MipsPipeline nu a fost testat pe placa sau pe simulator din cauza erorilor de execuție care au fost rezolvate doar ulterior laboratorului.

Codul assembly a fost rescris complet deoarece erorile din codul initial erau severe, dar codificarile pentru ALU au ramas neschimbate, cu exceptia:

* Instrucțiunii AND, care acum utilizeaza funcția “100100”
* Adaugarea instrucțiunii XOR (tip r)
* Adaugarea instrucțiunii SLTI (tip i)

Am atașat valorile semnalelor de control și un fisier modificat pentru detalierea instrucțiunilor.

XOR – Bitwise exclusive OR

Instrucțiunea realizează or exclusiv logic pe doua registre, memorand rezultatul in registrul $d.

|  |  |
| --- | --- |
| Format asamblare: | xor $d, $s, $t |
| RTL: | $d <- $s ^ $t; PC <- PC + 4; |
| Format binar: | 000000 sssss ttttt ddddd 00000 100110 |
| Valorile semnalelor de control: | AluSrc <= '0';  extOp <= '0';  MemWrite <= '0';  MemToReg <= '0';  branch <= '0';  jump <= '0';  regDest <= '1';  regWrite <= '1';  AluOp <= "010"; |

SLTI – Set on less than immediate

|  |  |
| --- | --- |
| Format asamblare: | slti $t, $s, imm |
| RTL: | PC <- PC + 4; if $s < SE(imm) then $t <-1 else $t <- 0; |
| Format binar: | 001010 sssss ttttt iiiiiiiiiiiiiiii |
| Valorile semnalelor de control: | AluSrc <= '0';  extOp <= '1';  MemWrite <= '0';  MemToReg <= '0';  branch <= '0';  jump <= '0';  regDest <= '0';  regWrite <= '1';  AluOp <= "010"; |

Referințe:

<https://opencores.org/projects/plasma/opcodes>

<https://inst.eecs.berkeley.edu/~cs61c/resources/MIPS_help.html>