Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*

Digitalni Elektronski Sistemi

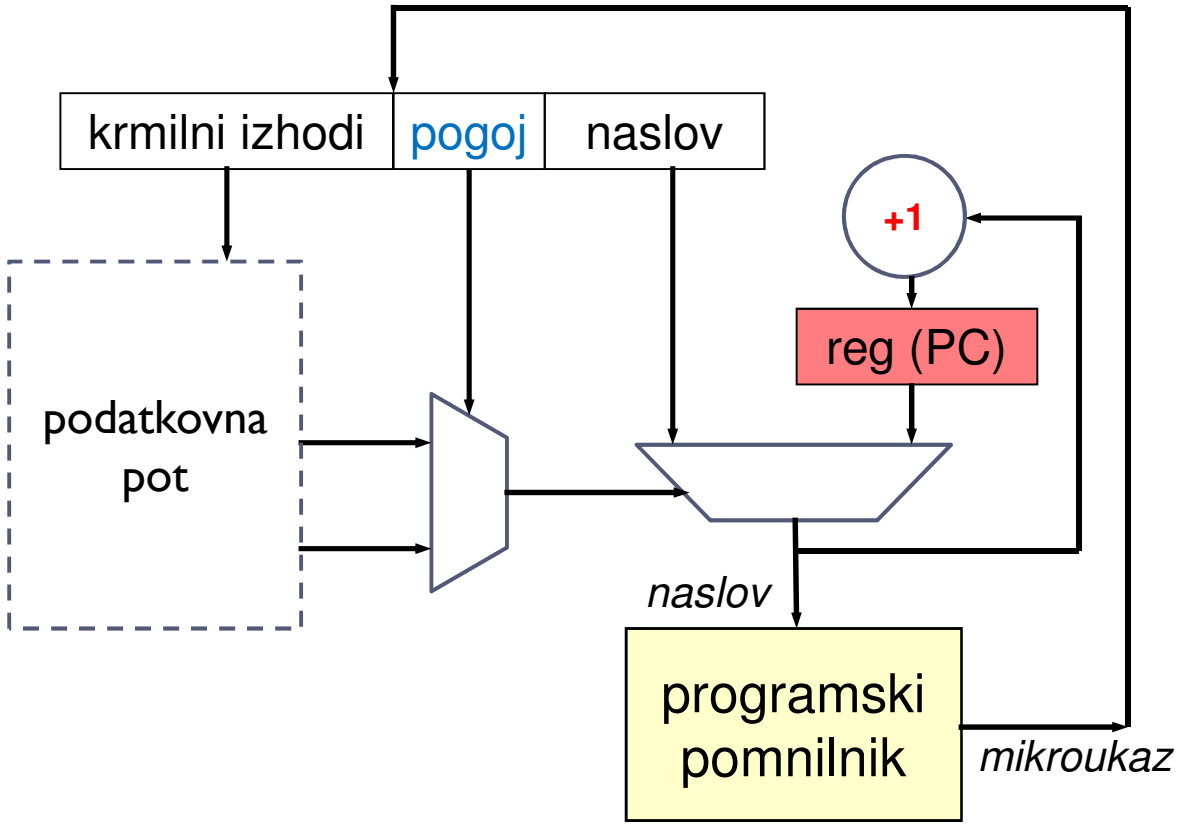
Procesorji

Mikro-programirana procesna enota, primeri

Mikro-programiran krmilnik s števcem

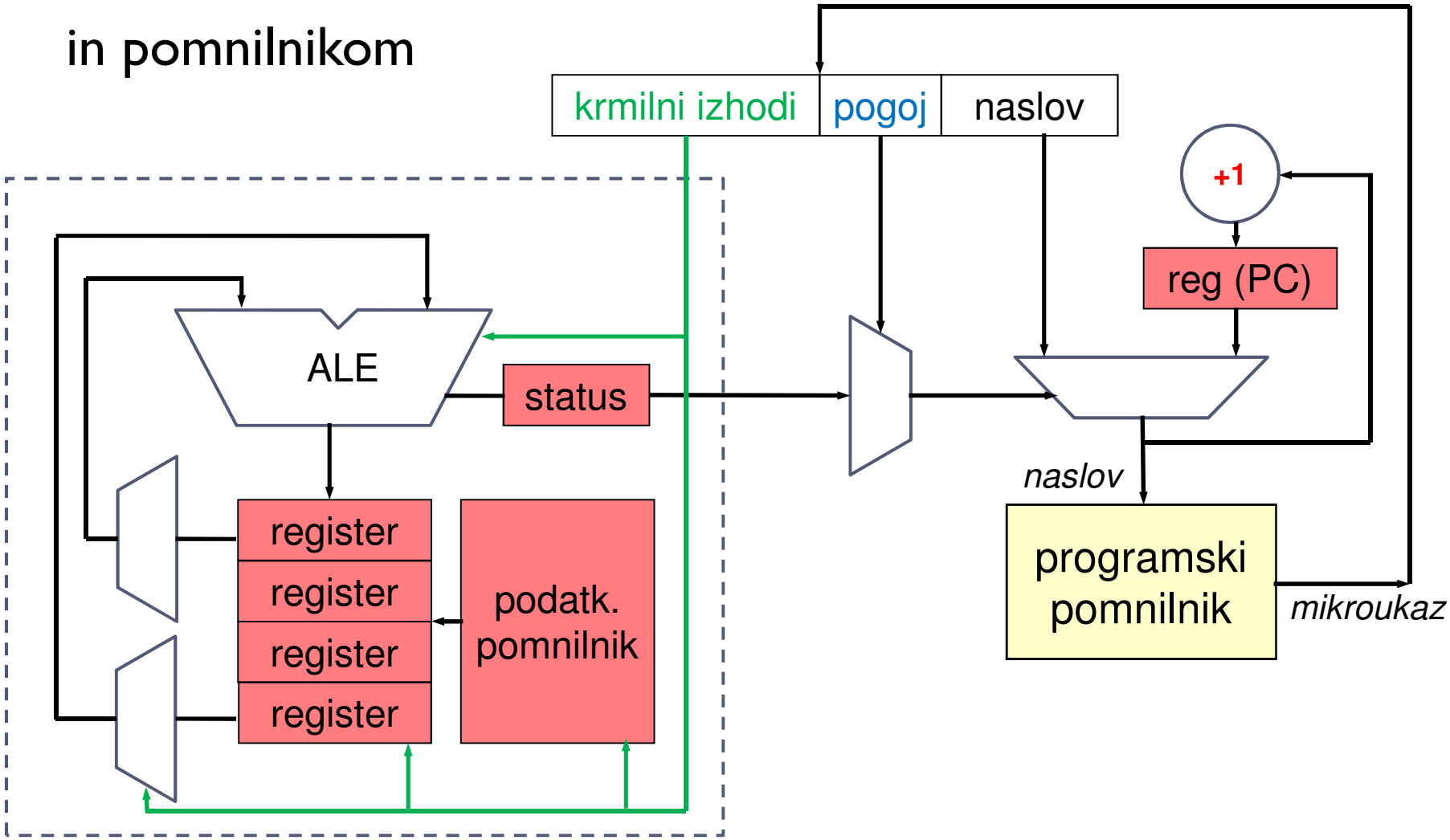
Mikroukazi

- mikroprogram se izvaja kot zaporedje mikroukazov
- mikroukazi določajo krmilne izhode
- *skočni ukazi* ob pogoju izvršijo skok na nov naslov



Mikro-programirana krmilna (procesna) enota

- izvršilna enota z ALE in pomnilnikom



Mikrooperacije

- Aritmetično-logična enota (ALE) izvaja računske mikrooperacije, npr.:

$$
\begin{array}{ll}
r3 <= r1 + ci & \mid p1 = 0,\ p0 = 0,\ ci = 1 \text{ (povečaj)} \\
r3 <= r1 + r2 + ci & \mid p1 = 0,\ p0 = 1,\ ci = 1 \text{ (seštej s prenosom)} \\
r3 <= r1 + (\text{NOT } r2) + ci & \mid p1 = 1,\ p0 = 0,\ ci = 1 \text{ (odštej)} \\
r3 <= r1 - 1 + ci & \mid p1 = 1,\ p0 = 1,\ ci = 1 \text{ (prenesi } r1)
\end{array}
$$

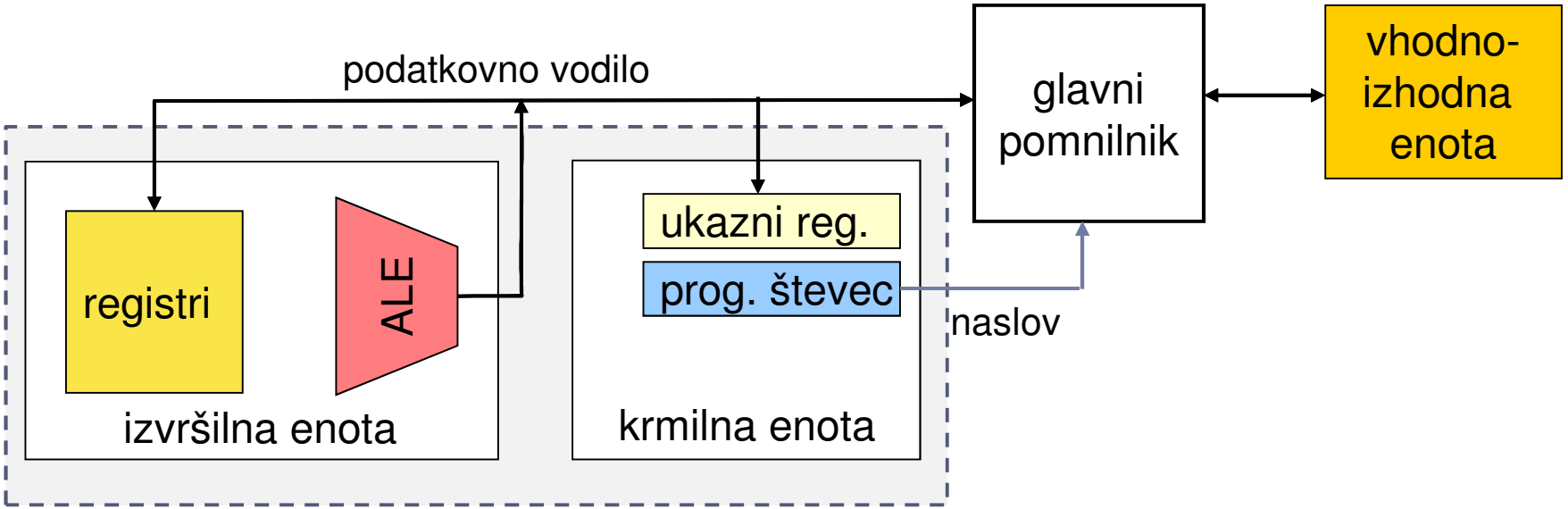
- Krmilni izhodi izbirajo operacijo, operande in izhodni register
  - operand je v registru ali v pomnilniku
- Skočne operacije spremenijo zaporedje izvajanja ukazov
  - brezpogojni skok, ali pogojni skok, glede na rezultat zadnje operacije – status: prenos, negativni rezultat, rezultat 0…)

Kode in koraki izvajanja mikrooperacij

- za direktno krmiljenje enot z mikroukazi potrebujemo veliko bitov
  - npr. 4 bite za ALE, 12 za registre, 16 za pomnilnik….
- krmilni signali so kodirani
  - potrebujemo ukazni dekoder
- mikroukazi se izvršijo v več urnih ciklih
  - bolj optimalna kritična pot in višja frekvenca ure
  - nekaterih operacij ne moremo narediti v enem ciklu
    - npr. branje in pisanje v pomnilnik

Von Neumannov model računalnika

- Centralno procesna enota (CPE) in pomnilnik
  - vhodno – izhodna enota skrbi za komunikacijo z zunanjostjo



- delovanje CPE določa nabor ukazov
  - ukazi so prilagojeni programskim jezikom (C/C++)

Osnovni gradniki mikroprocesorja

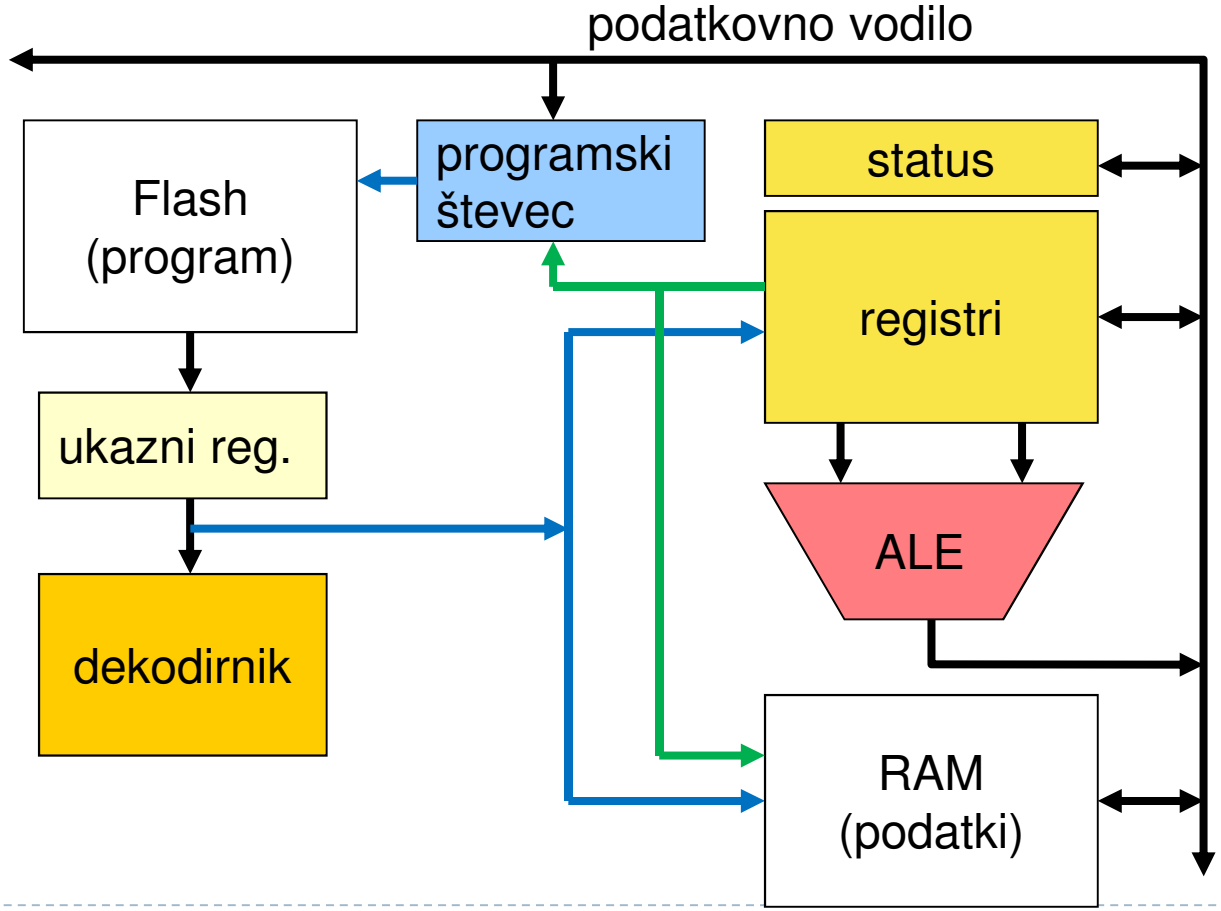
- Mikroprocesor na integriranem vezju vsebuje izvršilno in krmilno enoto
- Mikroprocesor potrebuje za delovanje:
  - zunanjo uro in reset
  - zunanji pomnilnik s programskimi ukazi in podatki
  - vhodno in izhodno (periferno) enoto za komunikacijo z okolico
    - periferna enota je lahko del pomnilnika (na določenih naslovih)
    - ali pa poteka komunikacija preko posebnih V/I ukazov
- V praksi potrebujemo vsaj dve vrsti pomnilnika
  - za program takšnega, ki ohranja vsebino (ROM, Flash)
  - za delovne podatke pa pomnilnik s hitrim branjem in pisanjem (RAM – Random Access Memory)

Mikroprocesorji za vgrajene naprave

- Mikrokrmilniki so sistemi na integriranem vezju, ki vsebujejo
  - mikroprocesorsko jedro (izvršilno in krmilno enoto),
  - programski in podatkovni pomnilnik,
  - ter različne V/I vmesnike:
    - vzporedna vrata (Port)
    - zaporedne komunikacijske vmesnike: I2C, SPI, UART
    - analogno / digitalne (A / D) in D / A pretvornike
    - modulatorje (PWM), časovnike, števce…
    - komunikacijske krmilnike: Ethernet MAC, USB
- Primer mikrokrmilnikov
  - Atmel AVR, 8-bitni procesor na razvojnem sistemu Arduino
  - ARM-7, 32-bitni procesor na razvojnem sistemu Š-ARM

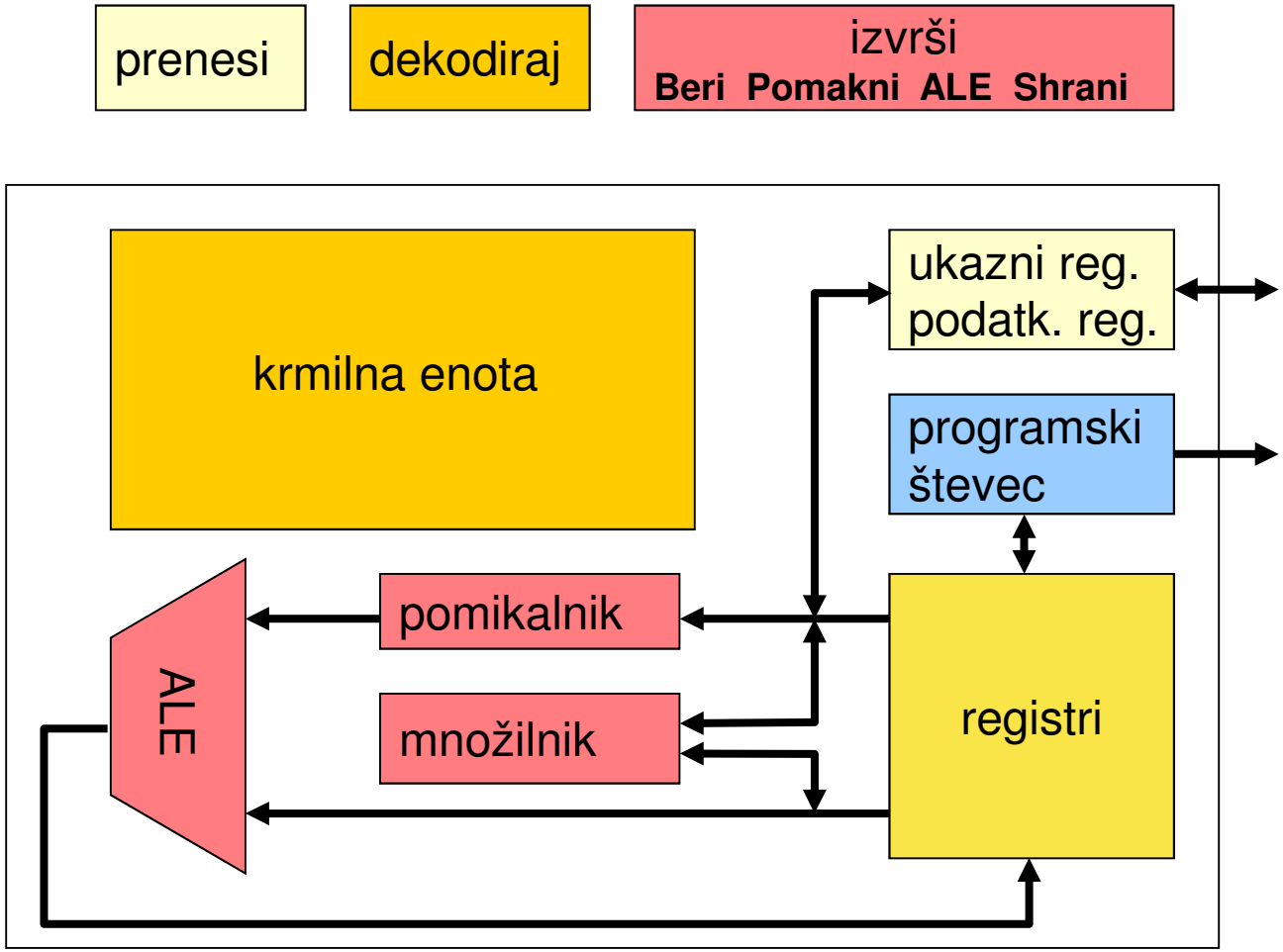
Procesorsko jedro AVR

- Ukaze izvaja v dveh korakih (urnih ciklih)
  1. prenesi ukaz iz pomnilnika v ukazni register (fetch)
  2. dekodiraj, izvrši ukaz (ALE) in shrani rezultat (execute)

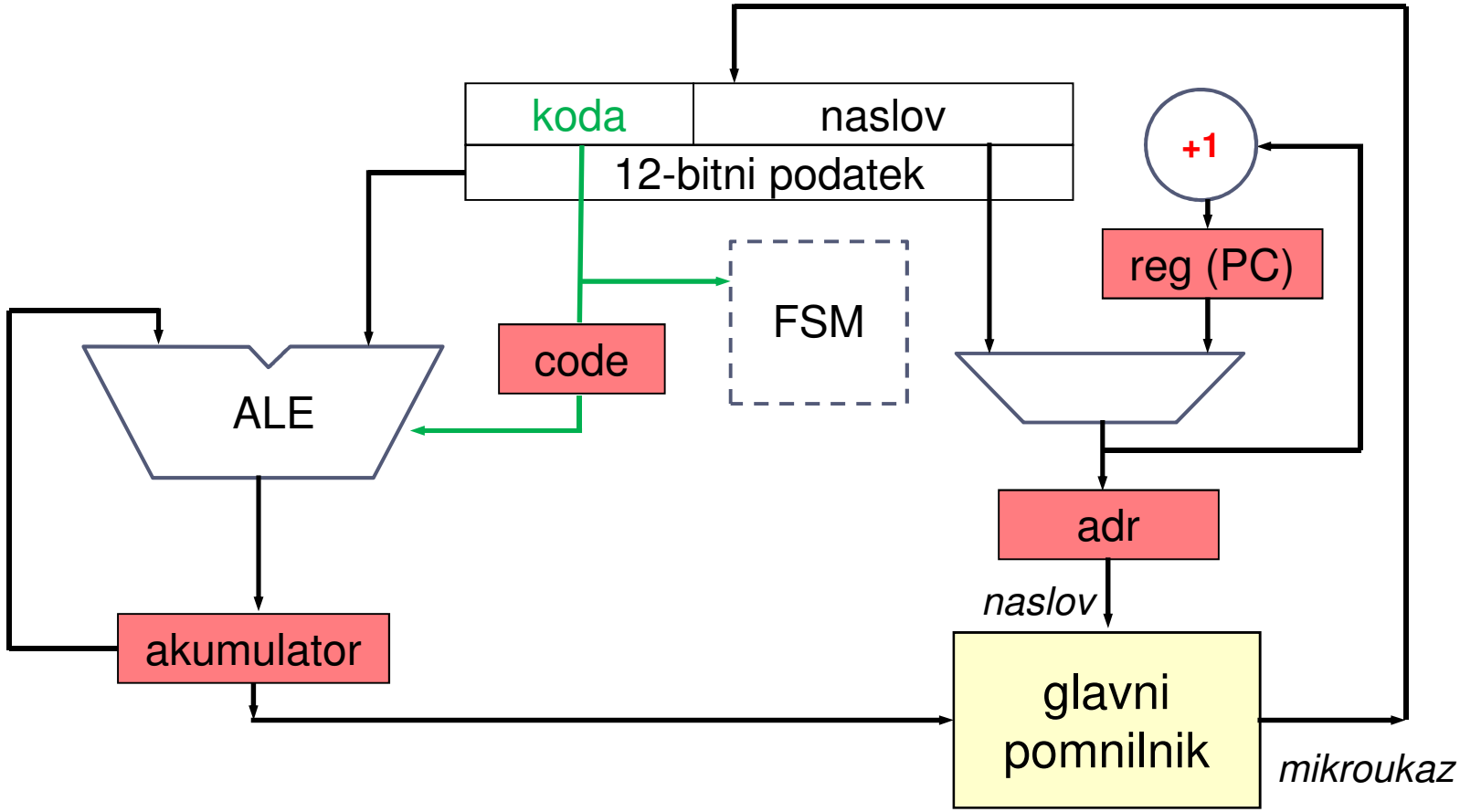


Procesorsko jedro ARM-7

- Ukaze izvršuje v treh ciklih:



Učni procesor: CPU-12

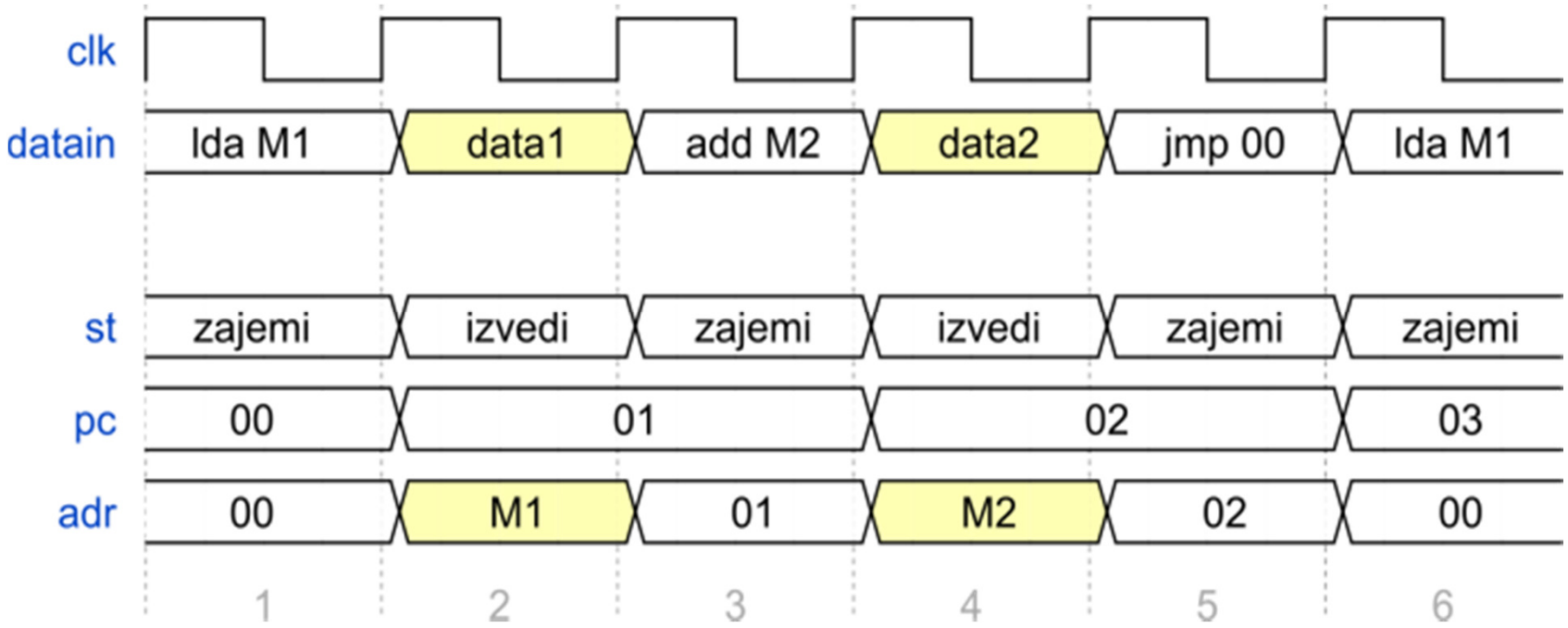


- ukazi z akumulatorjem, en operand je v pomnilniku
  1. prenesi 4-bitno ukazno kodo in naslov operanda iz pomnilnika
  2. prenesi operand (12 bitni podatek) in izvedi ukaz

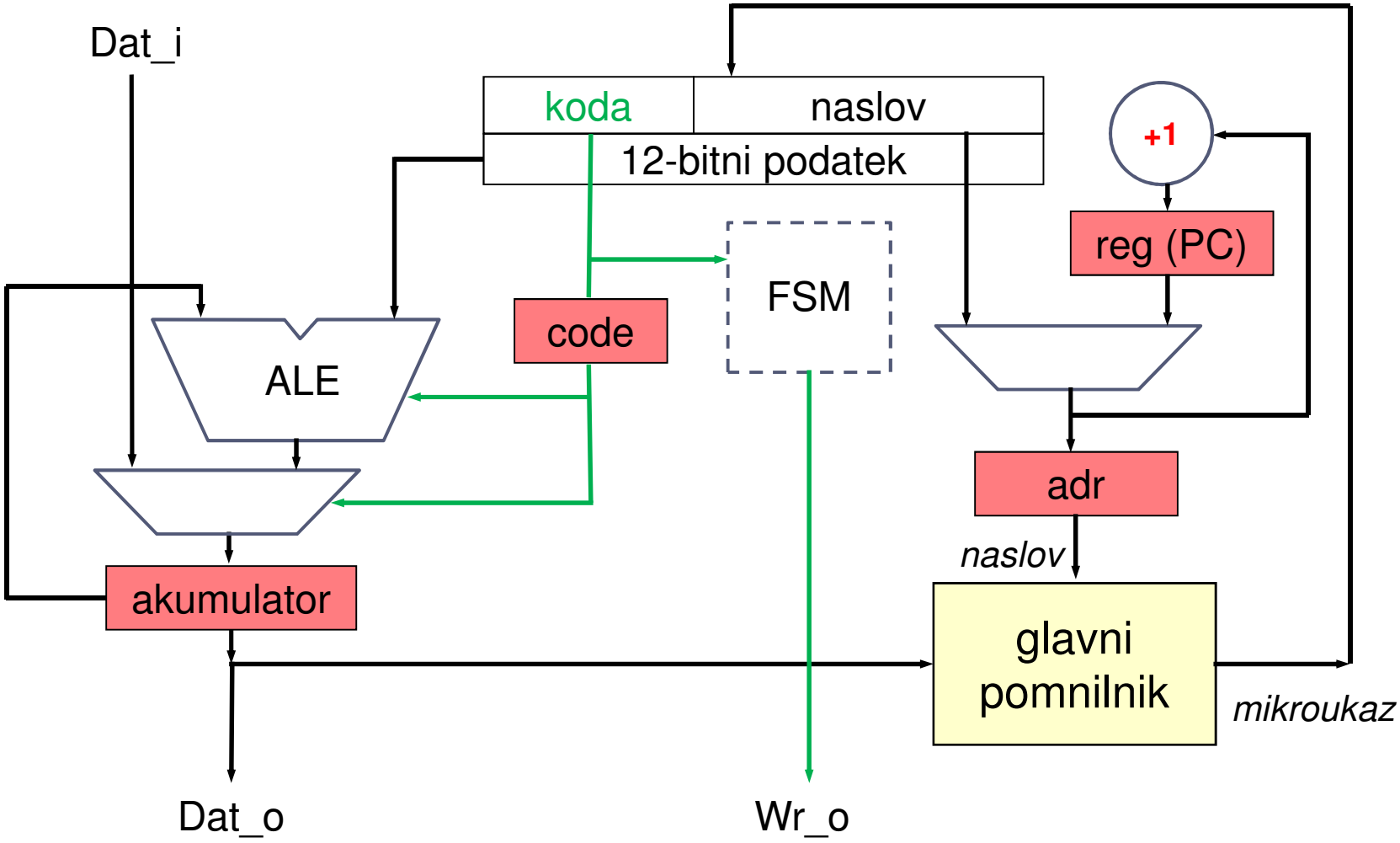
Potek izvedbe programa

```
0 :  lda   3
1 :  add  3
2 :  jmp  0
3 :    5
```

0001	0000 0011
1000	0000 0011
0100	0000 0000
0000 0000 0101	

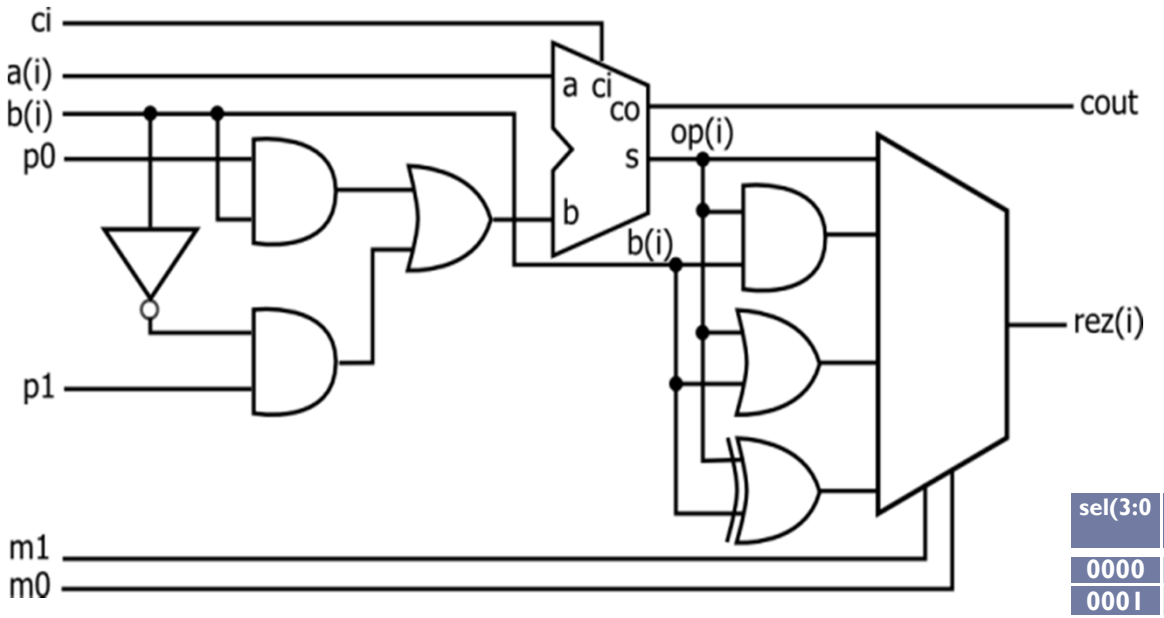


Nadgradnja: vhodno – izhodna enota



Nadgradnja ALE

- zgradimo univerzalne celice za posamezen bit
  - aritmetične operacije: seštej (s prenosom), odštej, prištej 1…
  - logične operacije: AND, OR, XOR



sel(3:0	p(1:0)	m(1:0)	Cprop	Cinv	Cset	ukaz
0000	00	00	0	0	0	lda
0001	01	00	0	0	0	add
0010	10	00	0	0	1	sub
0011	11	00	0	0	0	dec
0100	00	00	0	0	1	inc
0101	01	00	1	0	0	adc
0110	10	00	0	1	0	sbc
1x01	00	01	0	0	0	anda
1x10	00	10	0	0	0	ora
1x11	00	11	0	0	0	xora