DISEÑO DE UN PROCESADOR SENCILLO CON CARÁCTER DOCENTE: TEORÍA Y PRÁCTICA

M. P. PARRA[1], C. BAENA[1], I. CASADO Y M. VALENCIA[1]
[1]*Instituto de Microelectrónica de Sevilla (CNM-CSIC)*
tb con Dpto de Tecnología Electrónica. Fac. de Informática. Univ. de Sevilla. 41012-Sevilla. España.

La complejidad que supone el estudio de los procesadores reales puede ser suavizada mediante el estudio de un procesador de carácter docente. Nuestro objetivo con esta comunicación es presentar una propuesta de un procesador de estas características tanto a nivel de arquitectura como de funcionalidad. Por otra parte, mostraremos una herramienta informática consistente en un emulador del procesador cuyo objetivo es facilitar la realización de ejemplos prácticos.

1. Introducción

El estudio de los procesadores constituye un objetivo básico desde los primeros cursos de los estudios de Ingeniería Informática. Debido a la complejidad de los procesadores reales es muy conveniente introducir previamente un procesador sencillo con fines pedagógicos. En esta comunicación hacemos una propuesta en esta línea. En el apartado 2 detallamos esta propuesta a nivel funcional mostrando su repertorio de instrucciones. Posteriormente, en el apartado 3 nos centramos en la arquitectura presentando la unidad de datos utilizada. En el apartado 4 cubrimos el aspecto experimental presentando muy brevemente una herramienta informática que se desarrolla en profundidad en otra comunicación titulada "Emulador de un computador sencillo".

2. El computador simple: repertorio de instrucciones.

El procesador simple que se ha diseñado trabaja con palabras de 12 bits. Se trata de una mejora al que se viene impartiendo en clase por los autores [1] y que se ha desarrollado en el Proyecto Fin de Carrera [2]. Tanto las instrucciones que ejecuta como los datos son almacenados en una única memoria RAM de 2^8x12. La unidad de datos, que se describirá en el siguiente apartado, posee una arquitectura basada en registro acumulador. En cuanto al repertorio de instrucciones se han incluido instrucciones de movimiento de datos desde la memoria al registro acumulador y viceversa, instrucciones aritméticas como suma y resta, lógicas como rotaciones del registro acumulador, instrucciones de salto tanto condicional como incondicional, instrucciones para subrutinas, y de escritura y lectura en pila. El número total de instrucciones es 23, estas son de una palabra y tienen dos posibles formatos según se trate de instrucciones con o sin dato. En el caso de las instrucciones con dato se reservan los cuatro bits más significativos para el código de operación dejando los otros ocho para referirse al dato. En el caso de las instrucciones sin dato se utilizan los doce bits como código de operación. Se han incluido cinco modos de direccionamiento: inmediato, directo, indirecto, implícito o inherente y relativo al contador de programa. El modo de direccionamiento utilizado por cada instrucción es único y está implícito en su código de operación.

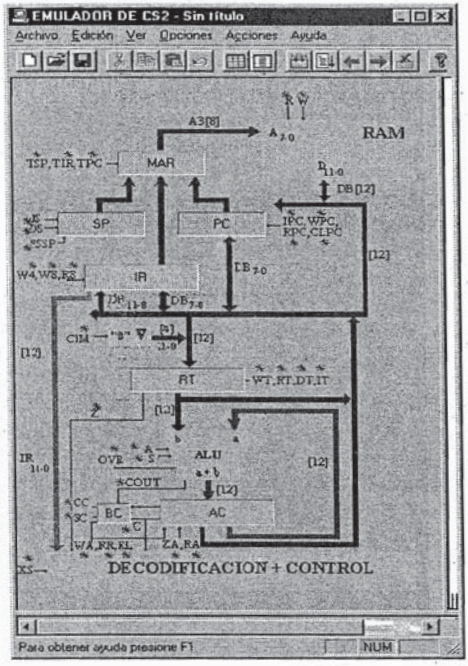
En la Tabla 1 se presenta un listado completo de las 23 instrucciones. Para cada una de ellas se da su código de operación (de 4 ó 12 bits), su sintaxis en lenguaje ensamblador, una descripción a nivel de transferencia de registros (descripción RT), y un breve comentario. Las 5 primeras instrucciones (LAIM, LDA, LDAI, STA, STAI) están dedicadas realizar transferencias de datos. LAIM permite la escritura de un dato inmediato en el acumulador. LDA y LDAI se dedican a movimientos desde memoria al acumulador con direccionamiento directo e indirecto respectivamente. Para movimientos de datos en sentido contrario se dispone de STA y STAI. Las siguientes instrucciones corresponden a operaciones aritméticas entre datos de memoria y el acumulador del sistema. En concreto son dos instrucciones para la suma: ADD y ADDI, con dos modos de direccionamiento a memoria distintos (directo e indirecto) y una para la resta, SUB, con direccionamiento directo. También se dispone de dos instrucciones para incrementar (INC) o decrementar (DEC) un dato en memoria. En cuanto a las instrucciones lógicas se realizan sobre el acumulador y/o sobre un biestable C que almacena el valor del acarreo de salida de la ALU del sistema. ROR y ROL son operaciones de rotación a derecha e izquierda, respectivamente, sobre el acumulador y el biestable C. El valor almacenado en este último puede modificarse con las instrucciones CLC (puesta a cero) y SEC (puesta a uno). Las instrucciones de salto son de tipo incondicional (JMP) o condicional (BCS, BZPC, DBZ). BCS condiciona el salto a que el bit de acarreo almacenado en C sea 1. Con BZPC se dispone de un salto relativo al contador de programa condicionado a que el acumulador (estrictamente, el registro RT) esté a cero. DBZ decrementa un dato de memoria, si el resultado es cero se salta la siguiente instrucción. Para salto y retorno de subrutina se dispone de las instrucciones JSR y RTS, y para manejo directo de la pila de las instrucciones PUSH (escritura) y POP (lectura). Finalmente, se dispone de una instrucción de fin de programa: STOP.

Código de operación	Sintaxis	Descripción RT	Comentario
0000	LAIM #AA	AC ← AA	Mueve dato inmediato a AC
0001	LDA $AA	AC ← M($AA)	Mueve dato de memoria a AC con direccionamiento directo
0110	LDAI $AA	AC ← M(M($AA))	Mueve dato de memoria a AC con direc. indirecto
0010	STA $AA	M($AA) ← AC	Mueve dato de AC a memoria con direccionamiento directo
0111	STAI $AA	M(M($AA)) ← AC	Mueve dato de AC a memoria con direc. indirecto
0011	ADD $AA	AC ← AC + M($AA)	Suma con direc. directo
0101	ADDI $AA	AC ← AC + M(M($AA))	Suma con direc. indirecto
0100	SUB $AA	AC ← AC - M($AA)	Resta con direc. directo
1011	INC $AA	M($AA) ← M($AA) + 1	Incrementa dato en memoria.
1100	DEC $AA	M($AA) ← M($AA) - 1	Decrementa dato en memoria
111100000100	ROL	AC ← SHL (AC,C), C ← AC_{11}	Rotación lógica a izquierda
111100000101	ROR	AC ← SHR (AC,C), C ← AC_0	Rotación lógica a derecha
111100000011	CLC	C ← 0	Puesta a cero de carry
111100000010	SEC	C ← 1	Puesta a uno de carry

1000	JMP $AA	GOTO $AA	Salto incondicional
1001	BCS $AA	C: GOTO $AA	Salto condicional
1101	BZPC #AA	Z: PC ← PC + AA	Salto relativo a PC
1010	DBZ $AA	M($AA) ← M($AA) – 1; Z: GOTO N+2	Decrementa dato en memoria y salta si cero
1110	JSR $AA	SP ← SP –1; M(SP) ← PC; GOTO $AA	Salto a subrutina
111100000000	RTS	PC ← M(SP); SP ← SP + 1	Retorno de subrutina
111100000111	PUSH	SP ← SP – 1; M(SP) ← AC	Mueve AC a pila
111100000110	POP	AC ← M(SP); SP ← SP + 1	Mueve dato de la pila a AC
111100000001	STOP	NOP	Fin de programa

Tabla 1: *Repertorio de instrucciones del procesador simple.*

3. El computador simple: unidad de datos



La unidad de datos, que se muestra en la Figura 1, está compuesta por una unidad aritmético lógica (ALU) de 12 bits y una serie de registros específicos. Los principales registros son: un contador de programa, PC, que proporciona la dirección de memoria donde se encuentra la próxima instrucción a realizar; un puntero de pila, SP; un registro de instrucciones, IR, que recibe las instrucciones procedentes de la memoria a través del bus interno del sistema; un registro de direcciones de memoria, MAR, que permite conectar los registros SP, PC e IR con el bus de direcciones de memoria y un acumulador (AC) conectado a la ALU para guardar los datos parciales y finales de las operaciones. El acarreo de salida procedente de la ALU se almacena en el biestable BC. En cuanto al sistema de interconexionado se utilizan buses de 12 bits de tipo bidireccional y unidireccional, tanto dedicados como compartidos.

Figura 1: *Unidad de datos del procesador*

4. Un emulador del computador simple para uso docente en laboratorio

Se ha diseñado una herramienta informática para la emulación del procesador simple. La Figura 1 muestra parte de su ventana principal. Con el emulador se pretende afianzar los nuevos conceptos y dar un enfoque práctico a la materia. La aplicación recibe como entrada un programa escrito a nivel de instrucciones que es compilado y ejecutado. Para ello se ha creado un ensamblador que facilita la escritura del programa. Este permite definir variables simples y *arrays*, insertar comentarios, usar distintos formatos numéricos, definir y usar etiquetas, etc. Por otra parte, el emulador proporciona una interfaz gráfica que permite observar la arquitectura del procesador, permitiendo de forma sencilla tener acceso a todas aquellas informaciones relevantes sobre su estado (registros, señales de control y de estado, memoria RAM ...). La herramienta facilita la ejecución de los programas de varias formas. La primera es la ejecución completa del programa, que posibilita comprobar el algoritmo planteado de forma rápida. Una segunda forma es la ejecución instrucción a instrucción. Finalmente, la ejecución por ciclos, en la que se accede al estado del procesador en cada ciclo de ejecución. Cuando la ejecución se realiza paso a paso se visualiza en pantalla los contenidos de todos y cada uno de los registros de la arquitectura así como las señales de control activadas. De esta forma, se puede seguir el flujo de datos en cada microinstrucción. En cada paso la herramienta puede mostrar la instrucción que se está ejecutando tanto en formato fuente como en hexadecimal y el contenido completo de la RAM.

Para cubrir el aspecto experimental, en el estudio de este procesador sencillo se proponen una serie de algoritmos sencillos que el alumno desarrolla a nivel de instrucciones y sirven, por un lado, para adentrarse en el manejo de la propia herramienta y por otro, al ser programas simples, para comprender el comportamiento temporal del procesador y de la memoria RAM instrucción a instrucción y ciclo a ciclo. A continuación, se propone un algoritmo algo más complejo, que combine subrutinas y genere un procesado de datos de mayor envergadura. Este programa será compilado y ejecutado mediante el emulador para lo cual, la ejecución de una sola pasada es la que gana interés.

5. Conclusiones

Se ha presentado un procesador simple con fines pedagógicos. Se ha detallado a nivel funcional (repertorio de instrucciones) y a nivel arquitectural (unidad de datos). También se ha presentado una herramienta informática para la emulación del procesador que permite la realización de prácticas con lo que se cubren los aspectos prácticos de esta materia.

Referencias

[1] C. Baena, J.I. Escudero, I. Gómez y M. Valencia. *Introducción a los Sistemas Digitales*. Dpto. de Tecnología Electrónica. Universidad de Sevilla (1997).

[2] I. Casado. *Emulador del computador simple 2*. Proyecto Fin de Carrera (1998).