# Selen ISA

*Список изменений*

|  |  |
| --- | --- |
| 16.12.15 | Исправлены описания действий инструкций I-type(R), добавлено более подробное описание sx |
|  |  |
|  |  |

*Используемые сокращения*

* ISA (instruction set architecture) – архитектура системы команд
* rd (destination register) – регистр назначения
* rs(source register ) – регистр который служит источником операнда
* sx (sign extension ) – знаковое расширение, пример для четырёхбитных immediate:
  + ‘b1001 🡪 ‘b...1111 1001
  + ‘b0111 🡪 ‘b...0000 0111
* ux (using extension) – расширение без знака

|  |  |  |
| --- | --- | --- |
| **Номер**  **регистра** | **Альтернативное**  **название** | **Описание** |
| 0 | Zero |  |
| 1 | $at | Временный для ассемблера зарезервирован ассемблером |
| 2-3 | $v0 - $v1 | (**v**aluxs) для записи результата функции |
| 4-7 | $a0 - $a3 | (**a**rguments)первые 4 параметра вызываемой функции. Не сохраняются в процессе вызова |
| 8-15 | $t0 - $t7 | (**t**emporaries) Регистры, сохраняемые вызывающей функцией, если необходимо. Не сохраняются в процессе вызова |
| 16-23 | $s0 - $s7 | (**s**aved valuxs) – регистры сохраняемые вызываемой функцией.  Подпрограмма при вызове любого из них должна сохранить их згачение в стеке. Сохраняются в процессе вызова |
| 24-25 | $t8 - $t9 | (**t**emporaries). Регистры, сохраняемые вызывающей функцией, если необходимо. Не сохраняются в процессе вызова |
| 26-27 | $k0 - $k1 | Регистры прерываний |
| 28 | $gp | **Глобальный указатель.** Указывает на середину 64к блока памяти в статическом сегменте памяти. |
| 29 | $sp | **Стек указатель**.Указывает на первое свободное место в стеке |
| 30 | $s8/$fp | **Сохраненное значение**/**фрейм указатель**  Сохраняется в процессе вызова подпрограммы |
| 31 | $ra | **Возвращаемый адрес** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 usx | | | 0000000 |  | | | |  | | | 000 | | |  | | | 0110011 | | | |
| R | SLT | | | If rs1 < rs2 rd =1 | | | 0000000 |  | | | |  | | | 010 | | |  | | | 0110011 | | | |
| R | SLTU | | | If rs1<rs2 rd=1(only then rd2 !=0) | | | 0000000 |  | | | |  | | | 011 | | |  | | | 0110011 | | | |
| R | AND | | | rd = rs1 & rs2 | | | 0000000 |  | | | |  | | | 111 | | |  | | | 0110011 | | | |
| R | OR | | | rd = rs1 | rs2 | | | 0000000 |  | | | |  | | | 110 | | |  | | | 0110011 | | | |
| R | XOR | | | rd = rs1 ^ rs2 | | | 0000000 |  | | | |  | | | 100 | | |  | | | 0110011 | | | |
| R | SLL | | | rd = rs1 << rs2[4:0] | | | 0000000 |  | | | |  | | | 001 | | |  | | | 0110011 | | | |
| R | SRL | | | rd = rs1 >> rs2[4:0] | | | 0000000 |  | | | |  | | | 101 | | |  | | | 0110011 | | | |
| R | SUB | | | rd =rs1 – rs2 | | | 0100000 |  | | | |  | | | 000 | | |  | | | 0110011 | | | |
| R | SRA | | | rd = rs1 >>> rs2[4:0] | | | 0100000 |  | | | |  | | | 101 | | |  | | | 0110011 | | | |
| R | AM | | | Rd = (rs1+rs2)/2 | | | 0100000 |  | | | |  | | | 010 | | |  | | | 0110011 | | | |
| I-type(R) | | | | | | | Imm | | | | | Rs1 | | | Funct3 | | | Rd | | | Opcode | | | |
| Количество бит | | | | | | | 12 | | | | | 5 | | | 3 | | | 5 | | | 7 | | | |
| I | | ADDI | | | Rd = sx(Imm) + Rs1 | |  | | | | |  | | | 000 | | |  | | | 0010011 | | | |
| I | | SLTI | | | If sx(Imm) > rs1 then rd = 1  else rd = 0 | |  | | | | |  | | | 010 | | |  | | | 0010011 | | | |
| I | | SLTIU | | | If sx(Imm) > rs1 then rd = 1  else rd = 0 | |  | | | | |  | | | 011 | | |  | | | 0010011 | | | |
| I | | ANDI | | | Rd = rs1 & sx(Imm) | |  | | | | |  | | | 111 | | |  | | | 0010011 | | | |
| I | | ORI | | | Rd =rs1 | sx(Imm) | |  | | | | |  | | | 110 | | |  | | | 0010011 | | | |
| I | | XORI | | | Rd = rs ^ sx(Imm) | |  | | | | |  | | | 100 | | |  | | | 0010011 | | | |
| I-type(R) | | | | | | | Imm | | Shamt | | | Rs1 | | | Funct3 | | | Rd | | | Opcode | | | |
| Количество бит | | | | | | | 7 | | 5 | | | 5 | | | 3 | | | 5 | | | 7 | | | |
| I | | SLLI | | | Rd = rs1<< shamt | | 0000000 | |  | | |  | | | 001 | | |  | | | 0010011 | | | |
| I | | SRLI | | | Rd= rs1>>shamt | | 0000000 | |  | | |  | | | 101 | | |  | | | 0010011 | | | |
| I | | SRAI | | | Rd=rs1>>>shamt | | 0100000 | |  | | |  | | | 101 | | |  | | | 0010011 | | | |
| U-type | | | | | | | Imm | | | | | | | | Rd | | | Opcode | | | | | | |
| Количество бит | | | | | | | 20 | | | | | | | | 5 | | | 7 | | | | | | |
| U | | LUI | | | Rd[31:12] = imm  Rd[11:0] = 0 | |  | | | | | | | |  | | | 0110111 | | | | | | |
| U | | AUIPC | | | Rd= pc+ A[31:0];  A[31:12]=imm;  A[11:0]=0 | |  | | | | | | | |  | | | 0010111 | | | | | | |
| 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 | |  | | |  | | | 1100011 | |
| SB | | BNE | | | Ветвление, если не равны | |  |  | |  | | |  | 001 | |  | | |  | | | 1100011 | |
| SB | | BLT | | | Ветвлении если rs1 < rs2(sign is impotent) | |  |  | |  | | |  | 100 | |  | | |  | | | 1100011 | |
| SB | | BLTU | | | if rs1 < rs2(sign is not impotent) | |  |  | |  | | |  | 110 | |  | | |  | | | 1100011 | |
| SB | | BGE | | | Brch is taken if rs1 > rs2(sign is impotent) | |  |  | |  | | |  | 101 | |  | | |  | | | 1100011 | |
| SB | | BGEU | | | Brch is taken if rs1 > rs2(sign is not impotent) | |  |  | |  | | |  | 111 | |  | | |  | | | 1100011 | |
| 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 =sx imm + pc; X1 = pc +4  (imm is sign offset in mull of 2 bytes) | |  |  | |  | | |  |  | | 1101111 | | | | | | | |
| I type | | | | | | | Imm[11:0] | | | | Rs1(base) | | | | Funct3 | | Rd | | | opcode | | | | |
| Количество бит | | | | | | | 12 | | | | 5 | | | | 3 | | 5 | | | 7 | | | | |
| I | | | JALR | | | Pc = sx imm + rs1 (LSB = 0)  Rd = pc + 4 |  | | | |  | | | | 000 | |  | | | 1100111 | | | | |
| 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 | |  | | | |  | | | | 010 | |  | | | 0000011 | | | | |
| I | | LH | | | Rd = sx mem[15:0] | |  | | | |  | | | | 001 | |  | | | 0000011 | | | | |
| I | | LHU | | | Rd = ze mem[15:0] | |  | | | |  | | | | 101 | |  | | | 0000011 | | | | |
| I | | LB | | | Rd = sx mem[7:0] | |  | | | |  | | | | 000 | |  | | | 0000011 | | | | |
| I | | LBU | | | Rd = ue mem[7:0] | |  | | | |  | | | | 100 | |  | | | 0000010031 | | | | |
|  | |  | | |  | |  | | | |  | | | |  | |  | | |  | | | | |
|  | |  | | |  | |  | | | |  | | | |  | |  | | |  | | | | |
| STORE | | | | | | | Imm[11:5] | | | | Rs1(basx) | | | | Rs2(src) | | Funct3 | | | Imm[4:0] | | | opcode | |
| Количество бит | | | | | | | 7 | | | | 5 | | | | 5 | | 3 | | | 5 | | | 7 | |
| S | | SW | | | Сохраняет слово в rs1 по адресу (adr = rs2 + sx imm) | |  | | | |  | | | |  | | 010 | | |  | | | 0100011 | |
| S | | SH | | | Сохраняет нижние 16 бит регистра в память | |  | | | |  | | | |  | | 001 | | |  | | | 0100011 | |
| S | | SB | | | Сохраняет нижние 8 бит регистра в память | |  | | | |  | | | |  | | 000 | | |  | | | 0100011 | |