

FLUJO DE DISEÑO ASÍNCRONO CON LA BIBLIOTECA DCVSL_LIB PARA ES2 ECPD10

L. Sintés, J. Escudero, M. Peña, O. Roig, J. Cortadella, J. Carrabina.

Teléfono : (93) 581.10.78 / (93) 581.30.82

FAX: (93) 581.30.33

E-mail: laure@cnm.es

RESUMEN.- En el presente trabajo se pretende abordar la metodología a seguir durante el flujo de diseño de un circuito asíncrono orientado a prestaciones, utilizando la biblioteca DCVSL_LIB para aplicaciones asíncronas que hemos desarrollado. Este proceso se aplicará desde un nivel semicustom. La concepción de los algoritmos del sistema se desarrollarán desde una metodología top-down, dadas las restricciones que por construcción debe contener un control asíncrono. Mientras que aspectos tales como el procesado del flujo de datos i control sobre el datapath se aplicaran con una jerarquía botton-up, facilitando la concepción de circuitos asíncronos sencillos i de elevadas prestaciones a partir de soluciones arquitecturales conocidas.

1.- INTRODUCCIÓN:

El uso cada vez más amplio de las aplicaciones microelectrónicas para usos generalizados, es una hecho indiscutible de nuestra sociedad actual. Si bien ello ha sido posible gracias al rápido aumento de las prestaciones y a la disminución de los precios de fabricación de los circuitos integrados (CI), cada vez se exige a estos dispositivos unas condiciones de trabajo más extremas y un rango de aplicaciones más amplio. Esto es así, debido a dos aspectos; por un lado el constante aumento de rendimiento en CIs mediante el aumento de las frecuencias de reloj, de la escala de integración y recientemente de la utilización generalizada de arquitecturas superescalares [1], por otro lado, la inclusión de la tecnología digital en dispositivos tales como telefonía, periféricos, etc. Todo esto fuerza en última instancia modificaciones sobre la tecnología, algo a priori caro. Otra posible solución consiste en replantearse el problema desde una óptica completamente distinta y no síncrona.

La lógica asíncrona ofrece una nueva concepción del diseño de CI versus lógica síncrona que actualmente cobra cada vez más importancia por las soluciones que ofrece en consumo y velocidad frente a diseños síncronos con una misma tecnología, como por ejemplo los diseñados por Phillips para el DCC, *Digital Compact Cassette* [2].

Como también se ha demostrado durante este trabajo la lógica asíncrona permite diseños mucho más robustos que los síncronos frente a modificaciones tales como variaciones tecnológicas, de temperatura, de tensión de alimentación, etc.

Otro importante "handicap" a la hora de diseñar circuitos asíncronos, es la ausencia de demanda de circuitos asíncronos aun habiendo unas necesidades reales de su utilización. Esta ausencia se basa en los siguientes factores:

- * Inexistencia de las herramientas necesarias para la realización de diseños prácticos.
- * Ausencia de personal cualificado. No se realiza una formación de los mismos o de realizarse hasta ahora esta anticuada y es totalmente teórica.
- * Desconocimiento general sobre las propiedades de los diseños asíncronos. Tanto en las peculiaridades del diseño, su modulabilidad, como en las desventajas que pueda conllevar su uso.

2.- OBJETIVOS DEL PROYECTO

Los objetivos fundamentales en este proyecto son dos :

Primero, la integración de las herramientas de concepción de circuitos asíncronos de tal forma que se pueda abordar su diseño desde especificaciones de alto nivel hasta su implementación eficiente a nivel físico. La utilización de este flujo automatizado nos debe garantizar la inexistencia de incoherencias en el complejo proceso de control asíncrono.

Segundo, verificar dicho flujo sobre un circuito demostrador. En concreto se ha implementado un circuito asíncrono para tratamiento de imágenes en tiempo real. El circuito implementa un filtro de ventana 3*3 con coeficientes programables. Este circuito presenta la ventaja adicional de que es utilizado por los alumnos que realizan en la asignatura Diseño de ASICs (4,5 créditos prácticos), en su versión síncrona, con lo que los conceptos asociados al asincronismo y los resultados comparativos tiene una visualización directa en el entorno docente.

Sobre el circuito demostrador se ha puesto especial énfasis en los siguientes puntos:

- * Obtener un circuito de elevadas prestaciones, en cuanto a velocidad, área y consumo.
- * Verificar la metodología de diseño de CI asíncronos, y contrastarla con su equivalente en el campo de los ASICs síncronos.

Así pues, se han utilizado un conjunto de herramientas y una metodología que ha permitido la concepción y el desarrollo de todas las etapas de un CI demostrador.

3.- FLUJO DE DISEÑO ASÍNCRONO ÓPTIMO A NIVEL FUNCIONAL

El diseño a alto nivel se basa en la utilización de herramientas de síntesis como por ejemplo TANGRAM [3,4], que parte de una especificación funcional y permite la descripción del control del circuito sobre grafos de transición de señales o STG ("signal transition graphs") o implementar directamente una netlist estructurada mediante una asociación entre la especificación funcional y la circuitería asociada. Sin embargo, esta alternativa, para arquitecturas clásicas (p.e. unidades de proceso), produce resultados poco óptimos comparados con los que se obtienen mediante una especificación estructural, debido a la

complejidad del control sobre el control del datapath y el control para generar el protocolo autotemporizado.

También cabe la posibilidad de conectar TANGRAM con una herramienta de síntesis, como por ejemplo ASSASSIN [5], SIS [6] o PETRIFY [7] que produce, a partir de los STG de entrada, una netlist a nivel estructural.

4.- BIBLIOTECA DE CELDAS PARA APLICACIONES ASÍNCRONAS

El diseño a nivel estructural se basa en la utilización de una biblioteca de celdas diseñadas expresamente para aplicaciones asíncronas, mucho más eficiente que anteriores implementaciones basadas en "Sea-of-Gates", como se ha confirmado durante el desarrollo del circuito demostrador.

En este nivel se trabaja con herramientas CAD estándar dentro de EURO PRACTICE como el kit de diseño de ES2 para Design Framework II de Cadence, con Verilog-XL como simulador, sobre el que se superpone nuestro kit de diseño. Este hecho facilita enormemente el aprendizaje y la metodología la realización de los diseños.

La biblioteca DCVSL_LIB está pensada para ser utilizada mediante una metodología ASIC semicustom clásica, y en conjunto disponemos de 4 conjuntos de celdas:

- * Celdas DCVSL: Son las mejores celdas para aplicaciones asíncronas siempre que se desee utilizar la lógica dual. Se emplean para la realización del datapath [9] tanto de las unidades de proceso como de control.
- * Celdas CMOS complementarias: Son celdas de apoyo especialmente concebidas para trabajar con las DCVSL. Se utilizan en árboles de detección de los nodos y en partes del control.
- * Celdas de Control: Se basa en las celdas C-MULLER [9] y MUTEX [8]. Confieren un control robusto y fiable al protocolo asíncrono del circuito.
- * Macrocelas: Agrupaciones básicas de celdas para algunas funciones específicas ("soft-macros"). Dentro de este conjunto se presentan algunas soluciones interesantes con el fin de simplificar el proceso de síntesis del circuito.
- * Generales: Celdas de conexión lateral para compatibilizar familias lógicas, así como celdas para aplicaciones de Place and Route (P&R).

La síntesis de control se basa en el modelo de Muller para la especificación de máquinas de estados asíncronas. El diseño del camino de datos está orientado a arquitecturas de tipo "micropipeline" [10], pero de tipo jerarquizado.

5.- METODOLOGIA DE DISEÑO ASÍNCRONO SOBRE EL CIRCUITO DEMOSTRADOR

A nivel funcional: Se ha diseñado la unidad de proceso ("datapath") basada en unidades autónomas funcionales o FAMs [11], sobre las que se ha definido el control. Los autómatas de

control, que utilizan un protocolo de 4 fases, han sido generados y verificados utilizando las herramientas TANGRAM para descripciones basadas en eventos, y ASSASSIN y/o SIS para la generación de una netlist a nivel de puertas. Estas herramientas aseguran la generación de circuitos óptimos en cuanto a su control.

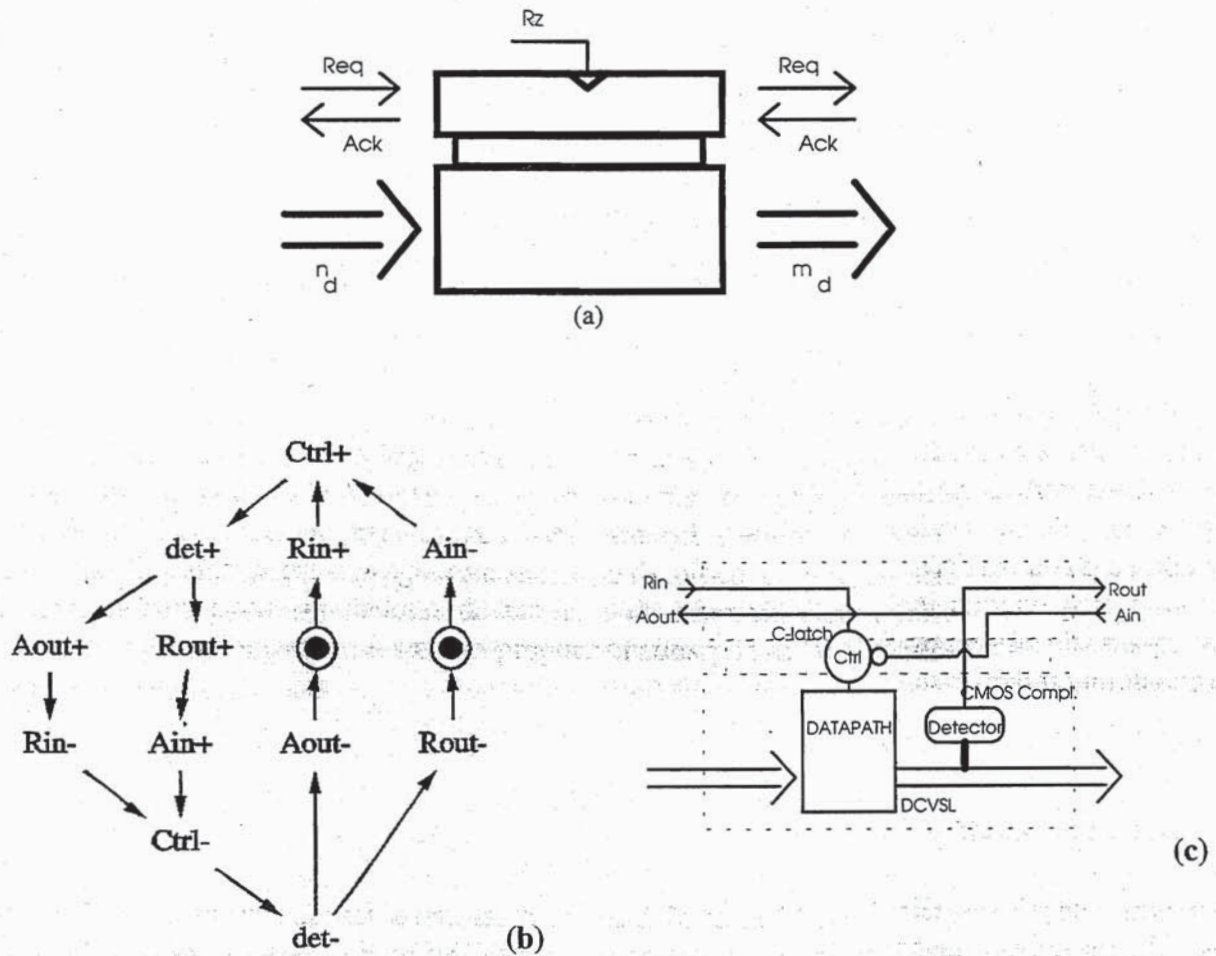


Figura 1.- Ejemplo de una FAM genérica basada en una arquitectura PC0 modificada: (a) símbolo genérico, (b) STG asociado, (c) Arquitectura PC0 modificada para dicha FAM

A nivel estructural: El circuito ha sido realizado dentro de CADENCE Design Framework II, y implementado a nivel de puertas utilizando la biblioteca ECPD10 de ES2, y la biblioteca de celdas DCVSL-LIB de la UAB. Las simulaciones del circuito se han realizado mediante el simulador Verilog, lo que ha permitido comprobar el comportamiento del asincronismo durante el flujo de datos, así como la verificación de las prestaciones de velocidad y la dependencia de la velocidad con los datos.

Las consideraciones de testabilidad se han tenido en cuenta mediante la introducción de un modo de test por el que se puede "congelar" un estado del circuito y controlarlo u observarlo mediante un "scan-path".

A nivel físico: Se ha utilizado la herramienta de "Place&Route" de Cadence "CellEnsamble" para obtener el layout del circuito, con un proceso adicional de separación de las celdas de

ambas bibliotecas mediante una "glue cell" de 4,5 micras de ancho [11]. El circuito resultante tiene 8 mm², con 32 pads.

6.- CONCLUSIONES

La metodología presentada nos permite obtener circuitos muy sencillos y altamente eficaces en velocidad, consumo y área. Alejándonos de concepciones arcaicas en donde se intenta sintetizar controles centralizados e ineficientes, tanto en su síntesis, como en velocidad y área.

Esta nueva visión da prioridad a un control totalmente distribuido y a la estructura de ejecución del datapath. Ya que modelando el tipo de relación entre los módulos funcionales autónomos (o FAM) simplifica enormemente la cantidad de control necesario y de este modo las herramientas de síntesis, como Tangram y Petrify, pueden general de manera fácil el control en varios niveles estructurales (estructuración que en el caso asíncrono presenta la propiedad de ser también de índole físico).

El análisis computacional sobre el circuito demostrador confirma valores de área un 24% menores y una velocidad un 74% mayor (valores estadísticos) que la mejor versión síncrona, realizada por los alumnos de la asignatura de Diseño de ASICs. Estos resultados confirman la idoneidad de la metodología utilizada y el alto grado de eficiencia de la biblioteca de celdas para aplicaciones asíncronas, DCVSL_LIB. De todos modos, estos resultados no son concluyentes al no tenerse en cuenta durante esta valoración la influencia del tipo de algoritmo en concreto utilizado. En estos momentos se está realizando una versión síncrona con los mismos algoritmos que hemos utilizado para la versión asíncrona. De este modo podremos comprobar las diferencias área-velocidad entre un mismo circuito pero distintas lógicas, síncrona-asíncrona.

7.- REFERENCIAS

- [1] "Nuevos Pentium de 90 y 100 Mhz". Barcelona. PCWORD N°98. Sección: Nuevos Productos. Abril 1994.
- [2] Berkel et al. "Asynchronous Circuits for Low Power: A DCC Error Corrector" IEEE Design&Test of Computers, Vol. 11, n°2. Summer 1994.
- [3] M. Peña. **Síntesis de circuitos asíncronos mediante la traducción de circuitos de sincronización a redes de Petri**. Tesina. UPC, Marzo 1995.
- [4] K. Berkel et al. "The VLSI programming language Tangram and its translation into handshake circuits". Proc. ECAD, 1991.
- [5] E. Sentovich et al. **SIS: A system for sequential circuit synthesis**. Technical Report UCB/ERL M92/41. 1992.
- [6] C.Ykman et al. **Concurrency Reduction Transformations on State Graphs for Asynchronous Circuit Synthesis**. IMEC-YCVL93, 1993.
- [7] J. Cortadella, M. Kishinevsky, L. Lavagno & A. Yakovlev, "Synthesizing Petri Nets from state-based models". Proc. of ICCAD'95, pages 164-171, Nov. 1995,
- [8] L. Sintes et al. "Diseño de una Biblioteca de Celdas Asíncronas compatibles con ES2" DCIS'94.
- [9] C. Seitz. "Systems Timming" in Mead & Conway "Introduction to VLSI systems". Ed. Addison-Wesley. 1980.

[10] T. Williams. **Latency and Throughput Tradeoffs in Self-Timed Speed-Independent Pipelines and Rings**. Stanford. Stanford University. Technical Report No.CSL-TR-90-431. August 1990.

[11] J.Escudero. **Metodología y diseño asíncrono. Estudio e implementación de un circuito para el tratamiento de imágenes**. Memoria del proyecto de fin de carrera de Ingeniería Informática. Barcelona, Universitat Autònoma de Barcelona. Febrero 1996. Pag. 224.