ENTORNO DOCENTE DE DISEÑO Y TEST DE UN MICROPROCESADOR

N. Cañellas, L.F. Marsal, J. Pallarès, J.L. Ramírez, P. Cabré, G. Bofill
Universitat Rovira i Virgili
Departament d'Enginyeria Electrònica
Escola Tècnica Superior d'Enginyeria
Autovia de Salou s/n 43006 Tarragona
Telf.: + 34 77 559625 Fax: + 34 77 559710
email: ncanyell@etse.urv.es

RESUMEN.- En este trabajo se presenta un sistema hardware-software para el diseño de un microprocesador sencillo, su implementación en FPGA y verificación de funcionamiento lógico. Se trata de facilitar a los alumnos de las asignaturas de Sistemas Digitales el paso entre el diseño de una máquina de estados finitos y el diseño de un microprocesador sencillo, de forma que el alumno pueda comprobar en el laboratorio el comportamiento interno del microprocesador, verificando la correcta temporización de los bits generados por la unidad de control, así como el adecuado flujo de datos por la unidad de proceso.

1.- INTRODUCCIÓN

En los últimos años, en las asignaturas de Sistemas Digitales correspondientes a las titulaciones de Ingeniería Técnica en Informática de Sistemas y Ingeniería Técnica Industrial especialidad Electrónica Industrial impartidas en la Escuela Técnica Superior de Ingeniería de la Universidad Rovira i Virgili, se han ido introduciendo los dispositivos programables de complejidad baja-media como parte fundamental del temario, tanto a nivel teórico, como en las prácticas docentes. Ahora se incorporan los dispositivos FPGA de Xilinx que, por sus elevadas prestaciones, són capaces de integrar sistemas digitales de considerable complejidad; aprovechándolos para poder realizar el diseño y la síntesis de un microprocesador sencillo, de forma similar a como se ha hecho en otras universidades [1], llenando el vacio que existía a nivel práctico entre la implementación de sistemas secuenciales sencillos y las prácticas con microprocesadores.

Para poder utilizar el entorno en diferentes cursos, se han realizado distintas descripciones del mismo diseño empleando herramientas diferentes: una descripción VHDL desde el programa Synopsys, una descripción esquemática desde Cadence, y una descripción a nivel lógico mediante el programa CUPL; todas ellas aplicaciones compatibles con el programa XACT de Xilinx mediante el cual se sintetiza el diseño y se implementa en uno o dos dispositivos tipo FPGA según el enfoque deseado para cada una de las prácticas.

Las distintas prácticas realizadas permiten, por ejemplo:

- Ampliar la capacidad de la unidad aritmético-lógica inicialmente propuesta.
- Modificar la unidad de control para soportar un repertorio de instrucciones más ámplio que aproveche las prestaciones de la nueva unidad aritmético-lógica.
- Diseñar la unidad de control cableada, definida a nivel esquemático o a nivel de máquina de estados finitos, o en forma microprogramada e implementada en una ROM.

2.- SOPORTE HARDWARE

Para la implementación se emplea la placa FPGA Demo board (Figura 1) suministrada por Xilinx [2] que incorpora dos dispositivos XC4003APC84 y XC3020APC68. Se han hecho accesibles, en los terminales de las FPGA, las señales internas más interesantes del microprocesador, y se han conectado a visualizadores para poder ir comprobando su evolución;

se ha utilizado un circuito de reloj externo para permitir la ejecución progresiva de los programas ejecutados por el microprocesador (tanto instrucción a instrución como ciclo de reloj a ciclo de reloj).
Aunque la sencillez del microprocesador diseñado permite su implementación completa en el dispositivo XC4003APC84, el empleo de las dos FPGA permite tener accesibles un mayor número de señales, así como el poder modificar por separado los diseños de la unidad de control y de proceso sin necesitar recompilar el diseño completo.

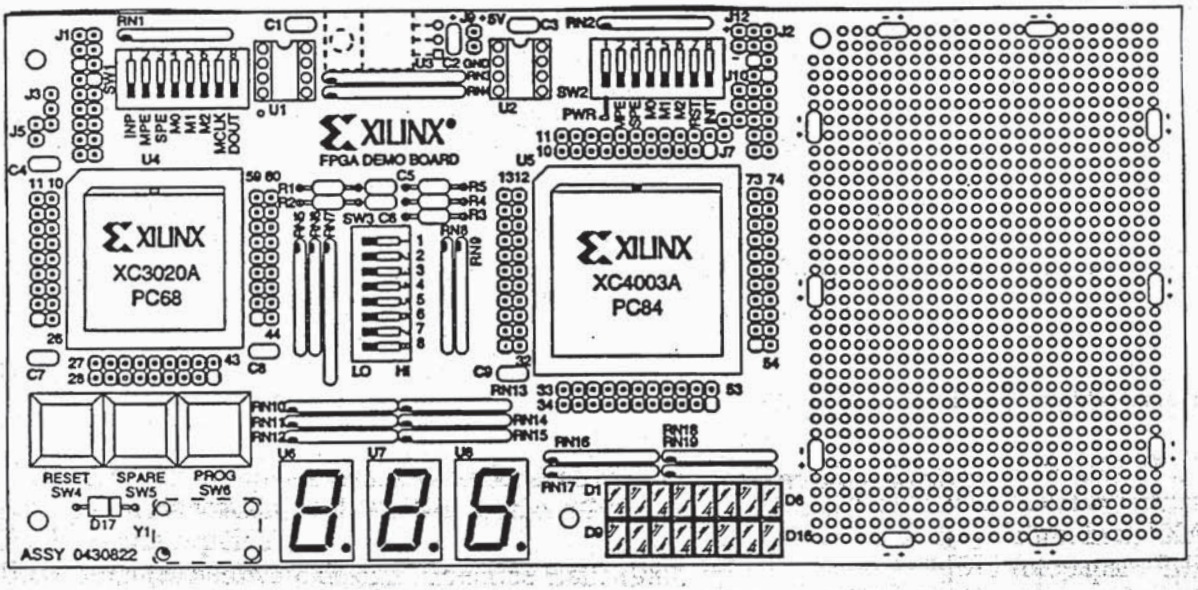

Figura 1.- Demo Board FPGA.

La grabación del diseño en la placa se realiza directamente a través del programa XCHECKER de xilinx mediante una conexión al puerto serie del PC, con lo que fácilmente se puede modificar el diseño y verificar el funcionamiento de la nueva implementación.
En la Tabla I se ven algunos de los informes de ocupación de los dispositivos programables obtenidos a través del programa XACT de XILINX, pudiéndose apreciar los márgenes de que se dispone para modificaciones o ampliaciones del diseño por parte de los alumnos.

UC Cableada en FPGA 3020APC68:	36 of 64 CLBs used
	18 of 58 I/O pins used
	0 of 6 internal IOBs used
	0 of 16 internal three-state signals used
	4 CLB flipflops used
UC Microprog. en FPGA 4003APC84:	66% utilization of I/O pins. (40 of 61)
	16% of CLB function generators. (32 of 200)
	15% utilization of CLB flip-flops. (30 of 200)
	20% utilization of bus resources. (8 of 40)
UP en FPGA 4003APC84:	80% utilization of I/O pins. (49 of 61)
	66% of CLB function generators. (131 of 200)
	18% utilization of CLB flip-flops. (35 of 200)
	20% utilization of bus resources. (8 of 40)
UC y UP en FPGA 4003APC84:	52% utilization of I/O pins. (32 of 61)
	82% of CLB function generators. (163 of 200)
	18% utilization of CLB flip-flops. (35 of 200)
	20% utilization of bus resources. (8 of 40)

Tabla I.- Estadísticas de ocupación de los dispositivos FPGA

3.- ARQUITECTURA

Se ha elegido, para su implementación en FPGA, una máquina sencilla (Figura 2) de buses de datos y direcciones de 8 bits, con un reducido juego de instrucciones (transferencia, compración, suma y salto condicional) que operan con la memória y el registro acumulador incorporado en la unidad de proceso.

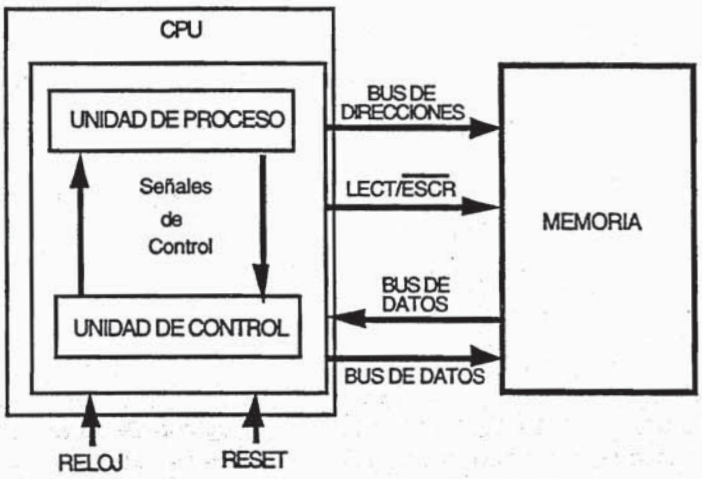

Figura 2.- Esquema de la máquina sencilla.

El procesador está diseñado para modificar fácilmente su estructura interna (Unidad de Control y/o Proceso), y la placa incorpora visualizadores para los valores de los distintos registros del procesador: contador de programa, registro de estado, acumulador, registro de instrucción y registro de direcciones, así como para las señales de control, también dispone de interruptores para modificar los valores del bus de datos, seleccionándose mediante un multiplexor el bus de datos propio o el contenido de los citados interruptores. Finalmente se dispone de pulsadores para permitir la ejecución de los programas paso a paso, pudiéndose así visualizar todas las señales después de cada ciclo de reloj o al final de cada instrucción.

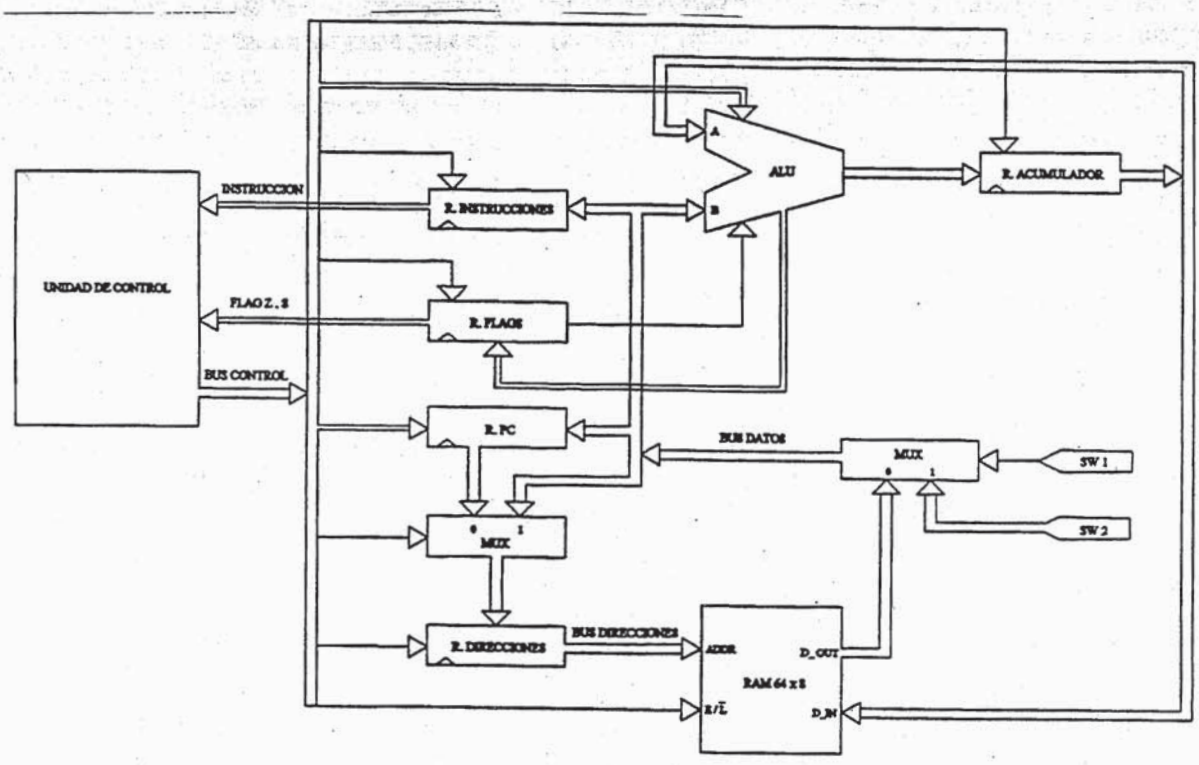

Figura 3.- Propuesta de Unidad de Proceso.

La Figura 3 presenta el esquema básico propuesto para la unidad de proceso, siendo el alumno el que diseñará las prestaciones finales deseadas modificandola convenientemente.
La unidad de control puede diseñarse desde distintas perspectivas (microprogramada o cableada), empleándose para cada una de ellas distintas herramientas de diseño VLSI. E el caso de una unidad de control microprogramada, se incorporará un esquema como el de la Figura 4 y se editará el contenido de la ROM.

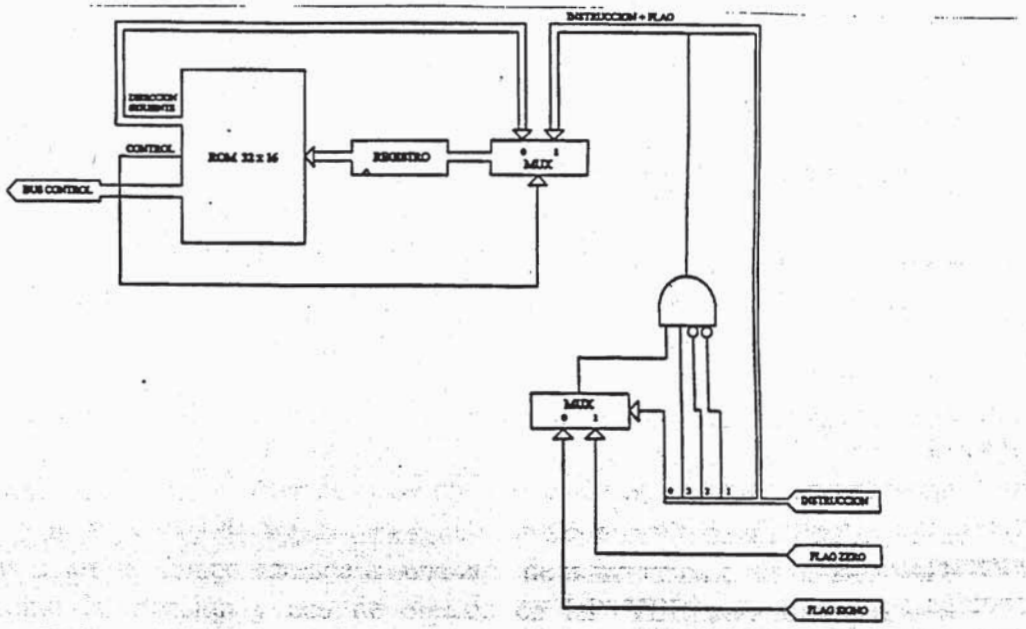

Figura 4.- Unidad de Control microprogramada.

En caso de implementarse una máquina de estados finitos puede, por ejemplo, especificarse en lenguaje VHDL mediante SYNOPSYS [4], o a través de la definición de seqüencia de estados mediante el programa CUPL [5].
Para minimizar los trabajos previos al test del diseño se ha previsto que siempre que el alumno diseña una única parte del procesador, por ejemplo la unidad de control, aprovechando un diseño previo del resto del procesador, la implementación final se realiza utilizando los dos dispositivos, uno de los cuales ya estaba programado al inicio de la practica.

4.- JUEGO DE INSTRUCCIONES

Se propone inicialmente un procesador capaz de ejecutar 11 instrucciones, pudiendo el alumno ampliar el diseño hasta 16 (4 bits para el registro de operación). Las instucciones incluyen suma aritmética, producto y comparación lógicos, transferencia de datos entre memoria y acumulador y salto condicional, tanto para direccionamiento absoluto como inmediato.
Se emplea una codificacion extendida para las instrucciones, de forma que el primer byte contiene la instrucción a ejecutar (se podría ampiar aún más el repertorio de instrucciones) y el segundo contiene la dirección absoluta o el valor inmediato a utilizar. Los códigos de operación propuestos se muestran en la tabla II.

CODIGO DE OPERACIÓN	INSTRUCCIÓN
000X	Suma aritmética
001X	And lógica
010X	Xor lógica
1000	Salto condicional al flag de signo
1001	Salto condicional al flag de cero
101X	Transferencia de datos hacia el acumulador
1100	Transferencia de datos desde el acumulador

Tabla II.- Codigos de operación propuestos.

Puesto que el entorno se emplea inicialmente para la realización de las prácticas de sistemas secuenciales, ésta es la parte con la que se relacionarán más los alumnos, modificando la unidad de control para modificar o ampliar este juego de instrucciones.

5.- CONCLUSIONES

La implementación de un microprocesador sencillo en FPGAs, permite al alumno conocer la estructura básica de un computador desde un punto de vista práctico, así como el realizar modificaciones hardware de este microprocesador de una manera sencilla, económica y didáctica. Además, la posibilidad de hacer accesibles en los terminales de los dispositivos todas las señales internas del microprocesador, permite la observación de los ciclos internos de funcionamiento, así como el flujo de datos en la unidad de proceso y los valores de las señales de control.
Finalmente debemos destacar que el diseño propuesto permite ser extrapolado a cualquier tipo de dispositivos lógicos programables, debiéndose sustituir en ese caso la placa de demostración de Xilinx por un prototipo con el conexionado adecuado y los zócalos donde el alumno pueda insertar los dispositivos lógicos programables en los que sintetize las distintas partes de su diseño.

4.- REFERENCIAS

[1] Brown, G. M., Vrana, N. "A computer architecture laboratory course using programable logic". IEEE Transactions on Education, Vol. 38, No. 2, May 1995.
[2] Xilinx Inc., "XACT reference guide".
[3] Protopapas. "Microcomputer hardware design". Prentice Hall, 1988.
[4] Synopsys Inc., "Synopsys online reference manual", 1994.
[5] Logical Devices Inc., "CUPL, Universal compiler for programmable logic", 1991