**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** | **Branch** | **Br\_gtz** | **Jump** | **JmpR** (opțional) | **MemWrite** | **MemtoReg** | **Reg Write** | **ALUOp (1:0)** | **func**  *Instr(2-0)* | **ALUCtrl (2:0)** |
| ADD | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 000 | 0000(+) |
| SUB | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 001 | 0001(-) |
| AND | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 010 | 0010(&) |
| OR | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 011 | 0011(|) |
| XOR | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 100 | 0100(^) |
| SRL | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 101 | 0101(>>) |
| SLL | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 110 | 0110(<<) |
| SLLV | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(R) | 111 | 0111(<<lv) |
| LW | 001 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 1 | 1 | 01(+) | 000 | 0000(+) |
| BEQ | 010 | 0 | 1 | 0 | 1 | 0 | 0 |  | 0 | 0 | 0 | 10(-) | 000 | 0001(-) |
| ADDI | 011 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 1 | 01(+) | 000 | 0000(+) |
| SW | 100 | 0 | 1 | 1 | 0 | 0 | 0 |  | 1 | 0 | 0 | 01(+) | 000 | 0000(+) |
| BGTZ | 101 | 0 | 1 | 0 | 0 | 1 | 0 |  | 0 | 0 | 0 | 10(-) | 000 | 0001(-) |
| SLTI | 110 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 1 | 11(<) | 000 | 1000(<) |
| J | 111 | 0 | 0 | 0 | 0 | 0 | 1 |  | 0 | 0 | 0 | 00® | 000 | 0000(+) |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

URL: <https://drive.google.com/open?id=1SI7x2Gp_2m3SEkwnXuGt4ns4voYzpGBH>