

# Implementación procesador MIPS

## Facultad de Ciencias Exactas, Físicas y Naturales

Sardoy, Juan Manuel<sup>1</sup>, Perez, Federico<sup>1</sup>

<sup>1</sup>Arquitectura de computadoras - FCEFYN  
Av. Velez Sarsfield 1600 – Córdoba – Argentina

jmsardoy@gmail.com, 0xfede07c8@gmail.com

**Resumen.** Este documento describe, de manera sintética, la implementación práctica del pipeline de un procesador que cumple con un subconjunto del set de instrucciones de la arquitectura MIPS IV de 32 bits. La implementación está basada en tecnología FPGA, mediante el lenguaje Verilog. Este trabajo tiene como objeto el aprendizaje de los diferentes elementos más comunes de la arquitectura de computadoras, dado el encuadre académico del mismo.

**Palabras clave:** MIPS, FPGA, pipeline.

**Abstract.** This document describes, in a synthetic way, a practical implementation of the pipeline of a processor that is compliant with a subset of the MIPS IV 32bits architecture instruction set. This job has as objective, to build the basic knowledge of the computer's architecture cope of study.

**Keywords:** MIPS, FPGA, pipeline.

## 1. Introducción

El trabajo, se encuentra dentro del marco de la asignatura de Arquitectura de Computadoras, de la FCEFYN de la Universidad Nacional de Córdoba, Argentina. Corresponde al trabajo final de la misma, el cual es requerimiento para su aprobación.

Además de la implementación del procesador en sí, se mostrará como implementar los elementos auxiliares requeridos para su funcionamiento y programación, entre otras utilidades. El hardware utilizado será un dispositivo FPGA, de la marca Xilinx, al ser de esta empresa, con los que cuentan los autores del trabajo.

Dentro de los elementos auxiliares, encontraremos el desarrollo de un sistema de comunicación directo con el procesador, mediante el protocolo UART. Además de éste, un mecanismo para cargar un programa en la RAM del procesador, como así también funcionalidad de debugging (ejecución paso a paso y salida de valores de registros internos). También se mostrará la implementación y uso de software auxiliar para el ensamblado de los programas a ejecutar.

Todo esto compondrá una suite básica de desarrollo de software de propósito general, cuyo entendimiento es el objetivo principal del trabajo y la materia.

## 2. Referencias Teóricas

La técnica de *pipeline* es la más común utilizada para la implementación de procesadores modernos de diferentes arquitecturas, dado el gran aumento de *IPC* (instrucciones por ciclo) que éste provee.

MIPS son las siglas de *Microprocessor without interlocked pipeline stages*. La primera versión de ésta arquitectura fue desarrollada en la Universidad de Stanford en 1981 por un equipo liderado por John. L. Hennessy. La idea era diseñar un procesador segmentado (idea ya muy conocida en esos momentos) pero sin interbloqueo entre capas. En esa época eran comunes los procesadores donde cada instrucción iba ejecutandose por cada capaz, mientras las otras permanecian inactivas. Esto llevaba a que el tiempo que demora en ejecutar la instrucción fuera el del tiempo en el que tarda en pasar por todas las capas, en contraste con el tiempo del camino crítico, como ocurre sin interbloqueo.

Por esto mismo este trabajo tendrá como cuerpo principal, la aplicación de dicha técnica, para lo cual, a modo de guía, se seguirá la bibliografía de *Patterson y Hennessy*, dado que expone y soluciona los problemas más comunes a la hora de la creación de un *pipeline* genérico. Sin embargo, muchas partes de la implementación, quedan a cargo, y serán solucionadas por los autores, lo cual lo hace una implementación única de la arquitectura.

### 3. Metodología

Se utilizará una metodología incremental y modular para la implementación del pipeline y sus módulos complementarios. Esto mejora el entendimiento y fragmentación del problema, así como también facilita su implementación, dado la complejidad del mismo. También esto es muy útil a la hora del testing y el debugado del sistema.

### 4. Requerimientos del trabajo

Los requerimientos provistos por la cátedra fueron los siguientes:

1. Implementar el pipeline de un procesador MIPS, segmentado en las siguientes etapas:
  - a) IF (Instruction Fetch): Búsqueda de la instrucción en la memoria de programa.
  - b) ID (Instruction Decode): Decodificación de la instrucción y lectura de registros.
  - c) EX (Execute): Ejecución de la instrucción propiamente dicha.
  - d) MEM (Memory Access): Lectura o escritura desde/hacia la memoria de datos.
  - e) WB (Write back): Escritura de resultados en los registros.
2. En dicha arquitectura, implementar las siguientes instrucciones:
  - a) R-type (SLL, SRL, SRA, SLLV, SRLV, SRAV, ADDU, SUBU, AND, OR, XOR, NOR, SLT)
  - b) I-Type (LB, LH, LW, LWU, LBU, LHU, SB, SH, SW, ADDI, ANDI, ORI, XORI, LUI, SLTI, BEQ, BNE, J, JAL)
  - c) J-Type (JR, JALR)
3. Debe poseer soporte (detección y manejo) de los siguientes tipos de riesgos:
  - a) Estructurales: Cuando dos o más instrucciones tratan de utilizar el mismo recurso en el mismo ciclo.
  - b) Datos: Una etapa desea utilizar un dato antes de que esté listo. Se debe mantener el orden estricto de las lecturas y escrituras.
  - c) Control: Capacidad para tomar una decisión sobre una condición no evaluada.

As Tabelas, Quadros e Figuras, assim como as legendas de Tabelas, Quadros e Figuras devem estar centralizadas se conterem apenas em uma linha (Figura 1), caso contrário devem estar tabuladas em 0.8cm em ambas as margens, como mostra a Figura 3.



Figura 1. Exemplo de figura

As legendas devem ser escritas na fonte “Helvetica”, Tamanho 10pts, negrito, com espaço de 6pts antes e depois de cada legenda. Sempre que possível, procure colocar a figura delimitada por um quadro (Figura 1 e Figura 3)

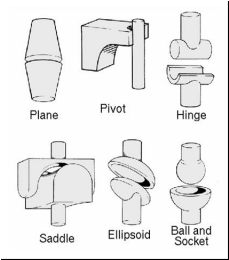


Figura 2. Essa figura foi referenciada na Seção ??.

Figura 3. Fonte: SBC.

Em tabelas, tente evitar o uso de fundos coloridos ou preenchidos, assim como linhas duplas na borda, ou linhas desnecessárias. Quando [David A. Patterson 2009] relatar dados empíricos, não faça uso de mais dígitos decimais do que o necessário. A legenda da tabela deve ser colocada antes da tabela (veja Tabela 1) e a fonte usada na legenda deve ser Helvetica, tamanho 10pts, negrito, com 6pts de espaço antes e depois de cada legenda.

Cuadro 1. Exemplo de tabela de 3 colunas e 2 linhas

|        | Value 1       | Value 2                                    |
|--------|---------------|--|
| Case 1 | $1.0 \pm 0.1$ | $1.75 \times 10^{-5} \pm 5 \times 10^{-7}$ |
| Case 2 | 0.003(1)      | 100.0                                      |

4.1. Código fonte

A inserção de código fonte deve ser por meio

```
1 |
2 | int main() {
```

```
3 | int a,b,c;  
4 | float x;  
5 | printf("informe o tamanho do lado do quadrado");  
6 | scanf("%d", &a);  
7 | printf("A area do quadrado %d", b=area(a));  
8 | printf("Duas vezes o valor do lado do quadrado %d", c=aumenta(a  
   | ));
```

## 5. Considerações Finais

Referências bibliográficas devem ser utilizadas dentro de um estilo uniforme e não ambíguo. A SBC sugere os seguintes formatos para referências: [David A. Patterson 2009], [?], e [?].

### Referencias

David A. Patterson, J. L. H. (2009). *Computer Organization and Design*. Morgan-Kaufmann, 4th edition.