

El ISA abierto RISC-V como eje vertebrador de las asignaturas de *Arquitectura de Computadores*

URJC

Katia Leal y Juan González-Gómez
katia.leal@urjc.es - objjuan.cube@gmail.com

RISC-V en la ETSIT-URJC

Estudiantes

GRADOS

- Ingeniería en Telemática, 3º
- Ingeniería en Tecnologías de la Telecomunicación, 3º
- Ingeniería en Sistemas de Telecomunicación, 3º
- Ingeniería de Robótica Software, 1º

GRADOS

- Ingeniería en Sistemas Audiovisuales y Multimedia, 4º

DOBLES GRADOS

- Ingeniería en Sistemas de Telecomunicación + Administración y Dirección de Empresas, 4º
- Ingeniería en Tecnologías de la Telecomunicación + Ingeniería Aeroespacial en Aeronavegación, 4º

Teoría

¿Por qué RISC-V?

- ¿MIPS, ARM, RISC-V?
- Similar a MIPS, la arquitectura de referencia utilizada en la mayor parte de los textos docentes
- David A. Patterson, John L. Hennessy, *Computer organization and design. The hardware/software interface, RISC-V Edition*. Morgan Kaufmann, 2018.
- Open-source hardware
- Simulador RARS

Alternativa prometedora

- Desarrollado originalmente en la Universidad de California en Berkeley, ofrece una versión simple, elegante y moderna de cómo deberían ser los repertorios de instrucciones actuales
- Más de 40 compañías se han unido a la Fundación RISC-V. La lista de patrocinadores incluye, virtualmente, a todos los actores principales, salvo ARM e Intel, como AMD, Google, Hewlett Packard Enterprise, IBM, Microsoft, NVIDIA, Oracle, y Qualcomm
- RISC-V Vs ARM/x86: diseño modular
- Flexibilidad: posibilidad de construir procesadores personalizados

Prácticas

- Material referenciado desde la Fundación RISC-V
- Publicación del curso en abierto en Github
- Videos en Youtube: 53
- 120 Ejercicios resueltos

<https://github.com/myTeachingURJC/2019-20-LAB-AO/wiki>

5 Prácticas / 12 sesiones:

- P1 - El simulador RARS: *Text, Static*
- P2 - E/S Mapeada. Llamadas al Sistema
- P3 - Bucles y saltos condicionales
- P4 - Subrutinas y Pila: *Stack*
- P5 - Memoria dinámica: *Heap*



Dirección memoria	Instrucción (Código máquina)
0x00400000	0x00002B7
0x00400004	0xAFF28293
0x00400008	0x0000C337
0x0040000C	0xA3A30313
0x00400010	0x0100006F
0x00400014	0x00000293
0x00400018	0x00010337
0x0040001C	0xFFFF3013
0x00400020	0x00A00893
0x00400024	0x00000073

Offset
jal x0, 0x08 (b salta)
Salto 4 instrucciones hacia adelante
¡IMPORTANTE!
El offset se expresa en medias palabras (16 bits)

Simulación

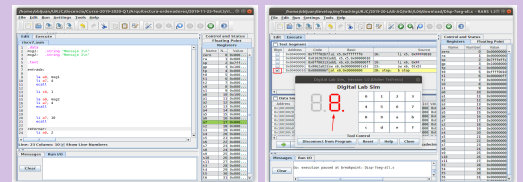
- Simulador RARS
- Software libre y multiplataforma
- Disponible en Github
- Nos permite modificarlo y adaptarlo
- RISC-V 32IM (32 bits)

<https://github.com/TheThirdOne/rars>

Depuración paso a paso

Dispositivos mapeados:

- Teclado hexadecimal
- Displays de 7 segmentos
- Compatible con GNU AS



Planes futuros

FPGAs libres

Síntesis en FPGAs libres:

- Prácticas para trabajar en RISC-V real
- Código máquina exportado desde RARS
- Carga en RISC-V en la FPGA
- Placa **Alhambra II** (Open Hardware)
- FPGA Lattice **ICE40HX8K**
- PicoRV32** (Clifford Wolf)
- Icestudio** (Software libre)

