**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** | Br\_ne | **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 | 00 (R) | 001 | 001 (+) |
| SUB | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00(R) | 010 | 010(-) |
| SLL | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00(R) | 011 | 011(<<l) |
| SRL | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00(R) | 111 | 111(>>l) |
| AND | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00(R) | 100 | 100(&) |
| OR | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00(R) | 101 | 101(|) |
| SRA | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00(R) | 110 | 110(>>a) |
| XOR | 000 | 1 | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00(R) | 000 | 000(^) |
| LW | 111 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 01(+) | XXX | 001(+) |
| SW | 010 | X | 1 | 1 | 0 | 0 | 0 | 0 | 1 | X | 0 | 01(+) | XXX | 001(+) |
| ADDI | 001 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 01(+) | XXX | 001(+) |
| BEQ | 011 | X | 1 | 0 | 1 | 0 | 0 | 0 | 0 | X | 0 | 10(-) | XXX | 010(-) |
| BGEZ | 100 | X | 1 | X | 1 | 1 | 0 | 0 | 0 | X | 0 | 10(-) | XXX | 010(-) |
| BNE | 101 | X | 1 | 0 | 1 | 0 | 0 | 1 | 0 | X | 0 | 10(-) | XXX | 010(-) |
| J | 110 | X | X | X | X | X | 1 | X | 0 | X | 0 | XX | XXX | XXX |