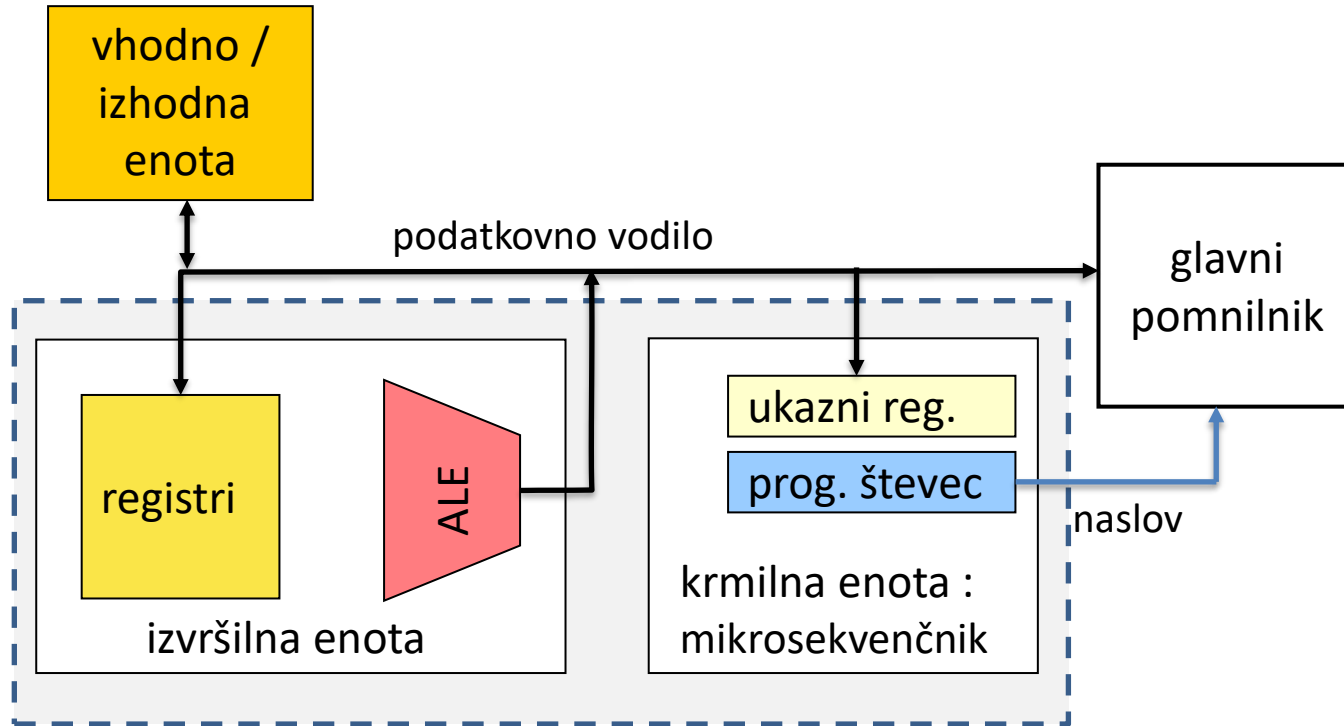


# Mikroprocesor

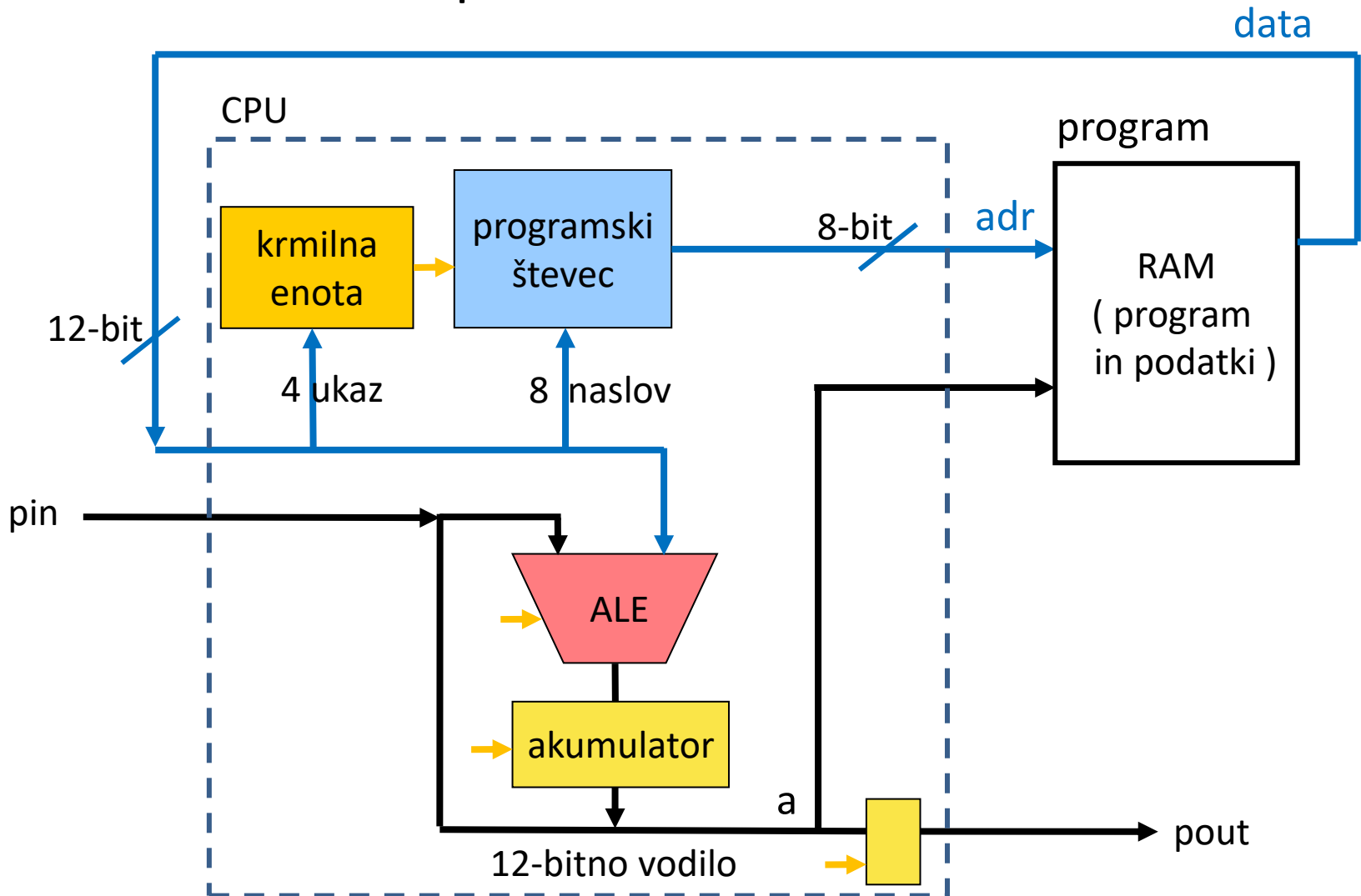
Von Neumannov model: centralna procesna enota (CPU), glavni pomnilnik in vhodno/izhodna enota



- delovanje CPU določa nabor ukazov
  - ▶ kompleksen nabor ukazov (CISC) za zmogljive računalnike ali
  - reduciran nabor ukazov (RISC) za vgrajene sisteme

# Učni mikroprocesor

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



# Delovanje učnega procesorja

## Arhitektura load-store (procesorji RISC)

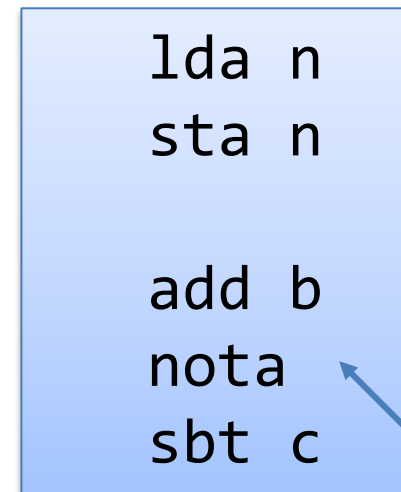
Nabor ukazov:

- **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.  $a = a - c$

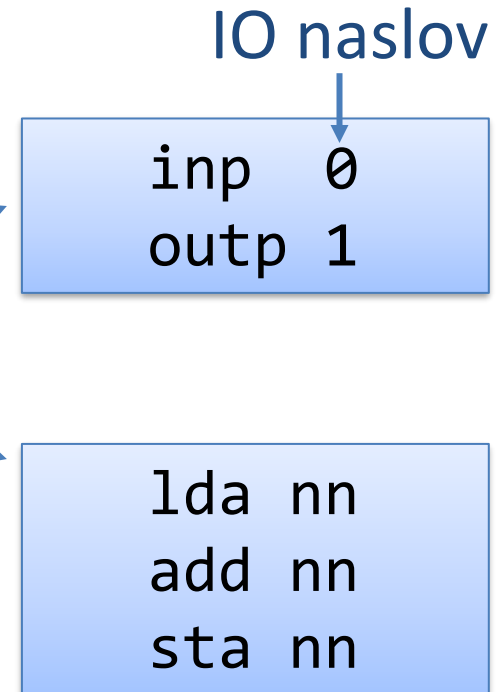
sbt c

# Vrste procesorskih ukazov

## Operacije in naslavljanje (addressing)

Naslov določa, kje je shranjen podatek

- 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**

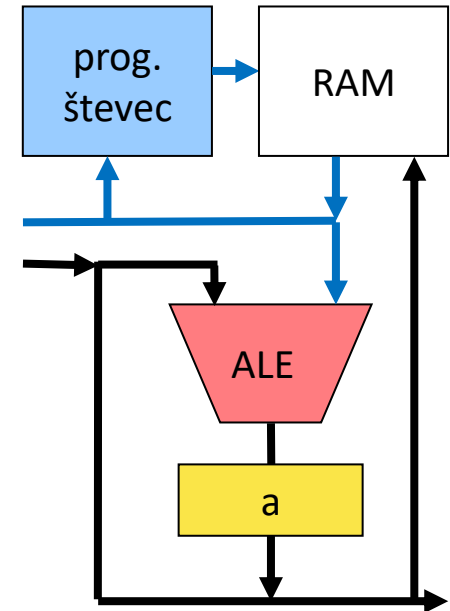


# Ukazi učnega procesorja

## Strojna koda

```
lda koda := "0001"; -- naloži iz RAM
sta koda := "0010"; -- shrani v RAM
add koda := "0100"; -- prištej
sub koda := "0101"; -- odštej
anda koda := "0110"; -- logična and
ora koda := "0111"; -- logična or
jmp koda := "1000"; -- skok na nov naslov
jze koda := "1001"; -- skok, če je a=0

inp koda := "0011"; -- naloži iz vhoda
outp koda := "0111"; -- shrani na izhod
```



Zbirnik (assembler) za pisanje ukazov v razumljivi obliki

```
      ukaz  operand
      ↓     ↓
oznaka → start: lda 30 ; naloži konstanto v akumulator
      ↓     ↓
      komentar
```

# Zbirnik in simulator CPU

oznaka vrstice

ukaz (mnemonik)

oz:      lda a ← **simbolična oznaka**  
          add a      (spremenljivka)  
          jmp oz

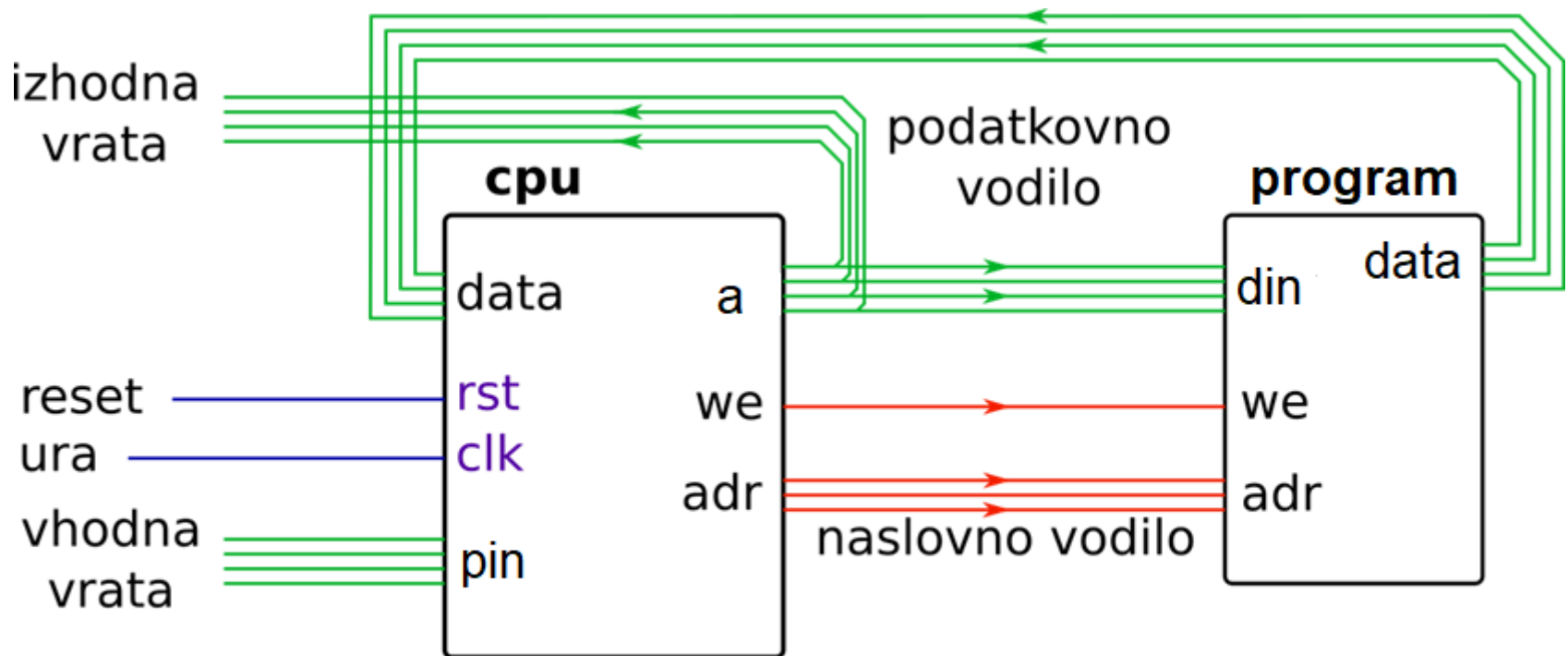
**a      db    5** ← **direktiva**  
                          (določi spremenljivko)

Prevedena zbirniška koda:

```
0=> lda & x"03",  
1=> add & x"03",  
2=> jmp & x"01",  
3=> x"005"
```

# Model mikroprocesorja

Veže vsebuje CPU logiko in vključuje programski pomnilnik z vpisanim programom



# 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

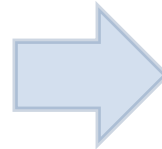
s = s + i

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
```