



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**Estructura y Programación de Computadoras**

*Grupo: 02 - Semestre: 2021-1*

**Tarea 1:  
Instruction Set Architecture**

**FECHA DE ENTREGA: 04/10/2020**

**Alumno:**  
**Téllez González Jorge Luis**



# Índice

<b>1. El conjunto de instrucciones - ISA</b>	<b>2</b>
1.1. RISC . . . . .	3
1.2. CISC . . . . .	4
1.3. CISC vs RISC . . . . .	5

# 1. El conjunto de instrucciones - ISA

El concepto de *instrucciones* se refiere de forma general a las diferentes acciones que pueden ser ejecutadas por una CPU, las cuales pueden ser operaciones aritméticas con diferentes tipos de datos (enteros, de punto flotantes, vectoriales), operaciones lógicas, de movimiento de datos, entre otras.

La mayoría de los programas que se escriben en lenguajes de alto nivel como C++, Java o Pascal deben de traducirse a un lenguaje que sea entendible para el procesador; diferente para cada familia de procesadores. *El conjunto de instrucciones que forman este lenguaje se denomina juego de instrucciones o repertorio de instrucciones.*



Figura 1: *El conjunto de instrucciones es la puerta de la comunicación CPU-Programador.*

Los conjuntos de instrucciones de una computadora deben poseer idealmente una serie de propiedades que se detallan a continuación:

1. Este conjunto debe de ser **completo**, tal que pueda escribirse un programa que permita evaluar una función computable utilizando una cantidad de memoria razonable y, además, emplear una cantidad de tiempo moderada. Por tanto, el número de instrucciones para ejecutar tal programa no debe ser demasiado elevado.
2. Deben ser eficientes, en el sentido de que las funciones más esenciales deben de poder realizarse en pocas instrucciones.
3. El conjunto de instrucciones debe de ser **regular**, es decir, debe de ser **simétrico** tal que, si existe una instrucción de desplazamiento a la izquierda, exista otra que permita desplazarse a la derecha. Además, debe de ser **ortogonal**, lo cual implica que deben de poder combinarse en la medida de lo posible todas las operaciones con todos los tipos de datos y modos de direccionamiento.
4. Es deseable que el conjunto de instrucciones de una CPU sea retrocompatible con modelos anteriores.
5. Cada una de las instrucciones debe de ser independiente de las otras y cada una de ellas debe de contener toda la información necesaria para ejecutarse por sí mismas.



Este concepto es conocido de forma general como *Instruction Set Architecture* o **ISA** y permite conocer el tipo de instrucciones que una CPU es capaz de reconocer y ejecutar. Los equipos de cómputo domésticos que utilizan procesadores de Intel o AMD tienen como norma general un **ISA x86** basado en las especificaciones del *Intel 8086*, mientras que en los equipos de cómputo móvil que utilizan procesadores de Apple, Huawei, MediaTek, Samsung y Qualcomm se encuentran basados en **el ISA ARM**.

## 1.1. RISC

Las máquinas que utilizan un *Reduced Instruction Set Computer* se enfocan principalmente en *mejorar en lo posible los casos más frecuentes*, es decir, enfocarse en el uso de las instrucciones de uso más frecuente en un determinado entorno (gestión, diseño asistido, cómputo científico, etc). Este enfoque se caracteriza por los siguientes puntos:

- Instrucciones sencillas de un solo ciclo.
- Pocas instrucciones y direccionamientos.
- Elevado número de registros en el procesador para las variables de los procedimientos.
- Acceso restringido a memoria; únicamente accesible a instrucciones de carga y almacenamiento (**LOAD y STORE**).
- Su complejidad reside en el compilador, pues utilizan compiladores optimizados a las características anteriores y a las propias características de la máquina en la que se ejecutan.

La ejecución de instrucciones en un solo ciclo de máquina es una de las propiedades más importantes de las máquinas RISC, pues permite implementar el uso de paralelismo *pipeline* en la ejecución de instrucciones, reduciendo los accesos continuos a la memoria. Como consecuencia de lo anterior, **RISC** tiene las siguientes ventajas:

- La CPU trabaja de forma más rápida al utilizar un menor número de ciclos de reloj para la ejecución de instrucciones.
- Utiliza un sistema de direcciones no destructivas en RAM, lo cual reduce la ejecución de nuevas operaciones.
- Cada instrucción puede ser ejecutada únicamente en un solo ciclo de máquina, como se mencionó previamente.

Los diseños basados en **RISC** prefieren el uso de un modelo de memoria *Harvard*, donde los conjuntos de instrucciones y los conjuntos de datos se encuentran conceptualmente separados. Esto permite la implementación eficaz de la paralelización en la ejecución de instrucciones y, en ciertos casos, una mejora sustancial de rendimiento de acuerdo a las instrucciones realizadas.



Figura 2: Consola de la supercomputadora CDC 6600, considerada la precursora del diseño RISC.

## 1.2. CISC

A diferencia del enfoque anterior, las computadoras que utilizan un *Complex Instruction Set Computer* implementan un conjunto amplio de instrucciones lentas de ejecutar. Tal amplitud permite operaciones complejas entre operandos situados en memoria o en registros internos, a diferencia de la arquitectura **RISC**.

Este tipo de arquitectura dificulta el uso del paralelismo entre instrucciones. Con el fin de mitigar tal deficiencia, la mayoría de equipos CISC de alto desempeño implementan un sistema que convierte dichas instrucciones complejas en varias instrucciones simples de tipo *RISC*, las cuales suelen denominarse como *microinstrucciones*. Las instrucciones compuestas son decodificadas internamente y ejecutadas con una serie de microinstrucciones almacenadas en una memoria ROM interna. Para cada microinstrucción, se requiere al menos de un ciclo de reloj.

Entre las características de este enfoque se encuentran:

- El uso extensivo de instrucciones complejas que requieren varios ciclos de ejecución.
- Hacen uso de muchas instrucciones complejas y direccionamientos.
- Cualquier instrucción puede acceder libremente a la memoria.

Los diseños **CISC** generalmente se encuentran en los procesadores más antiguos, como el Motorola 68000, el Zilog Z80 y, principalmente, la familia de procesadores Intel x86 que actualmente predomina el segmento de la computación de uso personal.

A pesar de las debilidades de este enfoque frente al **RISC**, las máquinas **CISC** han logrado perdurar de la mano de Intel aplicando ciertas ventajas del **RISC** para potenciar el diseño de sus arquitecturas, y si bien, este enfoque híbrido no consigue los mismos resultados que un diseño completamente puro, ofrece un rendimiento global competitivo y además, brinda una ventaja que resulta crítica en varios sistemas: **permite la ejecución de software viejo sin mayores modificaciones**.

Este enfoque también posee fortalezas a destacar, como son las siguientes:

- Reduce la dificultad de crear compiladores que estén adaptados para cada sistema en específico.
- Permite reducir costos en la construcción de un sistema computacional.
- Facilita la depuración de errores.

### 1.3. CISC vs RISC

La tendencia actual favorece al uso del conjunto de instrucciones basadas en **RISC** debido al alto desempeño que pueden lograr, sin embargo, el uso del **CISC** sigue vigente por motivos económicos y de retrocompatibilidad. Sin embargo, exponentes del diseño **RISC** como **ARM** han logrado importantes hitos respecto a su eficiencia energética y costo de producción.



Figura 3: Cada conjunto ofrece tanto ventajas como desventajas en su implementación.

Tras el nacimiento del **RISC**, muchos partidarios de este diseño aseguraban que la mejor forma de diseñar una computadora era tener un número reducido de instrucciones simples. Uno de sus argumentos era que, incluso si una máquina **RISC** requería cuatro o cinco instrucciones para hacer lo que una máquina **CISC** realiza una sola instrucción, si las instrucciones **RISC** eran 10 veces más rápidas de ejecutar, el ganador era evidente.

La batalla entre ambos enfoques continúa librándose y, al final, el propósito del sistema a desarrollar será uno de los factores decisivos al momento de elegir **el ISA adecuado** para el mismo.

## Referencias

- [1] Arquitectura CISC vs RISC. Recuperado de: <https://is603arquicom2016.wordpress.com/arquitectura-cisc-vs-risc/>. Fecha de consulta: 03/10/2020.
- [2] Arquitectura del Conjunto de Instrucciones del microprocesador. Recuperado de: <https://www.ticarte.com/contenido/>



- arquitectura-del-conjunto-de-instrucciones-del-microprocesador. Fecha de consulta: 03/10/2020.
- [3] Arquitectura RISC vs CISC. Recuperado de: [http://triton.astroscu.unam.mx/fruiz/introduccion/introduccion\\_computacion/Arquitectura%20RISC%20vs%20CISC.pdf](http://triton.astroscu.unam.mx/fruiz/introduccion/introduccion_computacion/Arquitectura%20RISC%20vs%20CISC.pdf). Fecha de consulta: 03/10/2020.
- [4] Conjuntos de instrucciones. Recuperado de: <https://www.infor.uva.es/~bastida/OC/conjunto.pdf>. Fecha de consulta: 03/10/2020.
- [5] Juego de instrucciones. Recuperado de: [http://cv.uoc.edu/annotation/8255a8c320f60c2bfd6c9f2ce11b2e7f/619469/PID\\_00218277/PID\\_00218277.html#w31aab7b9b9](http://cv.uoc.edu/annotation/8255a8c320f60c2bfd6c9f2ce11b2e7f/619469/PID_00218277/PID_00218277.html#w31aab7b9b9). Fecha de consulta: 03/10/2020.
- [6] RISC. Recuperado de: <https://esacademic.com/dic.nsf/eswiki/287472>. Fecha de consulta: 03/10/2020.
- [7] Tanerbaum, A. S. (2000). *Organización de computadoras: un enfoque estructurado*. Prentice Hall, 4th edition.

Los créditos de las fotografías pertenecen a sus respectivos autores. ©