

Documentație procesor MIPS32

varianta pipeline

Arhitectura Calculatoarelor

Student: Codorean Luca-Andrei, CTI-ro, 30223

Cadru didactic îndrumător: Nandra Constantin

Aprilie, 2024

Această documentație cuprinde elementele folosite în realizarea procesorului MIPS32 realizat în cadrul laboratorului de Arhitectura Calculatoarelor. Acest document reprezintă continuarea documentului AC_MIPS, document în care s-a prezentat implementarea procesorului folosind varianta ciclu unic. În această implementare, s-au folosit pipeline-uri.

Cuprins

Registrele procesorului MIPS32 Pipeline	3
Diagrama de execuție pipeline pentru programul de test MIPS32	4

Registreele procesorului MIPS32 Pipeline

Pentru implementarea variantei pipeline se folosesc 4 registre care au ca scop păstrarea stării curente a datelor, și ulterior distribuirea lor în etajele superioare ale logicii de calcul. Astfel, se îmbunătățește timpul de răspuns rezultat din varianta ciclului unic. Registrele implementate sunt:

1. REG_IF_ID;
2. REG_ID_EX;
3. REG_EX_MEM; respectiv
4. REG_MEM_WB.

Structura acestor registre este descrisă de tabelul atașat.

REG_IF_ID[63:0]	REG_ID_EX[158:0]	REG_EX_MEM[106:0]	REG_MEM_WB[70:0]
Instruction [63:32]	MEM_TO_REG [158]	MEM_TO_REG [106]	MEM_TO_REG [70]
PCp4 [31:0]	REG_WRITE [157]	REG_WRITE [105]	REG_WRITE [69]
	MEM_WRITE [156]	MEM_WRITE [104]	READ_DATA [68:37]
	BRANCH [155:154]	BRANCH [103:102]	ALU_RES [36:5]
	ALU_OP [153:151]	(PCp4 + (EXT_IMM << 2)) [101:70]	(INSTR [20:16] INSTR [15:11]) [4:0]
	ALU_SRC [150]	ZERO [69]	
	REG_DST [149]	ALU_RES [68:37]	
	PCp4[148:117]	READ_DATA_2 [36:5]	
	READ_DATA_1 [116:85]	(INSTR [20:16] INSTR [15:11]) [4:0]	
	READ_DATA_2 [84:53]		
	EXT_IMM [52:21]		
	INSTRUCTION [20:0]		

Diagrama de execuție pipeline pentru programul de test MIPS32

Implementarea pipeline-uri presupune rularea propriu-zisă a instrucțiunii după cinci cicluri de ceas. Acest aspect se datorează faptului că instrucțiunea trebuie să treacă prin toate etajele logicii pentru a putea fi executată. Astfel, pot să apară situații în care există hazard. Conceptul de hazard descrie un comportament necorespunzător al instrucțiunilor, în cazuri speciale. Există trei tipuri de hazard-uri pe care le abordăm.

1. Hazardul de date;
2. Hazardul structural; respectiv
3. Hazardul de control.

Hazardul de date presupune că situațiile de tip read-after-write, respectiv load data hazard. Acest tip de hazard poate să apară până la trei instrucțiuni în avans față de instrucțiunea curentă. Soluția generală de soluționare a acestui tip de hazard este introducerea unei întârzieri, simulate prin intermediul pseudo-instrucțiunii NoOp (*no operation*).

Pseudo-instrucțiunea NoOp nu este o instrucțiune propriu-zisă. În implementarea acestui procesor, aceasta este simulată de instrucțiunea *xor \$0, \$0, \$0*, care nu are nici un efect în logica de calcul.

Hazardul structural apare în situația în care se utilizează simultan o resursă (registru) de două operații. . Acest tip de hazard poate să apară între instrucțiunea curentă și cea la distanță 3 față de cea curentă. Soluționarea acestui caz de hazard este, în procesorul de față, realizarea scrierii la frontul descendent al semnalului de ceas în entitatea REGISTER FILE.

Hazardul de control apare în situațiile în care se realizează instrucțiuni de salt. În funcție de natura saltului, există mai multe modalități de soluționare. Un aspect important este că adresa la care se realizează saltul se modifică.

- 1) Tratarea instrucțiunilor de salt condiționat.
Hazardul generat de salturile condiționate (branch) este soluționat prin introducerea a trei pseudo-instrucțiuni de tip NoOp după instrucțiunea curentă.
- 2) Tratarea instrucțiunilor de salt necondiționat.

Modalitatea de tratare a hazardului în cazul instrucțiunilor de salt necondiționat (instrucțiuni de tip J), este în cazul de față, introducerea unei pseudo-instrucțiuni NoOp după aceasta.

20	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
21	0 XOR \$5, \$5, \$5	IF	ID	EX	MEM	WB(\$5)																	
22	1 ADDI \$6, \$0, 0		IF	ID	EX	MEM	WB(\$6)																
23	2 ADD \$1, \$0, \$0			IF	ID	EX	MEM	WB(\$1)															
24	3 ADDI \$4, \$0, 15				IF	ID	EX	MEM	WB(\$4)														
25	4 XOR \$2, \$2, \$2					IF	ID	EX	MEM	WB(\$2:D-6)													
26	5 XOR \$7, \$7, \$7						IF	ID	EX	MEM	WB(\$7)												
27	6 BEQ \$4, \$1, 8							IF	ID	EX	MEM(C)	WB											
28	7 LW \$3, 0(\$2)								IF	ID(\$2)	EX	MEM	WB(\$3:D-8)	WB(\$7)									
29	8 MODI \$7, \$3, 3									IF	ID(\$3)	EX	IF	ID	EX	MEM	WB(\$5)						
30	9 BNEQ \$7, \$0, 1										IF	ID	EX	MEM(C)	WB								
31	10 ADD \$5, \$3, \$5											IF	ID	EX	MEM	WB(\$5)							
32	11 ADD \$6, \$6, \$6												IF	ID	EX	MEM	WB(\$6)						
33	12 ADDI \$2, \$2, 4													ID	EX	MEM	WB(\$2)						
34	13 ADDI \$1, \$1, 1														IF	ID	EX	MEM	WB(\$1)				
35	14 J 6															IF	ID	EX	MEM	WB			
36	15 XOR \$7, \$7, \$7																IF	ID	EX	MEM	WB(\$7)		
37	16 OR \$7, \$6, \$5																	IF	ID	EX	MEM	WB(\$7)	
38	17 SW \$7, 60(\$0)																		IF	ID	EX	MEM	WB(\$0)

Figură 1. Identificarea hazardurilor

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	C23	C24	C25	C26	C27	C28	C29	C30	C31
0 XOR \$5, \$5, \$5	IF	ID	EX	MEM	WB(\$5)																										
1 ADDI \$6, \$0, 0		IF	ID	EX	MEM	WB(\$6)																									
2 ADD \$1, \$0, \$0			IF	ID	EX	MEM	WB(\$1)																								
3 ADDI \$4, \$0, 15				IF	ID	EX	MEM	WB(\$4)																							
4 XOR \$2, \$2, \$2					IF	ID	EX	MEM	WB(\$2)																						
5 XOR \$7, \$7, \$7						IF	ID	EX	MEM	WB(\$7)																					
6 BEQ \$4, \$1, 17							IF	ID	EX	MEM	WB																				
7 NoOp								IF	ID	EX	MEM	WB																			
8 NoOp									IF	ID	EX	MEM	WB																		
9 NoOp										ID	EX	MEM	WB																		
10 LW \$3, 0(\$2)										IF	ID	EX	MEM	WB																	
11 NoOp											ID	EX	MEM	WB																	
12 NoOp												IF	ID	EX	MEM	WB															
13 MODI \$7, \$3, 3													IF	ID	EX	MEM	WB(\$7)	WB													
14 BNEQ \$7, \$0, 4														ID	EX	MEM	WB														
15 NoOp															IF	ID	EX	MEM	WB												
16 NoOp																IF	ID	EX	MEM	WB											
17 NoOp																	IF	ID	EX	MEM	WB										
18 ADD \$5, \$3, \$5																		ID	EX	MEM	WB(\$5)										
19 ADD \$6, \$3, \$6																			ID	EX	MEM	WB(\$6)									
20 ADDI \$2, \$2, 4																			ID	EX	MEM	WB(\$2)									
21 ADDI \$1, \$1, 1																			ID	EX	MEM	WB(\$1)									
22 J 6																			ID	EX	MEM	WB									
23 NoOp																				IF	ID	EX	MEM	WB							
24 XOR \$7, \$7, \$7																					ID	EX	MEM	WB(\$7)							
25 OR \$7, \$6, \$5																					ID	EX	MEM	WB(\$7)							
26 SW \$7, 60(\$0)																					ID	EX	MEM	WB(\$0)							

Figură 2. Soluționarea hazardurilor

SCHEMA PROCESORULUI MIPS 32 PIPE LINE

CODOREAN LOCA-ANDREI
30223-1, CŢI=20

Gua 15.0

15.05.2024



