

Clase 11 - RISC - V - Parte 1

Profesor: **IIC2343 - Arquitectura de Computadores**

- Felipe Valenzuela González

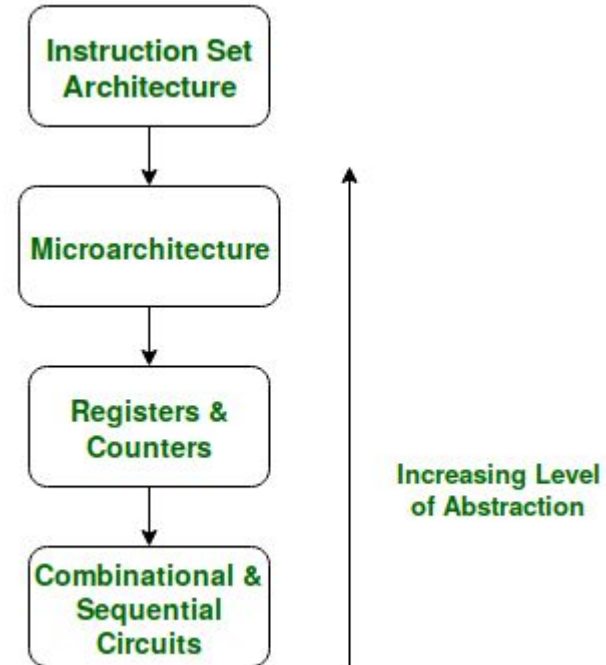
Correo:

frvalenzuela@alumni.uc.cl

Resumen de la clase pasada

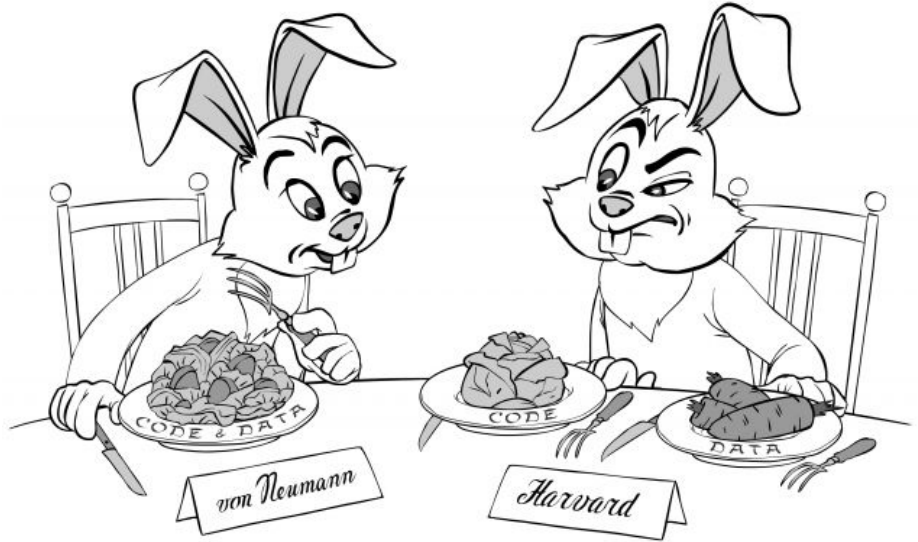
Arquitectura de Computadores: Instruction Set Architecture (ISA)

- Se refiere a tipo, formato, características de las instrucciones soportadas por **el computador**
- Especifica **cómo** escribir los programas en el computador
- Se compone básicamente de:
 - Tipos de instrucciones
 - Tipos de datos
 - Modos de direccionamiento de memoria
 - Formato de instrucción
 - Manejo del stack
 - Palabras por instrucción
 - Ciclos por instrucción



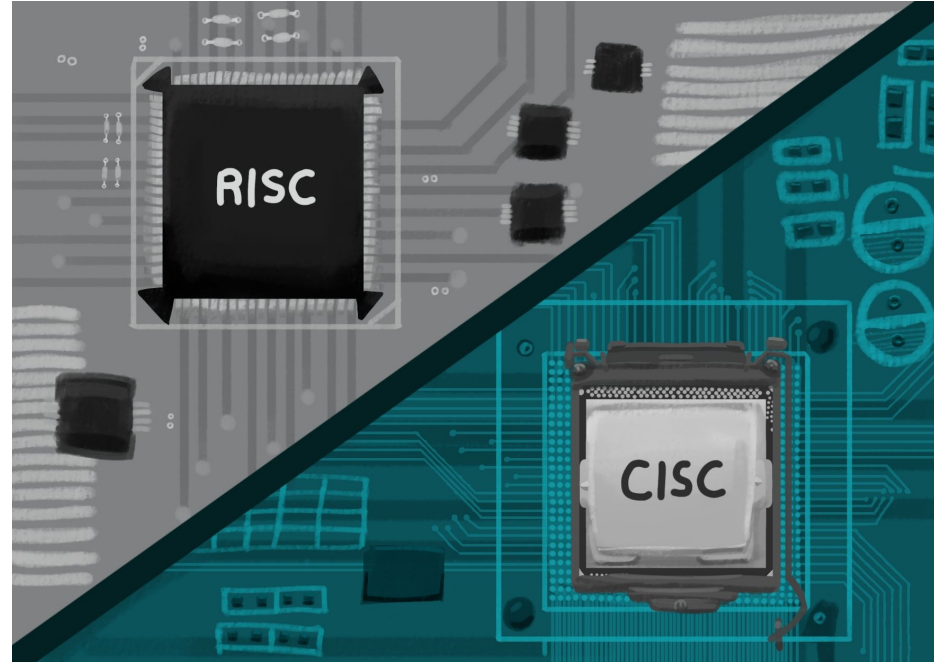
Arquitectura de Computadores: Microarquitectura - Paradigmas

- **Arquitectura Harvard:** Memoria de datos e instrucciones independientes
- **Arquitectura Von Neumann:** Memoria única que comparte datos e instrucciones. Permite escribir datos como si fueran instrucciones



Arquitectura de Computadores: ISA - Paradigmas

- **RISC: Reduced Instruction Set Computer.** Instrucciones pequeña y simples. Su diseño permite simplificar el hardware, poniendo énfasis en el software.
- **CISC: Complex Instruction Set Computer.** Muchas instrucciones y con complejidad alta. Énfasis en un hardware más complejo para poder ejecutarlas.



Arquitectura de Computadores: ISA - Industria - RISC-V

- **RISC-V** es una arquitectura de instrucciones (ISA) **libre**, basada en el diseño RISC, que optimiza el número de instrucciones
- A diferencia de otras ISAs, RISC-V **es abierta y no requiere regalías**, lo que permite a cualquiera diseñar, fabricar y vender chips y software
- Aunque no es la primera ISA abierta, destaca por su versatilidad y aplicación en una amplia variedad de dispositivos

Intel x86 32-bit



~1300 instructions
3 instructions / month
182 hours to read

ARM 32-bit



~500 instructions
79 hours to read

RISC-V RV32I



40 instructions
6 hours to read

¿Dudas?

RISC - V

- La arquitectura es de tipo Harvard, al igual que nuestro computador básico.
- El tipo load-store: existen instrucciones de acceso a memoria, pero las operaciones de la ALU se realizan exclusivamente con registros (o literal)



RISC - V

- Tiene 32 registros de propósito general, cada uno de 32 bits.
- Direcciones de memoria de 32 bits. Las palabras de memoria son de 32 bits (4 bytes), pero **las direcciones son por cada byte**
- Las instrucciones de operaciones aritméticas/lógicas tienen la estructura: palabra clave, registro de destino
- El **registro de destino** es donde se va a almacenar el valor de la operación.

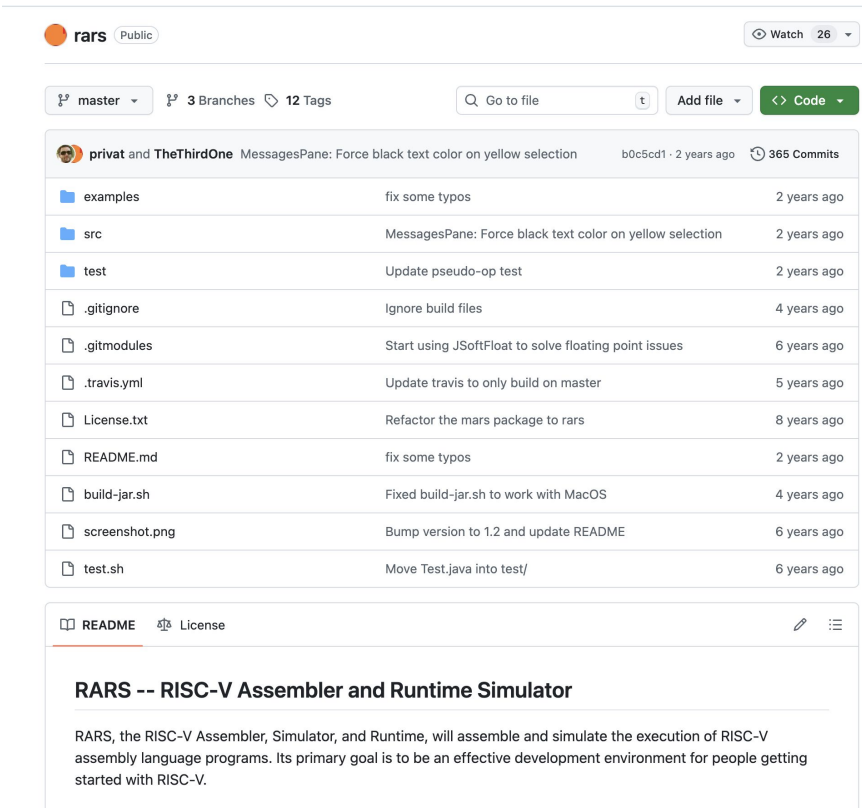
RISC - V

Registro	Nombre	Uso
x0	zero	Registro zero. Siempre almacena el valor 0, no se puede escribir.
x1	ra	Return Address. Se usa para guardar la dirección de retorno de una subrutina.
x2	sp	Stack Pointer. Almacena la dirección del último elemento del <i>stack</i> .
x3	gp	Global Pointer.
x4	tp	Thread Pointer.
x5-x7	t0 – t2	Registros temporales. Pueden perder su valor al llamar una subrutina.
x8-x9	s0 – s1	Registros guardados. Preservan su valor al llamar una subrutina.
x10-x11	a0 – a1	Argumentos de funciones / Valores de retorno.
x12-x17	a2 – a7	Argumentos de funciones.
x18-x27	s2 – s11	Registros guardados. Preservan su valor al llamar una subrutina.
x28-x31	t3 – t6	Registros temporales. Pueden perder su valor al llamar una subrutina.

¿Dudas?

RISC - V - RARS

- Emulador que usaremos en el curso
- Se usará en la actividad evaluada



The screenshot shows the GitHub repository for RARS (RISC-V Assembler and Runtime Simulator). The repository is public and has 3 branches and 12 tags. The commit history is displayed, showing a list of commits with their messages, commit IDs, and dates. The README section is visible at the bottom, providing an overview of the project.

Commit History:

Commit Message	Commit ID	Time Ago
privat and TheThirdOne MessagesPane: Force black text color on yellow selection	b0c5cd1	2 years ago
examples fix some typos		2 years ago
src MessagesPane: Force black text color on yellow selection		2 years ago
test Update pseudo-op test		2 years ago
.gitignore Ignore build files		4 years ago
.gitmodules Start using JSoftFloat to solve floating point issues		6 years ago
.travis.yml Update travis to only build on master		5 years ago
License.txt Refactor the mars package to rars		8 years ago
README.md fix some typos		2 years ago
build-jar.sh Fixed build-jar.sh to work with MacOS		4 years ago
screenshot.png Bump version to 1.2 and update README		6 years ago
test.sh Move Test.java into test/		6 years ago

README:

RARS -- RISC-V Assembler and Runtime Simulator

RARS, the RISC-V Assembler, Simulator, and Runtime, will assemble and simulate the execution of RISC-V assembly language programs. Its primary goal is to be an effective development environment for people getting started with RISC-V.

RISC - V - RARS

- Veamos código práctico !
- <https://github.com/TheThirdOne/rars/releases/tag/v1.6>

¿Dudas?

Clase 11 - RISC - V - Parte 1

Profesor: **IIC2343 - Arquitectura de Computadores**

- Felipe Valenzuela González

Correo:

frvalenzuela@alumni.uc.cl