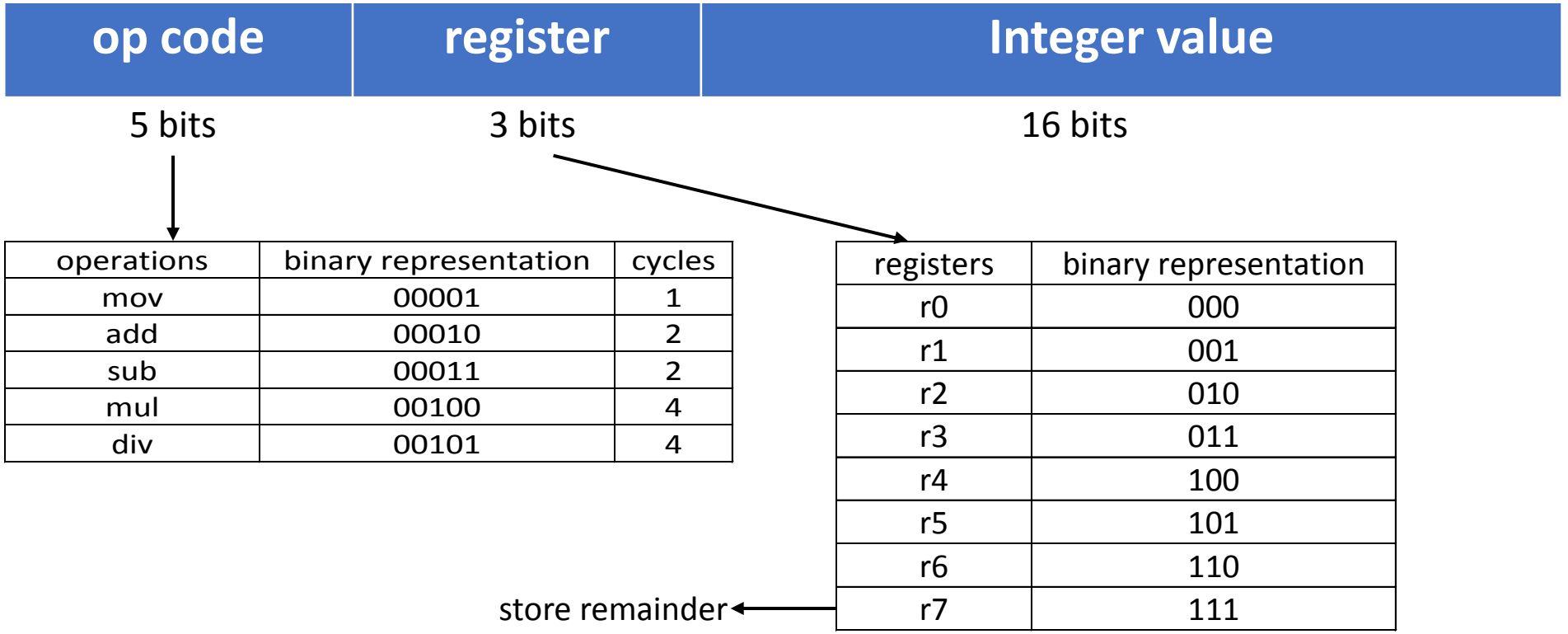
ISA

Sujan Gurung, ID: 5918127

24-bit instruction



operations	binary representation	cycles
mov	00001	1
add	00010	2
sub	00011	2
mul	00100	4
div	00101	4

registers	binary representation
r0	000
r1	001
r2	010
r3	011
r4	100
r5	101
r6	110
r7	111

example :-
decoded form: mov r0 15
encoded form: 00001 000 0000000000001111

Operations

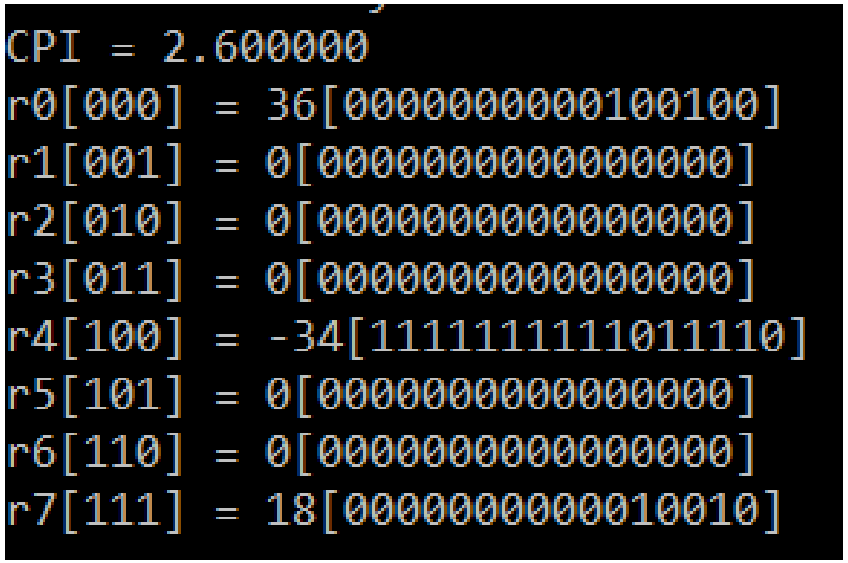
- instruction: opcode op1 op2
- op1 always a register value
- op2 can be register or integer value
- mov: op2 value is stored in op1 register
- add: op1 and op2 added
- sub: op2 subtracted from op1
- mul: op1 multiplied with op2
- div: op1 divided by op2, remainder stored in register 7

Program design

- written in python
- use a list to store register values
- variables: cycles, instruction count
- functions: mov, add, sub, mul, div, register finder, decimal/register/opcode to binary

```
160
161     instruction = input()
162     op, op1, op2 = instruction.split(' ')
163
164     while instruction != 'end 0 0':
165         result = []
166         if op == 'mov':
167             result = mov(op1, op2)
168         elif op == 'add':
169             result = add(op1, op2)
170         elif op == 'sub':
171             result = sub(op1, op2)
172         elif op == 'mul':
173             result = mul(op1, op2)
174         elif op == 'div':
175             result = div(op1, op2)
176
177         instruction_counter += 1
178         cycles += result[1]
```

```
55  def mov(a, b):
56      register1 = register_finder(a)
57      register2 = register_finder(b)
58      if register2 == -1:
59          register_list[register1] = int(b)
60      else:
61          value = register_list[register2]
62          register_list[register1] = value
63
64      return ['00001', 1]
```

Test scenario

- mov r0 18 [00001 000 0000000000010010], r0 = 18, cycles = 1
- add r4 90 [00010 100 0000000001011010], r4 = 0 + 90 = 90, cycles = 1 + 2 = 3
- mul r0 2 [00100 000 0000000000000010], r0 = 18*2 = 36[0000000000100100], cycles = 3 + 4 = 7
- div r4 r0 [00101 100 0000000000100100], r4 = 90/36 = 2, r7 = 18, cycles = 7 + 4 = 11
- sub r4 r0 [00011 100 0000000000100100], r4 = 2 – 36 = -34[1111111111011110], cycles = 11 + 2 = 13
- r0 = 36 [0000000000100100]
- r4 = -34 [1111111111011110]
- r7 = 18 [0000000000010010]
- CPI = 13/5 = 2.6