ORGANIZACIÓN DE UN COMPUTADOR ESPECÍFICO PARA BASES DE DATOS (CBD).

D. PEÑALOSA, C. BAENA, M.P. PARRA y M. VALENCIA
Dpto. de Tecnología Electrónica. E.T.S. Ing. Informática (Ed. Rojo), Univ. Sevilla Avdª. Reina Mercedes s/n Sevilla. . 41012 España.

Se presenta un computador diseñado para optimizar las aplicaciones de base de datos. Para ello se han generado un conjunto de instrucciones que realizan las operaciones características de estas aplicaciones, un direccionamiento de la memoria para organizar mejor la información almacenada y unas unidades funcionales específicas para operar con los datos.

1. Introducción

El manejo de bases de datos requiere procesar y administrar un volumen considerable de información. Para su manejo computacional existen lenguajes específicos, tales como SQL [1], que contienen instrucciones dedicadas a manipular ese tipo de información, lo que requiere muchos accesos a memoria y operaciones específicas como mostrar, borrar, actualizar o comparar datos. Además, la propia estructura de los datos en las bases de datos [2] difiere de la habitual en otros tipos de procesamiento. Así, los computadores estándar no resultan eficientes ya que ni su arquitectura ni su organización [3] optimizan la ejecución de esas tareas.

El enorme interés que suscita el manejo de bases de datos nos ha llevado a explorar el diseño de un computador específico para bases de datos (CBD). El computador CBD trata de optimizar las correspondientes operaciones organizando los datos de la base de forma adecuada, usando instrucciones especiales y diseñando el procesador con unidades especiales.

En este trabajo presentamos la arquitectura y organización de este computador, es decir, el tipo y funcionalidad de sus instrucciones y el diseño de su unidad de procesado y control a nivel RT [4]. En otro trabajo que presentamos en TAEE'2002 (*Diseño y simulación de CBD con ALLIANCE*), presentamos el entorno de diseño que hemos usado para su realización.

En el siguiente apartado describimos la arquitectura de CBD (instrucciones y registros a nivel de usuario) mientras que en el apartado 3 se presenta su organización haciendo mención de las unidades funcionales que hacen posible la ejecución de estas instrucciones. Por último, extraemos algunas conclusiones.

2. Arquitectura de CBD

Las bases de datos son unas aplicaciones donde, por un lado, se encuentra información

almacenada en unas estructuras llamadas “tablas” y, por otro, de programas que consultan, modifican, insertan y borran información en estas estructuras. Como se aprecia en la figura 1, cada tabla se organiza en filas y columnas. En el cruce de estas filas y columnas se encuentra una unidad de información llamada “hecho” [2]. En estas tablas, todos los “hechos” que conforman una fila tienen una relación semántica sobre el tipo de información que se desea almacenar.

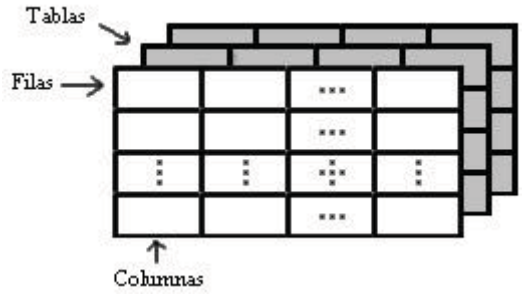

Figura 1: Estructura de almacenamiento de datos.

Nuestro computador va a poseer dos subconjuntos de instrucciones, uno con funcionalidad análoga al de cualquier procesador de propósito general (sumas, restas, transferencias, bifurcaciones, etc.) y otro subconjunto con instrucciones que realicen las operaciones fundamentales de los programas de base de datos, esto es, consultar, modificar, insertar y borrar información en tablas.

El computador CBD ha sido concebido para manejar una *memoria de datos* específica para la base de datos, separada de la memoria estándar que aquí denominamos *memoria de código*. En la memoria de código estará ubicado el programa que maneja la información de la base de datos, y está estructurada en cinco partes: una para controlar un sistema de memoria virtual; una segunda con rutinas de inicialización de la CPU; una tercera con rutinas de servicio de las interrupciones no enmascarables; una cuarta con el código de las aplicaciones y, por último, una pila para estas aplicaciones.

En la memoria de datos, por su parte, estará almacenada la información de la propia base de datos (no de los datos requeridos para ejecutar los programas). Esta memoria presenta una estructura análoga a la que se indica en la figura 1, es decir, está formada por tablas con columnas y filas. Con esto los programas que manipulan las bases de datos tendrán mejor accesibilidad a los datos, pero se requerirá de un direccionamiento especial; pues en vez de localizar una posición de esta memoria con una única dirección, ahora necesitamos tres direcciones: una que nos indique la tabla; otra que nos indique la columna; y una tercera para las filas. Además se reservarán en cada tabla dos posiciones de memoria: una con el número de columnas y otra con el número de filas de la tabla en cuestión. Para cada columna, se tendrá reservada una posición con el número máximo de palabras de la máquina que se requieren para almacenar un “hecho”.

Consideremos ahora los registros internos de propósito general en los que se almacenan los operandos de las instrucciones. Estos registros se encuentran en una unidad que recibe el nombre de “fichero de registros”, que consta de una memoria multipuerta de 32 bits de

anchura con 16 registros y 4 *constantes*, pudiendo ofrecer el contenido de dos de estos al mismo tiempo. De estos registros cabe destacar aquellos encargados de direccionar a la memoria de datos, su estructura es la mostrada en la figura 2 y están diseñados para almacenar en ellos de forma independiente cada una de las tres "subdirecciones" mencionadas con anterioridad. Las *constantes* que tiene la unidad corresponden a las direcciones donde se encuentran el tamaño de cada columna, el número de filas y el número de columnas de cada tabla.

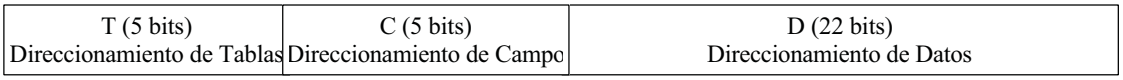

Figura 2: Estructura de los registros "punteros a memoria de datos".

A título de ejemplo veamos dos de las instrucciones específicas de base de datos que posee CBD:

"CCCI operando1, operando2, operando3": Esta instrucción realiza una operación de consulta (parecida a la instrucción SELECT de SQL). Para ello copia de una tabla origen a otra tabla destino, aquellas filas que cumplan una determinada condición. La condición que se ha de cumplir es que los "hechos" de las columnas de la tabla indicada por "operando1", coincidan con el "hecho" al que apunta "operando2". En "operando3" se indica la tabla destino.
"SSC operando1, operando2, operando3": Instrucción de salto (o bifurcación). Salta a la dirección de la memoria de código indicada por "operando3", si algunos de los "hechos" de la columna de la tabla indicada por el "operando1", coincide con el "hecho" al que apunta "operando2".

En total, CBD tiene definidas 72 instrucciones de las que 6 son específicas del manejo de bases de datos y con las que se pueden realizar tareas como actualizar contenidos, devolver direcciones de "hechos", etc.

3. Organización de CBD

En este apartado presentamos los diferentes componentes (a nivel RT) que forman el computador. Si observamos la figura 3, vemos que está compuesto por: nueve registros internos; tres unidades funcionales; dos buses; y una unidad de control (las memorias forman parte de un sistema externo). Pasemos a explicar cómo funcionan las partes más relevantes.

Las tres unidades funcionales que aparecen en la figura 3 son: El "Fichero de Registros", ya comentado en la sección anterior; la "Unidad Aritmético-Lógica" (ALU) que realiza operaciones estándar entre los datos de los dos buses acumulando el resultado en el registro RUF; y la "Unidad de Base de Datos" que comentamos seguidamente. La "Unidad de Base de Datos" está concebida para agilizar la unidad de control en las tareas de ejecutar las instrucciones específicas de base de datos. Estas instrucciones, aunque distintas, comparten ciertas funcionalidades como copiar y comparar "hechos", recorrer los diferentes "hechos" de una columna cualquiera de una determinada tabla, o llevar la cuenta del número de filas y de columnas tratados. Esta unidad funcional, relativamente compleja está compuesta por registros, comparadores, sumadores, etc., disponibles para que la unidad de control ejecute

estas funcionalidades de forma más rápida que si las hiciera con los registros internos y con la propia ALU del computador.

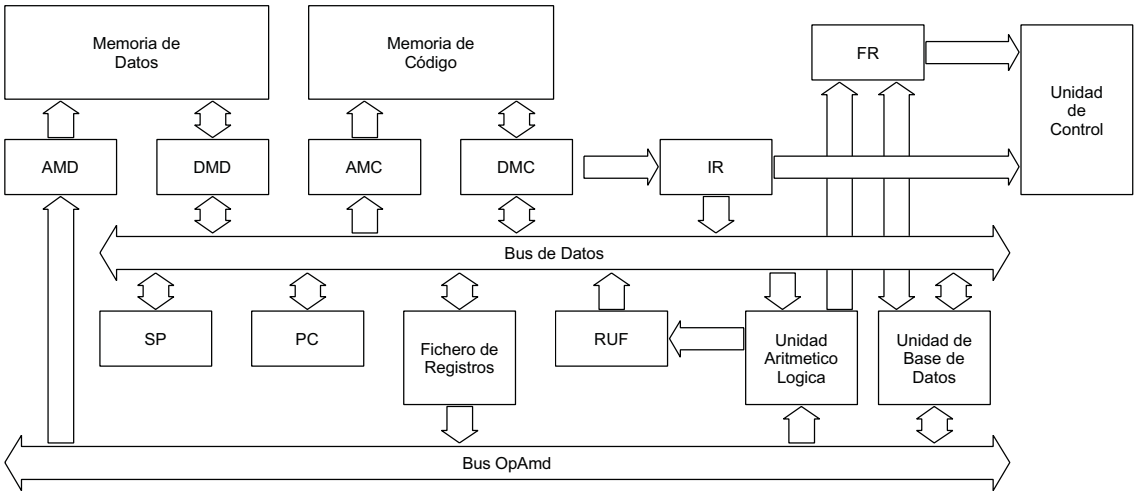

Figura 3: Organización a nivel RT de la CPU.

Por último y en relación a la unidad de control comentemos que dispone de: microinstrucciones capaces de realizar bifurcaciones; una pequeña pila para que el microprograma salte a subrutinas; un contador de las microinstrucciones en curso; y una ROM que contiene el microprograma de control.

4. Conclusiones

Este trabajo ha presentado un prototipo de computador orientado al manejo de base de datos, el cual está dotado de instrucciones capaces de insertar, modificar, eliminar y consultar información de una forma parecida a como lo hacen los lenguajes específicos. El computador está compuesto de una memoria de datos específica para la base de datos y una memoria de código. La información referente a base de datos está almacenada en forma de tablas y, para no limitar el tamaño de estas tablas y optimizar los accesos continuos a éstas, el computador dispone de los mecanismos necesarios para soportar caché y memoria virtual. En cuanto a su organización incorpora, como principales aspectos, el diseño de una unidad funcional específica para operar con los datos, un fichero de registros para direccionamiento múltiple y una compleja unidad de control microprogramada que permite bifurcaciones, saltos, y subrutinas para las microinstrucciones.

Referencias

[1] D. Hamilton. Inside ADABAS: Introduction to Direct Calls and ADASQL. Addison Wesley (1991)
[2] Prolog Development Center. PCD Prolog version 3.20. Ed. Prolog Develop. Center (1990)
[3] W. Stalling. Organización y Arquitectura de Computadores. Prentice Hall (2000).
[4] C. Baena, J.I. Escudero, I. Gómez y M. Valencia. Introducción a las Sistemas Digitales. Dpto de Tecnología Electrónica. Universidad de Sevilla (1997).
[5] D. Peñalosa. Diseño de un Procesador Específico para Bases de Datos. Proyecto Fin de Carrera (1998).