

**UNIVERSIDAD NACIONAL DE INGENIERIA  
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**

**1er. LABORATORIO DE EE604 O-P INTRODUCCIÓN A MICROCONTROLADORES  
(modalidad presencial)**

En esta experiencia de laboratorio, el estudiante obtendrá experiencia en resolver un algoritmo en lenguaje ensamblador para un procesador RISC con implementación monociclo usando la herramienta EDA **Quartus II** de Altera. Igualmente, el estudiante obtendrá experiencia en trabajar con memoria embebida de un FPGA usando la herramienta **In-System Memory Content Editor** de Quartus II. Por último, el estudiante obtendrá experiencia en el manejo de un analizador lógico basado en FPGA a fin de depurar la ejecución de programas binarios del procesador RISC. El procesador RISC a implementar es el de la arquitectura MIPS de 32 bits (arquitectura Harvard) en su versión monociclo [1].

El diagrama de bloques del procesador MIPS de 32 bits en su versión monociclo se muestra en la Figura 1. El procesador MIPS posee dos memorias separadas, una para las instrucciones y otra para los datos. Ambas memorias están implementadas en memoria RAM embebida del FPGA.

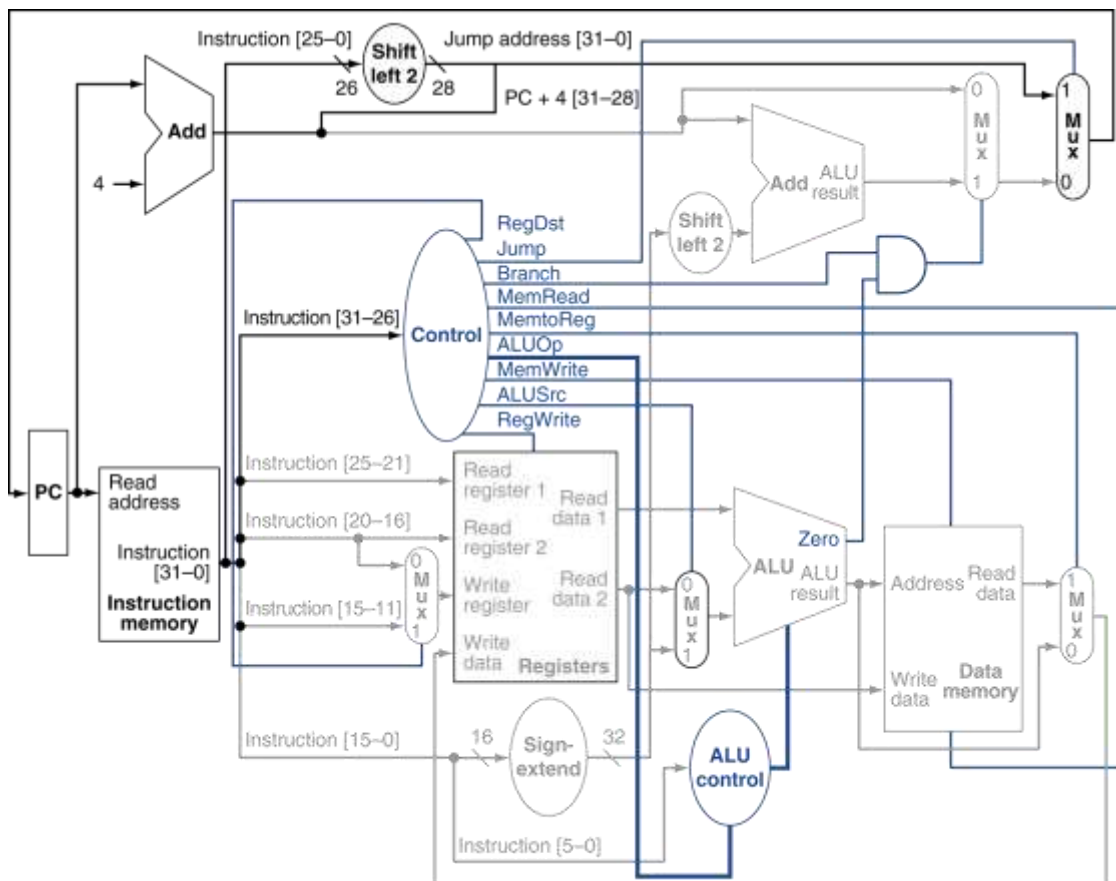


Figura 1.- Procesador RISC MIPS de 32 bits (implementación monociclo).

Las fases por las que pasa una instrucción en el procesador RISC MIPS son las siguientes:

- **IF** : Instruction **F**etch (búsqueda de instrucción).
- **ID** : Instruction **D**ecode (decodificación de instrucción).
- **EXE** : Instruction **E**XEcution (ejecución de instrucción).
- **MEM** : **M**EMory Access (ciclo de acceso a memoria de datos).
- **WB** : Escritura en registro destino del Register File (**W**rite**B**ack)

El bloque "**Control**" entrega las señales de control a diferentes bloques del procesador MIPS a fin de ejecutar adecuadamente los diferentes tipos de instrucciones. El bloque "**Control**" y los bloques "**Sign Extend**", "**Shift Left 2**", y "**ALU Control**" se implementan con lógica combinacional.

El procesador MIPS cuenta con un conjunto de 32 registros c/u de 32 bits (representado por el bloque “Registers”). Este conjunto de registros muestra dos valores a ser leídos (puertos “Read Data1” y “Read Data2”), y los números de estos registros están dados por los valores “Read Register1” y “Read Register2” (c/u de 5 bits), respectivamente, de forma tal que, el registro con número “0” siempre tiene el valor 00000000H y no puede ser modificado. Por otro lado, se puede actualizar un registro, cuyo número está dado por el valor “Write Register”. El nuevo dato a escribir en el registro seleccionado corresponde al valor que ingresa al puerto “Write Data”, siempre y cuando la señal “RegWrite” tenga un valor de “1” lógico. Toda actualización (escritura) o lectura de un registro ocurre en el **flanco de subida** de la señal de reloj (“clock”). Los valores de “PC” y “Address” en “Data Memory” son de 32 bits, pero la memoria de instrucciones “Instruction Memory” y la memoria de datos “Data Memory” solo contendrán 1024 posiciones (10 bits de dirección), y tendrán 32 bits de datos por cada dirección de memoria. La memoria de instrucciones será del tipo **ROM**, y la memoria de datos será del tipo **RAM**.

El procesador MIPS trabaja con instrucciones de longitud fija de 32 bits, donde se definen los siguientes formatos de instrucción (ver Tabla 1):

- Formato **R**: para ejecución de instrucciones aritméticas y lógicas entre Registros
- Formato **I**: para ejecución de instrucciones con un operando de memoria (dirección o Inmediato)
- Formato **J**: para ejecución de instrucciones tipo Jump

	Instruction [31 – 0]					
Formato	bits [31-26]	bits [25-21]	bits [20-16]	bits [15-11]	bits [10-6]	bits [5-0]
R	opcode	Rs	Rt	Rd	Shift	Function
I	opcode	Rs	Rt	Memory address / Immediate value		
J	opcode	Branch target address				

Tabla 1.- Formatos de instrucción para el procesador MIPS de 32 bits [2].

El juego de instrucciones disponible (con los respectivos formatos y valores por cada campo de la Tabla 1) para el procesador MIPS RISC de 32 bits se acompaña a la presente experiencia de laboratorio (ver documento “**MIPS\_Instruction\_Set.pdf**”) subido al repositorio del curso.

Para la realización de esta experiencia de laboratorio, se proveen los siguientes archivos de inicialización de memoria embebida como plantillas, cuyos contenidos deberán ser completados para el algoritmo solicitado:

- imemory.mif : archivo de inicialización de memoria de instrucciones
- dmemory.mif : archivo de inicialización de memoria de datos

El algoritmo que se debe implementar en esta experiencia de laboratorio corresponde a la obtención del número de Fibonacci de un dato **N**, recordando que  **$F(N) = F(N-1) + F(N-2)$** , donde **F(N)** es el número de Fibonacci. Se sabe que **F(0) = 0**, y **F(1) = 1**. Para este laboratorio, se asume que el número **N** será mayor o igual a 3. **La obtención de F(N) se debe hacer de manera recursiva**, como fuera explicada en clase (ver diapositivas 44 y 45 del capítulo 3) utilizando la instrucción **jal** (jump and link) y **jr** (jump register). Para la obtención de **F(N)**, utilizar las instrucciones del archivo “**MIPS\_Instruction\_Set.pdf**” que forma parte de la experiencia. El número **N** y **F(N)** no se deben destruir al término del algoritmo. El programa principal se ejecutará en la memoria de instrucciones a partir de la dirección **0000h**, el valor de **N** y **F(N)** estarán ubicados en direcciones de byte **0020h** y **0024h**, respectivamente, de la memoria de datos. El puntero de pila **sp (r29)** se inicializa en **0x0200** (512 en base 10). Un valor inicial de **N** estará previamente cargado en el archivo **dmemory.mif**. Al término de la obtención de **F(N)**, el procesador MIPS deberá permanecer en un “lazo infinito” de forma tal que la última instrucción del programa será la instrucción **j** (jump) a sí misma. El programa principal deberá utilizar obligatoriamente la instrucción **jal fibonacci**, donde la etiqueta “fibonacci” se ubica en la dirección de byte **0200h**. Se podrá evaluar y/o modificar el contenido de la memoria **RAM** y/o **ROM** en línea, utilizando la herramienta **In-System Memory Content Editor** del software Quartus II de Altera.

Como parte de la realización de esta experiencia de laboratorio, el estudiante deberá calcular el tiempo de ejecución del programa que implementa la obtención de **F(N)**, en función de **N**, tomando en cuenta que la frecuencia de reloj del procesador MIPS es de **10 MHz**. La asignación de pines del proyecto MIPS monociclo permite que el FPGA EP2C35F672C6 de la familia Cyclone II se pueda comunicar apropiadamente con otros dispositivos de la tarjeta de desarrollo DE2 (<http://de2.terasic.com/>) del fabricante Terasic.

El estudiante deberá usar el ensamblador preparado para el curso, y cuyo ejecutable es **mips.exe** (el cual se encuentra en la carpeta /Software/Assembler/ del repositorio del curso), el cual generará el archivo **imemory.mif** para inicializar la memoria **ROM** a partir de un archivo **imemory.asm** (que incluye las instrucciones del procesador MIPS que resuelve el algoritmo solicitado), a fin de cargarlo en el FPGA y ejecutarlo en el procesador MIPS (implementado en VHDL). Se proporciona para esto, un archivo plantilla de **imemory.asm** como ejemplo con una secuencia de instrucciones. De preferencia, usar **add r0, r0, r0** como primera instrucción a fin de evitar mensajes de error al momento de usar el ensamblador **mips.exe**. Los registros del procesador van del **r0** al **r31**, y todo valor inmediato debe especificarse como un número decimal. El ensamblador acepta etiquetas (pero evitar etiquetas que empiecen con la letra “r”), y deben seguirse las recomendaciones de cómo escribir el archivo \*.asm (usando Notepad o Notepad++) indicadas en la 2da página del archivo **MIPS\_Instruction\_Set.pdf**.

## Notas Generales para toda la experiencia

- Fecha de inicio de la experiencia de laboratorio: semana del 06/10/2025 (en los respectivos horarios).
- Duración de la experiencia de laboratorio: 2 sesiones consecutivas.
- Finalización de la experiencia de laboratorio: semana del 20/10/2025 (en los respectivos horarios).
- **Entrega del Informe Previo:** semana del 06/10/2025 (en los respectivos horarios).
  - **Contenido del Informe Previo:** explicación del diagrama de flujo de la solución, uso de registros para el algoritmo, archivos \*.mif en versión inicial. Todo debe incluirse en un solo archivo **zip** o **rar**.
- **Entrega del Informe Final:** semana del 20/10/2025 (en los respectivos horarios).
  - **Contenido del Informe Final:** archivos \*.mif en versión final. Cálculo del tiempo de ejecución y cuál es el número N más grande para el código diseñado sin que afecte los resultados. Informe Final, con las observaciones y conclusiones. Todo debe incluirse en un solo archivo **zip** o **rar**.
- **Test de la 1ra Experiencia de laboratorio:** semana del 20/10/2025 (en los respectivos horarios).
- La calificación para esta experiencia de laboratorio es la que se muestra en la Tabla 2.

Concepto	Descripción del concepto	Máximo puntaje
Asistencia	Asistencia a sesiones presenciales. Tolerancia máxima: 05 minutos	3.0
Test	Test sobre la experiencia: semana del 20/10/2025, 10 minutos	4.0
Informe Previo	Diagrama de flujo, archivos *.mif iniciales, etc.: semana del 06/10/2025	3.0
Semana 2	Configuración del FPGA y demostración experimental: semana del 20/10/2025	5.0
Informe Final	Archivos *.mif, tiempo de ejecución, conclusiones, etc.: semana del 20/10/2025	5.0
<b>Puntaje total</b>	<b>Máximo puntaje total de calificación de la 1ra experiencia de laboratorio</b>	<b>20.0</b>

Tabla 2.- Calificación para la 1ra experiencia de laboratorio del curso.

- La forma de entrega de los informes y archivos \*.mif será **vía el Aula Virtual de la UNI**.
- El nombre del archivo **zip** (o **rar**) debe tener la siguiente nomenclatura: **Lab-0X-Informe-YZ-Grupo-0W.zip**, debiendo reemplazar **0X** por 01, 02, etc. (número de la experiencia), **YZ** con **Previo** o **Final**, y **0W** con el número del grupo asignado (01, 02, etc).
- La fecha y hora de presentación de cada informe será dentro de la hora de clase de laboratorio, durante la semana de entrega de los informes indicados anteriormente. Basta que un integrante del grupo suba el informe al Aula Virtual de la UNI
- De haber retraso en la entrega de los informes, se calificará con **CERO (0.0)** dentro de la calificación final.
- Presentar los Informes de acuerdo a la forma sugerida en el documento "NORMAS PARA LA PRESENTACION DE INFORMES DE LABORATORIO DE INTRODUCCIÓN A MICROCONTROLADORES EE604", que se ha subido al repositorio digital del curso.
- El formato de presentación de los informes en formato IEEE es opcional, pero no es recomendable.
- Adherirse al "Código de Honor" del curso, el cual también se ha subido al repositorio digital del curso. Cualquier violación a dicho código será sancionado.

## **Referencias Bibliográficas**

1. David A. Patterson & John L. Hennessy, "Computer Organization & Design: The Hardware/Software Interface" 4th Edition, Morgan Kaufmann Publishers, 2009. ISBN: 978-0-12-374493-7.
2. MIPS Technologies Inc., "MIPS32® Architecture for Programmers Volume II: The MIPS32® Instruction Set", Document Number: MD00086, Revision 2.50, July 1, 2005.

El profesor  
Ing. A.F.M.V., Ph.D.

Lima, 28 de Setiembre de 2025