**Semnale control MIPS16 pentru Anexa 5**

<?> ϵ {\_gez, \_ne, \_gtz}

*Tipuri de operații care se pun în paranteză la ALUOp si ALUCtrl:* {(+), (-), (&), (|), (^), (<<*l*), (<<*lv*), (>>*l*), (>>*a*), (<)}, & - AND, | - OR, ^ *- XOR, l* *- logic, a - aritmetic, v - cu variabilă*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instrucțiune** | **Opcode** *Instr(15-13)* | **RegDst** | **ExtOp** | **ALUSrc** | **BranchEQ** | **BranchGTZ** | **Jump** | **JmpR** | **MemWrite** | **MemtoReg** | **Reg Write** | **ALUOp (1:0)** | **func**  *Instr(2-0)* | **ALUCtrl (2:0)** |
| ADD | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 000 | (+) 000 |
| SUB | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 001 | (-) 001 |
| AND | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 010 | (&) 010 |
| OR | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 011 | (|) 011 |
| XOR | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 100 | (^) 100 |
| SLL | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 101 | (>>) 101 |
| SLR | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 110 | (<<) 110 |
| SLT | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (R) 00 | 111 | (<) 111 |
| ADDI | 001 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (+) 01 | XXX | (+) 000 |
| BEQ | 010 | X | 1 | 0 | 1 | 0 | 0 | 0 | 0 | X | 0 | (-) 10 | XXX | (-) 001 |
| BGTZ | 011 | X | 1 | 0 | 0 | 1 | 0 | 0 | 0 | X | 0 | (-) 10 | XXX | (-) 001 |
| SW | 100 | X | 1 | 1 | 0 | 0 | 0 | 0 | 1 | X | 0 | (+) 01 | XXX | (+) 000 |
| LW | 101 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | (+) 01 | XXX | (+) 000 |
| J | 110 | X | X | X | X | X | 1 | 0 | 0 | X | 0 | XX | XXX | XXX |
| JR | 111 | X | X | X | X | X | X | 1 | 0 | X | 0 | XX | XXX | XXX |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

URL: <https://drive.google.com/file/d/1SI7x2Gp_2m3SEkwnXuGt4ns4voYzpGBH/view?usp=sharing>