Proyecto Puente de Lavado
Una Experiencia de Diseño en el Marco de la Colaboración Universidad-Industria

J. A. Boluda, S. Felici, J. J. Solano, E. de Ves, F. Pardo
Institut de Robòtica
Universitat de València

Persona de contacto: Jose A. Boluda
Institut de Robòtica-Universitat de València
C/ Hugo de Moncada, 4
46010 Valencia (ESPAÑA)
Tel: 34-6-3604484 Fax: 34-6-3616198
email: jboluda@glup.eleinf.uv.es

1 Resumen

*En este articulo se presenta el proyecto "Puente de lavado". Este proyecto es un ejemplo típico de colaboración Universidad-Empresa, en el marco de la promoción de la incorporación de la microelectrónica en la industria. El proyecto ha sido parcialmente subvencionado por la iniciativa Española GAME (**G**rupo de **A**ctivación de la **M**icroelectrónica en **E**spaña), y han intervenido ISTOBAL S.A. como usuario final del producto y SIDSA mas la Universitat de València como diseñadores del sistema, incluyendo los 2 ASICs implementados. En el artículo se describe el sistema completo: la funcionalidad de cada ASIC y el protocolo de comunicaciones diseñado especialmente para este sistema. Se expone la metodología de diseño utilizada haciendo un especial énfasis en las técnicas de síntesis y compilación, las ventajas de Verilog frente a VHDL, la inclusión de macroceldas y la compilación en FPGAs previa a la fabricación de las máscaras. Finalmente se muestran los beneficios económicos y abaratamiento del producto que ISTOBAL S.A. ha obtenido gracias a la incorporación de la microelectrónica.*

2 Descripción del sistema

2.1 La Máquina

El proyecto que se presenta tenía como objetivo el diseño de un nuevo sistema de control para un puente de lavado de coches. El sistema consiste en un conjunto de entradas y salidas tanto analógicas como digitales. Debido a la gran cantidad de características opcionales que puede incorporar la máquina, este conjunto de entradas y salidas puede oscilar entre 100 y 200 en total.

El sistema de control antiguo, de este conjunto de E/S, consistía en un autómata programable con una serie adicional de módulos de entrada/salida. La distancia física entre el módulo de control y una parte de las entradas y salidas obliga a una cantidad de cables excesiva, con todo lo que esto implica: alto coste, poca fiabilidad y lentitud en el montaje.

Otra característica fundamental es la gran cantidad de opciones diferentes que puede incorporar la máquina, además de la gran cantidad de programas diferentes que debe de poder ejecutar. Estas características imponen una solución microelectrónica que sea capaz de combinar flexibilidad y abaratamiento de costes.

El diseño implementado es un sistema maestro-esclavo, con una comunicación entre ellos por un solo cable coaxial (Figura 1). A continuación se describen tanto la unidad central (ASICUC), las unidades periféricas (ASICPER) y el protocolo de comunicaciones hecho a medida.

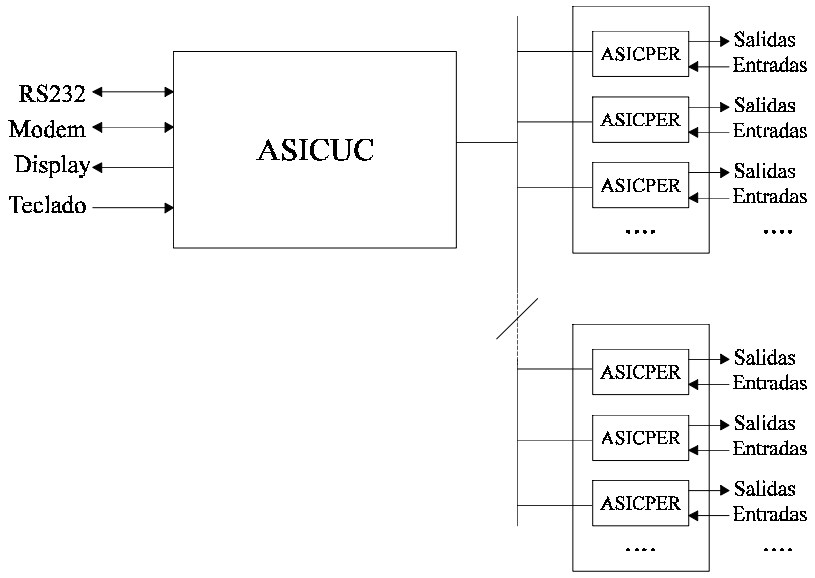

Figura 1. Esquema de bloques del sistema de control de la máquina de lavado

2.2 Las Comunicaciones: Protocolo y tolerancia a fallos

Con objeto de mejorar la calidad del producto aumentando la fiabilidad [10] y mejorando la competitividad reduciendo costes, se han conectado los elementos de entrada y de salida sobre

un único bus de dos hilos que atraviesa el puente de lavado, y va de columna a columna, conectando ASICUC con los ASICPER. A cada entrada y a cada salida se les asigna un ASICPER. Cada ASICPER tiene varias E/S y se diferencia de otro únicamente por su dirección configurada con interruptores. El ASICUC es el master y los ASICPER los esclavos, es decir, cada ASICPER solo contesta a requerimiento del ASICUC.

La comunicación tiene lugar a 9600 baudios, velocidad suficiente para hacer un *polling* de todas las E/S, como se describe posteriormente. Con el fin de obtener una comunicación robusta y evitar interferencias electromagnéticas [12], dado el entorno ruidoso en que está el sistema, se ha hecho una codificación redundante. Se ha utilizado una codificación Manchester, codificación por desplazamiento de frecuencias (FSK) y códigos redundantes de chequeo (CRC) [9]. Las frecuencias utilizadas para la codificación FSK son:

Cero	230.4	KHz
Uno	115.2	KHz
Sincronismo en mensajes enviados por ASICUC	460.8	KHz
Sincronismo en mensajes enviados por ASICPER	57.6	KHz

La trama consta de la información que podemos ver en la siguiente figura.

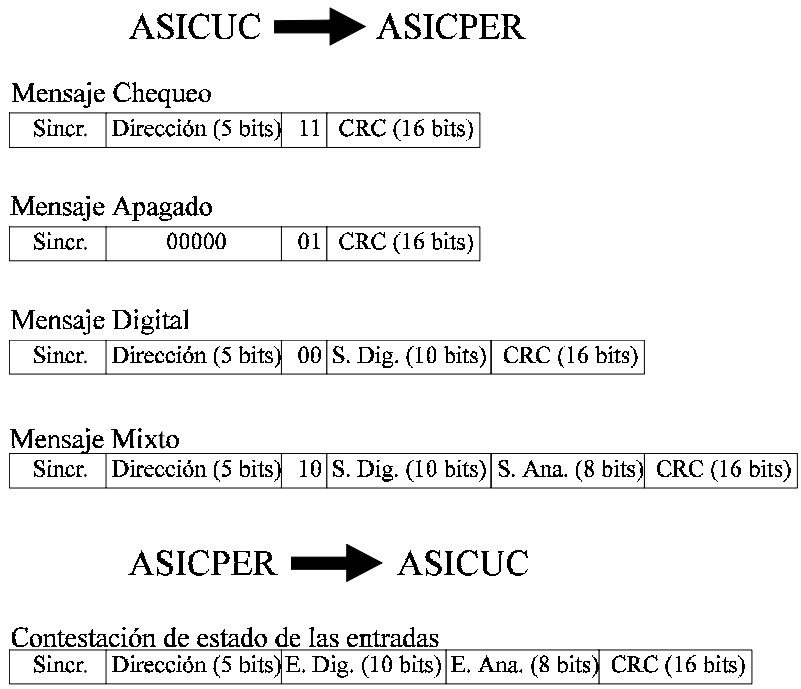

Figura 2. Descripción de mensajes

El módulo de comunicaciones está continuamente haciendo un barrido de los ASICPER. El modo de realizar dicho barrido es enviando mensajes de chequeo comenzando por el ASICPER con dirección 1 hasta el *nmax*. Si la comunicación con uno de los ASICs falla vuelve a enviar el mensaje de chequeo diez veces más. Si falla diez veces consecutivas activa la señal de fallo y deja de realizar el barrido hasta que el módulo microcontrolador envía algún mensaje a la línea, que será usualmente el de apagado general, dependiendo de la E/S que falle se puede hacer un apagado selectivo.

2.3 El ASICUC y el ASICPER

El ASICUC es el que se encarga de la gestión y control del sistema. La versatilidad se incorpora mediante el uso de un microcontrolador que junto con una ROM externa permiten el intercambio de programas y opciones de lavado. El ASICUC está formado por los siguientes módulos (ver figura 3):

- Microcontrolador 8051: Es una macrocelda desarrollada por TreeSoft, en tecnología ES2 CMOS de 1µ. Tiene absoluta compatibilidad con el microcontrolador 8051 de Intel, de esta manera se pueden aprovechar la gran cantidad de herramientas de desarrollo que existen para dicho microcontrolador. El microcontrolador ejecuta el programa externo guardado en ROM, encargándose del control de las E/S y la gestión de todo el sistema.

- Módulo URT: Desarrollado por SIDSA, es la **U**nidad **R**eceptor-**T**ransmisor de señales interfase entre el microcotrolador y las E/S. Gestiona la petición de mensajes por parte de la macrocelda 8051 y efectúa la transmisión-recepción, tal como se explica en el punto 2.2.

- RAM interna: Macrocelda generada por el sintetizador de Cadence. El tamaño es de 256 bytes.

- Módulo Combina: Desarrollado por la Universitat de València. Genera todas las señales de control de los buses y de selección de los elementos externos e internos a ASICUC.

- Reloj en tiempo real: Desarrollado por la Universitat de València. Se utiliza para poder llevar un control de los lavados y poder realizar históricos. Tiene una alimentación por batería externa.

El ASICPER, desarrollado en su totalidad por SIDSA, concentra 10 salidas digitales, 10 entradas digitales, 1 salida analógica y 1 entrada analógica. Debido a que se utilizan 5 bits para codificar la dirección de cada entrada o salida Podríamos tener hasta 32 E/S diferentes, de esta manera usaríamos menos ASICPER para una cantidad fija de E/S. El problema es que entonces el encapsulado para ASICPER pasaría de 44 pines, con lo que aumentaría considerablemente el coste por chip. Reduciendo el número de E/S a las expuestas se consigue mantener un encapsulado de 44 pines PLCC. Cada E/S está aislada galvánicamente mediante optoacopladores, para evitar de esta manera que corrientes inducidas puedan destruir los ASICs.

Todos los módulos excepto las macroceldas han sido descritos en HDL Verilog, simulados parcialmente y conjuntamente con los demás bloques. Finalmente se ha realizado un esquema para conectarlos entre si, este esquema es la parte mas alta de la jerarquía del diseño.

2.4 El software

El diseño específico del módulo URT permite simplificar las operaciones del 8051, de manera que para esta macrocelda, una activación o lectura de una señal, se convierte en una escritura o lectura en memoria.

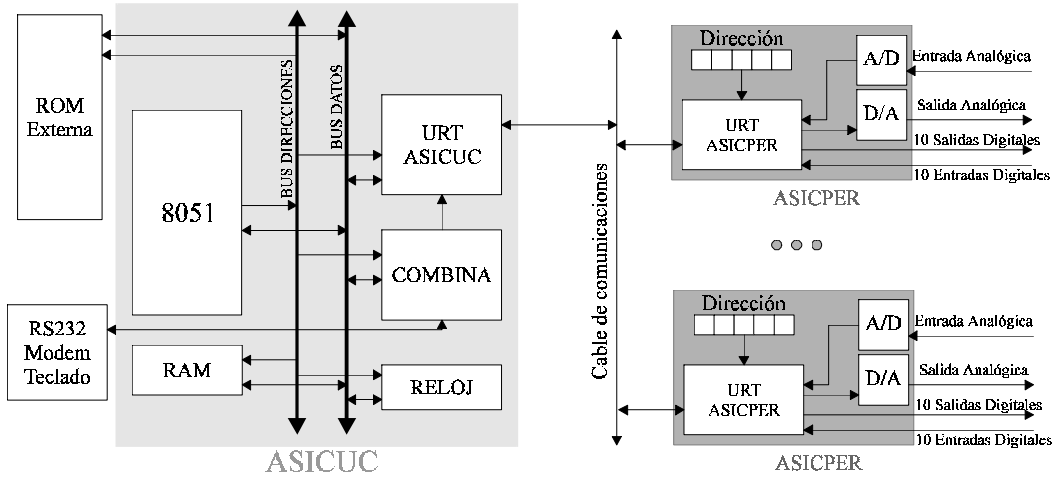

Figura 3. ASICUC y ASICPER

Es necesario conocer todos los procesos que deben ser controlados y las tareas a realizar. Para ello en primer lugar debemos saber qué periféricos habrá en el sistema y qué se espera de ellos. Debemos informarnos de todas las entradas posibles al sistema y cómo debemos actuar según estas. Las tareas a realizar por el software las podemos clasificar en:

- Las tareas de realización del lavado propiamente dicho.
- Control de los menús del display que sirven para configurar muchas de las variables del programa desde el exterior.
- Control de errores, debidos al mal funcionamiento de alguna parte del puente de lavado, o a un fallo en la comunicación con los ASICs periféricos.
- Comunicación a través del módem..
- Comunicación a través del puerto serie con un PC.

A la hora de construir el algoritmo, el mayor problema surge por el hecho de que el 8051 no puede quedarse en ningún momento esperando algo del exterior. Debe controlar demasiadas cosas al mismo tiempo. Por ello el algoritmo, a grandes rasgos, se basa en una pila de eventos. Esto es, el bloque principal de programa hace un *polling* de los posibles eventos del sistema. Si se produce cualquiera de ellos el programa entra en una subrutina que actua en consecuencia. Pero debe ser tan rápida que no se note en ningún momento que el programa no se ocupa del resto de entradas posibles.

Las entradas posibles al sistema, que se controlan mediante interrupciones, se van almacenando en varias pilas, una por cada interrupción. El bloque principal del programa chequea estas pilas. Hacer un algoritmo de este tipo es consecuencia de que el puente de lavado puede estar funcionando (atendiendo a un cliente), y a la vez se puede estar viendo la configuración actual del sistema a través del LCD. Además esto nos ayuda a controlar el display.

El programa total consta de las siguientes partes:

- Interrupción externa: Con ella se controla el teclado. Cada vez que se pulsa una tecla, se guarda en una pila circular. Esta pila es chequeada en el bloque principal. Según el estado del sistema está pulsación provocará acciones distintas.
- Interrupción del *timer*. Con ella se controla cada cierto tiempo las entradas que provocan error. Si se produce se actúa según proceda.
- Interrupción del puerto serie. En el bloque principal es donde realmente se decide qué hacer según los datos que llegan por el puerto serie.
- Bloque principal donde se controlan todos los sucesos .

3 Metodología de diseño

3.1 Compilación de Silicio

El término compilación o síntesis de silicio fue introducido por Johansen [2] en 1979 y significa que un sistema es capaz de producir silicio a partir de una descripción de tipo no máscara. Se puede realizar una clasificación de tipos de compiladores de silicio en función de la estructura del lenguaje utilizado para la descripción del sistema [3]. De esta manera, obtenemos tres tipos de compiladores que son:

- Compiladores de lenguajes estructurales
- Compiladores de lenguajes comportamentales
- Compiladores inteligentes

Los compiladores basados en lenguajes estructurales se limitan a traducir a máscaras un conexionado implícito en el texto. Los compiladores de lenguajes comportamentales se han concentrado en describir lenguajes algorítmicos como Pascal [4] [5] o lenguajes HDL como ISPS [6]. Finalmente los compiladores inteligentes utilizan técnicas mas refinadas que los simplemente comportamentales [7] [8].

No todos los HDLs son adecuados para síntesis. La mayoría de los HDLs están orientados a descripción del hardware y simulación. Earley [13] describe las características que debe de tener un HDL, para que la síntesis de silicio sea realizable de forma transparente al programador.

3.2 Síntesis con Verilog

El diseño tanto del ASICUC como del ASICPER han sido realizados con las herramientas del entorno Cadence DFWII. La descripción de los módulos que componen ambos ASICs ha sido realizada en Verilog XL. La elección de este lenguaje como el mas adecuado para realizar síntesis, viene plenamente justificada por su ventaja actual frente al VHDL [1]. En [13] J. Cooley testea la capacidad de síntesis de Verilog *versus* VHDL mediante la creación de un netlist para un contador up/down módulo 9 con una serie de características especiales.

El Verilog XL es un HDL comportamental, con ciertas restricciones en la síntesis, pero en cualquier caso con una compilación mucho mas directa que con VHDL. La descripción de los módulos está realizada a nivel RTL [11] o nivel de transferencia entre registros lo cual garantiza el que se pueda efectuar la síntesis. Además en el caso de *Synergy*, que es el

compilador de Cadence para Verilog, se pueden generar varias soluciones optimizando o bien la velocidad o bien el área de silicio utilizada.

Para realizar la síntesis se deben de seguir una serie de reglas que eviten en la descripción del circuito partes no sintetizables, de esta manera no se deben de incluir operaciones matemáticas complejas, o funciones lógicas no totalmente definidas. Finalmente se selecciona la librería de síntesis (ES2 CMOS 1μ) y se realiza la síntesis de los módulos por separado.

3.3 Síntesis en FPGAs

Después de la simulación satisfactoria de todo el sistema y previamente a la fabricación de las máscaras, un diseño de esta complejidad exige un paso intermedio. Este paso es la síntesis con lógica programable y con PCBs de prueba para testear el sistema en una máquina de lavado. Así conseguimos asegurarnos de que la funcionalidad de la máquina se corresponde exactamente con lo que el cliente (ISTOBAL S.A.) pretende obtener.

Para realizar la síntesis se ha sacado fuera del ASICUC el 8051, utilizando una de las versiones de Intel. El resto de módulos se han sintetizado conjuntamente en 2 FPGAs de ACTEL. Se han utilizado estas FPGAs debido a la disponibilidad de las librerías para el sintetizador *Synergy* de Cadence.

4 Beneficios industriales

Entre los beneficios obtenidos por ISTOBAL S.A. debido a la modernización de sus puentes de lavado mediante la introducción de la microelectrónica podemos enumerar los siguientes:

Abaratamiento de costes en materiales: El hecho de eliminar una gran cantidad de cableado (un cable para cada E/S) y de simplificar y reducir el tamaño de la electrónica permite una reducción del orden del 20% en el coste del sistema de control de la máquina.

Abaratamiento de costes en tiempo de montaje: El coste en horas de trabajo debido al montaje de cientos de cables se reduce totalmente. De esta manera se reduce la mano de obra y se abarata el producto.

Fiabilidad: Al reducir el numero de conexiones y de cables se aumenta la fiabilidad del sistema. Igualmente se han introducido aspectos de tolerancia a fallos en el control de E/S, tratamiento inteligente de fallos por medio de programa y diagnosis del problema remotamente con módem.

Mejora de prestaciones: Al renovar la electrónica de control se han introducido nuevas prestaciones para la máquina, como la gestión remota por módem, generación de históricos, conexión a PC mediante puerto serie, inclusión de display gráfico, etc.

Otras mejoras: Se ha conseguido la fácil programabilidad de la máquina, la protección frente a copia, la inclusión de nuevas tecnologías en la empresa privada y en general la mejora del producto y aumento de la competitividad frente a la competencia.

5 Conclusiones

Se ha presentado un ejemplo de aplicación de los ASICs a la industria. Se ha descrito el sistema de puente de lavado como una serie de E/S a controlar por un sistema central. Para ello se ha desarrollado un sistema de comunicaciones maestro-esclavo que solo necesita un cable coaxial que recorre la máquina. Se muestra el protocolo y la estructura de los mensajes, que implementan varias técnicas de tolerancia a fallos.

La gran cantidad de programas y opciones diferentes que la máquina puede incorporar implica una programabilidad que se incluye con la macrocelda 8051 de TreeSoft. Así se combina la minimización de componentes a usar para abaratar costes y la versatilidad que ofrece el software.

Se ha presentado la metodología de diseño utilizada, justificando el uso de la compilación de silicio a partir de Verilog. Éste es el método mas rápido de realizar diseños digitales a partir de descripciones de alto nivel junto con la utilización de librerías de celdas y macroceldas.

Los beneficios Industriales son claros e ISTOBAL S.A. ha obtenido un gran beneficio al incorporar la tecnología ASIC en su producto.

6 Agradecimientos

Este proyecto ha sido financiado en un 80% por el comité GAME con fondos ESPRIT de la CEE. El resto ha sido financiado por ISTOBAL. El consorcio lo han formado SIDSA como empresa interfase con la *foundry* y responsable de las comunicaciones, la Universitat de València como responsable del sistema central y del software, e ISTOBAL como usuario final que ha definido las especificaciones del sistema y se ha arriesgado en esta aventura de innovación tecnológica conjunta Universidad-Empresa.

7 Referencias

[1] B. Fuchs “Verilog HDL est bien preferable a VHDL”. *Electronique*, Nº 35. Febrero 1994.

[2] D. Johannsen “Bristle blocks: A silicon compiler”, *in Proc. 16th Design Automation Conf.,* San Diego, CA, 1971, pp.310-313.

[3] D. D. Gajski, N. D. Dutt, and B. M. Pangrle, “Silicon Compilation. (Tutorial)”, *in Proc. 1986 Custom Integrated Circuits Conf.*, 1986, pp. 102-109.

[4] H. W. Tickey, “Flamel: A High level hardware compiler”, *IEEE Trans. Computer-Aided Design*, vol. 6, pp 259-269, Mar. 1987.

[5] B. M. Pangrle and D. J. Gajski, “Design tools for intelligent silicon compilation”, *IEEE Trans. Computer-Aided Design*, vol. 6, pp 1098-1112, Nov. 1987.

[6] M. R. Barbacci and D. P. Siewiorek, *The Design and Analysis of instruction set processors*. New York: McGraw-Hill, 1982.

[7] M. Kahrs "Silicon Compilation of Very High Level Languaje", *IEEE Trans. Computer-Aided Design*, vol. 11, pp 1227-1246, Oct. 1992.

[8] R. Camposano and W. Rosenstiel "Synthesizing circuits from behavioral descriptions", *IEEE Trans. Computer-Aided Design*, vol. 8, pp 171-180, Feb. 1989.

[9] B. M. Cook and N. H. White, *Computer Peripherals*, Ed. Edward Arnorld. 1995.

[10] P. A. Lee and T. Anderson, *Fault Tolerance. Principles and Practice*. Ed. Springer-Verlag Wien. 1981.

[11] SYNERGY. Family of design synthesis technology. Verilog HDL Design Guide. © Cadence Design Systems 1992.

[12] F. Daura, "El ruido en sistemas digitales" *Mundo electrónico*. Nº 169-180, 1986-1988

[13] J. Cooley. "Verilog won and VHDL lost? You be the judge!". Netscape page at: http://www.chronologic.com/misc/reviews/verilog.review.html