Proiect MIPS 32
Pipeline

Nume Student: Barna Ioana Melisa

Grupa: 30225

CUPRINS

1. Tabelul register MIPS32 Pipeline
2. Tabel diagrama PipelineMIPS32
3. Cod fara hazarduri
4. Functionalitate proiect
5. Schema procesor Pipeline

1. *Tabelul register MIPS32 Pipeline*

a. Configurare registre MIPS32 Pipeline – varianta 1

REG_IF_ID[63:0]	REG_ID_EX[162:0]	REG_EX_MEM[110:0]	REG_MEM_WB[72:0]
Instruction [63:32]	regDst_id_ex [162:161]	branch_ex_mem [110:109]	memToReg_mem_wb [72:71]
PCp4 [31:0]	aluSrc_id_ex [160:159]	memWrite_ex_mem [108:107]	regWrite_mem_wb [70:69]
	branch_id_ex [158:157]	memToReg_ex_mem [106:105]	alulesire_mem_wb [68:37]
	aluOp_id_ex [156:155]	regWrite_ex_mem [104:103]	memData_mem_wb [36:5]
	memWrite_id_ex [154:153]	zero_ex_mem [102:101]	rd_mem_wb [4:0]
	memToReg_id_ex [152:151]	branchAddr_ex_mem [100:69]	
	regWrite_id_ex [150:149]	alulesire_ex_mem [68:37]	
	rd1_id_ex [148:117]	out_ex_mem [36:32]	
	rd2_id_ex [116:85]	rd2_ex_mem [31:0]	
	extImm_id_ex [84:53]		
	func_id_ex [52:47]		
	sa_id_ex [46:42]		
	rd_id_ex [41:37]		
	rt_id_ex [36:32]		
	pc_id_ex [31:0]		

b. Configurare registre MIPS32 Pipeline – varianta 2

IF/ID	ID/EX	EX/MEM	MEM/WB
instruction_if_id (32)	pc_id_ex (32)	memToReg_ex_mem (1)	rd_mem_wb (5)
pc_if_id (32)	rd1_id_ex (32)	regWrite_ex_mem (1)	memToReg_mem_wb (1)
	rd2_id_ex (32)	memWrite_ex_mem (1)	regWrite_mem_wb (1)
	extImm_id_ex (32)	branch_ex_mem (1)	alulesire_mem_wb (32)
	memToReg_id_ex (1)	zero_ex_mem (1)	memData_mem_wb (32)
	regWrite_id_ex (1)	alulesire_ex_mem (32)	
	memWrite_id_ex (1)	out_ex_mem (5)	
	branch_id_ex (1)	rd2_ex_mem (32)	
	aluOp_id_ex (3)	branchAddr_ex_mem (1)	
	aluSrc_id_ex (1)		
	regDst_id_ex (1)		
	sa_id_ex (5)		
	func_id_ex (6)		
	rt_id_ex (5)		
	rd_id_ex (5)		

2. *Tabel diagrama PipelineMIPS32*

a. Varianta program cu hazarduri

Adr.	Instr./Ciclu	C1	C2	C3	C4	C5	C6	C5	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
0	lw $1, n	IF	ID	EX	MEM	WB															
1	addi $2, $0, 0		IF	ID($0)	EX	MEM	WB($2)														
2	addi $3, $0, 0			IF	ID($0)	EX	MEM	WB($3)													
3	addi $4, $0, 0				IF	ID($0)	EX	MEM	WB($4)												
4	sll $5, $4, 2					IF	ID($4)	EX	MEM	WB($5)											
5	lw $6, A_addr($5)						IF	ID($5)	EX	MEM	WB($6)										
6	and $7, $6, 1							IF	ID($6)	EX	MEM	WB($7)									
7	beq $7, $0, is_even								IF	ID($0)	EX	MEM	WB($7)								
8	add $3, $3, $6									IF	ID($3)	EX	MEM	WB($3)							
9	j continue_loop										IF	ID (C)	EX	MEM	WB						
10	add $2, $2, $6											IF	ID($2)	EX	MEM	WB($2)					
11	addi $4, $4, 1												IF	ID($4)	EX	MEM	WB($4)				
12	add $8, $4, $1													IF	ID($4)	EX	MEM	WB($8)			
13	bne $8, $0, loop														IF	ID($0)	EX	MEM	WB($8)		
14	sw $2, suma_pare															IF	ID($2)	EX	MEM	WB	
15	sw $3, suma_impare																IF	ID($3)	EX	MEM	WB

b. Varianta program fara hazarduri

Adr.	Instr./Ciclu	C1	C2	C3	C4	C5	C6	C5	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30
0	lw $1, n	IF	ID	EX	MEM	WB																									
1	addi $2, $0, 0		IF	ID	EX	MEM	WB																								
2	addi $3, $0, 0			IF	ID	EX	MEM	WB																							
3	addi $4, $0, 0				IF	ID	EX	MEM	WB																						
4	sll $5, $4, 2					IF	ID	EX	MEM	WB																					
5	NoOp						IF	ID	EX	MEM	WB																				
6	lw $6, A_addr($5)							IF	ID	EX	MEM	WB																		
7	NoOp								IF	ID	EX	MEM	WB																	
8	and $7, $6, 1									IF	ID	EX	MEM	WB																
9	beq $7, $0, is_even										IF	ID	EX	MEM	WB															
10	NoOp											IF	ID	EX	MEM	WB														
11	NoOp												IF	ID	EX	MEM	WB													
12	NoOp													IF	ID	EX	MEM	WB												
13	add $3, $3, $6														IF	ID	EX	MEM	WB											
14	j continue_loop															IF	ID	EX	MEM	WB										
15	NoOp																IF	ID	EX	MEM	WB									
16	add $2, $2, $6																	IF	ID	EX	MEM	WB								
17	addi $4, $4, 1																		IF	ID	EX	MEM	WB							
18	NoOp																			IF	ID	EX	MEM	WB						
19	add $8, $4, $1																				IF	ID	EX	MEM	WB					
20	bne $8, $0, loop																					IF	ID	EX	MEM	WB					
21	NoOp																						IF	ID	EX	MEM	WB				
22	NoOp																							IF	ID	EX	MEM	WB			
23	NoOp																								IF	ID	EX	MEM	WB		
24	sw $2, suma_pare																									IF	ID	EX	MEM	WB	
25	sw $3, suma_impare																										IF	ID	EX	MEM	WB

3. *Cod fara hazarduri*

lw $1, n

addi $2, $0, 0

addi $3, $0, 0

addi $4, $0, 0

loop:

sll $5, $4, 2

lw $6, A_addr($5)

NOOP

and $7, $6, 1

beq $7, $0, is_even

NoOp

NoOp

NoOp

add $3, $3, $6

j continue_loop

NoOp

```
is_even:
   add $2, $2, $6

continue_loop:
   addi $4, $4, 1
   NoOp
   add $8, $4, $1
   bne $8, $0, loop
   NoOp
   NoOp
   NoOp

end_loop:
   sw $2, suma_pare
   sw $3, suma_impare
```

4. *Functionalitate proiect*

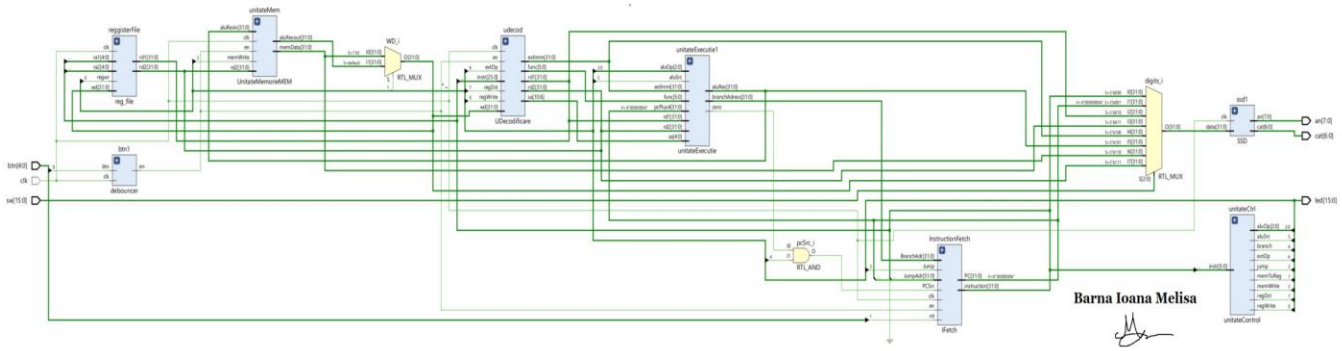
->codul nu a fost testat pe placuta;

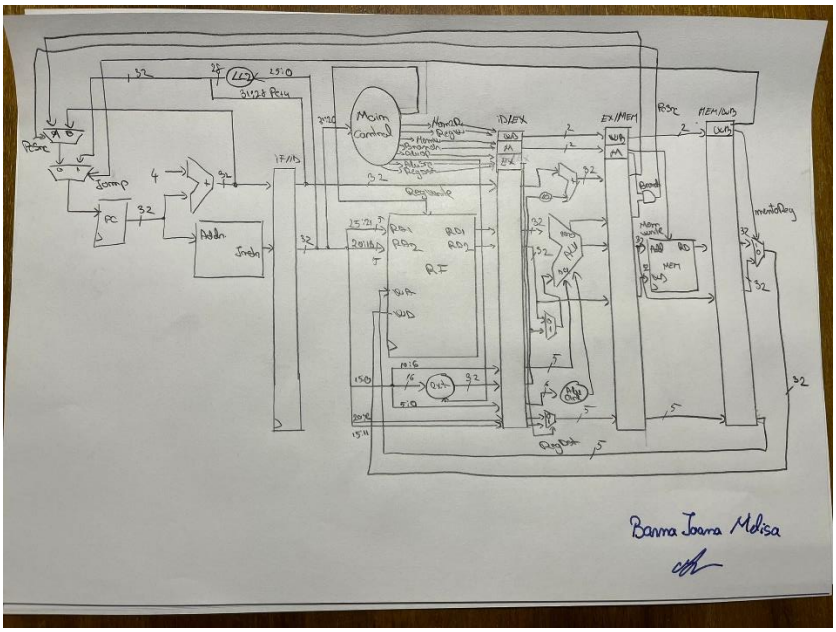
->codul nu prezenta erori de sintaxa;

-> a fost generat cu success: Synthesis, Implementation, Bitstream;

->codul prezinta schimbari pe partea de MIPS32 fata de varianta incarcata pe teams la assignment;

5. *Schema procesor Pipeline*


Barna Ioana Melisa


Barna Ioana Melisa