

MIPS 16

Pipeline

Modificari făcute pentru instrucțiunile alese suplimentar:

Doar în cazul instrucțiunilor BNE și BGEZ au fost făcute modificări ,și anume am adăugat două noi semnale în MainControl ,unul pentru BNE și altul pentru BGEZ , care vor intra în etajul de semnal M,și vor trece prin registrele ID/EX și EX/MEM .După ce vor iesi din EX/MEM vor forma fiecare câte o poartă and ,iar iesirea acestei porți va intra într-o poarta or,formând semnalul PCSrc.Această modificare poate fi observată mai jos ,pe poza cu schema procesorului.

Tabelul cu descriere registrilor:

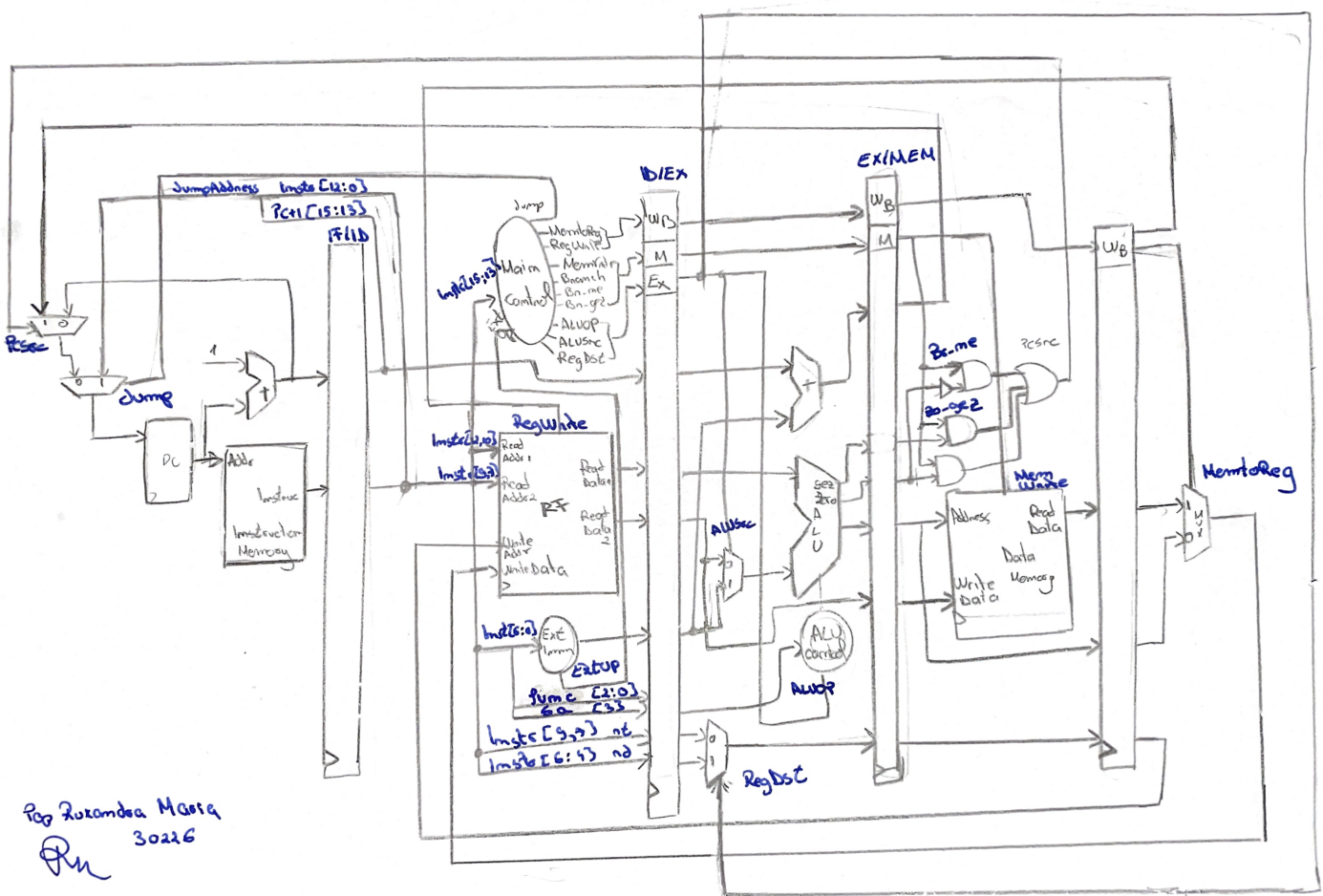
Am realizat prima variantă de tabel.În paranteză este reprezentat numărul de biți pe care îl ocupă fiecare semnal din registru.

- *Pentru implementarea registrul IF/ID am folosit 32 de biți.*
- *Pentru implementarea registrul ID/EX am folosit 84 de biți.*
- *Pentru implementarea registrul EX/MEM am folosit 59 de biți.*
- *Pentru implementarea registrul MEM/WB am folosit 37 de biți.*

REG_IF_ID(31 – 0)	REG_ID_EX(83– 0>)	REG_EX_MEM(58 – 0)	REG_MEM_WB(36 – 0)
Instr(31 – 16)	RegDst(83)	br_ne(58)	RegWrite(36)
PC + 1(15 – 0)(PCInc)	ALUSrc(82)	br_gez(57)	MemtoReg(35)
	Branch(81)	Branch(56)	ALUResOut(34-19)
	br_gez(80)	MemWrite(55)	MemData(18-3)
	br_ne(79)	MemtoReg(54)	rWA(2-0)
	ALUOp(78-77)	RegWrite(53)	
	MemWrite(76)	Zero(52)	
	MemtoReg(75)	gez(51)	
	RegWrite(74)	BranchAddress(50-35)	
	RD1(73-58)	ALURes(34-19)	
	RD2(57-42)	rWA(18-16)	
	Ext_Imm(41-26)	RD2(15-0)	
	func(25-23)		
	sa(22)		
	rd(21-19)		
	rt(18-16)		
	PC+1(15-0)		

Schema procesorului mips pipeline:

Am modificat schema din laborator ,astfel încât să corespundă procesorului pe 16 biți.Și se pot observa și semnalele de BNE și BGEZ ,despre care am discutat anterior.



Top Zuzomda Maseq

30226

A handwritten signature in blue ink, appearing to be "Rm".

Diagrama Pipeline:

Hazardele de tip structural au fost tratate prin frontul descendent de clock în blocul de registre. Așa că trebuia să identificăm numai hazardele de date și cele de control.

Pe schema de mai jos se pot observa hazardele de date/control pe care le-am întâlnit.

Adr.	Instrucțiune/Cik	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16	CC17	CC18	CC19
0	addi \$1,\$0,0	IF	ID	EX	MEM	WB														
1	addi \$2,\$0,4		IF	ID	EX	MEM	WB													
2	add \$3,\$0,\$0			IF	ID	EX	MEM	WB												
3	addi \$4,\$0,1				ID	EX	MEM	WB												
4	addi \$5,\$0,12				IF	ID	EX	MEM	WB											
5	addi \$6,\$0,1					ID	EX	MEM	WB											
6	beq \$1,\$5,8					IF	ID	EX	MEM	WB										
7	sw \$1,0(\$3)						IF	ID	EX	MEM	WB									
8	sw \$2,0(\$4)							ID	EX	MEM	WB									
9	lw \$1,0(\$3)								ID	EX	MEM	WB								
10	lw \$2,0(\$4)									ID	EX	MEM	WB							
11	add \$1, \$1,\$6									ID	EX	MEM	WB							
12	add \$2,\$2,\$1										ID	EX	MEM	WB						
13	sub \$5,\$5,\$6											ID	EX	MEM	WB					
14	j 6.												ID	EX	MEM	WB				

Am tratat aceste hazarde prin introducerea unor operații de tip NoOp, după cum se poate observa:

Adr.	Instrucțiune	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16	CC17	CC18	CC19	CC20	CC21	CC22	CC23	CC24	CC25	CC26	CC27	CC28	CC29
0	addi \$1,\$0,0	IF	ID	EX	MEM	WB																								
1	addi \$2,\$0,4		IF	ID	EX	MEM	WB																							
2	add \$3,\$0,\$0			IF	ID	EX	MEM	WB																						
3	addi \$4,\$0,1				ID	EX	MEM	WB																						
4	addi \$5,\$0,12				IF	ID	EX	MEM	WB																					
5	addi \$6,\$0,1					ID	EX	MEM	WB																					
6	NoOp						ID	EX	MEM	WB																				
7	beq \$1,\$5,14						IF	ID	EX	MEM	WB																			
8	NoOp							ID	EX	MEM	WB																			
9	NoOp								ID	EX	MEM	WB																		
10	NoOp									ID	EX	MEM	WB																	
11	sw \$1,0(\$3)								ID	EX	MEM	WB																		
12	sw \$2,0(\$4)									ID	EX	MEM	WB																	
13	lw \$1,0(\$3)										ID	EX	MEM	WB																
14	lw \$2,0(\$4)											ID	EX	MEM	WB															
15	NoOp												ID	EX	MEM	WB														
16	add \$1, \$1,\$6												ID	EX	MEM	WB														
17	NoOp													ID	EX	MEM	WB													
18	NoOp														ID	EX	MEM	WB												
19	add \$2,\$2,\$1													ID	EX	MEM	WB													
20	sub \$5,\$5,\$6														ID	EX	MEM	WB												
21	j 7															ID	EX	MEM	WB											
22	NoOp																ID	EX	MEM	WB										

După adaugarea operațiilor NoOp instrucțiunea beq va sari peste 14 instrucțiuni dacă valoarea celor 2 registre sunt egale, respectiv jump sare acum la instrucțiunea de la adresa 7 ,nu 6 cum era inițial.

Operația NoOp este o operație de adunare de forma : ,add \$0,\$0,\$0 ,care face ca citirea să se facă deodată cu scrierea sau ulterior,astfel încât să nu mai apară hazarde .

În continuare ,voi atasa o poză ,cu parcurgerea primei iterații a programului.Se pot observa deosebiri față de parcurgerea primei iterații ale procesorul mips cu ciclu unic,deoarece aici execuția programului se execută în mai multe cicluri,deci există o întârziere.Valoriile obținute sunt aceleași ,doar că întârziate.

		CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15	CC16	CC17	CC18	CC19	CC20	CC21	CC22	CC23
IF	Instr	2080	2104	0030	2201	228C	2301	0000	868E	0000	0000	0000	6C80	7100	4C80	5100	0000	0710	0000	0000	08A0	1751	E007	0000
PC		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F	0010	0011	0012	0013	0014	0015	0016	0017
EX	Rd1	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	0000	0001	0000	0000	0000	0000	0004	000C
	Rd2	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0004	0000	0004	0000	0001	0000	0000	0001	0001
	Ext Imm	0000	0000	0000	0004	0030	0001	000C	0001	0000	000E	0000	0000	0000	0000	0000	0000	0000	0000	0010	0000	0000	0020	0051
	Result All	0000	0000	0000	0004	0000	0001	000C	0001	0000	FFF4	0000	0000	0000	0000	0001	0000	0001	0000	0001	0000	0000	0005	000B
MEM	MemData	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0004	0000	0004	0000	0000	0000
WB	WriteData	0000	0000	0000	0000	0000	0004	0000	0001	000C	0001	0000	FFF4	0000	0000	0000	0000	0001	0000	0004	0000	0001	0000	0000

Elemente funcționale/nefuncționale:

Toate elementele funcționează corect. Pentru a realiza procesorul pipeline ,au fost necesare unele modificari în ID,EX,reg_file și în test_env.

- **ID**

Am eliminat semnalul de RegDst,și mux-ul corespunzator lui.Am adăugat un nou semnal de intrare WA care reprezintă write address din reg_file, am adăugat două semnale de iesire rt și rd.Și implicit a fost modificat și port map-ul corespunzător registrului reg_file.

- reg_file

Nu mai avem un clock ascendent ,ci unul descendent (falling_edge).Cu ajutorul acestuia sa rezolvat problema cu hazardul structural.

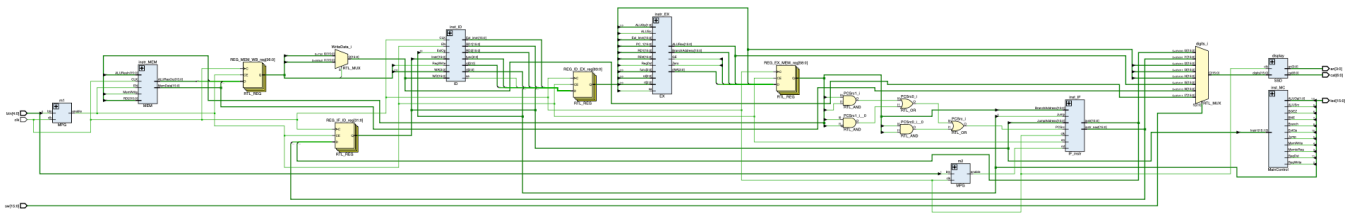
- **EX**

Am introdus 3 semnale de intrare ,care formează un nou mux,și un semnal de iesire rWa. Cele 3 semnale noi de intrare sunt:rt,rd,RegDst.Mux-ul nou format este întâlnit în cod cu eticheta mux2.

- **test_env**

A suferit mai multe modificări.Au fost adăugate 4 noi procese și 4 noi semnale , care simbolizează cele 4 registre :IF/ID,ID/EX,EX/MEM,MEM/WB.Dimensiunea alocată pentru fiecare registru în parte este prezentă în paragraful 2.Port map-urile au fost modificate astfel încât să corespundă cu semnalele registrelor

Schema RTL:



Nu am activități incomplete ,am reusit să implementez toate componentele necesare.Nu există erori.

Am testat pe placuță , și am comparat ceea ce se afisează ,cu tabelul scris de mine pentru prima interacție ,rezultatele erau aceleași.

