**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** | **BNE** | **Jump** | **JmpR** (opțional) | **MemWrite** | **MemtoReg** | **Reg Write** | **ALUOp (2:0)** | **func**  *Instr(2-0)* | **ALUCtrl (2:0)** |
| add | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 000(+) | 000(+) |
| sub | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 001(-) | 001(-) |
| sll | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 010(<<v) | 010(<<v) |
| srl | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 011(>>v) | 011(>>v) |
| and | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 100(&) | 100(&) |
| or | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 101(|) | 101(|) |
| subu | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 110(-) | 110(-a) |
| xor | 000 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 000 | 111(^) | 111(^) |
| addi | 001 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 1 | 001(+v) | XXX | 000(+) |
| lw | 010 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 1 | 1 | 010(+v) | XXX | 000(+) |
| sw | 011 | 0 | 1 | 1 | 0 | 0 | 0 |  | 1 | 0 | 0 | 011(+v) | XXX | 000(+) |
| beq | 100 | 0 | 1 | 0 | 1 | 0 | 0 |  | 0 | 0 | 0 | 100(-v) | XXX | 001(-v) |
| ori | 101 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 1 | 101(|v) | XXX | 101(|) |
| bne | 110 | 0 | 1 | 0 | 0 | 1 | 0 |  | 0 | 0 | 0 | 110(-v) | XXX | 001(-) |
| j | 111 | 0 | 0 | 0 | 0 | 0 | 1 |  | 0 | 0 | 0 | 111 | XXX | XXX |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

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