



# TRABAJO FINAL: PIPELINE PROCESADOR MIPS SIMPLIFICADO

Arquitectura de  
Computadoras  
2020

---

# Consigna

- Implementar el pipeline de 5 etapas del procesador MIPS

# Requerimientos

# Implementar el Procesador MIPS

## Segmentado en las siguientes Etapas

- ❑ **IF (Instruction Fetch):** Búsqueda de la instrucción en la memoria de programa.
- ❑ **ID (Instruction Decode):** Decodificación de la instrucción y lectura de registros.
- ❑ **EX (Execute):** Ejecución de la instrucción.
- ❑ **MEM (Memory Access):** Lectura o escritura desde/hacia la memoria de datos.
- ❑ **WB (Write back):** Escritura de resultados en los registros.

# Instrucciones a implementar

## □ R-type

~~SLL~~, ~~SRL~~, ~~SRA~~, ~~SLLV~~, ~~SRLV~~, ~~SRAV~~,  
~~ADDU~~, ~~SUBU~~, ~~AND~~, ~~OR~~, ~~XOR~~, ~~NOR~~, ~~SLT~~

## □ I-Type

~~LB~~, ~~LH~~, ~~LW~~, ~~LWU~~, ~~LBU~~, ~~LHU~~, ~~SB~~, ~~SH~~,  
~~SW~~, ~~ADDI~~, ~~ANDI~~, ~~ORI~~, ~~XORI~~, ~~LUI~~,  
~~SLTI~~, ~~BEQ~~, ~~BNE~~, ~~J~~, ~~JAL~~

## □ J-Type

~~JR~~, ~~JALR~~

# Riesgos

- El procesador debe tener soporte para los siguientes tipos:
  - **Estructurales.** Se producen cuando dos instrucciones tratan de utilizar el mismo recurso en el mismo ciclo.
  - **De datos.** Se intenta utilizar un dato antes de que esté preparado. Mantenimiento del orden estricto de lecturas y escrituras.
  - **De control.** Intentar tomar una decisión sobre una condición todavía no evaluada.

# Riesgos

- Para dar soporte a los riesgos nombrados se debe implementar las dos unidades riesgos:
  - Unidad de Cortocircuitos
  - Unidad de Detección de Riesgos

# Otros requerimientos

- El programa a ejecutar debe ser cargado en la memoria de programa mediante un archivo ensamblado.
  - Debe implementarse un programa ensamblador que convierte código assembler de MIPS a código de instrucción.
  - Debe transmitirse ese programa mediante interfaz UART antes de comenzar a ejecutar
- Se debe simular una unidad de Debug que envíe información hacia y desde el procesador mediante UART.



# Debug unit

- Se debe enviar a través de la UART:
  - Contenido de los 32 registros
  - PC
  - Contenido de la memoria de datos usada
  - Cantidad de ciclos de clock desde el inicio

# Modos de operación

- Antes de estar disponible para ejecutar, el procesador está a la espera para recibir un programa mediante la Debug Unit.
- Una vez cargado el programa, debe permitir dos modos de operación:
  - Continuo, se envía un comando a la FPGA por la UART y esta inicia la ejecución del programa hasta llegar al final del mismo (Instrucción HALT). Llegado ese punto se muestran todos los valores indicados en pantalla.
  - Paso a paso: Enviando un comando por la UART se ejecuta un ciclo de Clock. Se debe mostrar a cada paso los valores indicados.

# Modo de presentación

- Al no disponer de placas de desarrollo, el trabajo se debe presentar simulado.
- La simulación debe ser post-síntesis incluyendo información de timing.
- Mostrar reporte de timing, con máxima frecuencia de clock soportada.
- El clock del sistema debe crearse usando el ip-core correspondiente.



# Bibliografía

## ▣ Instrucciones:

- ▣ *MIPS IV Instruction Set*

## ▣ Pipeline:

- ▣ Computer Organization and Design 3rd Edition. Chapter 6. Hennessy- Patterson