Documentatie

Mips ciclu unic 32 de biti

Nume: Saca Victor-Valentin

Setul de instructiuni utilizat, cu codul RTL corespunzator

sw

add $d, $s, $t : d = s + t ; 000000 sssss ttttt ddddd 00000 000000

sub $d, $s, $t : d = s - t, pc += 4 000000 sssss ttttt ddddd 00000 100001

sll $d, $t, h : $d = $t << h, pc+= 4 ; 000000 00000 ttttt ddddd hhhhh 000010

srl $d, $t, h : d = t << h, pc += 4 ; 000000 00000 ddddd ttttt hhhhh 000011

and $d, $s, $t : d = s and t ; 000000 sssss ttttt ddddd 00000 000100

or $d, $s, $t : d = s and t, PC += 4 ; 000000 sssss ttttt ddddd 00000 000101

slt $d, $s, $t : if s < t then d = 1 else d = 0 ; 000000 sssss ttttt ddddd 00000 000110

noop : 0 = 0 << 0; 000000 00000 00000 00000 00000 000111

I

addi $t, $s, imm : t = s + SE(imm) ; 001000 sssss ttttt iiiiiiiiiiiiiii

lw $t, offset($s) : t = mem[s + SE(offset)]; pc += 4 ; 100011 sssss ttttt oooooooooooooooo

sw $t, offset($s) ; 101011 sssss ttttt oooooooooooooooo

beq $s, $t, offset : if s == t then pc += 4 + (SE(offset) << 2) else pc +=4 ; 000100 sssss ttttt oooooooooooooooo

bgez $s, offset : if $s >= 0 then PC += 4 + (SE(offset) << 2); 000001 sssss 00000 oooooooooooooooo bne $s, $t, offset: if $s ≠ $t then PC = (PC + 4) + (SE(offset) << 2) else PC = PC + 4; 000101 sssss ttttt oooooooooooooooo

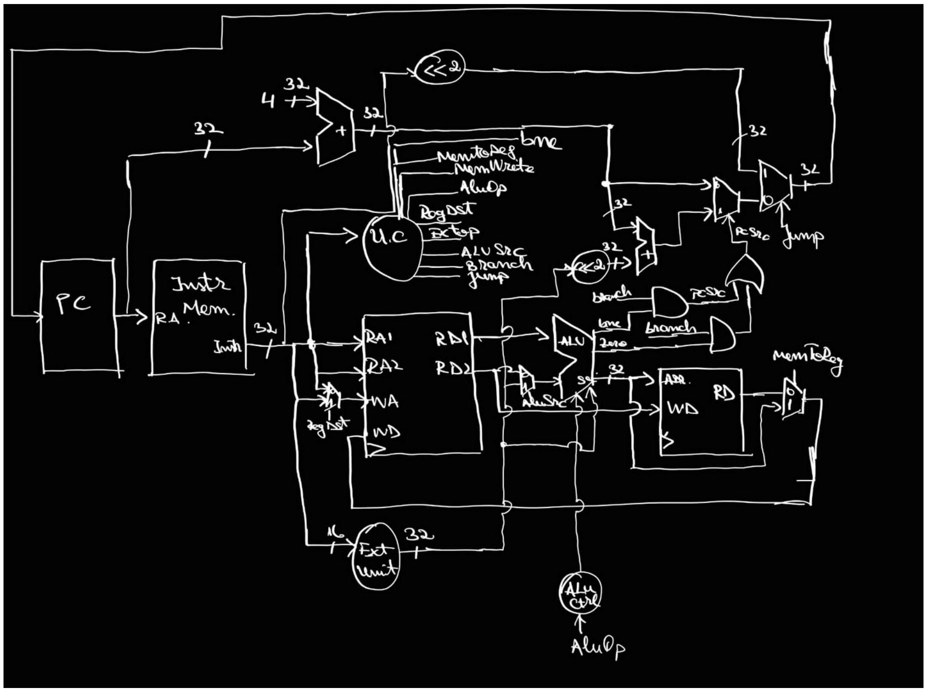
J

j addr : pc = (pc + 4)[31:28] || (addr << 2) ; 000010 aaaaaaaaaaaaaaaaaaaaaaaaaa

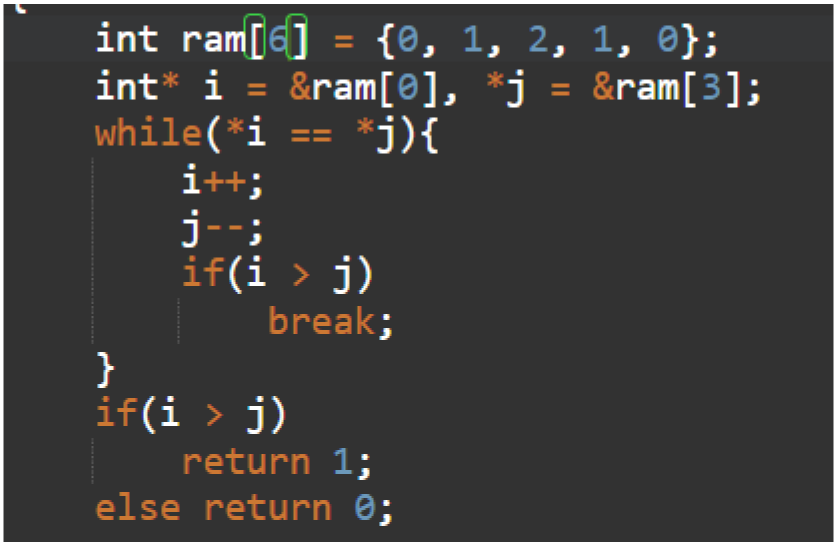
Setul de semnale de control:

instr	regDst	extOp	aluSrc	Branch	Jump	aluOp	memWrite	memReg	regWrite	branchnz
add	0	1	0	0	0	000	0	0	1	0
Addi	0	1	1	0	0	001	0	0	1	0
Lw	0	1	1	0	0	000	0	1	1	0
Sw	X	1	1	0	0	000	1	X	0	0
Beq	X	1	0	1	0	001	0	X	0	0
Bne	X	1	0	1	0	001	0	X	1	1

Schema detaliata a proiectului:



Programul de testare implementat: Verificare secventa palindromica



```
    int ram[6] = {0, 1, 2, 1, 0};
    int* i = &ram[0], *j = &ram[3];
    while(*i == *j){
        i++;
        j--;
        if(i > j)
            break;
    }
    if(i > j)
        return 1;
    else return 0;
```

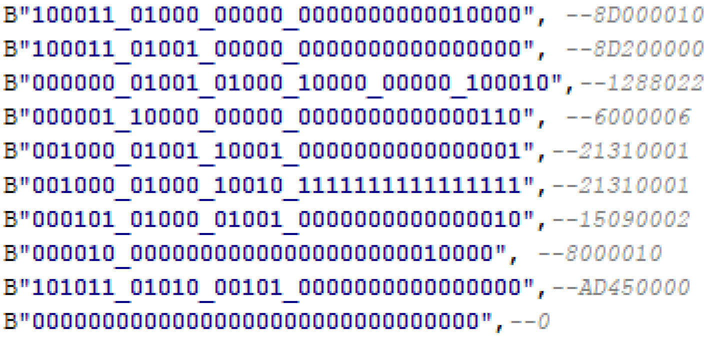
In cod assembly mips32:

lw t0, mem(4)

```
lw t1, mem(0)
sub s0, t1, t0
bgez s0, 28
addi s1, t1, 1
addi s2, t0, -1
bne $t0, $t1, 32
j 8
sw $t2, mem(0)
```

Programul nu merge corespunzator pe procesor, acesta nu merge pe ramurile bne si bgez.

Programul in cod masina:



```
B"100011_01000_00000_0000000000010000",  --8D000010
B"100011_01001_00000_0000000000000000",  --8D200000
B"000000_01001_01000_10000_00000_100010",--1288022
B"000001_10000_00000_0000000000000110",  --6000006
B"001000_01001_10001_0000000000000001",--21310001
B"001000_01000_10010_1111111111111111",--21310001
B"000101_01000_01001_0000000000000010",--15090002
B"000010_00000000000000000000010000",  --8000010
B"101011_01010_00101_0000000000000000",--AD450000
B"00000000000000000000000000000000",--0
```