|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | By default instruction is 1 byte length | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Can be changed to two, four or eight byte format by using OP2-8 prefixes. | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Instruction page 0 (default) description | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | bytes | Size | 0 | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | |
|  | nibbles | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prefixes: |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | OP2 |  | 0xC |  | defines two byte form | |  |  |  |  |  |  |  |  |  |  |  |  |
|  | OP4 |  | 0xD |  | defines four byte form | |  |  |  |  |  |  |  |  |  |  |  |  |
|  | OP8 |  | 0xE |  | defines eight byte form | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | NOP | 1 | 0x0 | 0x0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x0 | 0x0 | xx |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0xD | 0x0 | 0x0 | xx | xx | xx | xx | xx |  |  |  |  |  |  |  |  |
|  |  | 8 | 0xE | 0x0 | 0x0 | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx |
|  | note: xx - any value | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | RET | 1 | 0x0 | 0x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x0 | 0x1 | cnt:u4 | How many bytes to add to SP, before POP IP | | | | | | | | | | | |
|  |  | 4 | 0xD | 0x0 | 0x1 | cnt:u12 | | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | IRET | Same as RET, but opcode is 0x02 | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | INC | 1 | 0x0 | 0x3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x0 | 0x3 | reg:u4 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0xD | 0x0 | 0x3 | reg:u8 | | val:u12 | | |  |  |  |  |  |  |  |  |
|  |  | 8 | 0xE | 0x0 | 0x3 | reg:u8 | | val:u44 | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | DEC | Same as INC, but opcode is 0x04 | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | NOT | 1 | 0x0 | 0x5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x0 | 0x5 | reg:u4 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0xD | 0x0 | 0x5 | x | reg:u8 | | cnt:u8 | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | RS | 1 | 0x1 | reg:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x1 | reg:u8 | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | NS | Same as RS, opcode 0x2 | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | LI | 1 | 0x3 | val:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x3 | reg:u4 | val:u4 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0xD | 0x3 | reg:u8 | | val:u16 | | | |  |  |  |  |  |  |  |  |
|  |  | 8 | 0xE | 0x3 | reg:u8 | | val:u44 | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | JMP | 1 | 0x4 | ofs |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x4 | bcs:u4 | ofs:u4 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0xD | 0x4 | bcs:u8 | | ofs:u16 | | | |  |  |  |  |  |  |  |  |
|  |  | 8 | 0xE | 0x4 | bcs:u8 | | ofs:u44 | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | CALL | Same as JMP, opcode 0x5 | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | PUSH | 1 | 0x6 | reg:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x6 | reg:u8 | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0XD | 0x6 | reg:u8 | | cnt:u8 | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | POP | 1 | 0x7 | reg:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x7 | reg:u8 | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0XD | 0x7 | reg:u8 | | cnt:u8 | | ofs:u8 | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ALU | 1 | 0x8 | op:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x8 | op:u4 | adt:u4 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0xD | 0x8 | op:u4 | adt:u4 | a1:u4 | a2:u4 | r:u8 | |  |  |  |  |  |  |  |  |
|  |  | 8 | 0xE | 0x8 | op:u8 | | adt:u8 | | a1:u8 | | a2:u8 | | r::u8 | | ofs:u16 | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Note: basic ALU operations have scalar mode (VL == 0) or vector mode (VL > 0). In vector mode ALU performs VL operations on its arguments. Argument will be read from register, if its stride is 0, or from memory, next argument value will be fetched from next register or memory, incrementing pointer by its stride. Same for destination. | | | | | | | | | | | | | | | |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | INT | 1 | 0x9 | val: u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0x9 | val:u8 | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | IN | 1 | 0xA | ch:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0xA | ch:u8 | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0XD | 0xA | ch:u8 | | reg:u8 | | adt:u4 | fmt:u4 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | OUT | 1 | 0xB | ch:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0xB | ch:u8 | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 4 | 0XD | 0xB | ch:u8 | | reg:u8 | | adt:u4 | fmt:u4 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | EXT | 1 | 0xF | ipg:u4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 | 0xC | 0xF | ipg:u8 | |  |  |  |  |  |  |  |  |  |  |  |  |