



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

Procesorul MIPS16 ciclu unic

Remeș Daria-Maria

Grupa 30223

Cuprins

- a. Cele 4 instrucțiuni suplimentare
- b. Semnale de control MIPS16
- c. Cod sursa și descriere a programului implementat
- d. Trasare execuție
- e. Menționări activități
- f. Explicații descriere
- g. Detalii simulare

a. Cele 4 instrucțiuni suplimentare

1. Tipul R

- **XOR** – SAU exclusiv logic între două registre, memorează rezultatul în alt registru

ASM xor \$rd, \$rs, \$rt
RTL $RF[rd] \leftarrow RF[rs] \wedge RF[rt]$; $PC \leq PC+1$;

- **MULT** – înmulțire între două registre, memorează rezultatul în alt registru

ASM mult \$rd, \$rs, \$rt
RTL $RF[rd] \leftarrow RF[rs] * RF[rt]$; $PC \leq PC+1$;

2. Tipul I

- **BGTZ** – salt condiționat dacă un registru este mai mare ca 0

ASM bgtz \$rs, offset
RTL if $RF[rs] > 0$ $PC \leq PC+1+offset \ll 2$; else $PC \leq PC+1$;

- **ORI** – SAU logic între un registru și o valoare imediată, memorează rezultatul în alt registru

ASM ori \$rt, \$rs, imm
RTL $RF[rt] \leftarrow RF[rs] \vee imm$; $PC \leq PC+1$;

b. Semnale de control MIPS16

Instrucțiune	Opcode <i>Instr(15-13)</i>	RegDst	ExtOp	ALUSrc	Branch	<Br?> (opțional)	Jump	JumpR (opțional)	Mem Write	Memto Reg	Reg Write	ALUOp (1:0)	func <i>Instr(2-0)</i>	ALUCtrl (2:0)
ADD (tip R)	000	1	x	0	0		0		0	0	1	10 (+)	000	000 (+)
SUB (tip R)	000	1	x	0	0		0		0	0	1	10 (-)	001	001(-)
SLL (tip R)	000	1	x	0	0		0		0	0	1	10 (<<I)	010	010(<<I)
SRL (tip R)	000	1	x	0	0		0		0	0	1	10 (>>I)	011	010(<<I)
AND (tip R)	000	1	x	0	0		0		0	0	1	10 (&)	100	011(&)
OR (tip R)	000	1	x	0	0		0		0	0	1	10 ()	101	100()
XOR(tip R)	000	1	x	0	0		0		0	0	1	10 (^)	110	101(^)
MULT (tip R)	000	1	x	0	0		0		0	0	1	10 (*)	111	110(*)
ADDI (tip I)	001	0	1	1	0		0		0	0	1	00 (+)	X	000
LW (tip I)	010	0	1	1	0		0		0	1	1	00(+)	x	000
SW (tip I)	011	x	1	1	0		0		1	x	0	00(+)	x	000
BEQ (tip I)	100	x	1	0	1		0		0	x	0	01(-)	x	001
BGTZ (tip I)	101	x	1	0	1		0		0	x	0	01(-)	x	001
ORI (tip I)	110	0	1	1	0		0		0	0	1	11()	X	101()
J (tip J)	111	x	x	x	x		1		0	x	0	xx	x	111

c. Cod sursa și descriere a programului implementat

Programul implementat calculează suma pătratelor a 5 numere aflate în memorie.

Sursa in limbaj C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int A[5] = {1, 2, 3, 4, 5};
7      int sumpatrat = 0;
8      for(int i = 0; i < 5; i++)
9      {
10         A[i] = A[i] * A[i];
11         sumpatrat = sumpatrat + A[i];
12     }
13     return 0;
14 }
15
```

Cod mașină

```
B"000_000_000_001_0_000", --X"0010" --add $1,$0,$0 -0
B"001_000_100_0000101", --X"2205" --addi $4,$0,5 -1
B"000_000_000_101_0_000", --X"0050" --add $5,$0,$0 -2
B"100_001_100_0000110", --X"8606" --beq $1,$4,6(end_loop) -3
B"010_001_011_0000000", --X"4580" --lw $3,0($1)(A_addr) -4
B"000_011_011_110_0_111", --X"0DE7" --mult $6,$3,$3 -5
B"011_001_110_0000000", --X"6700" --sw $6,0($1)(A_addr) -6
B"000_101_110_101_0_000", --X"1750" --add $5,$5,$6 -7
B"001_001_001_0000001", --X"2481" --addi $1,$1,1 -8
B"111_0000000000011", --X"E003" --j 3(begin_loop) -9
B"011_000_101_0001010", --X"628A" --sw $5,10($0)(sum_addr) -10
```

```
others => X"0000"); --NoOp -11
```

d. Trasare execuție

0. add \$1,\$0,\$0	Instruction= 1,	RD1= 0,	RD2= 0,	Ext_imm= 0,	ALURes= 0,	WD= 0
1. addi \$4,\$0,5	Instruction= 2,	RD1= 0,	RD2= 0,	Ext_imm= 5,	ALURes= 5,	WD= 5
2. add \$5,\$0,\$0	Instruction= 3,	RD1= 0,	RD2= 0,	Ext_imm= 0,	ALURes= 0,	WD= 0
3. beq \$1,\$4,6	Instruction= 4,	RD1= 0,	RD2= 5,	Ext_imm= 6,	ALURes= FFFB,	WD= FFFB
4. lw \$3,0(\$1)	Instruction= 5,	RD1= 0,	RD2= 0,	Ext_imm= 0,	ALURes= 0,	WD= 1
5. mult \$6,\$3,\$3	Instruction= 6,	RD1= 1,	RD2= 1,	Ext_imm= 0,	ALURes= 1,	WD= 1
6. sw \$6,0(\$1)	Instruction= 7,	RD1= 0,	RD2= 1,	Ext_imm= 0,	ALURes= 0,	WD= 0
7. add \$5,\$5,\$6	Instruction= 8,	RD1= 0,	RD2= 1,	Ext_imm= 0,	ALURes= 1,	WD= 1
8. addi \$1,\$1,1	Instruction= 9,	RD1= 0,	RD2= 0,	Ext_imm= 1,	ALURes= 1,	WD= 1
9. j 3(begin_loop)	Instruction= A,	RD1= 0,	RD2= 0,	Ext_imm= 3,	ALURes= 0,	WD= 0
10. sw \$5,10(\$0)	Instruction= b,	RD1= 0,	RD2= 37,	Ext_imm= 10,	ALURes= 0,	WD= 0

e. Menționări activități

Nu există părți din procesor incomplete.

f. Explicații descriere

Nu există erori.

g. Detalii simulare

Am testat pe placă programul. Rezultatul final este cel asteptat, la adresa solicitată.