RISC vs. CISC

* Describa las características que diferencian a los procesadores RISC respecto de los CISC

# RISC

RISC (Reduced Instruction Set Computer): Repertorio reducido de instrucciones

* Repertorio de instrucciones limitado y con un formato fijo (sencillo)
* Número grande de registros o la utilización de un compilador que optimice el uso de éstos
* Énfasis en la optimización del cauce de instrucciones

RISC se presta a una segmentación eficiente, porque hay menos operaciones llevadas a cabo por instrucción y éstas son más previsibles. También se presta a la técnica de salto retardado, en la cual las instrucciones de salto se reubican entre otras instrucciones para mejorar la eficiencia del cauce.

## Características del RISC

* Una instrucción por ciclo: Se ejecuta una instrucción máquina cada ciclo máquina. Con instrucciones sencillas y de un ciclo, hay poca necesidad de microcódigo y las máquinas pueden estar cableadas (significa que no hay que acceder a la memoria de control de microprograma durante la ejecución de la instrucción)
* Operaciones registro a registro: La mayoría de las operaciones son así, lo que simplifica el repertorio de instrucciones y fomenta la optimización del uso de los registros (los operandos accedidos frecuentemente permanecen en el almacenamiento de alta velocidad)
* Modos de direccionamiento sencillos: se usa principalmente el direccionamiento a registro
* Formatos de instrucción sencillos: Se usan pocos formatos, la longitud de las instrucciones es fija y alineada en los límites de una palabra. Las posiciones de los campos, especialmente la del código de operación, son fijas. Todo esto simplifica la unidad de control.
  + Formato de instrucción fijo: como ventaja, la decodificación del código de operación y el acceso a los operandos en registros puede hacerse de forma simultánea

Pero también: Requiere mayor tiempo/esfuerzo de compilación

Se considera RISC a:

* Un único tamaño de instrucción (típicamente de 4 bytes)
* Pocos modos de direccionamiento de datos, normalmente menos que 5
* No se usa direccionamiento indirecto que requiera efectuar un acceso a memoria
* No hay operandos que combinen carga/almacenamiento con cálculos aritméticos (ej: suma desde o a memoria)
* No se direcciona más de un operando

# CISC

CISC (Complex Instruction Set Computer): Repertorio complejo de instrucciones

* Repertorio de instrucción más rico, con un número mayor de instrucciones y más complejas.

## Finalidad del CISC

* Facilitar el trabajo del escritor de compiladores
* Mejorar la eficiencia de la ejecución: secuencias complejas de operaciones en microcódigo
* Dar soporte a HLL más complejos

## Inconvenientes del CISC

* El software es mucho más caro que el hardware
* El nivel del lenguaje era cada vez más complicado
* Salto semántico: Diferencias entre operaciones HLL y operaciones de la arquitectura
* Todo esto conduce a:
  + Repertorios de instrucciones grandes
  + Más modos de direccionamiento
  + Varias sentencias de HLL implementadas en el hardware (Por ejemplo, el CASE del VAX)

## ¿Simplificación del compilador?

Las instrucciones máquina complejas son con frecuencia difíciles de aprovechar, ya que el compilador debe descubrir aquellos casos que se ajustan perfectamente a la construcción. Optimizar el código generado para minimizar su tamaño, reducir el número de instrucciones ejecutadas y mejorar la segmentación, es mucho más difícil con un repertorio complejo de instrucciones (CISC).

## ¿Programas más pequeños?

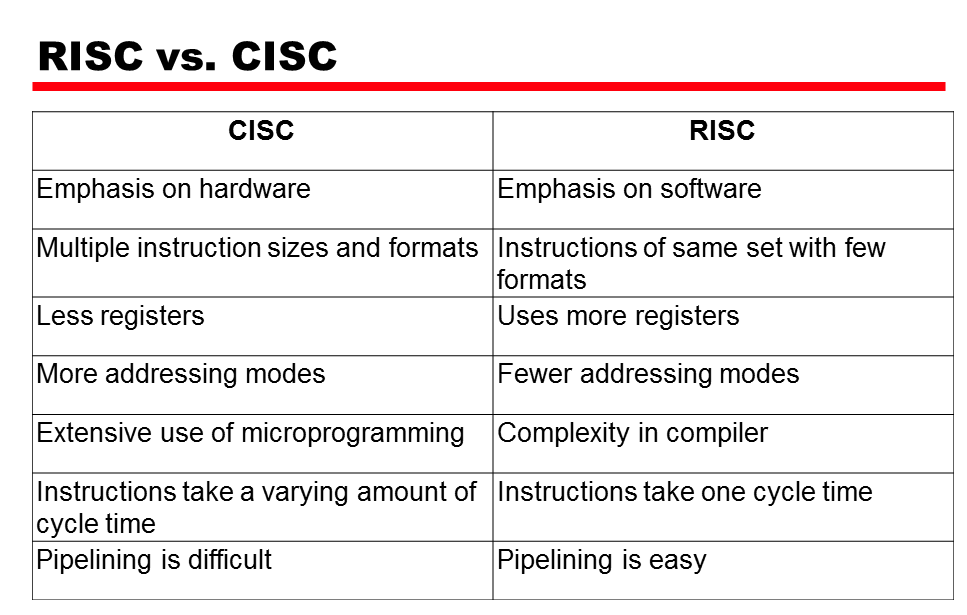
CISC busca que los programas sean más pequeños (ocupan menos memoria) y rápidos. Pero la memoria hoy en día es barata, por lo que no es una gran ventaja.

Si bien un programa para un CISC expresado en lenguaje máquina simbólico puede ser más corto (tiene menos instrucciones) que el RISC, el número de bits de memoria que ocupa no tiene por qué ser más pequeño. Al haber más instrucciones en un CISC, necesitan códigos de operación más largos (produciendo operaciones más largas). Además, RISC acentúa las referencias a registros en lugar de a memoria y los mismos necesitan menos bits.

## ¿Programas más rápidos?

CISC al ser propenso a usar instrucciones más sencillas, requiere de una Unidad de Control más compleja y/o la memoria de control del microprograma debe hacerse más grande, de modo que aumenta el tiempo de ejecución de las instrucciones simples.

# RISC frente a CISC



* No existe una clara barrera diferenciadora
* Muchos diseños incluyen características de ambos criterios (CISC y RISC)
  + Por ejemplo: PowerPC y Pentium II

## Controversia RISC y CISC

El trabajo que se ha hecho para evaluar las ventajas de la aproximación RISC se pueden agrupar en dos categorías:

* Cuantitativa: Intentos de comprar el tamaño de los programas y su velocidad de ejecución en máquinas RISC y CISC de similar tecnología
* Cualitativa: Revisión de asuntos tales como soporte de lenguajes de alto nivel y uso óptimo de recursos VLSI

## Problemas de las comparaciones:

* No existe un par de máquinas RISC y CISC directamente comparables
* No hay un conjunto de programas de prueba definitivo
* Es difícil separar los efectos del hardware de los del compilador
* La mayoría de los análisis comparativos se han hecho con máquinas de “juguete”, no con productos comerciales
* La mayoría de las máquinas son una mezcla de ambas

**Fuentes**:

* “Clase 06”
* “Capítulo 12 - Computadores de repertorio reducido de instrucciones” (Stalling 5ta ed. Págs 438, 451, 455 y 473)