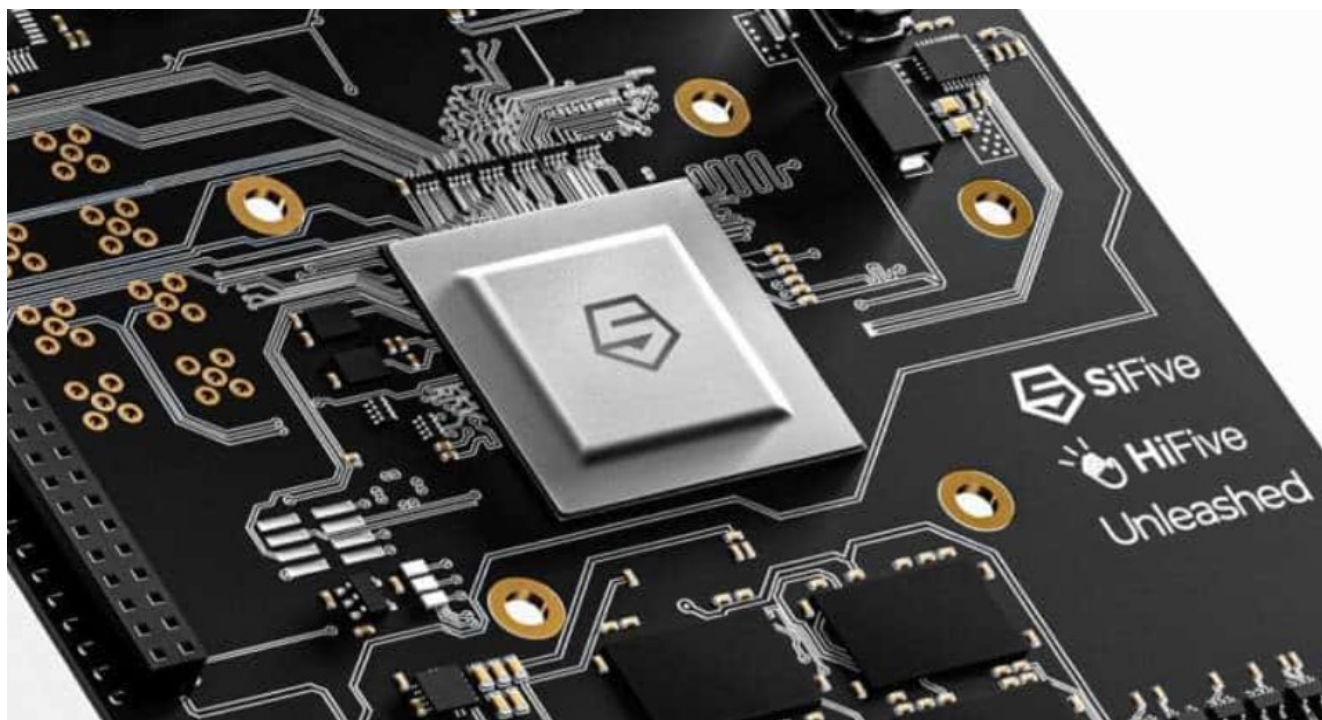


RISC-V, todo lo que debes saber de la alternativa libre que desafía a ARM y x86



Cuando hablamos de un **conjunto de instrucciones**, realmente estamos hablando del **idioma** que habla un procesador y lo que entiende al leer su código binario.

A lo largo de la historia de la informática ha habido varios sets de instrucciones de arquitectura o **ISA**, muchos de ellos tuvieron su momento de gloria y desaparecieron, pero en un mundo dominado por **x86** y **ARM** en todo tipo de dispositivos **RISC-V** empieza a despuntar como alternativa.

El conjunto de instrucciones **RISC-V** promete ser una revolución, en especial por el hecho que en muchos mercados relacionados con el hardware no solo permitirá el despliegue rápido de soluciones específicas, tanto para problemas generales o concretos, sino que además permite no tener que depender ni de **ARM** ni **x86**. En este artículo os hablaremos de las características de este conjunto de instrucciones y en qué

se diferencia del resto.

¿Qué es RISC-V?

Bajo el nombre de **RISC-V** se conoce a un conjunto de registros e instrucciones totalmente **libre** y que no tiene propiedad asignada a una empresa en concreto, lo cual tiene la ventaja de que se pueden crear no solo CPU con dicha **ISA** sin pagar a un tercero. Esto además permite modificarlo para las necesidades de diseños en concreto, es decir, se pueden crear chips con versiones adaptadas y recortadas de **RISC-V** para propósitos específicos como son **GPU** o **NPU**, por lo que en ese punto es muy parecido al concepto de software libre, pero de cara al hardware.

Por otro lado, **RISC-V** no define un tipo de núcleo, sino lo que hace cada instrucción, por lo que la implementación del set de registros e instrucciones depende de cada diseñador, pero al igual que ocurre con **Linux**, el objetivo en muchos casos es el de compartir las librerías, en este caso escritas en **VHDL** y **Verilog**, para acelerar así la creación de nuevos diseños y la adaptación acelerada de nuevos estándares. Es más, es incluso posible implementar un procesador **RISC-V** en un **FPGA**.

Pero en especial, **RISC-V** ha aumentado su popularidad en los últimos años por el hecho que promete romper la hegemonía de **ARM** en varios sectores. Y es que el set de instrucciones británico es el más utilizado a día de hoy, dado que se utiliza en todo tipo de dispositivos. Sin embargo, los cambios de políticas de **ARM Holdings** en los últimos tiempos y el susto que supuso que **NVIDIA** estuviese a punto de comprar dicha empresa hizo que **RISC-V** se convirtiese en una alternativa a tener en cuenta para el diseño de nuevas CPU y otros chips.

Su historia

El proyecto **RISC-V** nació en la facultad de ciencias de la computación de **Berkeley** de la mano del profesor **David Patterson** en 2008, quien ya había trabajado en los años 80 en la creación de la **ISA MIPS** y es famoso por ser autor de varios libros de texto de arquitectura de computadores. Los otros miembros del proyecto fueron el también profesor **Krste Asanovic** y los por aquel entonces alumnos de postgrado **Yunsup Lee** y **Andrew Waterman**.

La primera CPU la terminaron en 2011, un prototipo que fabricaron bajo el nodo de **28 nm** de **STMicroelectronics**, en un proyecto que tuvo como mecenas a empresas tan importante como **Intel** y **Microsoft**. Para 2015 ya eran **29 empresas** las que formaban parte de la **RISC-V Foundation** y 2018 empezaron a trabajar con la **Linux Foundation** para la creación del sistema operativo libre totalmente funcional bajo dicho set de instrucciones. Ya para 2020, con tal de ser totalmente independientes y estar fuera de la guerra comercial entre China y los Estados Unidos, movieron su sede a **Suiza**.

¿Cómo es el set de instrucciones RISC-V?

El set de instrucciones **RISC-V** no tiene una sola versión base, sino distintas variaciones que son las siguientes:

- **RV32I**: su unidad de ejecución principal opera con enteros de 32 bits y dispone de 32 registros.
- **RV32e**: diseñado para sistemas empujados, tiene la mitad de registros de propósito general que, en el caso anterior, 16.
- **RV64i**: es igual que la **RV32i**, pero con la capacidad de operar con enteros de 64 bits.
- **RV128i**: el mismo caso que el anterior, pero operando con instrucciones de 128 bits de tamaño.

Además, su forma de recorrer la memoria del sistema es **Little Endian**, esto significa que empezará a leer la memoria del sistema desde la dirección de memoria más baja. En todo caso, hay que tener en cuenta que de los sets de instrucciones base, el **RV32i** es el que se considera el estándar de facto, siendo el resto variaciones de este.

Extensiones y modularidad

Los sets de instrucciones base son fijos, esto significa que no se pueden añadir nuevas instrucciones al estándar desarrolladas por terceros. Sin embargo, la gran mayoría de instrucciones son opcionales, lo cual permite crear chips bajo la **ISA RISC-V** más especializados en una tarea concreta o de propósito general, según sea el caso. Dichas extensiones son las siguientes.

- **A:** instrucciones atómicas o que se pueden realizar con una simple operación.
- **B:** instrucciones para la manipulación de bits
- **C:** capacidad para operar con instrucciones de 16 bits.
- **D:** instrucciones de coma flotante a doble precisión (64 bits).
- **F:** lo mismo que el caso anterior, pero con precisión simple (32 bits).
- **H:** instrucciones para hipervisor, para el manejo de máquinas virtuales.
- **J:** instrucciones para lenguajes traducidos dinámicamente
- **L:** instrucciones para el cálculo en decimal.
- **M:** multiplicación y división de enteros
- **N:** instrucciones para la gestión de interrupciones a nivel de usuario.
- **P:** instrucciones SIMD en bloque
- **Q:** coma flotante de 128 bits
- **S:** instrucciones de supervisor
- **T:** instrucciones para el uso de memoria transaccional.
- **V:** instrucciones para hacer operaciones con vectores

Por lo que un chip **RISC-V** tendrá siempre uno de los conjuntos de instrucciones base, pero no tendrá todas las extensiones. Al contrario de lo que ocurre con **ARM** y **x86** donde cualquier CPU bajo dichas **ISA** tiene un set de instrucciones completo. En todo caso, en el caso de esos dos conjuntos de registros e instrucciones, también nos encontramos con extensiones que se pueden ver en ciertos procesadores y en otros no. Solo que el nivel de modularidad y adaptabilidad de **RISC-V** es mayor.

Decodificación de instrucciones

Una de las particularidades que tienen los sets de instrucciones reducidos es que su decodificador puede trabajar con instrucciones de tamaño fijo, lo cual significa que toma una cantidad de bits determinada en memoria y eso es una instrucción completa. Los **x86** en cambio, si bien tienen una **ISA** más completa, su decodificador ha de trabajar con diferentes tamaños de instrucción, provocando que en lo que es el FrontEnd del procesador se tengan que gastar más recursos y un manejo de instrucciones más estrecho.

Las instrucciones **RISC-V**, en cambio, funcionan en bloques de 16 bits. Esto significa que una de 64 bits realmente será decodificada como 4 instrucciones sucesivas de 16 bits, pero se solventará como una sola. Esto está pensado para reducir el consumo energético y hacer al conjunto de instrucciones ideal para situaciones de muy bajo consumo, como por ejemplo su uso en microcontroladores.

Niveles de privilegio y memoria virtual

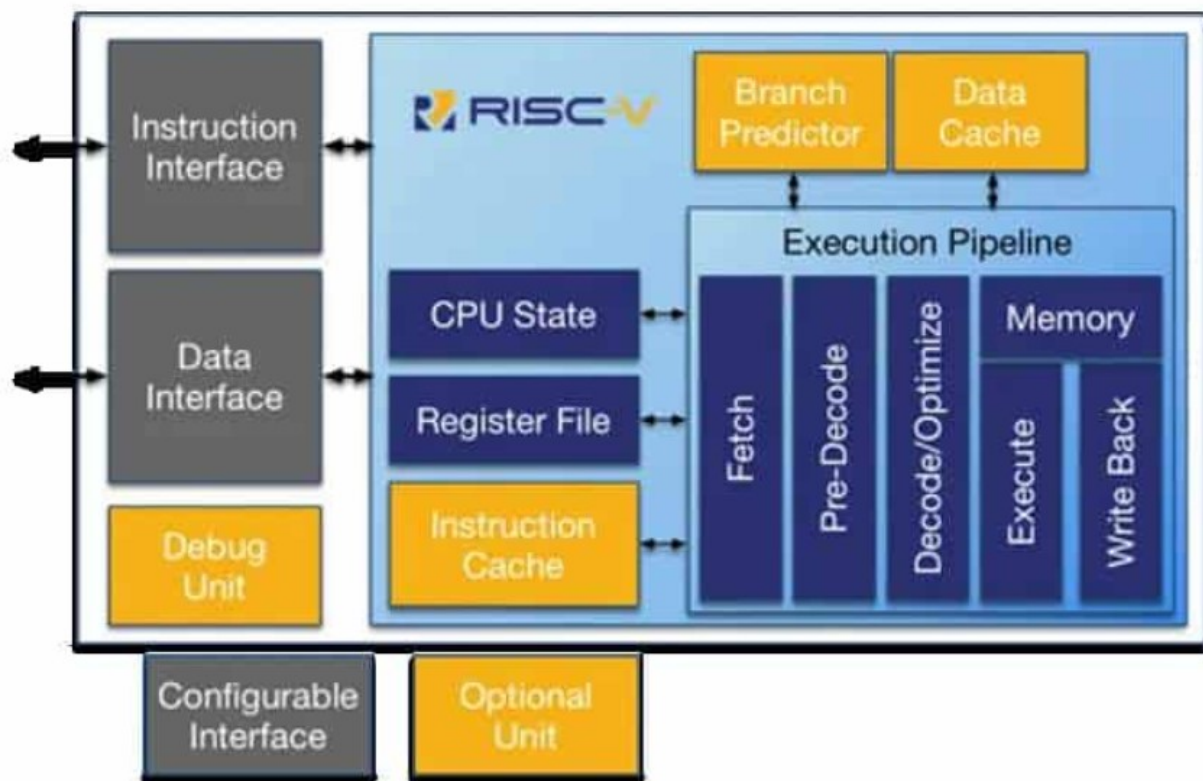
El conjunto de instrucciones **RISC-V** soporta hasta tres niveles de privilegio distintos en lo que a las instrucciones se refiere, lo cual es importante para evitar que una aplicación realice una operación ilegal.

- El primer nivel de privilegio es el modo máquina, es para cuando se usa la **ISA RISC-V** en sistemas que carecen

de sistema operativo y lo que hacen es ejecutar directamente el software.

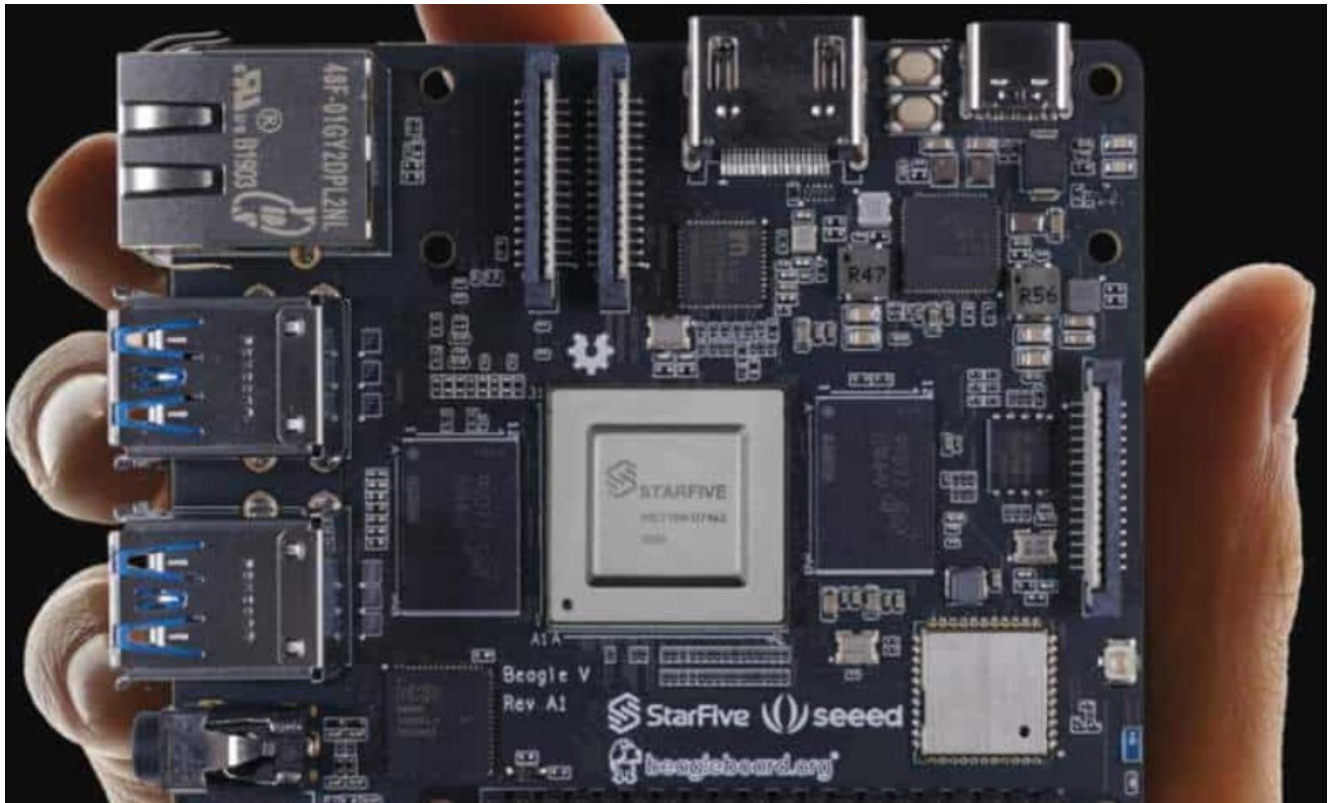
- El segundo nivel de privilegio es el del sistema operativo.
- El tercer nivel de privilegio es el que tienen las aplicaciones.

Se trata del mismo concepto que los anillos de ejecución de los **x86**, pero con la diferencia que el primer nivel es ideal para la creación de sistemas empotrados, como por ejemplo el microcontrolador de una lavadora.

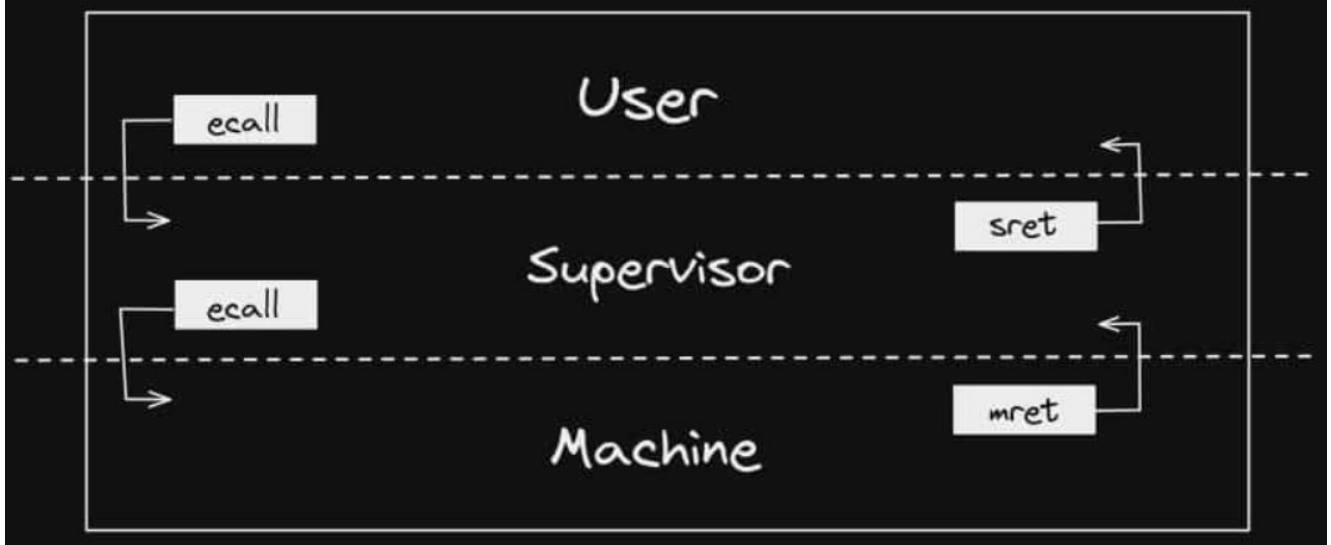


xxxxxxxxxxxxxxxxaa			16-bit ($aa \neq 11$)
xxxxxxxxxxxxxxxxxx		xxxxxxxxxxxxbbb11	32-bit ($bbb \neq 111$)
...xxx	xxxxxxxxxxxxxxxxxx	xxxxxxxxxxx011111	48-bit
...xxx	xxxxxxxxxxxxxxxxxx	xxxxxxxxxxx0111111	64-bit
...xxx	xxxxxxxxxxxxxxxxxx	xnnnxxxxx1111111	$(80+16*nnn)$ -bit, $nnn \neq 111$
...xxx	xxxxxxxxxxxxxxxxxx	x111xxxxx1111111	Reserved for ≥ 192 -bits

Byte Address: base+4 base+2 base



RISC-V Privilege Levels



En todo caso, esto no significa que sea un set de instrucciones simples, dado que **RISC-V** soporta memoria virtual por paginación, un requisito indispensable para el soporte de sistemas operativos complejos como **Linux**, **Windows** o **macOS**, aunque somos conscientes que de los últimos dos no hay versión para dicho conjunto de instrucciones, al menos, por el momento.

Fuente: elchapuzasinformatico.com.