



**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA**  
**Laboratorio de Languages de programación**

CARRERA	CURSO	AMBIENTE
Ingeniería de Sistemas	Languages de programación	305

PRACTICA No	NOMBRE DE PRÁCTICA	CODIGO DE LAB.	DURACION (HORAS)
01	Ensamblador	305	2

REVISION	FECHA	DESCRIPCIÓN
1	19/03/2019	Revisión de Guías de Laboratorio

### 1. OBJETIVOS

- Dar a conocer al estudiante de conceptos básicos de la programación en ensamblador
- Dar a conocer al estudiante el procesador MIPS y su simulador SPIM

### 2. TEMAS A TRATAR

- Lenguaje ensamblador en SPIM

### 3. INTRODUCCIÓN (MARCO TEÓRICO)



## **MIPS**

MIPS es un conjunto de arquitectura del conjunto de instrucciones reducido equipo instrucción desarrollada por MIPS Technologies. Las arquitecturas MIPS primeros eran de 32 bits, con versiones de 64 bits añadidos más tarde. Existen múltiples revisiones del conjunto de instrucciones MIPS, incluyendo MIPS I, MIPS II, MIPS III, MIPS IV, MIPS V, MIPS32 y MIPS64 - Las revisiones actuales son MIPS32 y MIPS64. MIPS32 y MIPS64 definen un registro de control conjunto, así como el conjunto de instrucciones [1].

## **Historia**

En 1981, un equipo dirigido por John L. Hennessy en la Universidad de Stanford comenzó a trabajar en lo que sería el primer procesador MIPS. El concepto básico era para aumentar el rendimiento mediante el uso de tuberías de instrucciones profundas. Canalización como técnica básica era muy conocido antes, pero no se convirtió en su máximo potencial. CPUs se construyen a partir de una serie de sub-unidades dedicadas como decodificadores de instrucciones, ALU, unidades de carga/almacenamiento, etc. En un diseño no optimizado tradicional, una instrucción particular en una secuencia del programa debe ser completado antes de la próxima puede ser emitida para su ejecución, en una arquitectura segmentada, las instrucciones sucesivas vez pueden superponerse en la ejecución. Por ejemplo, al mismo tiempo, una instrucción de matemáticas se alimenta en la unidad de coma flotante, la unidad de carga/tienda puede buscar la siguiente instrucción.

Un obstáculo importante para la canalización era que algunas instrucciones, como la división, necesitan más tiempo para completar, por lo que la CPU tiene que esperar antes de pasar a la siguiente instrucción en la tubería. Una solución a este problema es utilizar una serie de sistemas de bloqueo que permite a las etapas para indicar que están ocupados, haciendo una pausa las otras etapas aguas arriba. El equipo de Hennessy vieron estos bloqueos como una barrera importante el rendimiento, ya que tenían que comunicar a todos los módulos de la CPU que lleva tiempo, y parecía limitar la velocidad de reloj. Un aspecto importante del diseño de MIPS era para adaptarse a cada sub-fase, tales como memoria caché de acceso, de todas las instrucciones en un ciclo, eliminando de este modo cualquier necesidad de enclavamiento, y permitiendo que un solo ciclo de rendimiento.

Aunque este diseño elimina un número de instrucciones útiles, tales como multiplicar y dividir se estimó que el rendimiento global del sistema se podría mejorar drásticamente debido a que los chips podrían funcionar a frecuencias de reloj mucho más altas. Esta rampa de la velocidad sería difícil con enclavamiento involucrados, como el tiempo necesario para establecer bloqueos es tanto una función del tamaño de la pastilla como velocidad de reloj. La eliminación de estas instrucciones se convirtió en un punto discutible.



## UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

### Laboratorio de Languages de programación

La otra diferencia entre el diseño MIPS y Berkeley RISC competir involucrado el manejo de llamadas a subrutinas. RISC utiliza una técnica llamada ventanas de registro para mejorar el rendimiento de estas tareas muy comunes, pero esta limitada la profundidad máxima de las llamadas de multi-nivel. Cada subrutina llamada requiere su propio conjunto de registros, que a su vez requieren más espacio en el CPU y una mayor complejidad en su diseño. Hennessy consideró que un compilador cuidadoso podría encontrar registros libres sin tener que recurrir a una implementación de hardware, y que simplemente aumentando el número de registros no sólo haría que este sencillo, pero aumentar el rendimiento de todas las tareas.

En otros aspectos el diseño MIPS fue en gran medida un típico diseño RISC. Para guardar los bits de la palabra de instrucción, diseños RISC reducir el número de instrucciones para codificar. El diseño MIPS utiliza 6 bits de la palabra de 32 bits para el código de operación básico, y el resto puede contener una dirección de salto solo 26 bits o puede tener hasta cuatro campos de 5 bits que especifica hasta tres registros además de un valor de cambio combinado con otros 6 bits de código de operación; otro formato, entre varios, especifica dos registros combinados con un valor inmediato de 16 bits, etc Esto permitió a esta CPU para cargar hasta la instrucción y los datos que se necesitan en un solo ciclo, mientras que un no-diseño RISC, tales como el MOS Technology 6502, por ejemplo, requiere ciclos separados para cargar el código de operación y los datos. Esta fue una de las mejoras de rendimiento importantes que RISC ofrece. Sin embargo, los diseños modernos de no-RISC lograr esta velocidad por otros medios [2].

#### 4. MATERIALES, EQUIPOS, SOFTWARE

- Ordenador
- QTSPIM
- Editor de texto Sublime o Notepad ++.

#### 5. PROBLEMAS PROPUESTOS

1. Implementar un programa que lea dos números enteros y muestre la suma.
2. Implementar un programa que pida el ingreso de un número por teclado, si es mayor a 10, mostrara el texto "MAYOR", y mostrará el texto "MENOR" en caso contrario.



**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA**  
**Laboratorio de Lenguajes de programación**

3. Implementar un programa que muestre el promedio de dos números enteros ingresados por teclado.

## **6. BIBLIOGRAFÍA Y FUENTES PARA CONSULTAS**

- [1] Documentación SPIM, <http://www.dsi.unive.it/~archit/~/LAB/spim.htm>, extraído el 13-03-2019
- [2] Arquitectura MIPS, <http://arquitecturadelcomputadorsemestre6.blogspot.com/2014/10/procesadores-mips.html>, extraído el 13-03-2019