

Universidad de Guadalajara

Piensa y trabaja



Procesador MIPS

Equipo 6

Fase 1

- Alan Emmanuel Marin Lemus
- Denice Estefania Rico Morones
- Santiago Rafael Gomez Brisuela
- Ferran Prado Roesner

Turno: Matutino

Fecha: 22/04/24

Universidad de Guadalajara

Piensa y trabaja

Título: Data Path

Introducción:

Teniendo conocimiento sobre lo qué es un procesador tipo MIPS, comenzaremos con la construcción del “Data Path” el cual nos ayudará a decodificar un conjunto de instrucciones tipo **r**.

Esto estará conformador por:

- ALU (Unidad Aritmética Lógica).
- ALU Control.
- Memoria (Datos).
- Banco de Registros.
- Unidad de Control.
- Conjunto de instrucciones **32 bits**.

Ejecutaremos las siguientes instrucciones:

ADD, SUB, AND, OR Y SLT.

Desarrollo:

Un conjunto de instrucciones tipo **r** consta de 32 bits, las cuales son **Aritmético-Lógicas**; se dividen en las siguientes partes:

- Opcode (6 bits): código de operación.
- RS (5 bits): espacio de memoria donde se almacenará el primer operando.
- RT (5 bits): espacio de memoria donde se almacenará el segundo operando.
- RD (5 bits): indica el espacio de memoria donde se guardarán los datos.
- Shamt (5 bits): operaciones de desplazamiento.
- Funct (6 bits): identificador de la operación aritmética a realizar.

Universidad de Guadalajara

Piensa y trabaja

SLT como operación Aritmética-Lógica (Comparación):

(si ($rs < rt$) **entonces** ($rd = 1$)

en otro caso ($rd = 0$)).

Operador Condicional Ternario:

Evalúa una condición y retorna el valor de una función.

```
resultado = condición ? valor_si_cierto : valor_si_falso
```

Es equivalente a un **IF-ELSE**, esto evalúa un valor booleano lo que contiene:

- True o False.
- Variable Booleana.
- Llamada a una función que devuelve un booleano.

?: si es verdad devuelve el valor verdadero.

:: "si no" devuelve el valor falso.

Proceso de Compilación:

El proceso de compilación consiste en generar un código objeto que es equivalente a un programa fuente, esto solo ocurre cuando el archivo fuente esta libre de errores.

Cuando se habla de código objeto se trata de código en ensamblador o lenguaje máquina.

Esto se divide en distintas fases o pasos:

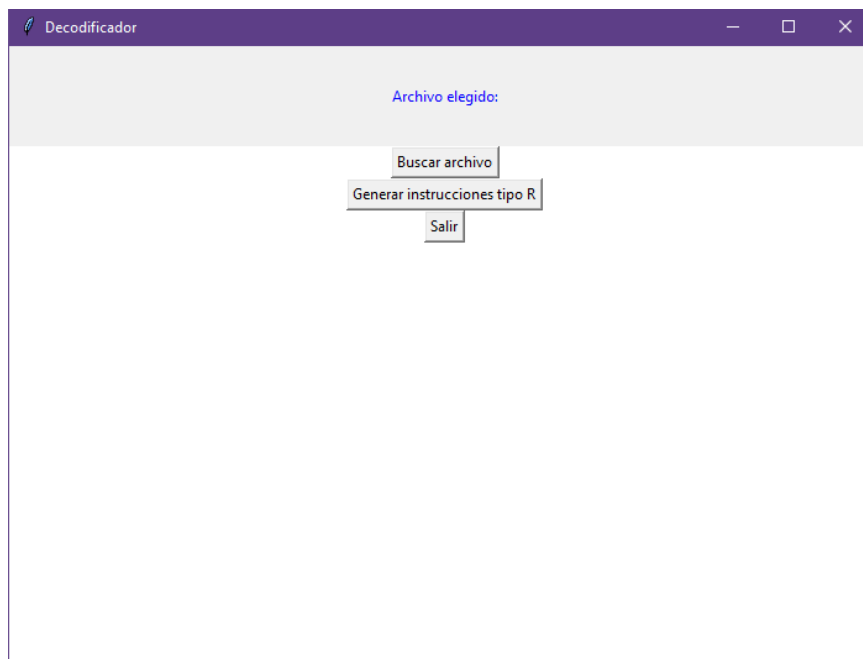
- 1) *Preprocesador: procesa el código fuente antes de su compilación.*
- 2) *Compilador: transforma el archivo fuente en lenguaje ensamblador.*
- 3) *Ensamblador: del código ensamblador lo traduce a binario.*
- 4) *Linker: ensambla los objetos para crear un ejecutable.*

Universidad de Guadalajara

Piensa y trabaja

1era Fase:

El decodificador es un programa hecho con Python utilizando la librería tkinter para crear un interfaz intuitivo para el usuario. En este programa podemos seleccionar cualquier archivo .asm de nuestro PC para posteriormente decodificar las instrucciones tipo R que este contenga. Esto se hace mediante la lectura de un archivo guardando su ruta, la lectura del archivo se hace por líneas, posteriormente crea un archivo de texto instrucciones_r.txt en modo escritura para que se le permita añadir datos. Las instrucciones de las líneas leídas se separan por partes, las instrucciones cambian a binario y el diccionario decodifica la función, se terminan de procesar todas las instrucciones y se almacenan en el archivo de tipo texto.



Conclusión:

Es muy importante conocer estos conceptos ya que nos ayudan a comprender lo que es el procesador que estamos construyendo, son herramientas que son

Universidad de Guadalajara

Piensa y trabaja

indispensables en la programación, así como conceptos básicos que nos facilitarán la implementación de todo esto.

Bibliografía:

Práctica 13. La unidad aritmético lógica (ALU). (s. f.).

https://www.hpca.ual.es/~vruiz/docencia/laboratorio_arquitectura/practicas/practica13/

Llamas, L. (2023, 1 junio). El operador ternario. Luis Llamas.

<https://www.luisllamas.es/programacion-operador-ternario/>

Prezi, M. C. O. (s. f.). PROCESO DE COMPILACION. prezi.com.

<https://prezi.com/qqn12utcx1yu/proceso-de-compilacion/>

Manual de Dev-C++ en español. (s. f.).

https://www.tel.uva.es/personales/josdie/fprog/Sesiones/manualDevCpp/compilacion_y_linkado.html