



Sección: D I 2  
Equipo I

# Seminario de Arquitectura de Computadoras

## Fase I de Proyecto Final

Integrantes:

Carlos Arturo González Ramírez

Estrada Huerta Félix Eduardo

Juan José Ortega Morales

# Procesadores MIPS

- Con el nombre de **MIPS** (siglas de ***M**icroprocessor **w**ithout **I**nterlocked **P**ipeline **S**tages*) se conoce a toda una familia de microprocesadores de arquitectura RISC desarrollados por MIPS Technologies.
- Supuestamente surgió a comienzos de los 80 en Stanford. Estos procesadores MIPS Sintetizan las principales características de la arquitectura RISC, que es una arquitectura simple y eficiente.

# ¿Qué es RISC?

- **RISC** (del inglés reduced instruction set computer) es un tipo de microprocesador con las siguientes características fundamentales:
  - Instrucciones de tamaño fijo y presentadas en un reducido número de formatos.
  - Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.
- El objetivo de diseñar máquinas con esta arquitectura es posibilitar la segmentación y el paralelismo en la ejecución de instrucciones y reducir los accesos a memoria.

# Organización de un MIPS

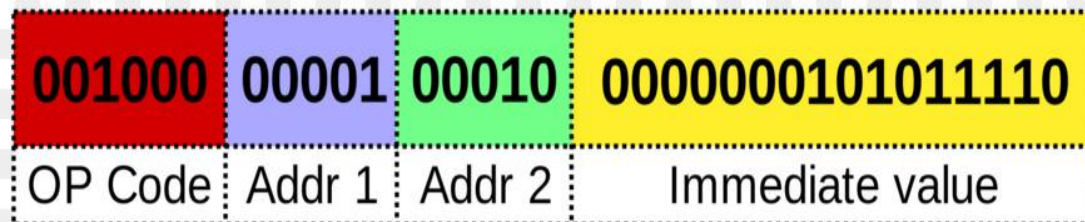
- Un MIPS se estructura en base de los siguientes elementos:
  - Unidad Aritmética y Lógica (ALU).
  - Unidad Aritmética entera, operaciones de multiplicación y división.
  - Unidad Punto Flotante.
  - Coprocesador dedicado al manejo de memoria caché y virtual
- La arquitectura MIPS requiere que el software implemente algunas limitaciones en el diseño que está normalmente considerado parte de la implementación del hardware. Este papel presenta resultados experimentales en la efectividad de este procesador como un programa anfitrión.

# Set de Instrucciones

- Es una especificación que detalla las instrucciones que una CPU puede entender y ejecutar, o el conjunto de todos los comandos implementados por un diseño particular de una CPU.
- El término describe los aspectos del procesador generalmente visibles para un programador, incluidos los tipos de datos nativos, las instrucciones, los registros, la arquitectura de memoria y las interrupciones, entre otros aspectos.

- Existen 3 principales tipos de set de instrucciones: CISC, RISC y SISC.
- La arquitectura del conjunto de instrucciones (ISA) se emplea a veces para distinguir este conjunto de características de la microarquitectura, que son los elementos y técnicas que se emplean para implementar el conjunto de instrucciones. Entre estos elementos se encuentran las microinstrucciones y los sistemas de caché.

### MIPS32 Add Immediate Instruction



Equivalent mnemonic: **addi** \$r1, \$r2, 350

# Tipos de Instrucciones

- **Formato R**

- Utilizado por las instrucciones aritméticas y lógicas

Tipo R (shamt: <i>shift amount</i> en instrucciones de desplazamiento)	Cód. Op.	Registro fuente 1	Registro fuente 2	Registro destino	Funct	
	xxxxxx	rs	rt	rd	shamt	funct
	6	5	5	5	5	6
	31-26	25-21	20-16	15-11	10-6	5-0

- **Formato I**

- Utilizado por las instrucciones de transferencia, las de salto condicional y las instrucciones con operando inmediatos

Tipo I (carga o almacenamiento, ramificación condicional)	Cód. Op.	Registro base	Registro destino	Desplazamiento
	xxxxxx	rs	rt	Inmediato
	6	5	5	16
	31-26	25-21	20-16	15-0

- **Formato J**

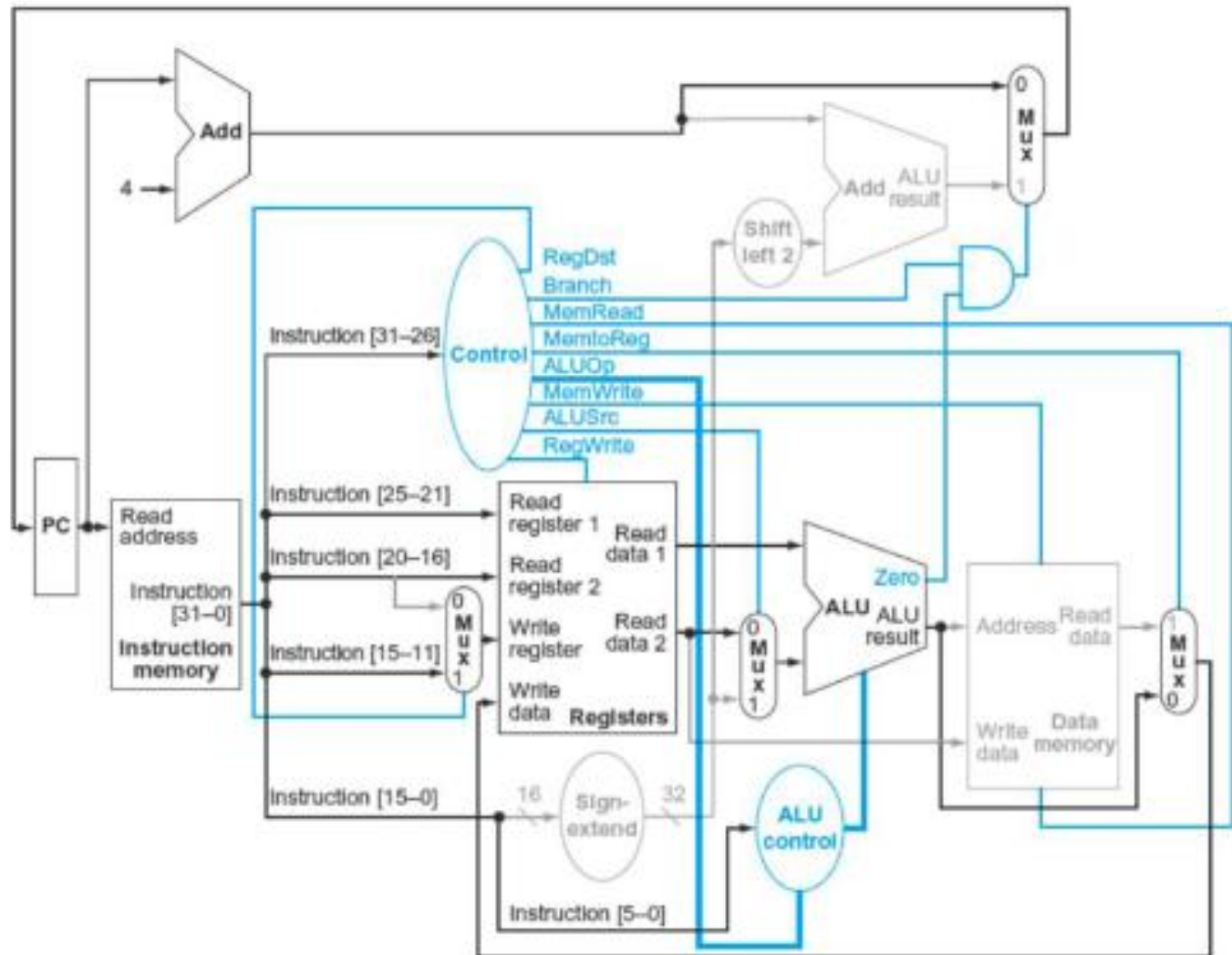
- Utilizado por las instrucciones de bifurcación



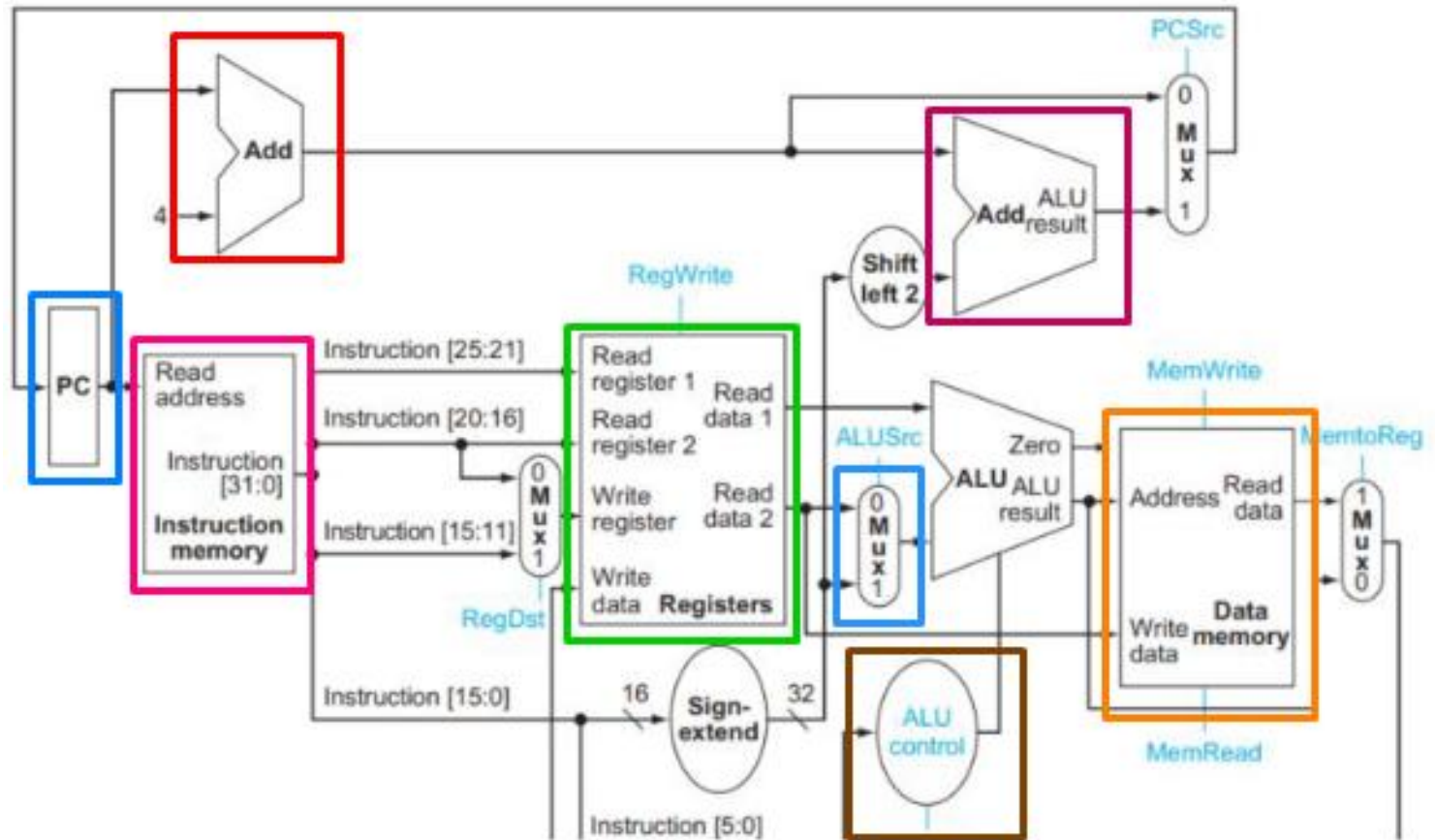
- Aunque tener múltiples formatos complica la circuitería, se puede reducir la complejidad guardándolos de forma similar. Por ejemplo, los tres primeros campos de los formatos de tipo R e I son del mismo tamaño y tienen los mismos nombres.
- Los formatos se distinguen por el valor del primer campo: a cada formato se le asigna un conjunto de valores distintos en el primer campo y por lo tanto la circuitería sabe si ha de tratar la última mitad de la instrucción como tres campos, es decir, como tipo R, o como un campo simple, tipo I, o si la instrucción es tipo J.



# Reporte de Proyecto Verilog



# Estructura Básica del Proyecto



# Índice de Colores de Diagrama

- ADD
- PC
- Instruction Memory
- Data Memory
- Register
- ALU Control
- MUXs Multiplexor
- ALU (Aritmetic Logic Unit)

# Reporte de Reuniones

- Reuniones:
  1. 20/05/2021 de 4pm - 5pm //Reunión General de Organización del Proyecto
  2. 21/05/2021 de 5pm - 6:30pm //Conocimiento básico sobre GIT
  3. 22/05/2021 de 7pm - 8pm //Inicio documentación
  4. 24/05/2021 de 5pm - 6pm //Detalles sobre documentación y código

Las primeras reuniones del proyecto se centraron en el correcto uso de Git para un mejor manejo de la información, así como evitar redundancias dentro de los mismos archivos que cada uno de los integrantes poseía

- Las reuniones consecuentes se utilizaron para el inicio de la documentación requerida por el profesor, así como el inicio de la codificación en verilog en la que se llevará a cabo el proyecto.
- Links de Apoyo durante las sesiones y reuniones:
  - <https://youtu.be/QOFAmkAQil0> /\* Clonación de Archivos \*/
  - <https://youtu.be/HiXLkL42tMU> /\* Comandos Básicos \*/
  - <https://www.youtube.com/watch?v=RReglKu-z3k> /\* Subir Proyecto a Git \*/
- links Páginas:
  - Clonar Archivo a Computador
    - <https://walkiriaapps.com/tecnologia/clonar-proyecto-github/>
  - Comandos Básicos
    - <https://codingpotions.com/git-comandos-basicos-avanzados>