



# RAČUNALNIŠKEA ARHITEKTURA

## 6 Centralna procesna enota



## 6 Centralna procesna enota

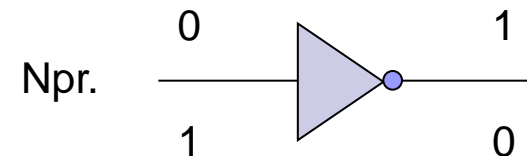
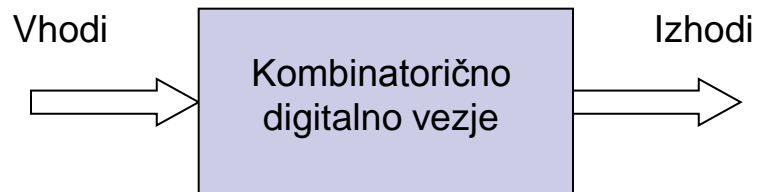
- Zgradba in delovanje CPE
- Hipotetični računalnik HIP - lastnosti
  - Primer: Procesor ARM9
- Hipotetični računalnik HIP - zgradba
  - Kontrolna enota
  - Podatkovna enota
- Izvajanje ukazov v procesorju HIP
- Paralelno izvajanje ukazov
- Cevovodna CPE
- Primer 5-stopenjske cevovodne CPE
- Večizstavitveni procesorji



## 6.1 Zgradba in delovanje CPE

- CPE (Centralna procesna enota ali tudi procesor) je enota, ki izvršuje ukaze, zato njena zmogljivost v veliki meri določa zmogljivost računalnika.
- Poleg CPE ima večina računalnikov še druge procesorje, večinoma v vhodno/izhodnem delu računalnika.
- Osnovni principi delovanja so za vse vrste procesorjev enaki.

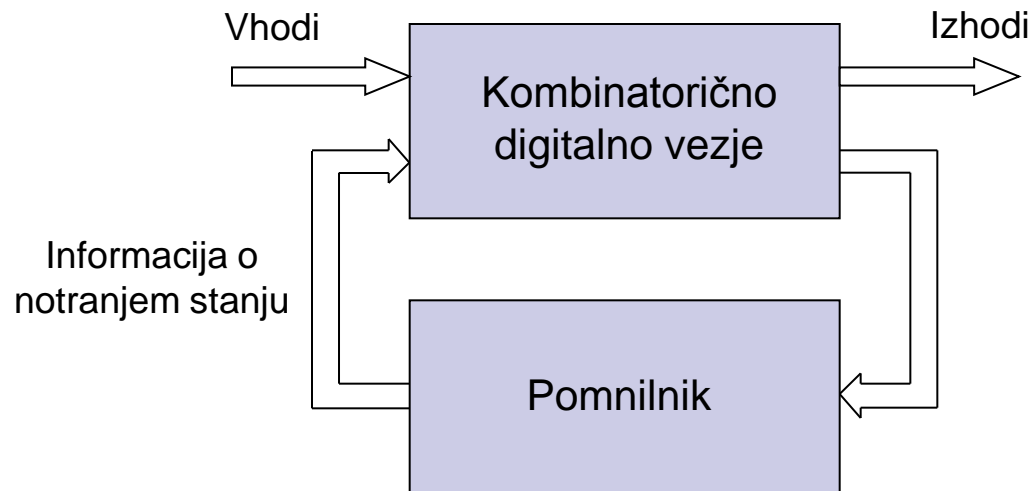
- CPE je digitalni sistem (zgrajena iz digitalnih elektronskih vezij) posebne vrste.
- Dve skupini digitalnih vezij:
  - Kombinatorična digitalna vezja
    - Stanje izhodov je odvisno samo od trenutnega stanja vhodov





□ Pomnilniška (sekvenčna) digitalna vezja

- Stanje izhodov je odvisno od trenutnega stanja vhodov in tudi od prejšnjih stanj vhodov
- Pomnilniška vezja si zapomnijo stanja
- Prejšnja stanja običajno označimo kot **notranja stanja**, ki odražajo prejšnja stanja vhodov

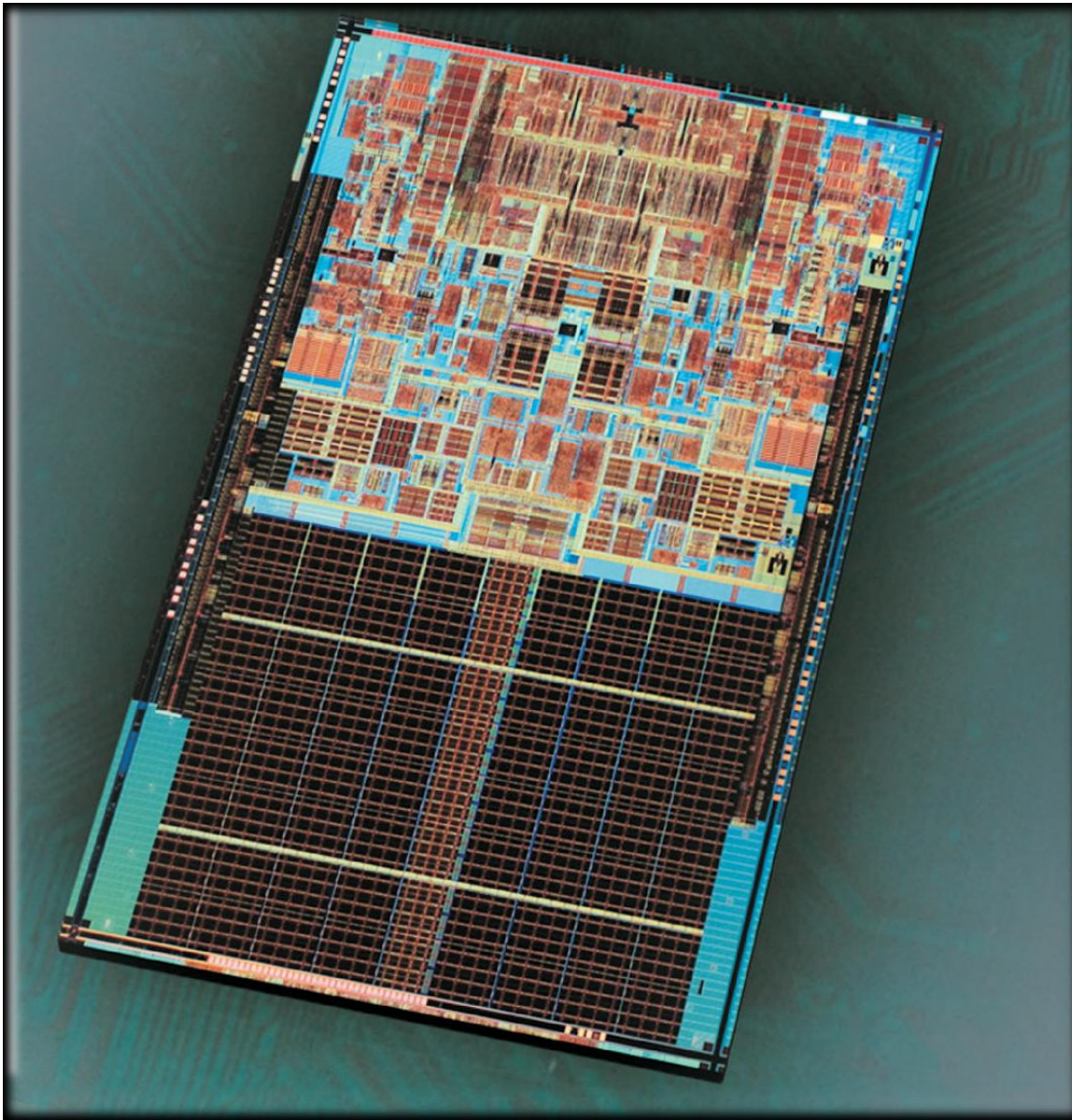




- Pomnilniška vezja:
  - Flip-flop – enobitna pomnilniška celica
  - Register
  - Števec
  - Pomnilnik
  
- CPE je zgrajena iz kombinatoričnih in pomnilniških digitalnih vezij.
  
- Trenutno stanje vseh pomnilniških vezij predstavlja **stanje CPE.**



- Delovanje CPE je v vsakem trenutku odvisno od trenutnega stanja vhodov v CPE in od trenutnega stanja CPE.
- Število vseh možnih notranjih stanj CPE je odvisno od velikosti (zmogljivosti) CPE.
- Število bitov, s katerimi so predstavljena notranja stanja CPE, je od nekaj 10 pa do 10.000 ali tudi več.
- Digitalna vezja iz katerih je narejena CPE so danes običajno na enem čipu.

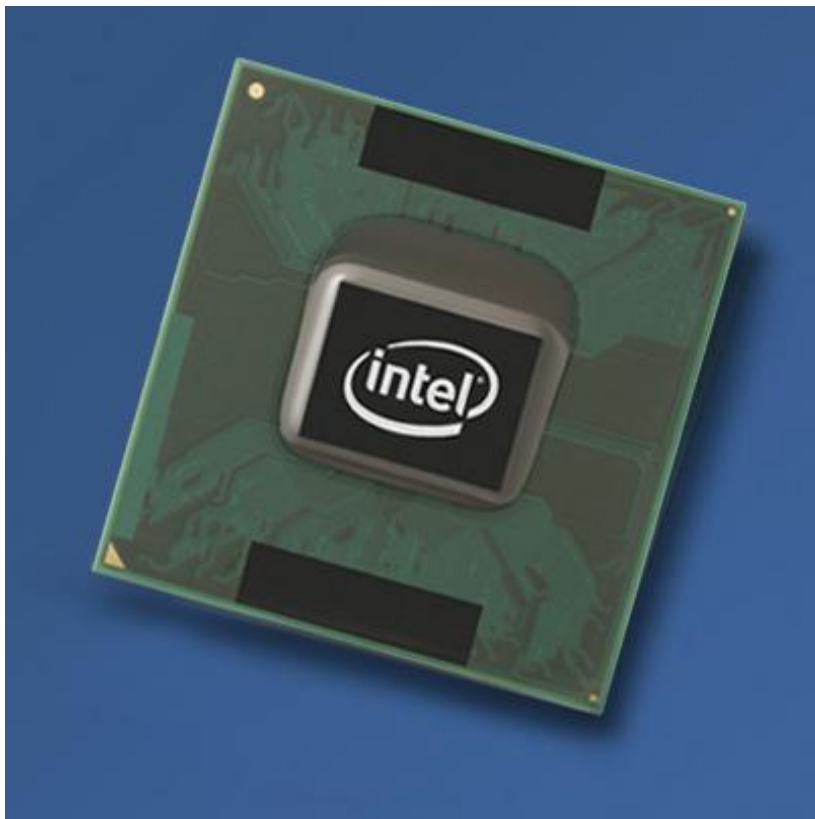


Intel Core2 Duo

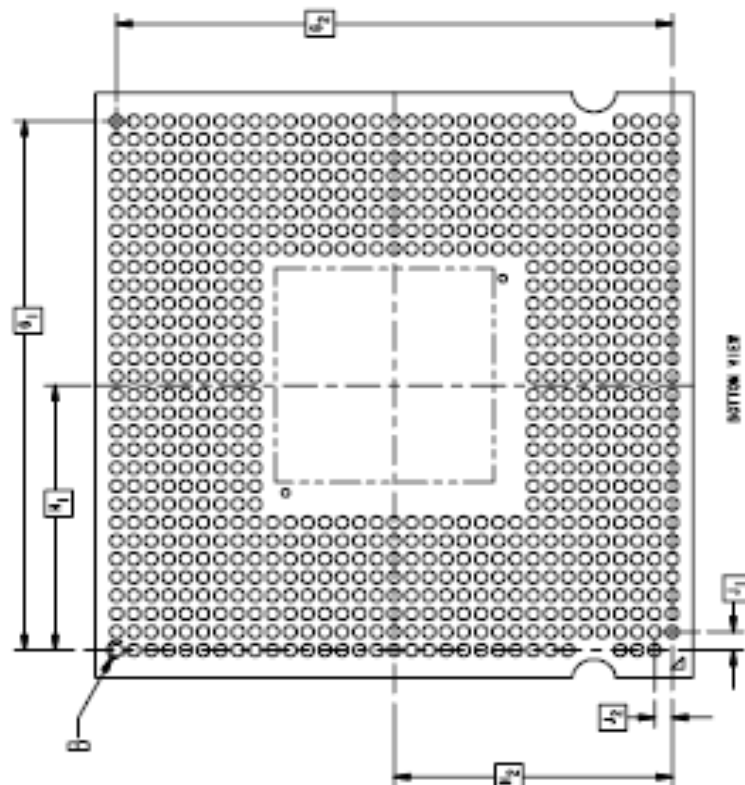




## Intel 80x86



Kontakti za povezavo  
čipa z matično ploščo

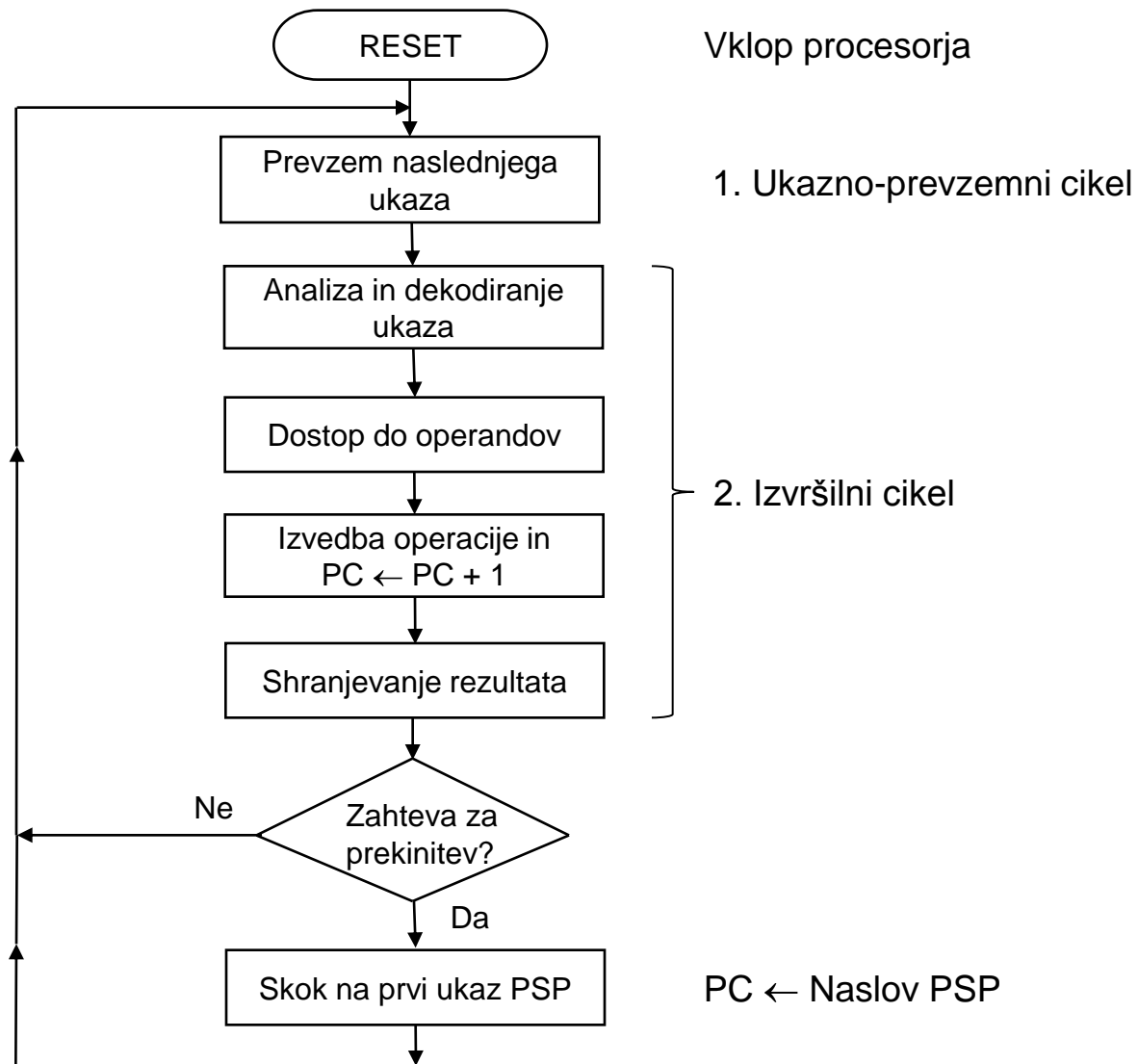




- Delovanje CPE v von Neumannovem računalniku smo opisali z dvema korakoma:
  - Jemanje ukaza iz pomnilnika (ukazno-prevzemni cikel), naslov ukaza je v programskem števcu (PC)
  - Izvrševanje prevzetega ukaza (izvršilni cikel):
    - Analiza (dekodiranje) ukaza
    - Prenos operandov v CPE (če niso že v registrih v CPE)
    - Izvedba z ukazom določene operacije
    - Shranjevanje rezultata (če je potrebno)
    - $PC \leftarrow PC + 1$  ali  $PC \leftarrow$  ciljni naslov pri skočnih ukazih

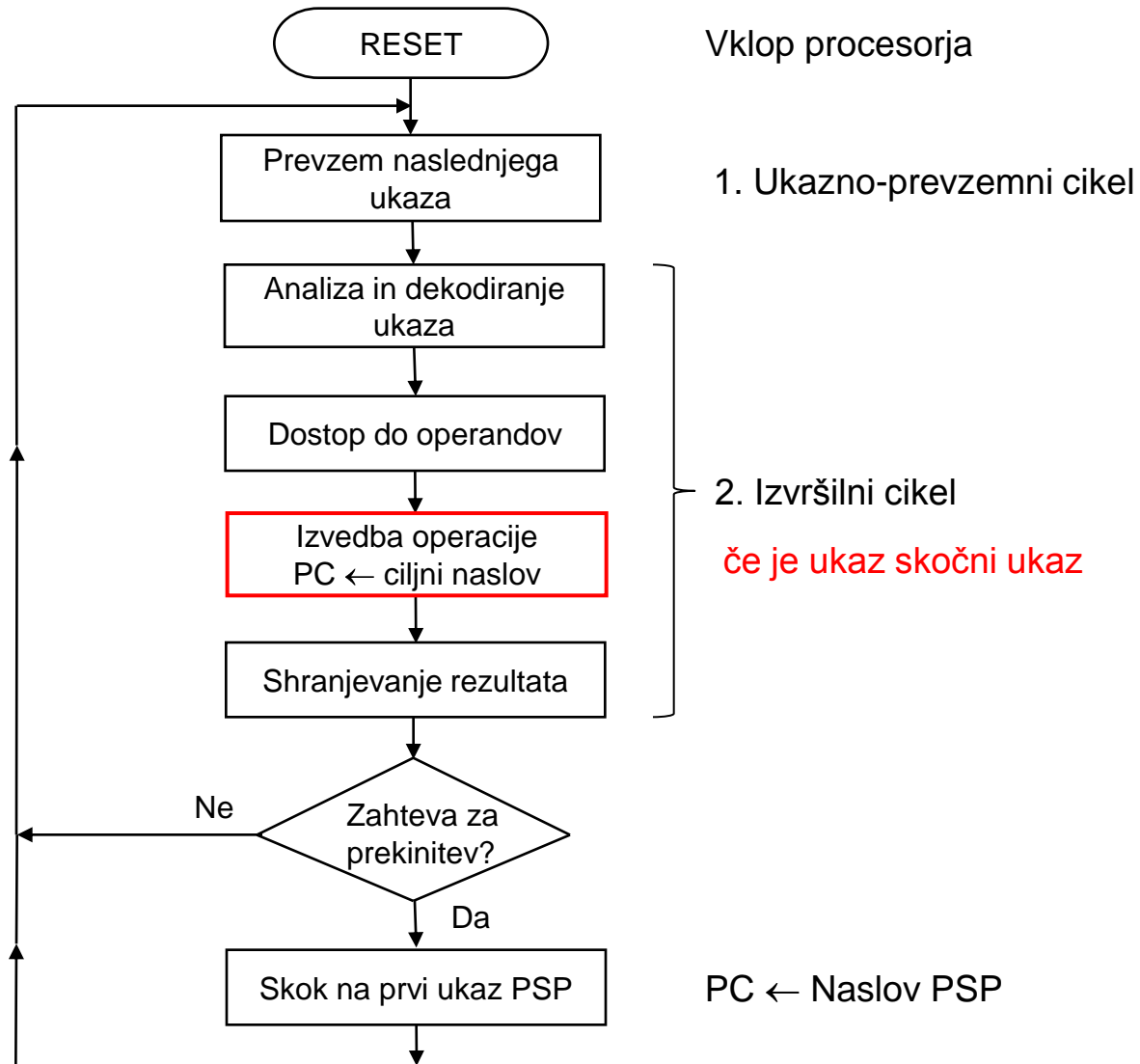


## Centralna procesna enota





## Centralna procesna enota

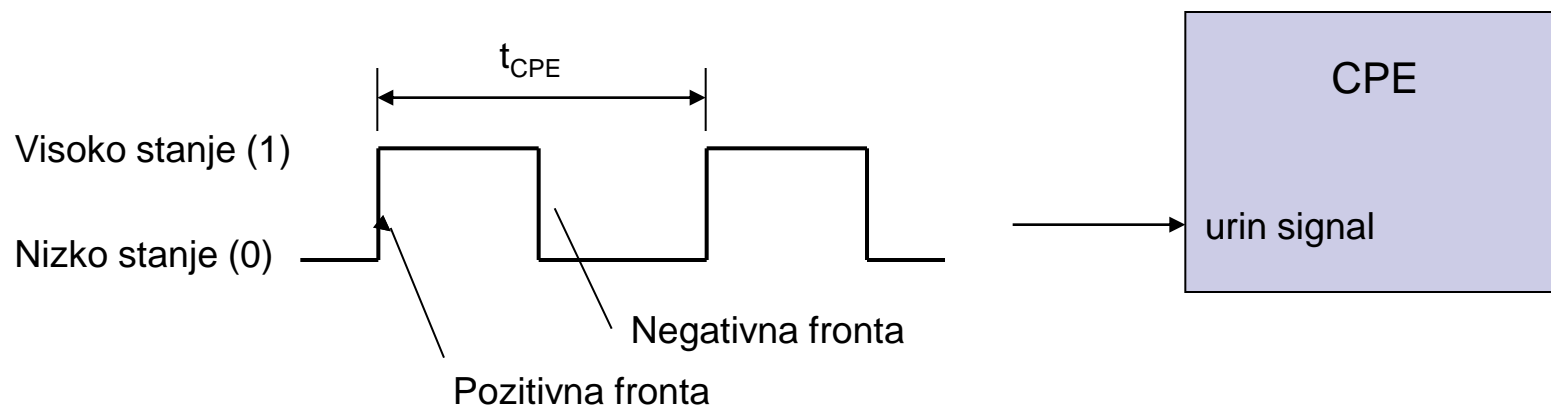




- Naslov prvega ukaza po vklopu (RESET) je določen z nekim pravilom.
- Po zaključku 2. koraka prične CPE zopet s 1. korakom, kar se ponavlja, dokler CPE deluje.
- Izjema je, kadar pride do prekinitve ali pasti.
- Takrat se namesto prevzema naslednjega ukaza izvrši skok na ukaz, katerega naslov je določen z načinom delovanja prekinitiev.

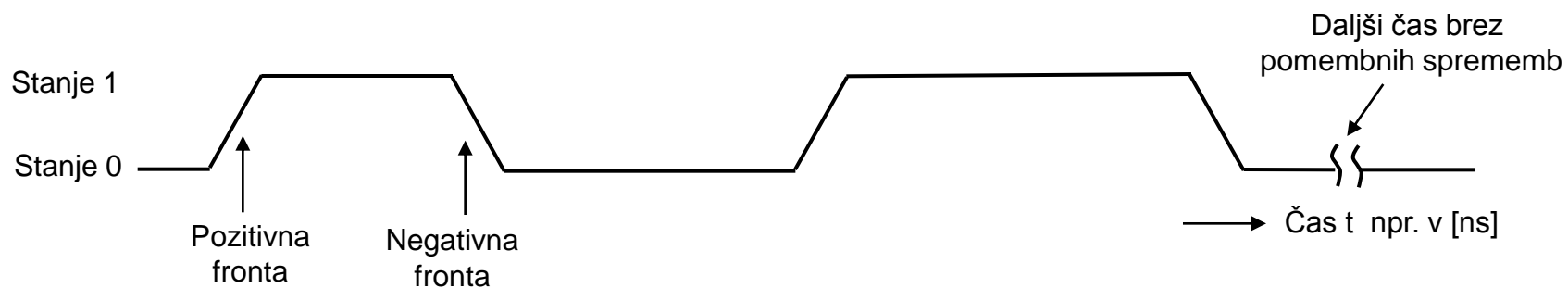
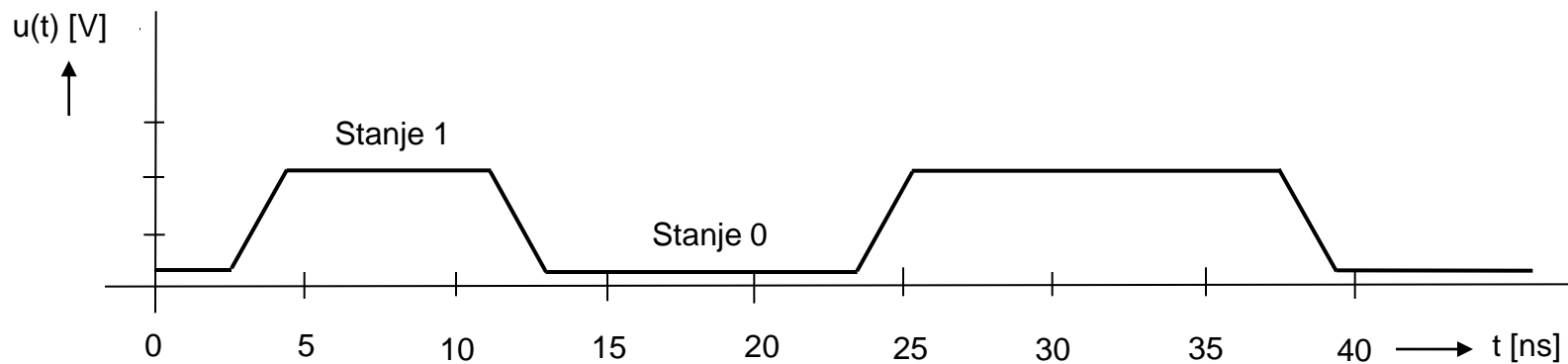


- Vsak od teh korakov je sestavljen iz bolj elementarnih korakov in realizacija CPE je realizacija teh elementarnih korakov.
- Vsak elementarni korak se opravi v eni ali več periodah urinega signala – CPE ure.





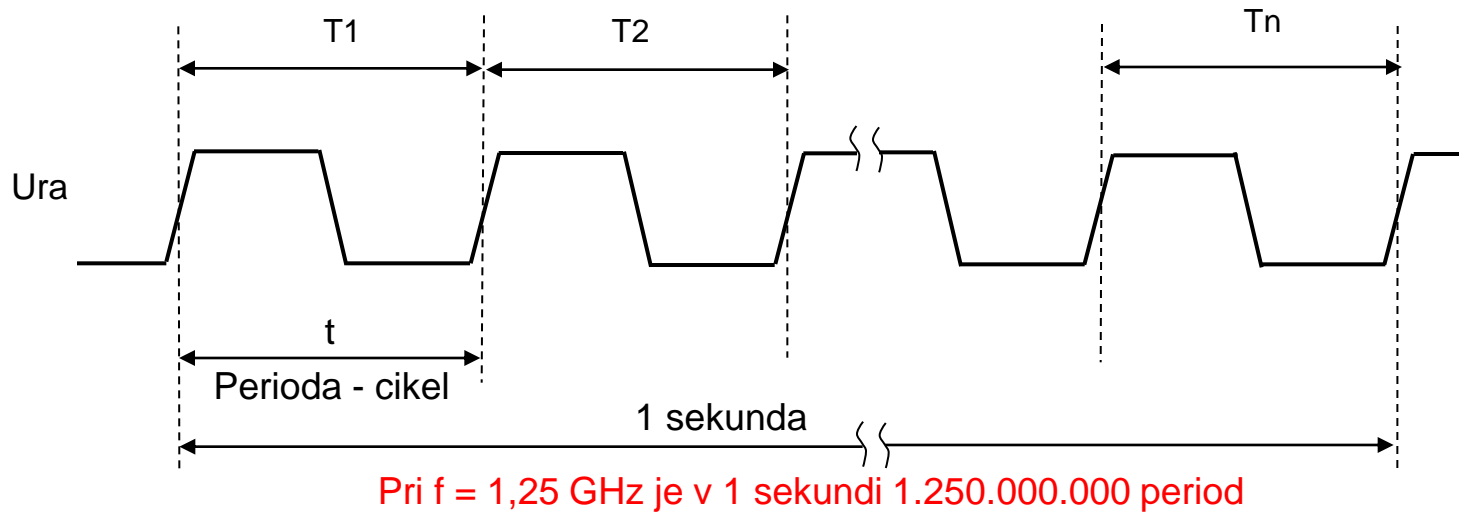
## Časovni diagram signala







## Urin signal - periodičen pravokoten signal



Frekvenca periodičnega signala  $f$  = število period (ciklov) v 1 sekundi

Enota za frekvenco je Hertz [Hz] :  $1 \text{ Hz} = 1 [\text{perioda/s}] = 1[1/\text{s}] = 1[\text{s}^{-1}]$

Čas trajanja ene periode  $t = 1 / f$

$$f = 1,25 \text{ GHz} \Rightarrow t = \frac{1}{f} = \frac{1}{1,25 * 10^9 \text{ /s}} = \frac{1}{1,25} * 10^{-9} \text{ s} = 0,8 * 10^{-9} \text{ s} = 0,8 \text{ ns}$$



- Stanje CPE se, tako kot stanja vseh digitalnih vezij, spreminja samo ob prehodu urinega signala iz enega v drugo stanje.
- Fronto ob kateri se dogajajo spremembe v CPE imenujemo **aktivna fronta**.
- CPE je lahko narejena tudi tako, da spreminja stanje ob pozitivni in negativni fronti, to pomeni, da sta aktivni obe fronti. V eni urini periodi se tako lahko izvršita dve spremembi stanja CPE.



- Frekvenca urinega signala, ki pride v CPE, se pogosto v CPE množi z nekim faktorjem (npr. 6, 12, 17 ...).
- Notranja ura ima tako višjo frekvenco (krajšo periodo) kot zunanja. Primer Intel i5 2400:
  - Frekvenca ure vodila: 100 MHz
  - Razmerje frekvence ure vodilo/jedro: 31
  - Frekvenca ure jedra: 3,1 GHz ( $100\text{MHz} \times 31 = 3100\text{MHz} = 3,1\text{GHz}$ )
- Stanje CPE se spreminja ob frontah notranje ure.
- Krajša urina perioda pomeni hitrejše delovanje CPE.



- Krajšanje urine periode (višanje frekvence) je pogojeno s hitrostjo uporabljenih digitalnih vezij in številom vezij skozi katera potuje signal.
- Najkrajše trajanje elementarnega koraka v CPE je ena urina perioda (ali tudi pol periode, če sta aktivni obe fronti, za kar pa je potrebno bolj komplicirano logično vezje).
- Ukazni in izvršilni cikel trajata vedno celo število urinih period.
- Število urinih period za izvedbo ukaza se med ukazi lahko precej razlikuje.



## 6.2 Hipotetični računalnik HIP - lastnosti

- RISC arhitektura
- 32-bitni računalnik
- 3-operandni registrsko-registrski (load/store) računalnik
- 32-bitni pomnilniški naslov
- 32-bitno podatkovno vodilo
- Dolžina pomnilniške besede 8 bitov
- 32 splošnonamenskih 32-bitnih registrov
- Dolžine pomnilniških operandov 8, 16 in 32 bitov

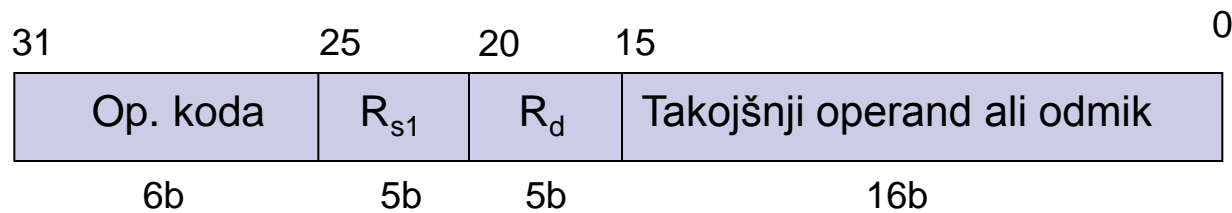


- Sestavljeni pomnilniški operandi so shranjeni po pravilu debelega konca.
- Cela števila so predstavljena v dvojiškem komplementu.
- Vsi ukazi so dolgi 32 bitov (4 bajte).
- Načini naslavljanja:
  - Takojšnje s 16-bitnim takojšnjim operandom
  - Bazno s 16-bitnim odmikom
- Dostop do pomnilniških operandov samo z ukazoma LOAD in STORE.

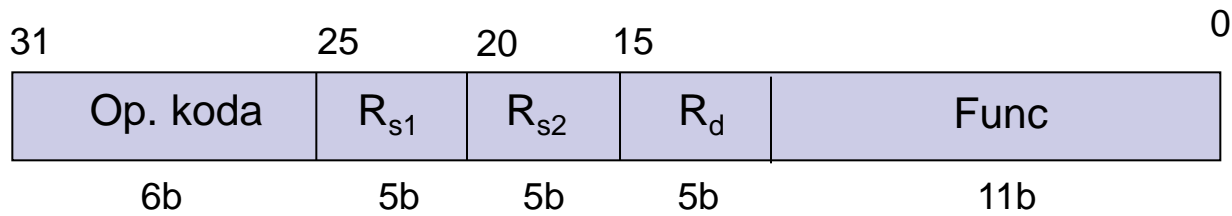


- Ukazi za pogojne skoke uporabljajo PC-relativno naslavljanje s 16-bitnim odmikom.
- Dva formata ukazov:

### Format 1



### Format 2





## Procesor ARM9:

- RISC arhitektura, vsi ukazi so dolgi 32 bitov
- 32-bitni računalnik
- 3-operandni registrsko-registrski (load/store) računalnik
- 32-bitni pomnilniški naslov
- 32-bitno podatkovno vodilo
- Dolžina pomnilniške besede 8 bitov
- 16 splošnonamenskih 32-bitnih registrov
- Dolžine pomnilniških operandov 8, 16 in 32 bitov
- Sestavljeni pomnilniški operandi so shranjeni po pravilu tankega konca
- Ukazi in operandi morajo biti v pomnilniku poravnani (shranjeni na naravnih naslovih)
- Cela števila so predstavljena v dvojiškem komplementu
- Realna števila po standardu IEEE-754



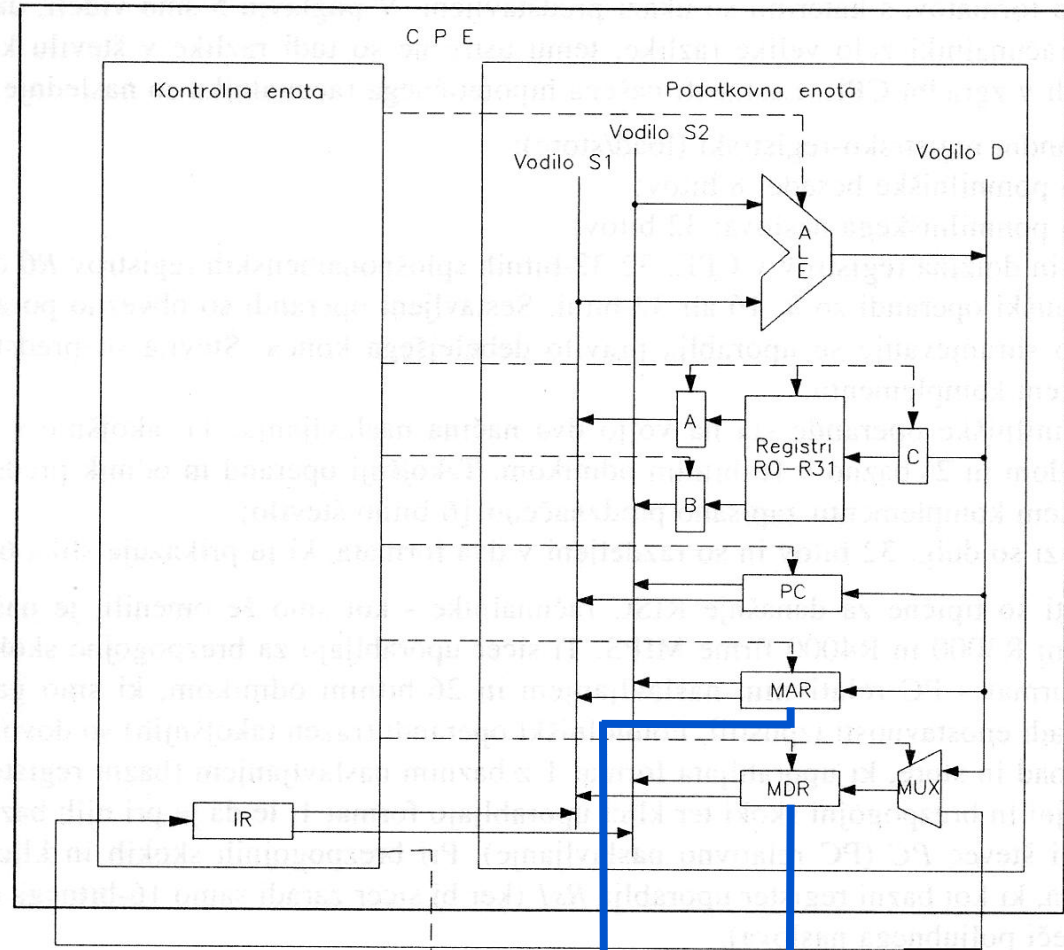


## 6.3 Hipotetični računalnik HIP - zgradba

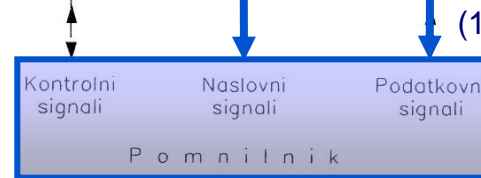
- Kontrolna enota
  
- Podatkovna enota:
  - ALE
  
  - Programsko dostopni registri



- 32-bitni pomnilniški naslov
- 32-bitno podatkovno vodilo:
- Pisanje operanda v pomnilnik (1)



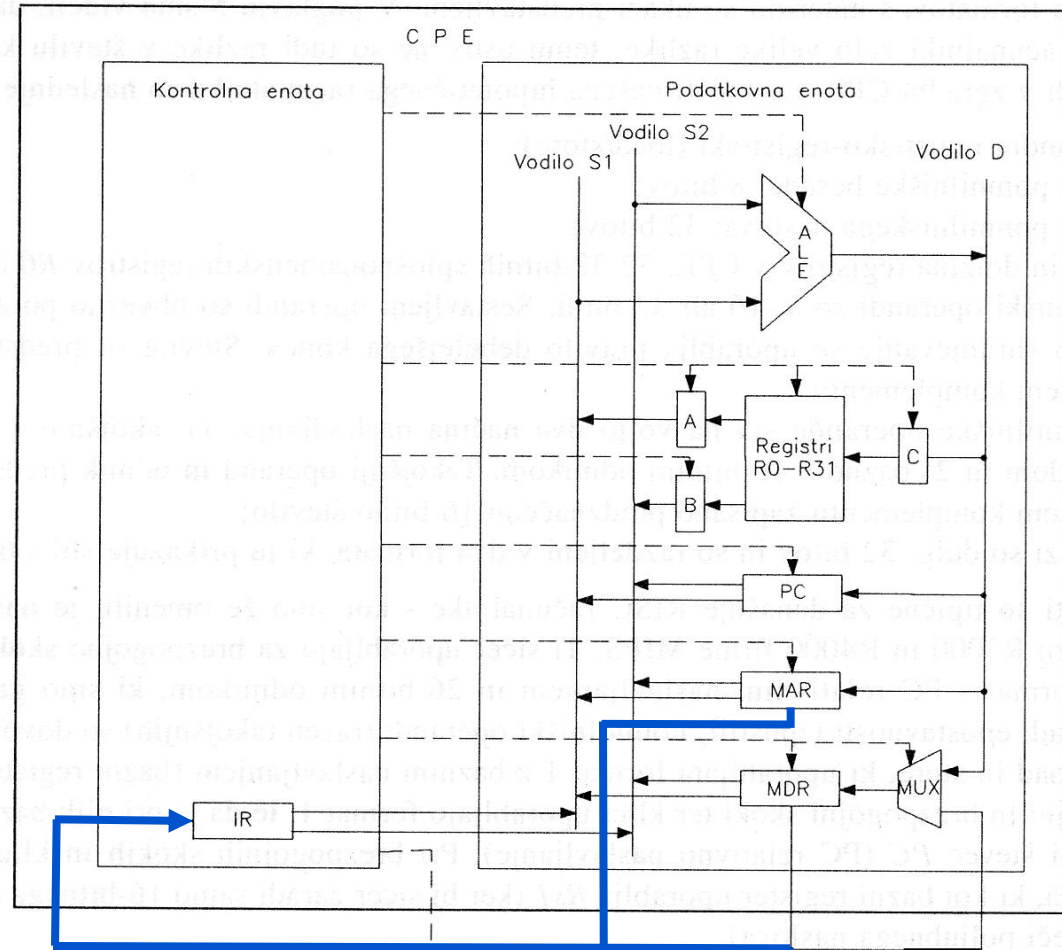
PC - Programski števec  
 MAR - Pomnilniški naslovni register  
 MDR - Pomnilniški podatkovni register  
 MUX - Multiplekser  
 IR - Ukazni register



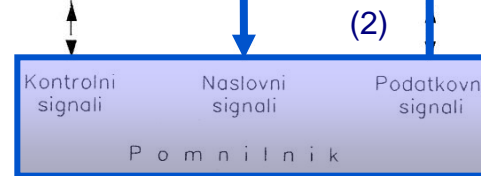
D.Kodek: Arhitektura računalniških sistemov



- 32-bitni pomnilniški naslov
- 32-bitno podatkovno vodilo:
- Pisanje operanda v pomnilnik
- Branje ukaza iz pomnilnika (2)



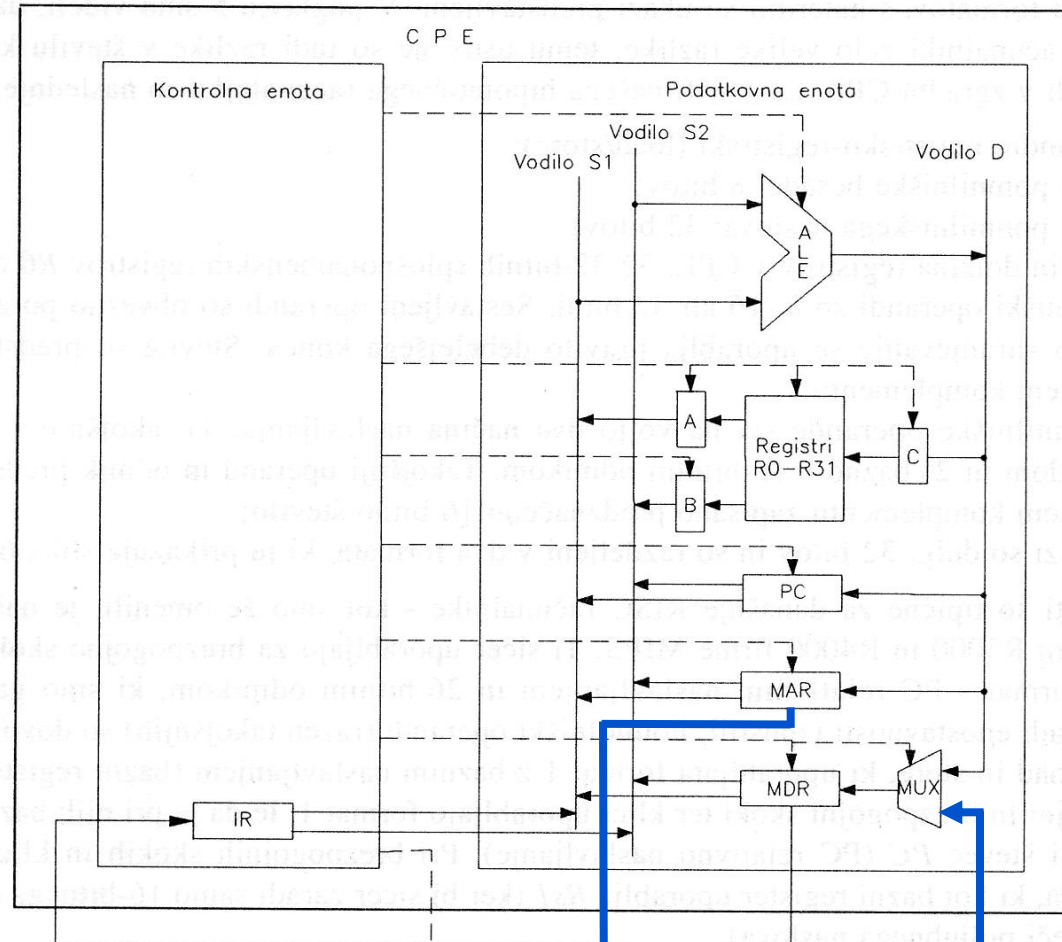
PC - Programski števec  
 MAR - Pomnilniški naslovni register  
 MDR - Pomnilniški podatkovni register  
 MUX - Multiplexer  
 IR - Ukazni register



D.Kodek: Arhitektura računalniških sistemov



- 32-bitni pomnilniški naslov
- 32-bitno podatkovno vodilo:
- Pisanje operanda v pomnilnik
- Branje ukaza iz pomnilnika
- Branje operanda iz pomnilnika (3)



PC - Programski števec  
 MAR - Pomnilniški naslovni register  
 MDR - Pomnilniški podatkovni register  
 MUX - Multiplekser  
 IR - Ukazni register



D.Kodek: Arhitektura računalniških sistemov

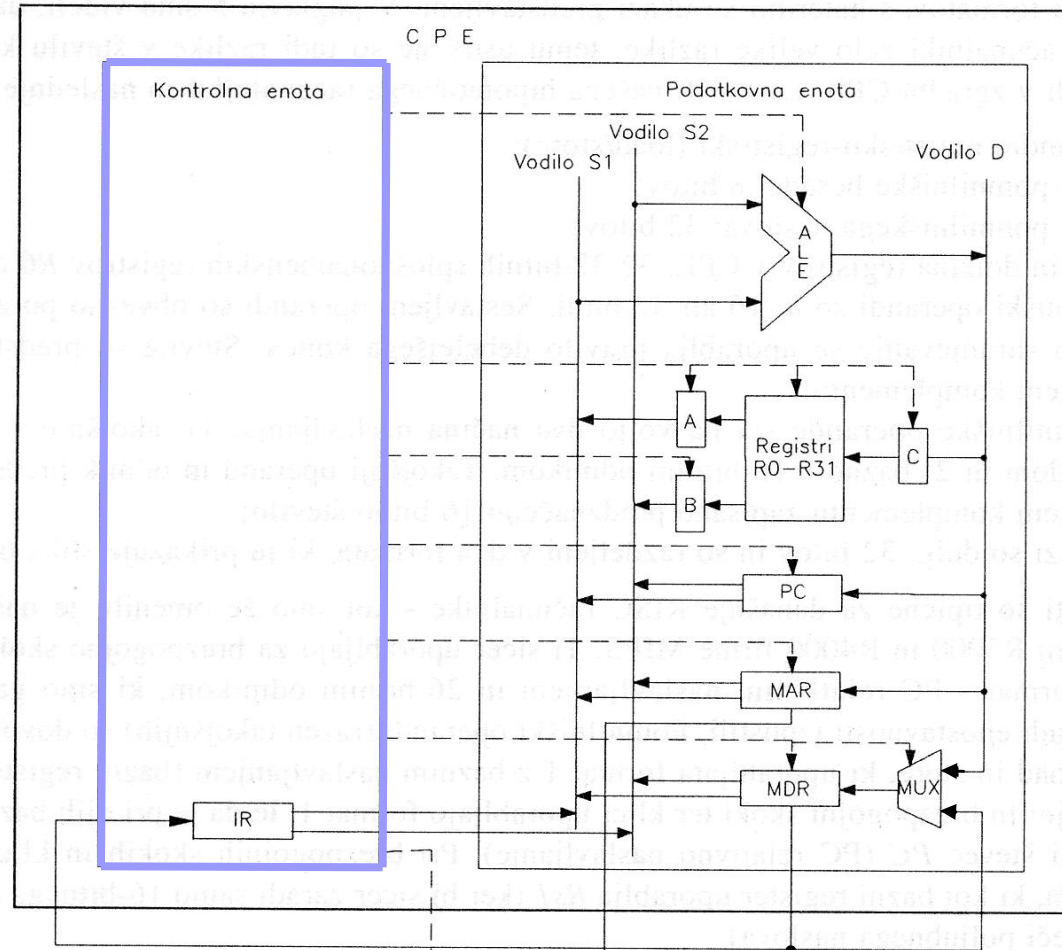


## ■ Kontrolna enota

- Digitalno vezje (pomnilniško + kombinatorično), ki na osnovi ukaza v ukaznem registru tvori kontrolne signale.
- Kontrolni signali sprožajo elementarne operacije v podatkovni enoti in s tem izvajanje ukaza.
- IR = 32-bitni ukazni register v katerega se v ukazno- prevzemnem ciklu prenese strojni ukaz iz pomnilnika.



# Kontrolna enota



- PC - Programski števec
- MAR - Pomnilniški naslovni register
- MDR - Pomnilniški podatkovni register
- MUX - Multiplekser
- IR - Ukazni register

Kontrolni signali

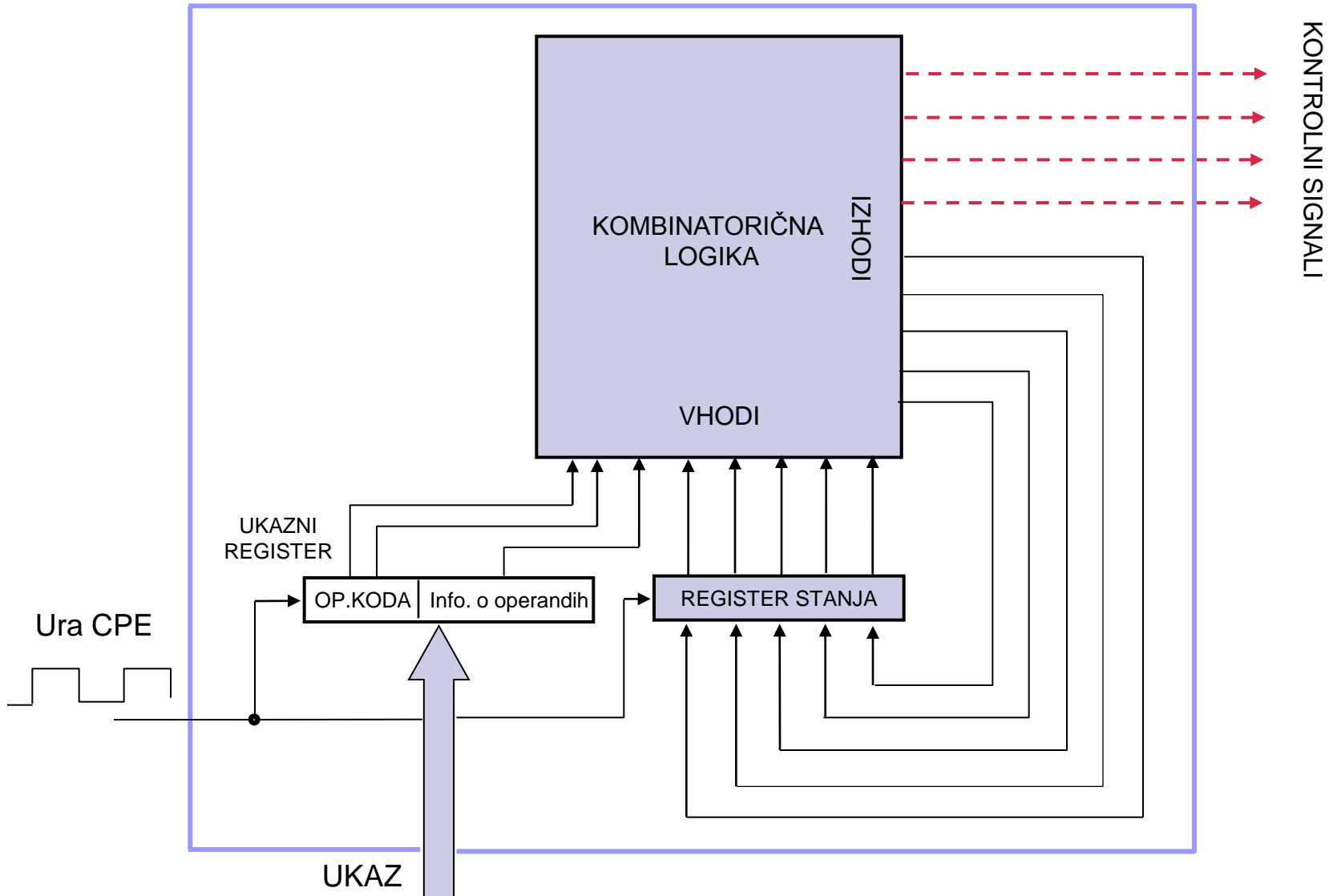
Naslovni signali

Podatkovni signali

P o m n i l n i k



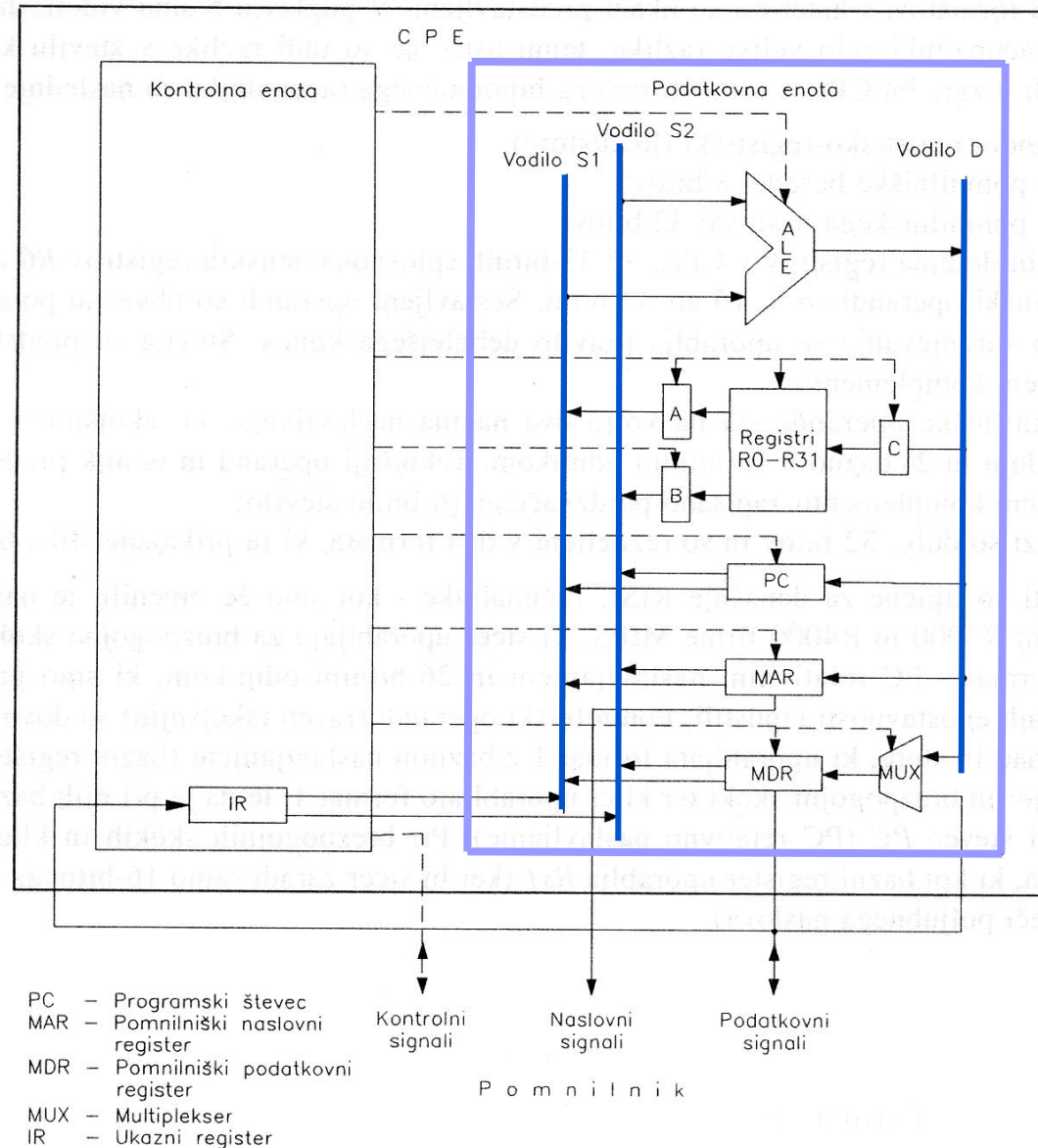
# Kontrolna enota (trdo ožičena)





## Podatkovna enota

- 32-bitna notranja vodila

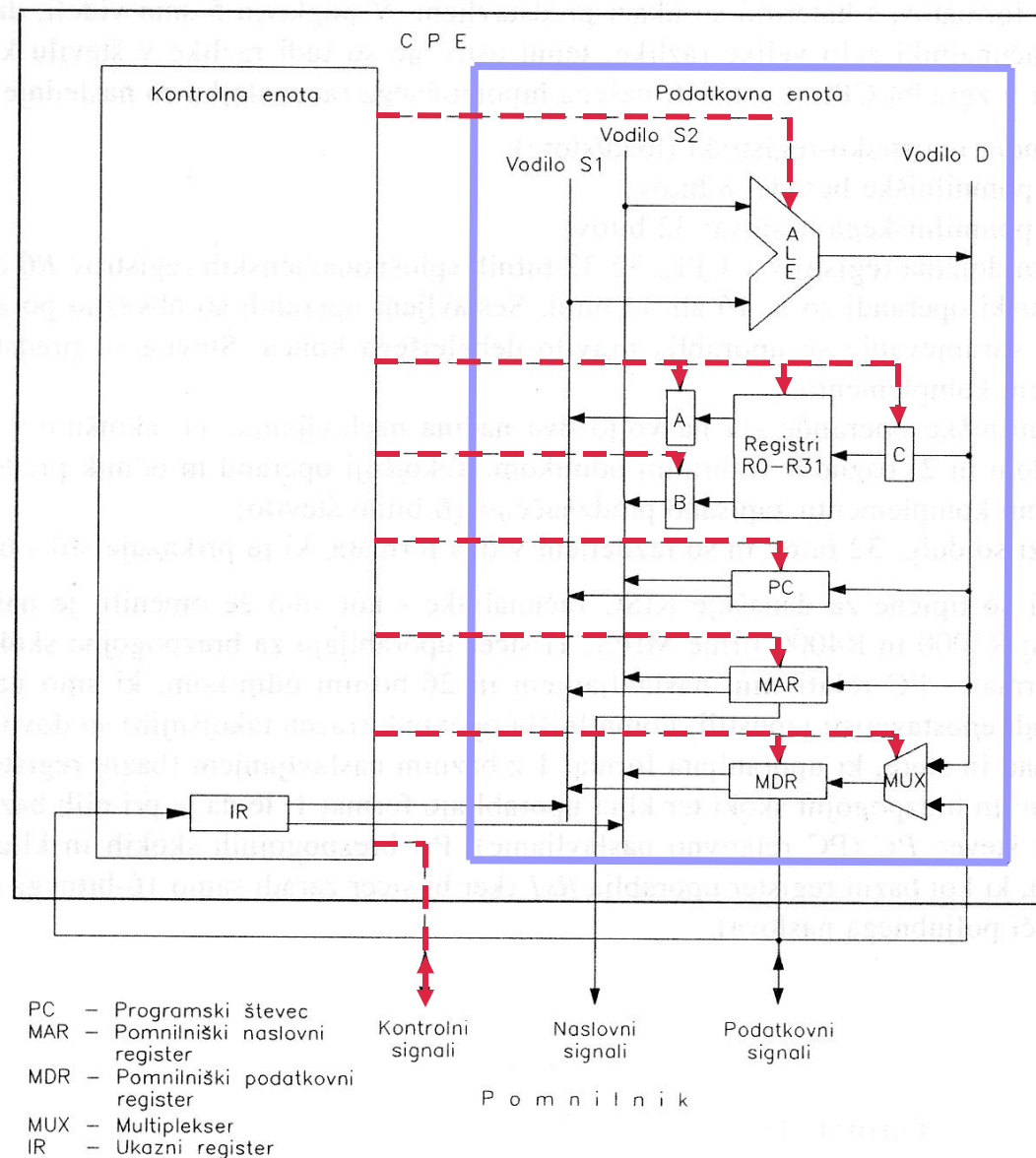






## Podatkovna enota

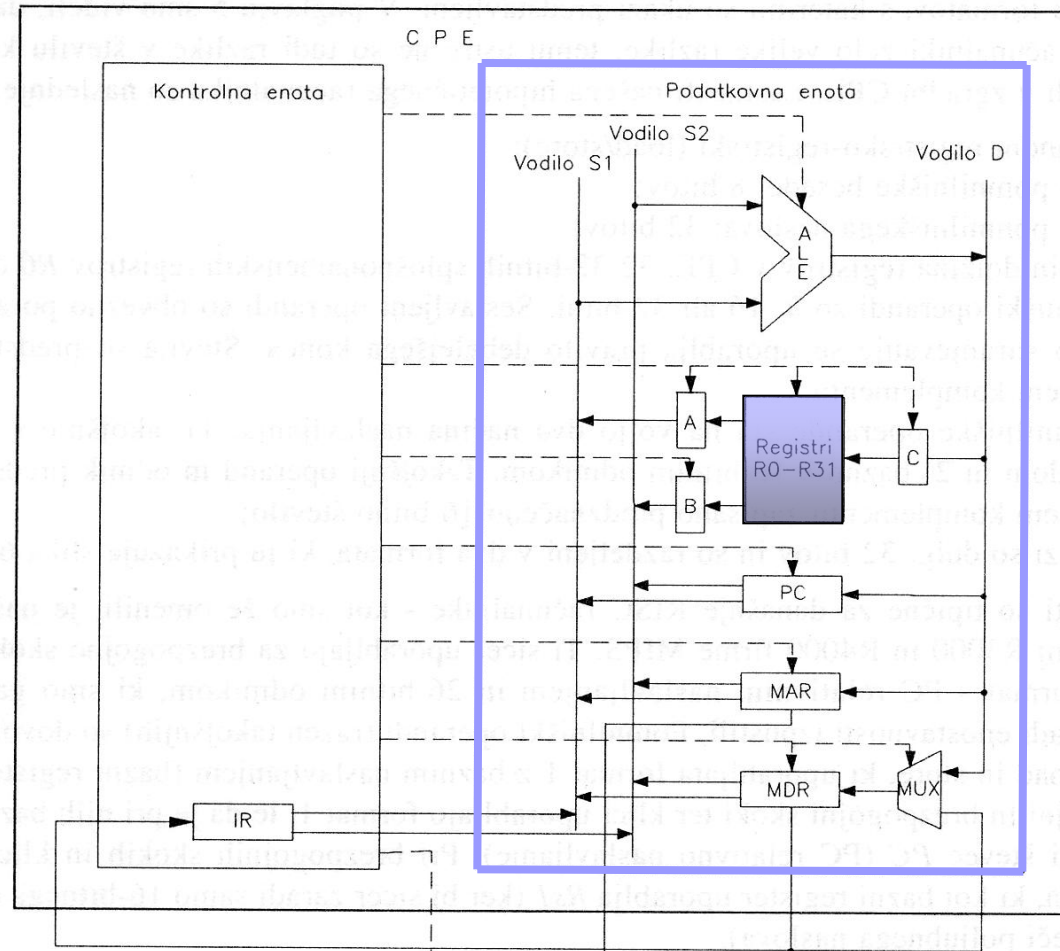
- 32-bitna notranja vodila
- Kontrolni signali





## Podatkovna enota

- 32-bitna notranja vodila
- Kontrolni signali
- 32 programsko dostopnih registrov R0 – R31



PC – Programski števec  
 MAR – Pomnilniški naslovni register  
 MDR – Pomnilniški podatkovni register  
 MUX – Multiplekser  
 IR – Ukazni register

Kontrolni signali

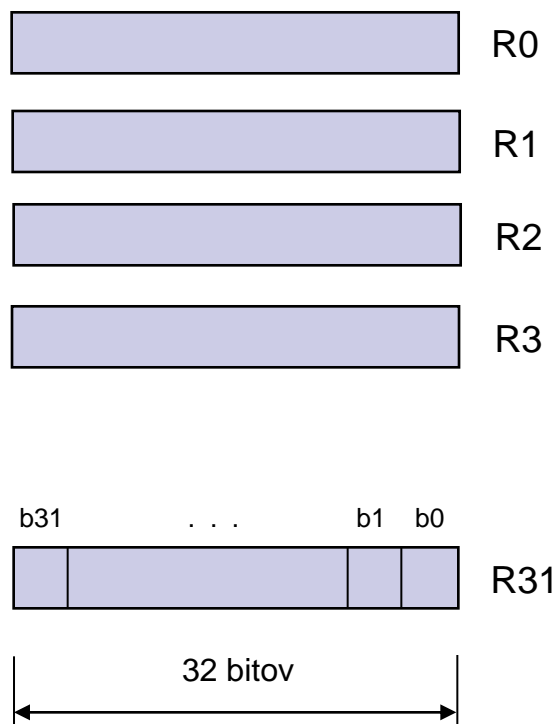
Naslovni signali

Podatkovni signali

P o m n i l n i k

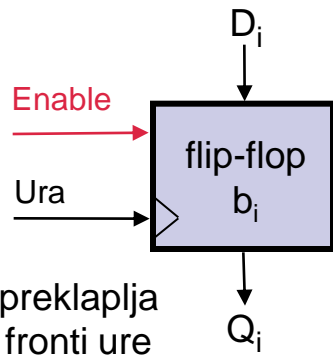
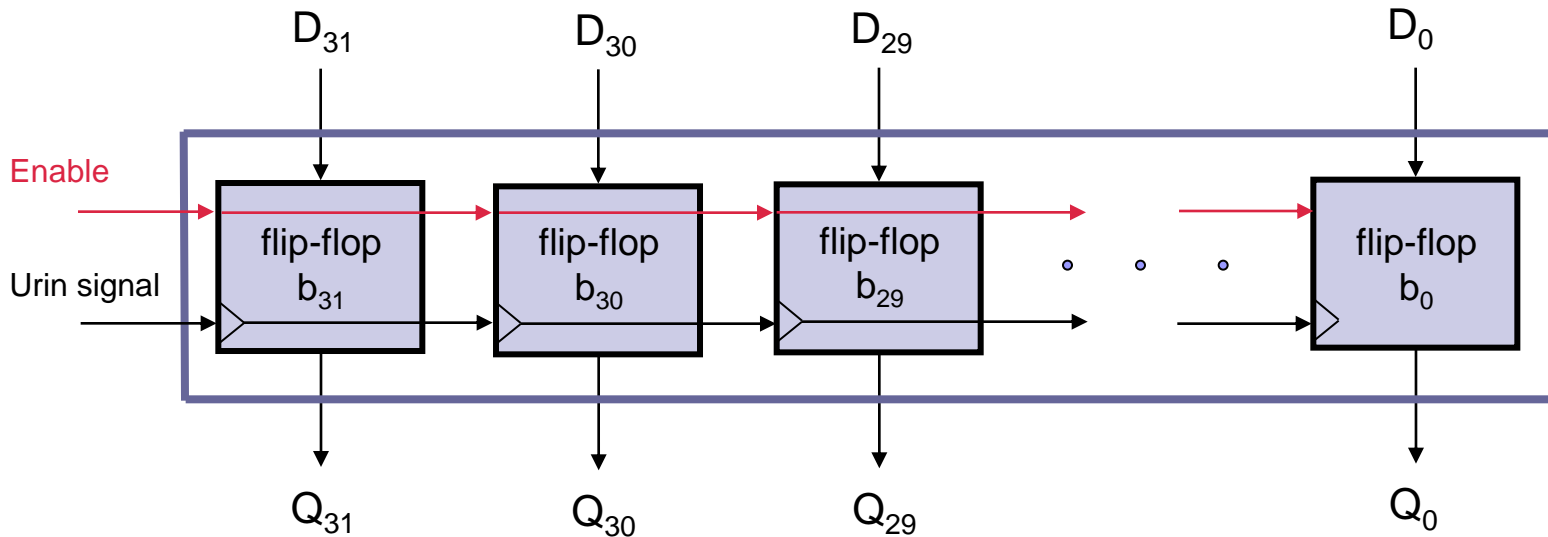


# Programsko dostopni registri procesorja HIP



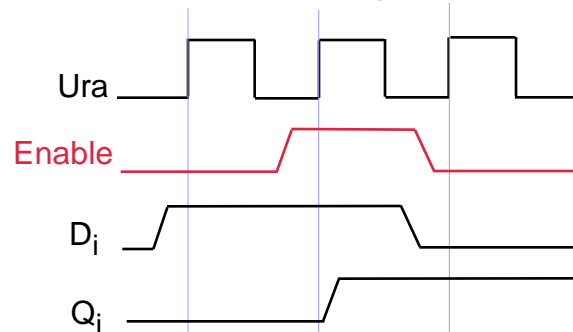


# 32-bitni register



Flip-flop, ki preklaplja ob pozitivni fronti ure

Časovni diagram



Pravilnostna tabela

Ura	En	$D_i$	$Q_i$
↑	0	0	Q
↑	0	1	Q
↑	1	0	0
↑	1	1	1



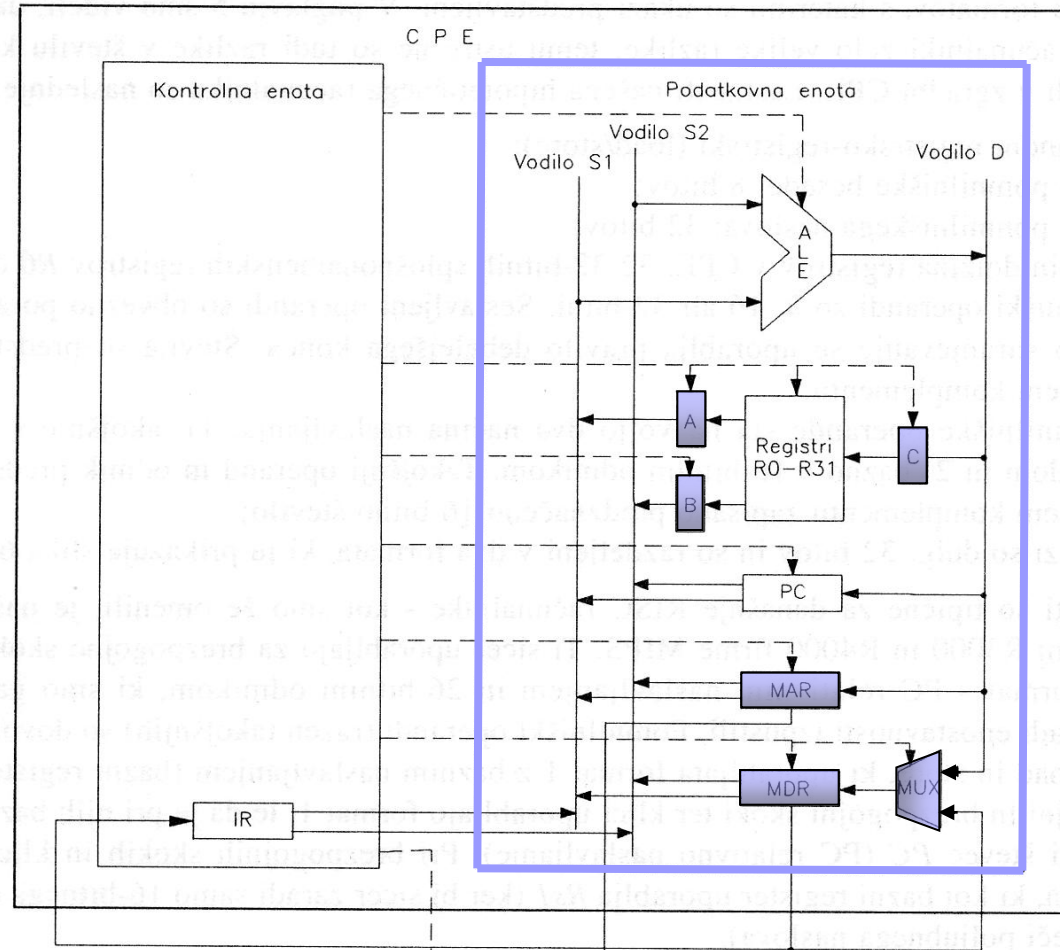
## Podatkovna enota

- 32-bitna notranja vodila
- Kontrolni signali
- 32 programsko dostopnih registrov R0 – R31

- Programsko nedostopni registri

- A, B in C
- MAR
- MDR

- Multiplekser



PC – Programski števec  
 MAR – Pomnilniški naslovni register  
 MDR – Pomnilniški podatkovni register  
 MUX – Multiplekser  
 IR – Ukazni register

Kontrolni signali

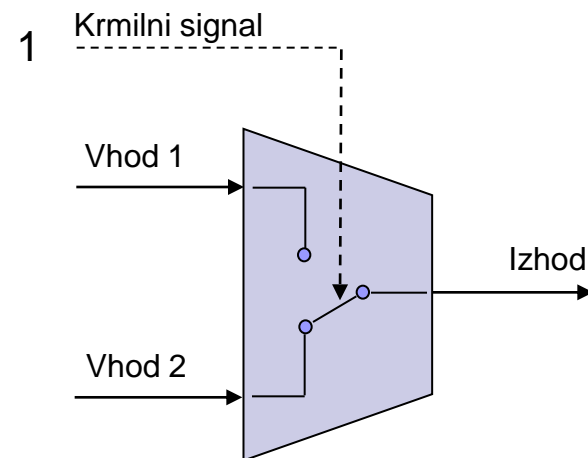
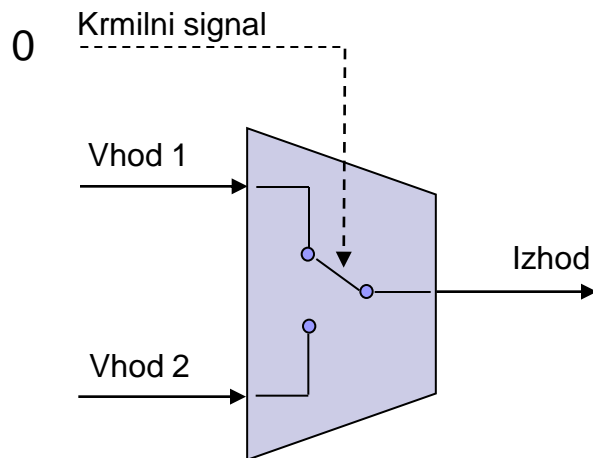
Naslovni signali

Podatkovni signali

P o m n i l n i k



- Multiplekser - MUX je digitalno vezje, ki iz več vhodnih signalov izbere enega in ga posreduje na izhod.
- Izbiro vhodnega signala določa krmilni signal.

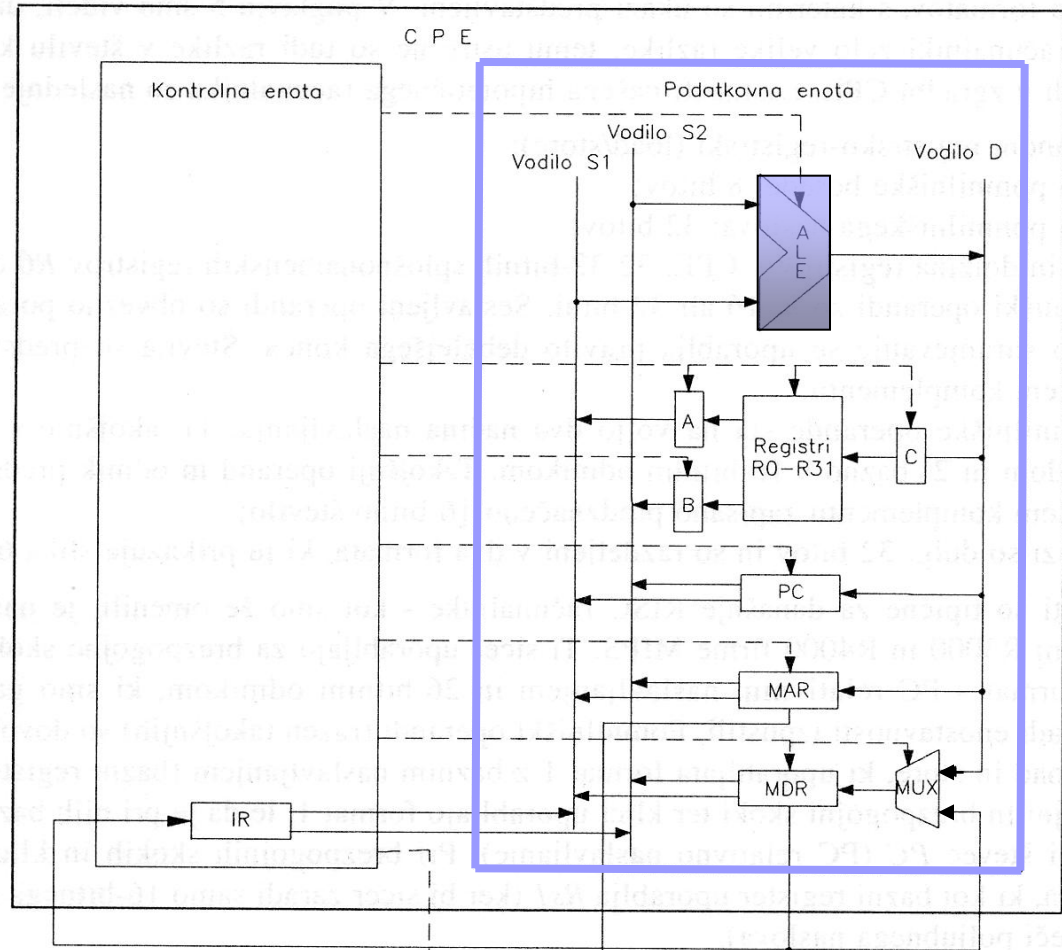




# Podatkovna enota

- 32 programsko dostopnih registrov R0 – R31

- ALE



- PC - Programski števec
- MAR - Pomnilniški naslovni register
- MDR - Pomnilniški podatkovni register
- MUX - Multiplexer
- IR - Ukazni register

Kontrolni signali

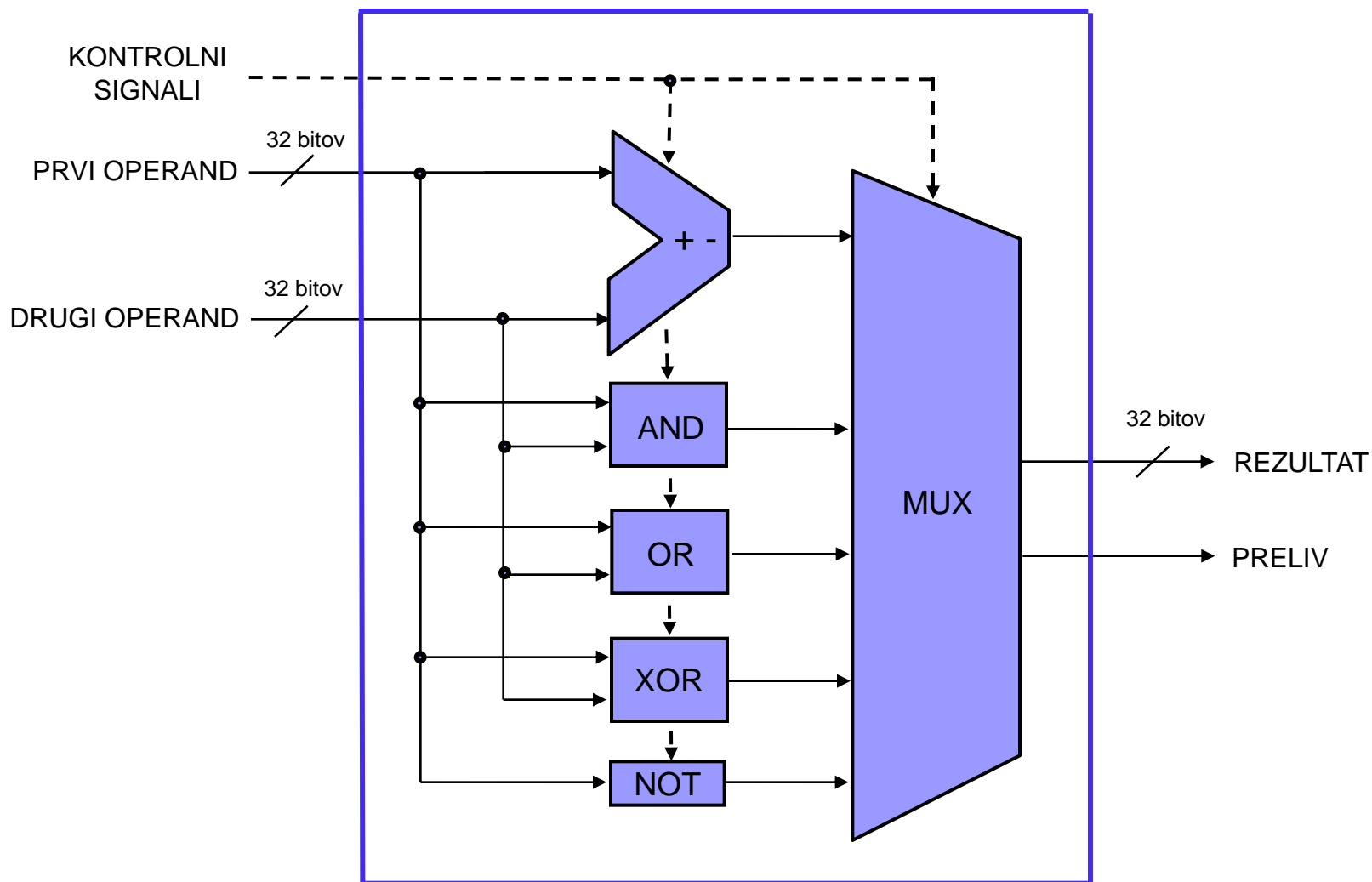
Naslovni signali

Podatkovni signali

P o m n i l n i k



# ALE

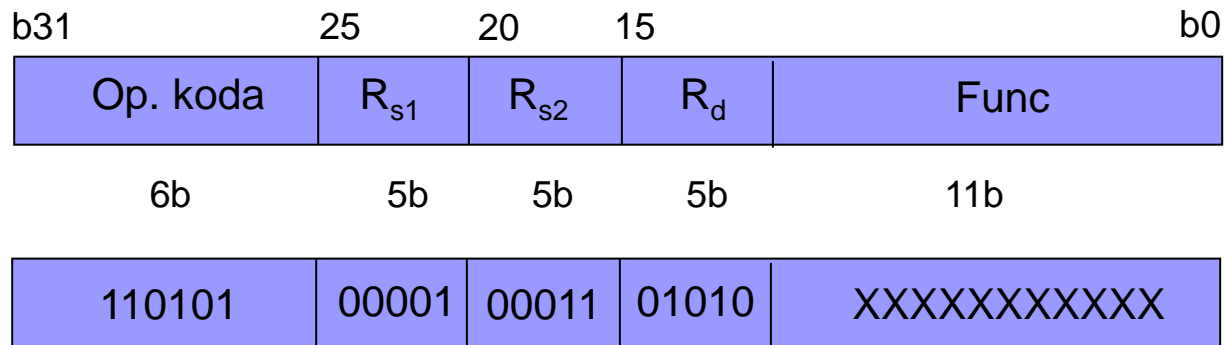




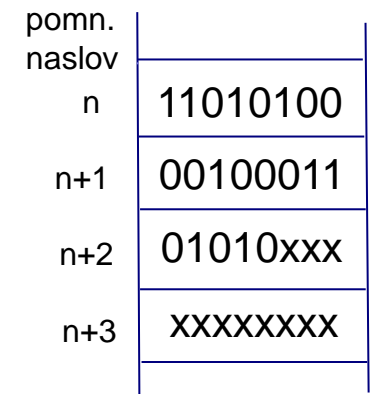


## 6.4 Izvajanje ukazov v procesorju HIP

- Primer izvajanja ukaza iz skupine ALE operacij:
- ADD R1, R3, R10      $R10 \leftarrow R1 + R3$
- Format ukaza:  
(če predpostavimo, da je op.koda ukaza ADD npr. 110101)



Strojni ukaz



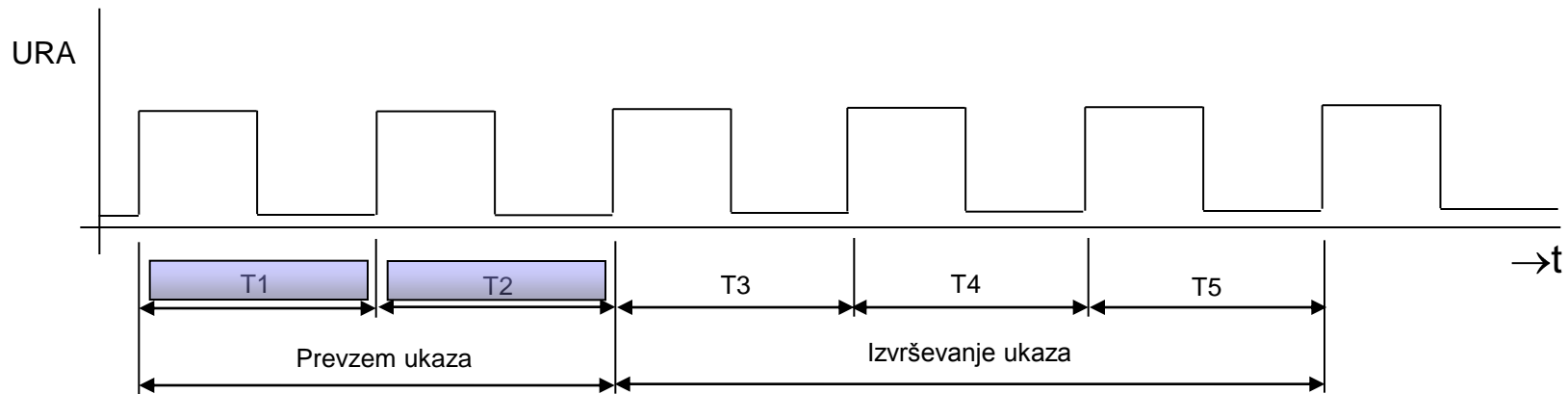
Strojni ukaz v pomnilniku  
po pravilu debelega konca



- Izvajanje ukazov lahko v našem hipotetičnem računalniku razdelimo na primer na 5 elementarnih korakov:
  - Prezem ukaza (IF - Instruction Fetch)
  - Dekodiranje ukaza in dostop do registrov (ID - Instruction Decode)
  - Izvrševanje ukaza (EX - Execute)
  - Dostop do pomnilnika (MA - Memory Access)
    - samo pri ukazih LOAD in STORE
  - Shranjevanje rezultata (WB - Write Back)



## Centralna procesna enota - izvajanje ukazov

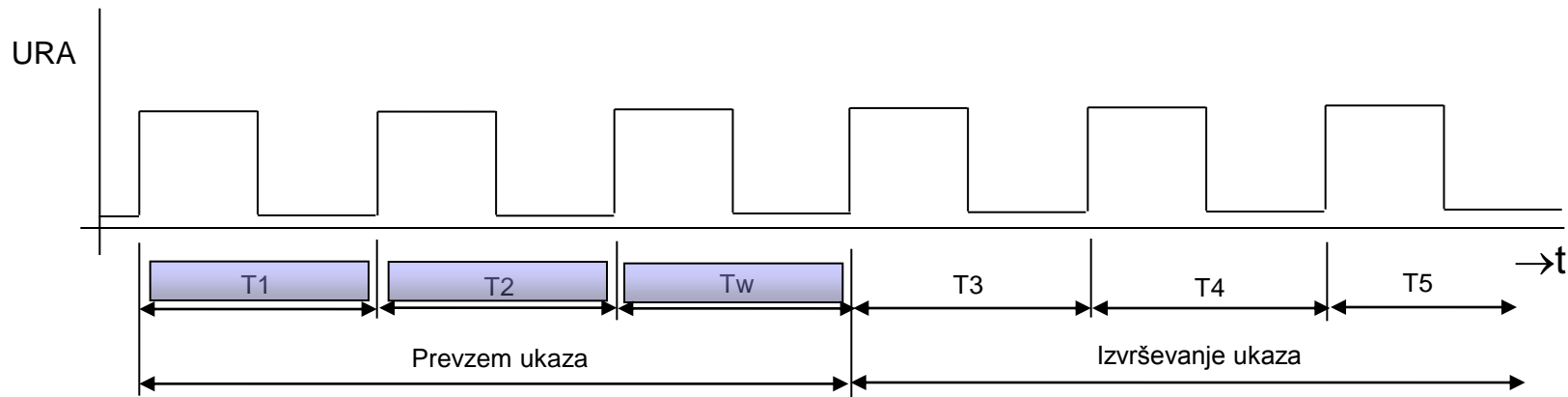


### ■ Izvajanje ukaza ADD traja 5 urinih period ( $CPI_{ALE}=5$ )

- T1: Dostop do ukaza v pomnilniku
- T2: Prenos ukaza iz pomnilnika v ukazni register



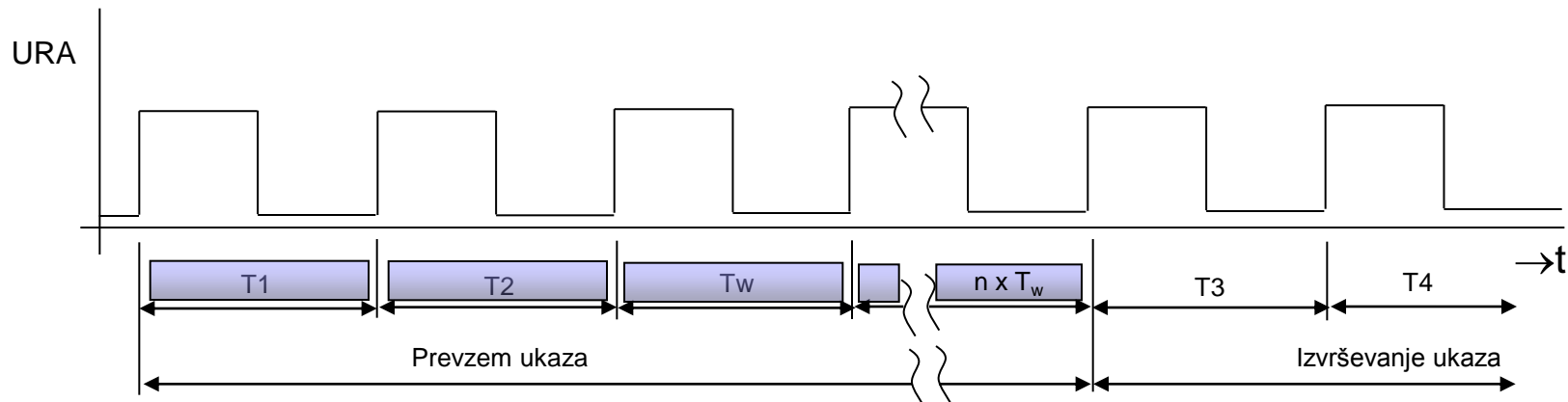
## Centralna procesna enota - izvajanje ukazov



- Izvajanje ukaza ADD traja 5 urinih period ( $CPI_{ALE}=5$ )
  - T1: Dostop do ukaza v pomnilniku
  - T2: Prenos ukaza iz pomnilnika v ukazni register
    - Pri prevzemu ukaza so lahko potrebne dodatne čakalne urine periode zaradi dostopa do pomnilnika



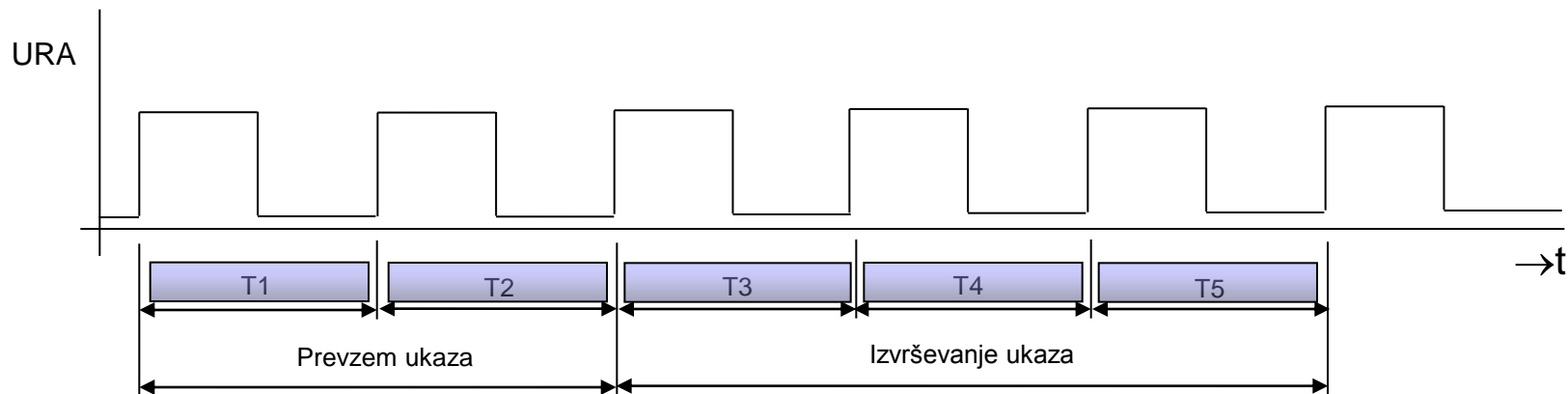
## Centralna procesna enota - izvajanje ukazov



- Izvajanje ukaza ADD traja 5 urinih period ( $CPI_{ALE}=5$ )
  - T1: Dostop do ukaza v pomnilniku
  - T2: Prenos ukaza iz pomnilnika v ukazni register
    - Pri prevzemu ukaza so lahko potrebne dodatne čakalne urine periode zaradi dostopa do pomnilnika



## Centralna procesna enota - izvajanje ukazov



- Izvajanje ukaza ADD traja 5 urinih period ( $CPI_{ALE}=5$ )
  - T1: Dostop do ukaza v pomnilniku
  - T2: Prenos ukaza iz pomnilnika v ukazni register
  - T3: Dekodiranje ukaza in dostop do operandov v registrih R1 in R3
  - T4: Izvrševanje operacije (seštevanje)
  - T5: Shranjevanje rezultata v register R10

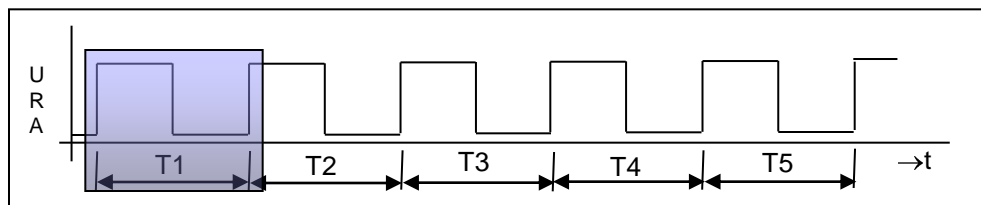
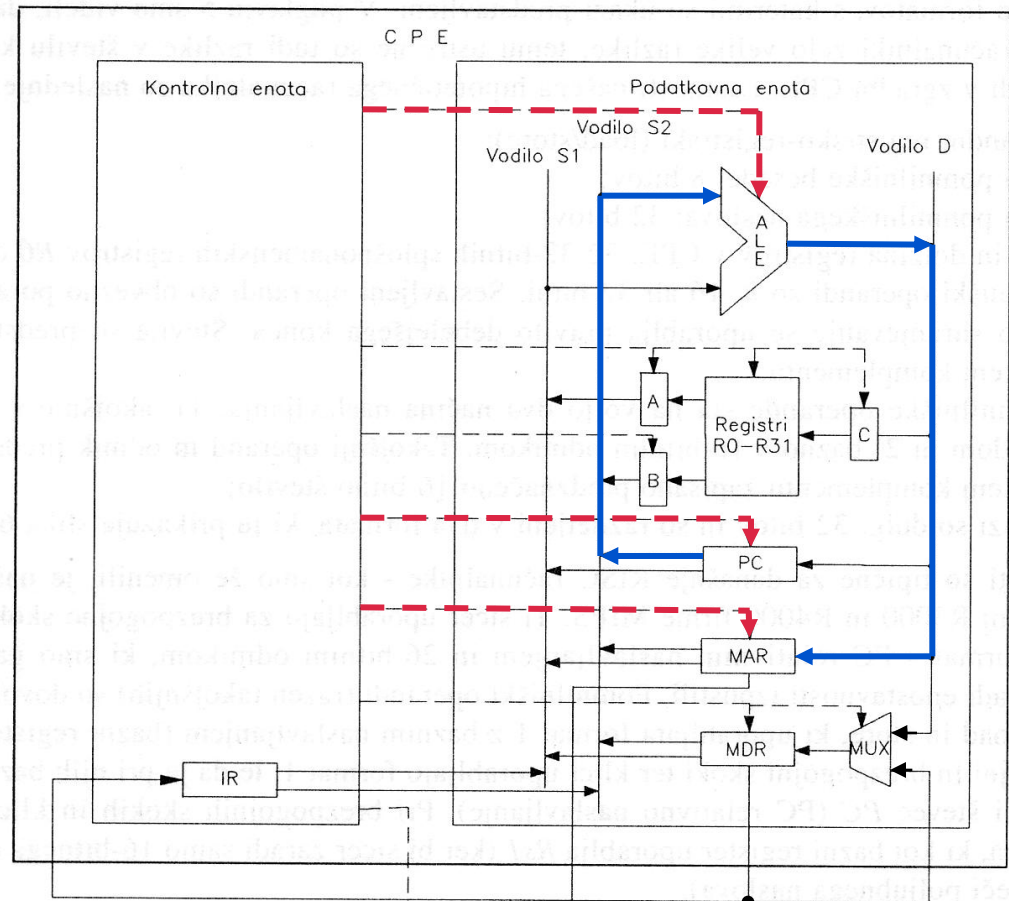


# Centralna procesna enota - izvajanje ukazov

## 1. urina perioda T1

MAR ← PC

Pomnilniški naslov ukaza se iz PC prenese v pomnilniški naslovni register MAR



Kontrolni signali

Naslovni signali

Podatkovni signali

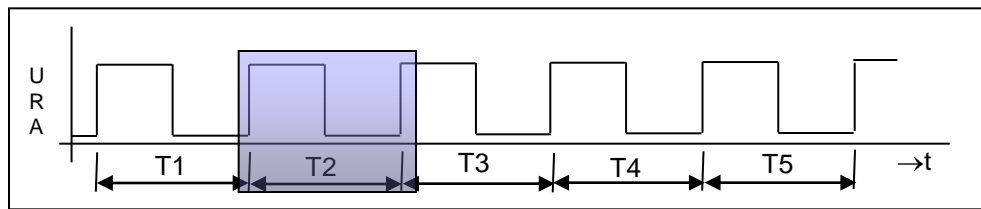
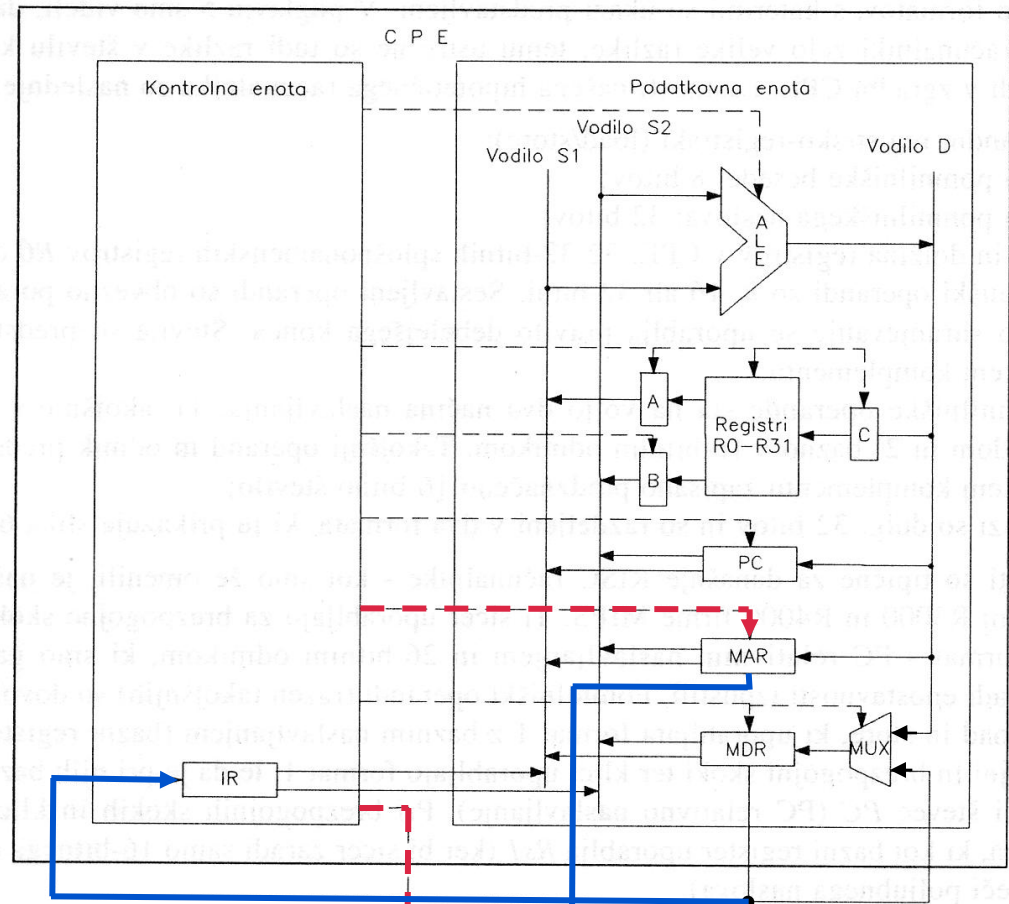
Pomnilnik



## 2. urina perioda T2

$$IR \leftarrow m[MAR]$$

Vsebinsko na tem naslovu  
CPE obravnava kot  
ukaz in se zato prenese  
v ukazni register IR  
kontrolne enote







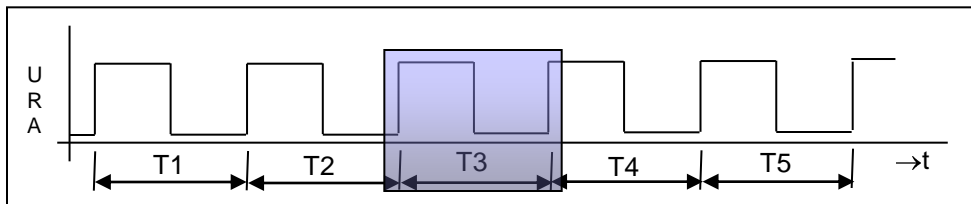
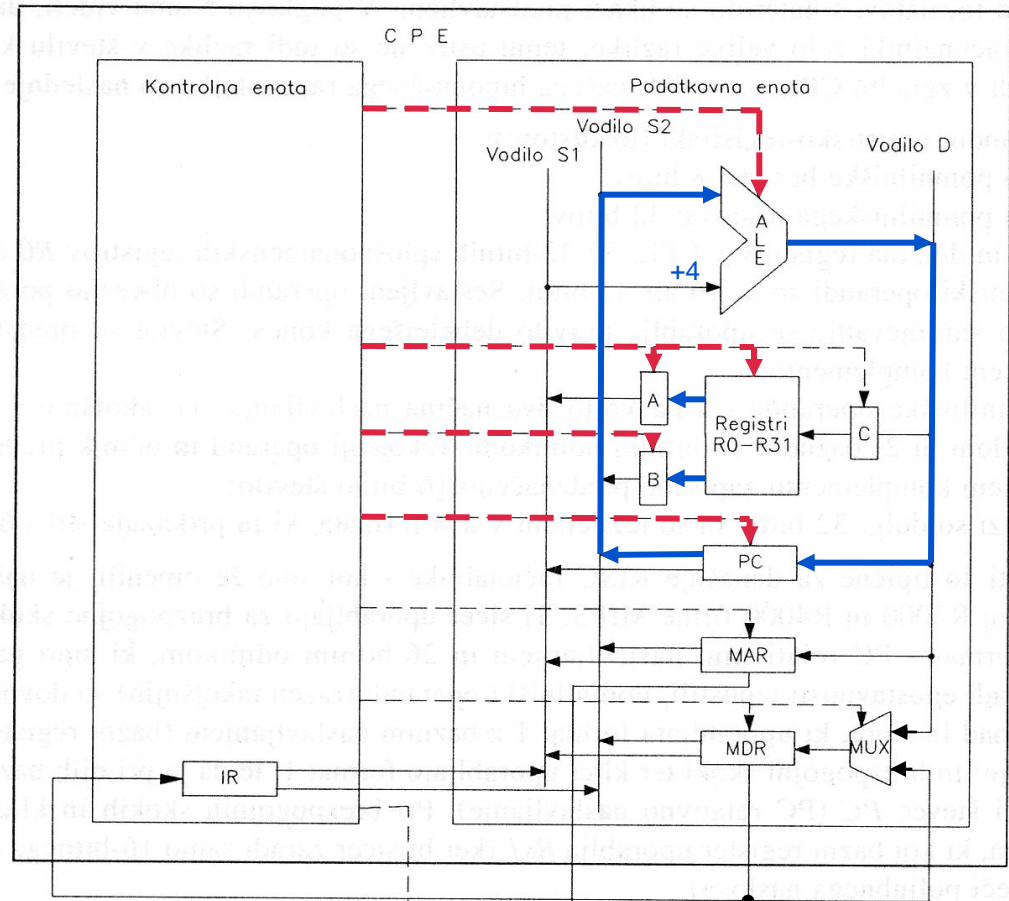
### 3. urina perioda T3

$A \leftarrow R_{s1}$  oz.  $A \leftarrow R1$   
 $B \leftarrow R_{s2}$  oz.  $B \leftarrow R3$   
 $PC \leftarrow PC + 4$

Dekodiranje ukaza  
 (op. kode)

Prenos operandov v  
 vmesna registra A in B

Povečanje vsebine PC



Kontrolni signali

Naslovni signali

Podatkovni signali

P o m n i l n i k

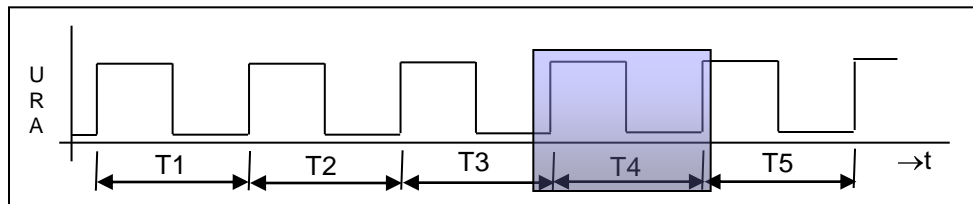
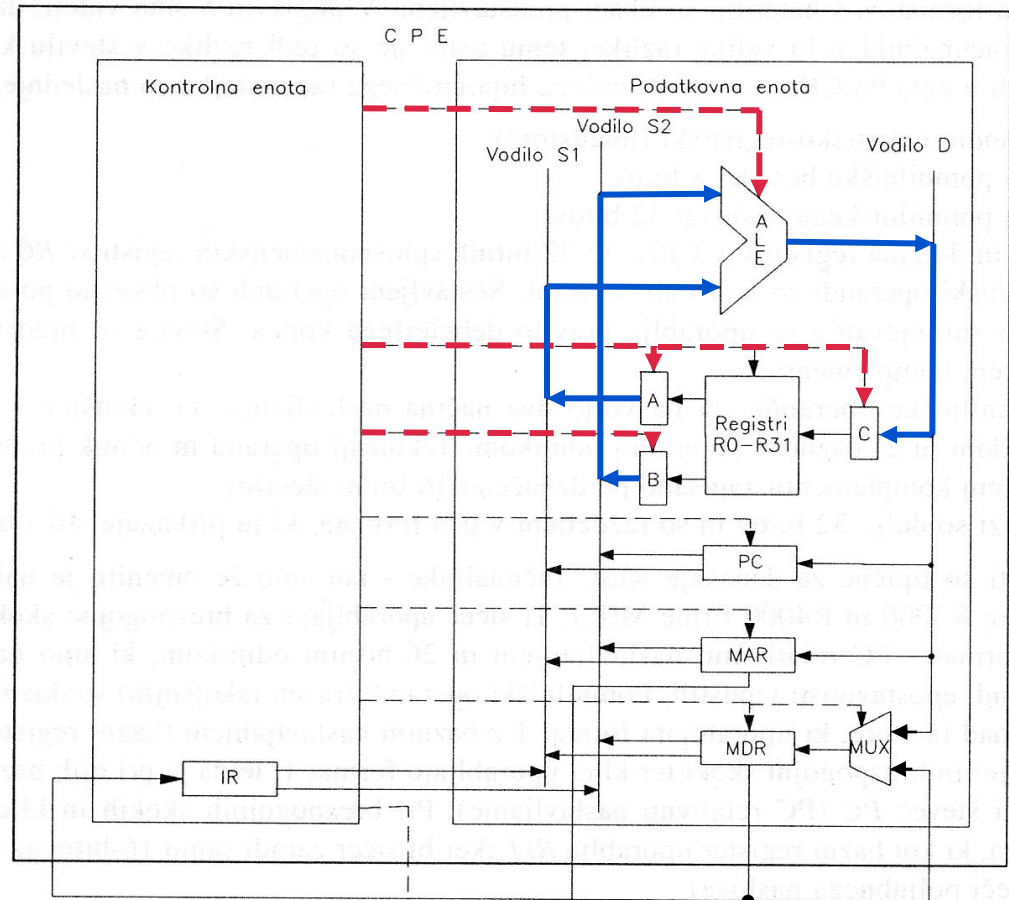


# Centralna procesna enota - izvajanje ukazov

## 4. urina perioda T4

$$C \leftarrow A + B$$

Izvrševanje operacije seštevanja



Kontrolni signali

Naslovni signali

Podatkovni signali

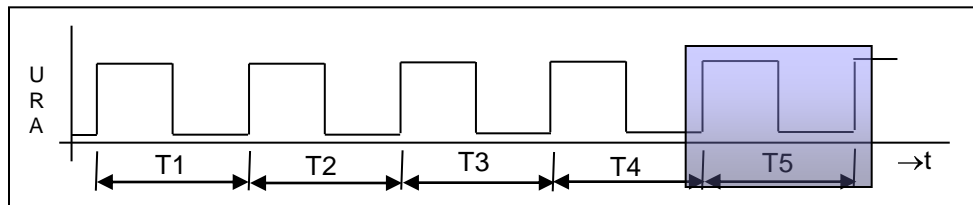
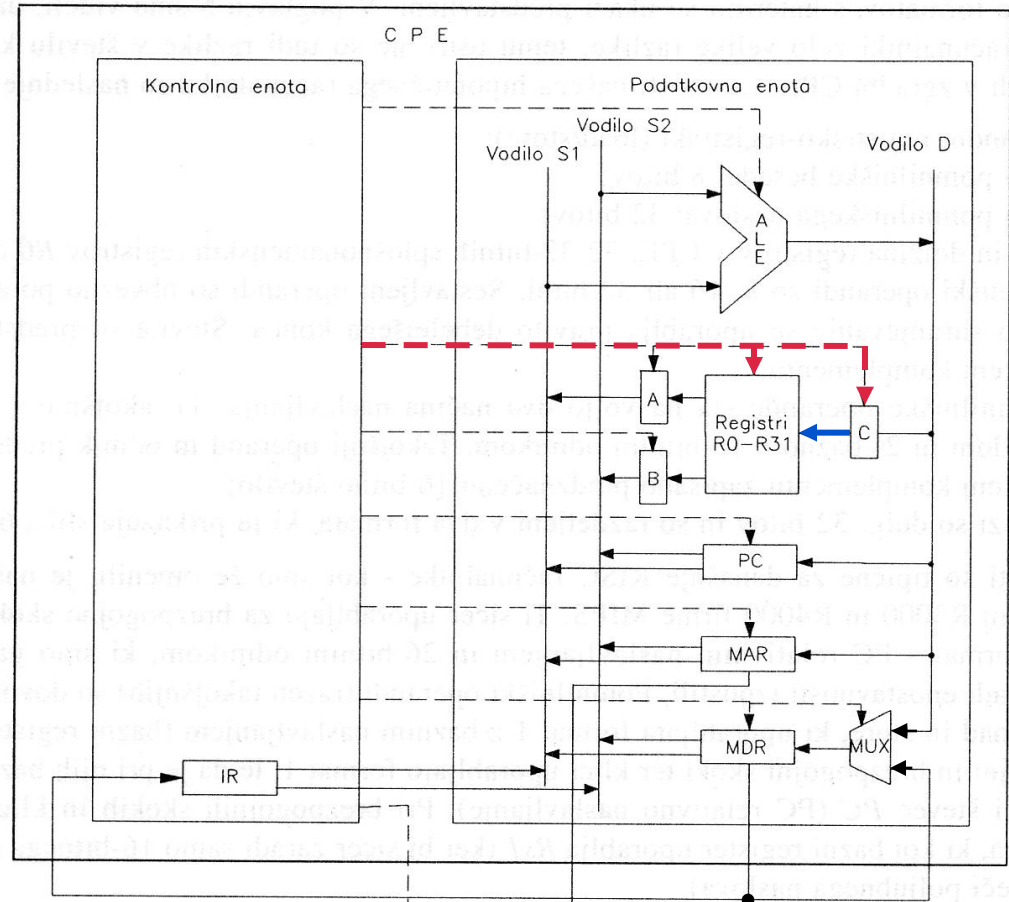
P o m n i l n i k



## 5. urina perioda T5

$$R_d \leftarrow C \text{ oz. } R_{10} \leftarrow C$$

Rezultat se iz vmesnega registra C shrani v register  $R_d$  ( $R_{10}$ )



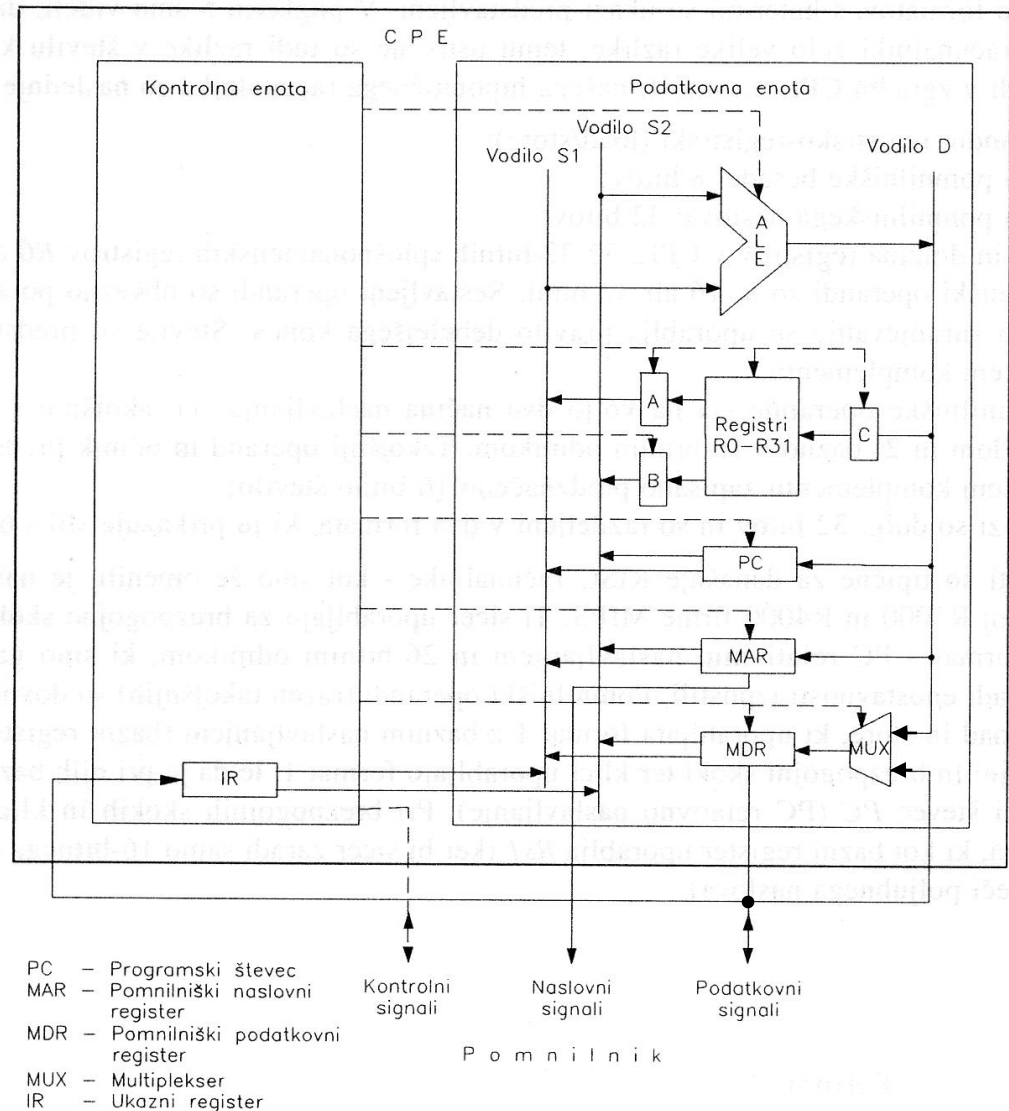
Kontrolni signali  
Naslovni signali  
Podatkovni signali

P o m n i l n i k



## Centralna procesna enota - izvajanje ukazov

- Ta rešitev ni edina možna.
- Ukaz ADD je mogoče realizirati tudi drugače, z več ali pa tudi manj urinimi periodami.
- Pri določenih ukazih lahko kakšen korak ni potreben.
- Čakalne periode pri dostopu do pomnilnika niso potrebne, če je informacija v predpomnilniku.





## 6.5 Paralelno izvajanje ukazov

- Običajna zgradba CPE – izvajanje strojnega ukaza traja najmanj 3 ali 4 urine periode, običajno pa precej več.
- Povprečno število ukazov, ki jih CPE izvede v eni sekundi (*IPS* – *Instructions Per Second*):

$$IPS = \frac{f_{CPE}}{CPI}$$

IPS je zelo veliko število, zato ga delimo z  $10^6$  in dobimo MIPS

$$MIPS = \frac{f_{CPE}}{CPI \cdot 10^6}$$

MIPS = Million Instructions per Second

$f_{CPE}$  = frekvenca CPE ure

CPI = Cycles per Instruction  
(povprečno število urinih period  
za izvedbo enega ukaza)



- Število ukazov, ki jih CPE izvede v eni sekundi, lahko povečamo:

$$\uparrow MIPS = \frac{\uparrow f_{CPE}}{\downarrow CPI \cdot 10^6}$$

- Z uporabo hitrejših elementov (povečanje  $f_{CPE}$  = več urinih period v sekundi)
- Z uporabo večjega števila elementov (zmanjšanje  $CPI$  = manj urinih period za izvedbo enega ukaza)
- Uporaba hitrejših elementov ne omogoča velikega povečanja hitrosti, s seboj pa prinaša še druge probleme.



- Če želimo učinkovito povečati hitrost CPE, moramo CPE narediti tako, da paralelno izvaja več funkcij, to pa pomeni povečanje števila logičnih elementov.
- Če izkoristimo možnost, da se nekateri ukazi v programu lahko izvršijo paralelno, imenujemo to **paralelizem na nivoju ukazov**.
- Največkrat uporabljena rešitev za izkoriščanje paralelizma na nivoju ukazov je izvedba CPE v obliki **cevovoda (pipeline)**.



- Prednost paralelizma na nivoju ukazov je, da niso potrebne spremembe v programih. Vendar v mnogih programih obstaja paralelizem tudi na višjih nivojih.
  
- Tak višje-nivojski paralelizem imenujemo **paralelizem na nivoju niti**.
  - Večnost
  
  - Večjedrni procesorji

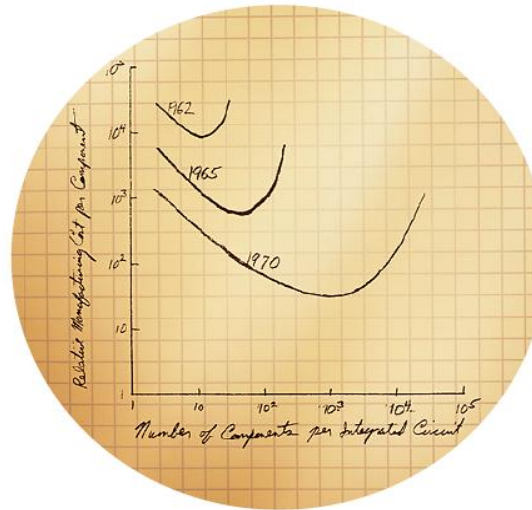




- Revija Electronic Magazine je leta 1965 objavila članek Gordona E. Moora, v katerem je napovedal, da se bo število tranzistorjev, ki so jih proizvajalci sposobni izdelati na čipu, podvojilo vsako leto.
- Leta 1975 je napoved popravil, da se bo število tranzistorjev podvojilo na vsaki dve leti.
- Kar je bilo takrat mišljeno kot izkustveno pravilo in naj bi veljalo naslednjih nekaj let, velja še danes in je poznano kot Moorov zakon.



# Moore's Law



**In 1965, Gordon Moore sketched out his prediction of the pace of silicon technology. Decades later, Moore's Law remains true, driven largely by Intel's unparalleled silicon expertise.**

According to Moore's Law, the number of transistors on a chip roughly doubles every two years. As a result the scale gets smaller and smaller. For decades, Intel has met this formidable challenge through investments in technology and manufacturing resulting in the unparalleled silicon expertise that has made Moore's Law a reality.



- Gordon E. Moore je danes častni predsednik Intela, v letu 1968 pa je bil soustanovitelj in izvršni podpredsednik Intela.
- Pri isti tehnologiji se je v zadnjih 20 letih najvišja hitrost logičnih elementov povečala za približno 10-krat.
- V istem času se je največje število elementov na enem čipu povečalo za približno 500 do celo 5000-krat pri pomnilniških čipih.



## Cevovodna CPE

Microprocessor	Year of Introduction	Transistors
4004	1971	2,300
8008	1972	2,500
8080	1974	4,500
8086	1978	29,000
Intel286	1982	134,000
Intel386™ processor	1985	275,000
Intel486™ processor	1989	1,200,000
Intel® Pentium® processor	1993	3,100,000
Intel® Pentium® II processor	1997	7,500,000
Intel® Pentium® III processor	1999	9,500,000
Intel® Pentium® 4 processor	2000	42,000,000
Intel® Itanium® processor	2001	25,000,000
Intel® Itanium® 2 processor	2003	220,000,000
Intel® Itanium® 2 processor (9MB cache)	2004	592,000,000

Intel Core i7 (Sandy Bridge) 1,16 milijarde (= 1.160.000.000) tranzistorjev

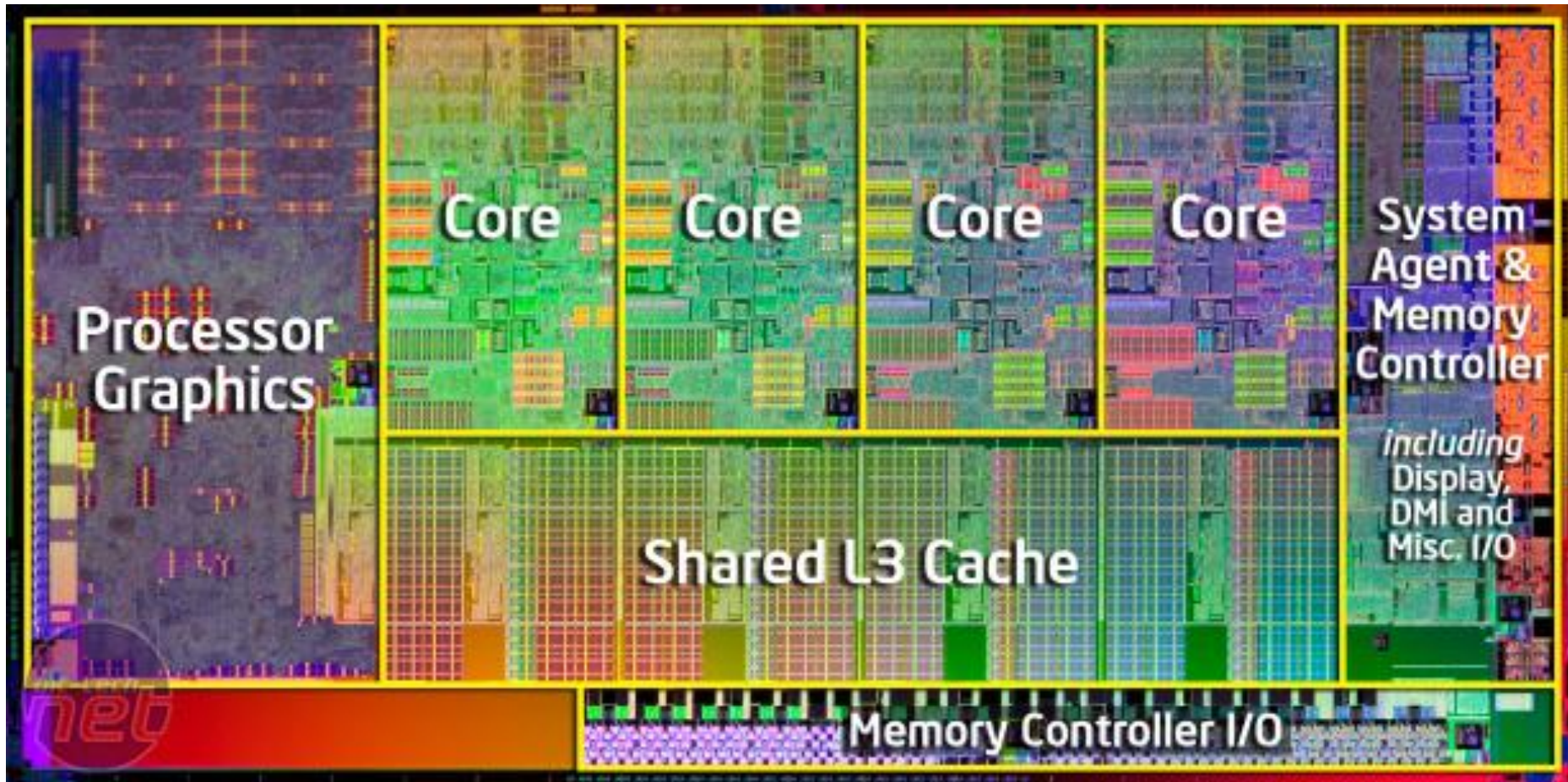


## ■ Intel Core i7 Sandy Bridge

- Feature size 32 nm ( =  $32 * 10^{-9}$  m)
- Število tranzistorjev 1,16 milijarde ( = 1.160.000.000)
- Velikost čipa 216 mm<sup>2</sup> (  $\approx 15 \times 15$  mm)
- Frekvenca ure od 2,8 GHz do 3,5 GHz
- Število jeder (CPE) 4
- Grafični procesor
- TDP (Thermal Design Power) od 65 W do 95 W
- Cena  $\approx 300$  \$

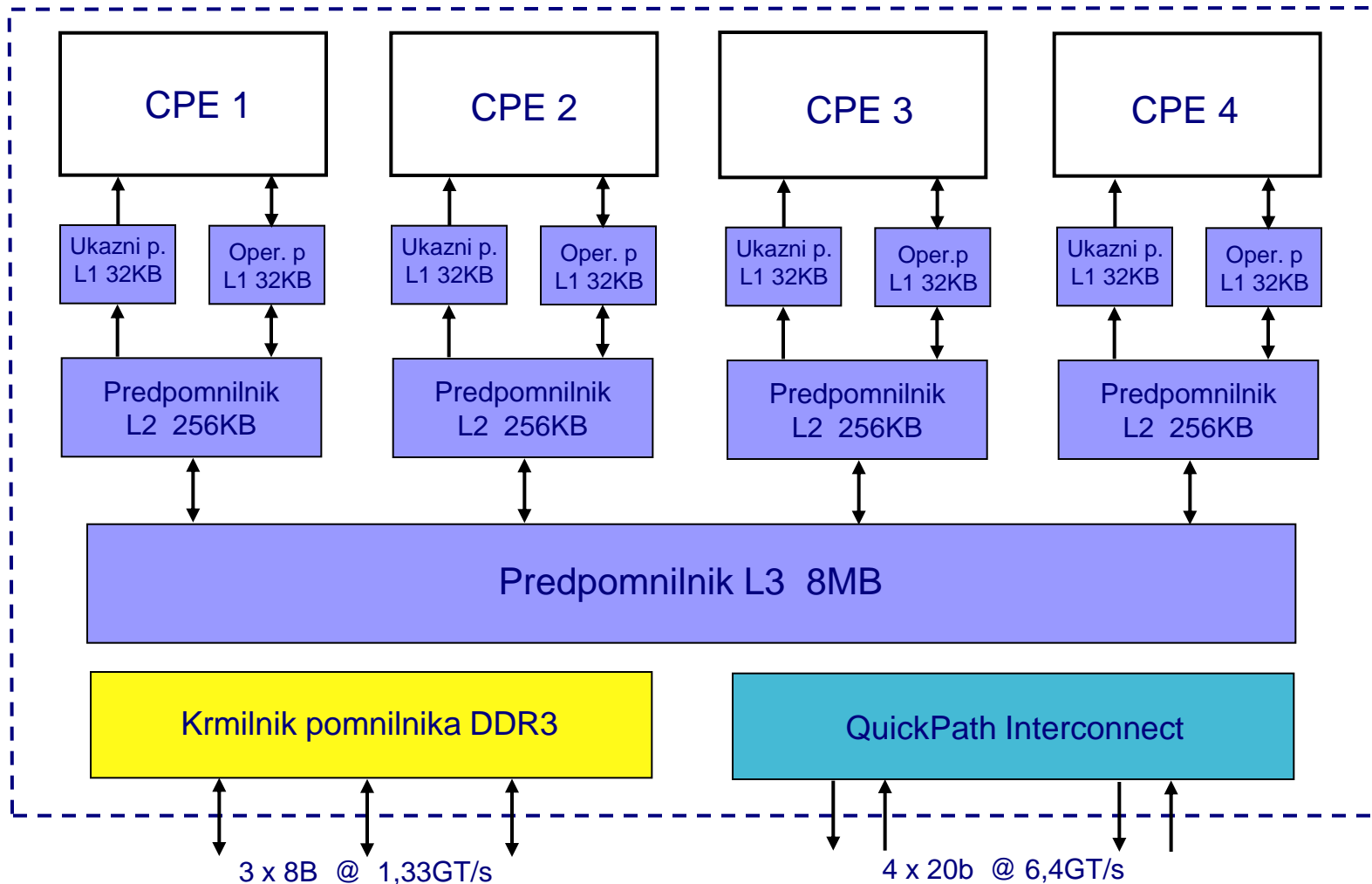


## Intel Core i7 (Sandy Bridge)





## Zgradba 4-jedrnega procesorja Intel Core i7 (Nehalem)





## 6.6 Cevovodna CPE (podatkovna enota)

- Je realizacija CPE, kjer se hkrati izvršuje več ukazov, tako da se posamezni koraki izvrše-vanja ukazov prekrivajo.
- V cevovodni CPE se ukazi izvršujejo podobno tekočemu traku v proizvodnji (npr. avtomobilov).
- Izvrševanje ukaza se razdeli na manjše **podoperacije**, za vsako je potreben samo del celotnega časa, ki je potreben za izvršitev ukaza.





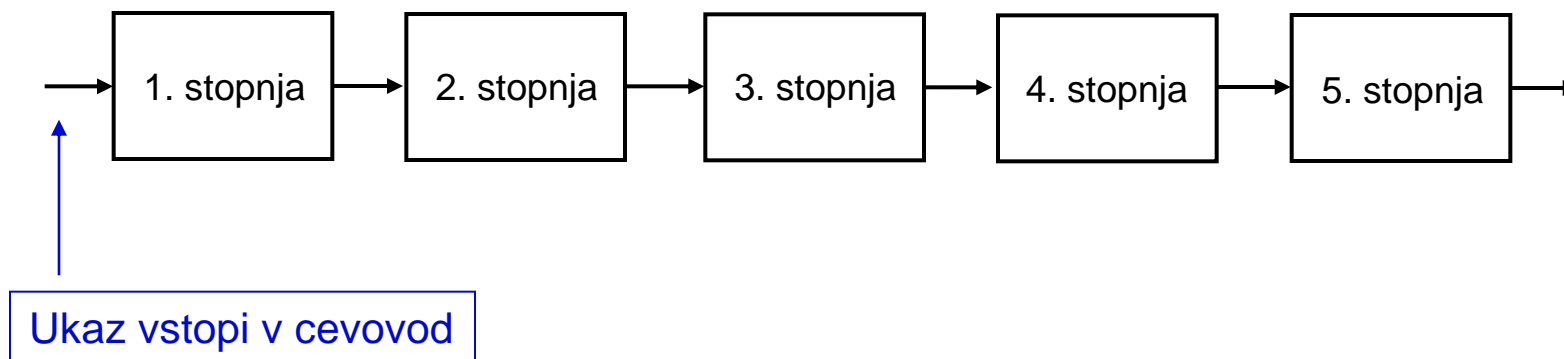
- CPE je razdeljena na **stopnje** ali **segmente cevovoda**, ki jih je toliko kot podoperacij v ukazu.
- Vsako podoperacijo izvrši določena stopnja ali segment cevovoda.
- Stopnje so med seboj povezane, ukazi na eni strani vstopajo, potujejo skozi stopnje, v katerih se izvršujejo posamezne podoperacije ukazov in na drugi strani izstopajo.
- V cevovodu je hkrati v izvrševanju toliko ukazov, kot je stopenj.



- Zmogljivost cevovodne CPE je določena s hitrostjo izstopanja ukazov iz cevovoda.
- Ker so stopnje med seboj povezane, se mora premik ukaza iz ene stopnje v drugo izvršiti pri vseh hkrati.
- Premik se običajno izvede vsako urino periodo.
- Trajanje urine periode  $t_{CPE}$  zato ne more biti krajše kot je čas, ki ga potrebuje za izvedbo podoperacije najpočasnejša stopnja cevovoda.

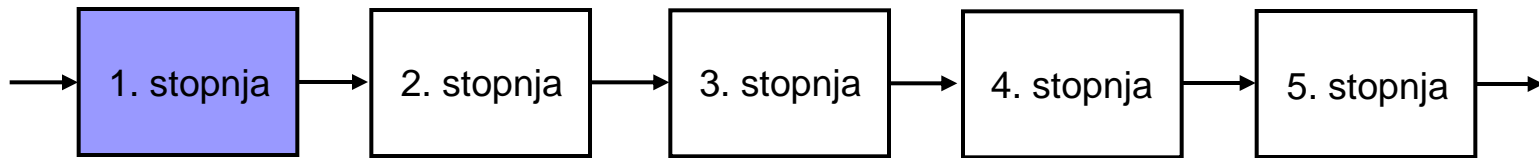


## Primer petstopenjske cevovodne CPE





## Primer petstopenjske cevovodne CPE

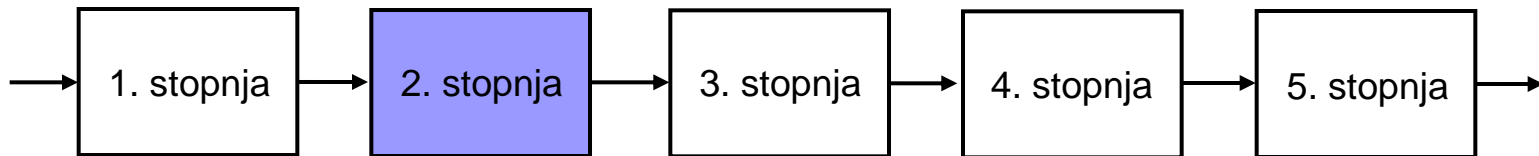


Prezem ukaza  
IF = Instruction Fetch

1. urina perioda



## Primer petstopenjske cevovodne CPE

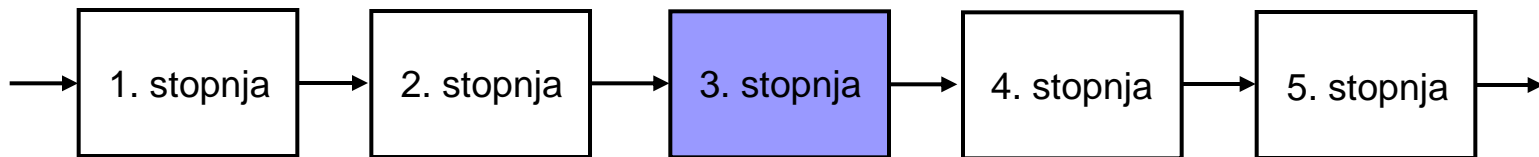


Dekodiranje ukaza in  
dostop do operandov  
v registrih  
ID = Instruction Decode

2. urina perioda



## Primer petstopenjske cevovodne CPE

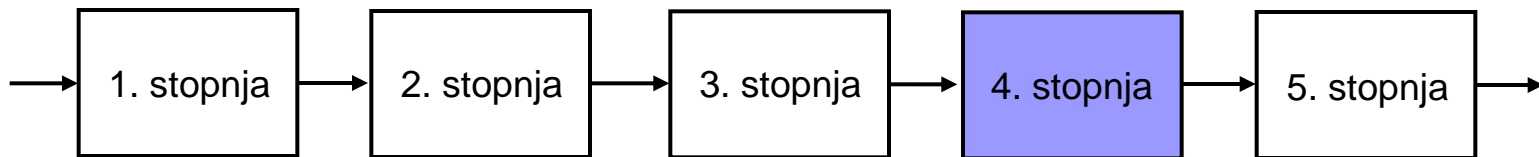


Izvrševanje operacije  
EX = execute

3. urina perioda



## Primer petstopenjske cevovodne CPE

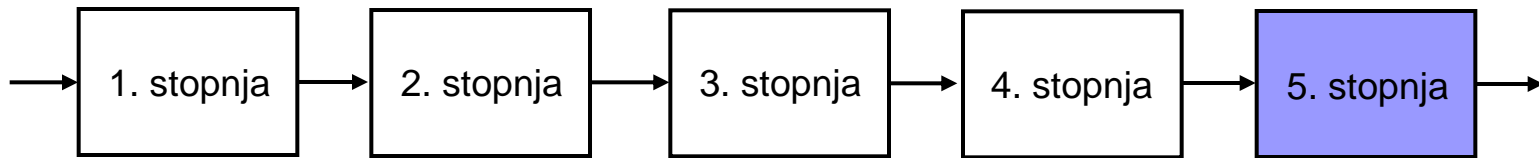


Dostop do operandov  
v pomnilniku (LOAD/STORE)  
MA = Memory Access

4. urina perioda



## Primer petstopenjske cevovodne CPE



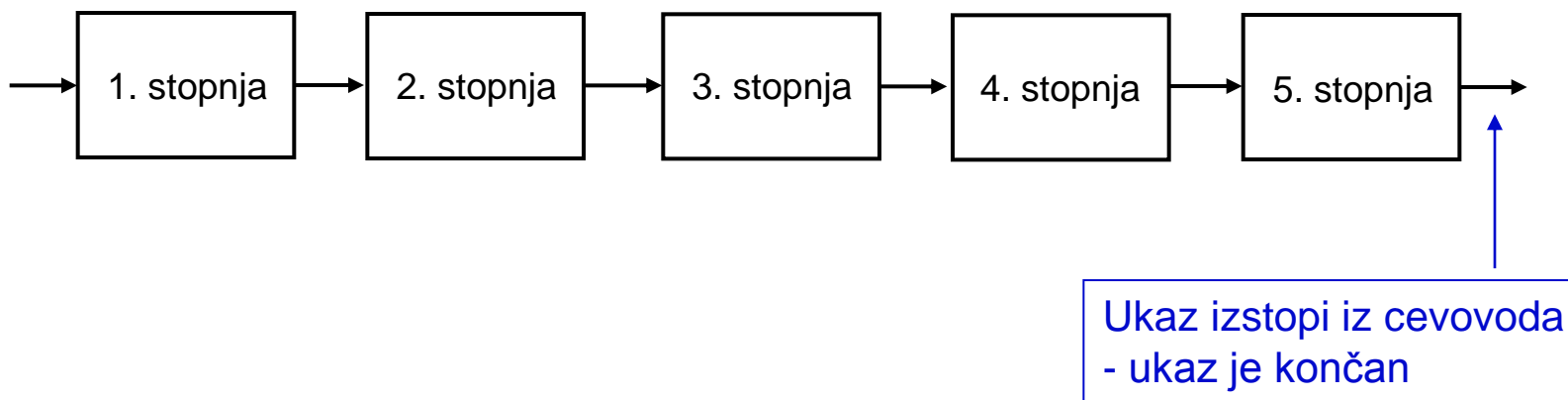
Shranjevanje rezultata  
(v register)  
WB = Write Back

5. urina perioda



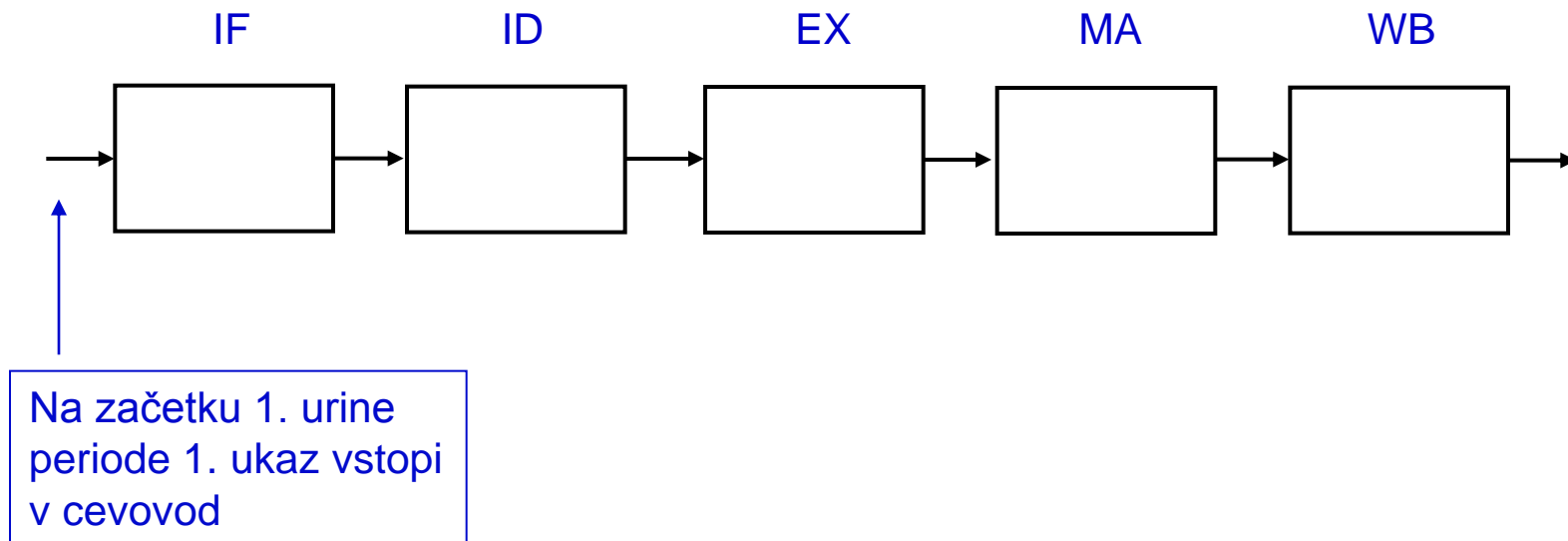


## Primer petstopenjske cevovodne CPE





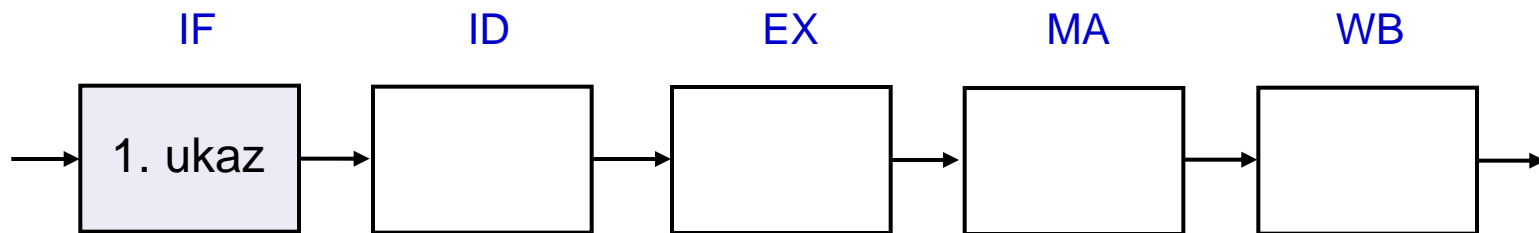
## Primer delovanja petstopenjske cevovodne CPE





# Primer delovanja petstopenjske cevovodne CPE

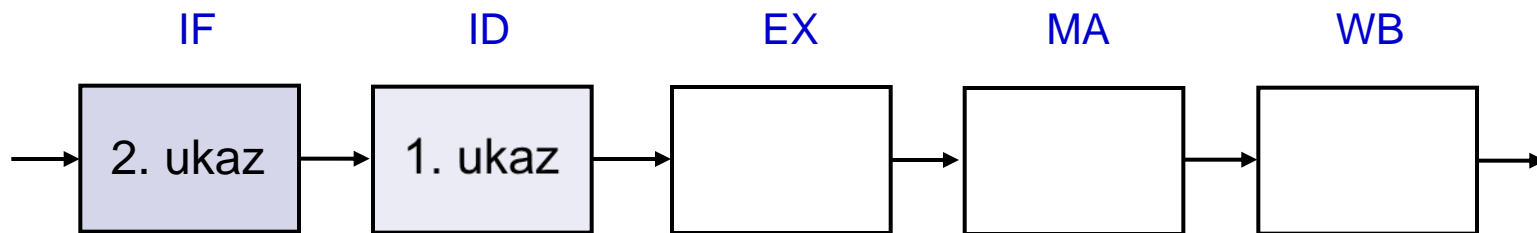
## 1. urina perioda





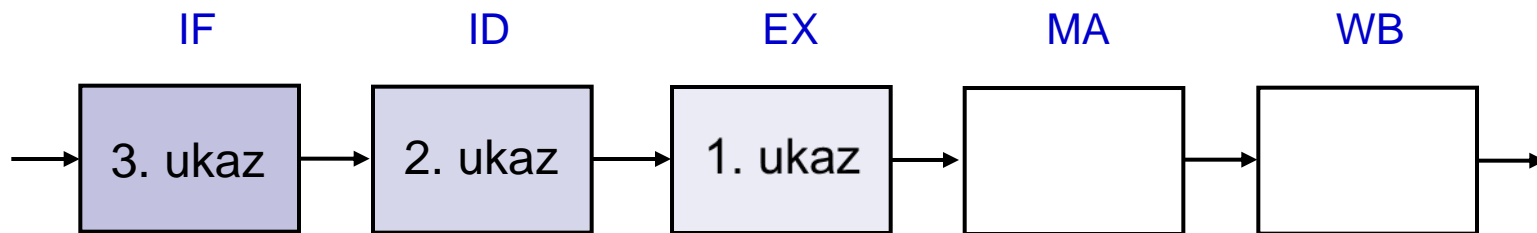
# Primer delovanja petstopenjske cevovodne CPE

## 2. urina perioda



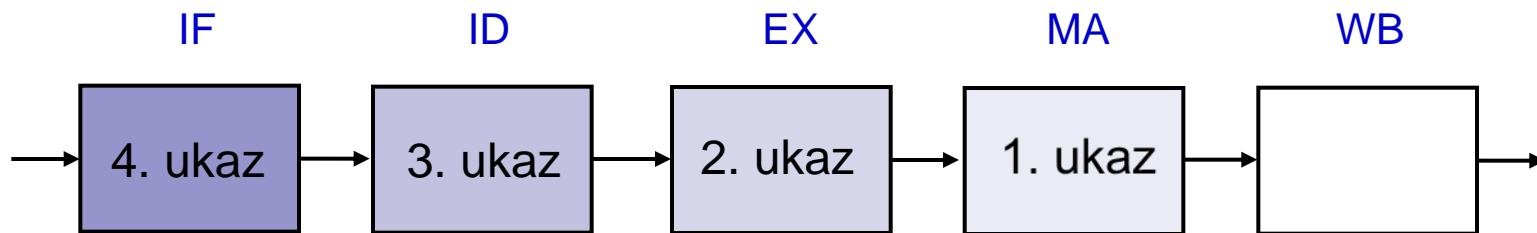


## Primer delovanja petstopenjske cevovodne CPE 3. urina perioda



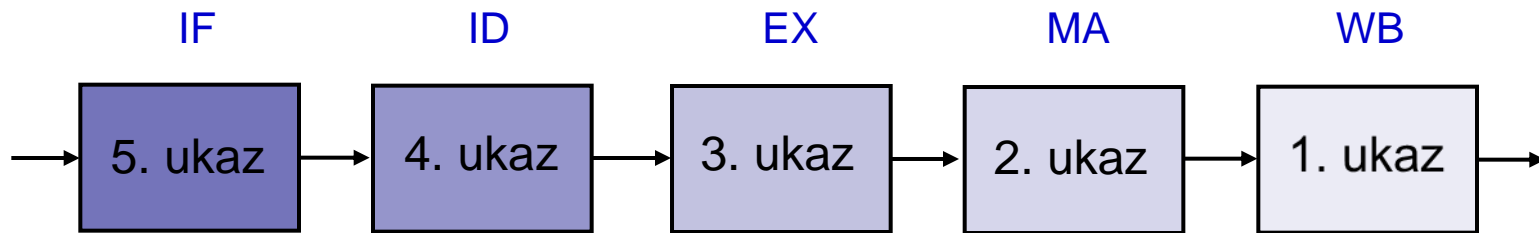


# Primer delovanja petstopenjske cevovodne CPE 4. urina perioda



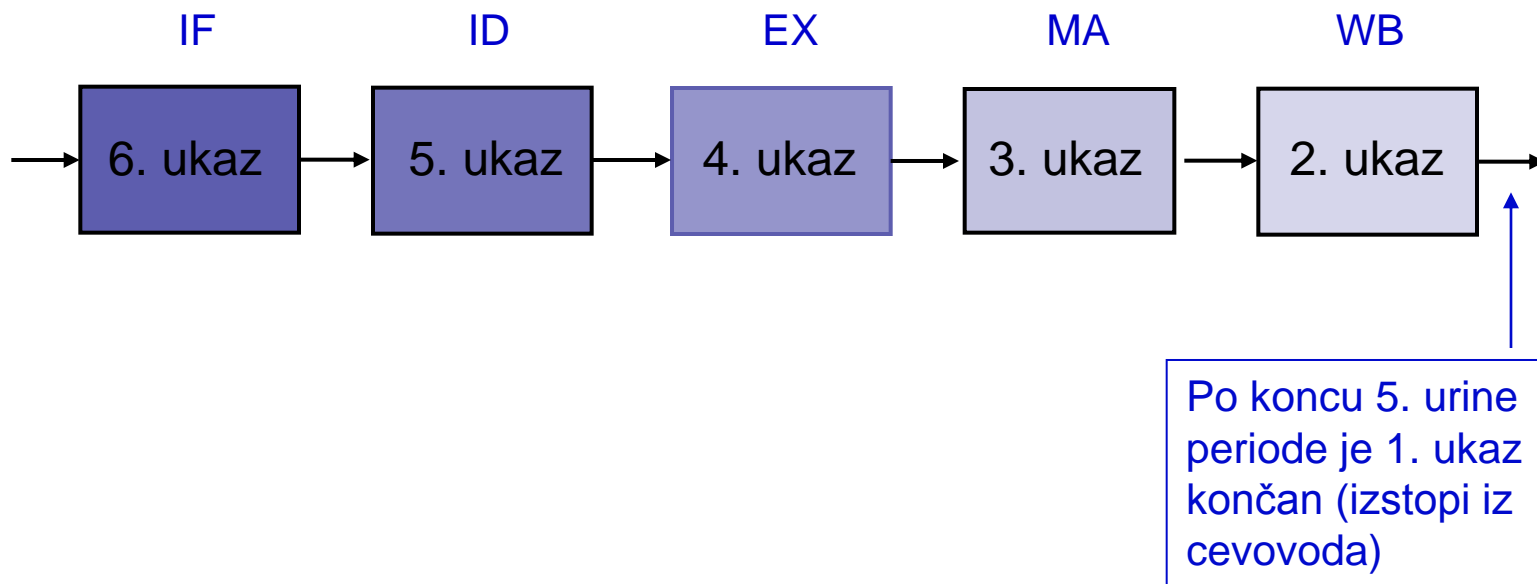


# Primer delovanja petstopenjske cevovodne CPE 5. urina perioda





## Primer delovanja petstopenjske cevovodne CPE 6. urina perioda

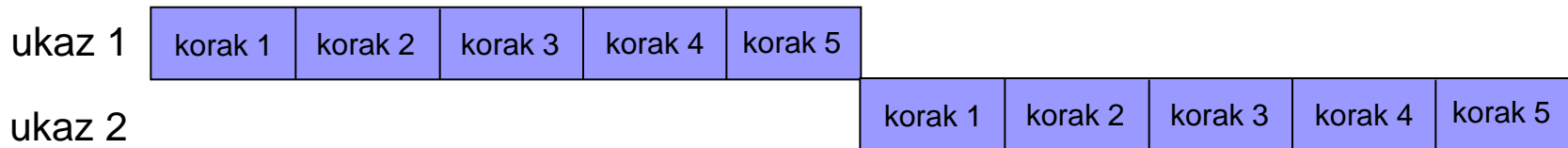
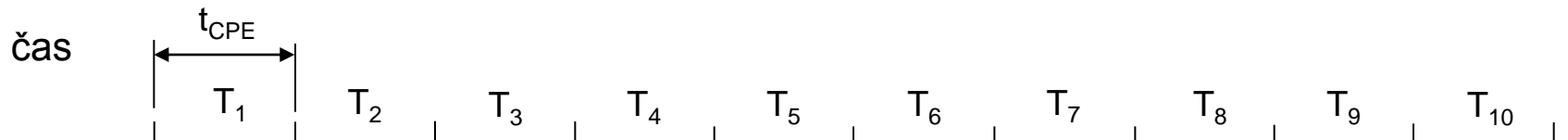




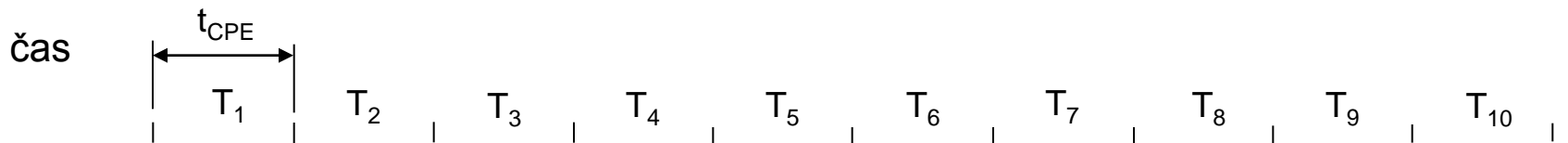


# Izvrševanje ukazov pri necevovodni in cevovodni CPE

## Necevovodna CPE



## Cevovodna CPE





- Danes so vsi zmogljivejši procesorji narejeni kot cevovodni procesorji.
- Pri razvoju cevovodne CPE se stremi za tem, da izvrševanje vseh podoperacij traja približno enako dolgo - uravnovežen cevovod.
- Pri idealno uravnoveženi CPE z  $N$  stopnjami ali segmenti je zmogljivost  $N$ -krat večja kot pri necevovodni CPE.
- Vsak posamezen ukaz se ne izvrši nič hitreje, se pa v cevovodu hkrati izvršuje  $N$  ukazov.



- Na izhodu cevovoda dobimo v enakem času  $N$ -krat več izvršenih ukazov kot pri necevovodni CPE.
- Povprečno število urinih period na ukaz ( $CPI$ ) je v idealnem primeru  $N$ -krat manjše kot pri necevovodni CPE.
- Trajanje izvrševanja posameznega ukaza (latenca) pa je enako  $N \times t_{CPE}$ , torej pri enaki urini periodi enako kot pri necevovodni CPE.



- Ali bi pri dovolj velikem številu stopenj  $N$  lahko naredili poljubno hitro CPE ( $N$ -krat hitrejšo)? Ne.
- Med delovanjem cevovoda prihaja do **cevovodnih nevarnosti** (pipeline hazards).
- Razlikujemo tri vrste cevovodnih nevarnosti:
  - **Strukturne nevarnosti** - kadar več stopenj cevovoda v isti urini periodi potrebuje isto enoto
  - **Podatkovne nevarnosti** - kadar nek ukaz potrebuje rezultat predhodnega ukaza, ki pa še ni končan



- **Kontrolne nevarnosti** - pri ukazih, ki spreminjajo vrednost PC (kontrolni ukazi: skoki, klici, ...)
  
- Zaradi cevovodnih nevarnosti se mora del cevovoda ustaviti dokler nevarnost ne mine (cevovod ne sprejema novih ukazov).
  
- Povečanje hitrosti zato **ni *N*-kratno**.
  
- Z večanjem števila stopenj  $N$  se cevovodne nevarnosti pojavljajo pogosteje in cevovod ni več tako učinkovit.

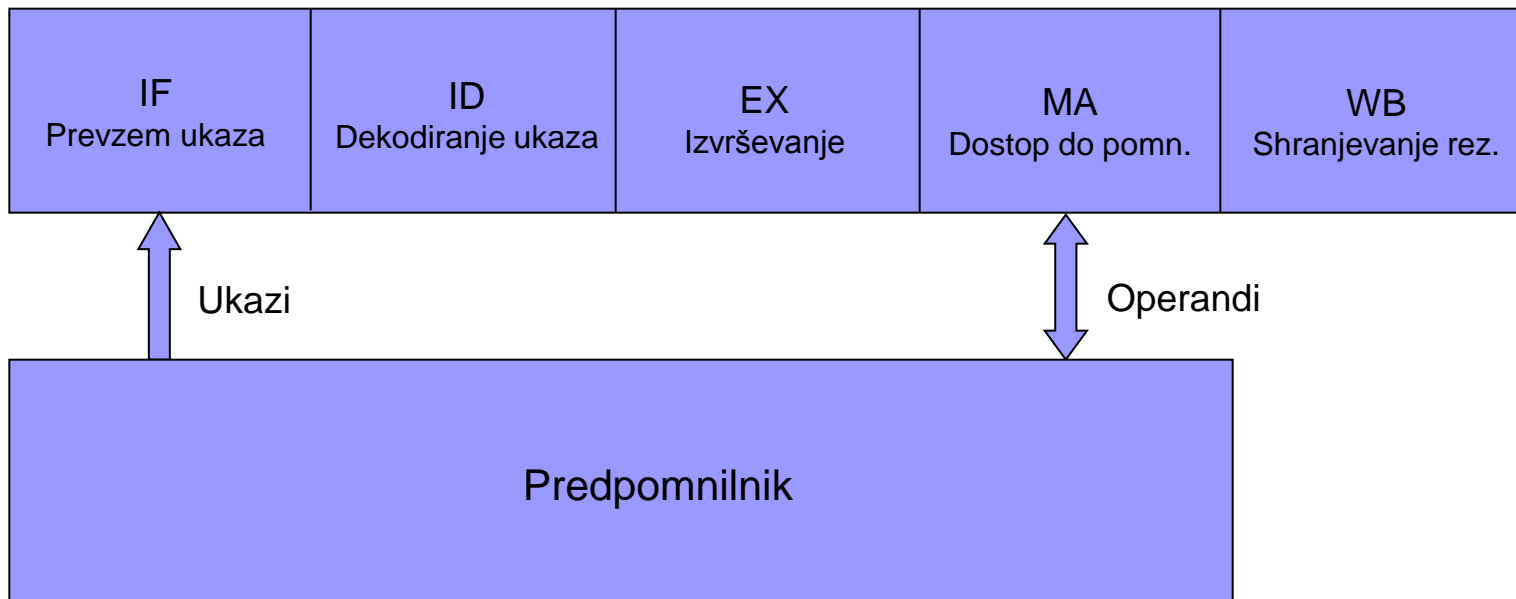


## 6.7 Primer petstopenjske cevovodne CPE

- Osnova naj bo izvajanje ukazov v petih korakih, kot smo ga opisali v prejšnjem poglavju.
- Izvrševanje ukaza razdelimo na pet podoperacij v skladu s koraki iz prejšnjega poglavja, CPE pa v pet stopenj oziroma segmentov:
  - Stopnja IF (Instruction Fetch) - prevzem ukaza
  - Stopnja ID (Instruction Decode) - dekodiranje ukaza in dostop do registrov
  - Stopnja EX (Execute) - izvrševanje operacije



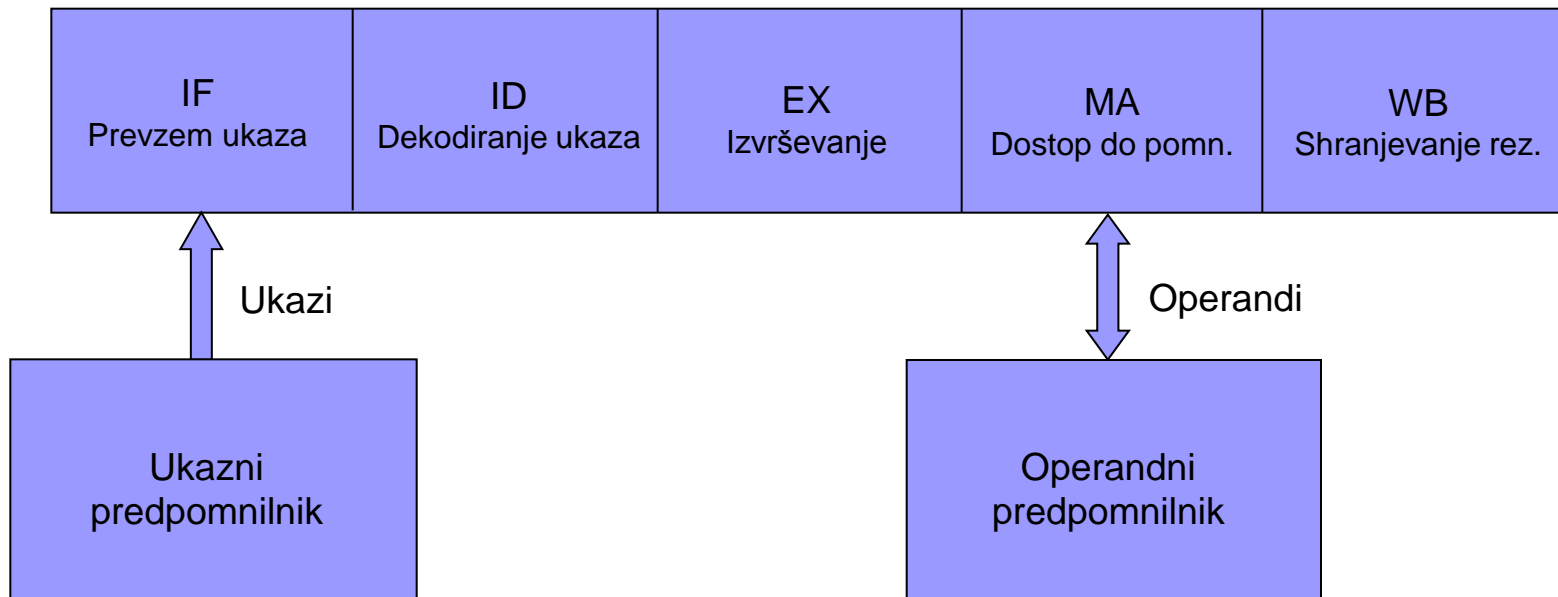
- Stopnja MA (Memory Access) - dostop do pomnilnika
- Stopnja WB (Write Back) - shranjevanje rezultata



Pri hkratnem dostopu do ukaza in operanda v pomnilniku pride do strukturne cevovodne nevarnosti



- Stopnja MA (Memory Access) - dostop do pomnilnika
- Stopnja WB (Write Back) - shranjevanje rezultata



Strukturna nevarnost je odpravljena s Harvardsko arhitekturo predpomnilnika





- Vsaka stopnja cevovoda mora izvršiti svojo podoperacijo v eni urini periodi.
- V stopnjah IF in MA lahko pride do hkratnega dostopa do pomnilnika (v isti urini periodi) - strukturna nevarnost.
- Predpomnilnik moramo zato razdeliti v ukazni in operandni predpomnilnik (Harvardska arhitektura).



- V stopnji IF se dostop do ukaznega predpomnilnika opravi vsako urino periodo, pri necevovodni CPE pa (v našem primeru) samo na vsakih pet urinih period.
- Hitrost prenosa informacij med predpomnilnikom in CPE mora biti zato v našem primeru petkrat večja kot pri necevovodni CPE.
- Pri načrtovanju cevovodne CPE je treba paziti, da od neke enote (register, ALE, ...) ne zahtevamo, da bi v določeni urini periodi morala delati dve različni operaciji.

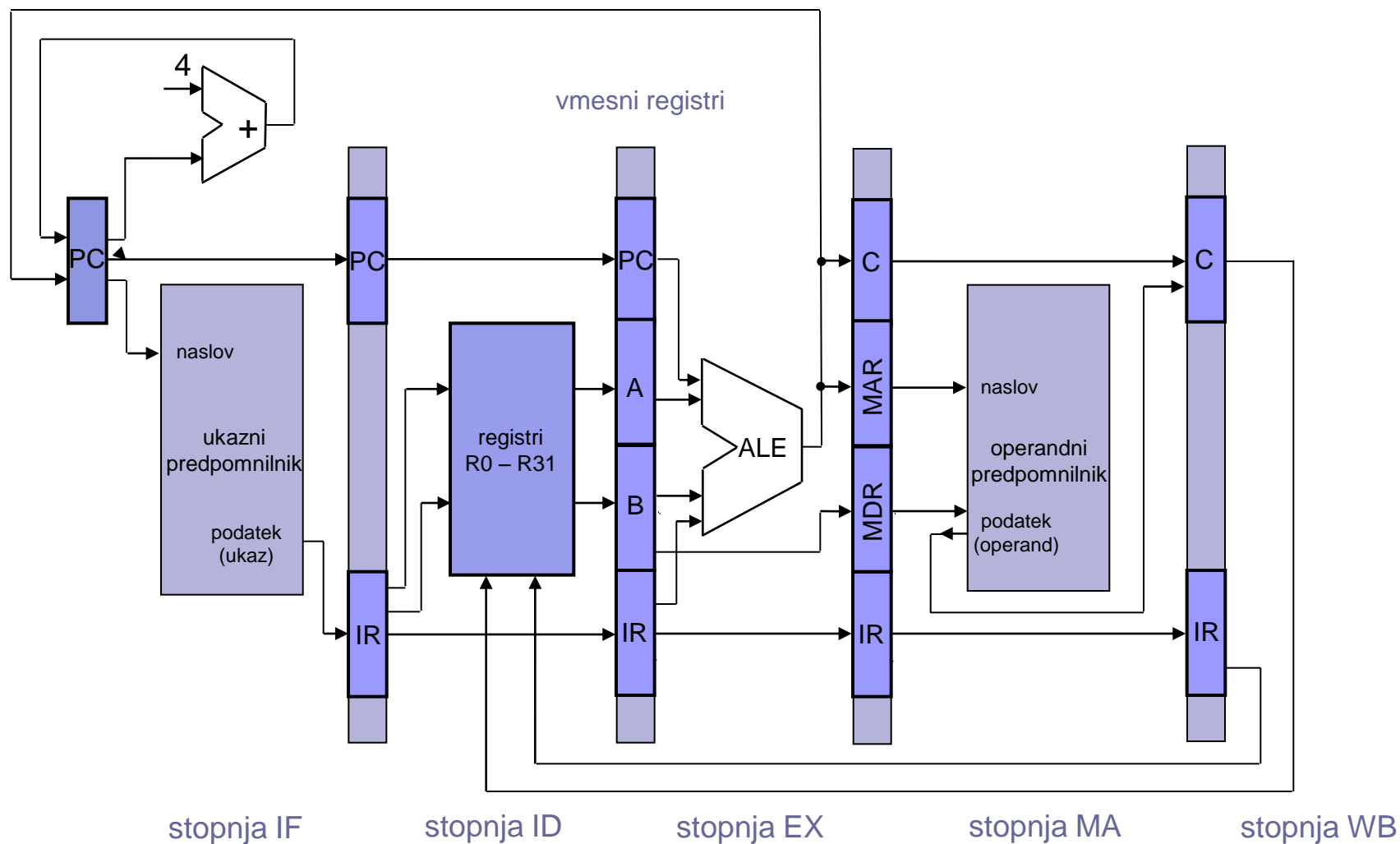


## Primer zgradbe 5-stopenjske cevovodne CPE

- Cevovod ima 5 stopenj, med njimi so vmesni registri, v katere se shranijo rezultati podoperacij vsake stopnje, ki jih potrebujejo druge stopnje.
- V stopnji IF se prebere ukaz, ki se prenese v ukazni register in za 4 poveča vsebina programskega števca PC(ukazi so dolgi 4 bajte).
- Programski števec je potrebno povečati v stopnji IF zato, ker se vsako urino periodo prevzame nov ukaz.



# Primer zgradbe 5-stopenjske cevovodne CPE





- Naslov ukaza, ki se izvaja (vsebina PC), se shranjuje v vmesne registre, ker je pri kontrolnih ukazih potreben v stopnji EX.
- Pri kontrolnih ukazih se v PC namreč vpiše nova vrednost (skočni naslov), ki se izračuna v stopnji EX.
- Vsaka stopnja izvršuje drug ukaz, zato je treba v vmesne registre IR med vsemi stopnjami vedno shranjevati tudi ukaz, ki se v stopnji IF vsako urino periodo prebere iz ukaznega predpomnilnika.



- V tabeli so v vsaki urini periodi napisane stopnje, ki v tej urini periodi delujejo pri določenem ukazu.
- Stopnje, ki v določeni urini periodi niso napisane, v tej urini periodi ne delujejo.

urine periode	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	T <sub>9</sub>
ukaz 1	IF <sub>1</sub>	ID <sub>1</sub>	EX <sub>1</sub>	MA <sub>1</sub>	WB <sub>1</sub>				
ukaz 2		IF <sub>2</sub>	ID <sub>2</sub>	EX <sub>2</sub>	MA <sub>2</sub>	WB <sub>2</sub>			
ukaz 3			IF <sub>3</sub>	ID <sub>3</sub>	EX <sub>3</sub>	MA <sub>3</sub>	WB <sub>3</sub>		
ukaz 4				IF <sub>4</sub>	ID <sub>4</sub>	EX <sub>4</sub>	MA <sub>4</sub>	WB <sub>4</sub>	
ukaz 5					IF <sub>5</sub>	ID <sub>5</sub>	EX <sub>5</sub>	MA <sub>5</sub>	WB <sub>5</sub>



- Vsako urino periodo se prevzame nov ukaz in potuje skozi stopnje cevovoda.
- Če se cevovod ne bi ustavljal zaradi cevovodnih nevarnosti, bi bila njegova zmogljivost 5-krat večja kot pri necevovodni CPE.



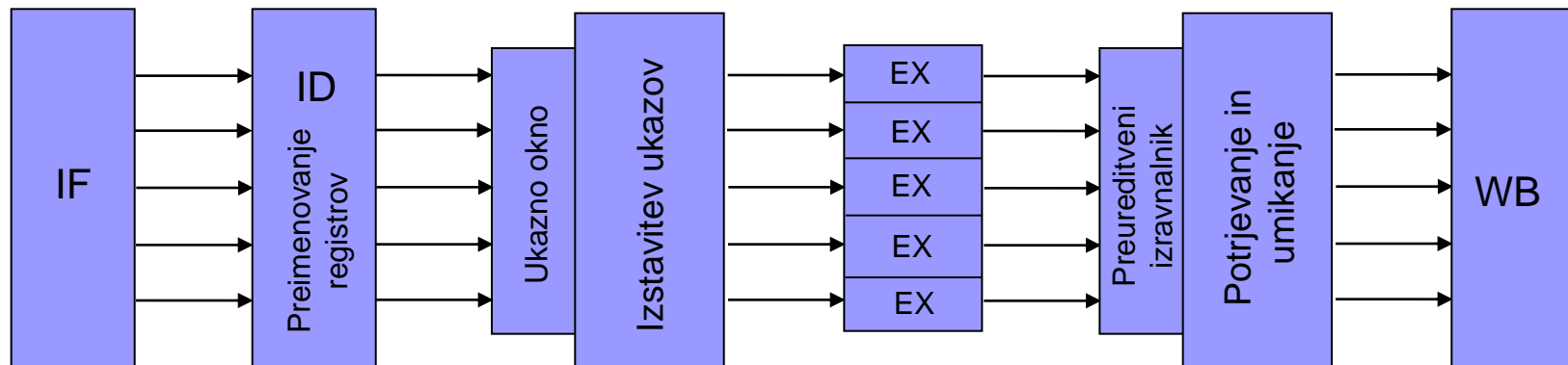
## 6.8 Večizstavitveni procesorji

- S cevovodno CPE in z odpravljanjem cevovodnih nevarnosti lahko dosežemo CPI, ki je blizu 1.
- Če želimo CPI zmanjšati pod 1, moramo v vsaki urini periodi prevzeti več ukazov (in jih tudi izvesti).
- Take procesorje označujemo z izrazom večizstavitveni procesorji in jih delimo na dve vrsti:
  - Superskalarni procesorji – ukaze, ki se paralelno izvajajo, določa logika v procesorju
  - VLIW procesorji – ukaze, ki se paralelno izvajajo, določa program (prevajalnik)





- **Superskalarni procesor** je cevovodni procesor, ki je sposoben hkrati prevzemati, dekodirati in izvrševati več ukazov.
- Število prevzetih in izstavljenih ukazov v eni urini periodi se med izvajanjem programa dinamično spreminja in ga določa logika v procesorju.
- Procesor, ki lahko izstavi največ  $n$  ukazov se imenuje *n-kratni* superskalarni procesor.
- Istočasno delovanje zahteva dodatne vmesnike in dodatne stopnje za potrjevanje (zavezovanje) in umikanje rezultatov.



Peonostavljena shema superskalarnega procesorja,  
ki ima za osnovo petstopenjski cevovod

- Ena od funkcijskih enot v stopnji EX je stopnja MA (funkcijska enota LOAD/STORE ali ločeni funkcijski enoti LOAD in STORE).



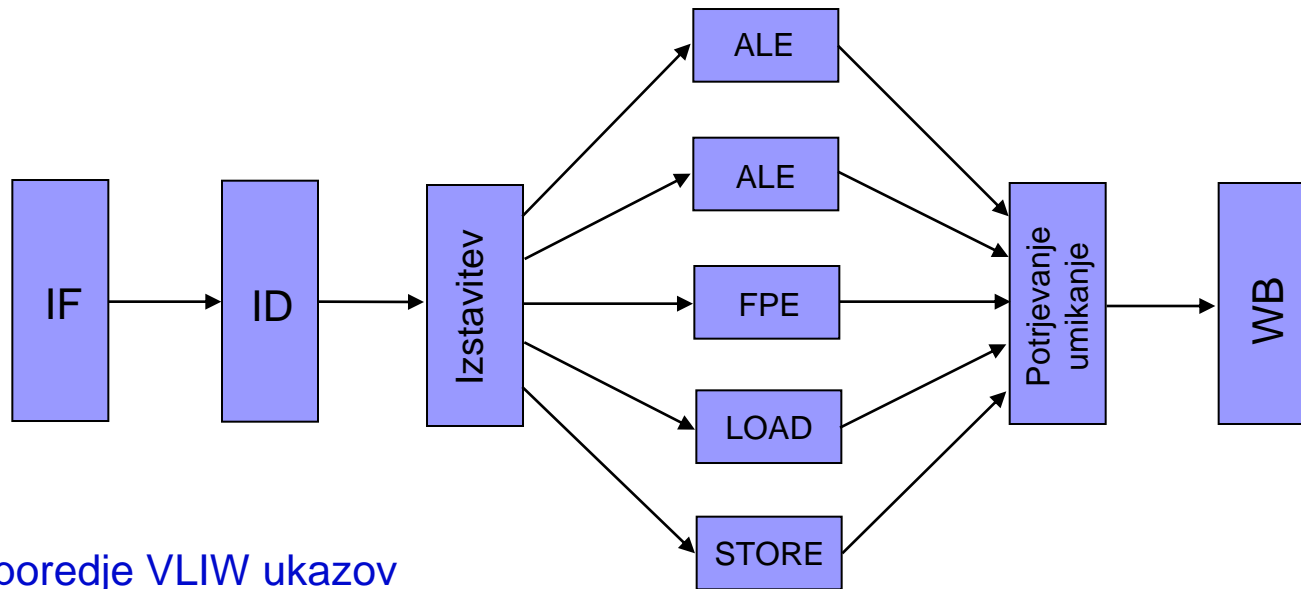
- **VLIW (Very Long Instruction Word)** procesorji izvršujejo dolge ukaze, ki so sestavljeni iz več običajnih strojnih ukazov, ki jih procesor lahko paralelno izvršuje v različnih funkcijskih enotah.
- V dolgem ukazu izvršuje vsaka funkcijska enota svoj ukaz.



VLIW ukaz sestavljajo ukazi za posamezne funkcijske enote



- Prevajalnik v programu išče med seboj neodvisne ukaze za funkcijske enote in z njimi sestavlja dolge ukaze.
- Število ukazov, ki so prevzeti in izstavljeni v eni urini periodi je določeno s programom in se med delovanjem ne spreminja.
- Če prevajalnik ne najde dovolj ukazov za vse funkcijske enote, da enoti, za katero ni našel ukaza, ukaz NOP.



Zaporedje VLIW ukazov



VLIW ukaz

- NOP ukaz

A = ALE ukaz  
 F = FPE ukaz  
 L = LOAD ukaz  
 S = STORE ukaz