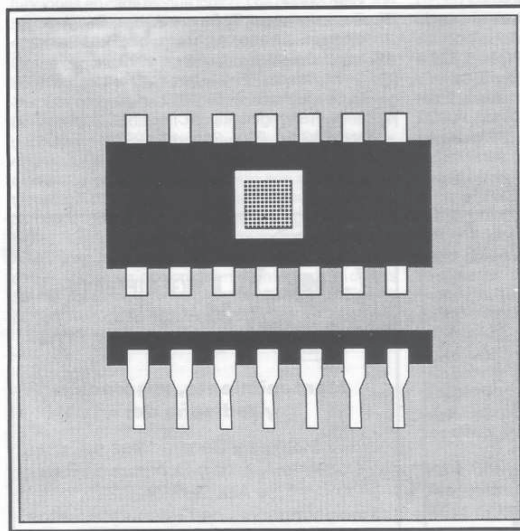


Microprocesadores con conjunto de instrucciones reducido y complejo



RESUMEN

La arquitectura de un microprocesador se determina por muchos factores tales como: la aplicación, la velocidad, la capacidad de cálculo matemático, la cantidad de direccionamiento de memoria, el consumo de potencia, resistencia a radiaciones, etcétera. Se ha ganado mucha experiencia en el diseño de microprocesadores y se han probado nuevas arquitecturas con el ánimo de hacer la máquina más eficiente; esto implica que se pueda procesar un mayor número de instrucciones por segundo. Actualmente existe una polémica entre dos arquitecturas llamadas CISC (Complex instruction set computer) y RISC (Reduced instruction set computer). En adelante se hará una discusión acerca de las características de estas arquitecturas, mostrando las filosofías en las cuales se basan sus diseñadores para concebir los dos tipos de microprocesadores que actualmente están en las máquinas de cómputo más populares en el mundo.

LOS PROCESADORES CISC

Los primeros microprocesadores poseían un conjunto de instrucciones pequeño, debido principalmente al poco desarrollo de la tecnología para la fabricación de circuitos

□ Luis A. Gómez Arizabaleta
Ingeniero Electricista
Profesor Instructor
Departamento de
Electricidad
Universidad del Valle

□ Humberto Loalza Correa
Ingeniero Electricista
Profesor Instructor
Departamento de
Electricidad
Universidad del Valle

integrados, que impedía que la circuitería para la ejecución de instrucciones fuera compleja, y porque éstas eran las primeras generaciones de máquinas para las cuales aún no existían aplicaciones que les exigieran gran eficiencia. El desarrollo de los procesadores CISC aparece como respuesta a dos cuestionamientos principales: Simplificar los compiladores e incrementar la eficiencia de la máquina. Además, se necesita un procesador capaz de proveer un lenguaje de máquina que soporte con facilidad los lenguajes de alto nivel.

El primer requerimiento aparece debido a un avance en las técnicas de programación donde surgen las llamadas instrucciones de alto nivel. Para la ejecución de cada instrucción de alto nivel se requiere una serie de instrucciones propias del microprocesador en que se trabaja.

La labor del compilador es traducir todos los comandos de alto nivel en una serie de comandos propios de la máquina. Esta traducción no es directa e implica que por cada instrucción de alto nivel se deba ejecutar todo un programa de bajo nivel que realiza en últimas la labor expresada en cada estamento. Por lo tanto, si el microprocesador posee un conjunto de instrucciones poco complejo, obliga a que las técnicas de compilación sean más complicadas y se generen programas muy largos que toman mucho tiempo en ejecutarse. Se requiere de una relación más directa entre las técnicas de compilación y las instrucciones propias del microprocesador, esto significa que la máquina debe sintetizar instrucciones de bajo nivel que son casi idénticas (o por lo menos se aproximan mucho) a las instrucciones de alto nivel. Una consecuencia directa de esto es que la circuitería que realiza cada instrucción se hace mucho más compleja y por lo tanto el área de pastilla de circuito integrado dedicada a las instrucciones es

mayor, dejando menos espacio para otras funciones del microprocesador como registros, circuitos aritméticos, circuitos de interfase, etcétera.

El segundo requerimiento está relacionado con el primero, ya que si existe un vínculo más directo entre las instrucciones de la máquina y el lenguaje de alto nivel, no sólo se logra una labor de compilación más veloz, sino que además el programa resultante en bajo nivel tiene un número de instrucciones mucho menor que el que se lograría con un microprocesador de características más pobres.

Por último, el hecho de que las instrucciones en bajo nivel se aproximen mucho a los estamentos y proposiciones de alto nivel resulta en un soporte directo entre los lenguajes de bajo nivel y los de alto nivel.

Las premisas anteriores no se cumplen en la realidad, debido principalmente a los factores que se mencionan a continuación.

Se ha encontrado que las máquinas CISC resultan difíciles de explotar, ésto es, no se encuentra una relación directa entre una estructura en alto nivel y una instrucción del procesador (Hennessy, Radin, Patterson). Otra evidencia se encontró cuando luego del análisis de varios compiladores se contabilizó el número y el tipo de instrucciones que se emplearon en la traducción de programas de alto nivel y se observó que la mayoría de las instrucciones empleadas eran las de formato más sencillo. Además, uno de los argumentos principales sobre los que se basa la arquitectura CISC, es que si se logran programas compilados de menor extensión se obtendrá un ahorro en circuitos de memoria. Este último argumento era válido en los inicios de la computación, cuando los circuitos de memoria eran costosos y la memoria un recurso escaso. Sin embargo, hoy la situación ha cambiado y este argumento ya no es válido, puesto que se pueden ge-

nerar programas relativamente grandes sin incurrir en un costo elevado en los circuitos de memoria. Además se encontró que no hay un ahorro significativo en memoria entre programas compilados por los procesadores CISC y aquellos generados por los RISC.

En realidad el supuesto ahorro en el número de instrucciones es una ilusión, ya que la codificación de las instrucciones para los procesadores CISC requiere un mayor número de BITS lo que implica un número mayor de BYTES (ver tabla No 1). Por último, la supuesta mayor eficiencia ganada por ejecutar programas más cortos (ejecución más rápida) se pierde ya que una instrucción compleja, codificada en un solo comando de bajo nivel, resulta en una serie de acciones que toman muchos ciclos de reloj y, por lo tanto, consumen mucho tiempo.

En resumen, las supuestas mejoras en la eficiencia de la máquina, que se ganarían con la arquitectura CISC no ocurrieron en la realidad, debido principalmente a que las expectativas del rendimiento de las máquinas se basaron en premisas que no tenían en cuenta el desarrollo futuro del hard y el soft. Como respuesta al fenómeno CISC, algunos investigadores de arquitectura de computadoras reorientaron sus esfuerzos tratando de corregir los errores del pasado. Nace una nueva arquitectura: la llamada RISC, o computador con conjunto de instrucciones reducido.

LOS PROCESADORES RISC

Actualmente existen muchas máquinas con características RISC y cada una presenta alguna variante en su arquitectura respecto a las otras, sin embargo, la mayoría poseen las características básicas de un procesador RISC. Estas características que fueron definidas por un grupo de investigadores en la Universidad de Berkeley son:

1. Ejecución de una instrucción por ciclo de reloj.

	11 PROGRAMAS EN C.	12 PROGRAMAS EN C.	5 PROGRAMAS EN C.
RISC I	1.0	1.0	1.0
VAX-11/780	0.8	0.67	-
M68000	0.9	-	0.9
Z8002	1.2	-	1.12
PDP-11/70	0.9	0.71	-

TABLA No 1: Tamaño del código generado por varias máquinas con relación a la máquina RISC I

- Operaciones registro - registro.
- Modos de direccionamiento sencillo.
- Formatos de instrucciones simples.

Estas características fueron derivadas luego de un estudio hecho sobre los procesadores CISC, algunas de ellas representan un cambio significativo en la arquitectura de los microprocesadores. En adelante se examinarán todas las características mencionadas mostrando sus ventajas y desventajas.

Ejecución de una instrucción por ciclo de reloj. Típicamente un ciclo de reloj corresponde al período de una onda cuadrada de voltaje, que sirve como referencia de tiempo para la ejecución de las labores de la máquina. Es común en los procesadores con arquitecturas menos recientes que cada instrucción de bajo nivel tome varios ciclos de reloj para ejecutarse. Cuanto más compleja es la instrucción más ciclos tomará y por lo tanto será más lenta.

De acuerdo con lo anterior, sería deseable que las instrucciones se ejecutaran en el menor número de ciclos posible. Esto no se logra con instrucciones complejas como la multiplicación y la división, donde duran alrededor de 16 ciclos de reloj como en el caso del microprocesador 8088 de Intel. Por lo tanto se hace una restricción y las instrucciones se ejecutan en un sólo ciclo. Esto obliga a que se cableen, dentro del microprocesador, instrucciones sencillas

(cuya ejecución pueda lograrse en un sólo ciclo de reloj), pero aún con estas instrucciones se corre el riesgo de que tomen más de un ciclo, lo que lleva a la máquina a presentar un hardware más complejo que pueda soportar este objetivo (mayor grado de paralelismo). Por ejemplo, una simple instrucción de suma expresada como:

SUM A,B,C

donde se indica que se debe sumar el contenido del registro A con el contenido del registro B y depositar el resultado en el registro C, tomaría tres ciclos de reloj si se ejecuta en un procesador con una arquitectura de bus único, sin embargo, tomaría dos ciclos de reloj si se ejecuta en un procesador con arquitectura de dos buses y por último tomaría "un" sólo ciclo de reloj si se ejecuta en un procesador de arquitectura de tres buses (ver figura 1).

Operaciones registro a registro. Otra característica importante de los procesadores RISC es que poseen una gran cantidad de registros internos de propósito general, en contraste con los procesadores tradicionales. Por ejemplo, un procesador 8088 de Intel posee ocho registros de propósito general, en contraposición, un procesador RISC como el SPARC posee hasta 1024 registros diferentes. La ventaja de poseer gran cantidad de registros internos consiste en que tales registros funcionan como una memoria "cache" (cache es un vocablo inglés que significa "escondi-

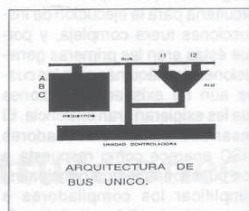


Figura 1. Diferentes arquitecturas de microprocesadores

Figura 1 A. La secuencia para la ejecución de la instrucción SUMAR A, B, C en una arquitectura de bus único es:

A → 11:
registro A a entrada de la ALU
1 ciclo de reloj.
B → 12:
registro B a entrada de la ALU
1 ciclo de reloj.
resultado → C:
resultado a registro C
1 ciclo de reloj.
TOTAL: se utilizaron 3 ciclos de reloj para completar la instrucción.

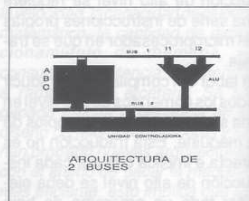


Figura 1 B. La secuencia para la ejecución de la instrucción SUMAR A, B, C en una arquitectura de 2 buses es:

A → 11:
registro A a entrada de la ALU
1 ciclo de reloj.
B → 12: suma;
resultado A a registro C
1 ciclo de reloj.
TOTAL: se utilizaron 2 ciclos de reloj para completar la instrucción.

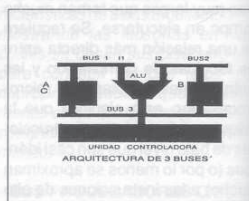


Figura 1 C. La secuencia para la ejecución de la instrucción SUMAR A, B, C en una arquitectura de 3 buses es:

A → 11; B → 12: suma;
resultado a registro C
1 ciclo de reloj.
TOTAL: se utilizó 1 ciclo de reloj para completar la instrucción.

te"). Una memoria "cache" permite almacenar datos e instrucciones a alta velocidad, lo cual permite que el acceso a los datos y códigos más utilizados en un programa sea muy rápido. Sin embargo, esta misma característica la hace sumamente costosa.

La memoria "cache" incrementa notablemente la eficiencia de un sistema, por lo que los diseñadores de procesadores prefieren tener una gran cantidad de registros internos, con el fin de lograr la ejecución de las instrucciones en forma mucho más veloz. Las instrucciones de los procesadores RISC son diseñadas para manipular los datos contenidos en los registros internos y hacer el menor número de accesos posibles a la memoria externa. Por lo tanto, las instrucciones de este tipo de procesadores son preferiblemente de registro a registro.

Modos de direccionamiento sencillo. Los modos de direccionamiento hacen referencia a la forma como el procesador accede a los operandos de una instrucción; estos pueden residir en la memoria externa o estar codificados dentro de la propia instrucción. En el primer caso se debe diseñar la máquina para que acceda a los operandos mediante el proceso de lectura de la memoria externa; esta opera-

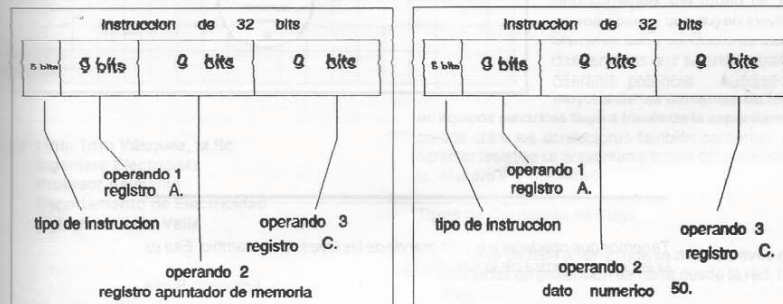
ción consume tiempo (ciclos de reloj adicionales) haciendo la instrucción más lenta. En el segundo caso sólo se hace la lectura de memoria para capturar el código de la instrucción que se va a ejecutar; por esto, los operandos deben estar especificados dentro de dicho código de operación de forma que no se acceda a la memoria de nuevo durante la ejecución de la instrucción. Los operandos de una instrucción pueden ser sólo registros como en el caso de la instrucción SUM A,B,C o pueden ser registros y números como en el caso de SUM A,50,C donde se suma el contenido del registro "A" con el dato numérico 50 y el resultado es llevado al registro "C". En esta última instrucción se deben "codificar" los registros que servirán como operandos, además de incluir el dato numérico "50" (ver figura 2). En conclusión se prefiere tener modos de direccionamiento lo más simples posible, esto significa un ahorro de tiempo en la ejecución de la instrucción y hace que la unidad de decodificación de instrucciones sea sumamente sencilla. Una consecuencia de la codificación de todos los operandos dentro del código de operación de las instrucciones es que obliga a que este último sea de un número elevado de bits. Típicamente los procesadores RISC poseen códigos de opera-

ción de 32 bits. Un número menor de bits haría imposible incluir dentro de la misma instrucción todos los operandos descritos.

Formatos de instrucciones simples. El formato de una instrucción tiene que ver con la partición de los bits que forman el código de operación en "campos", que indican a la unidad de decodificación el tipo de operación a realizar y los operandos que debe emplear. Un ejemplo del formato de una instrucción se ve en la figura 3. Una máquina con formato de instrucciones variable, es decir, donde la división por "campos" es diferente para cada grupo de instrucciones, hace que la unidad de decodificación dentro del procesador sea sumamente compleja; por esta razón, los diseñadores de máquinas RISC prefieren que el formato sea único y sencillo.

CONCLUSIONES

A través del desarrollo histórico de las computadoras siempre se ha visto el afán de los diseñadores por lograr máquinas superiores cada vez. Las primeras máquinas eran de relativa sencillez y su conjunto de instrucciones estaba direccionado al cálculo matemático; sin embargo, las aplicaciones que los usuarios de computadoras daban a estas máquinas exigían nuevas



Figuras 2 y 3. Formato de instrucciones.

metodologías de diseño y el desarrollo de muchos conceptos que podían ser útiles para lograr el objetivo. Así surgen las tendencias que hoy encontramos en las máquinas de cómputo. Tales tendencias no sólo ocurren en las computadoras de propósito general sino también en otras máquinas diseñadas para aplicaciones específicas, como los llamados procesadores de señales digitales (DSP), procesadores de video, multiprocesadores, etcétera.

Las máquinas CISC y RISC surgen de dos filosofías de diseño completamente diferentes, en la que RISC trata de solucionar los problemas que presenta la CISC. Sin embargo, no se puede determinar si las máquinas RISC son más o menos eficientes que las CISC, esto debido principalmente a que no existen parámetros de referencia para hacer mediciones entre máquinas de ambas tendencias y no se podría decir que para cada máquina CISC

existe su contraparte RISC, sobre las cuales se pueda hacer mediciones de eficiencia. Lo cierto es que ambas filosofías de diseño están en plena vigencia en el mundo y existen computadoras construidas con base en ambos tipos de procesadores, todas con gran acogida. Dos ejemplos típicos son: el procesador PENTIUM de Intel que es una máquina CISC y el procesador SPARC de SUN-microsystems, que es una máquina RISC. El primero es utilizado en computadores personales mientras que el segundo se emplea en estaciones de trabajo; ambas máquinas son muy populares y cada una cubre campos de aplicación diferentes. En últimas, son los usuarios los que deciden que máquina emplear en sus aplicaciones, de acuerdo con sus necesidades ya que independientemente de la máquina que ofrece las mejores características de eficiencia, los usuarios promedio de nuestro país

tienen limitaciones económicas como para adquirir el último grito de la moda en computadores y además muchas de sus aplicaciones no requieren de máquinas muy avanzadas.

2. BIBLIOGRAFIA

- [1] HENNESSY, J., ET AL. "Hardware/Software Tradeoffs for Increased Performance" Proceedings, Symposium on Architectural Support for Programming Languages and Operating Systems, March, 1982.
- [2] RADIN, G. "The 801 mini-computer." IBM Journal of Research and Development, May, 1983.
- [3] PATTERSON, D., and SEQUIN, C. "A VLSI RISC." Computer, September, 1982.