1. **XOR** -> realizează operația XOR între biți celor două registre, și pune rezultatul în primul registru

ASM

Xor $d, $s, $t

ex: xor $1, $1, $2

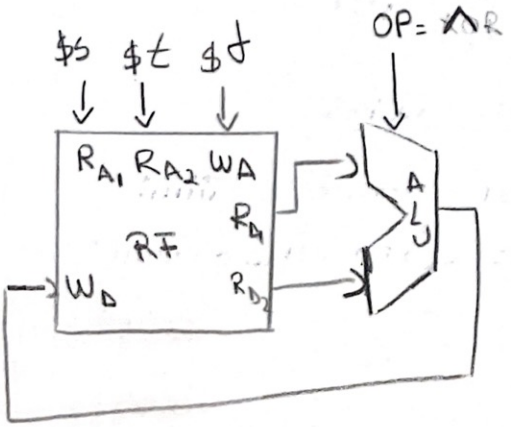
RTL

$d <- $s ^ $t; PC <- PC + 1

Cod mașină

B"000_sss_ttt_ddd_0_110"

ex: B"000_001_010_001_0_110"



2. **SLT** (Set on less than (signed), dacă $s < $t, $d este inițializat cu 1, altfel cu 0

ASM

slt $d, $s, $t

ex: slt $1, $1, $2

RTL

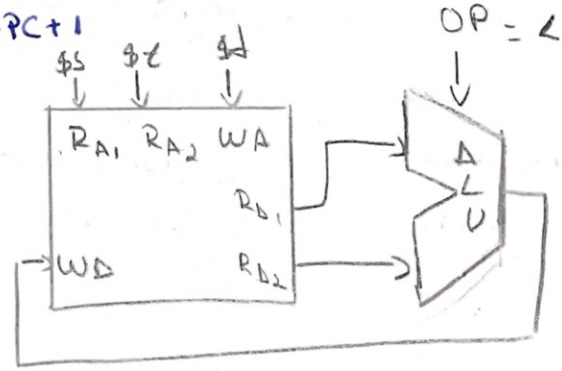
if $s < $t $d <- 1; else $d <- 0

PC <- PC + 1

Cod mașină

B"000_sss_ttt_ddd_0_111"

B"000_001_010_001_0_111"



3. BNE (Branch on not equal) salt conditionat dacă 2 registre sunt diferite. Dacă cele 2 registre sunt diferite sare la adresa ~~imm~~ dată de imm

ASM

bne $s, $t, imm

ex: bne $2, $3, 3

RTL

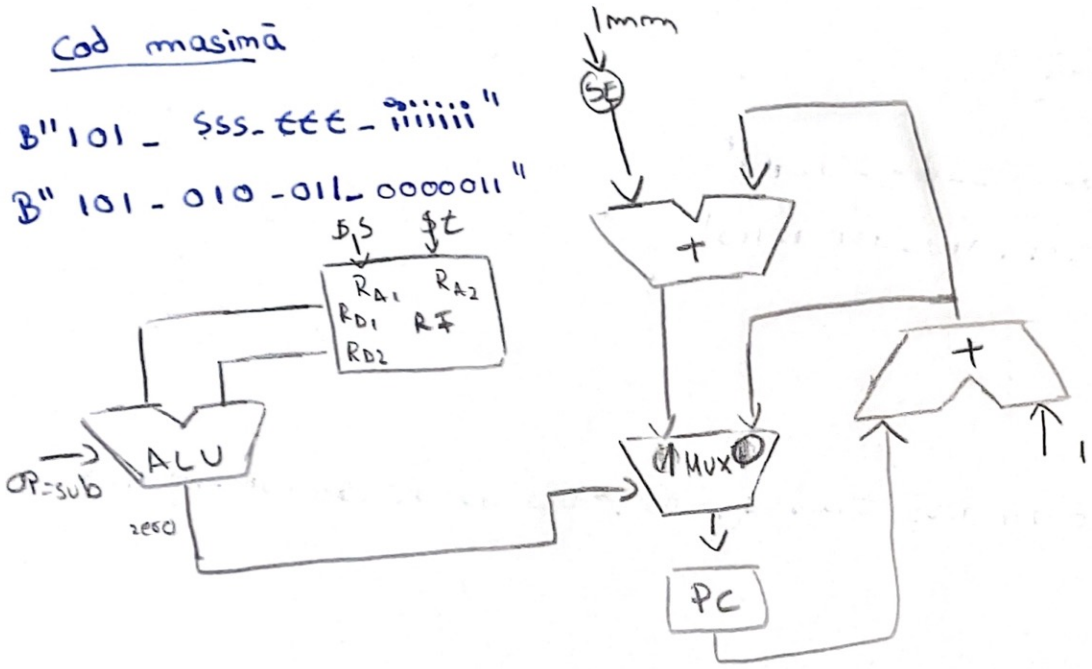
if ($s != $t) PC <- PC + 1 + SE(imm)

else PC <- PC + 1

Cod mașină

B"101 - sss - ttt - iiiiiiii"

B"101 - 010 - 011 - 0000011"



4. BGEZ (Branch on greater than on equal to zero), salt conditionat dacă un registru este mai mare sau egal cu 0

ASM

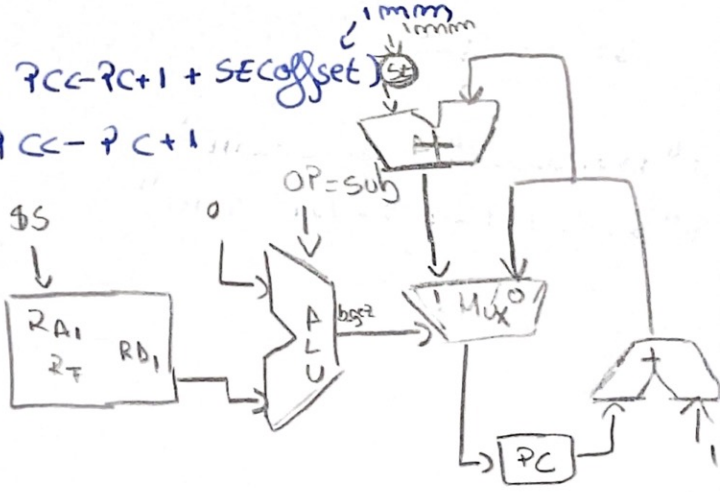
bgez $s, imm

ex bgez $4, 12

RTL

if $s ≥ 0 PC <- PC + 1 + SE(offset)

else PC <- PC + 1



Cod mașină

B"110 - sss - 001 - iiiiiiii"

B"110 - 100 - 001 - 0001100"