**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?>** (opțional) | **Jump** | **JmpR** (opțional) | **MemWrite** | **MemtoReg** | **Reg Write** | **ALUOp (2:0)** | **func**  *Instr(2-0)* | **ALUCtrl (2:0)** |
| ADD | 000 | 1 | X | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(+) | 000 | 000(+) |
| SUB | 000 | 1 | X | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(-) | 001 | 001(-) |
| SLL | 000 | 1 | 0 | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(<<) | 010 | 010(<<) |
| SRL | 000 | 1 | 0 | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(>>) | 011 | 011(>>) |
| AND | 000 | 1 | X | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(&) | 100 | 100(&) |
| OR | 000 | 1 | X | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(|) | 101 | 101(|) |
| XOR | 000 | 1 | X | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(^) | 110 | 110(^) |
| SLT | 000 | 1 | X | 0 | 0 |  | 0 |  | 0 | 0 | 1 | 000(<) | 111 | 111(<) |
| ADDI | 001 | 0 | 1 | 1 | 0 |  | 0 |  | 0 | 0 | 1 | 001(+) | XXX | 000(+) |
| LW | 010 | 0 | 1 | 1 | 0 |  | 0 |  | 0 | 1 | 1 | 001(+) | XXX | 000(+) |
| SW | 011 | X | 1 | 1 | 0 |  | 0 |  | 1 | 0 | 0 | 001(+) | XXX | 000(+) |
| BEQ | 100 | X | 1 | 0 | 1 |  | 0 |  | 0 | 0 | 0 | 010(-) | XXX | 001(-) |
| ANDI | 101 | 0 | 1 | 1 | 0 |  | 0 |  | 0 | 0 | 1 | 101(&) | XXX | 100(&) |
| ORI | 110 | 0 | 1 | 1 | 0 |  | 0 |  | 0 | 0 | 1 | 110(|) | XXX | 101(|) |
| J | 111 | X | X | 0 | 0 |  | 1 |  | 0 | 0 | 0 | XXX | XXX | XXX |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

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