**Computadores de repertorio reducido de instrucciones (RISC)**

* Los siguientes son uno de los principales avances desde el nacimiento del computador:

1. **El concepto de Familia:** introducido por la IBM en su System/360 en 1964 y seguido poco después por la DEC, en su PDP\_8. El concepto de familia separa la arquitectura de máquina de su implementación. Se ofrece un conjunto de computadores con distintas características en cuanto a precio/prestaciones, que presentan al usuario la misma arquitectura.
2. **Unidad de control microprogramada:** propuesta por Wilkes en 1951, e introducida por la IBM en la línea S/360 en 1964. La microprogramación facilita la tarea de diseñar e implementar la unidad de control y da soporte al concepto de familia.
3. **Memoria Caché:** introducida comercialmente por primera vez en el IBM S/360 Modelo 85 en 1968.
4. **Segmentación de cauce:** una manera de introducir paralelismo en la naturaleza esencialmente secuencial de una programa construido por la instrucción máquina. Dos ejemplos son la segmentación de instrucciones y el procesamiento vectorial.
5. **Múltiples procesadores.**
6. **Arquitectura de computador de repertorio reducido de instrucciones (RISC : Reduced Instruction set Computer).**

* Aunque los sistemas RISC se han definido y diseñados de diversas formas por parte diferentes grupos, los elementos principales compartidos por la mayoría de los diseños son:

1. Un gran número de registros de uso general, o el uso de tecnologías de compiladores para optimizar la utilización de los registros.
2. Un repertorio de instrucciones limitado y sencillo.
3. Un énfasis en la optimización de la segmentación de instrucciones.

**13.1 Características de la ejecución de instrucciones**

* El costo de hardware ha disminuido, el costo relativo del software ha ido creciendo. Otro elemento que se añade al coste y a otros inconvenientes es la falta de fiabilidad.
* Lenguajes de alto nivel (High-Level Languages HLL)
* Salto Semántico
* los supuestos síntomas de los saltos o huecos incluían:
* ineficiencia de la ejecución
* tamaño excesivo del programa en lenguajes de máquinas
* complejidad de los compiladores.
* Características principales:
* Repertorios de instrucciones grandes
* Docenas de modo de direccionamiento
* Determinadas sentencias HLL implemetandas en el hardware (ejemplo : la instrucción maquina CASE del VAX)
* Características de la ejecución de una instrucción : los aspectos cuyo calculo tiene interés son los siguientes :
* Operaciones realizadas
* Operando usados
* Secuenciamiento de la ejecución

**Operaciones**

**Operandos**

El estudio de Patterson indica que un candidato fundamental a optimizar es el mecanismo de almacenamiento y acceso a variables escalares locales.

**Llamadas a procedimientos**

* Hemos visto que las llamadas y retornos de procedimientos constituyen un aspecto importante de los programas en HLL, los datos de la tabla 13.2 indican que son las operaciones que consumen más tiempo en programas HLL
* Hay dos aspectos importantes

1. El número de parámetros y variables que manejan el procedimiento
2. La profundidad de anidamiento

**Consecuencias**

* Tres elementos que, por lo general, caracterizan a la arquitectura RICS
  1. Usar un gran número de registros o un compilador que optimice su tratamiento.
  2. Diseño de los cauces de instrucciones
  3. Repertorio simplificado (reducido) de instrucciones.

**13.2 Utilización de un amplio banco de registros**

* Se necesita una estrategia que permita que los operandos a los que se acceda con mayor frecuencia se encuentren en registros y se minimicen las operaciones, son posibles dos aproximaciones básicas :
  1. Basada en el software: consiste en confiar al computador la maximización del uso de los registros. El compilador intentara asignar registros a las variables que se usan más en un periodo de tiempo dado.
  2. Basada en el Hardware: consiste sencillamente en usar más registros que puedan mantenerse en ellos más variables durante periodos de tiempo largos.
* El primer producto RISC comercial: el Pyramid y se utiliza en la actualidad en la conocida arquitectura SPARC.

**Ventajas de los Registros**

* Los registros de parámetros contienen parámetros pasados al procedimiento actual desde el procedimiento que lo llamo, y los resultados a devolver a este. 76
* Los registros locales se unas para variables locales, según la asignación que realice el compilador
* Los registros temporales se usan para intercambiar parámetros y resultados con el siguiente nivel más bajo.

|  |  |  |
| --- | --- | --- |
| Registro de parámetros | Registros locales | Registros temporales |

Llamada/retorno

|  |  |  |
| --- | --- | --- |
| Registro de parámetros | Registros locales | Registros temporales |

Ventanas de registros solapadas

* La organización real del banco de registros es un buffer circulas de ventanas solapadas
* Puntero de ventana en curso (Curnt-Window Pointer, CWP): apunta a la ventana del procedimiento activo actualmente.
* El puntero de ventana Salvada (Saved-Window Pointer SWP): identifica la ventana guardada en memoria más recientemente.

**Variables Globales**

* La necesidad de almacenar las variables globales ue son las accedidas por mas de un procedimiento, se sugieren dos opciones :
  1. La primera es que el compilador asigne posiciones de memoria a las variables declaradas como globales en un HLL, y que todas las instrucciones máquina que referencien esas variables usen operadores referenciados en memoria.
  2. Una alternativa es incorporar al procesador un conjunto de registros globales.

**Un amplio banco de registros frente a una cache**

|  |  |
| --- | --- |
| **Banco de registro** | **Caché** |
| Todos los datos escalares locales | datos escalares locales recientemente usados |
| variables individuales | Bloques de memoria |
| variables globales asignadas por el compilador | Variables globales usadas recientemente |
| salvaguarda/restauración basadas en la profundidad de anidamiento | Salvaguarda/restauración basadas en el algoritmo de reemplazo |
| direccionamiento a registros | Direccionamiento a memoria |

Características de las organizaciones de un banco de registros amplio y de cache

* El banco de registros organizado en ventanas funciona como un buffer pequeño y rápido que contiene un subconjunto de todas las variables que probablemente se usen más

**13.3 optimización de registros basada en el compilador.**

* El objetivo del compilador es mantener en registro en lugar de en memoria las operandos necesarios para tantos cálculos como sea posible, y minimizar las operaciones de carga y almacenamiento.
* La técnica usada más comúnmente en los compiladores para RISC se conoce como coloreado de grafos.

**13.4 Arquitectura de repertorio reducido de instrucciones**

**¿Por qué CISC?**

* Hemos mencionado la tendencia hacia repertorios de instrucciones más ricos, que incluyen un número mayor de instrucciones e instrucciones más complejas. Estas tendencias ha sido motivada por dos razones principales:
  1. El deseo de simplificar los compiladores
  2. El deseo de mejorar las prestaciones
* La primera de las razones citadas, la simplificación de los compiladores
* La otra razón importante mencionada es la esperanza de que los CISC produzca programas más pequeños y rápidos.

**Características de las arquitecturas de repertorio reducido de instrucciones**

* Aunque ha habido diferentes aproximaciones a la arquitectura de repertorio reducido de instrucciones, hay ciertas características comunes a todas ellas :
* Una instrucción por ciclo: se ejecuta una instrucción de maquina cada ciclo de máquina.
* Operaciones registro a registro: la mayoría de las operaciones deben ser del tipo registro a registro con la excepción de sencillas operaciones LOAD y STORE para acceder a memoria.
* Modos de direccionamiento sencillos: casi todas las instrucciones RISC usan direccionamiento sencillo a registro.
* Formatos de instrucción sencillos.
* Un ciclo de maquina se define como el tiempo que se tarda en captar dos operandos desde dos registros. Realizar una operación de la ALU y almacenar el resultado en un registro.

**Características CISC frente a RISC**

* Tras el enunciado inicial por las maquinas RICS, ha habido una creciente convicción de que
  1. Los diseñadores RISC pueden sacar provecho de la inclusión de algunas características CISC
  2. Los diseños, CISC pueden sacar provecho de la inclusión de algunas características RISC.
* para el objetivo de esta comparación, se considera típico de un RISC clásico lo siguiente

1. un único tamaño de instrucción
2. ese tamaño es típicamente cuatro byte
3. un pequeño número de modos de direccionamiento de datos, típicamente menor que cinco.
4. No se usa direccionamiento indirecto que quiera efectuar un acceso a memoria para conseguir la dirección de memoria de otro operando.
5. No hay operaciones que combinen carga/almacenamiento con cálculos aritméticos (suma desde memoria, suma a memoria)
6. No se direcciona más de un operando de memoria por instrucción
7. Las operaciones de carga/almacenamiento no admiten una alienación de daros arbitraria
8. Un número máximo de usos de la unidad de gestión de memoria (Memory Managenent unit MMU) de una dirección de datos en cada instrucción
9. El número de bits de un campo designados de registros entero es de cinco o mas.
10. el número de bits de un campo designador de registro de coma flotante es de cuatro o más.

**13.5 Segmentación en RISC**

**Segmentación con instrucciones regulares**

* un ciclo de instrucción tiene las dos fases siguientes :
  1. I: captación de instrucción
  2. E: ejecución.
* Las operaciones de carga y almacenamiento necesitan tres fases:

1. I: captación de instrucción
2. E:ejecución
3. D:memoria

* Hay dos problemas que impiden que se consiga la máxima aceleración
  1. El primero es que suponemos que se usa una memoria de un único puerto y que solo es posible un acceso a memoria en cada etapa
  2. El segundo problema consiste en que una instrucción de salto interrumpe el flujo secuencial de ejecución.

**Optimización de la segmentación**

* El salto retardado, que es una forma de incrementar la eficiencia de la segmentación. Utiliza un salto que no tiene ligar hasta después de que se ejecute la siguiente instrucción (de ahí el termino retardado )
* La posición de la instrucción situada inmediatamente después de la instrucción de salto se conoce como espacio retardado.

**13.6 MIPS R4000**

**13.7 SPARC**

**13.8 La Controversia entre RISC y CISC**

* Durante muchos año la tendencia general en la arquitectura y organización de computadores ha sido incrementar la complejidad del procesador:
  1. Mas instrucciones
  2. Mas modos de direccionamiento
  3. Mas registros especializados
* El trabajo que se ha hecho para evaluar las ventajas de la aproximación RISC puede agrupar en dos categorías:
  1. Cuantitativa: intentos de comprar el tamaño de los programas y su velocidad de ejecución en máquinas RISC y CISC de similar tecnología.
  2. Revisión de asuntos tales como el soporte de lenguajes de alto nivel y el uso óptimo de los recursos VLSI.
* Cuando se intentan realizar tales comparaciones surgen varios problemas
  1. No hay una pareja de máquinas RISC y CISC que sean comprables en cuanto a coste del ciclo de vida, nivel de tecnología, complejidad a nivel de puertas, sofisticación del compilador, soporte para el sistema operativo.
  2. No existe un conjunto de programa de prueba definitivo. Las prestaciones varían según el programa.
  3. Es difícil separar los efectos del hardware de los efectos debidos a la habilidad en el diseño de compilador.
  4. La mayor parte de los análisis comparativos con RISC se han hecho con máquinas <<de juguete>> en vez de con productos comerciales.