

# **TECNOLOGIAS RISC y CISC**

## **8.1 - RISC**

### **8.1.1 - Definición**

La tendencia de hacer instrucciones de máquina cada vez más complejas, resulta en códigos de operación de tan alto nivel que tiende a romper la diferencia entre assembler y compilador.

Pero estudios al respecto revelaron elevados tiempos de diseño, aumento de errores, e implementaciones inconsistentes. Además se descubrió que las instrucciones muy complejas no eran utilizadas con mucha frecuencia y en muchos casos eran responsables del retardo de la performance de todo el sistema.

Esto último en particular debido a que la Unidad de Control debe ser más compleja y si es microprogramada provoca que todas las instrucciones demoren mayor tiempo en su ejecución.

Con el propósito de explotar nuevos caminos aparecen las tecnologías RISC (Reduced Instruction Set Computer) para oponerse a las CISC (Complex Instruction Set Computer) que son las que hemos visto hasta ahora.

En un intento de definir la tecnología RISC el grupo original de investigación en el proyecto RISC llevado a cabo en Berkeley (RISC I) produjo una filosofía de diseño que puede resumirse como sigue:

- 1) Analizar el objeto sobre el cual se desarrollan las aplicaciones a fin de determinar cuáles son las operaciones más frecuentes.
- 2) Optimizar los caminos que deben recorrer los datos para ejecutar las operaciones o instrucciones (del punto 1) tan rápido como sea posible.
- 3) Incluir otras instrucciones solo si forman parte de los caminos optimizados previamente y si son de relativa frecuencia, y si su inclusión no entorpece la ejecución de las instrucciones más frecuentes.
- 4) Aplicar una estrategia similar a los otros recursos del procesador. Incluir un recurso solo si éste está justificado por su frecuencia de uso, y su inclusión no entorpece a otros recursos más utilizados.
- 5) Tratar de trasladar lo más que se pueda la complejidad en tiempos de ejecución al momento de compilación, recargando el software de compilación y liberando al hardware de ejecución.

Algunas de las características más comunes que pueden verse en computadoras de tecnología RISC son:

- muchas de las instrucciones se ejecutan en un solo ciclo de máquina
- conjunto de instrucciones de cargar/almacenar (load/store). Es decir se accede a memoria exclusivamente mediante las instrucciones load y store, el resto de las instrucciones realizan sus operaciones entre registros
- decodificación de instrucciones hardwired (en oposición a la técnica de microprograma)
- existen relativamente pocas instrucciones y modos de direccionamiento
- todas las instrucciones deberían tener la misma longitud para facilitar la tarea de decodificación y homogeneizar los tiempos de carga de las mismas
- la complejidad se ha desplazado hacia los compiladores optimizados
- existe un alto grado de pipeline en los caminos de los datos para obtener mucha concurrencia
- gran cantidad de registros (windowed o no-windowed)
- muchos niveles de jerarquías de memoria
- conjunto de instrucciones diseñado para determinada clase de aplicaciones.
- soporte de lenguajes de alto nivel (Esto último debido a que como hay pocas instrucciones sobraría lugar en el chip para colocar elementos que ayuden a lenguajes de alto nivel, como manejo de listas, stacks, etc.)

Debe tenerse cuidado con una clasificación de computadoras de tecnología RISC, ya que, por ejemplo, muchas computadoras tienen decodificadores de instrucciones microcodificadas, un conjunto grande de instrucciones y un conjunto pequeño de registros pero sin embargo son de diseño definitivamente RISC.

La cuestión importante es que **esta filosofía RISC está respaldada en el diseño de un procesador para una aplicación específica.**

### **8.1.2 - Implementaciones de procesadores RISC**

Los diseños RISC están disponibles como:

- microprocesadores de un único chip
- conjuntos de chips de muy alta escala de integración con funciones más poderosas
- computadoras de placa única (single-board)
- superminicomputadoras

La mayor performance se está obteniendo de las implementaciones de arseniuro de galio (GaAs). Estos productos son los que tardarán más en llegar al mercado, los procesadores de GaAs tienen la ventaja de su inmensa velocidad.

Investigaremos primero el diseño de la CPU poniendo énfasis en el conjunto de instrucciones, decodificación de instrucciones, el camino de los datos, el diseño de registros, las unidades de ejecución, manejo de las bifurcaciones y el sistema de diseño de la memoria.

Luego investigaremos el RISC como un sistema, enfatizando las unidades de múltiple ejecución, el soporte de coprocesador, el multiprocesamiento, el soporte de sistema operativo y los lenguajes utilizados. Finalmente compararemos performance.

### 8.1.3 - CUESTIÓN CPU

#### 8.1.3.1 - Conjunto de instrucciones

El tamaño de los sets de instrucciones varía desde un mínimo de 16 instrucciones a aproximadamente 268.

Sin embargo el tamaño de la instrucción difiere en todas las máquinas RISC analizadas, todas utilizan formatos de instrucción que permiten una rápida decodificación utilizando un campo de código de operación consistente.

#### 8.1.3.2 - Decodificación de las instrucciones

En este análisis existen dos diseños básicos de los decodificadores de instrucción y algunas combinaciones interesantes.

Recuérdese que los ejemplos de RISC antiguos utilizaban todos decodificadores de lógica hardwired por la posible facilidad y rapidez del decodificado de la instrucción.

Muchos de los procesadores utilizan algún tipo de decodificador de instrucción estrictamente por hardware o utilizan decodificación de tipo hardwired diseñada con una lógica minimizada. Otros hacen uso combinado de decodificación de tipo hardwired y microcódigo, algunos de estos casos tienen una unidad separada para obtención anticipada del código de operación (prefetch) y decodificación del mismo. La instrucción decodificada se carga luego en una memoria cache de instrucciones.

#### 8.1.3.3 - El camino de los datos (datapath)

El diseño del datapath es del todo complejo en todos los procesadores analizados, tienen pipelines de una profundidad que varían desde 2 estadios (stages, etapas) hasta 7 estadios en los procesadores de arseniuro de galio.

En forma general, cuanto más corto es el tiempo del ciclo más profundo es el pipeline. Este fenómeno se debe a dos factores:

- 1º) todos los ejemplos de procesadores intentan comenzar a ejecutar una nueva instrucción cada golpe de reloj (clock cycle).
- 2º) los procesadores tienen accesos a memoria y retardos electrónicos que consumen una gran porción del período del ciclo de reloj.

Luego deben lograrse más accesos a memoria para obtener las instrucciones.

La Fig. 8.1 ilustra un pipe de 4 instrucciones de uno de los modelos de los procesadores RISC (el MIPS)

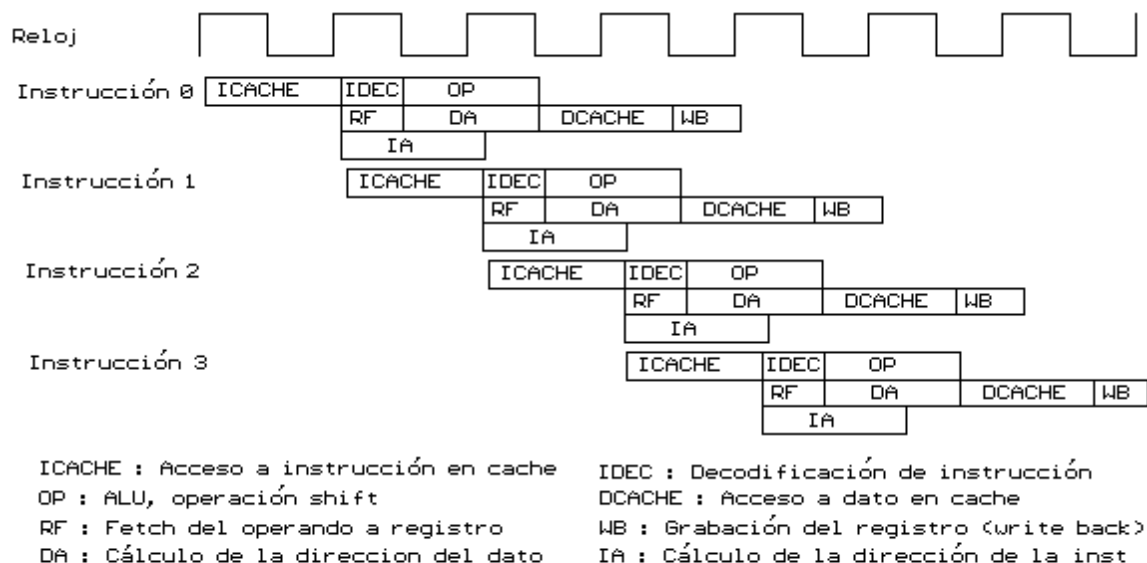


Fig. 8.1. - Un pipeline de 4 instrucciones de ejemplo.

cuyas instrucciones tienen todas una longitud de 32 bits.

Es bastante dificultoso mostrar una figura del número-de-etapas para este diseño debido a que existen estadios de medio-ciclo y ciclo-completo, y actividades concurrentes dentro de la ejecución de cada instrucción.

El ciclo de reloj es de 60 nanosegundos divididos en dos fases de 30 nanosegundos.

*La complejidad está parcialmente justificada para ejecutar todas las instrucciones en un único ciclo.*

#### 8.1.3.4 - Diseño de registros

Se ha comprobado mediante mediciones que la mayoría de los procesadores CISC :

- gasta el 70 % de su tiempo accediendo a los operandos
- los operandos accedidos con más frecuencia son escalares
- cerca del 80 % de las referencias a escalares son locales al procedimiento
- el pasaje de parámetros representa alrededor del 10 % del total del tráfico de memoria

Y considerando que el porcentaje de llamadas a procedimientos ronda el 25 a 40 % del tiempo total de ejecución, esto sumado al ahorro que puede obtenerse en cuanto al manejo de las variables locales y al pasaje de parámetros indujo a que en todos los diseños RISC se intentara sacar provecho de la performance mediante:

- 1) la velocidad que se obtiene al almacenar variables dentro del chip
- 2) la habilidad del compilador para aprovechar en forma efectiva la propiedad de localidad de las variables de un programa

Generalmente existe, en consecuencia, una gran cantidad de registros que están organizados en forma de ventanas múltiples solapadas y de tamaño fijo.

Existen conjuntos de registros de ambas modalidades: windowed y no windowed.

La técnica de **windows de registros** permite que un nuevo conjunto de registros esté disponible para cada procedimiento, permitiendo una superposición de unos pocos registros para el pasaje de argumentos.

Cuando un procedimiento agotó la cantidad de windows, una de esas windows se libera salvando los datos que contiene en la memoria.

Si el dato fuera a ser requerido nuevamente se restaura en la window desde memoria.

Las condiciones en que una window es salvada o restaurada se denominan overflow y underflow respectivamente.

El hardware para windows está organizado como un buffer circular que cubre siempre la parte superior del stack de ejecución.

El microprocesador RISC I posee un total de 138 registros (numerados de 0 a 137), que utiliza de la manera que se aprecia en la Fig. 8.2.

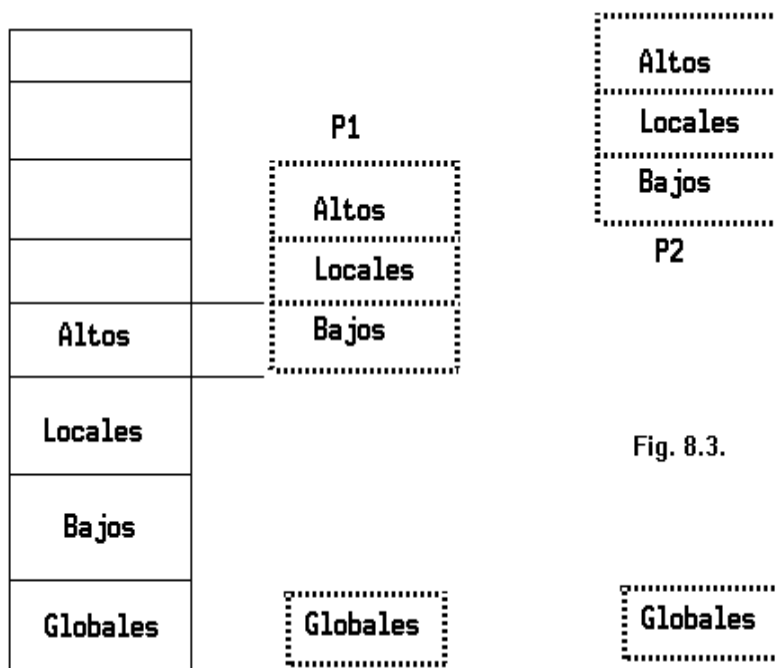
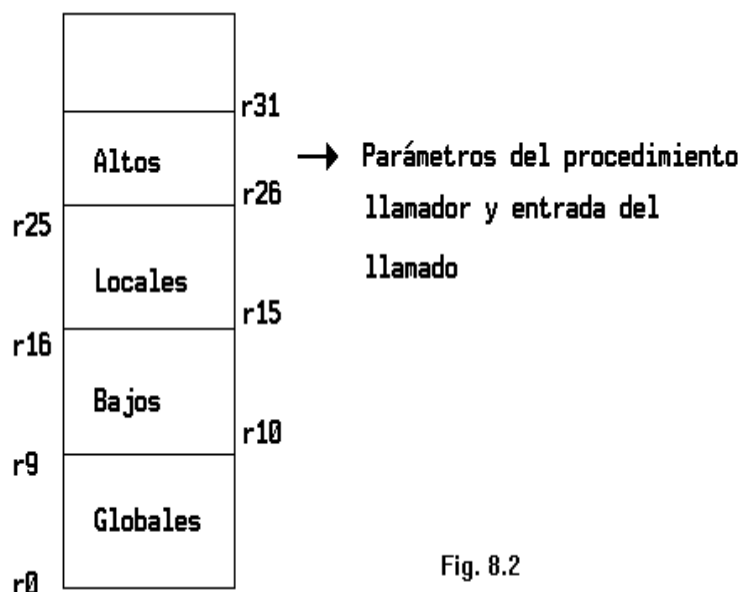
Como sistema de ventanas los usa de la forma que puede visualizarse en la figura 8.3.

O sea el procedimiento llamador coloca sus parámetros de llamada en sus registros Altos, los cuales se transforman en Bajos para el procedimiento llamado, o sea que el procedimiento llamado antes de comenzar su ejecución ya tiene sus parámetros a disposición.

RISC I adoptó el sistema de 8 ventanas, pues con ese número de determinó que se producirían problemas de overflow sólo en el 1% de los casos.

#### 8.1.3.5 - Unidades de ejecución

Existen procesadores que tienen unidades separadas para la ejecución concurrente de operaciones ALU y de E/S, en varios de ellos la unidad de E/S tiene una ALU separada para cálculo de direcciones o tienen dos unidades



lógicas separadas, una para Prefetch y Decodificación y otra para Ejecución o coprocesadores para realizar tareas complejas tales como manejo de cache, administración de memoria y cálculos de punto flotante.

#### 8.1.3.6 - Manejo de las bifurcaciones.

Varios estudios han demostrado que las ramas dentro de las bifurcaciones constituyen una fracción significativa del número de instrucciones ejecutadas.

En 1984 J. K. F. Lee y A. J. Smith realizaron un estudio sobre la cantidad de bifurcaciones en 26 programas totalizando una cantidad de 94 millones de instrucciones. El estudio consistió de una mezcla de programas de compilador, comerciales, científicos y de sistemas operativos corridos en una IBM S/370, programas educativos corridos en una PDP11-70 y varios programas científicos corridos en una CDC 6400.

Esos programas mostraban que en promedio alrededor de un 10 a un 30 por ciento de las instrucciones en un programa típico son instrucciones de bifurcación, de las cuales la bifurcación es tomada en un 60 a un 70 por ciento de las veces. Esto último provoca que un pipeline de instrucción trabaje efectivamente al 66 % de su máxima capacidad.

El diseño de las instrucciones de salto es crucial en toda arquitectura aunque es de mucha más relevancia en arquitecturas pipelined. En realidad el elemento decisivo de este tipo de instrucciones se obtiene en tiempo de ejecución.

Existen procesadores que utilizan predicción sobre las bifurcaciones al momento de compilación, basado en asociar un bit que indica la probabilidad de que dicha bifurcación se produzca o no.

La técnica del "desdoblamiento" de las bifurcaciones usa un campo generado dinámicamente de "próxima instrucción" para cada instrucción decodificada.

Otros utilizan la técnica de los slots, mediante la cual se llenan unos o dos slots que continúan a la ejecución de la bifurcación con las instrucciones siguientes. Cuanto más niveles tiene el pipeline más dificultoso se torna llenar estos slots.

La tecnología RISC utiliza la técnica de prebúsqueda de instrucciones, pero como sabemos esto causa problemas con las instrucciones de salto, por lo tanto la técnica más utilizada es la de "salto demorado", de modo que el salto se realice luego de la siguiente instrucción (Ver Fig. 8.4).

Las máquinas con salto tradicionales ejecutan la secuencia de instrucciones (primer columna). Para obtener el mismo efecto en RISC, el compilador inserta una instrucción de no-operación (NOP) después de cada salto (segunda columna). Como el salto retardado asegura que la bifurcación se efectúa sólo después que las siguientes instrucciones han sido ejecutadas, los compiladores RISC incluyen frecuentemente un optimizador que intenta reemplazar este NOP con una instrucción que pertenezca al bloque precedente a la transferencia (tercera columna).

Las nuevas versiones de RISC ya utilizan la detección de salto incondicionales en la Unidad I.

	<b>Salto Normal</b>	<b>Salto Retardado</b>	<b>Salto Optimizado</b>
<b>100</b>	<b>Load X, A</b>	<b>Load X, A</b>	<b>Load X, A</b>
<b>101</b>	<b>Add 1, A</b>	<b>Add 1, A</b>	<b>Jump 105</b>
<b>102</b>	<b>Jump 105</b>	<b>Jump 106</b>	<b>Add 1, A</b>
<b>103</b>	<b>Add A, B</b>	<b>Nop</b>	<b>Add A, B</b>
<b>104</b>	<b>Sub C, B</b>	<b>Add A, B</b>	<b>Sub C, B</b>
<b>105</b>	<b>Store A, Z</b>	<b>Sub C, B</b>	<b>Store A, Z</b>
<b>106</b>		<b>Store A, Z</b>	

**Fig. 8.4.**

#### 8.1.3.7 - Sistema de Memoria

Debido a la necesidad de retener las instrucciones y los datos que deben proveerse al procesador, los sistemas de memoria de los procesadores RISC son necesariamente muy complejos.

Se utilizan diversos niveles de jerarquía de memoria, muy a menudo con separación de datos e instrucciones.

Todos los procesadores analizados proveen el manejo de memoria virtual.

Por ahora tenemos diferentes estructuras de memoria, pero las más comunes incluyen un buffer de instrucciones dentro-del-chip de un tamaño suficiente para albergar unas pocas de las próximas instrucciones.

Este buffer se llena totalmente mediante una lógica de prefetch.

Algunos procesadores tienen una cache para datos y una cache para instrucciones, en tanto que otros solamente tienen una cache para todo o directamente carecen de ella.

Finalmente la memoria principal existe en construcciones fuera-del-chip y muy a menudo fuera del board del procesador.

De más está decir que el tiempo de acceso aumenta a medida que aumenta la distancia hasta la CPU o cuando aumenta el tamaño de la memoria.

#### 8.1.4 - División de funciones

Casi todos los RISC cuentan con coprocesadores de punto flotante (uno o dos).

Algunas implementaciones de los multiprocesadores cuentan con :

- un array de varios procesadores RISC con memoria propia en un solo circuito. Una unidad de ejecución escalar obtiene las instrucciones y provee el control para este array.
- dos chips para manejo de cache y de memoria; uno maneja los accesos a datos a cache y a memoria, y el otro maneja los accesos a instrucciones en cache y memoria, esto permite por tanto superponer los accesos a datos e instrucciones.

#### 8.1.4.1 - Sistemas Operativos / Lenguajes Soportados

El sistema operativo dominante es el UNIX con lenguajes: C, Fortran, Pascal, Ada y Cobol.

#### 8.1.5 - Notas de performance.

Los procesadores en este panorama pueden dividirse en dos grupos de performance.

El mayor de los grupos contiene a aquellos procesadores con implementaciones de silicio.

En la Fig. 8.5 podemos ver que la mayoría de los procesadores tienen ciclos de reloj en el rango de 30 a 400 nanosegundos.

Procesador	Ciclo/Reloj	Promedio Instrucciones (MIPS)
Accel	100 ns	3.2 MIPS
ARM	8 MHz	3-4 MIPS
AMD2900	125 ns	4-5 MIPS
CAP	I : 10 MHz II : 25 MHz	12.5 MIPS maximo, unidad escalar
Clipper	33 MHz	5 MIPS
CRISP	16 MHz	> 10 MIPS
Dragon	10 MHz	5 MIPS por CPU
MIPS	16.6 MHz	8 MIPS
Pyramid	125 ns	2-4 MIPS
Ridge 32	125 ns	1-4 MIPS
ROMP	170 ns	2 MIPS
Spectrum	30 MHz	10.8 MIPS
Transputer	50 ns	10 MIPS
Whetstone	50 ns	5-13.3 MIPS
MIPS-X	20 MHz	> 10 MIPS
CD GaAs	5 ns	91 MIPS
McD GaAs	10 ns	100 MIPS
RCA GaAs	200 MHz	200 MIPS maximo

**Fig. 8.5. - Performance de procesadores RISC.**

El dato que figura en la columna de Promedio de Instrucciones es, en la medida de lo posible, el promedio de tiempos de procesamiento de mediciones o benchmarks simulados, y no representa los valores pico.

Los 10 MIPS del Transputer pueden parecer engañosos, ya que se basan en el supuesto de que todas las instrucciones y operandos residen en el chip de RAM sin que existan demoras de acceso a memorias externas.

El segundo grupo de performance incluye los procesadores de arseniuro de galio.

Los tres procesadores fueron diseñados para un ciclo de reloj de 200 Megahertz. Los promedios de ciclo y de instrucciones son de una magnitud mayor que la de aquellas implementaciones basadas en el silicio.

Pocos de los procesadores vistos poseen cada una de las características atribuidas a los diseños RISC.

Muchos comparten alguna de las características de las CISC, agregando capacidad de procesamiento adicional para una dada aplicación.

De hecho, es interesante hacer notar que cada procesador de los analizados contiene características arquitecturales atribuidas típicamente a los CISC, y que cada característica de las CISC está representada en al me-

nos alguno de los diseños RISC, indicando que en el futuro los buenos diseños y conceptos arquitectónicos útiles sobrevivirán.

Es desde ya obvio que un compilador optimizado es una parte integral de cualquier diseño RISC.

El desarrollo de compiladores optimizados y reorganizadores va a la zaga muy a menudo de los nuevos desarrollos de hardware de cómputo.

## 8.2. - **CONTROVERSIA**

Existe una gran controversia cuando se quiere comparar computadoras con conjunto de instrucción reducido y computadoras con conjunto de instrucción complejo (RISC vs. CISC). Esta misma controversia puede dividirse en dos grandes categorías:

1º) Qué diferencia una RISC de una CISC ?

2º) Cómo puede uno hacer mediciones razonables y útiles de performance para compararlas ?

Muchas de las características de las RISC han sido muy utilizadas en computadoras CISC. Características tales como el pipeline de datos, memoria cache, y windowing de registros son vistos muy a menudo como atributos de un diseño RISC.

Originalmente los diseños RISC se realizaron apuntando a aplicaciones específicas y debido a eso fueron optimizados para la ejecución de una clase bien definida de programas.

De forma característica, las CISC fueron diseñadas para un amplio rango de aplicaciones y consiguientemente incluyen el soporte para muy diversos entornos de programación.

Muchas de las más populares técnicas de medición de performance son de un valor cuestionable cuando se trata de medir performance entre RISC y CISC. Típicamente, los efectos del overhead del sistema operativo, la optimización del compilador y los conjuntos de registros múltiples no son considerados en forma apropiada.

Los benchmarks relativos a la cantidad de transacciones de la aplicación por segundo tienen más significado que la simple medición de las instrucciones ejecutadas por segundo.

Actualmente aún existen fuertes disputas entre los defensores de RISC y CISC. De acuerdo al avance del mercado se detectan RISC sólo de nombre (ya que poseen más de 180 tipos diferentes de instrucciones) pero que mantienen la llamada a procedimientos desde hardware y lo realmente novedoso que es el sistema de ventanas.

Existen ya versiones que realizan operaciones de punto flotante en el llamado ciclo de máquina.

Se define el tiempo de ciclo de máquina como el tiempo que lleva leer y sumar el contenido de 2 registros y guardar el resultado en un tercero.

## 8.3 - **Una definición de las arquitecturas CISC**

Los criterios que siguen a continuación pueden utilizarse tanto para definir computadoras RISC como computadoras CISC.

Aquí los utilizaremos para las computadoras CISC, con las precisiones correspondientes en cada caso:

- 1) Cantidad de instrucciones en lenguaje máquina (en el caso de las CISC, tan grande como sea posible)
- 2) Cantidad de modos de direccionamiento (aquí también tan grande como sea posible)
- 3) Cantidad de formatos de instrucción (nuevamente, tan grande como sea posible)
- 4) Muchas instrucciones requieren más de un ciclo para su ejecución
- 5) Varios tipos de instrucciones tienen acceso a memoria (en el caso de los sistemas RISC las instrucciones LOAD/STORE son las únicas que tienen acceso a memoria)
- 6) Existencia de registros de propósito específico
- 7) Control microprogramado
- 8) Instrucciones de máquina de un relativo alto-nivel (cercano al alto nivel de las sentencias de los lenguajes de alto nivel)

Los criterios (1) y (2) pueden especificarse en valores numéricos, en tanto que los otros se especifican por Si o No.

Asumamos los siguientes requerimientos a efectos de cumplir los criterios (1) y (2):

- 1) la cantidad de instrucciones debe ser mayor a 100
- 2) los modos de direccionamiento son más de 4

La Fig. 8.6 presenta los ocho criterios para las siguientes arquitecturas CISC:

- El Motorola MC68020
- Intel 80386
- Clipper de Fairchild
- Zilog Z80000
- AT&T WE32100
- Focus de Hewlett-Packard
- Serie NS32000
- DEC VLSI de VAX

La Fig. 8.7 indica, para cada máquina en particular si los 8 criterios han sido satisfechos (S) o si han sido violados (N).



Según surge de las Fig. 8.6 y 8.7 muchos de los procesadores satisfacen muchos de los requerimientos y pueden ser caracterizados entonces como máquinas CISC.

SISTEMA	C r i t e r i o s							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Motorola 68020	109	16	Si	Si	Si	Si(16)	Si	Si
Intel 80386	111	8	Si	Si	Si	Si(6)	Si	Si
Clipper	101	9	Si	No	No	No(16)	No	No
Zilog Z80,000	110	9	Si	Si	Si	No(16)	Si	Si
AT&T WE32100	169	16	Si	Si	Si	No(16)	Si	Si
Focus de HP	230	10	Si	Si	Si	Si(28)	Si	Si
Series NS32000	86	14	Si	Si	Si	No(8)	Si	Si
DEC VLSI VAX	304	21	Si	Si	Si	Si(16)	Si	Si

\* Bajo ciertas condiciones los valores numéricos de esta tabla pueden variar

Fig. 8.6. - Ocho arquitecturas evaluadas en términos de los criterios que caracterizan a las computadoras CISC.

SISTEMA	C r i t e r i o s							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Motorola 68020	Si	Si	Si	Si	Si	Si	Si	Si
Intel 80386	Si	Si	Si	Si	Si	Si	Si	Si
Clipper	Si	Si	Si	No	No	No	No	No
Zilog Z80,000	Si	Si	Si	Si	Si	No	Si	Si
AT&T WE32100	Si	Si	Si	Si	Si	No	Si	Si
Focus de HP	Si	Si	Si	Si	Si	Si	Si	Si
Series NS32000	No	Si	Si	Si	Si	No	Si	Si
DEC VLSI VAX	Si	Si	Si	Si	Si	Si	Si	Si

Fig. 8.7. - Ocho arquitecturas evaluadas en términos de si satisfacen [Si] o violan [No] los criterios que caracterizan a las computadoras CISC.

La única excepción la constituye el Clipper de Fairchild, el cual combina características de máquinas CISC (gran cantidad de modos de direccionamiento y formatos de instrucciones) y características de máquinas RISC (incluyendo ejecución de instrucción en un solo ciclo, arquitectura LOAD/STORE y control de tipo hardwired).

Algunas de las características arquitecturales son erróneamente consideradas como típicamente de computadoras CISC; como por ejemplo, la gran cantidad de registros, el soporte para la administración de memoria y de memoria virtual, la existencia de una memoria cache, y una gran cantidad de transistores en el chip.

Estas características arquitecturales están reflejadas en la Fig. 8.8.

La columna (a) indica dónde está ubicada la unidad de administración de memoria, la columna (b) define dónde está ubicada la memoria cache, y la columna (c) especifica en forma conjunta la cantidad de transistores y la cantidad de chips utilizados en la construcción del procesador.

Los diseñadores de máquinas CISC y RISC, tienden a incorporar una gran cantidad de registros en sus máquinas.

Sin embargo, en el caso de las RISC tales registros tienden a ser de propósito general, en tanto que en el caso de las CISC una cierta cantidad de los registros son de propósito específico. Este hecho se ve reflejado en las Fig. 8.6 y 8.7.

Es cierto que las CISC están caracterizadas típicamente por una gran cantidad de transistores dentro del chip, a diferencia de las RISC. Sin embargo, una RISC con una gran cache dentro del chip puede tener incorporados una cantidad aún mayor de transistores.

La existencia de una memoria cache y del soporte para la administración de memoria y de memoria virtual son ambas, de una importancia comparable para las CISC y las RISC. Sin embargo, es cierto, que muchas (aun-

que no todas) las máquinas RISC proveen la administración de memoria fuera del chip en tanto que la mayoría de las CISC intentan incorporar al menos algunos de los elementos de la administración de memoria dentro del chip.

SISTEMA	Características de Arquitectura		
	(a)	(b)	(c)
Motorola 68020	Fuera del chip	Dentro del chip	190 K (1)
Intel 80386	Dentro del chip	Fuera del chip	275 K (1)
Clipper	Fuera del chip	Dentro del chip	(3) *
Zilog Z80,000	Dentro del chip	Dentro del chip	(1) *
AT&T WE32100	Fuera del chip	Dentro del chip	146 K (3)
Focus de HP	Fuera del chip	Dentro del chip	450 K (1)
Series NS32000	Dentro del chip	Dentro del chip	(1) *
DEC VLSI VAX	Dentro del chip	Fuera del chip	1.2 M (9)

**\* Datos no disponibles en la literatura**

**Fig. 8.8. - Ocho arquitecturas se evalúan aquí en términos de sus características arquitecturales.**

## **EJERCICIOS**

- 1) Cómo puede definirse la tecnología RISC y cuáles son sus características más comunes ?
- 2) Comente la controversia existente sobre las diferencias entre procesadores RISC y CISC.
- 3) Porqué la mayoría de los procesadores RISC tiene una lógica de decodificación de tipo hardwired ?
- 4) Verdadero o falso : Muchos procesadores RISC tienen unidades separadas para la ejecución concurrente de operaciones aritméticas y de E/S.
- 5) Qué es el windowing de registros en los procesadores RISC ?
- 6) Cómo se manejan las bifurcaciones en los procesadores RISC ? Comente cómo cree que se implementa.
- 7) Porqué existen coprocesadores en la mayoría de los procesadores RISC ?
- 8) Comente si la siguiente frase le parece o no razonable : "Dado un problema específico, es menos eficiente resolverlo con un procesador RISC que en una CISC de propósito general".
- 9) El uso de registros en los procesadores RISC es de uso general o específico ?
- 10) Los procesadores RISC tienen unidades de control microprogramadas o hardwired ? Justifique.
- 11) Porqué es usual que los procesadores RISC cuenten con pipelines de instrucción ?
- 12) Porqué resulta importante el desarrollo de los compiladores para máquinas RISC? Justifique.
- 13) Comente por lo menos tres criterios que usted utilizaría para diferenciar una computadora CISC de una RISC.
- 14) Cuál es la diferencia entre cómo se utilizan los registros internos de la CPU en computadoras RISC y CISC ? Utilizaría este concepto para diferenciar estas arquitecturas ?