

RISC vs CISC

De los tres componentes principales de una CPU, conjunto de registros conectados en bus, unidad aritmética/lógica y unidad de control, ésta última es la mas compleja de diseñar pues debe proporcionar las señales de control para cargar registros, borrar su contenido, incrementarlo, seleccionar operaciones en la ALU, crear las banderas de estado etc, y esto es para cada instrucción diferente con que haya sido diseñada la ALU.

Cuando el número de instrucciones de ensamblador llega a ser demasiado grande (de 150 a 400 instrucciones), el diseño de la ALU es tan complejo que se recurre al uso de un sistema microprogramado; es decir, el circuito de combinación de la Unidad de Control se sustituye por una memoria de control y un microcódigo que generarán las señales de control requeridas.

El uso de una memoria, aunque muy rápida le añade retardos en el procesamiento pues puede llegar al caso de desperdiciar pulsos de reloj mientras se accesa la memoria.

La manera en la que se puede mantener una unidad de control alambrada (diseñada con compuertas si memoria de por medio) que trabaja mucho mas rápido que la unidad microprogramada es que no existan demasiadas instrucciones de ensamblador. Algunos autores acotan esta cantidad a un rango entre 75 y 90 instrucciones. Veamos un ejemplo:

Supongamos que para realizar la prueba de saltar hacia cierta etiqueta contamos con el siguiente conjunto de instrucciones de salto (recuerda que estas instrucciones se escriben después de una instrucción CMP CX, dato):

JE etiqueta ; salta si la resta del registro CX menos el contenido de la variable dato es cero (*)

JG etiqueta ; salta si (CX – dato) es mayor que cero (*)

JGE etiqueta ; salta si (CX – dato) es mayor o igual a cero

JL etiqueta ; salta si (CX – dato) es menor que cero (*)

JLE etiqueta ; salta si (CX – dato) es menor o igual a cero

JNE etiqueta ; salta si (CX – dato) es diferente a cero

Si por ejemplo el programador desea saltar al inicio de un ciclo mientras el resultado de la operación CX – dato sea mayor o igual que cero, solo deberá escribir una sola instrucción. Pero, si solo se contara con las instrucciones marcadas con (*), entonces se debe recurrir a escribir las instrucciones:

JE etiqueta

JG etiqueta

En forma consecutiva para obtener el mismo resultado. Así, mientras el conjunto de instrucciones sea pequeño, el programador deberá escribir programas en ensamblador con mas líneas pero el CPU podrá tener una unidad de control alambrada (hardware 100%) más rápida.

Veamos otro ejemplo, si solo se contara con las instrucciones ADD (sumar), COM (obtener el complemento a uno) e INC (incrementar en uno un registro), se podría obtener la resta de dos datos escribiendo el código como sigue:

LD R1, minuendo

LD R2, sustraendo

COM R2

INC R2

ADD R1, R2

Esta misma operación la obtenemos en una computadora CISC con el siguiente código:

MOV AX, minuendo

SUB AX, sustraendo

Lo que desde el punto de vista del programador le simplifica el trabajo, a cambio de consumir mas tiempo para ejecutar este último código que las 5 líneas anteriores. También habrás notado que en las instrucciones RISC se maneja una nomenclatura diferente de los registros, pues en este tipo de CPU existe una mayor cantidad de registros procesadores con propósitos múltiples, lo que ayuda a no guardar constantemente resultados parciales que serán empleados en instrucciones subsiguientes, ahorrando una considerable cantidad de pulsos de reloj que son desperdiciados cuando se accesa la RAM, algunos autores estiman que se pierden hasta 7 ciclos por cada acceso a memoria RAM aunque esto dependerá de la rapidez de acceso.

Veamos las ventajas de RISC contra CISC

- Mayor velocidad de ejecución del código aunque este sea de mucho mas líneas
- Diseño alambrado de la unidad de control que genera menor consumo de tiempo y energía

Como desventaja está el hecho de que si se desea incrementar una sola instrucción se deberá rediseñar completamente la unidad de control.

Ventajas CISC

- Código mas pequeño
- Mayor facilidad de programación
- Si se aumenta una instrucción solo debe actualizarse el microcódigo sin rediseñar la unidad de control.

A propósito, ¿sabes lo que significan los términos CISC y RISC?

Para complementar este tema analiza el video adjunto