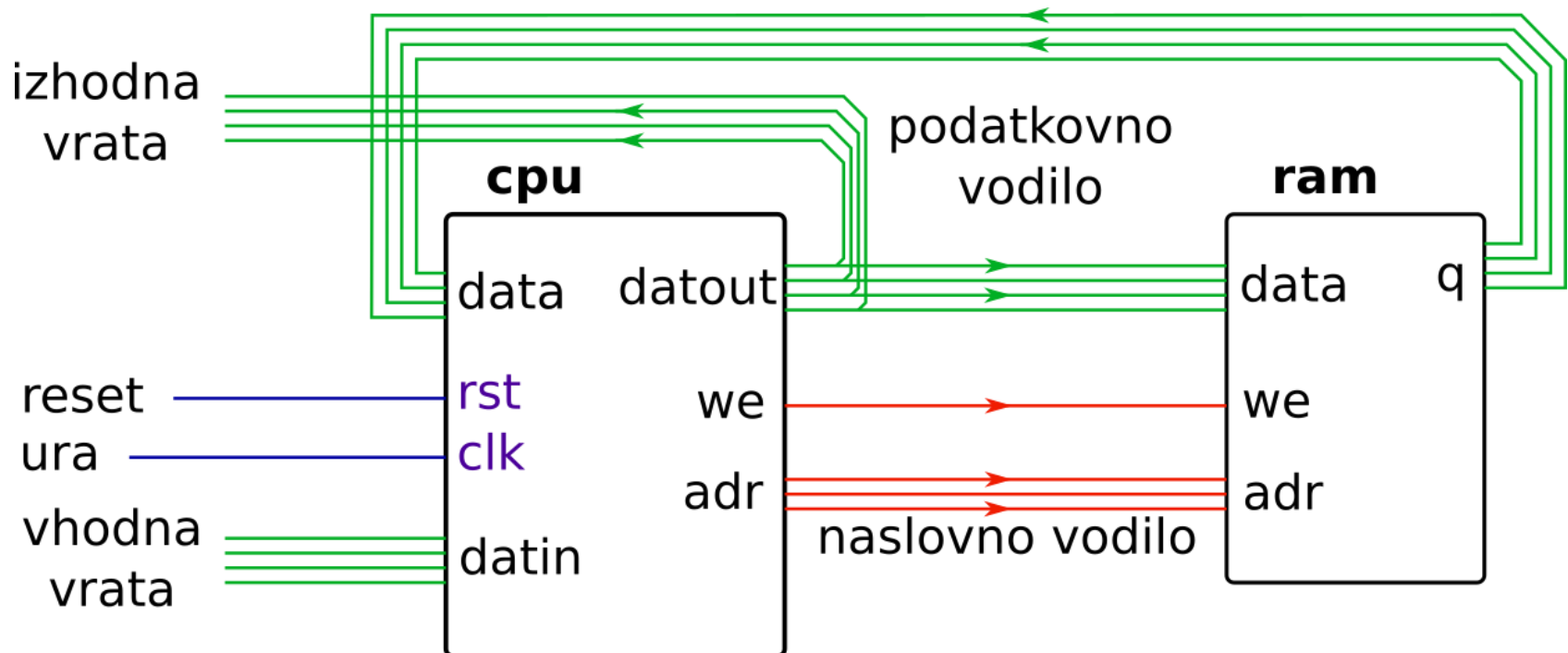


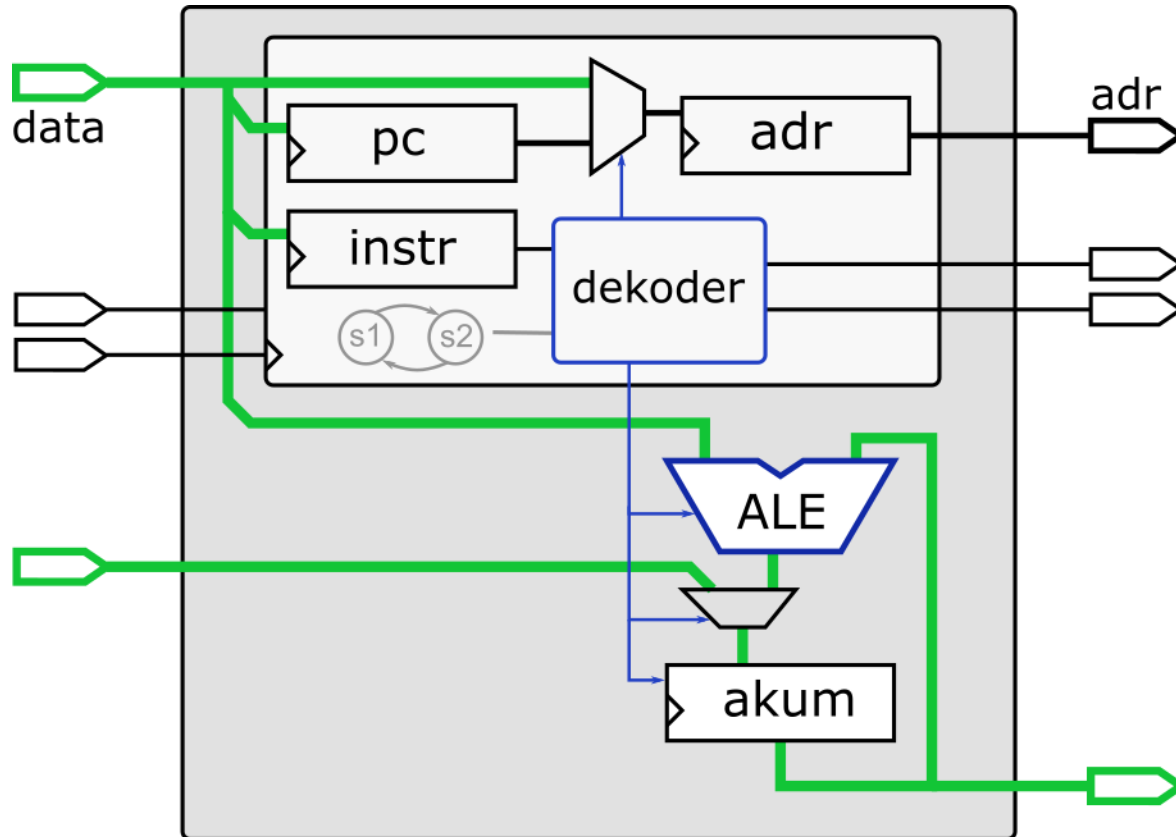
Mikroprocesor

12-bitni procesor z majhnim naborom ukazov (RISC)
in 256 besedami pomnilnika RAM



Logična zgradba CPU

- krmilna enota (pc, avtomat, ukazni register in dekodeer)
- aritmetično logična enota
- delovni register (akumulator)



Delovanje procesorja

Arhitektura load-store (procesorji RISC)

Nabor ukazov:

- **LOAD** prenesi iz RAM
- **STORE** shrani v RAM
- **ADD, ...** operacije ALE, ki uporabljajo registre

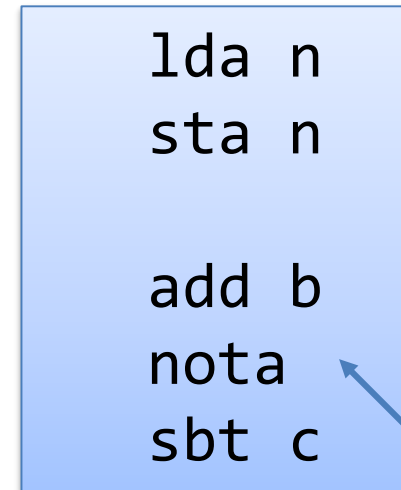
Registri:

- izvor in ponor podatkov ALE
- npr. $r1 = r1 \text{ or } r2$;

OR $r1, r2$

```
lda n
sta n

add b
nota
sbt c
```



En register (akumulator)

npr. $a = a - c$

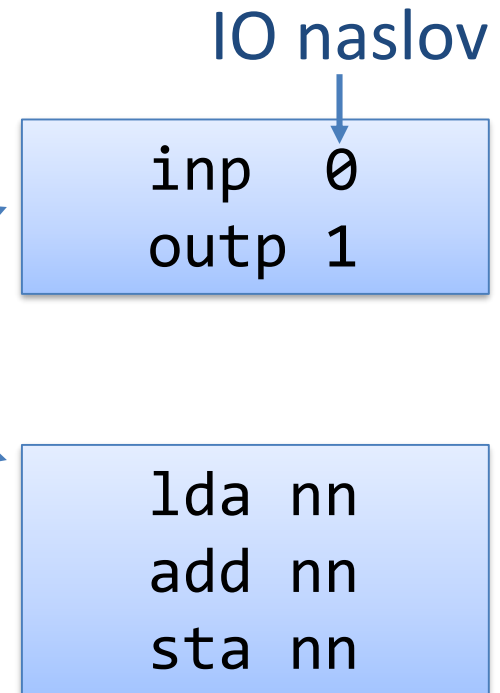
sbt c

Delovanje procesorja

Naslavljanje (addressing)

Naslov določa, kje je shranjena spremenljivka

- v registru: **INC R16**
- direktno v IO: **OUT PORTA, R16**
- direktno v RAM: **LDS R17, \$006a**
- indirektno v RAM: **LD R4, X**
- takoj v ukazu: **LDI R16, 255**



Kako CPU izvaja program ?

Primer: seštej števila od 1 do 10 in izpiši vsoto

```
s = 0
```

```
for i = 1 to 10
```

```
    s = s + i
```

```
print s
```



```
    s = 0; i = 1  
loop: if i=10 goto next  
    s = s + i  
    i = i + 1  
    goto loop  
next:  
    print s
```

Program v zbirniku

$i-10 = 0$ (jze)

~~$s = 0; i = 1$~~

loop: if i=10 goto next

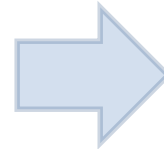
s = s + i

i = i + 1

goto loop

next:

print s



```
loop: lda i
      sbt 10
      jze next
      lda s
      add i
      sta s
      lda i
      add 1
      sta i
      jmp loop
next: lda s
      outp 0
```