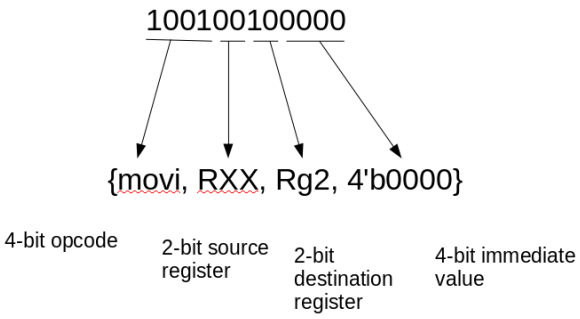
Instruction format



Defined values

```
parameter // register names
   Rg0 = 2'b00, Rg1 = 2'b01, Rg2 = 2'b10,
   Rg3 = 2'b11, RXX = 2'b00;
parameter  // opcodes
  jmp  = 4'b0000, jz   = 4'b0010,
  jnz = 4'b0011,
  movi = 4'b1001, mov  = 4'b1000,
  addi = 4'b1011, add  = 4'b1010,
  subi = 4'b1101, sub  = 4'b1100,
  cmpi = 4'b1111, cmp  = 4'b1110;
```

Instruction set

```
{jmp,  RXX, RXX, imm};
      jump to the instruction at address pc+imm

{jz,  RXX, RXX, imm};
      jump to the instruction at address pc+imm if the zero flag is set

{jnz,  RXX, RXX, imm};
      jump to the instruction at address pc+imm if the zero flag is not set

{movi,  RXX, dst, imm};
      store the value imm into the register dst

{mov,  src, dst, 4'b0000};
      store the value in register src into the register dst

{addi,  RXX, dst, imm};
      add the value imm to the value in register dst, store the result in dst

{add,  src, dst, 4'b0000};
      add the value in register src to the value in register dst, store the result
      in dst

{subi,  RXX, dst, imm};
      subtract the value imm from the register dst, store the result in dst

{sub,  src, dst, 4'b0000};
      subtract the value in register src from the value in register dst

{cmpi,  RXX, dst, imm};
      compare the value imm to the value in register dst; set the zero flag  to 1
      if they are equal, to 0 otherwise

{cmp,  src, dst, 4'b0000};
      compare the value in register src to the value in register dst; set the zero
      flag  to 1 if they are equal, to 0 otherwise
```

All arithmetic instructions (add, addi, sub, subi, cmp, cmpi) set the zero flag on the basis of the ALU output: if the result is zero, the zero flag is set to one; otherwise, it is set to zero.