Procesorul MIPS pipeline 16 biti

Student: Rad Vladut

Grupa: 30225

Cele 4 instructiuni suplimentare:

Tipul R:

1. XOR $rd, $rs, $rt => RF[rd]<- RF[rs] ^ RF[rt]
2. SRA $rd, $rs, sa => RF[rd] <-RF[rs]>>sa

Tipul I:

1.ANDI $rt, $rs, imm -> RF[rt] <- RF[rs] & Z_Ext(imm)

2.ORI $rt, $rs, imm -> RF[rt]<- RF[rs] or Z_Ext(imm)

Nu au fost facute modificari particulare la mips pipeline pentru cele 4 instructiuni suplimentare alese.

Trasarea programului

0: xor $0, $0, $0 - registrul 0 este initializat la 0

1: xor $1, $1, $1 - registrul 1 este initializat la 0

2: xor $2, $2, $2 - registrul 2 este initializat la 0

3: xor $3, $3, $3 registrul 3 este initializat la 0

4: NoOp

5:NoOp

6: NoOp

7: addi $3, $0, 20 -in registrul 3 se adauga 20(suma primelor 5 numere pare)

8: addi $2, $2, 2 -registrul 2 se incrementeaza din 2 in 2 pentru a avea numerele pare formate

9: ADD $1, $1, 2 -in registrul 1 se calculeaza suma

10: NoOp

11: NoOp

12: NoOp

13: beq $1, $3, 21 -daca am ajuns la 20 se sare peste instructiunea urmatoare (se iese din for)

14: NoOp

15: NoOp

16:NoOp

17: jmp 8 -face salt la linia 5

18: NoOp

B”000_000_000_000_0_110”, X”6”

B”000_001_001_001_0_110”, X”0496”

B”000_010_010_010_0_110”, X”0926”

B”000_011_011_011_0_110”, X”0db6”

B"001_000_000_0000000", --NoOp

B"001_000_000_0000000", --NoOp

B"001_000_000_0000000", --NoOp

B”001_000_011_0010100”, X”2194”

B”001_010_010_0000010”, X”2902”

B”000_001_010_001_0_000”, X”0510”

B"001_000_000_0000000", --NoOp

B"001_000_000_0000000", --NoOp

B"001_000_000_0000000", --NoOp

B”100_001_011_0000001”, X”8581”

B"001_000_000_0000000", --NoOp

B"001_000_000_0000000", --NoOp

B"001_000_000_0000000", --NoOp

B”111_0000000000101”, X”C005”

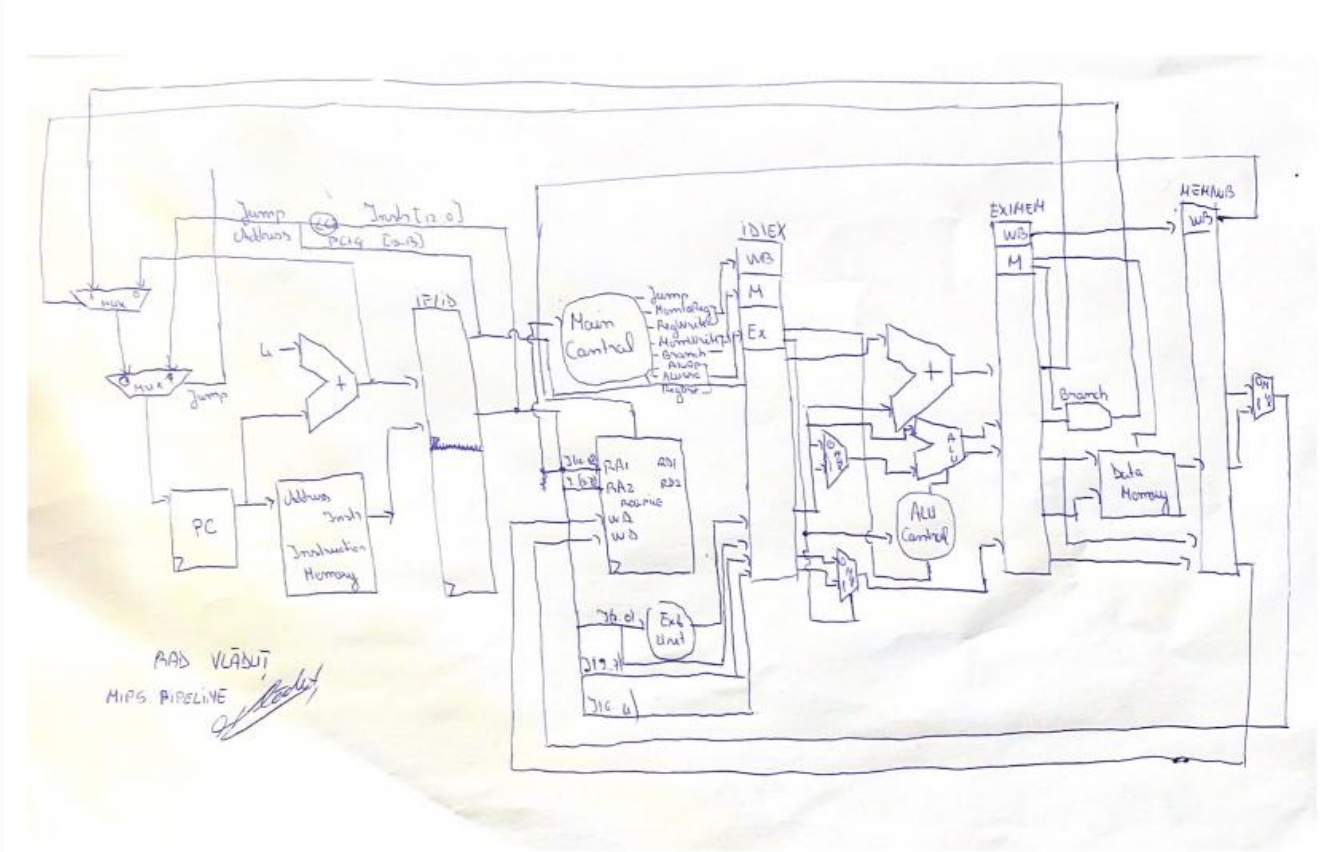
B"001_000_000_0000000", --NoOp

Diagramele pipeline si identificarile hazardurilor

Adr.	Instrucțiune/Clk	CC1	CC2	CC3	CC4	CC5	CC6	CC5	CC8	CC9	CC10	CC11	CC12	CC13
0	XOR $0, $0, $0	IF	ID	EX	MEM	WB								
1	XOR $1, $1, $1		IF	ID	EX	MEM	WB							
2	XOR $2, $2, $2			IF	ID	EX	MEM	WB						
3	XOR $3, $3, $3				IF	ID	EX	MEM	WB					
4	ADDI $3, $0,20					IF	ID	EX	MEM	WB				
5	ADDI $2, $0, 20						IF	ID	EX	MEM	WB			
6	ADD $1, $1, 2							IF	ID	EX	MEM	WB		
7	BEQ $1, $3, 21								IF	ID	EX	MEM	WB	
8	JMP 5									IF	ID	EX	MEM	WB

Varianta de program fără hazarduri

Adr.	Instrucțiune																					
0	XOR $0, $0, $0	IF	ID	EX	MEM	WB																
1	XOR $1, $1, $1		IF	ID	EX	MEM	WB															
2	XOR $2, $2, $2			IF	ID	EX	MEM	WB														
3	XOR $3, $3, $3				IF	ID	EX	MEM	WB													
4	NoOp					IF	ID	EX	MEM	WB												
5	NoOp						IF	ID	EX	MEM	WB											
6	NoOp							IF	ID	EX	MEM	WB										
7	ADDI $3, $0,20								IF	ID	EX	MEM	WB									
8	ADDI $2, $0, 20									IF	ID	EX	MEM	WB								
9	ADD $1, $1, 2										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	BEQ $1, $3, 21														IF	ID	EX	MEM	WB			
14	NoOp															IF	ID	EX	MEM	WB		
15	NoOp																IF	ID	EX	MEM	WB	
16	NoOp																	IF	ID	EX	MEM	WB
17	JMP 8																		IF	ID	EX	MEM
18	NoOp																			IF	ID	EX



IF/ID

Instruction_out(16) Pc_nxt_out(16)

ID/EX

Instruction_out(16) Ext_imm_out(16) Rd1_out(16) Rd2_out(16) Pc_nxt_out(16) Ex_out(5) M_out(5) Wb_out(2)

EX/MEM

Rde_out(16) Address_out(16) Write_Address_out(3) Zero_out(1) M_out(2) Wb_out(2) Branch_out(16)

MEM/WB

Write_data_out(16) Address_out(16) Wb_out(2) Write_Address_out(3)