

## **Unidad II: Estructura y funcionamiento de la CPU**

### **2.1 Organización del procesador**

Un procesador, incluye tanto registros visibles por el usuario como registros de control/estado. Los registros visibles por el usuario pueden ser de uso general o tener una utilidad especial, mientras que los registros de control y estado se usan para controlar el funcionamiento del procesador, un claro ejemplo es el contador de programa.

Los procesadores utilizan la segmentación de instrucciones para acelerar la ejecución. La segmentación de cauce se puede dividir en ciclo de instrucción en varias etapas separadas que operan secuencialmente, tales como la captación de instrucción, decodificación de instrucción, cálculo de direcciones de operando, ejecución de instrucción y estructura del operando resultado.

### **2.2 Estructura de registros**

El Registro está organizado en una estructura jerárquica compuesta por subárboles con sus respectivas claves, subclaves y entradas.

El contenido del Registro puede variar considerablemente de un equipo a otro, en función de los dispositivos, servicios y programas instalados en cada equipo.

Las claves pueden contener subclaves que, a su vez, pueden contener otras subclaves. Aunque la mayor parte de la información del Registro se almacena en disco y se considera permanente, algunos datos almacenados en *claves volátiles* se sobrescriben cada vez que se inicia el sistema operativo.

#### **2.2.1 Registros visibles para el usuario**

Minimizan el acceso entre el microprocesador y cpu, debido a que se encuentran dentro del CPU, son manipulables por lenguaje de máquina. Aquí tenemos los 2 principales:

1) Registros de Datos. (Manipulables a bajo nivel).

2) Registro de Direcciones (Manipulables a bajo nivel).

- Registro de índice.

Sirve para direccionamiento, es similar al "PC", pero tiene un valor base de autoincremento: Ejemplo: Base 3  $\diamond$  3, 6, 9...n, n+1; que se irá sumando para obtener una dirección más efectiva.

## 2.2.2 Registros de control y de estados

Hay diversos registros del procesador que se emplean para controlar su funcionamiento. La mayoría de ellos, en la mayor parte de las máquinas, no son visibles por el usuario. Algunos de ellos pueden ser visibles por ciertas instrucciones máquina ejecutadas en un modo de control de sistema operativo.

Naturalmente, máquinas diferentes tendrán distintas organizaciones de registros y usarán distinta terminología. A continuación se presenta una lista razonablemente completa de tipos de registros, con una breve descripción.

Son esenciales cuatro registros para la ejecución de una instrucción:

- Contador de programa (Program Counter, PC): contiene la dirección de la instrucción a captar.
- Registro de instrucción (Instruction Register, IR): contiene la instrucción captada más recientemente.
- Registro de dirección de memoria (Memory Address Register, MAR): contiene la dirección de una posición de memoria.
- Registro intermedio de memoria (Memory Buffer Register, MBR): contiene la palabra de datos a escribir en memoria o la palabra leída más recientemente.

No todos los procesadores tienen registros internos designados como MAR y MBR, pero es necesario algún mecanismo de almacenamiento intermedio

equivalente mediante el cual se dé salida a los bits que van a ser transferidos al bus del sistema y se almacenen temporalmente los bits leídos del bus de datos.

Típicamente, el procesador actualiza PC después de cada captación de instrucción de manera que siempre apunta a la siguiente instrucción a ejecutar. Una instrucción de bifurcación o salto también modificará el contenido de PC. La instrucción captada se carga en IR, donde son analizados el código de operación y los campos de operando. Se intercambian datos con la memoria por medio de MAR y de MBR. En un sistema con organización de bus, MAR se conecta directamente al bus de direcciones, y MBR directamente al bus de datos. Los registros visibles por el usuario repetidamente intercambian datos con MBR.

Los cuatro registros que se acaban de mencionar se usan para la transferencia de datos entre el procesador y la memoria. Dentro del procesador, los datos tienen que ofrecerse a la ALU para su procesamiento.

La ALU puede tener acceso directo a MBR y a los registros visibles por el usuario. Como alternativa, puede haber registros intermedios adicionales en torno a la ALU; estos registros sirven como registros de entrada y salida de la ALU e intercambian datos con MBR y los registros visibles por el usuario.

### **2.2.3 Ejemplos de organización de registros de CPU reales**

En algún diseño concreto de procesador es posible encontrar otros registros relativos a estado y control. Puede existir un puntero a un bloque de memoria que contenga información de estado adicional (por ejemplo, bloques de control de procesos). En las máquinas que usan interrupciones vectorizadas puede existir un registro de vector de interrupción. Si se utiliza una pila para llevar a cabo ciertas funciones (por ejemplo, llamada a subrutina), se necesita un puntero de pila del sistema. En un sistema de memoria virtual se usa un puntero a la tabla de

páginas. Por último, pueden emplearse registros para el control de operaciones de E/S.

En el diseño de la organización de los registros de control y estado entran en juego varios factores.

Una cuestión primordial es el soporte del sistema operativo. Algunos tipos de información de control son de utilidad específica para el sistema operativo. Si el diseñador del procesador posee una comprensión funcional del sistema operativo que se va a utilizar, la organización de los registros puede adaptarse hasta cierto punto a ese sistema operativo.

Otra decisión importante en el diseño es la distribución de información de control entre registros y memoria. Es frecuente dedicar los primeros (más bajos) pocos cientos o miles de palabras de memoria para fines de control. El diseñador debe decidir cuánta información de control debiera estar en registros y cuánta en memoria. Se presenta el compromiso habitual entre coste y velocidad.

## **2.3 El ciclo de instrucción**

### **2.3.1 Ciclo Fetch-Decode-Execute**

Un ciclo de instrucción (también llamado ciclo de *fetch-and-execute* o ciclo de *fetch-decode-execute* en inglés) es el período que tarda la unidad central de proceso (CPU) en ejecutar una instrucción de lenguaje máquina.

Comprende una secuencia de acciones determinada que debe llevar a cabo la CPU para ejecutar cada instrucción en un programa. Cada instrucción del juego de instrucciones de una CPU puede requerir diferente número de ciclos de instrucción para su ejecución. Un ciclo de instrucción está formado por uno o más ciclos máquina.

Para que cualquier sistema de proceso de datos basado en microprocesador (por ejemplo un ordenador) o microcontrolador (por ejemplo un reproductor de MP3) realice una tarea (programa) primero debe buscar cada instrucción en la memoria principal y luego ejecutarla.

## **2.3.2 Segmentación de instrucciones**

### **2.3.3 Conjunto de instrucciones: Características y funciones**

La segmentación de instrucciones es similar al uso de una cadena de montaje en una fábrica de manufacturación. En las cadenas de montaje, el producto pasa a través de varias etapas de producción antes de tener el producto terminado. Cada etapa o segmento de la cadena está especializada en un área específica de la línea de producción y lleva a cabo siempre la misma actividad. Esta tecnología es aplicada en el diseño de procesadores eficientes.

A estos procesadores se les conoce como pipeline processors. Estos están compuestos por una lista de segmentos lineales y secuenciales en donde cada segmento lleva a cabo una tarea o un grupo de tareas computacionales. Los datos que provienen del exterior se introducen en el sistema para ser procesados. La computadora realiza operaciones con los datos que tiene almacenados en memoria, produce nuevos datos o información para uso externo.

Las arquitecturas y los conjuntos de instrucciones se pueden clasificar considerando los siguientes aspectos:

Almacenamiento de operandos en la CPU: dónde se ubican los operandos aparte de la memoria.

Número de operandos explícitos por instrucción: cuántos operandos se expresan en forma explícita en una instrucción típica. Normalmente son 0, 1, 2 y 3.

Posición del operando: ¿Puede cualquier operando estar en memoria?, o deben estar algunos o todos en los registros internos de la CPU. Cómo se especifica la dirección de memoria (modos de direccionamiento disponibles).

Operaciones: Qué operaciones están disponibles en el conjunto de instrucciones.

Tipo y tamaño de operandos y cómo se especifican.

### **2.3.4 Modos de direccionamiento y formatos**

El campo de operación de una instrucción especifica la operación que se debe realizar. Esta debe ser ejecutada sobre algunos datos almacenados en registros del computador o en palabras de memoria, es decir, sobre los operandos. El modo de direccionamiento especifica la forma de interpretar la información contenida en cada campo de operando para localizar, en base a esta información, el operando.