|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тип | Код | | | Действие | | | Формат | | | | | | | | | | | | | | | | | |
| R-type | | | | | | | Funct7 | Rs2 | | | | Rs1 | | | Funct3 | | | Rd | | | Opcode | | | |
| Количество бит | | | | | | | 7 | 5 | | | | 5 | | | 3 | | | 5 | | | 7 | | | |
| R | ADD | | | rd = rs1 + rs2  If overflows low bits use | | | 0000000 |  | | | |  | | | 000 | | |  | | | 0000011 | | | |
| R | SLT | | | If rs1 < rs2 rd =1 | | | 0000000 |  | | | |  | | | 001 | | |  | | | 0000011 | | | |
| R | SLTU | | | If rs1<rs2 rd=1(only then rd2 !=1) | | | 0000000 |  | | | |  | | | 010 | | |  | | | 0000011 | | | |
| R | AND | | | rd = rs1 & rs2 | | | 0000000 |  | | | |  | | | 011 | | |  | | | 0000011 | | | |
| R | OR | | | rd = rs1 | rs2 | | | 0000000 |  | | | |  | | | 100 | | |  | | | 0000011 | | | |
| R | XOR | | | rd = rs1 ^ rs2 | | | 0000000 |  | | | |  | | | 101 | | |  | | | 0000011 | | | |
| R | SLL | | | rd = rs1 << rs2[4:0] | | | 0000000 |  | | | |  | | | 110 | | |  | | | 0000011 | | | |
| R | SRL | | | rd = rs1 >> rs2[4:0] | | | 0000000 |  | | | |  | | | 111 | | |  | | | 0000011 | | | |
| R | SUB | | | rd =rs1 – rs2 | | | 0100000 |  | | | |  | | | 000 | | |  | | | 0000011 | | | |
| R | SRA | | | rd = rs1 >>> rs2[4:0] | | | 0100000 |  | | | |  | | | 001 | | |  | | | 0000011 | | | |
| R | AM | | | Rd = (rs1+rs2)/2 | | | 0100000 |  | | | |  | | | 010 | | |  | | | 0000011 | | | |
| I-type(R) | | | | | | | Imm | | | | | Rs1 | | | Funct3 | | | Rd | | | Opcode | | | |
| Количество бит | | | | | | | 12 | | | | | 5 | | | 3 | | | 5 | | | 7 | | | |
| I | | ADDI | | | Rd = rs1+ se imm | |  | | | | |  | | | 000 | | |  | | | 1000011 | | | |
| I | | SLTI | | | If se imm > rs1 then rd = 1 | |  | | | | |  | | | 001 | | |  | | | 1000011 | | | |
| I | | ANDI | | | Rd = rs1 & se imm | |  | | | | |  | | | 010 | | |  | | | 1000011 | | | |
| I | | ORI | | | Rd =rs1 | se imm | |  | | | | |  | | | 011 | | |  | | | 1000011 | | | |
| I | | XORI | | | Rd = rs ^ se imm | |  | | | | |  | | | 100 | | |  | | | 1000011 | | | |
| I-type(R) | | | | | | | Imm | | Shamt | | | Rs1 | | | Funct3 | | | Rd | | | Opcode | | | |
| Количество бит | | | | | | | 7 | | 5 | | | 5 | | | 3 | | | 5 | | | 7 | | | |
| I | | SLLI | | | Rd = rs1<< shamt | | 0000000 | |  | | |  | | | 101 | | |  | | | 1000011 | | | |
| I | | SRLI | | | Rd= rs1>>shamt | | 0000000 | |  | | |  | | | 110 | | |  | | | 1000011 | | | |
| I | | SRAI | | | Rd=rs1>>>shamt | | 0100000 | |  | | |  | | | 111 | | |  | | | 1000011 | | | |
| U-type | | | | | | | Imm | | | | | | | | Rd | | | Opcode | | | | | | |
| Количество бит | | | | | | | 20 | | | | | | | | 5 | | | 7 | | | | | | |
| U | | LUI | | | Rd[31:12] = imm  Rd[11:0] = 0 | |  | | | | | | | |  | | | 0100011 | | | | | | |
| U | | AUIPC | | | Rd= pc+ A[31:0];  A[31:12]=imm;  A[11:0]=0 | |  | | | | | | | |  | | | 0100011 | | | | | | |
| SB-type | | | | | | | Imm[12] | Imm[10:5] | | Rs2 | | | Rs1 | Funct3 | | Imm[4:1] | | | Imm[11] | | | Opcode | |
| Количество бит | | | | | | | 1 | 6 | | 5 | | | 5 | 3 | | 4 | | | 1 | | | 7 | |
| SB | | BEQ | | | Ветвление, если равны | |  |  | |  | | |  | 000 | |  | | |  | | | 0010011 | |
| SB | | BNE | | | Ветвление, если не равны | |  |  | |  | | |  | 001 | |  | | |  | | | 0010011 | |
| SB | | BLT | | | Ветвлении если rs1 < rs2(sign is impotent) | |  |  | |  | | |  | 010 | |  | | |  | | | 0010011 | |
| SB | | BLTU | | | if rs1 < rs2(sign is not impotent) | |  |  | |  | | |  | 011 | |  | | |  | | | 0010011 | |
| SB | | BGE | | | Brch is taken if rs1 > rs2(sign is impotent) | |  |  | |  | | |  | 100 | |  | | |  | | | 0010011 | |
| SB | | BGEU | | | Brch is taken if rs1 > rs2(sign is not impotent) | |  |  | |  | | |  | 101 | |  | | |  | | | 0010011 | |
| UJ- type | | | | | | | Imm[20] | Imm[10:1] | | Imm[11] | | | Imm[19:12] | Rd | | Opcode | | | | | | | |
| Количество бит | | | | | | | 1 | 10 | | 1 | | | 8 | 5 | | 7 | | | | | | | |
| UJ | | JAL | | | Pc =se imm; X1 = pc +4 | |  |  | |  | | |  |  | | 0001011 | | | | | | | |
| I type | | | | | | | Imm[11:0] | | | | Rs1(base) | | | | Funct3 | | Rd | | | opcode | | | | |
| Количество бит | | | | | | | 12 | | | | 5 | | | | 3 | | 5 | | | 7 | | | | |
| I | | | JALR | | | Pc = se imm + rs1 (LSB = 0)  Rd = rd + 4 |  | | | |  | | | | 000 | |  | | | 0000111 | | | | |
| LOAD | | | | | | | Imm[11:0] | | | | Rs1(base) | | | | Funct3 | | Rd | | | Opcode | | | | |
| Количество бит | | | | | | | 12 | | | | 5 | | | | 3 | | 5 | | | 7 | | | | |
| I | | LW | | | Загрузить из памяти в регистр 32 бита | |  | | | |  | | | | 001 | |  | | | 0000111 | | | | |
| I | | LH | | | Rd = se mem[15:0] | |  | | | |  | | | | 010 | |  | | | 0000111 | | | | |
| I | | LHU | | | Rd = ze mem[15:0] | |  | | | |  | | | | 011 | |  | | | 0000111 | | | | |
| I | | LB | | | Rd = se mem[7:0] | |  | | | |  | | | | 100 | |  | | | 0000111 | | | | |
| I | | LBU | | | Rd = ze mem[7:0] | |  | | | |  | | | | 101 | |  | | | 0000111 | | | | |
|  | |  | | |  | |  | | | |  | | | |  | |  | | |  | | | | |
|  | |  | | |  | |  | | | |  | | | |  | |  | | |  | | | | |
| STORE | | | | | | | Imm[11:5] | | | | Rs1(src) | | | | Rs2(base) | | Funct3 | | | Imm[4:0] | | | opcode | |
| Количество бит | | | | | | | 7 | | | | 5 | | | | 3 | | 5 | | | 4 | | | 7 | |
| S | | SW | | | Сохраняет слово в rs1 по адресу (adr = rs2 + se imm) | |  | | | |  | | | |  | | 00001 | | |  | | | 0001111 | |
| S | | SH | | | Сохраняет нижние 16 бит регистра в память | |  | | | |  | | | |  | | 00010 | | |  | | | 0001111 | |
| S | | SB | | | Сохраняет нижние 8 бит регистра в память | |  | | | |  | | | |  | | 00011 | | |  | | | 0001111 | |

|  |  |  |
| --- | --- | --- |
| **Register Number** | **Alternative  Name** | **Description** |
| 0 | zero | the value 0 |
| 1 | $at | (**a**ssembler **t**emporary) reserved by the assembler |
| 2-3 | $v0 - $v1 | (**v**alues) from expression evaluation and function results |
| 4-7 | $a0 - $a3 | (**a**rguments) First four parameters for subroutine. Not preserved across procedure calls |
| 8-15 | $t0 - $t7 | (**t**emporaries) Caller saved if needed. Subroutines can use w/out saving. Not preserved across procedure calls |
| 16-23 | $s0 - $s7 | (**s**aved values) - Callee saved.  A subroutine using one of these must save original and restore it before exiting. Preserved across procedure calls |
| 24-25 | $t8 - $t9 | (**t**emporaries) Caller saved if needed. Subroutines can use w/out saving. These are in addition to $t0 - $t7 above. Not preserved across procedure calls. |
| 26-27 | $k0 - $k1 | reserved for use by the interrupt/trap handler |
| 28 | $gp | **g**lobal **p**ointer.  Points to the middle of the 64K block of memory in the static data segment. |
| 29 | $sp | **s**tack **p**ointer  Points to last location on the stack. |
| 30 | $s8/$fp | **s**aved value / **f**rame **p**ointer Preserved across procedure calls |
| 31 | $ra | **r**eturn **a**ddress |