

UNIVERSIDAD DE SANTIAGO DE CHILE  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA



## Laboratorio 2

Integrantes: Raúl Andrés Olivares Pasten  
Curso: Organización de computadores  
Sección L3  
Profesor(a): Alfonso Guzman  
Daniel Wladdimiro

2 de Mayo de 2017

# Tabla de contenidos

|   |          |
|---|----------|
| <b>1. Introducción</b>                  | <b>1</b> |
| 1.1. Contexto . . . . .                 | 1        |
| 1.2. Problema . . . . .                 | 1        |
| 1.3. Motivación . . . . .               | 1        |
| 1.4. Objetivos . . . . .                | 1        |
| 1.5. Propuesta de solución . . . . .    | 2        |
| 1.6. Herramientas . . . . .             | 2        |
| 1.7. Estructura del informe . . . . .   | 2        |
| <b>2. Marco teórico</b>                 | <b>3</b> |
| 2.1. Pipeline . . . . .                 | 3        |
| 2.2. Hazard de Datos . . . . .          | 3        |
| 2.3. Hazard de Control . . . . .        | 4        |
| <b>3. Desarrollo</b>                    | <b>5</b> |
| 3.1. Elaboración del programa . . . . . | 5        |
| 3.2. Experimento y Resultados . . . . . | 5        |
| 3.3. Análisis . . . . .                 | 6        |
| <b>4. Conclusiones</b>                  | <b>7</b> |
| <b>Bibliografía</b>                     | <b>8</b> |

# 1. Introducción

## 1.1. Contexto

A partir de la materia vista en la cátedra para curso de organización de computadores lo mas esencial es el camino de datos(datapath) que lleva la información necesaria a través de las unidades funcionales para ejecutar una instrucción de forma correcta. Ampliando más el tema del camino de datos en este laboratorio se aplicara los buffers en el camino de datos que son necesarios para realizar ejecutar correctamente las instrucciones en un procesador con segmentación(pipeline).

## 1.2. Problema

El programa a construir debe recibir dos archivos de entrada, uno con un programa en lenguaje MIPS y otro archivo con los valores de cada uno de los registros. Poder ejecutar el archivo MIPS como si se tratase de un procesador con pipeline y forwarding y que genere dos archivos de salida, uno con la traza de los registros por cada ciclo de reloj y el otro con los valores que tienen los buffers en cada ciclo de reloj.

## 1.3. Motivación

Con el tiempo además de aumentar la potencia del procesador se a buscado mejorar la eficiencia de este, a través de distintas medidas, una de ellas la segmentación que consiste en descomponer la ejecución de cada instrucción en varias etapas y trabajar con varias de ellas a la vez aumento la eficiencia del procesador. El motivo de este laboratorio es apreciar de forma practica la importancia de este método por parte del alumno y otros lectores de este documento.

## 1.4. Objetivos

- Objetivo general: Aplicar los conceptos de forwarding y pipeline aprendidos en clase implementado un programa que represente y realice las acciones de un procesador con pipeline y forwarding.

- Objetivos específicos:

1. Aplicar correctamente el uso del camino de datos en el procesador.
2. Poder implementar el concepto de buffer y que estos almacenen información entre etapas.
3. Ejecutar un programa escrito en MIPS correctamente y con la capacidad de usar forwarding para obtener los valores correctos de los registros antes que estos sean actualizados.

## **1.5. Propuesta de solución**

Como solución para el problema se decidió realizar un programa en lenguaje C que cumpliera las características requeridas, creando estructuras de datos para representar los registros y memoria de un procesador y los distintos buffers que tiene este, con la opción de a partir de los archivos entregados ejecutar varias instrucciones en un mismo ciclo y ir imprimiendo los valores requeridos en los archivos de salida con formatos de tablas.

## **1.6. Herramientas**

- El programa se desarrollo usando el lenguaje de programación C.
- Sistema operativo GNU/Linux

## **1.7. Estructura del informe**

El siguiente informe esta estructurado en capítulos siguiendo la siguiente forma, empezando por la Introducción del documento, relatando de que se abordara este informe y los objetivos que se buscan cumplir en este laboratorio. Un marco teórico documentando los conceptos generales que se deben dominar y que son necesarios para entender el problema. A continuación sigue el capitulo de Desarrollo con la explicaciones de por que se desarrollo el programa así y la forma de este, además de la ejecución del programa y los resultados que este dio, para luego finalizar con una Conclusión observando si se cumplieron o no los objetivos planteados anteriormente.

## 2. Marco teórico

### 2.1. Pipeline

La segmentación consiste en descomponer la ejecución de cada instrucción en varias etapas para poder empezar a procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez (Wikipedia, 2017c). Nos podemos encontrar las siguientes Etapas de ejecución en un procesador:

- Instruction fetch(IF): Búsqueda de la instrucción.
- Instruction decode and register fetch(ID): Decodificación de la instrucción.
- Execute(EX): Ejecución en la unidad aritmético lógica.
- Memory access(MEM): Búsqueda o escritura en memoria.
- Register write back(WB): Escritura de registros.

Cada una de estas etapas de la instrucción usa en exclusiva un hardware determinado del procesador, de tal forma que la ejecución de cada una de las etapas en principio no interfiere en la ejecución del resto, dando como resultado que se pueda iniciar la ejecución de la siguiente instrucción tras acabar la primera etapa de la instrucción actual.

### 2.2. Hazard de Datos

se conoce como dependencia de datos aquella situación en que las instrucciones de un programa se refieren a los resultados de otras anteriores que aún no han sido completadas (Wikipedia, 2017a).

En el caso del procesador con pipeline se da comúnmente la situación donde se necesita un dato que aún no ha sido calculado, por ejemplo:

1.  $R2 = R1 + R3$

2.  $R4 = R2 + R3$

En el ejemplo anterior se puede apreciar como la instrucción 2 ocupa el registro R2, el cual debe esperar a finalizar la instrucción 1 para tomar el valor correcto de este, este tipo de dependencia recibe el nombre de RAW(Read After Write).

### **2.3. Hazard de Control**

Los hazard de control ocurren cuando el procesador se ve obligado a saltar a una instrucción que no tiene por qué ser necesariamente la inmediatamente siguiente en el código. En ese caso, el procesador no puede saber por adelantado si debería ejecutar la siguiente instrucción u otra situada más lejos en el código(Wikipedia, 2017b).

Esto puede resultar en acciones no deseadas por parte de la CPU.

## 3. Desarrollo

### 3.1. Elaboración del programa

El programa consta de varios archivos .h y .c para ayudar a separar las funciones del programa, existe un archivo estructuras.h que posee todas las estructuras de datos, otro archivo llamado instrucción.c con las funciones para la estructura de datos Instrucción, así también existen archivos para separar las funciones de Memoria, Registros y Buffers. Mientras en el archivo principal main.c existen las funciones correspondientes a las Etapas y que actúan sobre los buffers igualmente, además de la función principal donde se interactúa con el usuario.

Estructura de Registros: Se decidió implementar las variables de registros como variables de tipo long, para así poder operar números y valores mayores si es que se llegaban a necesitar.

Branch no tomado: Se decidió que la siguiente instrucción a entrar al procesador en la etapa IF después de un branch sea la que se encuentra en la siguiente posición del PC y que si en la etapa MEM se toma que la condición del branch se cumple ahí saltar al PC correspondiente, agregando nop's en los buffers anteriores(IF/ID y ID/EX)

Valores líneas de control: ya que algunas líneas de control realizan una opción con el valor 0 y otra con el valor 1, se decidió tomar para todas las líneas de control el valor 2 como valor nulo.

Nombre de la instrucción en el archivo de salida: Por comodidad se decidió agregar a la estructura de datos de cada buffer una variable con el nombre de la instrucción, esta variable no influye en el camino de datos, ni en las etapas de ciclo de reloj, solo esta para que sea más fácil de imprimir en el archivo con los valores de los buffer en cada ciclo y así saber a que instrucción corresponde los valores que toma cada buffer.

### 3.2. Experimento y Resultados

A continuación se realizara una prueba de como funciona el programa, ingresando los archivos de prueba entregados en el moodle de usach virtual, inputCode.asm con el código MIPS del programa y el archivo inputRegistros.txt

A continuación una imagen del archivo de salida que muestra los valores de los registros en los ciclos de reloj iniciales1.

**Tabla registros.**

| Ciclo | PC | zero | \$at | \$v0 | \$v1 | \$a0 | \$a1 | \$a2 | \$a3 | \$t0 | \$t1 | \$t2 | \$t3       | \$t4 | \$t5 | \$t6    | \$t7       | \$s0     | \$s1       | \$s2 | \$s3 | \$s4     |
|-------|----|------|------|------|------|------|------|------|------|------|------|------|------------|------|------|---------|------------|----------|------------|------|------|----------|
| C1    | 4  | 0    | 0    | 15   | 0    | 1386 | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C2    | 84 | 0    | 0    | 15   | 0    | 1386 | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C3    | 88 | 0    | 0    | 15   | 0    | 1386 | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C4    | 8  | 0    | 0    | 15   | 0    | 1386 | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C5    | 12 | 0    | 0    | 15   | 0    | 1386 | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C6    | 16 | 0    | 0    | 15   | 0    | 6    | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C7    | 20 | 0    | 0    | 15   | 0    | 6    | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C8    | 24 | 0    | 0    | 15   | 0    | 6    | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C9    | 28 | 0    | 0    | 15   | 0    | 6    | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C10   | 32 | 0    | 0    | 15   | 0    | 6    | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C11   | 36 | 0    | 0    | 15   | 0    | 6    | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C12   | 8  | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 7    | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C13   | 12 | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 12   | 105  | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C14   | 16 | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 12   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C15   | 20 | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 12   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C16   | 24 | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 12   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C17   | 28 | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 12   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C18   | 32 | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 12   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C19   | 36 | 0    | 0    | 15   | 0    | 5    | 3    | 0    | 0    | 12   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C20   | 8  | 0    | 0    | 15   | 0    | 4    | 3    | 0    | 0    | 12   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C21   | 12 | 0    | 0    | 15   | 0    | 4    | 3    | 0    | 0    | 16   | 17   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C22   | 16 | 0    | 0    | 15   | 0    | 4    | 3    | 0    | 0    | 16   | 20   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |
| C23   | 20 | 0    | 0    | 15   | 0    | 4    | 3    | 0    | 0    | 16   | 20   | 0    | 2433469440 | 0    | 0    | 4515151 | 3722304989 | 20488191 | 4294901760 | 0    | 8    | 85976270 |

Figura 1: Archivo de salida: valores de registros.

|     |     |   |   |     |   |   |   |   |   |    |    |   |            |   |   |         |            |          |            |   |   |          |
|-----|-----|---|---|-----|---|---|---|---|---|----|----|---|------------|---|---|---------|------------|----------|------------|---|---|----------|
| C92 | 96  | 0 | 0 | 720 | 0 | 6 | 3 | 0 | 0 | 22 | 22 | 0 | 2433469440 | 0 | 0 | 4515151 | 3722304989 | 20488191 | 4294901760 | 0 | 8 | 85976270 |
| C93 | 100 | 0 | 0 | 720 | 0 | 6 | 3 | 0 | 0 | 22 | 22 | 0 | 2433469440 | 0 | 0 | 4515151 | 3722304989 | 20488191 | 4294901760 | 0 | 8 | 85976270 |
| C94 | 104 | 0 | 0 | 720 | 0 | 6 | 3 | 0 | 0 | 22 | 22 | 0 | 2433469440 | 0 | 0 | 4515151 | 3722304989 | 20488191 | 4294901760 | 0 | 8 | 85976270 |
| C95 | 108 | 0 | 0 | 720 | 0 | 6 | 3 | 0 | 0 | 22 | 22 | 0 | 2433469440 | 0 | 0 | 4515151 | 3722304989 | 720      | 4294901760 | 0 | 8 | 85976270 |

Figura 2: Valores de registros: Ultimo ciclo de reloj

### 3.3. Análisis

Como primer punto de la experiencia se aprecia que se ejecuto correctamente el programa dando como resultado del factorial de 6 el valor en el registro v0 de 720.

No es seguro si es un error que se produce por el tamaño de la variable que guarda el registro, pero en los buffer de datos en las ocasiones que interactúa el registro Zero, aparece con otro registro añadido seguido de este, por ejemplo en el ciclo C56, en el buffer ID.EX aparece Rs: \$zero\$v0 aunque esto no afecta el funcionamiento del programa, ya que al buscar el valor de un registro X, este solo revisa los primero elementos del string registro.



## 4. Conclusiones

El programa construido pudo implementar correctamente las funciones requeridas, logrando que el programa pueda ejecutar correctamente las instrucciones del programa dado, pudiendo trabajar una instrucción diferente en cada etapa del procesador al utilizar la segmentación(pipeline) de este. Uno de los problemas de aplicar la segmentación es la necesidad de obtener valores que todavía no estaban actualizados en los registros, para esto se logro implementar utilizando los buffers del procesador el método de forwarding para obtener valores que no estén todavía guardados en los registros y así lograr ejecutar el programa correctamente.

# Bibliografía

Wikipedia (2017a). Dependencia de datos. [Online] [https://es.wikipedia.org/wiki/Dependencia\\_de\\_datos](https://es.wikipedia.org/wiki/Dependencia_de_datos).

Wikipedia (2017b). Hazard (computer architecture). [Online] [https://en.wikipedia.org/wiki/Hazard\\_\(computer\\_architecture\)](https://en.wikipedia.org/wiki/Hazard_(computer_architecture)).

Wikipedia (2017c). Instruction pipelining. [Online] [https://en.wikipedia.org/wiki/Instruction\\_pipelining](https://en.wikipedia.org/wiki/Instruction_pipelining).