**Semnale control MIPS16**

<?> 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** | **Branch1** | **Branch2** (opțional) | **Jump** | **Branch3**(opțional) | **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 |  | 000 | 000 |
| Sub | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00 | 001 | 001 |
| Sll | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00 | 010 | 010 |
| Srl | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00 | 011 | 011 |
| And | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00 | 100 | 100 |
| Or | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00 | 101 | 101 |
| Xor | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00 | 110 | 110 |
| Sllv | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00 | 111 | 111 |
| Addi | 001 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 01 | - | 000 |
| Lw | 010 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 01 | - | 000 |
| Sw | 011 | X | 1 | 1 | 0 | 0 | 0 | 0 | X | X | 0 | 01 | - | 000 |
| Beq | 100 | X | 1 | 1 | 1 | 0 | 0 | 0 | 0 | X | 0 | 10 | - | 001 |
| Bgtz | 110 | X | 1 | 0 | 0 | 0 | 0 | 1 | 0 | X | 0 | 10 | - | 100 |
| Bne | 101 | X | 1 | 1 | 0 | 1 | 0 | 0 | 0 | X | 0 | 10 | - | 001 |
| J | 111 | X | X | X | X | X | 1 | X | 0 | X | 0 | 11 | - | 100 |