
**Universidade Federal da Paraíba**
**Centro de informática - CI**

---

# *Arquitetura de Computadores*
# *Trabalho Final*

***Guilherme Moreira - 20160105205***
***Luciana Serrão e Silva - 2016085380***

**2019**

## Especificações.:

Suponha que houvesse uma instrução MIPS do tipo I, chamada **sdw** (store double word), que fizesse uma escrita duplicada de uma palavra em dois endereços consecutivos da memória (por exemplo, os bytes de 3-6 – primeira cópia – e os bytes de 7-10 – segunda cópia). O endereço onde o armazenamento deverá iniciar estará armazenado no registrador apontado no campo **rs** da instrução, e o dado a ser gravado estará no registrador apontado no campo **rt** da instrução. Perceba que essa instrução é uma variante da instrução **sw** (é importante que você entenda bem como funciona sw). Um exemplo de uso dessa instrução seria **sdw $t2, $t1(5)**, considerando que o registrador **$t1** (campo rs) contém o endereço base onde a gravação deve começar, 5 é um offset de 16 bits que será somado com o valor do registrador base para a obtenção do endereço inicial da escrita, e o registrador **$t2** (campo **rt**) é aquele que armazenará a palavra a ser escrita de forma duplicada. Ainda de acordo com esse exemplo, a segunda palavra a ser escrita deverá iniciar no endereço **$t1 + 5 + 4**.

## Escreva um relatório que inclua pelo menos os seguintes itens:

• Se foi necessária alguma alteração no caminho de dados MIPS estudado em sala de aula;

• O diagrama de estados relativo ao controle da instrução **sdw** (os estados **não** necessitam conter os valores exatos para os sinais de controle – eles podem conter uma descrição textual do “efeito” que o estado deve alcançar);

• A relação de microinstruções para a implementação da instrução **sdw**

• Descrição geral da solução adotada (escolha das microinstruções, possíveis consequências para a implementação de outras instruções, etc.)

## Caminho de dados MIPS.:

| Instruction opcode | ALUOp | Instruction operation | Funct Field | Desired ALU action | ALU control input |
|---|---|---|---|---|---|
| LW | 00 | load word | XXXXXX | add | 0010 |
| SW | 00 | store word | XXXXXX | add | 0010 |
| SDW | 00 | store double word | XXXXXX | add | 0010 |
| Branch equal | 01 | branch equal | XXXXXX | subtract | 0110 |
| R-type | 10 | add | 100000 | add | 0010 |
| R-type | 10 | subtract | 100010 | subtract | 0110 |
| R-type | 10 | AND | 100100 | and | 0000 |
| R-type | 10 | OR | 100101 | or | 0110 |
| R-type | 10 | set on less than | 101010 | set on less than | 0111 |

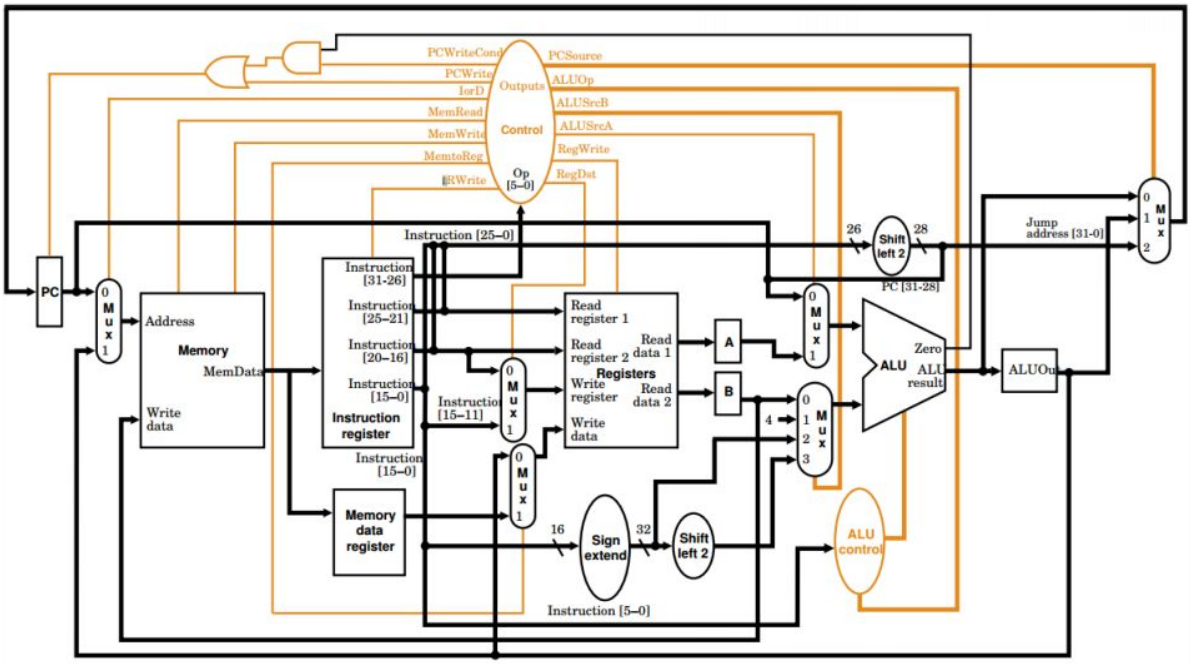
Caminho de dados é a parte do processador que contém a **ULA** e todas as suas **entradas** e **saídas**.
Tivemos que acrescentar a instrução **sdw** na tabela de caminho de dados **MIPS,** pois **sdw** será uma nova entrada da **ULA**.

## Relação de Microinstruções.:

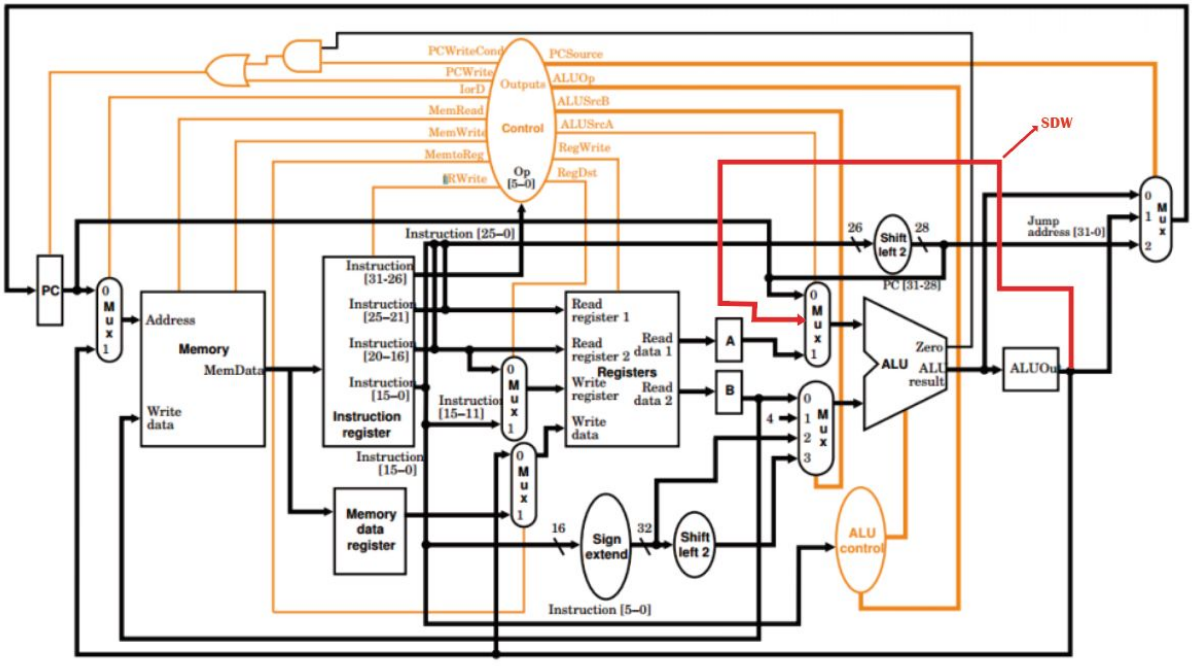
| Label | ALU Control | SRC1 | SRC2 | RegisterControl | Memory | PCWrite Control | Sequencing |
|---|---|---|---|---|---|---|---|
| *Fetch* | *Add* | *PC* | *4* | | *Read PC* | *ALU* | *Seq* |
| | *Add* | *PC* | *Extshft* | *Read* | | | *Dispatch 1* |
| *Mem1* | *Add* | *A* | *Extend* | | | | *Dispatch 2* |
| *LW2* | | | | | *Read ALU* | | *Seq* |
| | | | | *Write MDR* | | | *Fetch* |
| *SW2* | | | | | *Write ALU* | | *Fetch* |
| *Rformat 1* | *func code* | *A* | *B* | | | | *Seq* |
| | | | | *Write ALU* | | | *Fetch* |
| *BEQ1* | *Subt* | *A* | *B* | | | *ALUOut-cond* | *Fetch* |
| *JUMP1* | | | | | | *JUMP Address* | *Fetch* |
| *SDW* | | *ALUOut* | | *Wrtite MDR* | | *ALU* | *Fetch* |

Em SRC1 foi adicionado **ALUOut**, pois iremos pegar a primeira **saída** da ALU e adicionar novamente a sua **entrada**, porém somando mais **4 bytes** no endereço da memória para que ele não seja o mesmo que o anterior.
Como **sdw** é uma instrução de **escrita** duplicada que irá armazenar o primeiro valor da saída da ALU, usamos **WriteMDR** que irá escrever tal valor no campo **rt** do banco de registradores.

## Microarquitetura original:




## Microarquitetura com SDW:

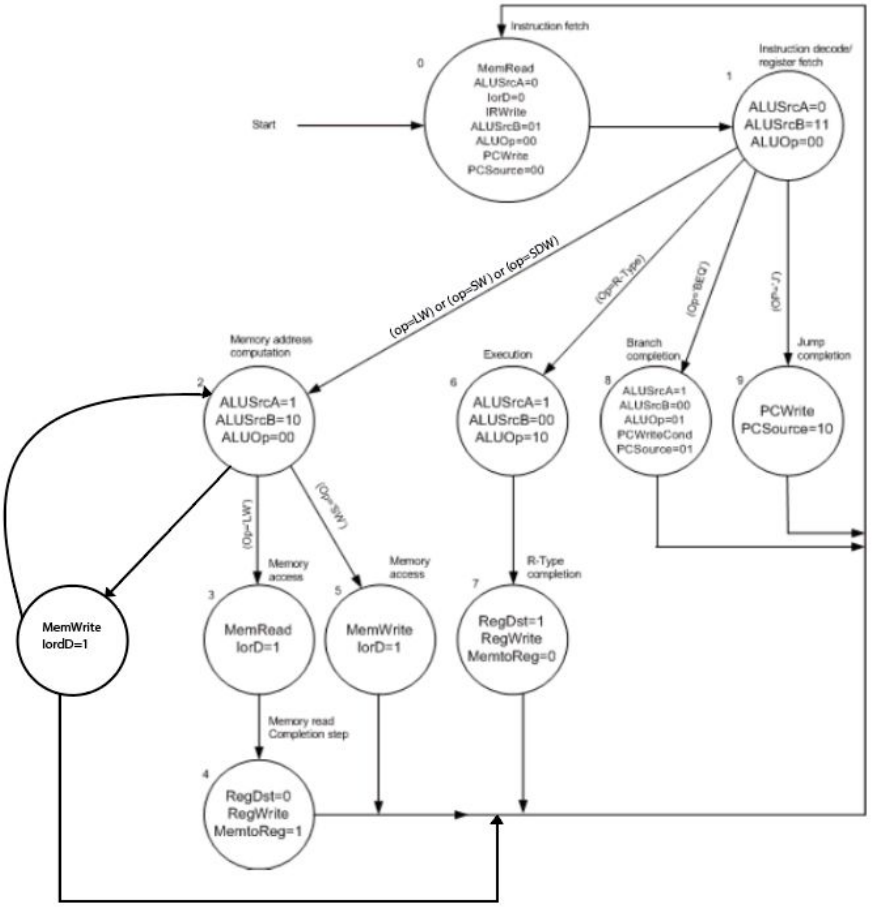


## Instrução SDW.:

A instrução SDW (store double word) é uma instrução MIPS do tipo I, que realiza uma escrita duplicada de uma palavra em dois endereços de memória diferentes.

O seu funcionamento na microarquitetura é o seguinte, inicialmente é pego o primeiro valor da saída da **ALUout** (valor total que foi processado pela ULA), esse valor é reinserido no primeiro **mux** (que passou de **2x1** pra **3x1**), tal valor é armazenado no campo **rt** do banco de registradores, o endereço da instrução está apontado no campo **rs.** Durante cada processo iremos somar **4 bytes** no campo **rs** do banco de registrador para que o endereço da instrução mude e não sobrescreva a mensagem. Sendo assim a **ALUout** irá ser atualizada com o mesmo valor anterior porém em um endereço de memória diferente.

## Diagrama de Estados.:



## Referências.:


1. *SLIDES APRESENTADOS EM AULA*
2. *Organização e Projeto de Computadores" (3 ed.)*