

MIPS Pipeline

-- Versiunea pe 32 de biți --

1. Componente folosite și funcționalitatea acestora

Pentru realizarea proiectului MIPS pipeline, am implementat o versiune pe 32 biti care conține următoarele componente:

- MPG.vhd – reprezintă un Generator de Monopuls Sincron
- SSD.vhd – reprezintă Afișoare pe 7 segmente (Seven Segment Display)
- IFetch.vhd – reprezintă Unitatea de extragere a instrucțiunilor
- ID.vhd – reprezintă Unitatea de Decodificare a instrucțiunilor
- UC.vhd – reprezintă Unitatea de Control a procesorului
- EX.vhd – reprezintă Unitatea de Execuție a instrucțiunilor
- MEM.vhd – reprezintă Unitatea de Memorie
- test_env.vhd – reprezintă Arhitectura completă a procesorului

Generator de Monopuls Sincron(MPG)

Este compus dintr-un numărător și 3 bistabile D Flip-Flop. Rolul circuitului MPG (Generator de Monopuls Sincron) este de a genera un singur impuls la o apăsare de buton.

Afișor pe 7 segmente(SSD)

Acesta este compus din anodi și catodi. Pentru afișarea unei cifre se folosesc 7 led-uri active (catodi) iar fiecare cifră este activată de un semnal (anod). Din motive legate de economie a semnalelor alocate, catodii sunt comuni tuturor afișoarelor active, iar pentru a afișa cifre diferite pe afișoare, acestea vor fi activate alternativ, în mod ciclic, pentru perioade scurte de timp (0.16ms).

Unitatea de extragere a instrucțiunilor(IFetch)

Unitatea de extragere a instrucțiunilor (IFetch) primește pe intrările de date adresele de salt și pune la dispoziție, pe ieșiri, adresa imediat următoare (PC+4), respectiv conținutul instrucțiunii curente.

Unitatea de Decodificare a instrucțiunilor(ID)

Unitatea de decodificare a instrucțiunilor (ID) primește pe intrările de date instrucțiunea curentă și valoarea WD, care se scrie în RF, ambele pe 32 de biți. ID pune pe ieșiri, operanzii RD1, RD2 și imediatul extins Ext_Imm(valorile fiind pe 32 de biți). Suplimentar, pe ieșire mai apar câmpurile *function* (6 biți) și *sa* (5 biți) din instrucțiune. Semnalul de control RegDst selectează registrul (adresa) în care se scrie valoarea WD atunci când semnalul de control RegWrite este activ.

Unitatea de Control(UC)

Unitatea de control (UC) , generează semnalele care determină funcționalitatea unităților din calea de date.

Unitatea de Execuție(EX)

Unitatea de execuție (EX) realizează operațiile aritmetice și logice necesare instrucțiunii. Aceasta primește pe intrările de date registrele RD1 și RD2 de la blocul de registre, imediatul extins Ext_imm și adresa de instrucțiune imediat următoare PC+4, codificate pe 32 de biți. Suplimentar, apar câmpurile *func* și *sa* din instrucțiunea curentă, pe 6 biți, respectiv 5 biți. EX pune la dispoziție rezultatul ALU cu semnalul de validare Zero și adresa de salt condiționat Branch Address, calculată astfel:

$$\text{Branch Address} \leq (\text{PC}+4) + (\text{Ext_imm} \ll 2)$$

Unitatea de Memorie(MEM)

Unitatea de memorie (MEM) are rol de stocare a datelor, pe 32 de biți. Scrierea în memorie este sincronă pe frontul de ceas ascendent și citirea este asincronă, ca la blocul de registre RF.

Arhitectura completă a procesorului(test_env)

Acestă entitate reprezintă arhitectura întregului procesorului și are rolul de a asambla întregul procesor folosind componentele mai sus enumerate.

Execuția lentă la varianta cu ciclu unic este datorată perioadei de ceas, care trebuie să acopere propagarea semnalelor pe calea cea mai lungă (calea critică), întâlnită în cazul instrucțiunii load word (LW). În consecință, restul instrucțiunilor vor suferi aceeași întârziere. Durata perioadei se poate reduce prin separarea diferitelor elemente din calea de date combinațională cu ajutorul unor registre de memorare a rezultatelor intermediare (Registrele Pipeline).

Registrele pipeline vor fi identificate conform cu etapele de execuție pe care le separă, astfel: IF/ID, ID/EX, EX/MEM, MEM/WB. Ele memorează rezultatele de la etajul anterior și le pun la dispoziție elementelor funcționale din etajul următor. În acest fel, în pipeline se pot executa concomitent până la 5 instrucțiuni consecutive, fiecare în etape diferite.

Toate componentele mai sus enumerate sunt funcționale, niciuna nu prezintă probleme de funcționare. Acestea au fost testate integral pe plăcuță pe parcursul laboratoarelor și problemele apărute au fost remediate pe parcurs. De asemenea unele componente au fost testate și în simulatorul din VIVADO (IFetch.vhd, EX.vhd, ID.vhd, test_env.vhd) pentru a fii sigur de funcționalitatea proiectului.

Sigura problemă a proiectul ar fii la încărcarea pe plăcuța Nexys A7, la prima iterație, când execută prima instrucțiune beq acesta sare la ultima instrucțiune din program, după care programul se execută corect fără nici-o problemă și este calculat și rezultatul dorit.

Pe parcursul realizării proiectului s-au întâmpinat diferite probleme de proiectare, valori puse greșit sau greșeli la proiectarea programului MIPS32. Acestea au fost remediate cu ajutorul cadrului didactic de la laborator sau prin testarea fiecărei componente în parte și remediarea problemelor pas cu pas. Hazardurile au fost rezolvate în întregime (se poate observa la Diagrama de execuție pipeline) iar singura problemă rămasă ar fii cea menționată mai sus.

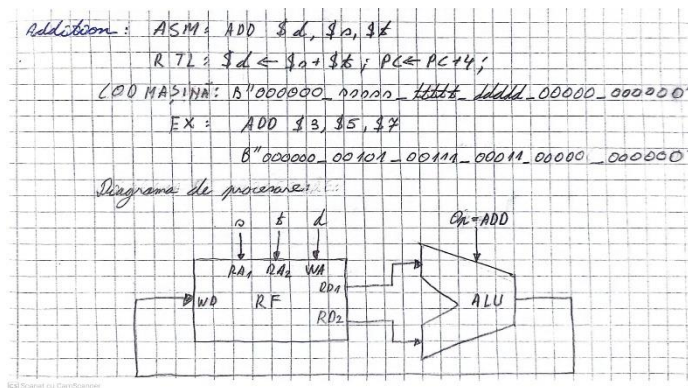
2. Instrucțiuni pentru MIPS 32

Setul de instrucțiuni ales pentru implementarea procesorului MIPS 32 pipeline este după cum urmează:

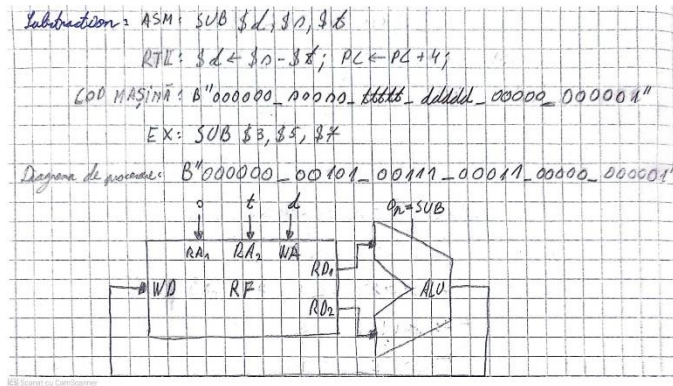
TIP R	Addition	add
	Subtraction	sub
	Shift Left Logical	sll
	Shift Right Logical	srl
	Logical AND	and
	Logical OR	or
	Logical Exclusive-OR	xor
	Set on Less Than	slt
TIP I	Add Immediate	addi
	Load Word	lw
	Store Word	sw
	Branch on Equal	beq
	OR Immediate	ori
	Set on Less Than Immediate	slti
TIP J	Jump	j

Cele 4 instrucțiuni ale suplimentar sunt: Logical Exclusive-OR (xor), Set on Less Than (slt), OR Immediate (ori) și Set on Less Than Immediate (slti).

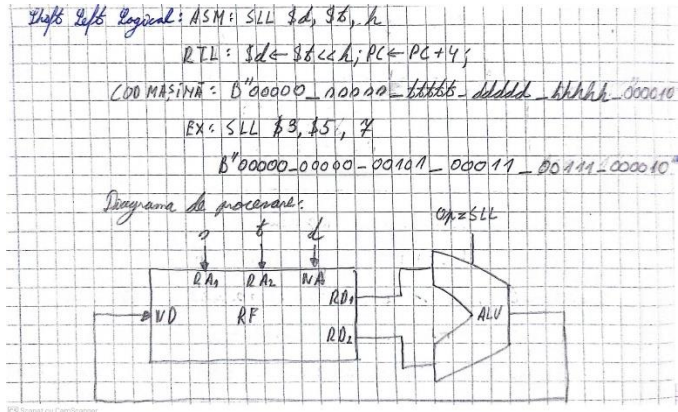
Addition



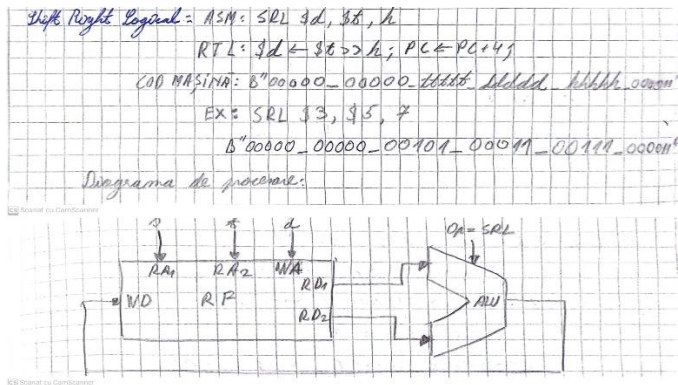
Subtraction



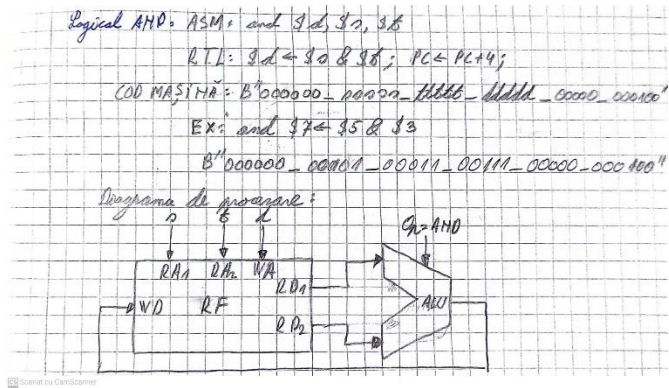
Shift Left Logical



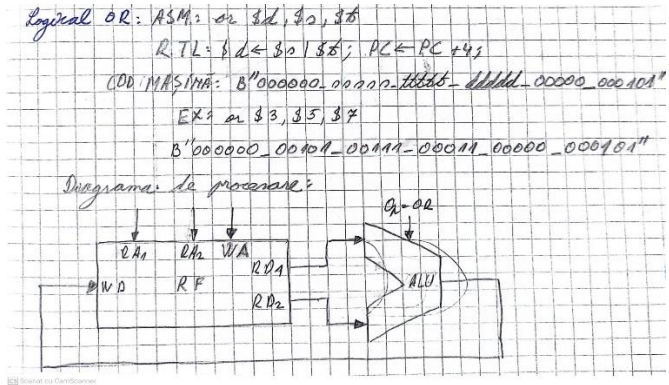
Shift Right Logical



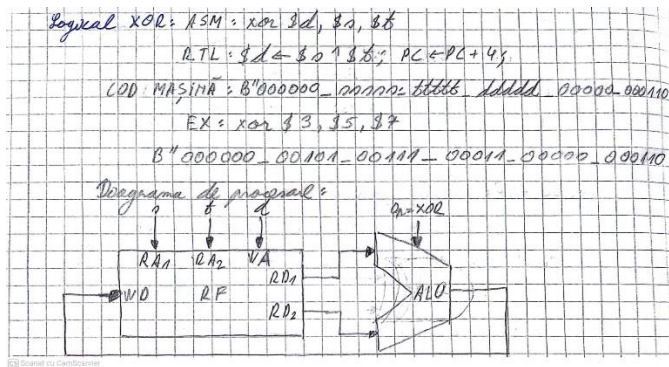
Logical And



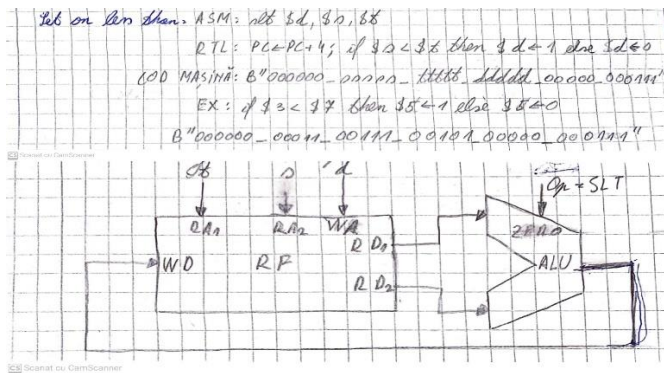
Logical OR



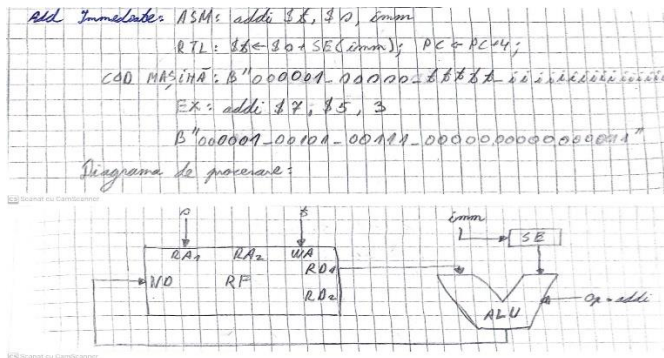
Logical Exclusive-OR



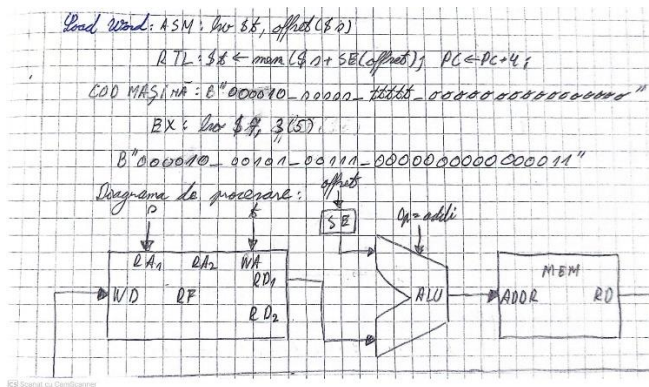
Set on Less Than



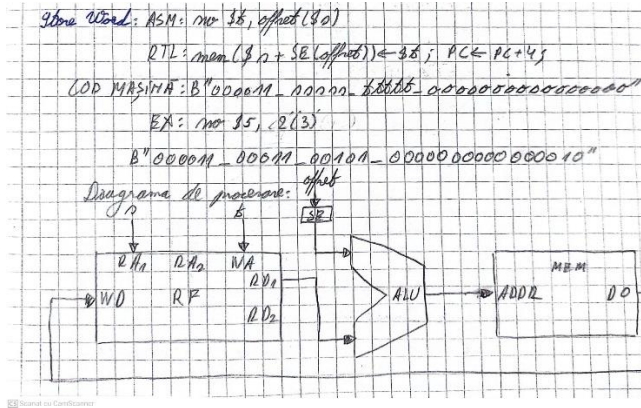
Add Immediate



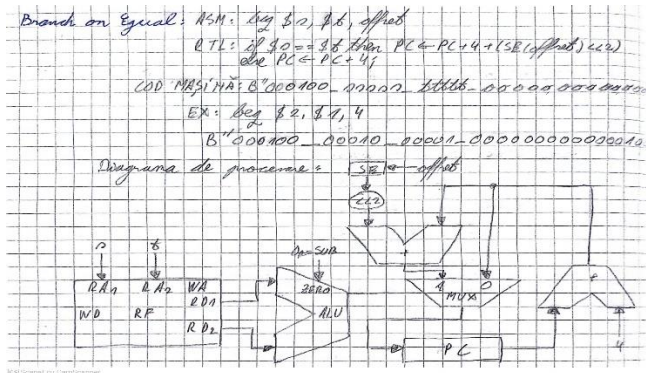
Load Word



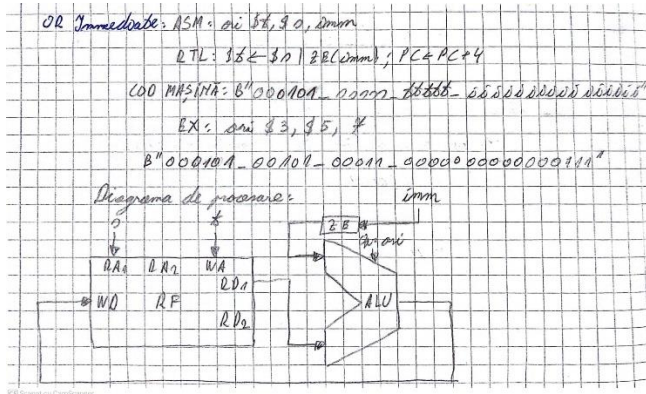
Store Word



Branch on Equal



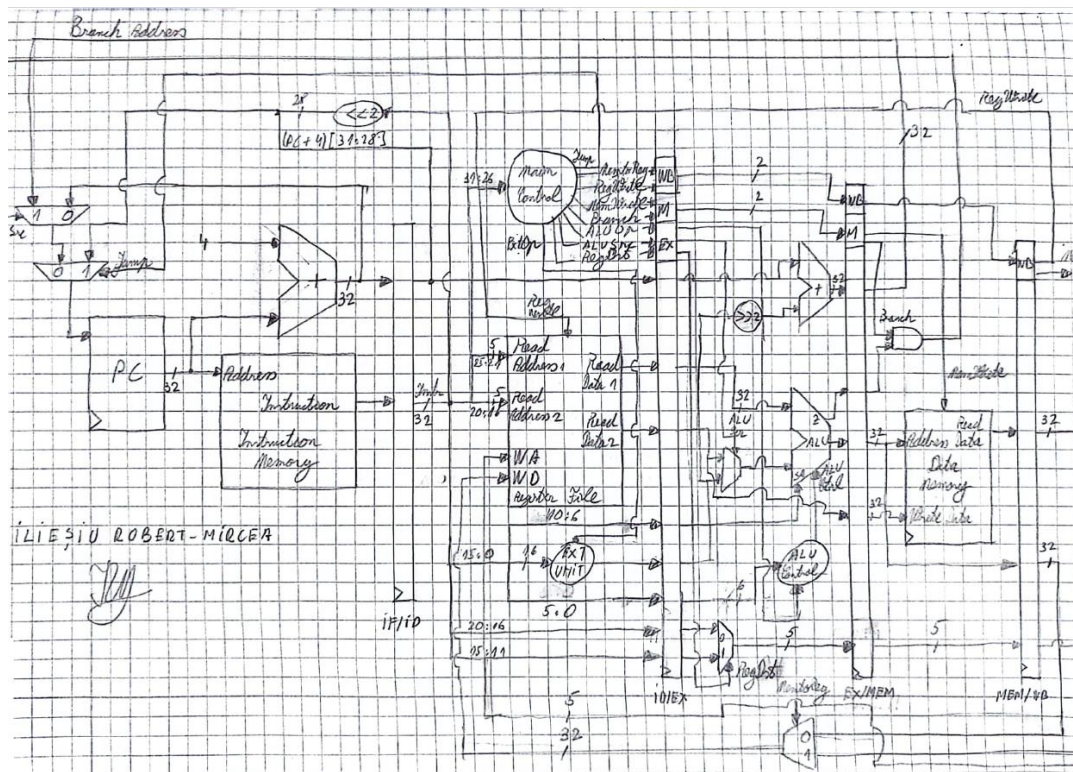
OR Immediate



Set on Less Than Immediate

[illegible]

3. Schema procesorului pipeline



4. Proiectarea registrelor pipeline

Mai jos avem tabelul cu proiectarea registrelor pipeline pentru fiecare semnal în parte:

Configurare registre MIPS32 Pipeline – varianta 2

[illegible]

5. Diagrama de execuție pipeline

Adr. Instr./Ciclu	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22
0 addi \$1,\$0,0	IF																					
1 addi \$2,\$0,0		ID																				
2 lw \$3,\$4(\$0)			ID																			
3 add \$4,\$0,\$3				ID																		
4 add \$5,\$0,\$0					ID																	
5 addi \$8,\$0,0						ID																
6 beq \$1,\$4,0							ID															
7 lw \$6,\$12(\$2)								ID														
8 addi \$10,\$0,1									ID													
9 and \$7,\$6,\$10										ID												
10 beq \$7,\$10,3											ID											
11 slt \$1,\$5,\$6												ID										
12 beq \$1,\$0,1													ID									
13 add \$5,\$6,\$0														ID								
14 addi \$2,\$2,4															ID							
15 addi \$1,\$1,1																ID						
16 j 6																	ID					
17 sw \$5,8(\$0)																		ID				
18																			ID			
19																				ID		
20																					ID	
21																						ID
22																						WB

Adre	Instr./Ciclu	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35	C36
0	lw \$3,8(\$0)	IF	ID	EX	MEM	WB																															
1	NoOp		ID	EX	MEM	WB																															
2	NoOp			ID	EX	MEM	WB																														
3	addi \$4,\$0,13				ID	EX	MEM	WB																													
4	lw \$1,4,28					ID	EX	MEM	WB																												
5	NoOp						ID	EX	MEM	WB																											
6	NoOp							ID	EX	MEM	WB																										
7	NoOp								ID	EX	MEM	WB																									
8	lw \$6,\$12(\$2)									ID	EX	MEM	WB																								
9	addi \$10,\$0,1										ID	EX	MEM	WB																							
10	NoOp											ID	EX	MEM	WB																						
11	NoOp												ID	EX	MEM	WB																					
12	and \$7,\$6,\$10													ID	EX	MEM	WB																				
13	NoOp														ID	EX	MEM	WB																			
14	NoOp															ID	EX	MEM	WB																		
15	NoOp																ID	EX	MEM	WB																	
16	NoOp																	ID	EX	MEM	WB																
17	NoOp																		ID	EX	MEM	WB															
18	NoOp																			ID	EX	MEM	WB														
19	lw \$8,\$5,\$6																				ID	EX	MEM	WB													
20	NoOp																					ID	EX	MEM	WB												
21	NoOp																						ID	EX	MEM	WB											
22	lw \$18,\$0,4																							ID	EX	MEM	WB										
23	NoOp																								ID	EX	MEM	WB									
24	NoOp																									ID	EX	MEM	WB								
25	NoOp																										ID	EX	MEM	WB							
26	addi \$5,\$6,40																											ID	EX	MEM	WB						
27	addi \$2,\$2,4																												ID	EX	MEM	WB					
28	addi \$1,\$1,1																													ID	EX	MEM	WB				
29	j 4																														ID	EX	MEM	WB			
30	NoOp																															ID	EX	MEM	WB		
31	sw \$5,8(\$0)																																ID	EX	MEM	WB	

6. Schema RTL

