Instituto Tecnológico de Estudios Superiores de Occidente – ITESO

ITESO

Universidad Jesuita
de Guadalajara

Materia: Arquitectura de computadoras – Primavera 2019

Maestro: Rodrigo Aldana López

Fecha: 26 de abril de 2019

Tarea 11

Autor(es): Díaz Meda Ricardo Ian - IS710007
Ávalos Guzmán Luis Joaquín - IS708601

Modificaciones hechas

Las modificaciones que hicimos fue primeramente cambiar en el módulo de Register lo siguiente:

```
module Register
#(
    parameter N=32
)
(
    input clk,
    input reset,
    input enable,
    input  [N-1:0] DataInput,


    output reg [N-1:0] DataOutput
);

always@(negedge reset or negedge clk) begin
    if(reset==0)
        DataOutput <= 0;
    else
        if(enable==1)
            DataOutput<=DataInput;
end

endmodule
```

Esto con la finalidad de arreglar el Hazard estructural en la memoria.

Además, agregamos el siguiente módulo llamado Pipeline para poder hacer la división de las rutas críticas:

```
//asereje

module Pipeline
#(
    parameter N = 32
)
(
    // Inputs
    input clk,
    input reset,
    input enable,
    input nopper,
    input      [N-1:0] dataInput,
    output reg [N-1:0] dataOutput
);
always@(negedge reset or posedge clk or posedge nopper or posedge enable) begin // Para que escriba en flanco de subida
    if(reset==0 || nopper == 1)
        dataOutput <= 0;
    else
        dataOutput <= dataInput;
end

endmodule
```

Después añadimos las cuatro instancias necesarias para cada fase, añadimos bastantes wires para hacer el control de dichos Pipelines, y además cambiamos wires para que se adapten a la parte de la ruta crítica en la que están. Por ejemplo, cuando estamos en la parte que está en la ALU, le pasamos el wire de Alu_Result pero de la fase de EX.

```
//Pipeline IF/ID
Pipeline
#(
	.N(64)
)
Pipeline_IF_ID
(
	.clk(clk),
	.reset(reset),
	.enable(1'b1),
	.nopper(1'b0),
	.dataInput({Instruction_wire, PC_4_wire}),
	.dataOutput({ID_instruction_wire, ID_PC_4_wire})
);

Pipeline
#(
	.N(174)
)
Pipeline_ID_EX
(
	.clk(clk),
	.reset(reset),
	.enable(1'b1),
	.nopper(1'b0),
	.dataInput({RegDst_wire,
                            BranchNE_wire,
                            MemReadWire,
                            BranchEQ_wire,
                            MemWriteWire,
                            MemtoRegWire,
                            ALUOp_wire,
                            ALUSrc_wire,
                            RegWrite_wire,
                            jump_wire,
                            jal_wire,
                            ID_PC_4_wire,
                            ReadData1_wire,
                            ReadData2_wire,
                            InmmediateExtend_wire,
                            ID_instruction_wire
                            }),
	.dataOutput({EX_RegDst_wire,
                             EX_BranchNE_wire,
                             EX_MemReadWire,
                             EX_BranchEQ_wire,
                             EX_MemWriteWire,
                             EX_MemtoRegWire,
```

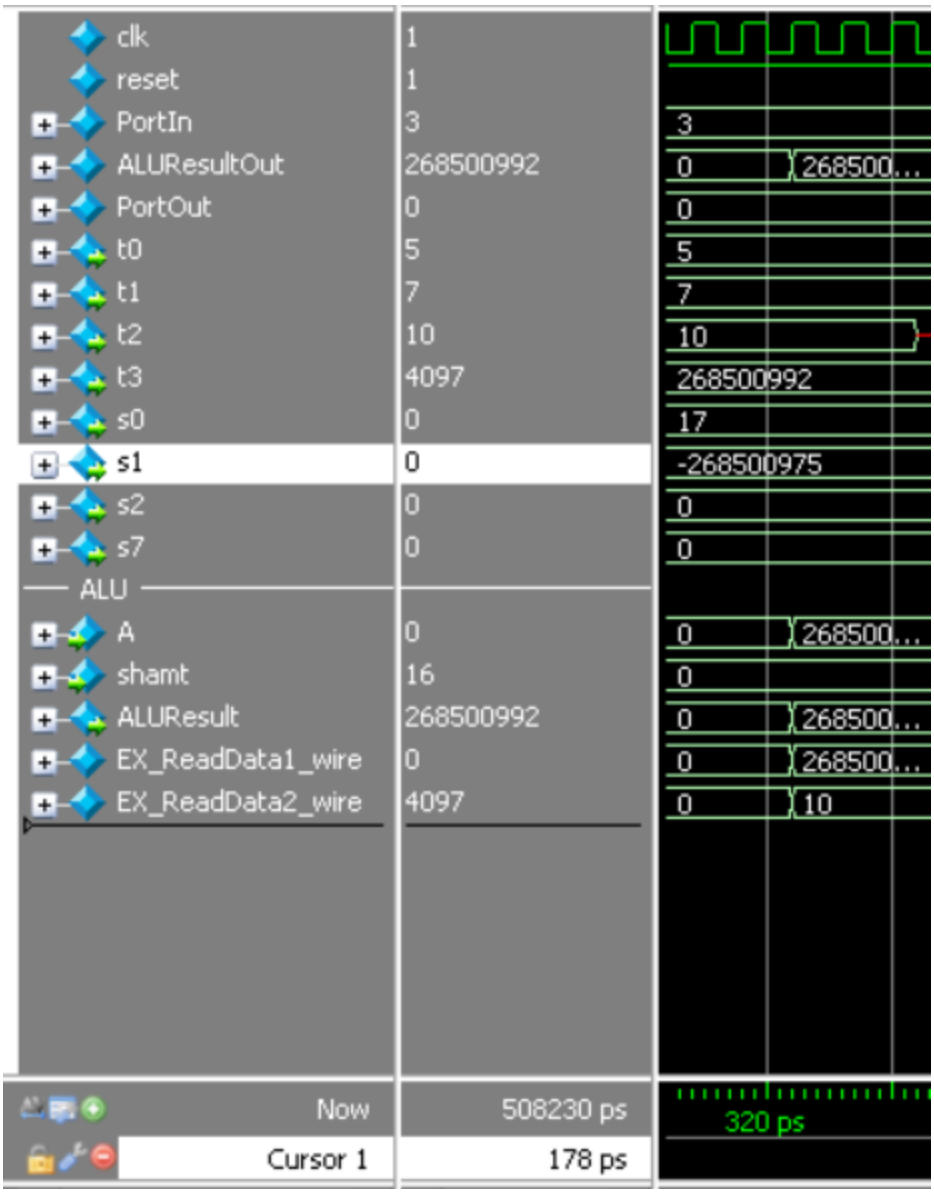
```
                                    EX_ALUOp_wire,
                                    EX_ALUSrc_wire, //
                                    EX_RegWrite_wire,
                                    EX_jump_wire,
                                    EX_jal_wire,
                                    EX_PC_4_wire,
                                    EX_ReadData1_wire,
                                    EX_ReadData2_wire,
                                    EX_InmmediateExtend_wire,
                                    EX_instruction_wire
                                    })
);


Pipeline
#(
      .N(137)
)
Pipeline_EX_MEM
(
      .clk(clk),
      .reset(reset),
      .enable(1'b1),
      .nopper(1'b0),
      .dataInput({
                        InmmediateExtendAnded_wire,
                        ALUResult_wire,
                        EX_ReadData2_wire,
                        WriteRegister_wire,
                        MUX_PC_wire,
                        EX_MemReadWire,
                        EX_MemWriteWire,
                        EX_MemtoRegWire,
                        EX_RegWrite_wire,
                        EX_jal_wire
                        }),
      .dataOutput({
                        MEM_InmmediateExtendAnded_wire,
                        MEM_ALUResult_wire,
                        MEM_ReadData2_wire,
                        MEM_WriteRegister_wire,
                        MEM_MUX_PC_wire,
                        MEM_MemReadWire,
                        MEM_MemWriteWire,
                        MEM_MemtoRegWire,
                        MEM_RegWrite_wire,
                        MEM_jal_wire})
);

//Pipeline MEM/WB
Pipeline
#(
      .N(105)
)
Pipeline_MEM_WB
(
      .clk(clk),
```

```
        .reset(reset),
        .enable(1'b1),
        .nopper(1'b0),
        .dataInput({
                            ramDataWire,
                            MEM_ALUResult_wire,
                            MEM_MemWriteWire,
                            MEM_MemtoRegWire,
                            MEM_RegWrite_wire,
                            MEM_jal_wire,
                            MEM_WriteRegister_wire,
                            MEM_MUX_PC_wire ////////////////////
                            }),
        .dataOutput({
                            WB_ramDataWire,
                            WB_ALUResult_wire,
                            WB_MemWriteWire,
                            WB_MemtoRegWire,
                            WB_RegWrite_wire,
                            WB_jal_wire,
                            WB_WriteRegister_wire,
                            WB_MUX_PC_wire})
);
```

Simulación ModelSIM

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	5
$t1	9	7
$t2	10	10
$t3	11	268500992
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	17
$s1	17	-268500975
$s2	18	8
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	32
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194448
hi		0
lo		0



Código MIPS

```
addi $t0,$zero,5

add $zero,$zero,$zero# NOP
add $zero,$zero,$zero# NOP

add $t1,$t0,$zero

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

addi $t1,$t1,2

add $zero,$zero,$zero# NOP
add $zero,$zero,$zero#NOP
```

```
addi $t2,$t1,3

addi $t3, $t3, 0x1001

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

sll $t3,$t3,16

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

sw $t2,0($t3)

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

add $s0,$t2,$t1

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

sub $s1,$s0,$t3

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

lw $t2, 0($t3)

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

addi $s2,$t2,-2

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

or $s2,$s2,$t4

add $zero,$zero,$zero#NOP
add $zero,$zero,$zero#NOP

sll $s7,$s2,2

exit:
```