MPD: MICROPROCESADOR DIDÁCTICO EN VHDL

Javier Hernández[1], César Sanz[2], Antonio Carpeño[3] y Bernardo Merino[4]

[1]Universidad Politécnica de Madrid. jhernan@sec.upm.es
[2]Universidad Politécnica de Madrid. cesar@sec.upm.es
[3]Universidad Politécnica de Madrid. acruiz@sec.upm.es
[4]Universidad Politécnica de Madrid. merinobe@eductrade.com

RESUMEN

Esta comunicación describe el mPd (**m**icro**P**rocesador **d**idáctico) desarrollado en VHDL así como sus herramientas de ensamblado y depuración. El mPd da soporte a una asignatura sobre microprocesadores, con una carga lectiva muy baja, que no permite abordar el estudio de microprocesadores o microcontroladores comerciales y sus herramientas de desarrollo comerciales. El esfuerzo realizado en el diseño de microprocesadores didácticos en VHDL, está dando resultados muy positivos en el aprendizaje por parte de los estudiantes del funcionamiento de los sistemas digitales basados en microprocesador; estos diseños se sintetizan en la tarjeta PRINCE, sobre la cual se realizan prácticas de manejo de periféricos, interrupciones y desarrollo completo del software de sistemas basados en microprocesador.

1. INTRODUCCIÓN

Este trabajo se encuadra en la iniciativa de mejora de los recursos docentes para la enseñanza de la Electrónica Digital en el Dpto. de Sistemas Electrónicos y de Control de la EUITT-UPM. Las prácticas de laboratorio de las diferentes asignaturas se desarrollan sobre una plataforma común: la tarjeta PRINCE[1]; con el objetivo de optimizar la eficacia del aprendizaje y reutilizar el conocimiento y manejo de herramientas CAD (MAX+plus II).

La tarjeta PRINCE dispone, básicamente, de un FPGA de 30000 puertas y un variado conjunto de periféricos: teclado hexadecimal, *displays*, LCD, etc.

En el primer cuatrimestre de segundo curso de "Ingeniero Técnico de Telecomunicación especialidad Sistemas Electrónicos" del plan 2000 se imparten las asignaturas:

- "Electrónica Digital" con 1,5 créditos de teoría y 6 créditos de laboratorio
- "Sistemas Digitales I" con 3 créditos de teoría y 1,5 créditos de laboratorio

En la asignatura "Electrónica Digital" se estudian las técnicas de diseño digital y el ciclo de diseño por medio de CAD electrónico, mientras que en el laboratorio se realizan diseños lógicos que se sintetizan en el FPGA de la tarjeta PRINCE.

En "Sistemas Digitales I" se estudian los sistemas digitales basados en microprocesador. Esta asignatura plantea el reto de ajustar los contenidos de una asignatura clásica de microprocesadores a una carga lectiva mucho menor de lo común en asignaturas similares de otros planes de estudios (habitualmente: 6 créditos de teoría y 3 de laboratorio). Estas condiciones hacen inabordable el estudio de un microprocesador o microcontrolador comercial y la utilización en el laboratorio de un entorno de desarrollo comercial para dichos microprocesadores. La opción de utilizar algún microprocesador clásico de arquitectura sencilla (como Z80 ó 68000) presenta los inconvenientes de desmotivación de los estudiantes ante un microprocesador obsoleto y la falta de herramientas actuales para dichos microprocesadores.

La disponibilidad de código fuente HDL sintetizable de los microprocesadores abrió la posibilidad de utilizar microprocesadores sencillos, que el estudiante pudiera analizar y/o incluso diseñar en detalle, como medio para entenderlos mejor y para acortar la distancia entre el diseño digital con puertas y uso del microprocesador como caja negra.

La opción que consideramos más adecuada, para "Sistemas Digitales I", fue contar con un microprocesador didáctico que permitiera su estudio en profundidad y el desarrollo de prácticas de programación en lenguaje ensamblador en el laboratorio; dicho microprocesador está desarrollado en VHDL y se sintetiza con las mismas herramientas que el estudiante utiliza en el laboratorio de "Electrónica Digital": MaxPlus II y tarjeta PRINCE. Mientras que en la asignatura "Electrónica Digital" se realizan algunas de las prácticas que consisten en el diseño de subsistemas del microprocesador (la ruta de datos, la unidad aritmético-lógica, etc.); lo cual tiene la ventaja añadida para los estudiantes de estar diseñando partes del microprocesador a la par que realizan prácticas de programación de dicho microprocesador.

En una asignatura posterior: "Sistemas Digitales II" se aborda el estudio en profundidad de un microcontrolador comercial (de la familia 8051), y se realizan prácticas de programación del mismo en un entorno comercial (μVision2).

A continuación se describe el microprocesador mPd (**m**icro**P**rocesador **d**idáctico) que se utilizará en el curso 2004/05 en la asignatura "Sistemas Digitales I", así como las herramientas de ensamblado, simulación y depuración que utilizarán los estudiantes en el laboratorio de esa asignatura. Este microprocesador es una evolución de otro, denominado microSEC, que se ha utilizado en los dos cursos anteriores en dicho laboratorio y aporta mejoras significativas entre las cuales cabe citar:

- Temporización de las instrucciones más cercana a los microprocesadores comerciales
- Ampliación y mejora del repertorio de instrucciones
- Disponibilidad de un simulador
- Disponibilidad de herramientas de depuración en tarjeta

2. mPd: MICROPROCESADOR DIDACTICO

El mPd es un microprocesador de 8 bits, diseñado en VHDL, con arquitectura CISC (Complex Instruction Set Computer). La orientación didáctica del mismo hace que su comprensión y manejo sean sencillos, adecuados a los estudiantes que se acercan por primera vez al estudio de un microprocesador; sin embargo se han mantenido, en su diseño, muchas características (temporización, repertorio de instrucciones, etc.) similares a otros microprocesadores comerciales, en particular el Z80.

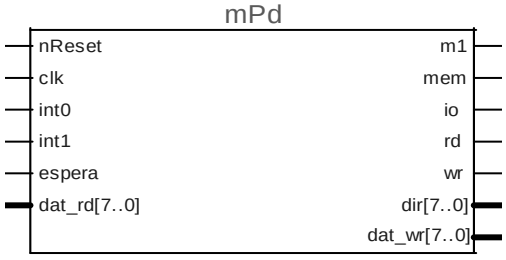

Figura 1: Conexiones del mPd

Las principales características del mPd, en cuanto al conexionado con circuitos de memoria y dispositivos de entrada/salida (ver figura 1), son:

- Bus de datos de 8 bits
- Bus de direcciones de 8 bits

- Mapa de memoria y de entrada salida independientes de 256 direcciones cada uno
- Línea M1: activa a nivel alto en los ciclos de búsqueda del código de operación
- Dos líneas de petición de interrupción: INT0 e INT1, activas a nivel alto
- Una línea de petición de estados de espera, activa a nivel alto

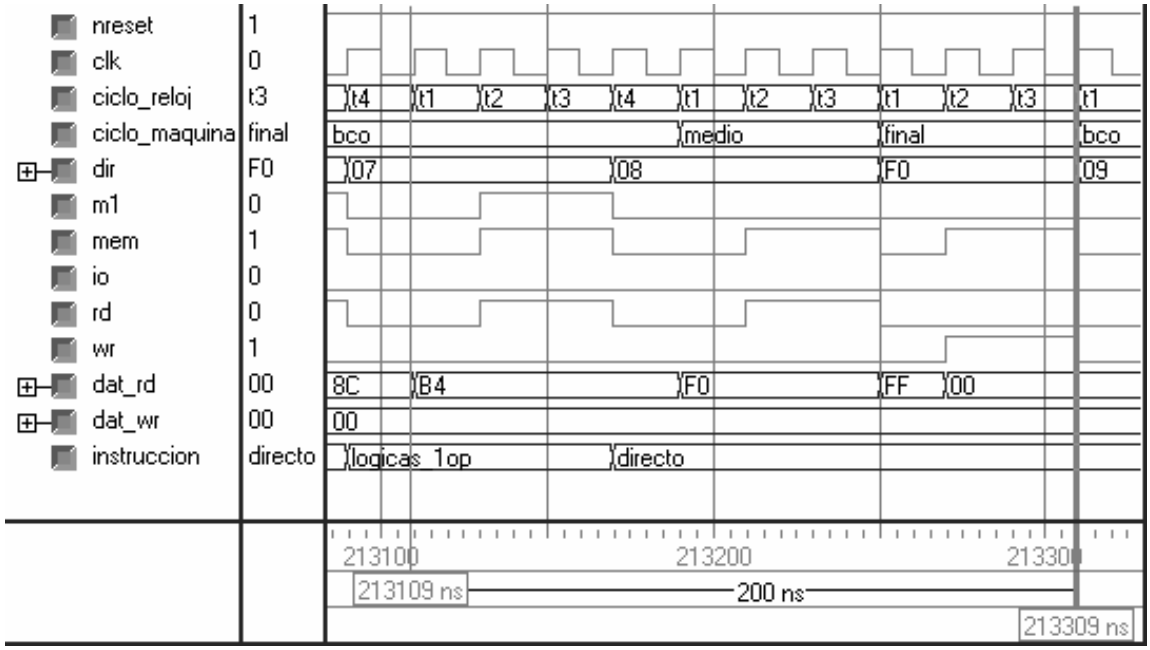

Figura 2: Temporización de la instrucción MOV A, (0Fh)

La temporización de las instrucciones se realiza en ciclos máquina; se distinguen los siguientes ciclos máquina:

- Búsqueda del código de operación: se realiza en 4 ciclos de reloj (T1, T2, T3 y T4); los tres primeros para leer el código de operación de la memoria y el cuarto para decodificar la instrucción
- Lectura de memoria: se realiza en tres ciclos de reloj (T1, T2 y T3)
- Escritura en memoria: se realiza en tres ciclos de reloj (T1, T2 y T3)
- Lectura de entrada/salida: se realiza en tres ciclos de reloj (T1, T2 y T3)
- Escritura en entrada/salida: se realiza en tres ciclos de reloj (T1, T2 y T3)

Para ilustrar la temporización del mPd, la figura 2 muestra la ejecución de la instrucción `MOV A, (F0h)` (copiar el registro A a la dirección de memoria F0h, instrucción perteneciente al grupo de instrucciones con direccionamiento directo); en dicha figura se observan claramente los ciclos máquina de búsqueda del código de operación (bco), lectura de memoria (medio) y escritura en memoria (final). La instrucción se encuentra almacenada en memoria en las direcciones 07h y 08h, y el dato almacenado en el registro A (acumulador) es 00h. El reloj utilizado es de 20MHz y la instrucción se desarrolla en 10 ciclos de reloj (200ns).

En cuanto al modelo de programación del mPd, cuenta con siete registros, todos ellos de 8 bits (ver figura 3):

A: es el registro acumulador, en el que se guardan los resultados de la ALU

B: registro de datos

C: registro de datos; se utiliza, además, en las instrucciones DJNZ (decrementar C y salto relativo si C es distinto de cero)

I: registro de datos; se utiliza, además, en las instrucciones con direccionamiento indexado e indirecto por registro

PC: contador de programa
SP: puntero de la pila
ST: registro de estado, almacena los indicadores del resultado de la última operación realizada por la ALU:
Z: indicador de que el resultado fue cero
C: indicador de que se produjo acarreo, bit de acarreo inicial
S: indicador de signo negativo
O: indicador de que se produjo desbordamiento (Overflow)
También almacena los permisos de interrupciones:
E0: habilitación de las interrupciones INT0
E1: habilitación de las interrupciones INT1

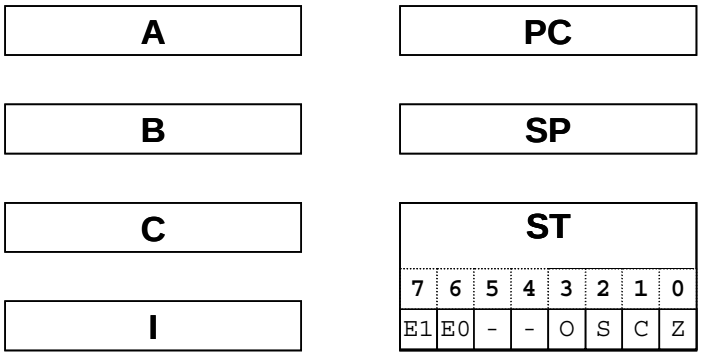

Figura 3: Modelo de programación del mPd

Todos los registros se inicializan con 00h tras el reset, salvo el puntero de la pila (SP) que se inicializa con el valor FFh. Esto impone que las primeras posiciones de la memoria estén dedicadas a almacenar el programa que se ejecutará después de un reset (memoria de programa, no volátil) y las últimas posiciones del mapa de memoria estén dedicadas a memoria de datos (RAM), sobre la cual se realizan las operaciones con la pila.

El mapa de memoria implementado en el sistema basado en mPd para el laboratorio de "Sistemas Digitales I" está formado por 224 posiciones de memoria ROM (desde la 00h a la DFh) y 32 posiciones de memoria RAM (desde la E0h a la FFh).

Las direcciones de comienzo de las rutinas de atención a las interrupciones son la 02h para INT0 y la 04h para INT1.

El repertorio de instrucciones del mPd está formado por 19 grupos de instrucciones, numeradas del 0 al 18. Los grupos de instrucciones del 1 al 18 ocupan en total 240 códigos de operación, quedando disponibles para ampliaciones del repertorio los códigos del F0h al FFh, estos códigos actualmente se interpretan como instrucciones de no operación (NOP).

El grupo de instrucciones 0 (ATENDER_INT) consiste en el desarrollo de los ciclos máquina necesarios para atender las interrupciones INT0 e INT1; por lo que no tienen ningún código de operación asociado (se desarrollan cuando hay una petición por una línea de interrupción y esa interrupción está permitida). Desde el punto de vista del controlador del microprocesador operan igual que una instrucción: se detectan la petición en un ciclo máquina M1 (en el cual se deshecha el código de operación leído y no se incrementa el contador de programa), al cual le sigue un ciclo de salvar en la pila el registro de estado y otro de salvar en la pila el contador de programa (quedando el contador de programa cargado con la dirección de comienzo de la rutina de atención). Las interrupciones INT0 son prioritarias frente a las interrupciones INT1.

En la figura 4 se muestran los grupos de instrucciones, su denominación en el código VHDL, el número de códigos ocupados por cada grupo de instrucciones (columna OCUP.), así como el valor correspondiente del código de operación (REGISTRO DE INSTRUCCIÓN); en el código de operación se indican los valores binarios de los bits que identifican la instrucción (los bits más significativos) así como el significado del resto de los bits que completan el código (por ejemplo: R/W = 0 en operaciones de lectura, = 1 en operaciones de escritura).

			REGISTRO DE INSTRUCCIÓN							
DEC	GRUPO	OCUP.	7	6	5	4	3	2	1	0
0	ATENDER_INT									
1	SALTAR_REL	64	0	0	I/C	d4	d3	d2	d1	d0
2	INDEXADO	64	0	1	R/W	A/B	d3	d2	d1	d0
3	COMPARAR	4	1	0	0	0	0	0	f1	f0
4	ROT_Y_DESP	4	1	0	0	0	0	1	c1	c0
5	INC_Y_DEC	4	1	0	0	0	1	0	A/I	c0
6	LOGICAS_1OP	4	1	0	0	0	1	1	c1	c0
7	LOGICAS_2OP	16	1	0	0	1	c1	c0	f1	f0
8	ARITMETICAS	16	1	0	1	0	c1	c0	f1	f0
9	DIRECTO	16	1	0	1	1	M/IO	R/W	r1	r0
10	COPIAR_REG	16	1	1	0	0	f1	f0	r1	r0
11	SALTAR_CON	8	1	1	0	1	0	S/R	b1	b0
12	BITS_ESTADO	8	1	1	0	1	1	S/R	b1	b0
13	LLEVAR_PILA	4	1	1	1	0	0	0	r1	r0
14	TRAER_PILA	4	1	1	1	0	0	1	r1	r0
15	INMEDIATO	4	1	1	1	0	1	0	r1	r0
16	RETORNO	2	1	1	1	0	1	1	0	c0
17	LLAMADA	1	1	1	1	0	1	1	1	0
18	SALTAR_ICON	1	1	1	1	0	1	1	1	1
19	NOP	0	1	1	1	1	X	X	X	X

Figura 4: Grupos de instrucciones del mPd

Los grupos de instrucciones del 3 al 8, ambos incluidos, corresponden a las operaciones en la ALU. En la figura 5 se muestran los nemotécnicos de las instrucciones de cada uno de estos grupos; las instrucciones de comparar CP (grupo: COMPARAR) realizan una resta, sin guardar el resultado, entre el registro A y: el registro C, el registro B, un dato inmediato (dat) o bien el byte de memoria apuntado por el registro I. Cada una de las instrucciones lógicas de 2 operandos (grupo: LOGICAS_2OP) y de las aritméticas (grupo: ARITMETICAS) admiten los mismos cuatro modos de direccionamiento detallados para la instrucción de comparar.

GRUPO	NEMOTECNICOS			
COMPARAR	CP A, C	CP A, B	CP A, dat	CP A, (I)
ROT_Y_DESP	SHR A	ROR A	SHL A	ROL A
INC_Y_DEC	DEC A	INC A	DEC I	INC I
LOGICAS_1OP	RESET A	NOT A	FLAGS A	SET A
LOGICAS_2OP	XNOR	AND	OR	XOR
ARITMETICAS	SUB	ADD	SBC	ADC

Figura 5: Instrucciones para el manejo de la ALU

3. HERRAMIENTA DE PROGRAMACIÓN EN ENSAMBLADOR DEL mPd

La programación del mPd se realiza en lenguaje ensamblador mediante una hoja de Excel que permite elegir, mediante el nemotécnico, cada una de las instrucciones que forman el programa, así como el operando de la instrucción, en caso de que la instrucción elegida lo requiera. La figura 6 muestra el aspecto de esta hoja de cálculo con la realización de un programa para sumar números de 32 bits; en dicha figura se observa la lista desplegable que da acceso a elegir una de las instrucciones del repertorio del mPd, una vez elegido el nemotécnico, la instrucción se completa con el operando (columna: DIR o DAT) en hexadecimal (hex) o en decimal (dec). Inmediatamente se obtiene el código de operación y el operando de la instrucción en código máquina (columnas: COD. MAQ.), así como la ubicación en memoria de dicha instrucción (columna: DIR HEX). De esta forma se facilita la elaboración de los programas, y se muestra claramente la relación entre el lenguaje ensamblador y el código máquina.

	A	H	I	J	L	M	N	O	P	Q
1		**SUMA DE 32 BITS**								
3										
5		DIR	COD. MAQ.			ORG.	INSTRUCCIÓN EN	DIR o DAT		
6		HEX	C.O.	OP		hex	NEMOTECNICO	hex	dec	COMENTARIOS
7		00	EB	E0		00	MOV dat, I	E0		direccion comienzo de numeros
8		02	EA	04			MOV dat, C	04		numero de bytes a sumar
9		04	D8				CLR CY			acarreo inicial a cero
10		05	44				MOV (I+d), A	04		byte segundo numero
11		06	A7				ADD A, (I)			suma con byte primer numero
12		07	68				MOV A, (I+d)	08		guardar el resultado
13		08	8B				INC I			incrementar puntero a numeros
14		09	3B				DJNZ dir	05		bucle
15		0A	87				ROL A			acarreo final al registro A
16		0B	96	01			AND A, dat	01		borrar el resto de los bits de A
17		0D	68				MOV A, (I+d)	08		guardar el acarreo final
18		0E	1F				JR dir	0E		bucle infinito
19		0F					JR dir			
20		0F					MOV (dir), A			
21		0F					MOV (dir), B			
22		0F					MOV (dir), C			
23		0F					MOV (dir), I			
24		0F					MOV (I+d), A			
25		0F					MOV (I+d), B			
							MOV A, (dir)			

Figura 6: Hoja de cálculo para la programación en ensamblador

Adicionalmente, en la hoja de cálculo se permite marcar una dirección de comienzo de la instrucción (columna: ORG. Hex), similar a las clásicas directivas de ensamblador: ORG. También se permite incluir tablas de datos escribiendo valores en las columnas DIR o DAT sin haber elegido ninguna instrucción en nemotécnico (similar a la clásica directiva DB de los ensambladores). En las instrucciones de salto se puede poner como operando (columna: DIR o DAT) la referencia a la celda de la hoja de cálculo que tiene la dirección de memoria a la que se quiere saltar (columna: DIR HEX), de manera que se mantiene dicha referencia si posteriormente se cambia el programa o se copian las instrucciones a otra parte del programa (con lo que se consigue una funcionalidad similar a las etiquetas de los ensambladores).

4. HERRAMIENTAS PARA LA DEPURACIÓN EN mPd

El fichero de Excel con el programa realizado sirve de entrada a las herramientas de depuración del mPd. Dichas herramientas se integran en una aplicación desarrollada para su ejecución bajo entorno Windows, y constan de:

- Simulador
- Depuración en la tarjeta PRINCE
- Analizador lógico

El simulador consiste en la representación del sistema basado en mPd sobre el cual se desarrollan las prácticas en el laboratorio y la visualización de la ejecución de las instrucciones. En una primera fase se puede observar y comprender el flujo de datos que se desencadena en la ejecución de los programas, visualizando el sistema como controlador, ALU, los registros registros del modelo de programación, la memoria y la entrada/salida (ver figura 7). El funcionamiento del simulador en este modo está inspirado en el programa MSX88 [2].

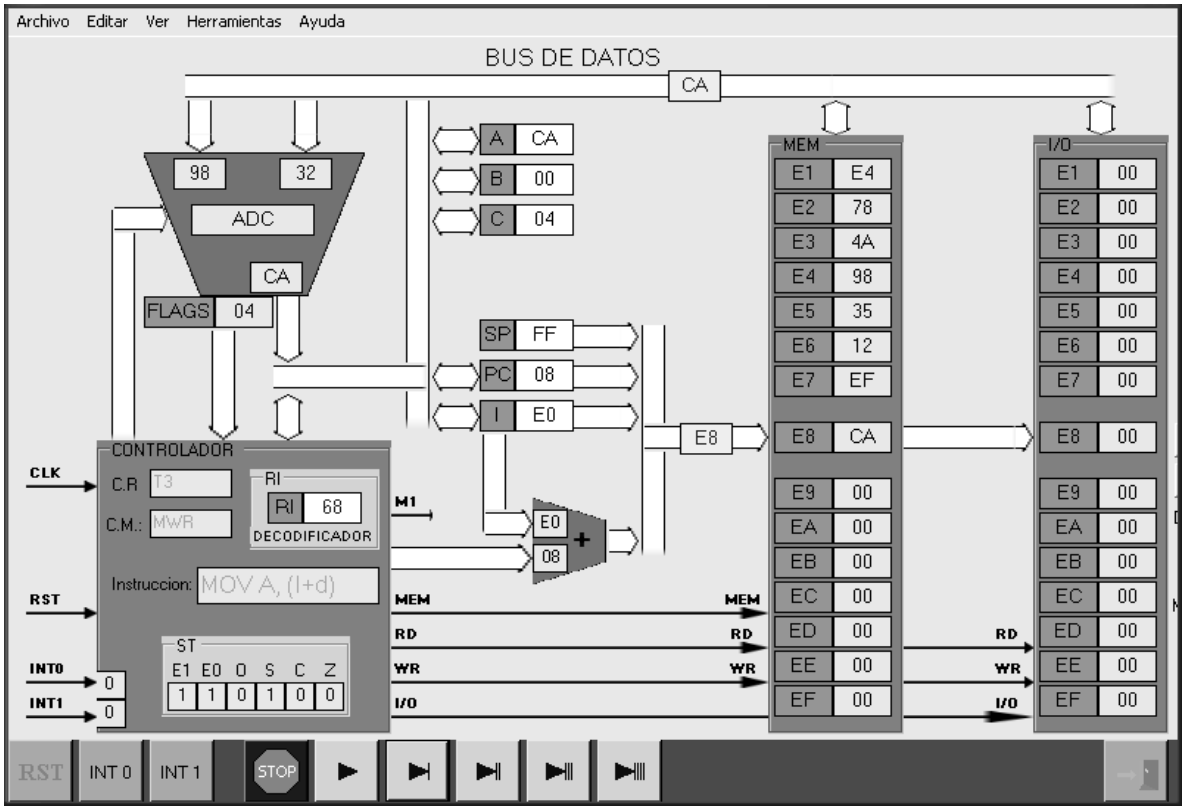

Figura 7: Simulador del mPd

En la figura 7 se muestra el aspecto del simulador durante la ejecución de la instrucción `MOV A, (I+8)` del programa de la figura 6: Suma de números de 32 bits.

En el diseño del mPd en VHDL se ha incluido la lógica necesaria para poder realizar la depuración del microprocesador en la tarjeta PRINCE desde el ordenador personal, mediante una línea serie RS-232. Concretamente se ha incluido en el diseño:

- Un módulo de comunicaciones serie asíncronas (UART)
- Un módulo DEBUG: para la depuración; incluye la lógica para realizar el reset del microprocesador, la carga de programas en memoria de código, la lectura y

escritura en registros internos, la lectura y escritura en memoria de datos y en dispositivos de entrada/salida, etc.

- Un módulo ANALIZADOR LÓGICO: con un analizador lógico sencillo en el cual se puede programar la palabra disparo y realiza la captura de los niveles lógicos en los buses del mPd en cada ciclo de reloj, rellenando con dichos valores una memoria de captura. Una vez completada la memoria se pueden enviar los datos por la línea serie al ordenador personal y visualizarlos en forma de cronogramas o tablas de estados

Desde el entorno de desarrollo se podrá configurar que las herramientas de depuración (DEBUG y ANALIZADOR LÓGICO) operen sobre el simulador o bien sobre la PRINCE, a través de uno de los puertos de comunicaciones serie del ordenador: COM1 o COM2.

La herramienta DEBUG permitirá depurar los programas en un entorno similar a otras herramientas de depuración en ensamblador para microcontroladores: MPLAB [3] (para los microcontroladores PIC) o Vision2 [4] (para la familia de microcontroladores 8051). Pudiendo resetear el mPd, poner puntos de ruptura, ejecutar paso a paso cada instrucción, visualizar y modificar los registros y la memoria, etc.

El manejo del analizador lógico será similar a los analizadores lógicos comerciales, facilitando de esta manera el conocimiento de estas herramientas, como soporte al análisis y detección de fallos en el hardware de los sistemas digitales, especialmente de los basados en microprocesador.

5. VENTAJAS DE LA UTILIZACIÓN DEL mPd

La experiencia de utilizar en el laboratorio, durante dos cursos académicos, un microprocesador diseñado en VHDL y sintetizado en la tarjeta PRINCE, ha sido muy enriquecedora, tanto para los estudiantes como para los profesores; especialmente al cursar los estudiantes simultáneamente la asignatura "Electrónica Digital", en la que aprenden las técnicas de diseño lógico sobre dispositivos lógicos programables (FPGA).

Algunas de las ventajas en el aprendizaje de los microprocesadores mediante las prácticas de laboratorio con estos microprocesadores didácticos (microSEC en cursos anteriores y mPd en el próximo) son:

- Realización de sistemas basados en microprocesador en entornos de simulación y síntesis conocidos por los estudiantes
- Visualización y análisis del funcionamiento tanto de la interfaz externa (los buses de direcciones, datos y control) como de la arquitectura interna
- Percepción del impacto de la arquitectura interna del microprocesador en la incorporación de nuevas instrucciones y modos de direccionamiento
- Mayor comprensión de la temporización de los microprocesadores: ciclos máquina, principio de búsqueda y ejecución de la instrucción, etc.
- Mejor aprendizaje del funcionamiento de un sistema programado mediante la realización de programas en lenguaje ensamblador
- Realización de prácticas de integración hardware/software
- Adaptación a diversas necesidades: inclusión de periféricos, cambio del repertorio de instrucciones, cambio de la temporización de los buses, etc.
- Estudio de un microprocesador completo, y desarrollo de prácticas de programación en ensamblador, manejando diversos periféricos reales, en una asignatura con muy poca carga lectiva (3 horas de teoría y 1 hora de laboratorio semanales)

6. BIBLIOGRAFÍA

[1] C. Sanz, F. Pescador, M. A. Freire, M. Garrido y M. C. Rodríguez, "Recursos para la enseñanza de la Electrónica Digital," *Libro de Actas V Congreso TAEE*, Las Palmas de Gran Canaria, pp. 127-140, Febrero 2002

[2] R. de Diego, "MSX88: una herramienta para la enseñanza de la estructura y funcionamiento de los ordenadores", *Libro de Actas I Congreso TAEE*, Madrid, pp. 397-409, Julio 1994

[3] Microchip, http://www.microchip.com/

[4] Keil Software, http://www.keil.com/