

Arquitectura de computadores

Gustavo Gutiérrez-Sabogal

Información General

- **Duración:** 14 semanas
- **Intensidad horaria:** 5 horas semanales (70 horas totales)
- **Nivel:** Pregrado

Descripción del Curso

Este curso introduce los principios fundamentales de la arquitectura de computadores, abarcando desde la representación de datos y aritmética computacional hasta el diseño de procesadores y sistemas de memoria. Los estudiantes aprenderán cómo se organizan y diseñan los componentes de hardware de un computador. Se explorarán conceptos clave como el lenguaje ensamblador, la arquitectura del conjunto de instrucciones (ISA) y el diseño de procesadores con pipeline.

El curso combina teoría con práctica mediante el diseño e implementación de dos variantes de un procesador RISC-V RV32I utilizando Verilog como lenguaje de descripción de hardware (HDL): primero un procesador monociclo y posteriormente una versión con pipeline de 5 etapas.

Objetivos de Aprendizaje

Al finalizar el curso, los estudiantes serán capaces de:

- Comprender la organización y funcionamiento interno de un computador.
- Analizar y diseñar componentes básicos de un procesador (ALU, registros, control)
- Programar en lenguaje ensamblador RISC-V y comprender la arquitectura del conjunto de instrucciones RV32I
- Diseñar e implementar un procesador RISC-V RV32I monociclo en Verilog
- Diseñar e implementar un procesador RISC-V RV32I con pipeline de 5 etapas
- Analizar y resolver riesgos (hazards) en procesadores con pipeline
- Entender la jerarquía de memoria y sus principios de diseño
- Evaluar el rendimiento de arquitecturas de computadores

Conocimientos previos

En este curso se asume que el estudiante ya ha tenido contacto con cursos de electrónica digital. En particular, el estudiante debe conocer sobre los siguientes temas:

- Representación de datos en sistema decimal, binario y hexadecimal.
- Circuitos electrónicos básicos como compuertas lógicas, sumadores, multiplexores. En general circuitos asíncronos.

- Circuitos algo más avanzados como flip flops, registros y memorias. En general circuitos síncronos.
- Descripción de circuitos en algún lenguaje de descripción de hardware, por ejemplo VHDL, Verilog, etc..
- Simulación de circuitos síncronos.

Contenido del Curso

Semana 1: Todo comienza en el compilador

- Del código de alto nivel al código máquina: el rol del compilador
- Niveles de abstracción: aplicaciones, sistema operativo, ISA, microarquitectura, hardware
- La interfaz hardware/software: ¿qué debe entender el hardware?
- Motivación: ¿por qué un ingeniero de sistemas debe entender arquitectura?
- Introducción al lenguaje ensamblador como producto del compilador

Semana 2-3: RISC-V

- Para construir un procesador necesitamos elegir una arquitectura, por qué RISC-V?
- Lenguaje ensamblador RISC-V.
- Los compiladores producen el ensamblador, pero para ejecutarlo nosotros debemos entenderlo.
- Cómo un programa en C puede ser transformado en ensamblador?
 - Cómo se representa una variable?
 - Cómo se representa un condicional?
 - Cómo se representa un ciclo (e.g. for, while, etc.)?
 - Cómo se representa una función?
- Conjunto de instrucciones RV32I
- Comienza desarrollo de ensamblador que transforma a binario.

Semana 4-5-6: Comienza la construcción de una CPU.

- Un procesador es un circuito que ejecuta un programa. Qué componentes tiene este circuito?
 - Contador de programa y un sumador.
 - Memoria de instrucciones.
 - Unidad de registros,
 - Unidad de ejecución (ALU y dos multiplexores).
 - Unidad de memoria.
 - Unidad de escritura (Writeback).
- Cómo estos componentes interactúan para ejecutar cada tipo de instrucción?
- Ejecución de instrucciones tipo R.
- Ejecución de instrucciones tipo I (aritméticas y lógicas).
- Ejecución de instrucciones tipo I (carga).
- Ejecución de instrucciones tipo S.

Semana 7-8: Continua construcción de una CPU.

- Ejecución de instrucciones tipo B.

- Ejecución de instrucciones tipo I (salto).
- Ejecución de instrucciones tipo J.
- Finaliza desarrollo de ensamblador.

Metodología

El curso se desarrolla mediante:

- **Clases magistrales** (2 horas/semana): Presentación de conceptos teóricos.
- **Laboratorios** (2 horas/semana): Implementación práctica en Verilog y simulación.

Evaluación

El componente práctico del curso se centra en el diseño e implementación de procesadores RISC-V RV32I en Verilog:

| Componente | Porcentaje |
|----------------|------------|
| Ensamblador | 34% |
| CPU monociclo | 33% |
| CPU segmentada | 33% |

Proyecto 1: Procesador Monociclo (Semana 10)

Implementación de un procesador RISC-V RV32I de ciclo único que ejecuta el conjunto base de instrucciones. Los estudiantes diseñarán el datapath, la unidad de control y verificarán el funcionamiento mediante testbenches.

Proyecto 2: Procesador con Pipeline (Semanas 12-14)

Extensión del procesador monociclo para incorporar un pipeline de 5 etapas (IF, ID, EX, MEM, WB). Se implementarán mecanismos de detección y resolución de riesgos mediante forwarding y stalling.

Recursos Bibliográficos

Textos principales:

- Patterson, D. A., & Hennessy, J. L. *Computer Organization and Design: The Hardware/Software Interface* (RISC-V Edition). Morgan Kaufmann.
- Harris, S. L., & Harris, D. *Digital Design and Computer Architecture* (RISC-V Edition). Morgan Kaufmann.

Textos complementarios:

- Hennessy, J. L., & Patterson, D. A. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann.
- Null, L., & Lobur, J. *The Essentials of Computer Organization and Architecture*. Jones & Bartlett Learning.

Herramientas:

TODO