”.

Previamente a la generación del reloj, no sólo la función lógica (FL) ha evaluado el nuevo valor del registro, sino que la lógica de control de flujo (FCL) ha habilitado la transición de aquellos flip-flops T asociados a los canales seleccionados. De esta forma, la señal de reloj provocará un cambio de paridad que se propagará hasta el otro extremo de los enlaces seleccionados con unas características temporales fijadas por el retardo del canal usado. Cabe remarcar que nosotros usamos cadenas de latches para implementar retardos, pero esto puede lograrse de muchas maneras.

Si centramos nuestra atención en enlaces de comunicación, los cambios de paridad inducidos señalizan el movimiento de los permisos de un lado del enlace al otro. Mientras, en los enlaces de lazo cerrado, una transición indica la pérdida del permiso asociado hasta que el cambio de paridad cruce a través del bucle. Gracias a las especiales características de esta señalización por paridad global, los procesos de transición de permisos serán válidos independientemente de los valores adoptados por los flip-flops T de manera individual a lo largo del tiempo.

Antes de acabar esta sección debemos notar que la cantidad de recursos de la FPGA usados por un GPRM pueden ser optimizados fuertemente en función de las características del módulo de rendezvous en sí mismo. Así, no sólo la función lógica de detección de paridad es mapeada de forma óptima en tablas de look-up, sino que diferentes flip-flops T en un GPRM pueden ser redundantes y por lo tanto susceptibles de ser optimizados también. Actualmente, estamos explorando distintas técnicas de particionamiento del sistema en las que herramientas de síntesis convencionales se encargan de efectuar estas optimizaciones de forma automática.

A modo de ejemplo, la Figura 5 muestra la implementación de dos segmentos de tubería straight-forward y sus GPRMs asociados.

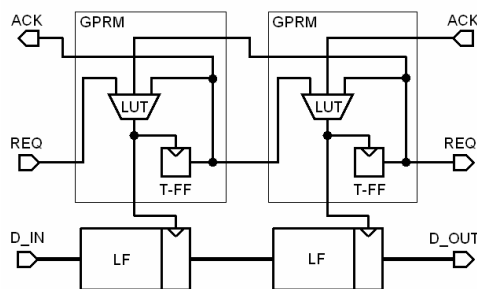


Figura 5. Implementación en FPGA de dos segmentos de tubería asincrónica

3. Resultados experimentales

En esta sección, presentaremos algunos de los más interesantes resultados experimentales obtenidos en las pruebas de laboratorio que hemos efectuado. Hemos trabajado sobre FPGAs Spartan3 y Virtex4 de Xilinx usando el entorno de software gratuito ISE Webpack y, aunque hemos implementado una gran variedad de diseños para validar nuestra metodología [4, 9], la mayoría de las conclusiones empíricas introducidas derivan del comportamiento de tuberías FIFO y en anillo.

3.1. Rendimiento temporal

Como hemos visto en la discusión previa, como cualquier aproximación GALS, nuestra metodología se basa en un conjunto de bloques síncronos convencionales dedicados a tareas de procesamiento coordinados por circuitería asincrónica. En las pruebas de laboratorio que hemos conducido, hemos observado que el rendimiento temporal de nuestra tecnología es lo suficientemente alto como para controlar cualquier bloque síncrono implementado sobre la FPGA. Los valores límite de velocidad alcanzados por los enlaces a 25°C son de 175 MDI/s (comunicación) y 300MHz (lazo cerrado) para una Spartan3-4 y de 550 MDI/s (comunicación) y 700 MHz (lazo cerrado) para una Virtex4-4.

3.2. Mitigación de EMI

La habilidad de nuestros GPRMs para seleccionar la evolución temporal de las transferencias de datos a través del sistema nos permite actuar sobre el espectro de potencia de la interferencia electromagnética (EMI) que la FPGA emite tanto en modo conducido como radiado. En la serie de pruebas SuperpipeS3, analizamos la EMI conducida de un dispositivo Spartan3 que estaba programado con una profunda tubería conmutada síncrona frente a la producida por el mismo diseño usando nuestra aproximación GALS. En esta última tubería, implementamos una técnica de ensanchamiento de espectro que permitía a uno sólo de los enlaces de comunicación tener la habilidad de escoger entre dos posibles valores de retardo mediante la generación de una secuencia pseudo-aleatoria en la lógica de control de flujo asociada [2]. En la Figura 6 podemos ver tanto los resultados experimentales para la EMI conducida en el caso de la tubería síncrona como los de la aproximación GALS -se puede apreciar que una radiación externa produce una interferencia adicional en torno a los 100MHz-

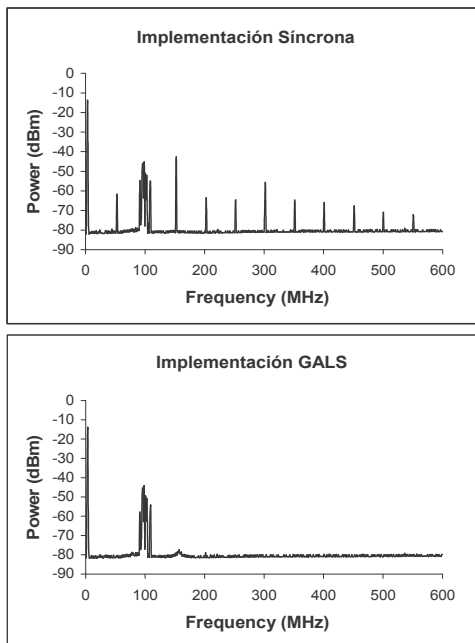


Figura 6. EMI conducida producida por una tubería en configuración síncrona y GALS

3.3. Adaptación a los cambios en el entorno

Las características temporales de la tecnología CMOS dependen en parte de las condiciones de operación del dispositivo en cuestión, siendo remarcable la relación con la temperatura o el voltaje de alimentación [1, 5]. Si centramos nuestra atención en la temperatura, implementamos la circuitería asíncrona con las mismas partes que los bloques de procesado, por lo que cuando ocurre un sobrecalentamiento de la FPGA causando el aumento de los retardos, podemos esperar que la lógica asíncrona se frene y disminuya su velocidad de funcionamiento adaptándose en parte a las nuevas condiciones. Para evaluar este efecto, diseñamos la serie de pruebas HotterV4. En la Figura 7, podemos ver los resultados obtenidos para la variación del throughput en una tubería extremadamente profunda controlada por un GPRM e implementada sobre una Virtex4 cuando forzamos una situación sobrecalentamiento causada por un fallo de refrigeración. Conforme la temperatura del dispositivo sube, el sistema automáticamente va frenando reduciendo la cantidad de calor que el encapsulado debe disipar, alcanzándose finalmente un valor estacionario para throughput y temperatura. Actualmente, estamos conduciendo pruebas similares enfocadas al estudio de la adaptación automática a cambios en el voltaje de alimentación.

En aquellos casos en los que la adaptación automática no es suficiente, otra opción es dar al sistema la habilidad de medir su temperatura o voltaje de alimentación mediante un dispositivo externo. Mediante el uso de esta información, nuestra aproximación GALS puede seleccionar el rendimiento temporal que mejor se adapta a las condiciones del entorno, evitando así el riesgo de fallos asociados a derivas externas.

4. Conclusión

En este artículo hemos explicado en un primer momento cuál es el origen del reciente interés en implementar sistemas GALS sobre dispositivos lógicos programables y cómo las aproximaciones convencionales no son capaces de lidiar de forma eficiente con la arquitectura de las FPGAs disponibles en el mercado. Para superar este problema, hemos introducido un nuevo paradigma

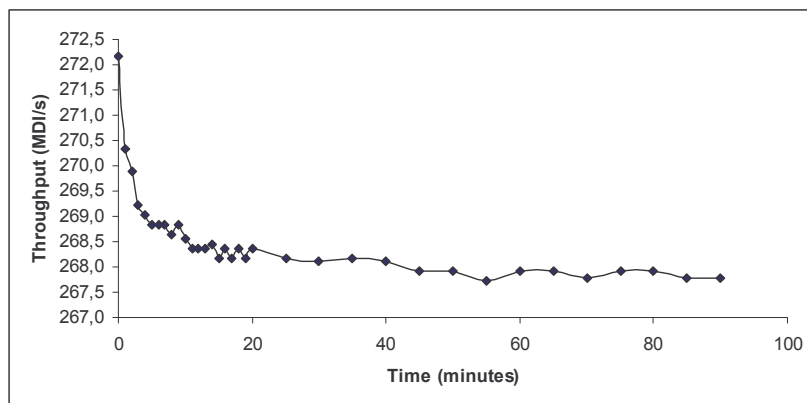


Figura 7. Autocompensación del throughput en una prueba de fallo de refrigeración

GALS basado en el concepto de microtubería y su circuitería de control asociada.

De esta forma, hemos propuesto una partición del sistema en bloques de procesamiento autónomos que descansan en un protocolo de handshaking de dos fases basado en cambios de paridad para las tareas de comunicación y generación de reloj. Este nuevo protocolo de señalización tiene la funcionalidad añadida de selección en tiempo real de retardos. También hemos presentado una implementación eficiente de esta aproximación GALS sobre arquitecturas FPGA comerciales.

Para validar nuestra tecnología, hemos mostrado las conclusiones empíricas más interesantes obtenidas en las pruebas experimentales que hemos llevado a cabo sobre dispositivos Spartan3 y Virtex4 de Xilinx. Así, hemos encontrado que nuestros circuitos sobrepasan ampliamente los requerimientos de velocidad más exigentes. Adicionalmente, y mediante el uso de la habilidad de selección de retardos en tiempo real de nuestros circuitos, hemos desarrollado técnicas eficientes de mitigación de EMI basadas en ensanchamiento de espectro y hemos apuntado hacia una nueva metodología que permite la adaptación inteligente del rendimiento temporal a los cambios en el entorno del dispositivo.

Referencias

- [1] Z. Feng, H. Zhijun, T. Jiarong, and T. Pushan. "An analytical delay model for SRAM-based FPGA interconnections". In *Proceedings of the ASP-DAC '99. Asia and South Pacific Design Automation Conference, 1999*. Page(s): 101 – 104 vol.1. 1999.
- [2] A. Miller, and M. Gulotta. "PN Generators Using the SRL Macro". Xilinx Application Note. XAPP211. 2004
- [3] M. Najibi, K. Saleh, M. Naderi, H. Pedram, and M. Sedighi. "Prototyping globally asynchronous locally synchronous circuits on commercial synchronous FPGAs". In *The 16th IEEE International Workshop on Rapid System Prototyping, 2005*, Page(s): 63 – 69. 2005. <http://www.xilinx.com>
- [4] <http://www.opencores.org>
- [5] M. Ruffoni, and A. Bogliolo. "Direct Measures of Path Delays on Commercial FPGA Chips". IEEE CNF. Page(s): 157 – 159. 2002
- [6] J. Sparso, and S. Furber. "Principles of Asynchronous Circuit Design – A System Perspective". Kluwer Academic Publishers, 2002.
- [7] I. E. Sutherland, and S. Fairbanks. "GasP: a minimal FIFO control". In *ASYNC 2001, Seventh International Symposium on Asynchronous Circuits and Systems*, Page(s): 46 – 53. 2001.
- [8] I. E. Sutherland. "Micropipelines". In *Communications of the ACM, Volume 32, No.6*, pp. 720-738. 1989.
- [9] <http://www.xilinx.com>