

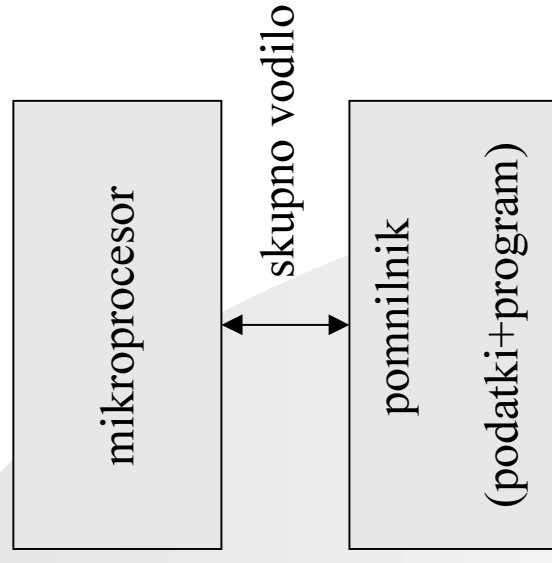
2. Arhitektura mikroprocesorjev

- Model mikroprocesorja
 - ◆ krmilna enota
 - ◆ aritmetično/logična enota (ALE)
 - ◆ Registri
- Delovanje mikroprocesorja
 - ◆ Faze delovanja
 - ◆ Oblika strojnih ukazov
 - ◆ Implementacija krmilne enote

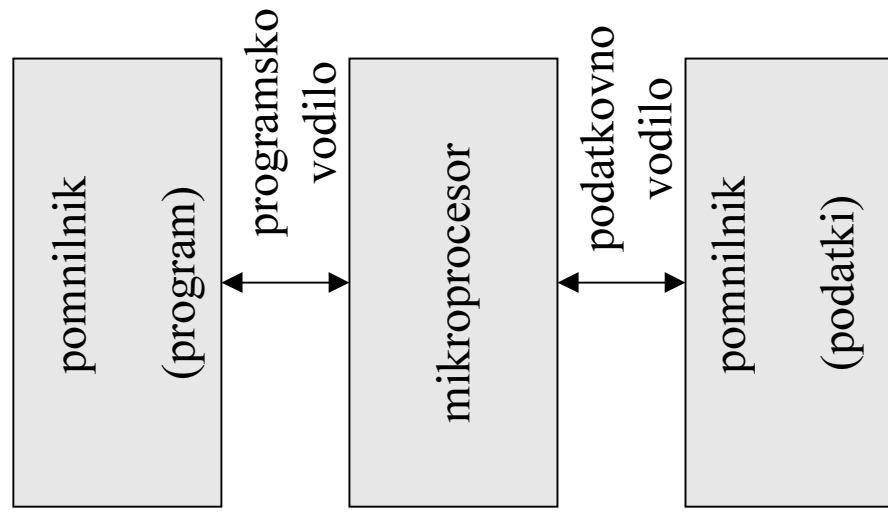
Model mikropcesorja

- Arhitekture mikropcesorjev se zelo razlikujejo
 - ◆ von Neumannov model: podatki in koda v istem pomnilniku (Pentium)
 - ◆ Harwardski model: podatki in koda v ločenih pomnilnikih (PIC)

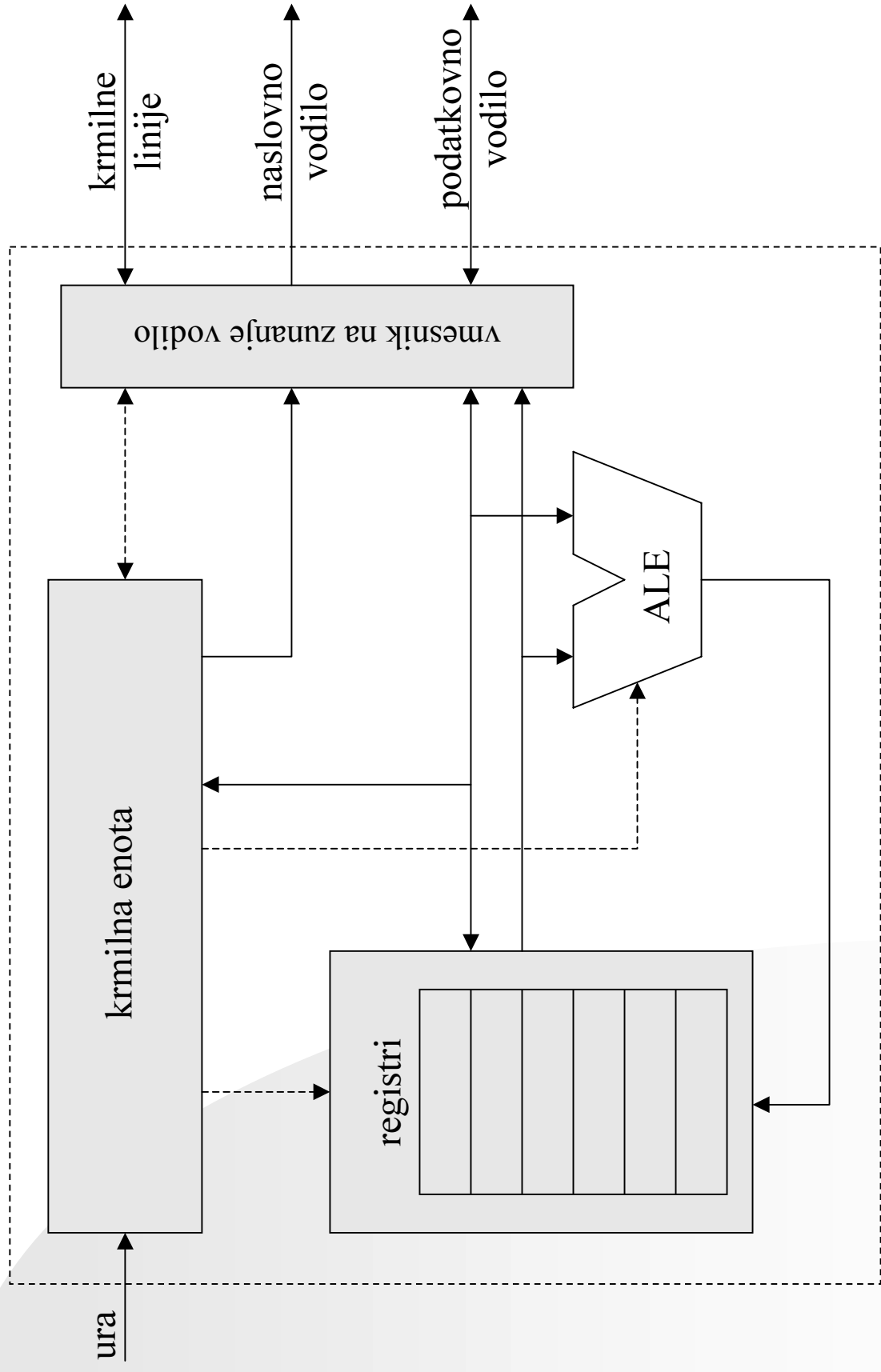
von Neumannov model

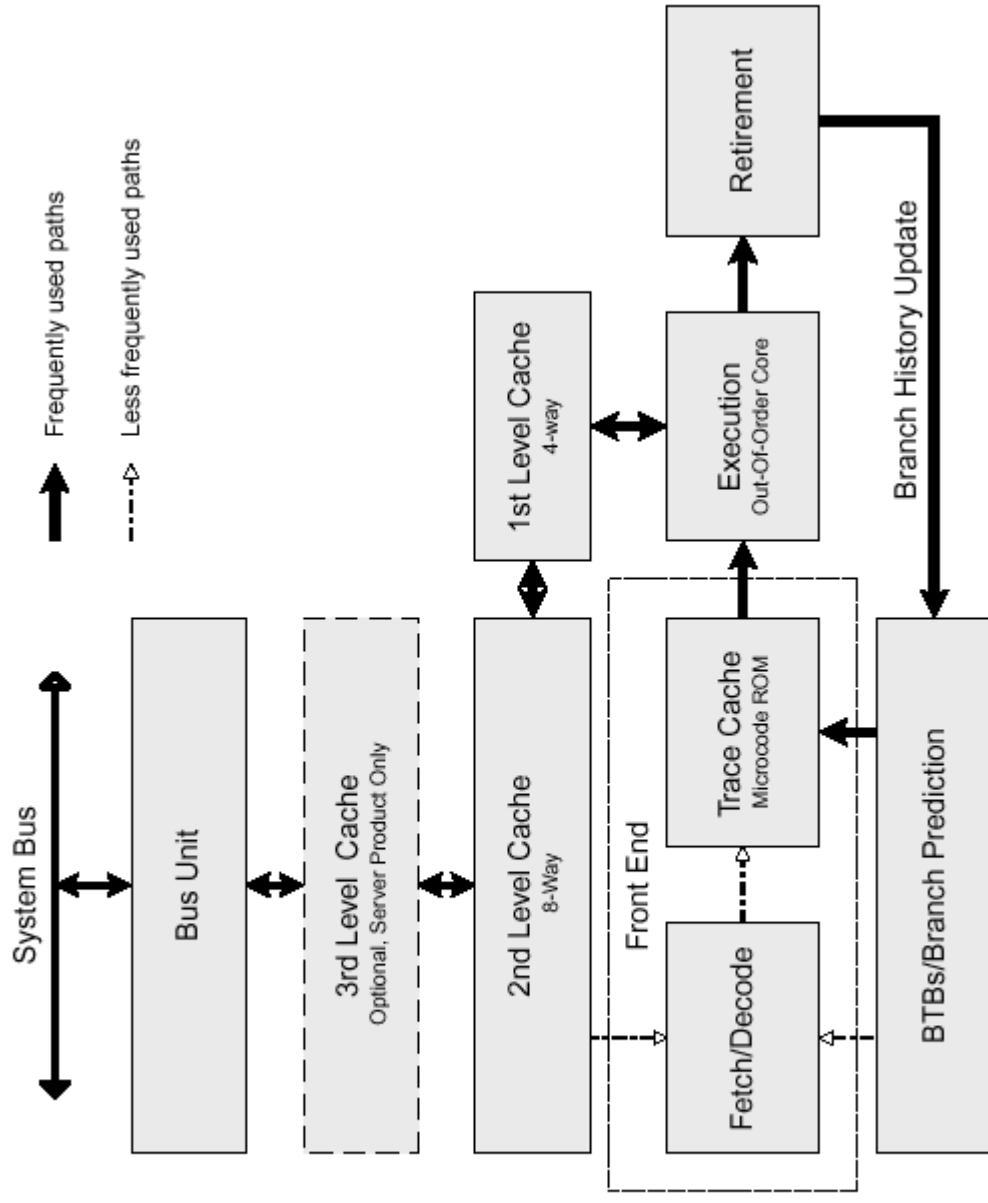


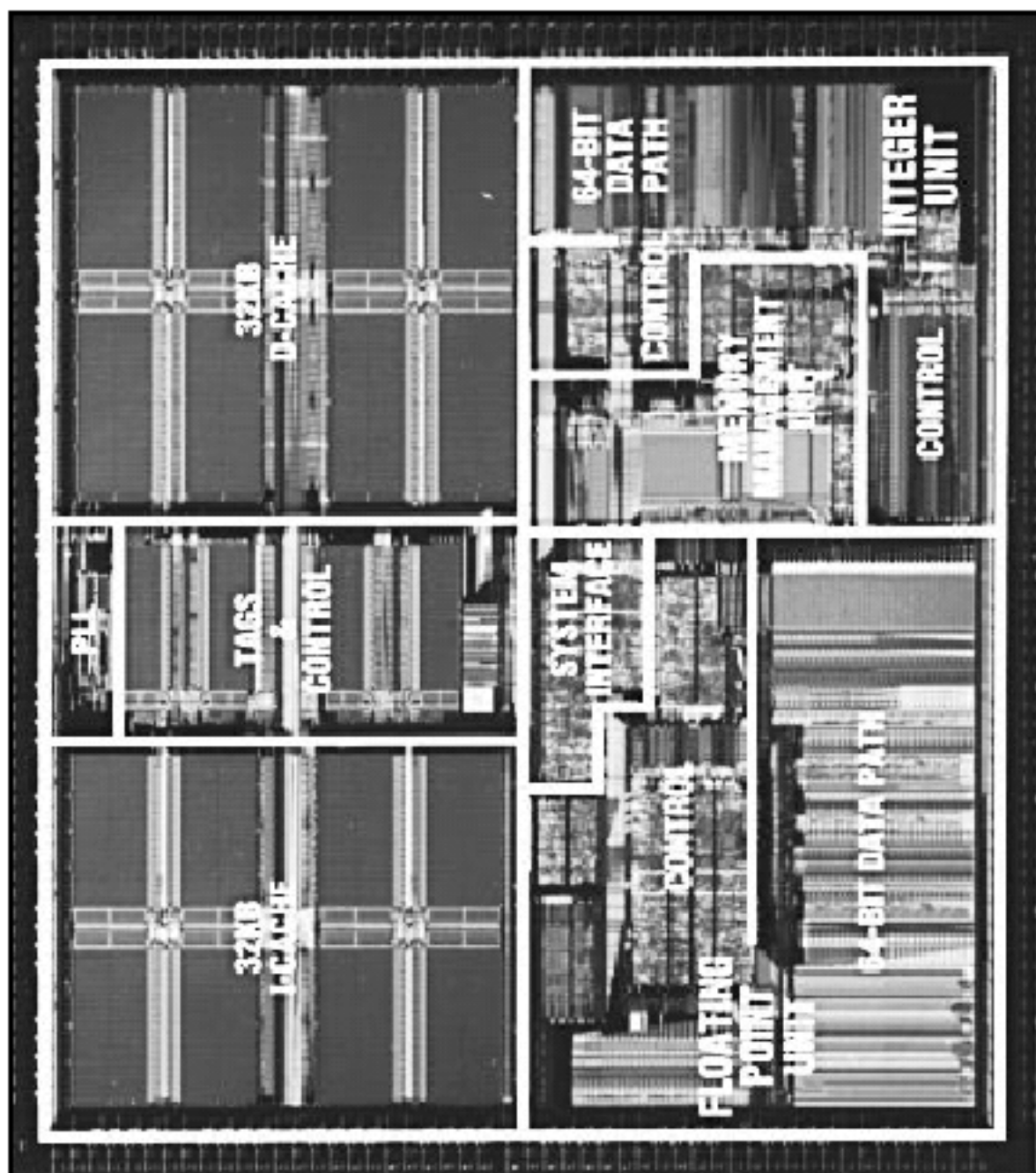
Harwardski model



Hipotetični model mikropcesorja

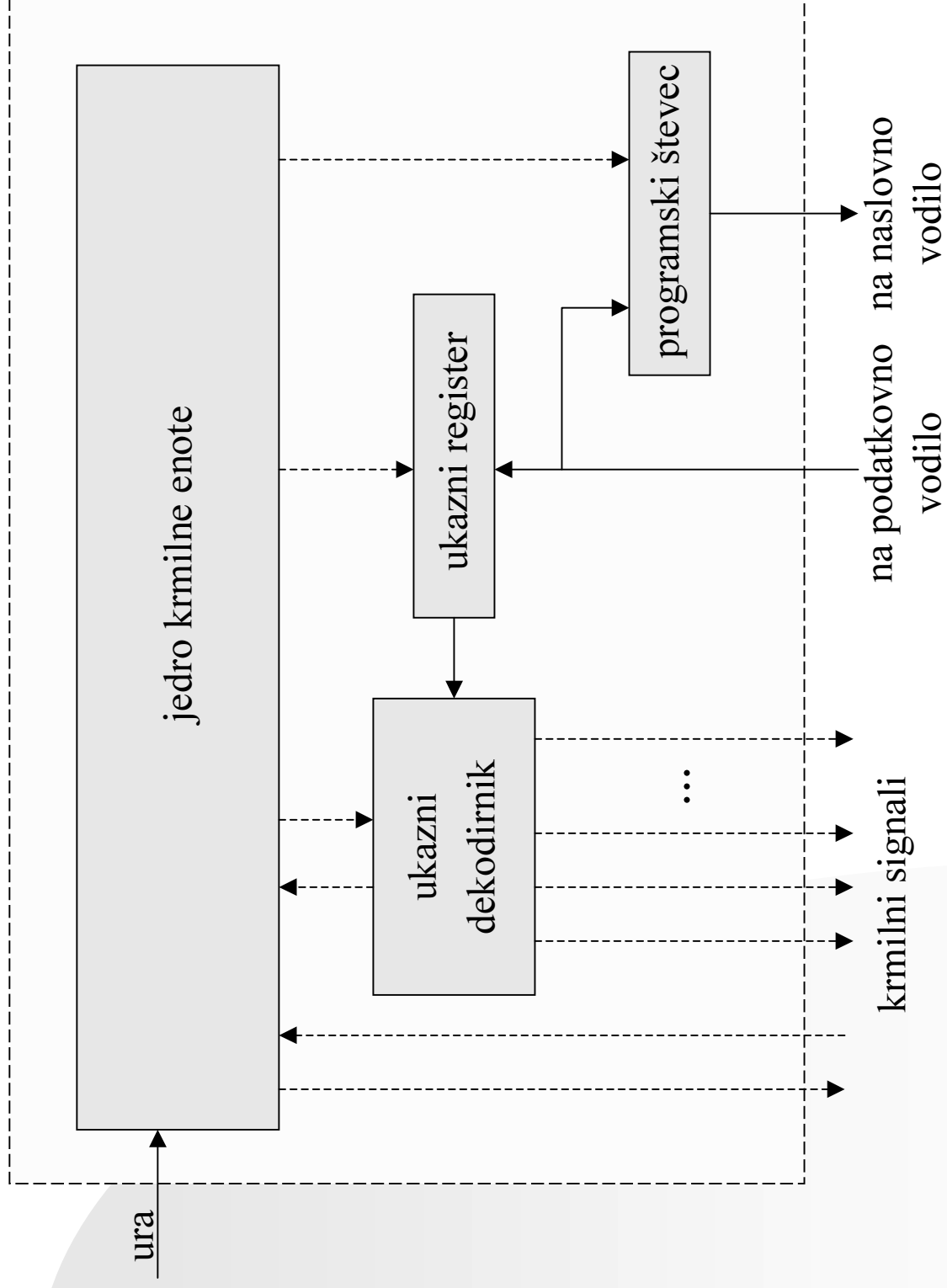




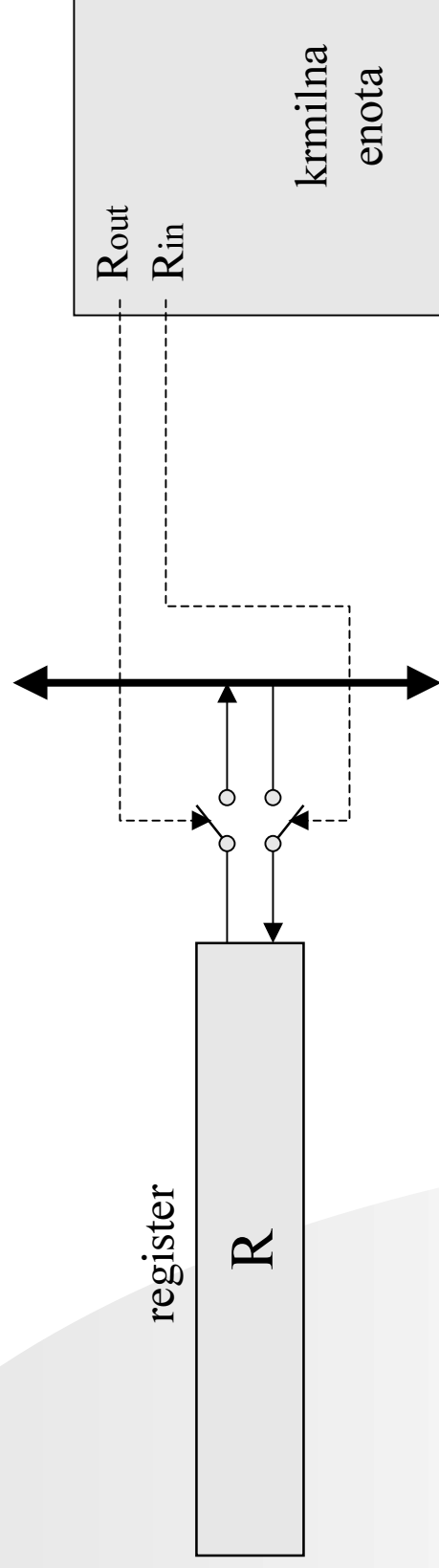


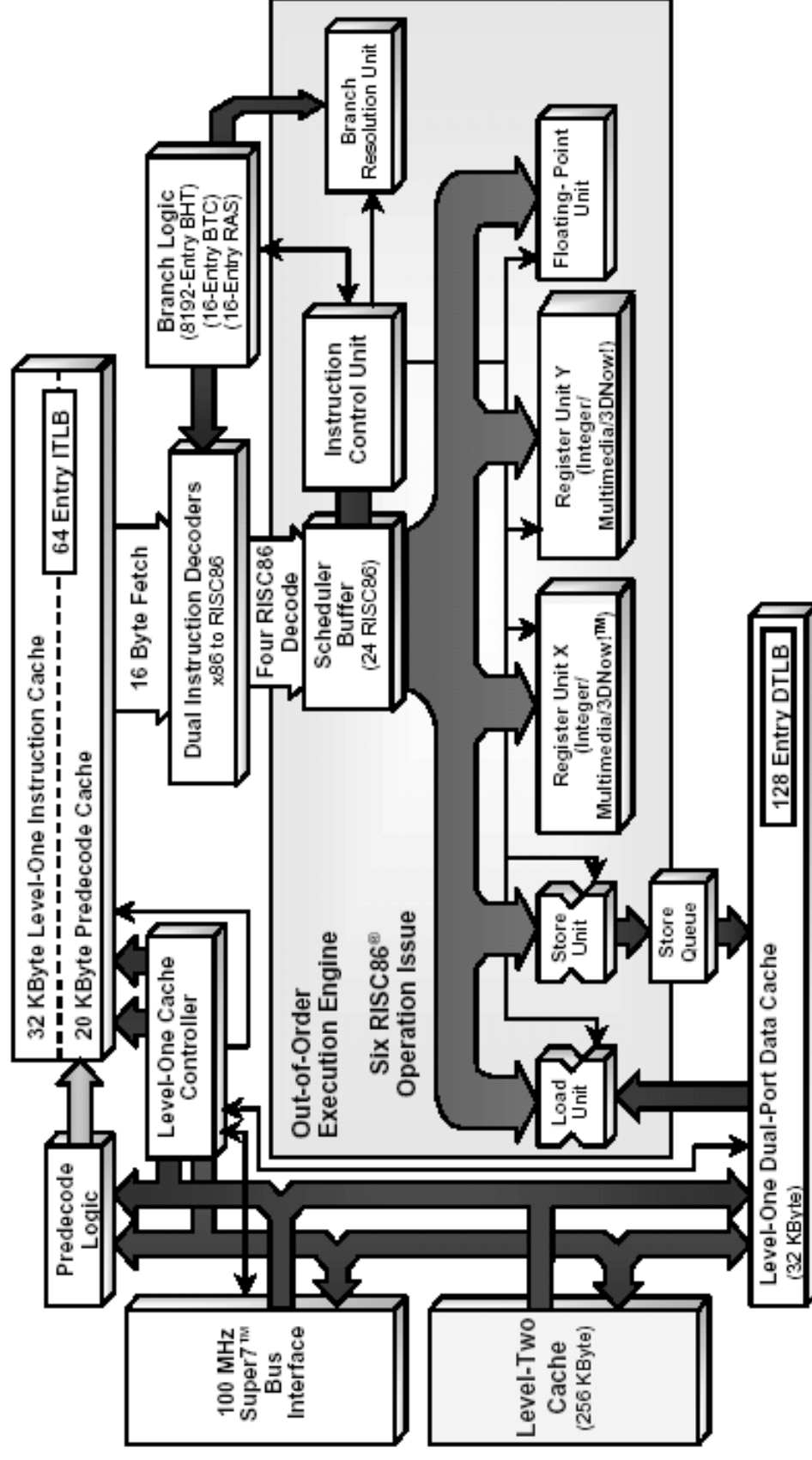
Krmilna enota

- Upravlja delovanje mikroprocesorja
- Sinhrono vezje, ki deluje po taktu ure
- Sestavljena iz:
 - ◆ jedra krmilne enote: krmili delovanje mikroprocesorja
 - ◆ ukaznega registra: sprejme naslednji ukaz, ki ga mora mikroprocesor izvesti
 - ◆ ukaznega dekodirnika: dekodira ukaz in generira krmilne signale za njegovo izvedbo
 - ◆ programskega števec: določa naslov naslednjega ukaza, ki se mora izvesti

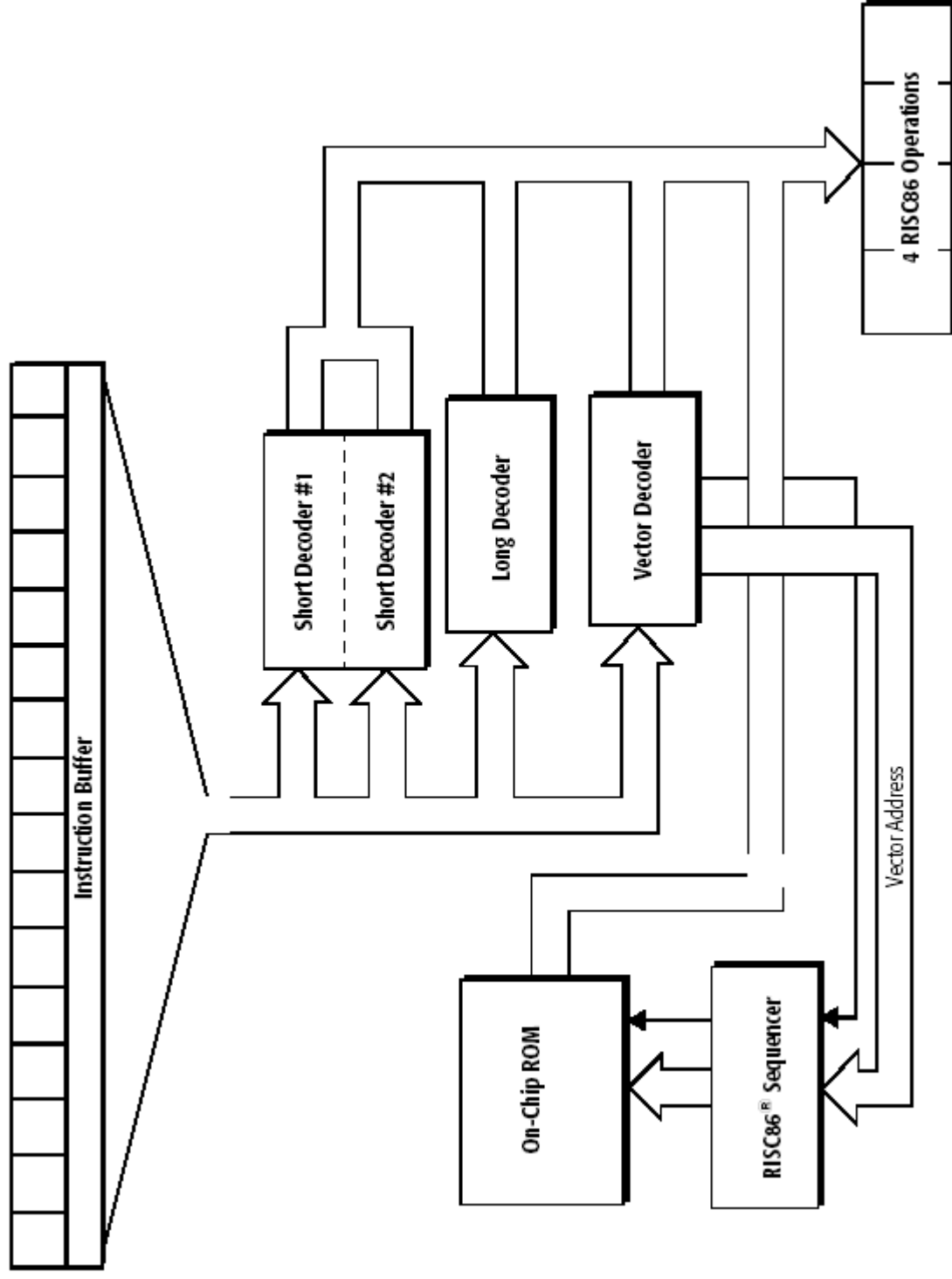


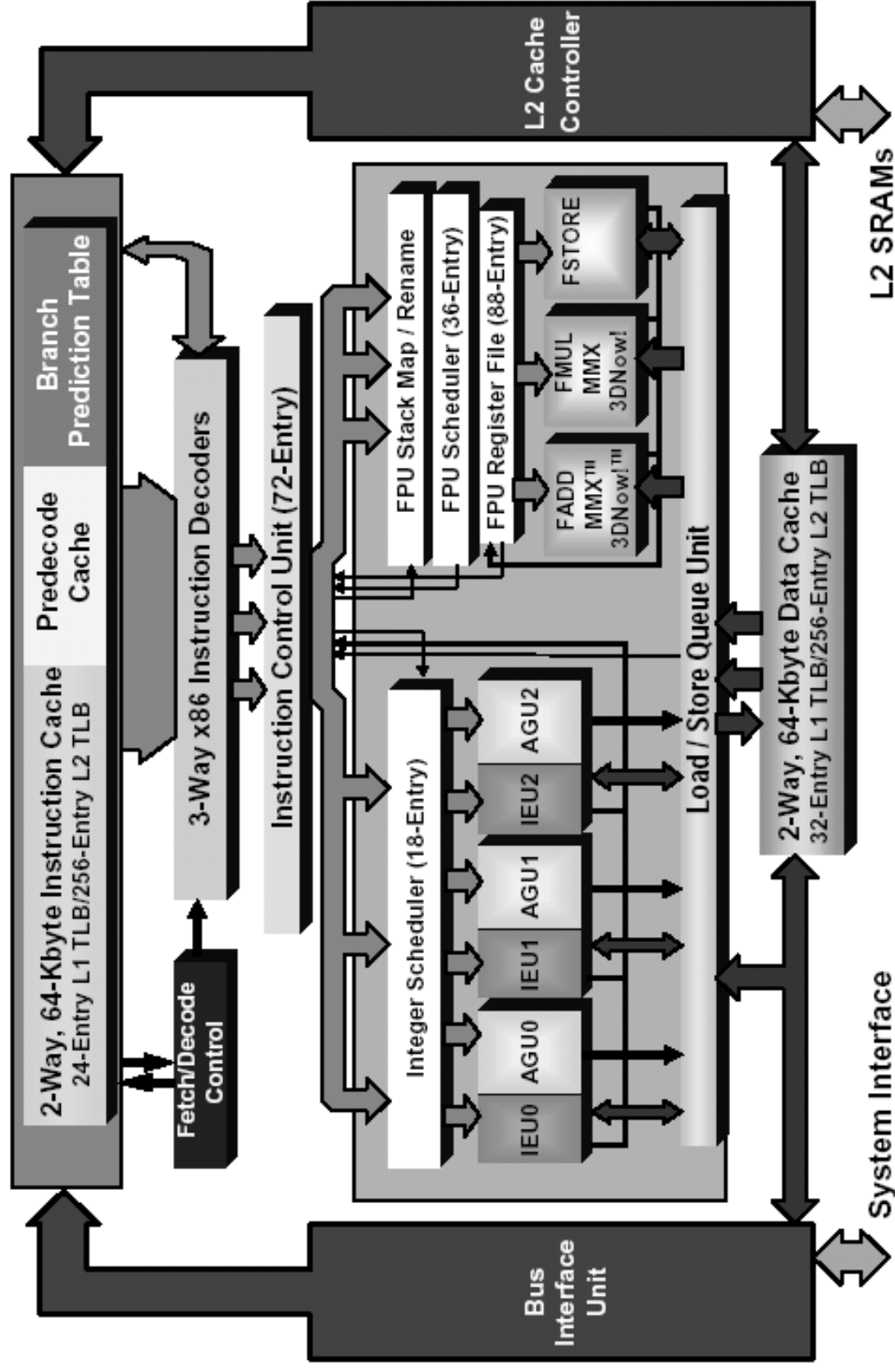
Prikaz delovanja krmilnih signalov





AMD-K6®

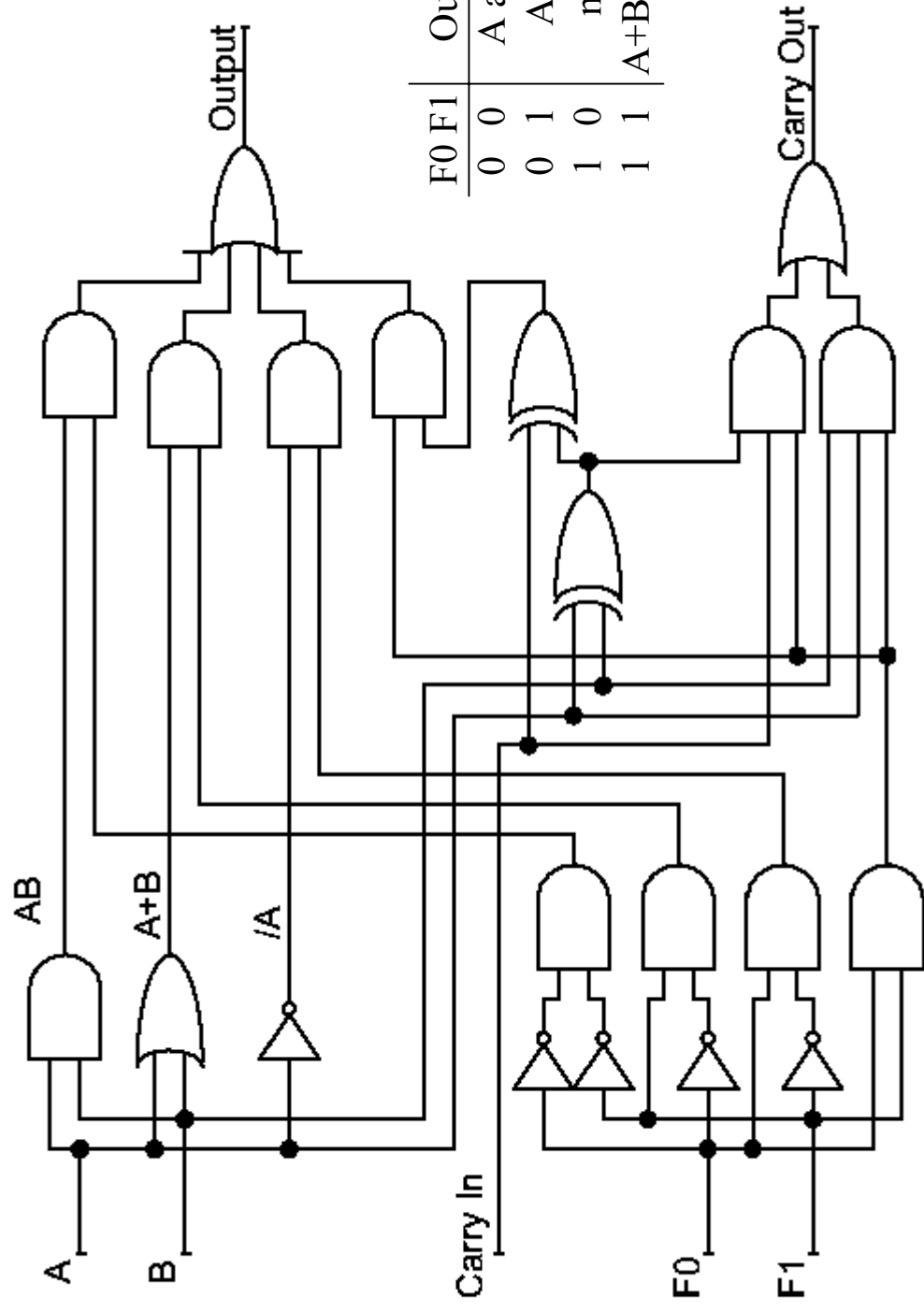




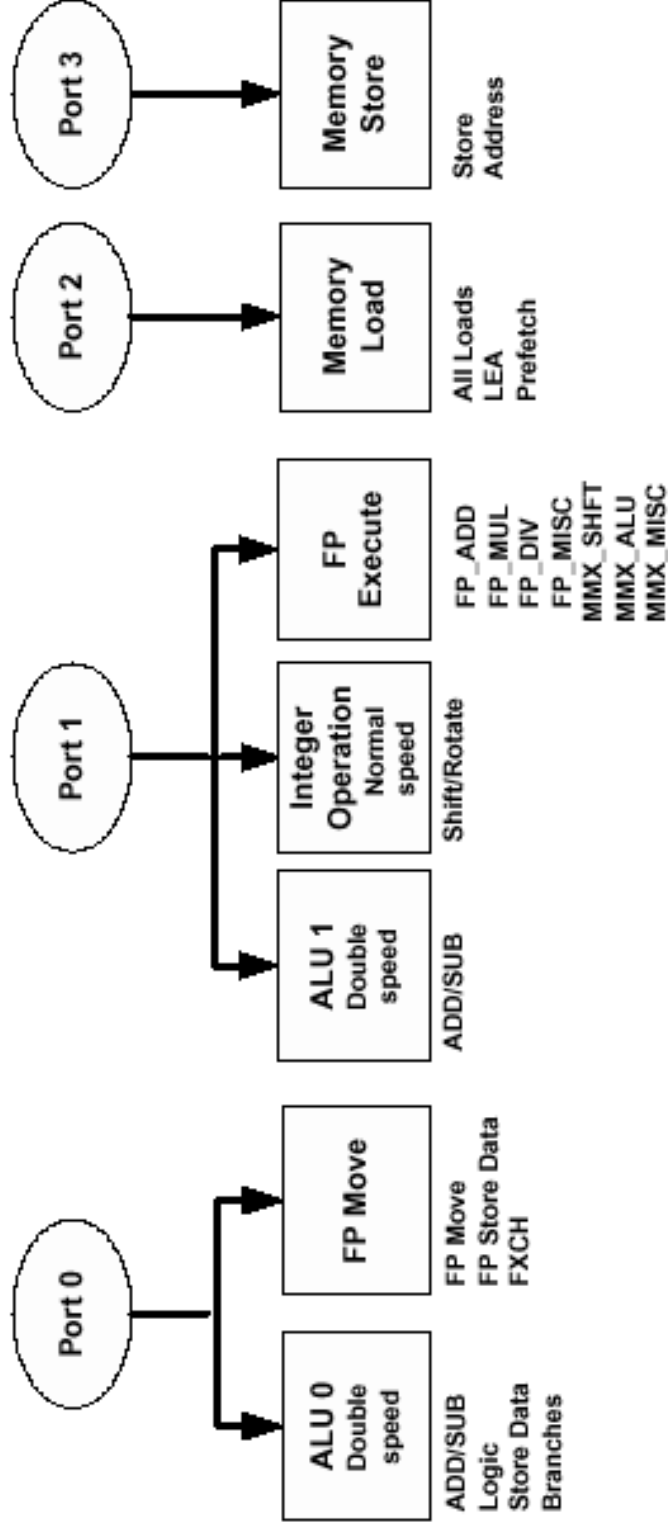
AMD Athlon™

Aritmetično/logična enota (ALE)

- Izvajanje operacij nad operandi mikroprocesorja (matematične in logične operacije)
- Sodobnejši mikroprocesorji imajo več ALE (za cela in realna števila) in podpirajo kompleksnejše operacije (sin, log, ...)
- Izvedena kot čisto preklopno vezje



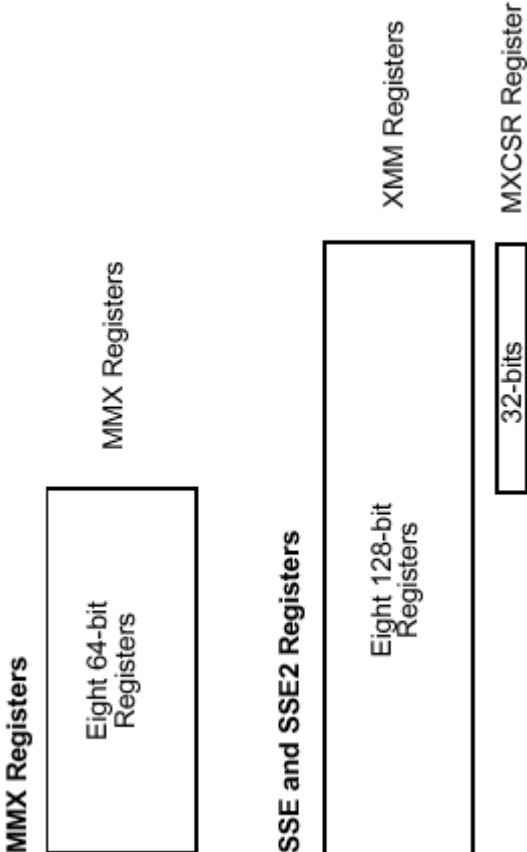
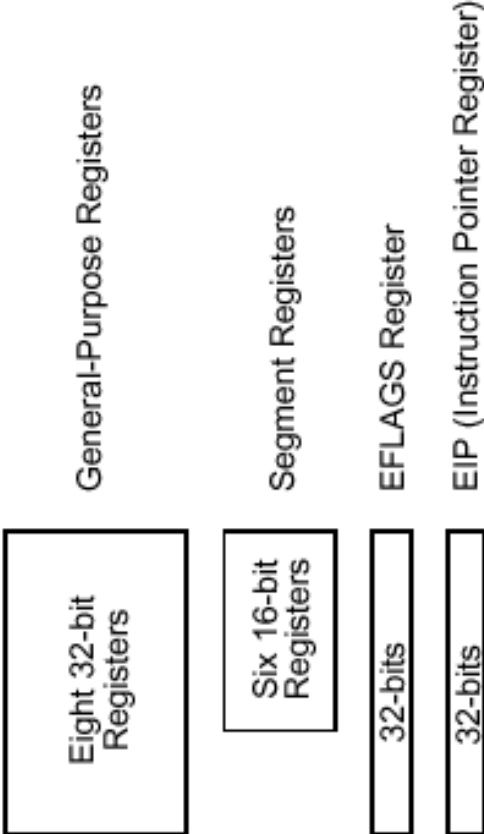
F0	F1	Output	CarryOut
0	0	A and B	0
0	1	A or B	0
1	0	not A	0
1	1	A+B+CarryIn	CarryOut



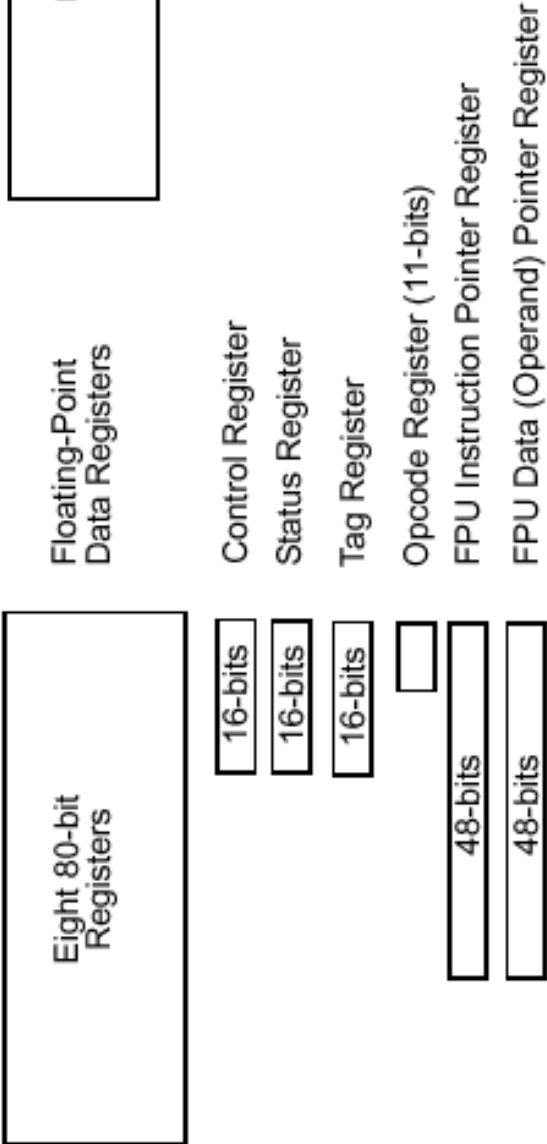
Registri

- Hitre pomnilniške celice znotraj mikroprocesorja
- Osnovne skupine:
 - ◆ Podatkovni registri: operandi pri ukazih, shramba
 - ◆ Naslovni registri: določajo operande v pomnilniku
 - ◆ Posebni registri: PC, UR, ...

Basic Program Execution Registers

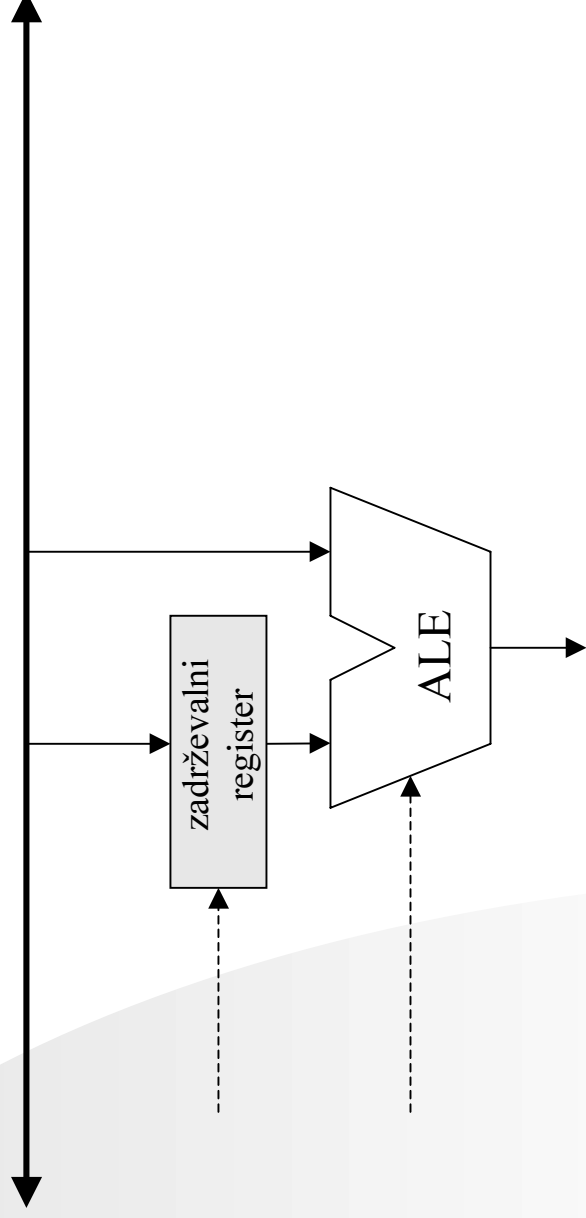


FPU Registers



Prenos podatkov znotraj mikroprocesorja

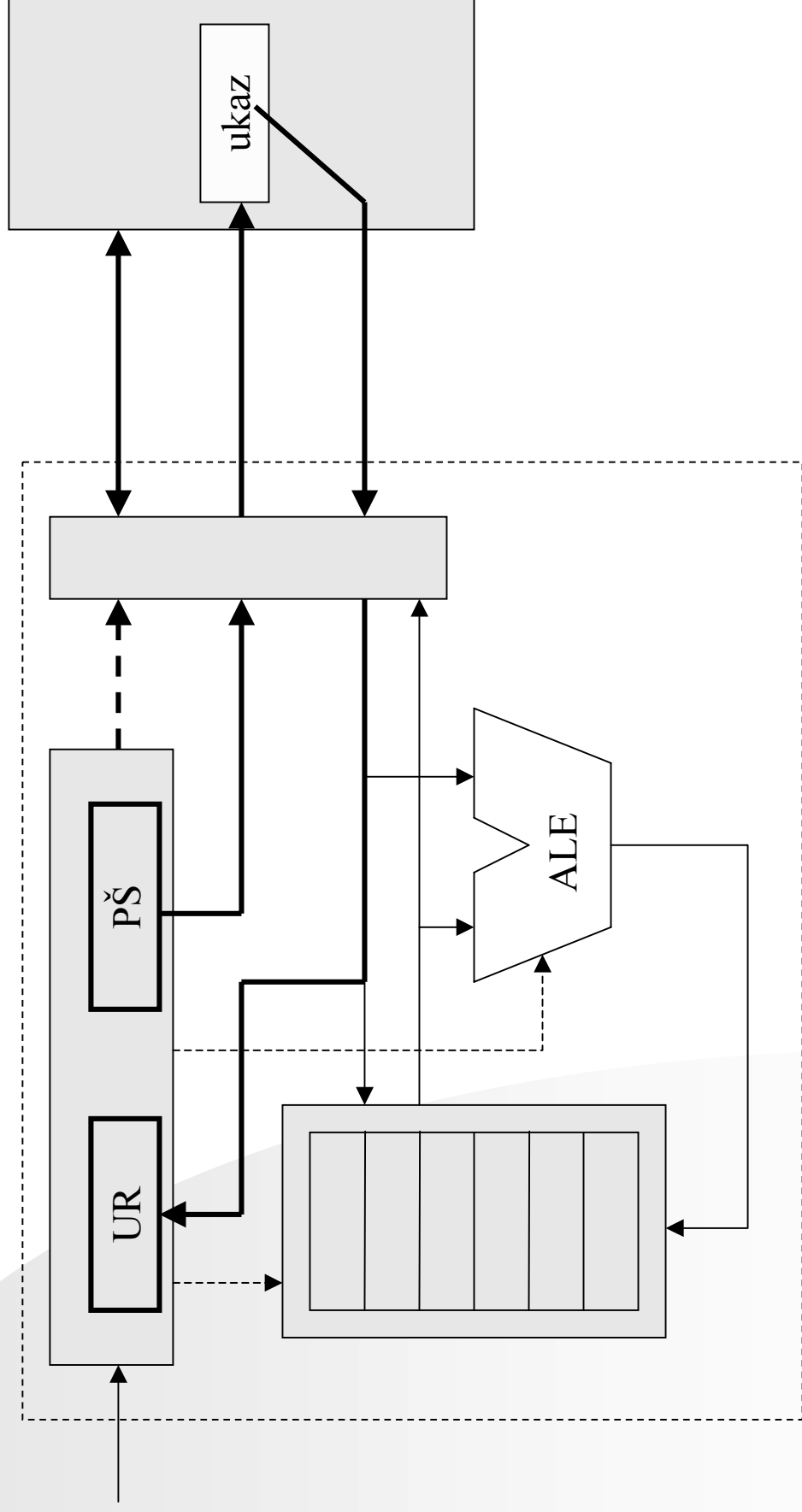
- Interna vodila
- ◆ skupno notranje vodilo



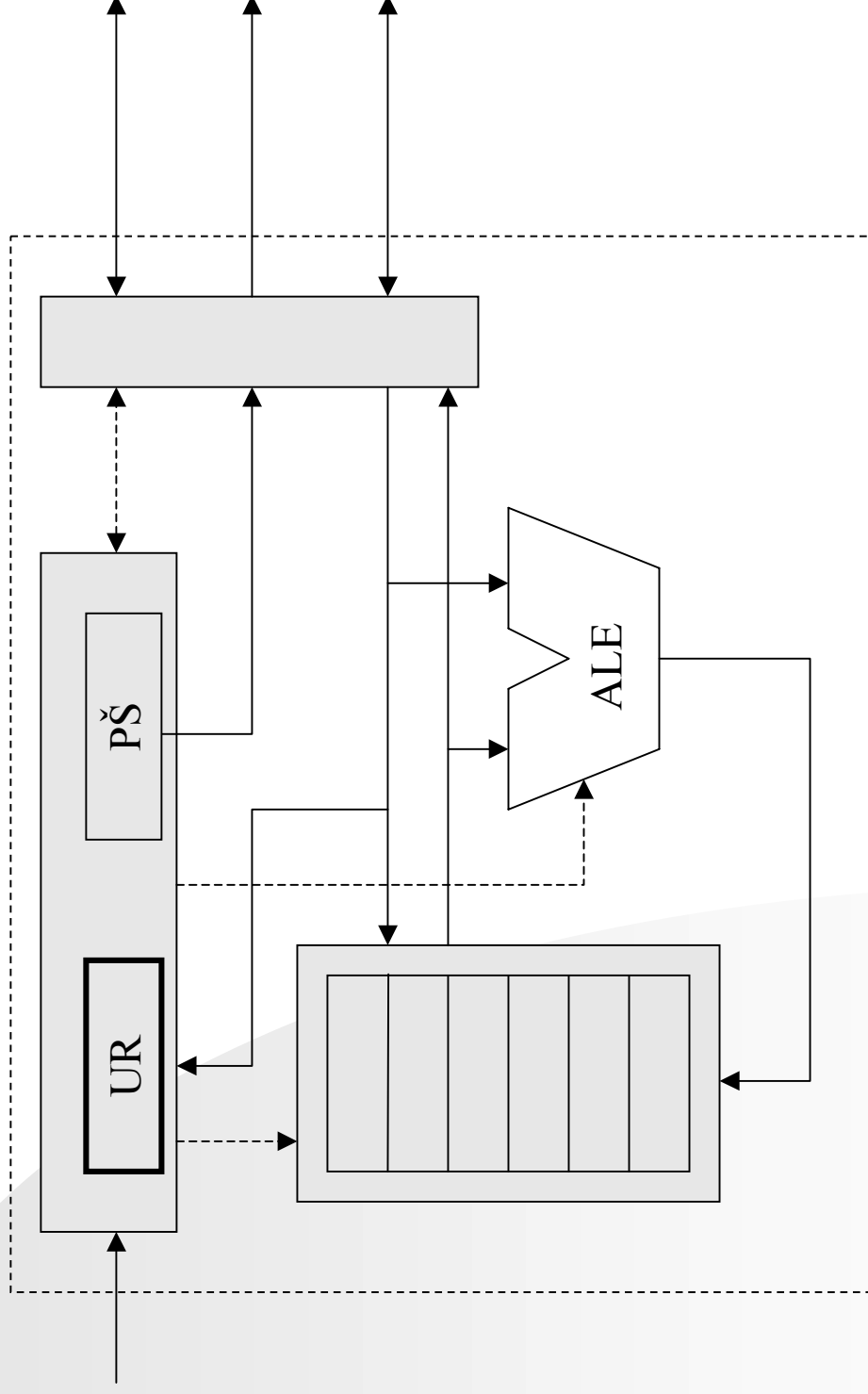
- ◆ ločena notranja vodila

Faze delovanja mikroprocesorja

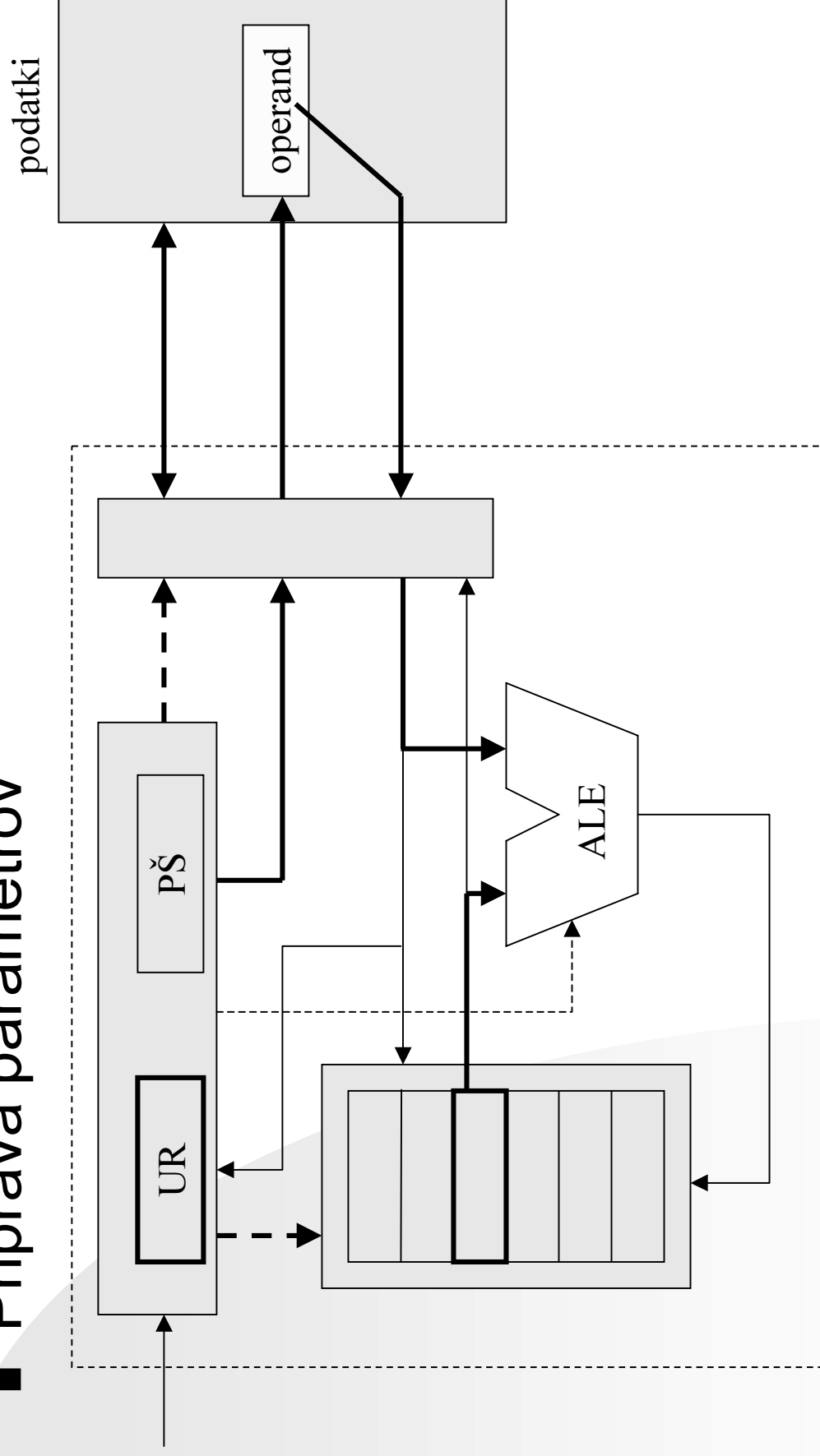
- Prevzem ukaza



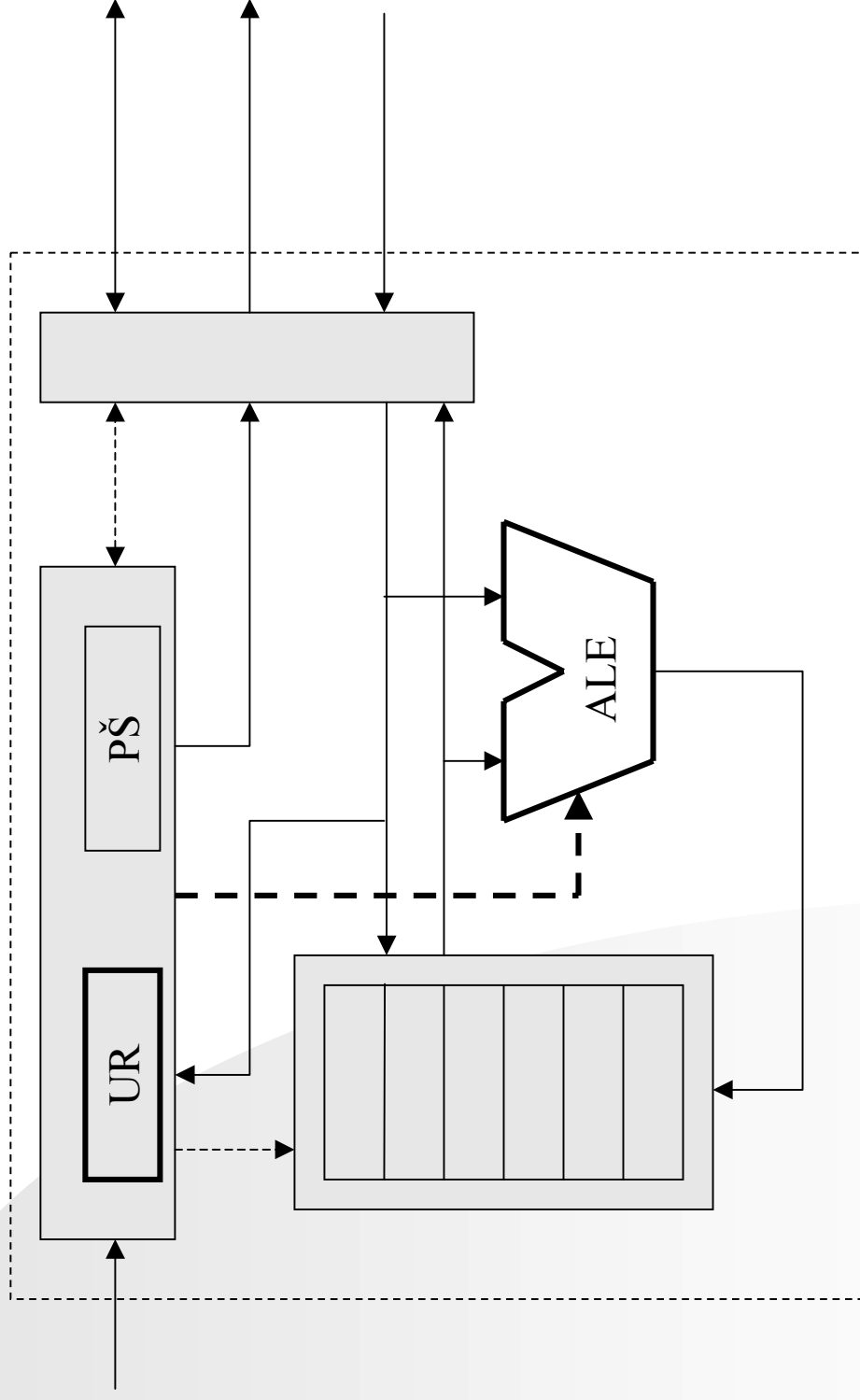
■ Dekodiranje ukaza



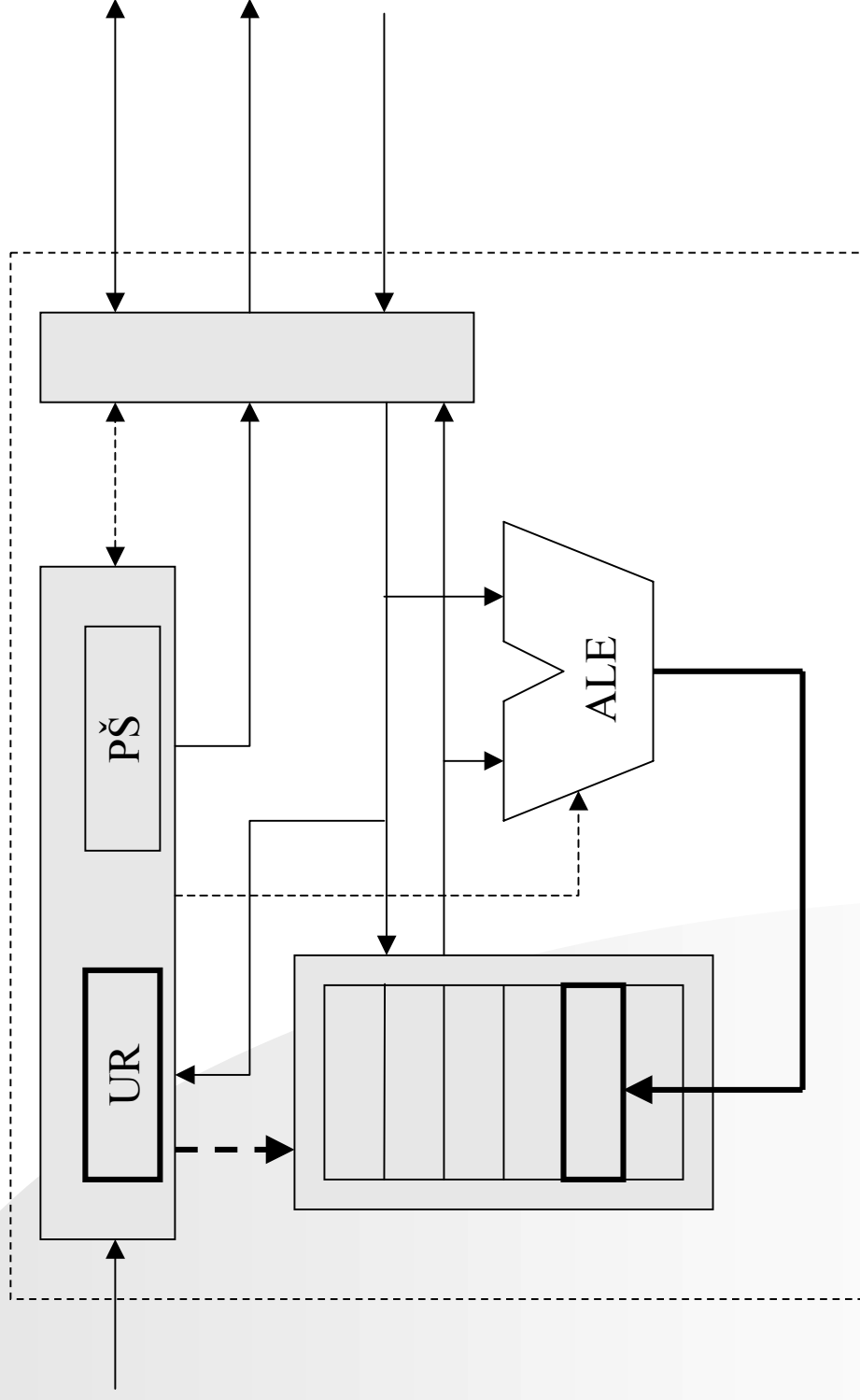
■ Priprava parametrov



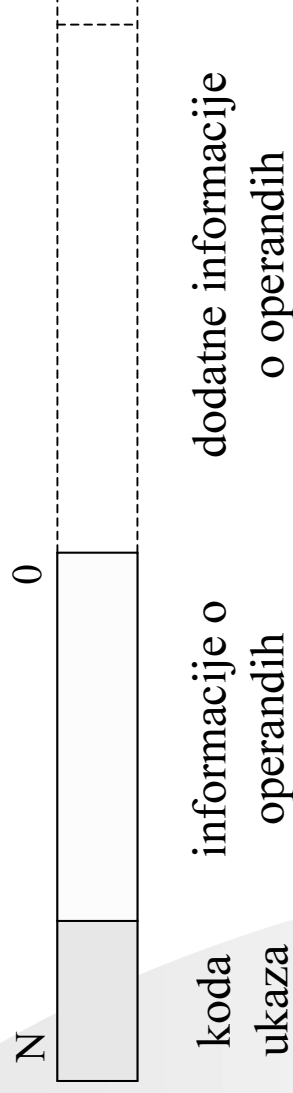
■ Izvedba ukaza



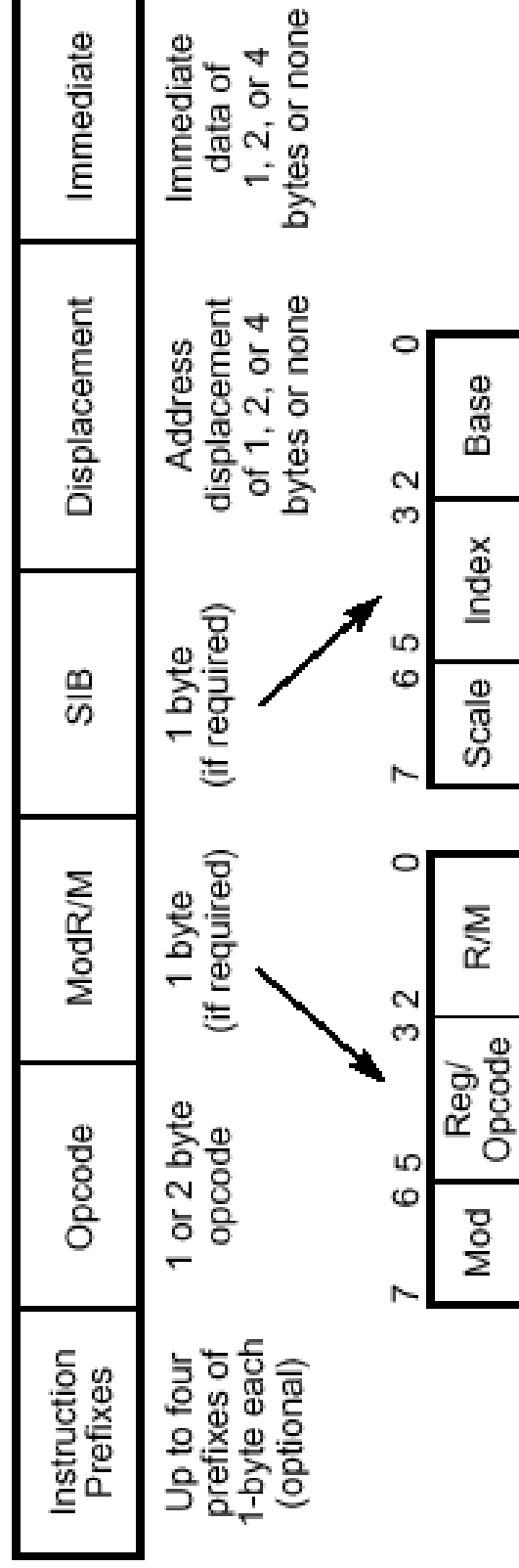
■ Shranjevanje rezultatov



Oblika strojnih ukazov



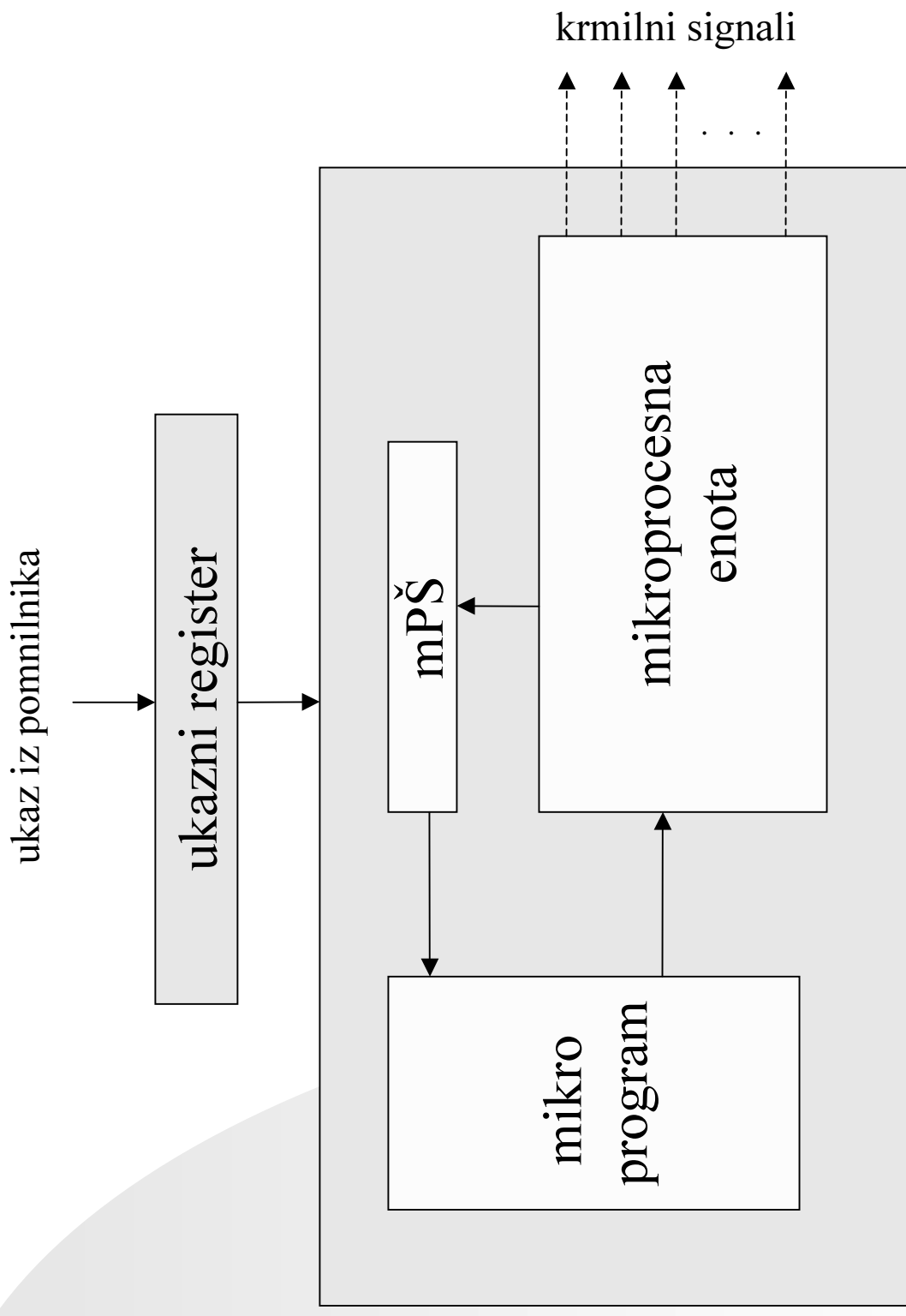
- Ukazi spremenljive dolžine
- Ukazi fiksne dolžine



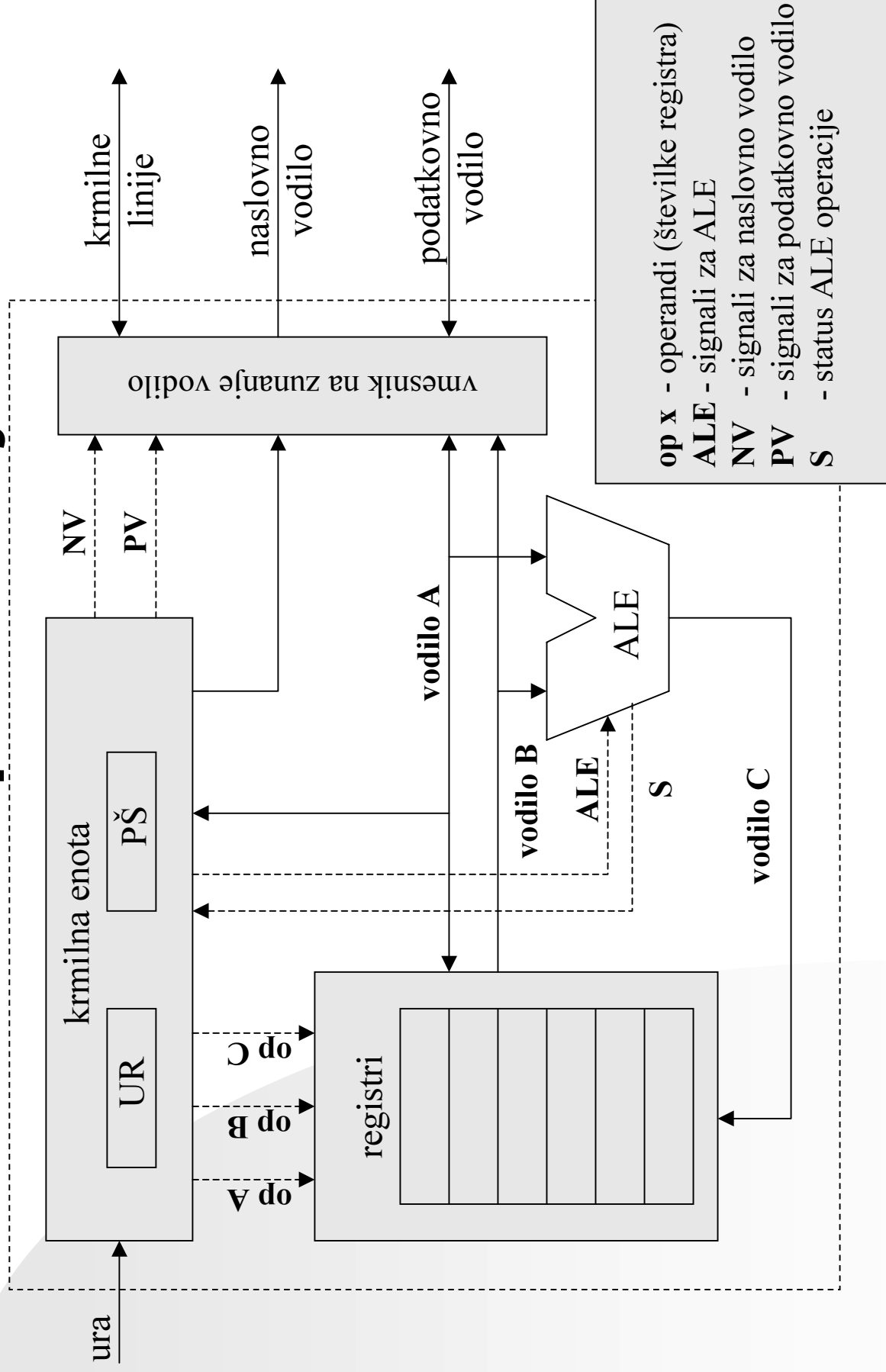
Implementacija krmilne enote

- Dve osnovni izvedbi:
 - ◆ kombinacijska logika
 - ◆ mikroprogramiranje

Zgled mikroprogramirane krmilne enote



Zgled delovanja hipotetičnega mikropcesorja



Nabor ukazov

op.koda	op A	op B	op C
---------	------	------	------

0000 - ADD a,b,c
 0001 - SUB a,b,c
 0010 - OR a,b,c
 0011 - AND a,b,c
 0100 - NEG a,b
 0101 - NOT a,b
 0110 - MOV [a],b
 0111 - MOV a,[b]

$R_C = R_A + R_B$
 $R_C = R_A - R_B$
 $R_C = R_A \text{ or } R_B$
 $R_C = R_A \text{ and } R_B$
 $R_C = -R_A$
 $R_C = \text{not } R_A$
 $R_C = [R_A]$
 $[R_A] = R_B$

[x] - vsebina pomnilnika

op.koda	A	naslov/konstanta
---------	---	------------------

1000 - MOV naslov,A
 1001 - MOV A,naslov
 1010 - MOV #konst,A

$[n] = R_A$
 $R_A = [n]$
 $R_A = k$

MOV a,b \equiv OR a,a,b

PŠ	ALE	A
PŠ	00 - brez spre	
	01 - povečaj z	
	10 - naloži iz U	
	11 - naloži iz v	
ALE	000 - brez spre	
	001 - $C = A$	
	010 - $C = A+B$	
	011 - $C = A-B$	
	100 - $C = A$ or	
	101 - $C = A$ and	
	110 - $C = \text{not } A$	
	111 - $C = -A$	
A,B,C	0 - neaktivno	
	1 - dostop do r	branje, C p

PŠ	ALE	A
PŠ	00 - brez spre	
	01 - povečaj z	
	10 - naloži iz U	
	11 - naloži iz v	
ALE	000 - brez spre	
	001 - $C = A$	
	010 - $C = A+B$	
	011 - $C = A-B$	
	100 - $C = A$ or	
	101 - $C = A$ and	
	110 - $C = \text{not } A$	
	111 - $C = -A$	
A,B,C	0 - neaktivno	
	1 - dostop do r	branje, C p

PŠ	ALE	A
PŠ	00 - brez spre	
	01 - povečaj z	
	10 - naloži iz U	
	11 - naloži iz v	
ALE	000 - brez spre	
	001 - $C = A$	
	010 - $C = A+B$	
	011 - $C = A-B$	
	100 - $C = A$ or	
	101 - $C = A$ and	
	110 - $C = \text{not } A$	
	111 - $C = -A$	
A,B,C	0 - neaktivno	
	1 - dostop do r	branje, C p

PŠ	ALE	A
PŠ	00 - brez spre	
	01 - povečaj z	
	10 - naloži iz U	
	11 - naloži iz v	
ALE	000 - brez spre	
	001 - $C = A$	
	010 - $C = A+B$	
	011 - $C = A-B$	
	100 - $C = A$ or	
	101 - $C = A$ and	
	110 - $C = \text{not } A$	
	111 - $C = -A$	
A,B,C	0 - neaktivno	
	1 - dostop do r	branje, C p

Izvajanje ukaza

op.koda	op A	op B	op C
---------	------	------	------

0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 ADD R0,R5,R8

PŠ	ALE	A	B	C	UR	PV	NV	ostali signali
----	-----	---	---	---	----	----	----	----------------

1. 0 0 0 0 0 0 0 0 0 0 0 0 0 1 vsebina PŠ se prenese na naslovno vodilo

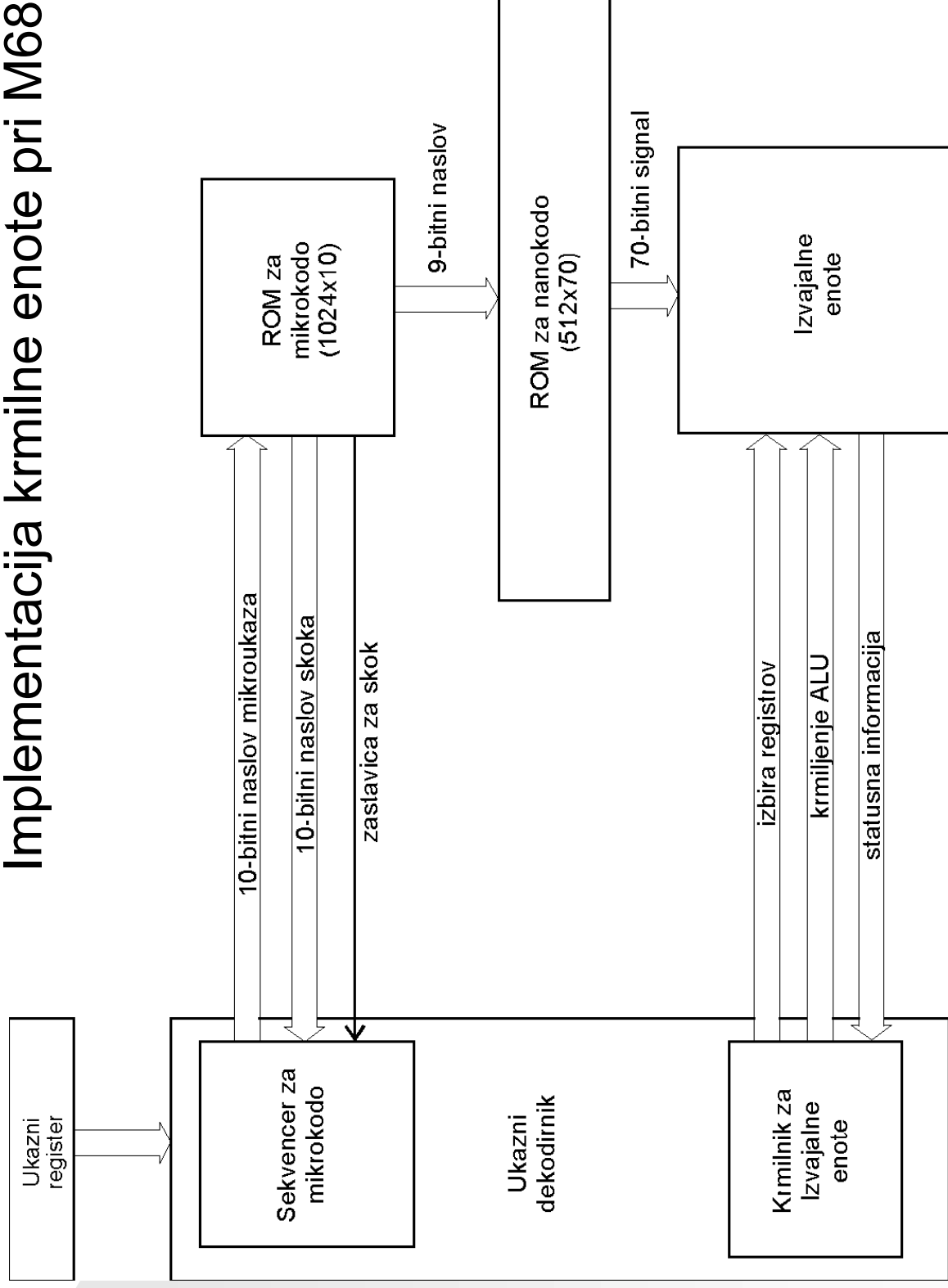
2. 0 1 0 0 0 0 0 0 0 1 0 1 0 0 vrednost is podatkovnega vodila se prepiše v UR, dekodiranje ukaza, PŠ = PŠ+1

3. 0 0 0 1 0 1 1 1 0 0 0 0 0 0 na vodilo A se prenese R0, na vodilo B se prenese R5, ALE izvede operacijo seštevanja, rezultat se shrani v R8

1. 0 0 0 0 0 0 0 0 0 0 0 0 0 1 vsebina PŠ se prenese na naslovno vodilo

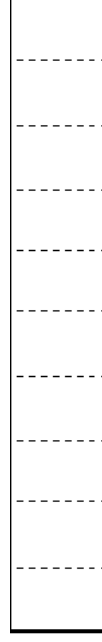
2. 0 1 0 0 0 0 0 0 0 1 0 1 0 0 vrednost is podatkovnega vodila se prepiše v UR, dekodiranje ukaza, PŠ = PŠ+1

Implementacija krmilne enote pri M68000



Oblika mikroukaza

Format 1:



10-bitni naslov
mikroukaza (skok)

Format 2:



9-bitni naslov
nanoukaza

zastavica za skok v naslednjem ukazu