**Semnale de control MIPS32**

Semnale de Branch opționale: ? ϵ {gez, ne, gtz}, se va înlocui ? cu o valoare din paranteză, dacă e cazul

Tipuri de operații care se pun în paranteză la ALUOp si ALUCtrl:

(+), (-), (&), (|), (^), (<<*l*), (<<*lv*), (>>*l*), (>>*a*), (<)

Semnificații: & - AND, | - OR, ^ *- XOR, l* *- logic, a - aritmetic, v - cu variabilă*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instrucțiune** | **Opcode**  *Instr[31-26]* | **RegDst** | **ExtOp** | **ALUSrc** | **Branch** | **Br\_GTZ** | **Jump** | **JmpR** (opțional) | **MemWrite** | **MemtoReg** | **Reg Write** | **ALUOp[1:0]** | **function**  *Instr[5-0]* | **ALUCtrl[2:0]** |
| ADD | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 000001 | 000(+) |
| SUB | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 000010 | 001(-) |
| SLL | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 000100 | 100<<l |
| SRL | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 000101 | 101>>l |
| AND | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 100000 | 010& |
| OR | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 100001 | 011| |
| SRA | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 000111 | 110>>a |
| SLT | 000000 | 1 | X | 0 | 0 | 0 | 0 |  | 0 | 0 | 1 | 10(R) | 001010 | 111(<) |
| ADDI | 001000 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 1 | 00(+) | xxxxxx | 000(+) |
| LW | 100010 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 1 | 1 | 00(+) | Xxxxxx | 000(+) |
| SW | 100011 | 0 | 1 | 1 | 0 | 0 | 0 |  | 1 | 0 | 0 | 00(+) | Xxxxxx | 000(+) |
| BEQ | 000100 | 0 | 1 | 0 | 1 | 0 | 0 |  | 0 | X | 0 | 01(-) | Xxxxxx | 001(-) |
| BGTZ | 000111 | 0 | 1 | 0 | 0 | 1 | 0 |  | 0 | X | 0 | 01(-) | Xxxxxx | 001(-) |
| SLTI | 001010 | 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 1 | 11(<) | Xxxxxx | 111(<) |
| JUMP | 000010 | X | X | X | X | X | 1 |  | 0 | X | 0 | XX | Xxxxxx | 000(+) |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |