Procesorul MIPS 32
varianta Pipeline

Nume: Rujac Roxana
Grupa : 30224

1. Componente

Procesorul implementat rezolva problema găsirii celui mai mic număr impar dintr-un șir de numere. Pentru aceasta am avut nevoie de următoarele componente:

-o unitate de control (UC)
-o memorie (MEM) -un generator monopols (MPG)
-un afisor pe 7 segmente (SSD)
-o unitate de decodificare a instrucțiunilor (ID)
-o unitate de extragere a instrucțiunilor (IFetch)
-o unitate de execuție a instrucțiunilor (EX)

Toate acestea sunt componente folosite pentru a crea arhitectura completa a procesorului în fișierul test_env.

2. Programul ASM

0	lw $2, 8($0)
1	lw $3, 4($0)
2	lw $4, 0($0)
3	addi $5, $0, 0
4	beq $1, $3, 27
5	NoOp
6	NoOp
7	NoOp
8	lw $6, 12($5)
9	addi $7, $0, 1
10	NoOp
11	NoOp
12	and $8, $6, $7
13	NoOp
14	NoOp
15	beq $8, $0, 11
16	NoOp

17	NoOp
18	NoOp
19	slt $9, $6, $2
20	NoOp
21	NoOp
22	beq $9, $0, 4
23	NoOp
24	NoOp
25	NoOp
26	add $2, $6, $0
27	addi $1, $1, 1
28	addi $5, $5, 4
29	j 4
30	NoOp
31	sw $2, 8($0)

3. Instructiuni adaugate

1. XOR

Descriere: SAU-Exclusiv logic între două registre, memorează rezultatul în alt registru
RTL: $d $s ^ $t; PC PC + 4;
Sintaxă asamblare: xor $d, $s, $t
Format: 000000 sssss ttttt ddddd 00000 100110

2. SLT

Descriere: Dacă $s < $t, $d este inițializat cu 1, altfel cu 0
RTL: PC PC + 4; if $s < $t then $d 1 else $d 0;
Sintaxă asamblare: slt $d, $s, $t
Format: 000000 sssss ttttt ddddd 00000 101010

3. BGEZ

Descriere: Salt condiționat dacă un registru este mai mare sau egal cu 0
RTL: if $s ≥ 0 then PC (PC + 4) + (SE(offset) << 2) else PC PC + 4;
Sintaxă asamblare: bgez $s, offset
Format: 000001 sssss 00000 oooooooooooooooo

4. SLTI

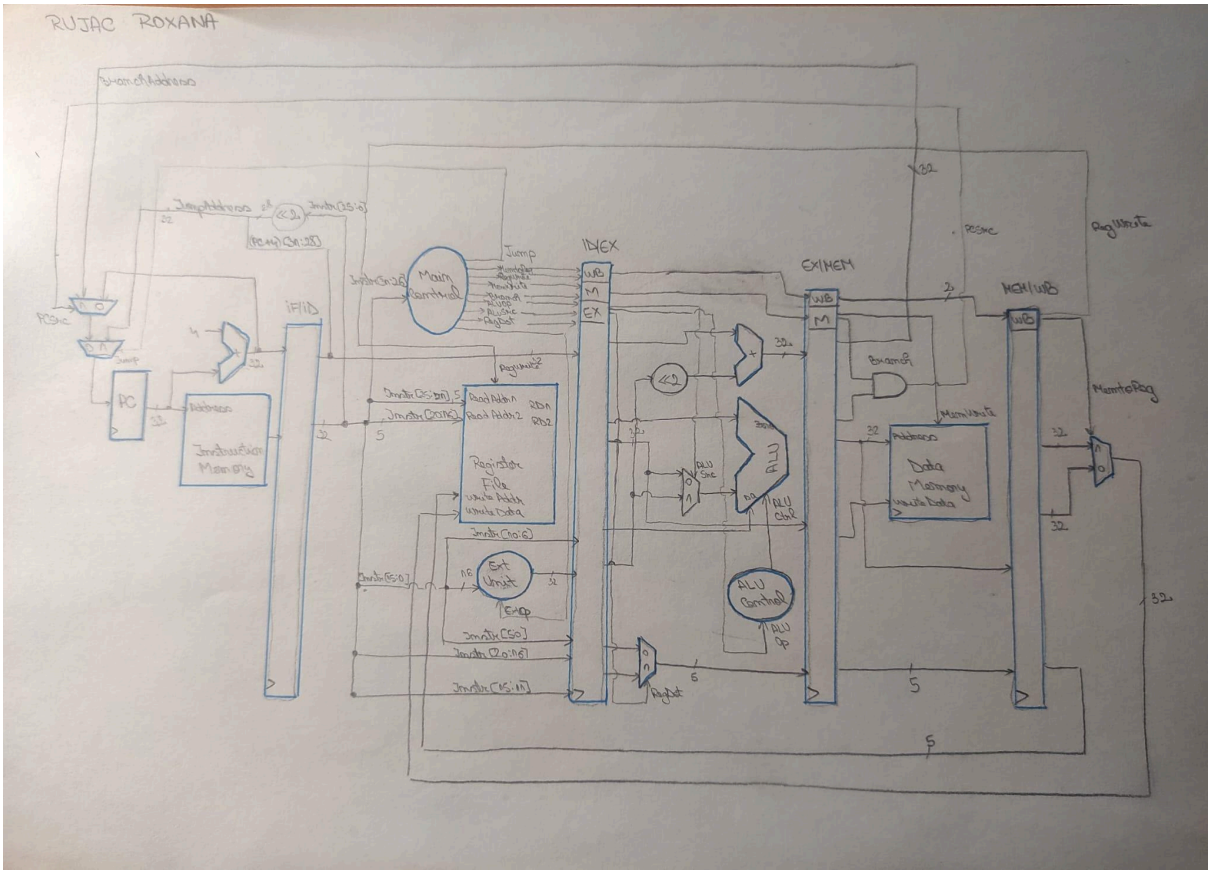
Descriere: Dacă $s este mai mic decât un imediat, $t este inițializat cu 1, altfel cu 0
RTL: PC PC + 4; if $s < SE(imm) then $t 1 else $t 0;
Sintaxă asamblare: slti $t, $s, imm
Format: 001010 sssss ttttt iiiiiiiiiiiiiiii

4. Implementarea PIPELINE

Registrele pipeline sunt denumite în funcție de etapele de execuție pe care le separă: IF/ID, ID/EX, EX/MEM, MEM/WB. Aceste registre memorează rezultatele de la etapa anterioară și le pun la dispoziție elementelor funcționale din etapa următoare. Astfel, în pipeline pot fi executate simultan până la 5 instrucțiuni consecutive, fiecare în etape diferite. Registrele contribuie la îmbunătățirea performanței prin facilitarea execuției simultane a mai multor instrucțiuni.

-Reducerea duratei perioadei de ceas
-Execuție concomitentă a instrucțiunilor
-Utilizarea eficientă a unităților funcționale
-Minimizarea intârzierilor

5.Schema procesorului Pipeline



6. Proiectarea registrelor Pipeline

Configurare registre MIPS32 Pipeline – varianta 2

Se introduc pe coloane numele utilizate în codul VHDL pentru semnalele de date și control implementate ca registre, pe categorii. În paranteză se introduce dimensiunea în biți.

IF/ID	ID/EX	EX/MEM	MEM/WB
Instruction_IF_ID (32)	RegDst_ID_EX (1)	Branch_EX_MEM (1)	RegWr_MEM_WB (1)
PCplus4_IF_ID (32)	ALUSrc_ID_EX (1)	MemWrite_EX_MEM (1)	MemtoReg_MEM_WB (1)
	Branch_ID_EX (1)	MemtoReg_EX_MEM (1)	ALURes_MEM_WB (32)
	ALUOp_ID_EX (1)	RegWrite_EX_MEM (1)	MemData_MEM_WB (32)
	MemWrite_ID_EX (1)	Zero_EX_MEM (1)	WA_MEM_WB (5)
	MemtoReg_ID_EX (1)	BranchAddress_EX_MEM (32)	
	RegWrite_ID_Ex (1)	ALURes_EX_MEM (32)	
	RD1_ID_EX (32)	WA_EX_MEM (5)	
	RD2_ID_EX (32)	RD2_EX_MEM (32)	
	Ext_Imm_ID_EX (6)		
	func_ID_EX (5)		
	sa_ID_EX (5)		
	rd_ID_EX (5)		
	PC_ID_EX (32)		

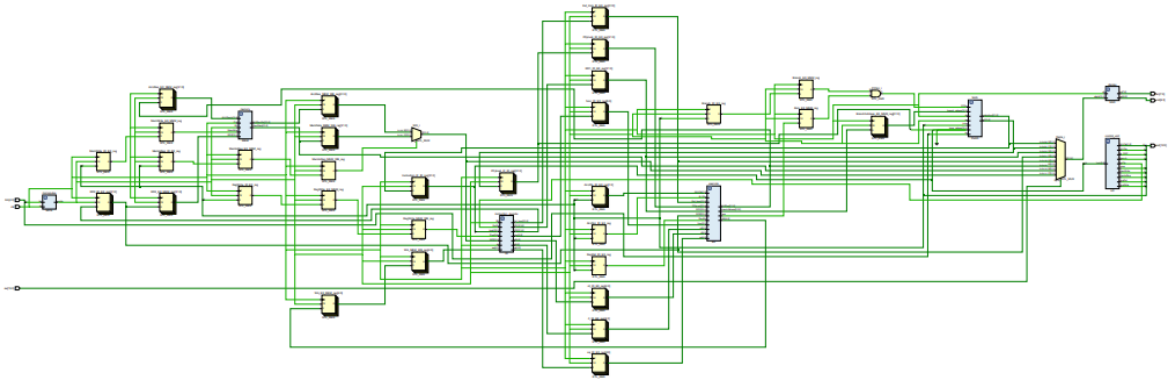
7. Diagrama de executie

Adr.	Instr./Ciclu	C1	C2	C3	C4	C5	C6	C5	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18		
0	xor $1, $0, $0	IF	ID	EX	MEM	WB															
1	lw $2, 8($0)		IF	ID	EX	MEM	WB														
2	lw $3, 4($0)			IF	ID	EX	MEM	WB ($3 -S:2, 5)													
3	lw $4, 0($0)				IF	ID	EX	MEM	WB												
4	addi $5, $0, 0					IF	ID	EX	MEM	WB											
5	beq $1, $3, 10						IF	ID($3)	EX	MEM (C)	WB										
6	lw $6, 12($5)							IF	ID	EX	MEM	WB ($6 -D: 6, 8)									
7	addi $7, $0, 1								IF	ID	EX	MEM	WB ($7 -D:7, 8)								
8	and $8, $6, $7									IF	ID ($6, $7)	EX	MEM	WB ($8 -D:8, 9)							
9	beq $8, $0, 3										IF	ID($8)	EX	MEM	WB						
10	slt $9, $6, $2											IF	ID	EX	MEM	WB ($9 -D:10, 11)					
11	beq $9, $0, 1												IF	ID (&9)	EX	MEM	WB				
12	add $2, $6, $0													IF	ID	EX	MEM	WB			
13	addi $1, $1, 1														IF	ID	EX	MEM	WB		
14	addi $5, $5, 4															IF	ID	EX	MEM	WB	
15	j 5																IF	ID (C)	EX	MEM	WB
16	sw $2, 8($0)																	IF	ID	EX	MEM

arianta de program fără hazarduri

dr.	Instr./Ciclu	C1	C2	C3	C4	C5	C6	C5	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18														
0	lw $2, 8($0)	IF	ID	EX	MEM	WB																											
1	lw $3, 4($0)		IF	ID	EX	MEM	WB																										
2	lw $4, 0($0)			IF	ID	EX	MEM	WB																									
3	addi $5, $0, 0				IF	ID	EX	MEM	WB																								
4	beq $1, $3, 27					IF	ID	EX	MEM	WB																							
5	NoOp						IF	ID	EX	MEM	WB																						
6	NoOp							IF	ID	EX	MEM	WB																					
7	NoOp								IF	ID	EX	MEM	WB																				
8	lw $6, 12($5)									IF	ID	EX	MEM	WB																		
9	addi $7, $0, 1										IF	ID	EX	MEM	WB																	
10	NoOp											IF	ID	EX	MEM	WB																
11	NoOp												IF	ID	EX	MEM	WB															
12	and $8, $6, $7													IF	ID	EX	MEM	WB														
13	NoOp														IF	ID	EX	MEM	WB													
14	NoOp															IF	ID	EX	MEM	WB												
15	beq $8, $0, 11																IF	ID	EX	MEM	WB											
16	NoOp																	IF	ID	EX	MEM	WB										
17	NoOp																		IF	ID	EX	MEM	WB									
18	NoOp																			IF	ID	EX	MEM	WB								
19	slt $9, $6, $2																				IF	ID	EX	MEM	WB							
20	NoOp																					IF	ID	EX	MEM	WB						
21	NoOp																						IF	ID	EX	MEM	WB					
22	beq $9, $0, 4																							IF	ID	EX	MEM	WB				
23	NoOp																								IF	ID	EX	MEM	WB			
24	NoOp																									IF	ID	EX	MEM	WB		
25	NoOp																										IF	ID	EX	MEM	WB		
26	add $2, $6, $0																											IF	ID	EX	MEM	WB	
27	addi $1, $1, 1																												IF	ID	EX	MEM	
28	addi $5, $5, 4																													IF	ID	EX	
29	j 4																														IF	ID	
30	NoOp																															IF	
31	sw $2, 8($0)																																

8. Schema RTL



9. Evaluare functionalitate

Programul este testat integral pe placuta, atat componentele separate cat și programul final. Desi procesorul functioneaza corespunzator in varianta single-cycle, varianta pipeline implementata de mine nu este una corecta deoarece nu urmeaza instructiunile corect. Testand pe placuta am observat faptul ca programul imi ruleaza doar primele 5 instructiuni, imi incarca in memorie datele corepunzatoare, insa apoi se intoarce la prima instructiune (inclusiv PC devine 0). Am verificat codul insa nu am reusit sa gasesc problema.