

METODOLOGÍA BASADA EN LA UTILIZACIÓN DE UN “COMPUTADOR BÁSICO” COMO REFERENCIA EN LA COORDINACIÓN TEMÁTICA EN LA ASIGNATURA DE ESTRUCTURA DE COMPUTADORES.

Jesús García-Consuegra Bleda(*). Antonio Martínez Núñez (*) y Francisco José Quiles Flor (*)

(*) Arrea de Arquitectura de Computadores
Dpto. de Informática
Escuela Universitaria Politécnica de Albacete
Universidad de Castilla-La Mancha
Campus Universitario s/n. 02076
Albacete.

Resumen. - En este trabajo se presenta una metodología para la docencia de la asignatura "Estructura de Computadores" correspondiente a los estudios de Ingeniería Técnica de Informática de Gestión y Sistemas.

Dicha Metodología está fundamentada en la referencia continua a un "Computador Básico", presentado al comienzo de la asignatura, basado en la arquitectura de Von Neumann.

1. INTRODUCCIÓN.

La experiencia acumulada, tras años de docencia en la asignatura, ha revelado que una metodología inapropiada puede provocar en el alumnado una concepción parcialista de la misma, es decir, entender las diversas unidades temáticas como partes independientes de un todo.

Esto provoca una degradación del objetivo principal de la asignatura: el conocimiento del funcionamiento global del Computador y el análisis y estudio de la interacción entre las diversas Unidades Funcionales que lo constituyen.

El uso de la metodología propuesta ayuda al alumno a entender, a grandes rasgos, desde el principio, el funcionamiento y la interrelación de cada uno de los bloques temáticos que se imparten durante el curso.

La aplicación de la metodología presentada se refuerza con la realización de un ejercicio sobre dicho "Computador básico", a la finalización del primer capítulo ("Introducción a la Estructura de Computadores"), obligando al alumno a su estudio y comprensión desde el comienzo de la asignatura.

2.- ANTECEDENTES.

Previo al empleo de esta metodología, antes de abordar un nuevo capítulo, se realizaba una introducción en la que se presentaban los conceptos a tratar y su interrelación con las otras Unidades Temáticas.

Aunque este proceso es correcto, su utilidad pierde eficacia cuando el alumnado se enfrenta al estudio y comprensión de cada capítulo, provocándole una concepción parcialista de la asignatura.

3.- OBJETIVOS QUE SE PERSIGUEN.

Se pretende mostrar el funcionamiento **global** de un computador basado en la arquitectura de Von Neumann, examinando la **interacción** entre sus Unidades Funcionales. Durante la presentación de la Máquina Básica se **introducen los siguientes conceptos**:

Introducción a la memoria: capacidad, palabra de memoria, ancho de palabra, tipo de memoria y empleo de la memoria.

Funcionamiento de la memoria

Concepto de direccionamiento

Bus del Sistema (datos, direcciones y control).

Señales del bus de control (señales estado y de control).

Diferenciación lógica entre datos e instrucciones almacenados en memoria.

Unidades de E/S

Funcionamiento ALU y su relación con el registro de estado (F).

CPU

Relación ancho del bus de direcciones-PC-MAR-posiciones de memoria

Relación ancho de palabra de un computador-bus de datos-MDR- registros internos-Registro de instrucción-palabra de memoria

Bus interno a la CPU

Juego de instrucciones

Ensamblador

Formato de instrucción

Codificación de las instrucciones

Correspondencia ensamblador - código máquina

Código Máquina

Relación UC-juego de instrucciones

Función de un programa traductor

Programación en código máquina

Programa cargador

Fases de la instrucción (captura, decodificación y ejecución).

Relación UC-fases de ejecución de la instrucción

Secuenciamiento de las instrucciones

4.- DESCRIPCIÓN DE LA METODOLOGÍA BASADA EN LA UTILIZACIÓN DEL COMPUTADOR BÁSICO. CONCEPTOS RELACIONADOS

Esta metodología se puede descomponer en varias fases: 1) descripción de los componentes más significativos de cada Unidad Funcional de la arquitectura de Von Neumann; 2) definición de un

juego de instrucciones para la realización de un programa; y, 3) funcionamiento del Computador.

4.1.- Descripción de los componentes más significativos de cada Unidad Funcional de la arquitectura de Von Neumann.

A partir de la definición de la arquitectura de Von Neumann, se procede a describir, con cierta profundidad, los componentes más significativos de cada Unidad Funcional. El orden y el contenido con el que se aborda esta fase es la siguiente:

- El primer paso consiste en introducir la **memoria principal** (MP): su capacidad, palabra de memoria, ancho de palabra, tipo de memoria y empleo de la memoria. Así, la memoria principal se explica como una organización en tabla (filas y columnas) de biestables, capaces de almacenar un bit. Al número de columnas se le denomina **ancho de palabra de memoria**, y a cada fila, **posición o palabra de memoria**. Se define como **capacidad de memoria** al producto del ancho de palabra por el número de filas. La selección o direccionamiento de cada una de las palabras se realiza mediante el concurso de un *Decodificador*.

A las Unidades Funcionales de la Arquitectura de Von Neumann hay que añadir los elementos (**buses**) de comunicación. Estos buses se encargan de la transferencia de información entre las Unidades Funcionales. Dependiendo del tipo de información los buses se pueden clasificar en: **bus de direcciones**, transmite la dirección de la posición de memoria con la que se desea trabajar; **bus de datos**, por él viajan los datos y las instrucciones; y, el **bus de control**, el cual aglutina todas las señales de control y estado del sistema. Al conjunto formado por los tres tipos de buses se denomina **bus del sistema**.

En nuestro caso, el número de entradas al decodificador (suficientes para direccionar la totalidad de la memoria) va a determinar, a su vez, el ancho (expresado en bits) del *bus de direcciones*.

Mediante dos señales, *Leer* y *Escribir*, se le indica a la MP si debe recoger o volcar, el contenido de la posición seleccionada por el decodificador, del o al exterior a través del *Bus de Datos*. La MP indicará la finalización de la lectura o escritura mediante la señal *Listo*.

- En un segundo paso, se presenta la **Unidad Funcional de Entrada** como un periférico (por ejemplo, un teclado) que nos capacita para introducir los datos y nuestros programas. Cada dato o instrucción se vuelca en el Bus de Datos como respuesta a la señal *Entrar*.

La **Unidad Funcional de Salida** (pantalla) tendrá un comportamiento similar, aunque esta vez, se recogerá el dato del Bus de datos y lo mostrará al usuario, cada vez que reciba la señal *Salir*.

- El tercer paso se dedica a la Unidad Aritmetico-lógica (ALU) y a los registros específicos asociados: Acumulador (A) y Registro de estado (F). Para nuestro computador, la ALU realizará la suma de dos números contenidos en el Acumulador y en el registro MDR (explicado con posterioridad) volcando el resultado en el propio Acumulador, cada vez que ésta reciba la señal Sumar. Reflejando, en el registro F, alguna de las consecuencias implícitas en dicha operación: resultado igual a cero, positivo/negativo, acarreo, etc..
- En un cuarto paso, se introduce a la Unidad de Control (UC) como la unidad encargada de

controlar y secuenciar los pasos necesarios para la correcta ejecución de los programas previamente almacenados en la MP. Para lo cual, se apoya en una serie de registros específicos como: el Contador de Programa (PC) encargado de apuntar a la posición de memoria que almacena la siguiente instrucción a ejecutar; Registro de Dirección de Memoria (MAR), registro con la dirección de memoria, codificada en binario, con la que deseamos trabajar, conectado al decodificador de la MP mediante el bus de direcciones; el Registro de Datos de Memoria (MDR), encargado de recoger, o volcar, los datos o instrucciones procedentes, o con destino, a la posición de memoria especificada por el MAR o de las Unidades de E/S; Registro de Instrucción (IR), almacena únicamente la instrucción en ejecución para su interpretación por la UC; y, por último, una batería de registros auxiliares (Ri), usados como almacenamiento temporal de datos y direcciones.

La UC, dependiendo de las señales de estado (Listo) del resto de las unidades, del código de operación de la instrucción almacenada en el IR y del registro F, envía señales de control a las unidades y circuitos para establecer el secuenciamiento necesario, para ejecutar dicha instrucción, realizando las operaciones que esto implica (microoperaciones).

El Computador Básico (fig. 1) se obtiene como consecuencia de la incorporación de los componentes, a cada Unidad funcional, a medida que se han ido presentado.

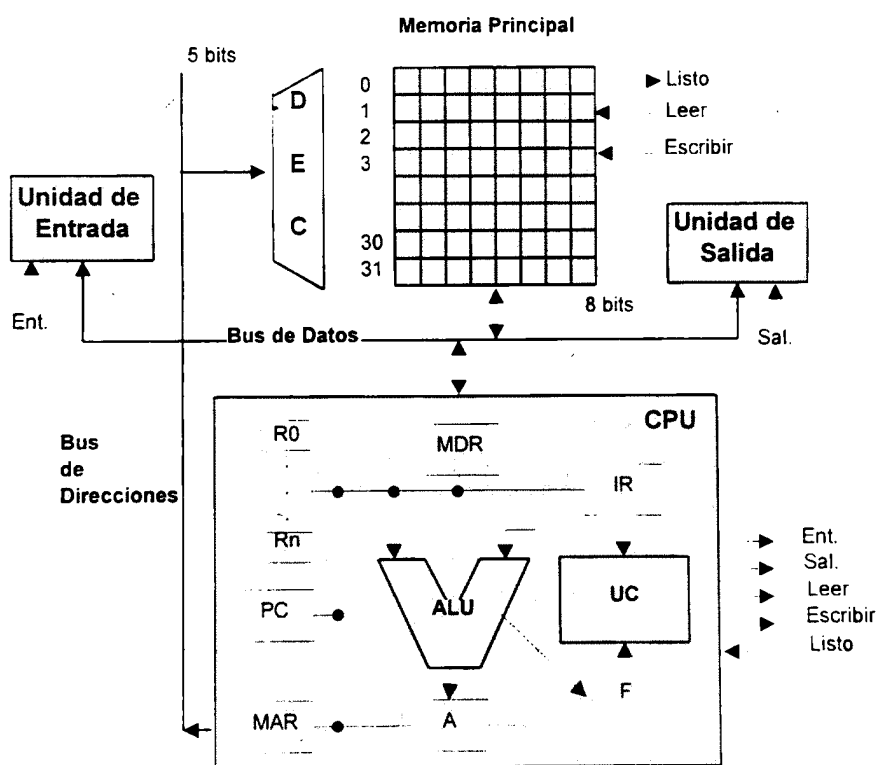


Fig. 1: Esquema del Computador Básico.

4.2.- Definición de un juego de instrucciones para la realización de un programa.

En esta fase, como ejemplo del funcionamiento de nuestro Computador, se propone la realización de un programa (Tabla 2), basado en una modificación del propuesto en [1], que permita visualizar el resultado, por la Unidad de Salida, de sumar dos números, introducidos por la Unidad de Entrada. Para ello, se diseñan el mínimo número de instrucciones necesarias, a las que llamaremos **Juego de Instrucciones**, y cuya descripción es:

- ENT: entrada de información al Acumulador desde la Unidad de Entrada. *ENT*
- SAL: salida de información desde el Acumulador a la Unidad de Salida. *SAL*
- CAR: carga al Acumulador el contenido de la posición de memoria especificada. *CAR A,M(m)*.
- MEM: memoriza el contenido del Acumulador en la posición de memoria especificada en la instrucción. *MEM M(m),A*
- SUM: suma al contenido del Acumulador el dato almacenado en la posición de memoria especificada en la instrucción. *SUM A,M(m)*

Como el computador funciona con una lógica binaria, se hace necesario la codificación del **juego de instrucciones** presentando al **lenguaje máquina** del computador, formado por unos y ceros. Como pueden apreciar los alumnos, es más difícil trabajar en este lenguaje máquina. Esta dificultad ha provocado el desarrollo de otros lenguajes, como el anterior (**lenguaje ensamblador**), más fáciles de manejar por el programador. Siendo el lenguaje ensamblador el más próximo al lenguaje máquina.

En este paso introducimos los conceptos de **Formato de la instrucción**, **correspondencia ensamblador-código máquina**, **codificación de las instrucciones**, **concepto de direccionamiento**, **relación Unidad de Control-juego de instrucciones-formato de instrucción**, y **programa traductor ensamblador-código máquina**.

Así, para la codificación de las instrucciones propuestas, se definen dos elementos (**campos**) a codificar: el identificador (código) de la instrucción y la dirección de memoria con la que se desea trabajar. Para la **codificación de las 5 instrucciones** (tabla 1) se precisan 3 bits, y otros 5 más para codificar la dirección completa de posición de memoria, es decir, el ancho de bus de dirección de la máquina (ver fig. 1). Con lo que nos quedan instrucciones con un tamaño de 8 bits, que pueden almacenarse en cada fila o posición de memoria principal.

ENT	
SAL	
CAR	A, M(m)
MEM	M(m),A
SUM	A, M(m)
Mnemónico	Dirección
Instrucciones Ensamblador	

Codificación

←→

8 Bits	
3 Bits	5 Bits
000	XXXXX
001	XXXXX
010	dddd
011	dddd
100	dddd
Campo Código de operación	Campo de direc. de los operandos
Instrucciones máquina	

Tabla 1. Juego de instrucciones ensamblador y sus instrucciones máquina equivalentes. Donde por "d" se entiende cada uno de los bits que forma la dirección de la posición de memoria con la que se desea trabajar. Y, por XXXXX se debe entender que no usa el campo de dirección, por lo que se codificará como 00000.

ENT	
MEM	16.A
ENT	

8 Bits	
3 Bits	5 Bits
000	00000
011	10000
000	00000

MEM	17.A		011	10001
CAR	A. 16		010	10000
SUM	A. 17		100	10001
SAL			001	00000
Mnemónico	Dirección (decimal)	Codificación ↔	Campo Código de operación	Campo de direc. de los operandos (binario)
Instrucciones Ensamblador			Instrucciones máquina	

Tabla 2. Programa para sumar dos números, introducidos por teclado y almacenados en las posiciones 16 y 17 de memoria. El resultado se saca por pantalla.

El programa de la tabla 2 se carga en memoria a partir de la posición 0 (ver tabla 3). Este proceso que se realiza manualmente, se encarga un programa del sistema operativo denominado *cargador*. El cargador introduce el programa en posiciones consecutivas de memoria, a partir de una dada.

Dir.	Contenido de la Memoria								Tipo de Contenido
0	0	0	0	0	0	0	0	0	Instrucciones
1	0	1	1	1	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	1	1	1	0	0	0	1	
4	0	1	0	1	0	0	0	0	
5	1	0	0	1	0	0	0	1	
6	0	0	1	0	0	0	0	0	
...									...
16	Sumando 1								Datos
17	Sumando 2								
...									...
31									
32									

Tabla 3. Memoria Principal con el programa cargado.

4.3.- Funcionamiento del Computador. Ejecución de un programa.

En esta tercera, y última, fase se reproducen las acciones tomadas por la UC para interpretar cada una de las instrucciones: llevar el contenido de la posición de memoria apuntada por el PC al registro IR (**captar**); interpretar (**decodificar**) el contenido del IR; y, tomar las acciones para la ejecución (**ejecutar**) de la instrucción teniendo presente las señales de estado y el contenido del registro F. Con ello se pretende que el alumno siga el proceso de una forma visual, entendiendo la interrelación de los componentes del Computador Básico en cada una de las **fases de ejecución de una instrucción**.

Una vez cargado el programa (manualmente o mediante el cargador), se le indica a la UC que pase a su ejecución. Para lo cual se inicializa el contenido del PC con la dirección de comienzo del programa, en nuestro caso, en la dirección 00000b. A partir de este instante, la UC repetirá los pasos siguientes hasta finalizar la ejecución del programa:

- Llevar el contenido del PC a MAR. Enviar la señal de lectura a memoria. Recoger el dato del bus de datos en el MDR. Volcar el contenido de MDR en IR. Incrementar PC en 1.

- b) Interpretar el contenido del IR
- c) Tomar las acciones oportunas para la ejecución de cada instrucción.

Estos pasos se realizan en clase para cada una de las instrucciones. repitiendo las mismas acciones que debería tomar la UC.

5.- CONCLUSIONES

El uso de una u otra metodología a la hora de impartir clase en asignaturas que tratan de partes de un mismo tema como es el caso de "Estructura de Computadores". condiciona en gran manera la comprensión por parte del alumno de cada bloque temático dentro del contexto de lo que es un computador.

El alumno estudia los diversos conceptos separados en el tiempo varias semanas. esto puede provocar la incapacidad de comprender e interrelacionar los diversos bloques temáticos como partes de un todo (el computador).

Por eso es eficaz. desde el principio. definir el entorno dentro del cual nos vamos a mover durante toda la asignatura. Al emplear una metodología de enseñanza que comience estudiando el concepto global de lo que es un ordenador y sus principios básicos de funcionamiento. facilitamos que al particularizarlo en un ejemplo concreto, bien mediante la "pizarra" o en cualquier otra herramienta didáctica. se sedimente en el alumno una gran cantidad de conceptos nuevos. básicos en el estudio de la carrera.

6.- BIBLIOGRAFÍA

- [1] Prieto, E. A., Lloris, R.A. y Torres, C. J.C. (1989): "Introducción a la Informática", edit. McGraw-Hill.
- [2] Prieto, E. A., Lloris, R.A. y Torres, C. J.C. (1995): "Introducción a la Informática", segunda edición, edit. McGraw-Hill.
- [3] Martínez, N. A. (1996): "Didáctica en las Enseñanzas Experimentales", ISBN: 84-8499-812-6
- [4] De Miguel, A. (1992): "Fundamentos de los Computadores", edit. Paraninfo.
- [5] Tanenbaum, A. S. (1992): "Organización de Computadoras. Un Enfoque Estructurado", 3ª edición, edit. Prentice-Hall.
- [6] Angulo, J.M. (1996): "Estructura de Computadores". edit. Paraninfo.
- [7] Patterson, D.A. y Hennessy, J.L. (1994): "Organización y Diseño de Computadores", edit. McGraw-Hill.