Arquitectura de Computadoras

CURSO 2024

Turno:

Clase 7

Resumen clase 7

Procesadores RISC y CISC

- Hitos relevantes en la evolución de los sistemas de cómputo
- Procesadores CISC
- Estudios sobre el comportamiento de los programas
- Procesadores RISC
- Ventana de registros: concepto, implementación
- Optimización por software
- Ventajas y desventajs de los procesadores CISC
- Comparación entre procesadores RISC y CISC
- Ejemplos de procesadores RISC

Evolución histórica

Algunos hitos significativos en la evolución de los sistemas de cómputo

- Familia de computadoras: el concepto de familia de computadoras fué introducido por IBM a partir de su modelo System/360 en 1964, basado en la disponibilidad de un conjunto de máquinas con una arquitectura común y prestaciones variables de acuerdo a las necesidades.
- Arquitectura e implementación: son 2 conceptos separados que introdujo DEC en su máquina PDP-8, referidos a las características visibles al programador y las prestaciones del sistema.

Evolución histórica

Algunos hitos significativos en la evolución de los sistemas de cómputo

- 1964: Unidad de control microprogramada

Idea propuesta por Wilkes en 1951 para el desarrollo sistemático de las unidades de control, e introducida por IBM en la línea S/360.

- 1968: Memoria cache

Pequeña memoria de alta velocidad para balancear la capacidad de la CPU y la velocidad de transferencia de la Memoria principal. Introducida en la IBM S/360 Modelo 85.

- 1970: Memoria de estado sólido:

Desarrollo de memorias semiconductoras en reemplazo de las memorias magnéticas, lentas y complicadas de manejar.

Evolución histórica

Algunos hitos significativos en la evolución de los sistemas de cómputo

- 1971: i4004 primer microprocesador
- 1972: i8008
- 1974: i8080
- 1978: i8086

Segmentación de cauce: Introduce el paralelismo en la naturaleza secuencial de los programas.

Procesadores múltiples: desarrollo de sistemas con múltiples procesadores para aumentar la potencia de cómputo.

- 1980/1981/1982: primeras máquinas tipo RISC

Diseño conceptualmente opuesto a los procesadores CISC.

Procesadores CISC

Evolución de los procesadores

- El desarrollo de la tecnología de fabricación de circuitos integrados de muy alta escala de integración (VHSI) permitió incorporar cada vez mas funcionalidades dentro de un chip.
- La incorporación de nuevas funcionalidades se orientó principalmente en mejorar la eficiencia en la ejecución de programas.
- Un aspecto de esta mejora consistió en brindar un mayor soporte a los lenguajes de alto nivel (HLL), sobre todo los más complejos.

Procesadores CISC

Evolución de los procesadores

- Dar mayor soporte a los HLL tiende a facilitar el proceso de compilación (y también el trabajo de elaboración de compiladores), sobre todo considerando que el costo del "software es típicamente mucho más caro que el hardware".
- La "distancia" entre un HLL y el lenguaje de máquina se denomina GAP semántico, y básicamente es la relación entre la cantidad de sentencias de un programa en HLL y la cantidad de instrucciones de máquina que se requieren para resolverlo.
- El mayor soporte tiende a reducir el "GAP semántico", agregando instrucciones de máquina más complejas que resuelvan las sentencias del HLL más directamente.

Procesadores CISC

Procesadores CISC (Complex Instruction set computer)

- Los conceptos anteriores sentaron las bases de diseño de los procesadores “convencionales”:
  - Repertorios de instrucciones grandes.
  - Formato de instrucción complejo y de longitud variable.
  - Muchos modos de direccionamiento.
  - Capacidad de implementar sentencias de HLL con muy pocas instrucciones de máquina. Esta idea consiste básicamente en resolver una sentencia de HLL “por hardware”.
- De los conceptos anteriores surge el modelo de procesador conocido como **CISC**, es decir, procesador con repertorio de instrucciones complejo

Procesadores CISC

Procesadores CISC (Complex Instruction Set Computer)

- Con el uso masivo de la tecnología VHSI y el desarrollo de procesadores tipo CISC, se empezaron a realizar diferentes estudios sobre el comportamiento de los procesadores tipo CISC.
- Los estudios se orientaron a:
  - 1. Uso del repertorio de instrucciones
  - 2. Uso de los operandos
  - 3. Secuencia de ejecución de instrucciones
- Los estudios se realizaron de 2 maneras distintas:
  - Estáticos
  - Dinámicos

Procesadores CISC

Procesadores CISC (Complex Instruction set computer)

- Los estudios sobre operaciones estaban orientados a determinar el uso de las instrucciones y, por lo tanto, su interacción con la memoria.
- Los estudios sobre operandos básicamente intentaban determinar cuales eran los tipos de datos usados y su frecuencia de uso.
- Los estudios sobre el secuenciamiento de la ejecución de instrucciones tenían como objetivo determinar el impacto de disponer de un repertorio de instrucciones largo y complejo, en la Unidad de Control (que debe resolver las instrucciones), y en el cauce de datos que tiene relación con la eficiencia en la ejecución de una secuencia de instrucciones.

Procesadores CISC

Procesadores CISC (Complex Instruction set computer)

- Los estudios estáticos computaban cantidades basados en el código objeto ejecutable. Es una forma sencilla de obtener métricas, pero tienen el problema de que no tienen en cuenta el flujo del programa ejecutado, y pueden resultar bastante equivocados. Por ejemplo, la cantidad de instrucciones que componen un lazo no tiene nada que ver con las veces que se ejecuta.
- Los estudios dinámicos computan cantidades en base a la ejecución de diferentes programas (típicamente programas de prueba o “benchmarks”). Son más realistas porque los resultados tienen en cuenta la situación real por la que pasan los procesadores.

Procesadores CISC

Estudio sobre el uso de instrucciones

- Este estudio permitió medir el uso de las instrucciones, y su interacción con la memoria. Se efectuó sobre VAX, PDP-11 y MC68000.
- Las operaciones (instrucciones) se dividieron en:
  - Asignación: movimiento de datos, aritméticas y lógicas
  - Sentencias condicionales: IF, LOOP
  - Sentencias incondicionales: GOTO
  - Llamadas y retornos de subrutinas: CALL
  - Otras
- Los resultados obtenidos se pueden ver en la siguiente tabla.

Procesadores CISC

Estudio sobre el uso de instrucciones - Resultados

	Aparición dinámica		Instruc. máquina (Ponderadas)		Referencias a memoria (Ponderadas)	
	Pascal	C	Pascal	C	Pascal	C
Assign	45	38	13	13	14	15
Loop	5	3	42	32	33	26
Call	15	12	31	33	44	45
If	29	43	11	21	7	13
GoTo	-	3	-	-	-	-
Otras	6	1	3	1	2	1

Procesadores CISC

Estudio sobre el uso de instrucciones - Resultados

- En la tabla anterior se muestran los resultados para programas escritos en Pascal y C.
- Los resultados son medidos de 3 maneras (columnas):
  - Aparición dinámica: cuenta la cantidad de sentencias del HLL ejecutadas.
  - Instrucciones de máquina - ponderadas: computa las instrucciones de máquina para las sentencias del HLL ponderadas al tiempo de ejecución de las mismas.
  - Referencias a memoria - ponderadas: computa la cantidad de accesos a memoria.

Procesadores CISC

Estudio sobre el uso de instrucciones - Resultados

- Se puede apreciar del estudio mostrado que:
  - Las sentencias de asignación son las más numerosas, seguidas por las de llamadas a procedimientos y las sentencias condicionales (primer columna).
  - Cuando se hace la ponderación basado en la cantidad de instrucciones de máquina que resuelven las sentencias de alto nivel, las sentencias condicionales y llamadas a procedimientos son las que requieren más instrucciones de máquina (segunda columna).
  - Cuando se hace la ponderación basado en la cantidad de accesos a memoria, siguen predominando las sentencias condicionales y llamadas a procedimiento (tercer columna).

Procesadores CISC

Estudio sobre el uso de instrucciones - Resultados

- Los accesos a memoria tienen mucho impacto en el tiempo de ejecución del programa.
- Por lo tanto, los llamados a procedimiento consumen mucho tiempo y tiene que hacerse lo más eficiente posible para optimizar los accesos a memoria.

Procesadores CISC

Estudios sobre llamadas a procedimientos - Resultados

- En varios estudios se analizó el comportamiento en llamadas y retornos a procedimientos.
- En un alto porcentaje (>98%) se pasan menos de 6 datos.
- En un alto porcentaje (>92%) la cantidad de variables locales es menor a 6.
- En general el nivel de anidamiento es menor a 7.
- Conclusiones:
  - La mayoría de los programas tienen una secuencia corta de llamadas seguida por la secuencia de retornos.
  - La mayoría de las variables son locales.
  - Las referencias a operandos están muy localizadas.

Procesadores CISC

Estudios sobre uso de operandos - Resultados

La siguiente tabla muestra algunos resultados sobre el uso de operandos

	Pascal	C	Promedio
Constantes enteras	16	23	20
Variables escalares	58	53	55
Matrices/estructuras	26	24	25

Además, en otros estudios se observó:

- Uso de variables locales (80%) del procedimiento.
- Uso de punteros del tipo variable local y escalar para manejo de estructuras de datos.
- Uso intensivo de registros (no reflejado en este estudio).

Procesadores CISC

Conclusiones finales

Las conclusiones finales a las que se arribaron luego de muchos estudios fueron:

1) Optimización: se necesita optimizar el tiempo de ejecución de las instrucciones más usadas y las que consumen más tiempo.
2) Simplificación: de las instrucciones, para resolverlas más rápidamente.
3) Ajuste: del cauce de datos para resolver las instrucciones más usadas lo más rápido posible.
4) Registros: usar los registros en forma intensiva, para manejo de operandos. Disponer de un banco de registros significativo para reducir los accesos a memoria.

Procesadores RISC

- Los estudios realizados sobre el comportamiento de los programas, condujeron a un nuevo enfoque sobre el diseño de procesadores, a los que se identificó genéricamente como procesadores RISC (Reduced Instruction Set Computer).
- Los principales referentes de este nuevo enfoque fueron Patterson y Hennesey (“Arquitectura de computadoras – Un enfoque estructurado”).
- Conceptualmente, los primeros procesadores RISC surgieron a mediados de los 80 en las Universidades de Berkeley (David Patterson, proyecto RISC 1) y Stanford (John Hennesey, proyecto MIPS-XMP).

Procesadores RISC

- Los primeros RISC tenían, entre otras, las siguientes características relevantes:
  - Repertorio de instrucciones reducido y básico.
  - Formato de instrucción fijo y sencillo (todas las instrucciones tienen la misma cantidad de bits o bytes).
  - Unidad de control “cableada” (no microprogramada).
  - Diseño del cauce optimizado para ejecutar 1 instrucción por ciclo (uso preciso de la segmentación)
  - Máquina tipo LOAD-STORE: acceso a memoria solo con instrucciones de movimiento de datos. Instrucciones aritméticas y lógicas solo entre registros.
  - Uso intensivo de registros (de uso general) con optimización por software.

Procesadores RISC

Comparación entre CISC y RISC

En la tabla siguiente se muestra una comparativa entre 3 procesadores CISC y 2 procesadores RISC.

	Complex Instruction Set (CISC)Computer			Reduced Instruction Set (RISC) Computer	
Characteristic	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000
Year developed	1973	1978	1989	1987	1991
Number of instructions	208	303	235	69	94
Instruction size (bytes)	2–6	2–57	1–11	4	4
Addressing modes	4	22	11	1	1
Number of general-purpose registers	16	16	8	40 - 520	32
Control memory size (Kbits)	420	480	246	—	—
Cache size (KBytes)	64	64	8	32	128

Procesadores RISC

- En la tabla comparativa anterior se puede apreciar:
  - Repertorio de instrucciones: los CISC tienen muchas más instrucciones que los RISC.
  - Tamaño de instrucción: en los CISC es muy variable, mientras que en los RISC es fija.
  - Modos de direccionamiento: los CISC tiene una buena variedad de modos de direccionamiento, mientras que los RISC son mínimos.
  - Banco de registros: en los RISC son amplios, y más limitados en los CISC.
  - Memoria de control: típicamente microprogramada en los CISC, lógica “cableada” en los RISC.

Procesadores RISC - Ventana de registros

Registros en procesadores RISC

- Un aspecto esencial de los procesadores RISC es disponer de un banco de registros amplio, para reducir los accesos a memoria, y simplificar las instrucciones.
- La optimización del banco de registros se puede hacer de 2 maneras:
  - Por Hardware: agregando más registros
  - Por Software: optimizando la asignación de registros a las variables que se usen más en un período de tiempo dado.

Procesadores RISC - Ventana de registros

Solución por hardware de los registros en procesadores RISC

Tener un banco de registros amplio tiene ventajas y desventajas:

- Ventajas:
  - Reducción de accesos a memoria.
  - Disponibilidad para más variables escalares locales.
- Desventajas:
  - Más bits en la instrucción para identificar el registro.
  - En llamadas y retornos de subrutina hay que considerar que se requiere tiempo para pasar parámetros, asignar espacio para las variables locales, y devolver los resultados.

Procesadores RISC - Ventana de registros

Llamadas y retornos de subrutinas - Variables

- Por los estudios realizados, los llamados y retornos de subrutina son los que más tiempo consumen.
- Por lo tanto, es necesario hacer muy eficiente el manejo de este tipo de instrucciones.

Procesadores RISC - Ventana de registros

Llamadas y retornos de subrutinas – Variables

En general, cada subrutina necesita, para operar, 4 tipos de datos:

- Datos de entrada: argumentos de entrada
- Datos propios: datos (variables) locales
- Datos de salida: para pasar argumentos
- Datos globales (vistos por todas las subrutinas)

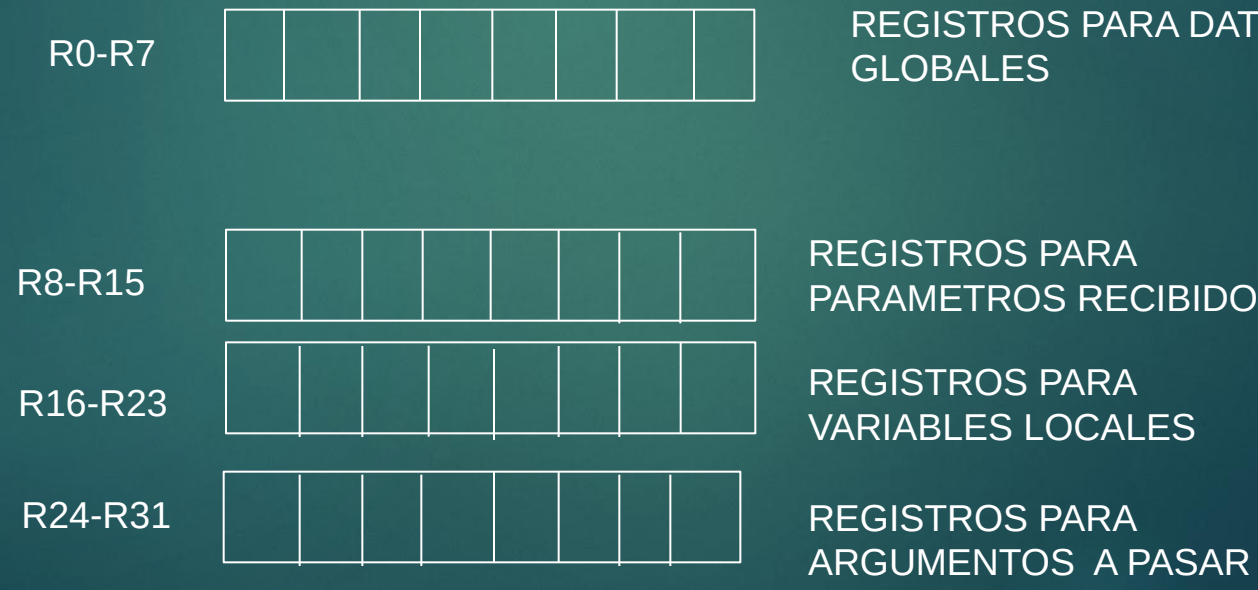
Los estudios realizados permiten inferir que:

- En general, las subrutinas usan pocas variables locales.
- Cada llamada recibe y pasa pocos argumentos.
- Las subrutinas usan pocas variables globales.

Procesadores RISC - Ventana de registros

Llamadas y retornos de subrutinas – Variables y registros

Si la subrutina usara registros para manipular todos los tipos de datos mencionados anteriormente, necesitaría disponer de la siguiente estructura de registros (asumiendo 8 variables por tipo de dato).



Procesadores RISC - Ventana de registros

Llamadas y retornos de subrutinas – Variables y registros

Cada tipo de dato requiere un banco de registros:

- Datos de entrada: registros de parámetros
- Datos propios: registros locales
- Datos de salida: registros temporales
- Datos globales: registros globales

Procesadores RISC - Ventana de registros

Llamadas y retornos de subrutinas – Variables y registros

- Si cada subrutina administra “su” banco de registros (que le “pertenece”), excepto los globales que son comunes a todos, la cantidad real de registros de cada subrutina se limita a un número no excesivamente grande (24 en el ejemplo anterior), y por lo tanto, requiere pocos bits para identificarlo (5 bits).
- Los registros globales son comunes a todas las subrutinas, y accesibles por todas ellas.

Procesadores RISC - Ventana de registros

Por otra parte, los problemas que aparecen con el uso de subrutinas son:

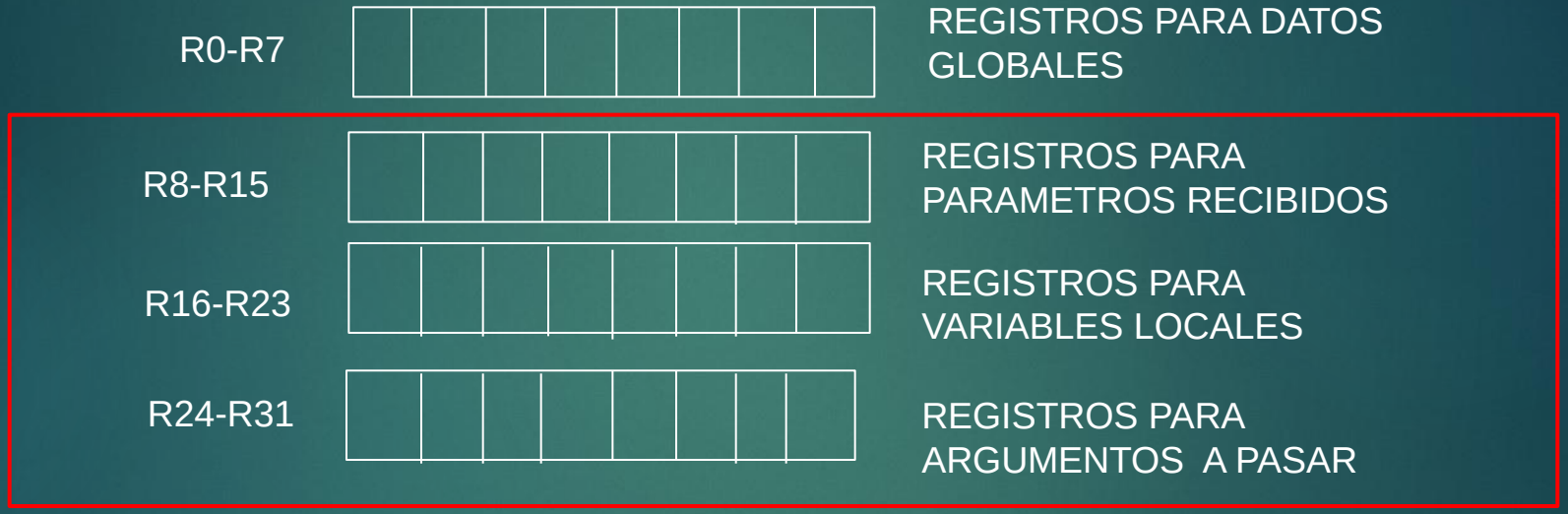
- Cada vez que una subrutina invoca otra subrutina (anidamiento de subrutinas) debe pasarle los argumentos, lo que produce un gasto de tiempo considerable.
- Cuanto más registros tenga la subrutina, más tiempo se consume en pasar la información.
- Cuantos mas registros necesite localmente, mas tiempo se requiere en reservarlos.

Para solucionar los problemas en el uso de Procedimientos y pasaje de parámetros se puede usar una estrategia conocida como Ventana de registros.

Procesadores RISC - Ventana de registros

La solución de usar una Ventana de registros consiste en:

1°) Asignar a cada subrutina un banco de registros propio:



Por ejemplo, los registros R8 a R31 son los registros accesibles por la subrutina.

Los registros R0 a R7 son globales y accesibles por todas las subrutinas.

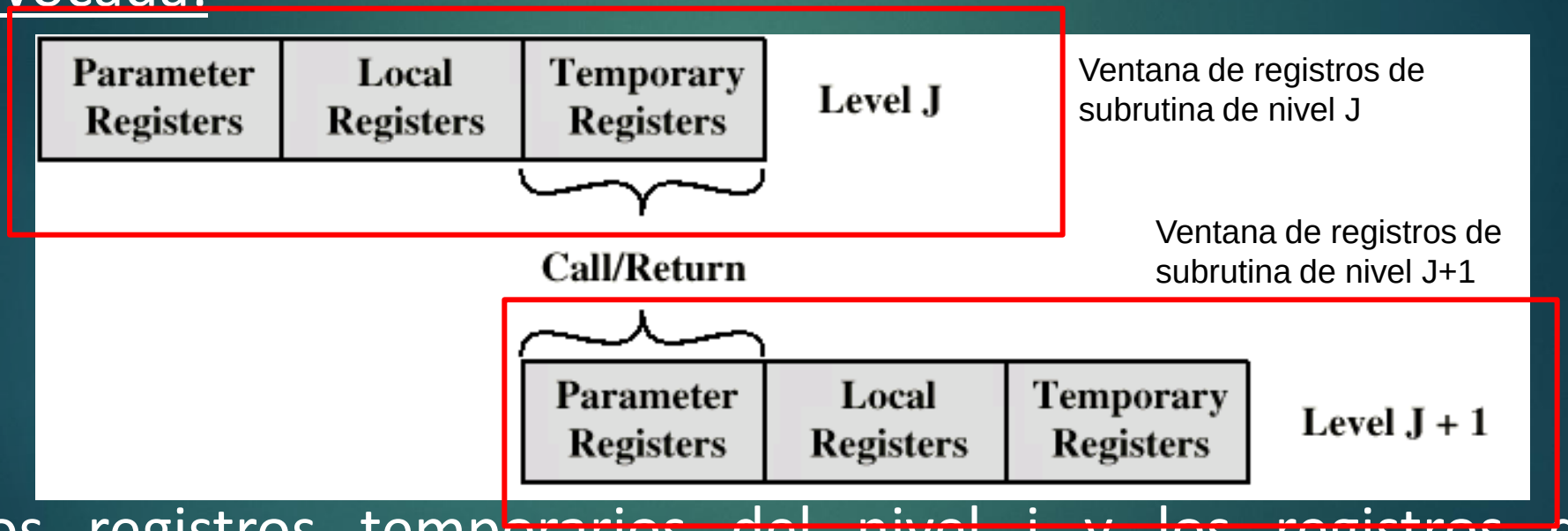
Procesadores RISC - Ventana de registros

2°) Superponer los registros donde recibe los parámetros una subrutina con los registros donde pasa los argumentos la subrutina que la llama. Por ejemplo:

- Una subrutina de nivel j invoca a una subrutina (nivel j+1).
- La subrutina de nivel j pasa los argumentos a la de nivel j+1 a través de sus registros “lógicos” R24 a R31.
- La subrutina de nivel j+1 recibe los parámetros de la subrutina de nivel j en sus registros “lógicos” R8 a R15.
- Si se usan los mismos registros “físicos”, el pasaje de parámetros se reduce a que la subrutina de nivel j escriba los argumentos en sus registros R24 a R31 y la de nivel j+1 los lea en sus registros R8 a R15. Esta situación se muestra a continuación.

Procesadores RISC - Ventana de registros

Esquema de solapamiento de los registros para pasar parámetros desde una subrutina j que llama, a otra j+1 que es invocada:



Los registros temporarios del nivel j y los registros de parámetros del nivel j+1 son físicamente los mismos.

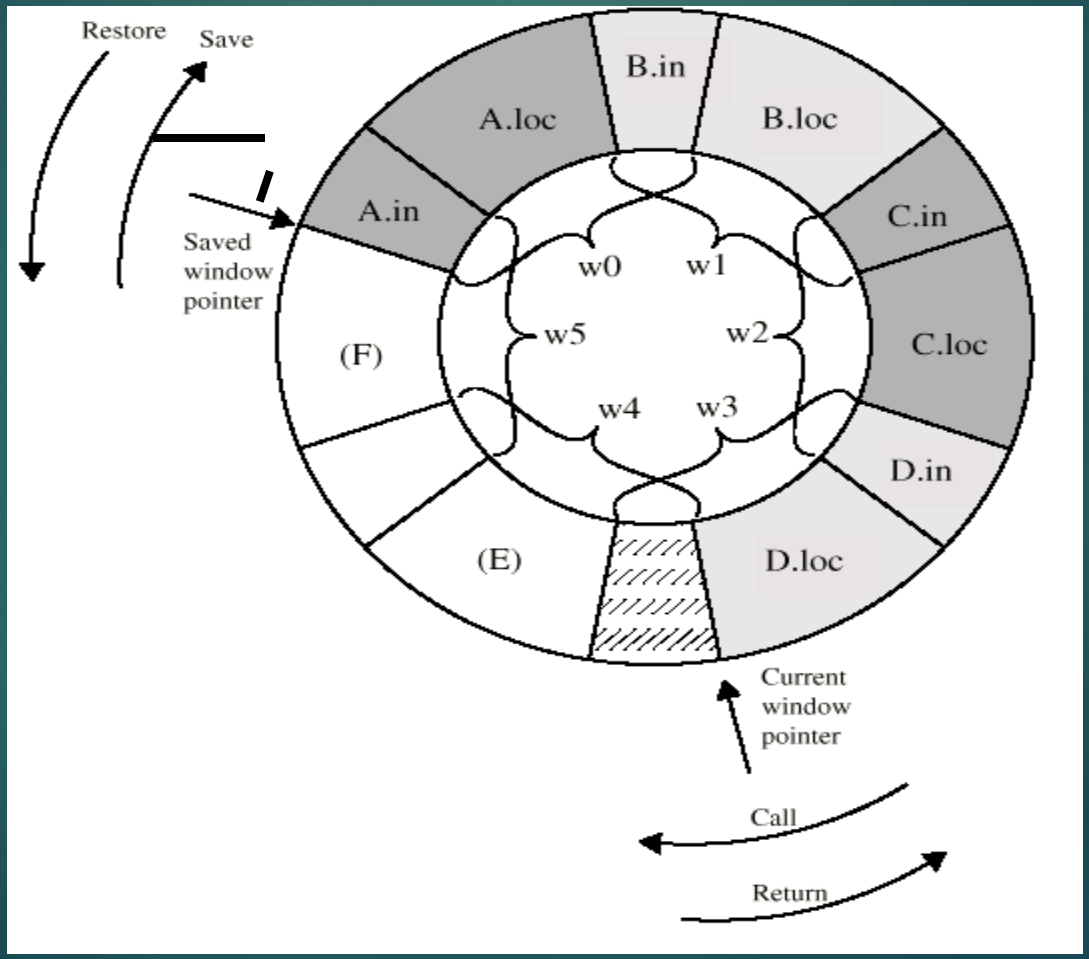
En cambio, los registros locales son físicamente distintos para cada subrutina y solo accesibles por ella.

Procesadores RISC - Ventana de registros

- De esta manera, cada subrutina accede a una ventana de registros, de los cuales los primeros y los últimos están solapados (superpuestos), y son usados para pasaje de parámetros.
- El banco de registros se implementa como un buffer de tipo circular, con una capacidad determinada de “ventanas”, basada en la “profundidad “ de anidamientos admitidas por el procesador (basada en los estudios antes analizados, podría ser, por ejemplo, del orden de 7)
- El esquema del banco de registros de ventanas solapadas se muestra en la figura siguiente.

Procesadores RISC - Ventana de registros

Implementación de la Ventana de registros con un buffer circular



Procesadores RISC - Ventana de registros

En la imagen anterior se muestra el buffer circular usado en un esquema de ventana de registros.

- En el nivel 0 (w0), los registros accesibles son A_{in}, A_{loc} y B_{in}.
- En el nivel 1 (w1), los registros accesibles son B_{in}, B_{loc} y C_{in}.
- Los registros B_{in} son accesibles por la rutina de nivel 0 (por donde pasa los parámetros) y la rutina de nivel 1 (por donde recibe los parámetros).
- Cada invocación de una subrutina mueve un puntero que apunta al siguiente banco de registros. El movimiento del puntero es tal que quedan superpuestos (solapados) los registros usados para pasaje de parámetros.
- La cantidad de registros accesibles por cada subrutina es la misma, pero cambian los registros físicos a los que accede.

Procesadores RISC - Ventana de registros

Variables globales

Para el manejo de variables globales (es decir compartidas por todas las subrutinas) se pueden usar 2 soluciones:

1. El compilador asigne posiciones de memoria a las variables: es algo ineficiente para variables globales a las que se accede frecuentemente porque requiere accesos a memoria que son inherentemente lentos.
2. Incorporar en el procesador un conjunto de registros para variables globales accesible en todos los niveles. Este banco de registros no está mostrado en la ventana de registros.

Procesadores RISC - Ventana de registros

Ventana de registros vs. Memoria cache

Banco de registros amplio	Cache
➢ Todos los datos son escalares y locales	➢ Datos escalares locales reciente-mente usados
➢ Acceso individual a variables	➢ Acceso a bloques de memoria
➢ Variables globales asignadas por el compilador	➢ Variables locales y globales usadas recientemente
➢ Salvaguarda/restauración basadas en la profundidad de anidamiento	➢ Salvaguarda/restauración basadas en el algoritmo de reemplazo
➢ Direccionamiento de registro	➢ Direccionamiento de memoria

Procesadores RISC - Técnicas por software

Técnicas de optimización por software

- También se puede mejorar las prestaciones en los procesadores RISC, optimizando el uso de los registros.
- Pero los lenguajes HLL no tienen referencias explícitas a los registros.
- Como la asignación de registros se hace en la compilación, el uso optimizado de registros es responsabilidad del compilador (no del programador).
- Las estrategias de optimización tratan de asignar los pocos registros a las variables más usadas o las que más tiempo permanecerán en el registro.

Procesadores RISC - Técnicas por software

Técnicas de optimización por software

- Una estrategia consiste en suponer que se disponen de infinitos registros (llamados “virtuales”). A cada variable del programa se le asigna un registro virtual (que son ilimitados).
- El compilador mapea (asigna) el número ilimitado de registros simbólicos a un número fijo de registros reales.
- Los registros simbólicos que no se solapan pueden compartir el registro real. Si se agotan los registros reales, algunas de las variables se asignan a posiciones de memoria.
- En la optimización se usa una técnica denominada ‘coloreado de grafos’.

Procesadores RISC

Características relevantes de los procesadores RISC

- Formatos de instrucción sencillos (fijo).
- Modos de direccionamiento sencillos.
- Diseño de la Unidad de control del tipo “cableado” (sin microcódigo).
- Operaciones registro a registro.
- Instrucciones a memoria únicamente LOAD y STORE.
- Una instrucción por ciclo (segmentación eficiente).
- Típicamente máquinas tipo Harvard.
- Mayor tiempo/esfuerzo de compilación.

Procesadores RISC y CISC

Ejemplos de procesadores RISC y CISC

Processor	Number of instruction sizes	Max instruction size in bytes	Number of addressing modes	Indirect addressing	Load/store combined with arithmetic	Max number of memory operands	Unaligned addressing allowed	Max Number of MMU uses	Number of bits for integer register specifier	Number of bits for FP register specifier
AMD29000	1	4	1	no	no	1	no	1	8	3[a]
MIPS R2000	1	4	1	no	no	1	no	1	5	4
SPARC	1	4	2	no	no	1	no	1	5	4
MC88000	1	4	3	no	no	1	no	1	5	4
HP PA	1	4	10[a]	no	no	1	no	1	5	4
IBM RT/PC	2[a]	4	1	no	no	1	no	1	4[a]	3[a]
IBM RS/6000	1	4	4	no	no	1	yes	1	5	5
Intel i860	1	4	4	no	no	1	no	1	5	4
IBM 3090	4	8	2[b]	no[b]	yes	2	yes	4	4	2
Intel 80486	12	12	15	no[b]	yes	2	yes	4	3	3
NSC 32016	21	21	23	yes	yes	2	yes	4	3	3
MC68040	11	22	44	yes	yes	2	yes	8	4	3
VAX	56	56	22	yes	yes	6	yes	24	4	0
Clipper	4[a]	8[a]	9[a]	no	no	1	0	2	4[a]	3[a]
Intel 80960	2[a]	8[a]	9[a]	no	no	1	yes[a]	—	5	3[a]

a RISC that does not conform to this characteristic.

b CISC that does not conform to this characteristic.

Procesadores RISC y CISC

Ventajas y desventajas de los procesadores CISC

- Ventaja: El compilador es mas sencillo por disponer de un repertorio amplio de instrucciones y modos de direccionamiento
- Desventaja: las instrucciones de máquina complejas son difíciles de aprovechar por el compilador, es decir las usa poco nada.
- Desventaja: la optimización es más difícil de realizar.
- Ventaja: los programas son más pequeños, tiene menos instrucciones, lo que probablemente implique que ocupa menos memoria. Sin embargo, la memoria hoy día es muy barata, por lo que esta ventaja es muy relativa.

Procesadores RISC y CISC

Ventajas y desventajas de los procesadores CISC

- El número de bits de memoria que ocupa no tiene porqué ser más pequeño al tener menos instrucciones
- Desventaja: al tener repertorio de instrucciones más amplio, los campos de código de operación son más largos y aumentan el tamaño de la instrucción.
- Las referencias a registros necesitan menos bits.

Procesadores RISC y CISC

Velocidad de ejecución

- El objetivo principal en el desarrollo de los procesadores es mejorar la velocidad de ejecución de los programas.
- Se supone que aumentar la complejidad del procesador debería mejorara su velocidad en ejecutar los programas.

Procesadores RISC y CISC

Velocidad de ejecución

- Pero:
  - Los procesadores CISC casi no usan las instrucciones más complejas.
  - El repertorio de instrucciones complejo exige una Unidad de control más compleja y lenta. En particular la memoria de control en la Unidad de control (microprogramada) es muy grande y “lenta”.
  - Una Unidad de control más lenta aumenta el tiempo de ejecución de las instrucciones simples, es decir, penaliza las instrucciones que podrían hacerse más rápido.
  - En definitiva, no está comprobado que la tendencia hacia CISC fuera la apropiada.

Procesadores RISC y CISC

Comparación entre RISC y CISC

- Se han realizado numerosas mediciones comparativas entre procesadores RISC y CISC.
- Pero las comparaciones tienen varios problemas:
  - No existe un par de máquinas RISC y CISC directamente comparables.
  - No hay un conjunto de programas de prueba definitivo.
  - EN los análisis, es muy difícil separar los efectos del hardware de los del compilador.
  - En muchos casos, las comparaciones se realizan usando prototipos o simulaciones en un “ambiente” controlado, y pocas veces con productos comerciales.
  - La mayoría de las máquinas son una mezcla de ambas.

Procesadores RISC y CISC

- Por otra parte es necesario definir qué parámetros se van a medir. En general, las evaluaciones pueden ser:
  - Cuantitativas: básicamente comparando los tamaños de los programas y su velocidad de ejecución
  - Cualitativa: revisión de soporte de lenguajes de alto nivel y uso óptimo de los recursos VLSI.

Conclusiones:

- No existe una marcada diferencia de performance entre uno y otro
- No está clara la barrera que separa uno u otro estilo.
- Muchos diseños incluyen características de ambos criterios, por ejemplo PowerPC y Pentium II.

Procesadores RISC

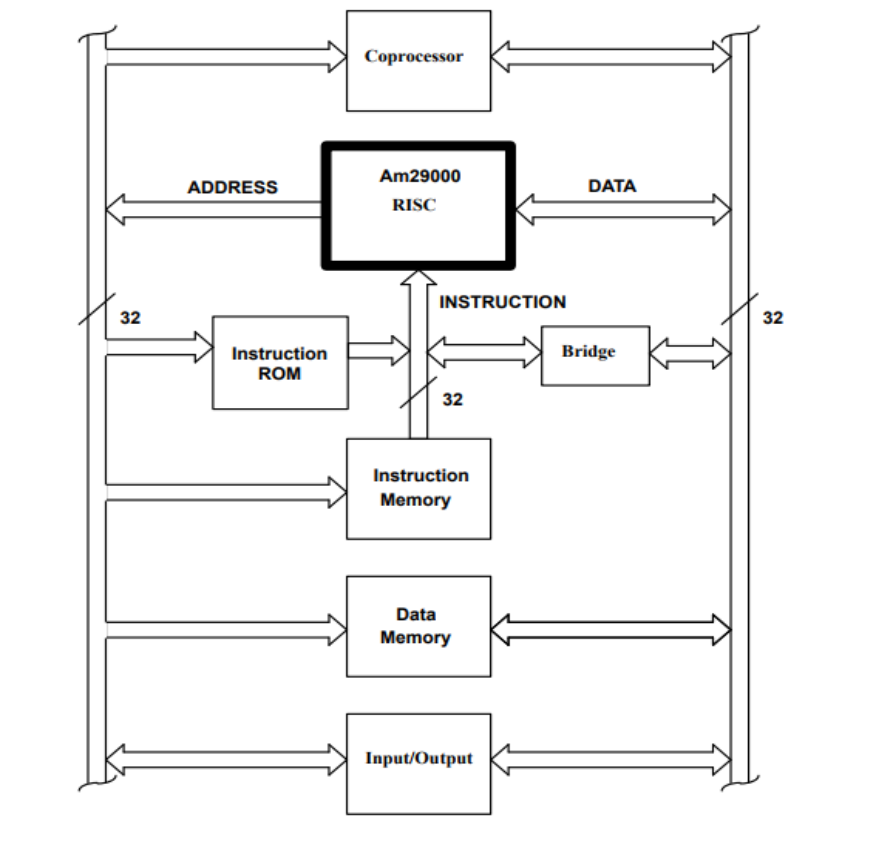
Algunos ejemplos comerciales

- SPARC: arquitectura tipo RISC abierta, diseñada por Sun microsystems y licenciada para Texas Inc, Cypress Sem., Fujitsu, etc. Uso de ventana de registros de con de 2 a 32 ventanas, y de 40 a 520 registros físicos.
- MIPS: arquitectura RISC desarrollada por MIPS Tech. Usado en sistemas SGI, y dispositivos tipo routers, videoconsolas, Play Station, etc. Varias revisiones (desde MIPS I hasta MIPS V, esta última de 64 bits)
- PA-RISC: diseñado por Hewelett Packard para estaciones gráficas.
- M88000: desarrollado por Motorola.
- I860: desarrollado por Intel en 1989. Fracasó por problemas técnicos, y de implementación (baja capacidad de cómputo).

Procesadores RISC

Procesador AMD 29000

- El AMD 29000 fué introducido en 1988. Tenía memorias de datos e instrucciones separadas (como el nanoMIPS), y un ancho de bus de 32bits.



Referencias

- Organización y Arquitectura de Computadoras, William Stallings, Capítulo 12, 5ta ed.
- Diseño y evaluación de arquitecturas de computadoras, M. Beltrán y A. Guzmán, Capítulo 1, 1er ed.
- Arquitectura de computadores – Un enfoque estructurado, Hennessy y Patterson.