

Građa računala

3. Logička vrata, tablice istine i
Boolean-ova algebra. Pojednostavljeni
model CISC i RISC procesora.




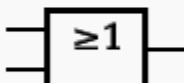

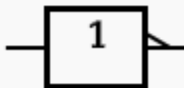


Preddiplomski izvanredni stručni studij
Informacijske tehnologije




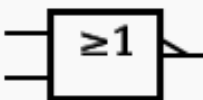
Boolean-ova algebra

- matematička osnova za digitalnu logiku
- pogodan alat u analizi opisivanja funkcija digitalnih krugova
- osnove Boolean-ove algebre:
- varijable mogu imati vrijednost:
 - 1(TRUE) ili 0 (FALSE)
- osnovne funkcije: AND, OR i NOT


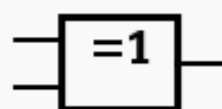

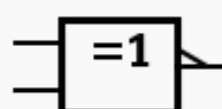
Osnovni logički sklopovi

Operacija	Simbol (ANSI)	Simbol (IEC)	Booleov izraz	Tablica istine																		
I (AND)			$A \cdot B$	<table><tr><th colspan="2">ULAZ</th><th>IZLAZ</th></tr><tr><th>A</th><th>B</th><th>A I B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	ULAZ		IZLAZ	A	B	A I B	0	0	0	0	1	0	1	0	0	1	1	1
ULAZ		IZLAZ																				
A	B	A I B																				
0	0	0																				
0	1	0																				
1	0	0																				
1	1	1																				
ILI (OR)			$A + B$	<table><tr><th colspan="2">ULAZ</th><th>IZLAZ</th></tr><tr><th>A</th><th>B</th><th>A ILI B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	ULAZ		IZLAZ	A	B	A ILI B	0	0	0	0	1	1	1	0	1	1	1	1
ULAZ		IZLAZ																				
A	B	A ILI B																				
0	0	0																				
0	1	1																				
1	0	1																				
1	1	1																				
NE (NOT)			\overline{A}	<table><tr><th>ULAZ</th><th>IZLAZ</th></tr><tr><th>A</th><th>NE A</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	ULAZ	IZLAZ	A	NE A	0	1	1	0										
ULAZ	IZLAZ																					
A	NE A																					
0	1																					
1	0																					

Izvedeni logički sklopovi

Operacija	Simbol (ANSI)	Simbol (IEC)	Booleov izraz	Tablica istine																		
NI (NAND)			$\overline{A \cdot B}$	<table><tr><th colspan="2">ULAZ</th><th>IZLAZ</th></tr><tr><th>A</th><th>B</th><th>A NI B</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	ULAZ		IZLAZ	A	B	A NI B	0	0	1	0	1	1	1	0	1	1	1	0
ULAZ		IZLAZ																				
A	B	A NI B																				
0	0	1																				
0	1	1																				
1	0	1																				
1	1	0																				
NILI (NOR)			$\overline{A + B}$	<table><tr><th colspan="2">ULAZ</th><th>IZLAZ</th></tr><tr><th>A</th><th>B</th><th>A NILI B</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	ULAZ		IZLAZ	A	B	A NILI B	0	0	1	0	1	0	1	0	0	1	1	0
ULAZ		IZLAZ																				
A	B	A NILI B																				
0	0	1																				
0	1	0																				
1	0	0																				
1	1	0																				

Izvedeni logički sklopovi

Operacija	Simbol (ANSI)	Simbol (IEC)	Booleov izraz	Tablica istine																		
XILI (XOR)			$A \oplus B$	<table><tr><th colspan="2">ULAZ</th><th>IZLAZ</th></tr><tr><th>A</th><th>B</th><th>A XILI B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	ULAZ		IZLAZ	A	B	A XILI B	0	0	0	0	1	1	1	0	1	1	1	0
ULAZ		IZLAZ																				
A	B	A XILI B																				
0	0	0																				
0	1	1																				
1	0	1																				
1	1	0																				
XNILI (XNOR)			$\overline{A \oplus B}$	<table><tr><th colspan="2">ULAZ</th><th>IZLAZ</th></tr><tr><th>A</th><th>B</th><th>A XNILI B</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	ULAZ		IZLAZ	A	B	A XNILI B	0	0	1	0	1	0	1	0	0	1	1	1
ULAZ		IZLAZ																				
A	B	A XNILI B																				
0	0	1																				
0	1	0																				
1	0	0																				
1	1	1																				

Boolean-ova algebra

- Osnovni aksiomi Boolean-ova algebre
- **A.1. Neutralni element**
 - a) $A+0 = A$
 - b) $A \cdot 0 = 0$
- **A.2. Komplement**
 - a) $A+\bar{A} = 1$
 - b) $A \cdot \bar{A} = 0$

Boolean-ova algebra

- Osnovni aksiomi Boolean-ova algebre

- **A.3. Komutativnost**

a) $A+B = B+A$

b) $A \cdot B = B \cdot A$

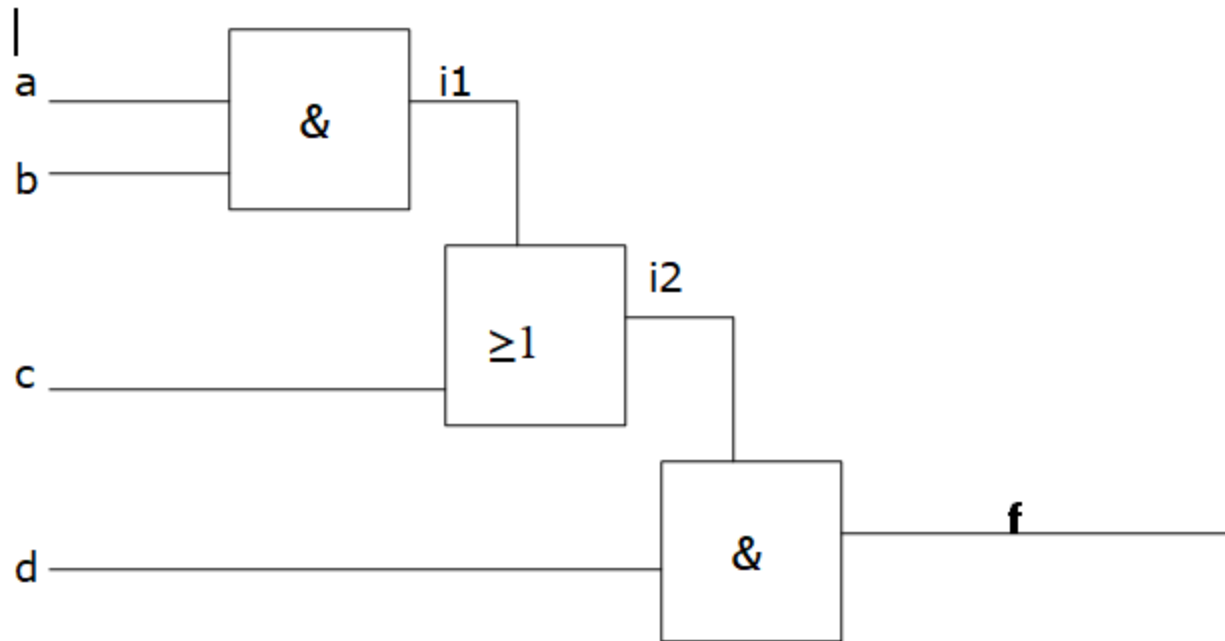
- **A.4. Distributivnost**

a) $A \cdot (B+C) = A \cdot B + A \cdot C$

b) $A + B \cdot C = (A+B) \cdot (A+C)$

Boolean-ova algebra

- Primjer: *Na slici je prikazan logički sklop ostvaren upotrebom logičkih sklopova I i ILL. Koju logičku funkciju sklop ostvaruje ?*



Boolean-ova algebra

- Označimo sa i_1 i i_2 međurezultate:
- $i_1 = a \cdot b$
- $i_2 = i_1 + c$
- $f = i_2 \cdot d = (i_1 + c) \cdot d = (a \cdot b + c) \cdot d$

Boolean-ova algebra

- *Primjer:* Napišite tablicu logičkih stanja za sklop iz prethodnog primjera.

i	a	b	i1	c	i2	d	f
0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0
2	0	0	0	1	1	0	0
3	0	0	0	1	1	1	1
4	0	1	0	0	0	0	0
5	0	1	0	0	0	1	0
6	0	1	0	1	1	0	0
7	0	1	0	1	1	1	1
8	1	0	0	0	0	0	0
9	1	0	0	0	0	1	0
10	1	0	0	1	1	0	0
11	1	0	0	1	1	1	1
12	1	1	1	0	1	0	0
13	1	1	1	0	1	1	1
14	1	1	1	1	1	0	0
15	1	1	1	1	1	1	1

Boolean-ova algebra

- Ako imamo zadanu tablicu kombinacija za logičku funkciju tada je možemo prikazati kao:
 - sumu standardnih produkata (mintermi)
 - umnožak standardnih suma (makstermi)

Boolean-ova algebra

- Primjer: *Zadana je tablica kombinacija za logičku funkciju. Prikažite logičku funkciju kao zbroj mintermi.*

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Boolean-ova algebra

- U tablici kombinacija promatramo minterme za koje logička funkcija daje vrijednost 1.
- To su minterme m_4 , m_5 , m_7 , pa je suma standardnih produkata:

$$f = m_4 + m_5 + m_7$$

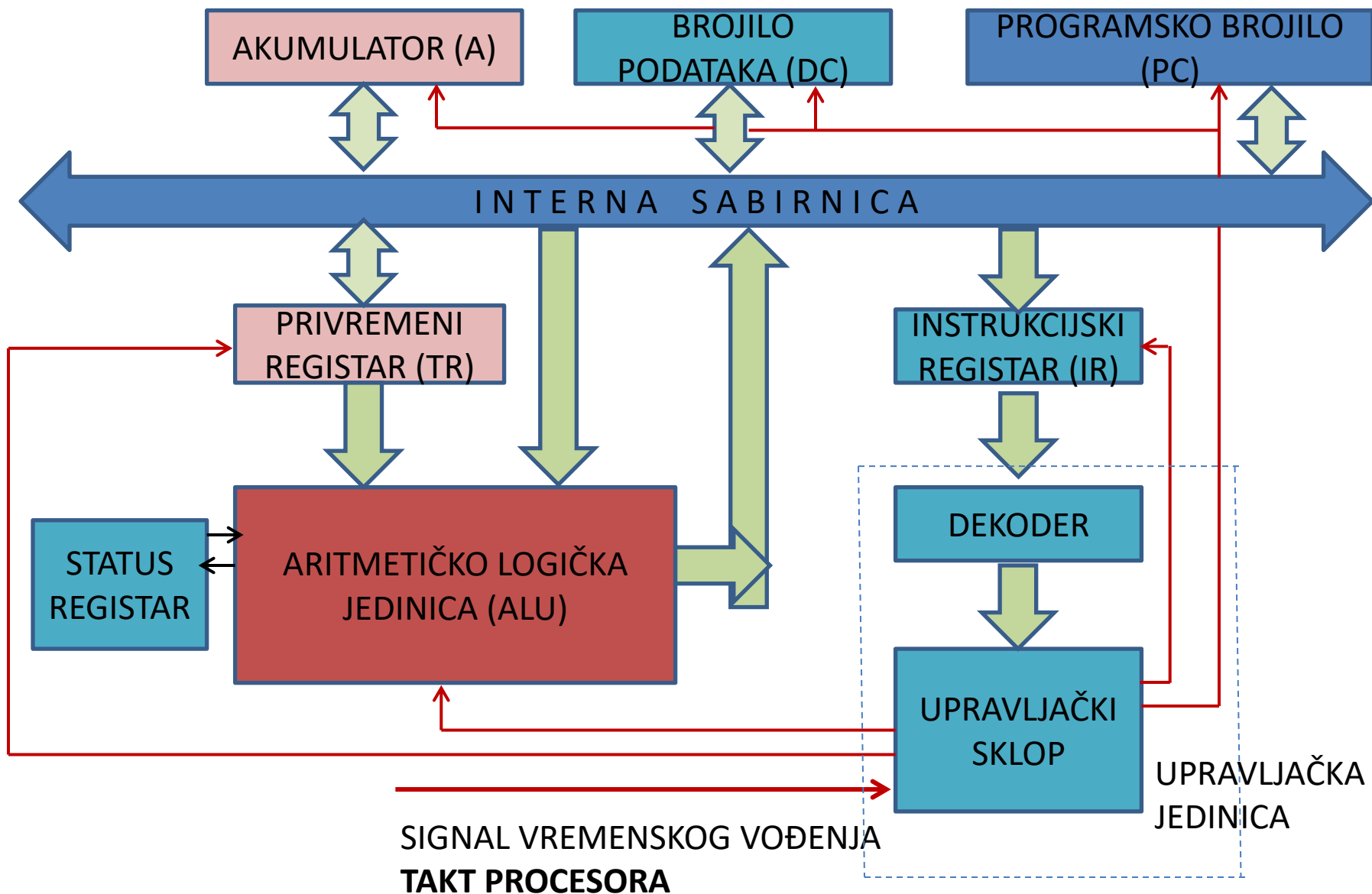
A	B	C	f	mintermi
0	0	0	0	m_0
0	0	1	0	m_1
0	1	0	0	m_2
0	1	1	0	m_3
1	0	0	1	m_4
1	0	1	1	m_5
1	1	0	0	m_6
1	1	1	1	m_7

Boolean-ova algebra

- U retku koji pripada mintermu m_4 , vrijednosti varijabli su slijedeće: $A = 1$, $B = 0$, $C = 0$. Budući da minterm odgovara umnošku varijabli, a u tom retku funkcija mora poprimiti vrijednost 1, slijedi da umnožak mora biti 1, tj. $m_4 = 1 = 1 \cdot 1 \cdot 1 = A \bullet \bar{B} \bullet \bar{C}$
- Sličnim razmatranjem se dobije za:
 $m_5 = A \bullet \bar{B} \bullet C$
 $m_7 = A \bullet B \bullet C$
- Tražena funkcija tada glasi:

$$f(A,B,C) = A \bullet \bar{B} \bullet \bar{C} + A \bullet \bar{B} \bullet C + A \bullet B \bullet C = \Sigma(4,5,7)$$

Model procesora CISC arhitekture



Model procesora CISC arhitekture

- Komponente modela:
 - Akumulator A
 - Programsko brojilo PC
 - Instrukcijski registar IR
 - Brojilo podataka DC
 - Privremeni registar PR
 - Status-registar (Registar stanja)
 - Aritmetičko-logička jedinica (ALU)
 - Interna sabirnica
 - Upravljačka jedinica

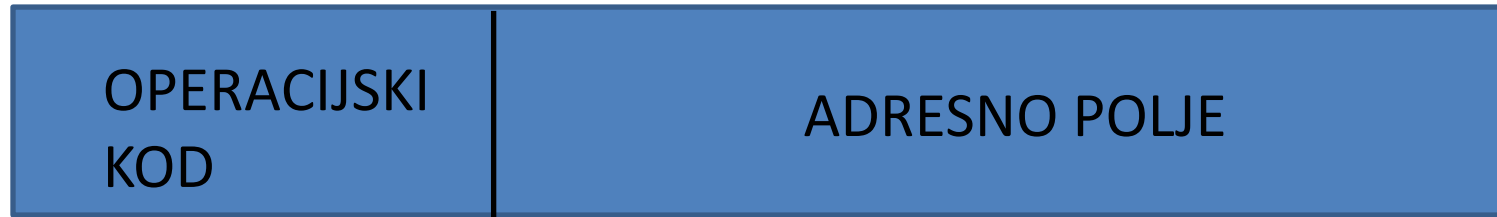
Model procesora CISC arhitekture

- Upravljačka jedinica na temelju dekodiranja strojne instrukcije generira sve potrebne upravljačke signale za vremensko vođenje i upravljanje ostalim jedinicama računala
- Ti se signali dovode u tzv. upravljačke točke i njima se aktiviraju sklopovi u pojedinim funkcijskim jedinicama.

Model procesora CISC arhitekture

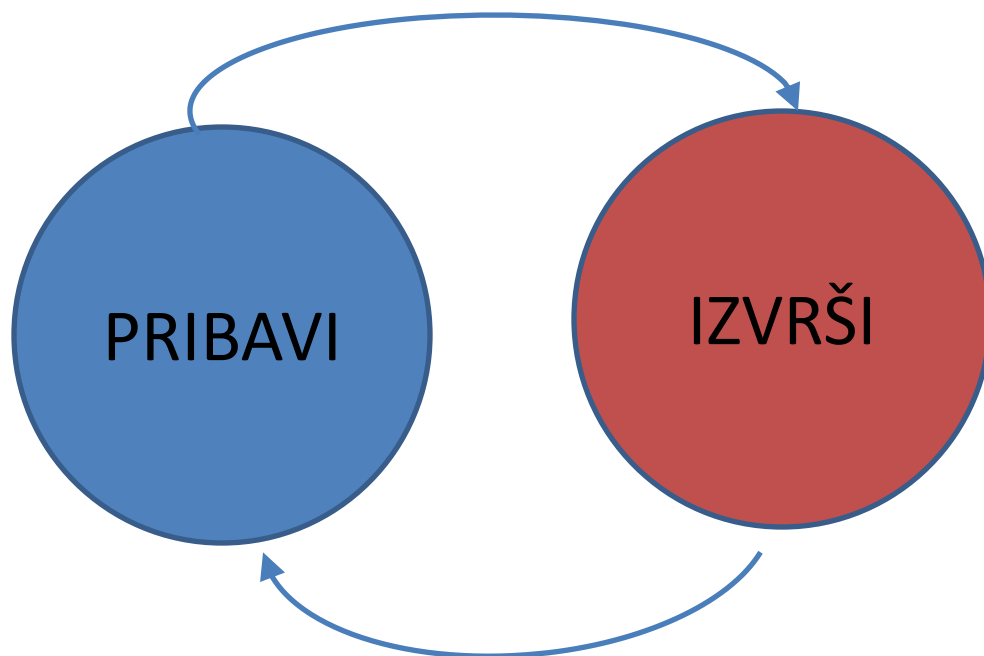
- Upravljačka jedinica zadužena je za automatsko izvršavanje programa - upravljanje slijedom izvršavanja instrukcija kojima je predodčen algoritam obrade
- Svaki je korak algoritma predstavljen jednom strojnom instrukcijom ili slijedom strojnih instrukcija.
- One određuju elementarne operacije koje sklopovlje može izvesti.

Format instrukcije



- **polje operacijskog koda** –što treba izvršiti
- **adresno polje** – nad kojim podacima treba izvršiti operaciju

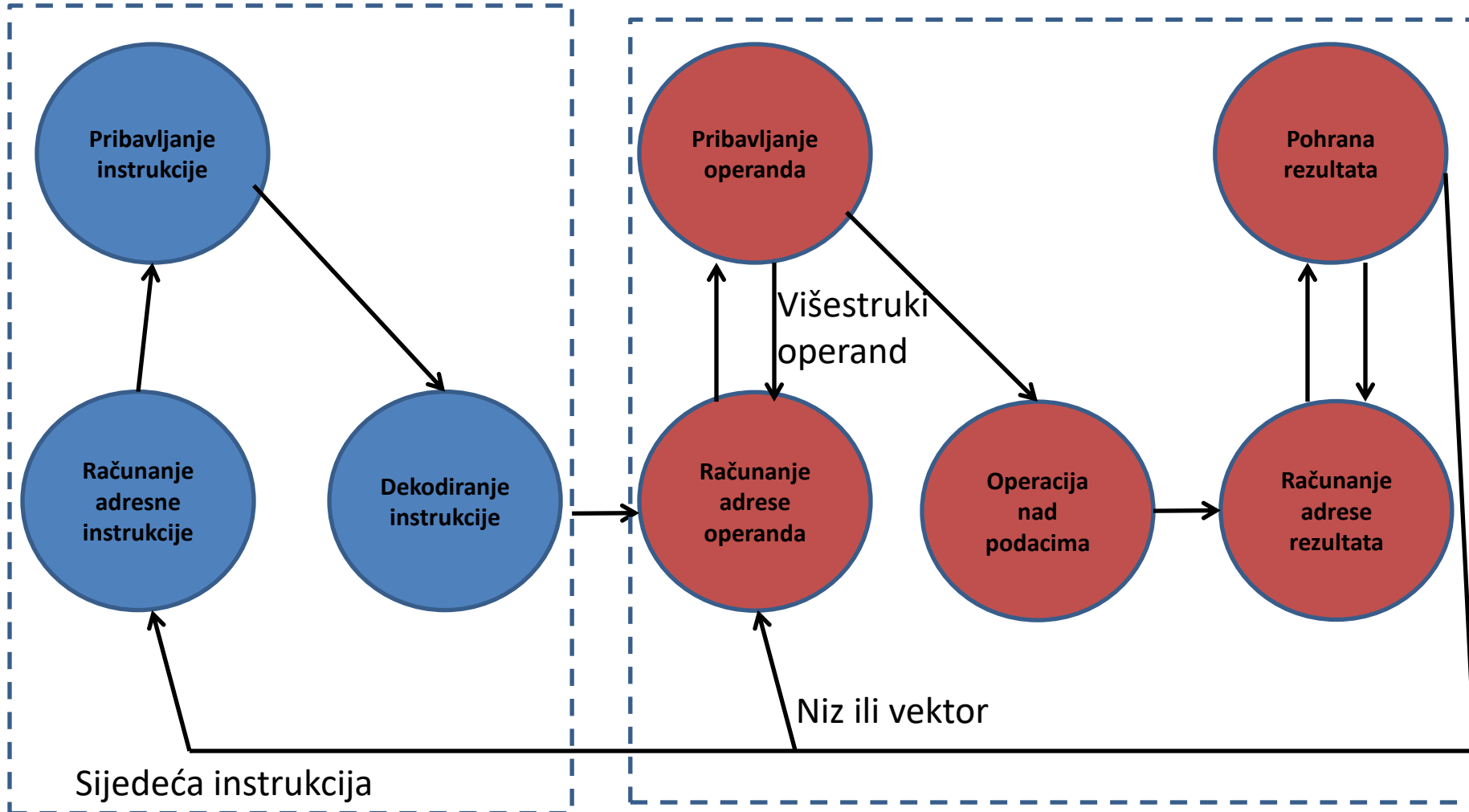
Redoslijed izvršavanja instrukcije



Instrukcijski ciklus

PRIBAVI

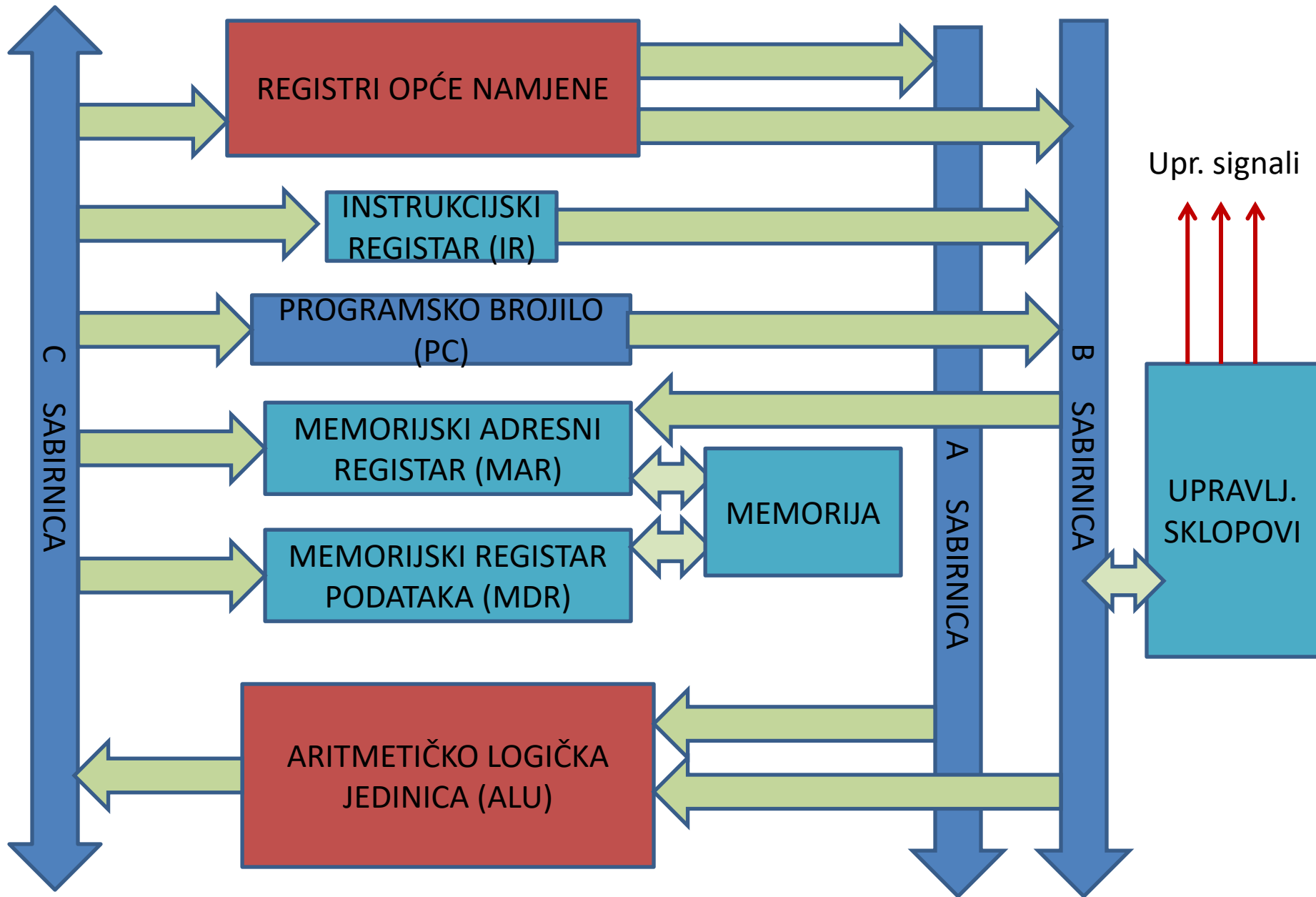
IZVRŠI



Instrukcijski ciklus

- U fazi pribavljanja na osnovi adrese zapisane u registru PC pribavlja se instrukcija iz memorije i prenosi u instrukcijski registar
- U drugoj fazi se instrukcija izvršava tako da se dekodira i da upravljački sklop generira signale potrebne za izvršavanje instrukcije
- Zadnji korak izvršavanja instrukcije je povećanje adrese u registru PC za jedan
- Nakon toga se postupak ponavlja sve dok se ne generira signal koji označava kraj niza instrukcija, odnosno programa

Model procesora RISC arhitekture



Usporedba CISC i RISC

- CISC –Complex Instruction Set Computers
 - Arhitektura računala sa **kompleksnim** skupom instrukcija
- RISC –Reduced Instruction Set Computers
 - Arhitektura računala sa **reduciranim** skupom instrukcija

Obilježja CISC arhitekture

- Bogati skup instrukcija, od jednostavnih do vrlo složenih
- Instrukcija se izvršava u više od jednog takta
- Instrukcije različitih dužina
- Različiti i složeni načini adresiranja memorije
- Mikroprogramirana upravljačka jedinica
- Bez cjevovodne (no pipelining) arhitekture

Obilježja CISC arhitekture

- Dobre performanse sa jednostavnim prevodiocima (složene instrukcije)
- Segmentirana memorija
- Malo registara
- Slabije performanse FPU (od engl. FloatingPoint Unit) –jedinica za operacije s pomičnim zarezom
- **Kompatibilnost sa prethodnicima**

Predstavnici CISC arhitekture

- DEC VAX
- IBM System/360, System/370
- Burroughs B5000, B6000, B7000
- Motorola MC6800, MC68000, ..., MC68040,...
- Intel 8080, 8086, 80286, 80386, 80486, Pentium

Obilježja RISC arhitekture

- Jednostavne instrukcije i malo načina adresiranja
- Instrukcije se izvode u jednom taktu
- Uniformirana dužina instrukcija i fiksni instrukcijski format
- Adresiranje memorije sa “load” i “store”
- Sklopovska realizacija upravljačke jedinice
- Cjevovodna (pipelining) arhitektura

Predstavnici RISC arhitekture

- HP PA-RISC
- University Berkeley: RISC I , RISC II
- Stanford University: MIPS R2000
- AcornARM 6
- IBM 801, RS6000, RT-PC
- Intel i860, i960
- Motorola 88K
- Motorola/IBM PowerPC
- Sun SPARC

Predstavnici RISC arhitekture

- danas najpoznatiji – **ARM (Advanced RISC) procesori**
- temelje na dizajnu britanske korporacija *ARM Holdings* koja ne proizvodi ove procesore nego ih samo dizajnira i potom naplaćuje licencu od proizvođača

Primjeri izvedbe ARM arhitekture

- Apple Axx.. procesori – iPhone, iPad
- Samsung – mobilni uređaji, tableti
- Sustavi na jednom procesoru:
 - Qualcomm Snapdragon
 - Raspberry Pi
 -
- Televizori, automobili, mrežni uređaji, gotovo svi danas poznati pametni telefoni
...

Primjer RISC procesora

- MIPS R2000 – mikroprocesor razvijen od strane tvrtke MIPS Computer Systems
- Koristit ćemo ga kao primjer na predavanjima iz ovog predmeta
- Zašto je važan – prvi komercijalni RISC procesor, predstavljen 1986. godine

Bitna svojstva MIPS R2000

- 32 registra opće namjene po 32 bita
- Programsko brojilo –32 bita
- Dva 32 bitna registra koja sadrže rezultate cjelobrojnog dijeljenja ili množenja
- Procesor izravno podržava tri tipa podataka:
 - 32-bitnu riječ
 - 16-bitnu riječ
 - 8-bitni bajt

Formati instrukcija MIPS R2000

- Instrukcije su sve iste duljine –32 bita
- Postoje samo tri formata instrukcija:
 - I-tip (Immediate) –sadrži 16-bitni usputni podatak
 - J-tip (Jump) –26 bita koristi za ciljnu adresu grananja
 - R-tip (Register) –troadresni format

	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
Register	op	reg1	reg2	des	shift	funct
Immediate	op	reg1	reg2	16-bit constant		
Jump	op	26-bit constant				

Formati instrukcija MIPS R2000

- Prvih šest bitova svake instrukcije nazivaju se **op poljem**. Polje op određuje o kojoj se vrsti instrukcije radi (register, immediate, jump)
- Ovisno o tome što je op, dijelovi ostatka instrukcije mogu predstavljaju imena registara, konstantne memorijske adrese, 16-bitne cijele brojeve ili opis druge op instrukcije

	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
Register	op	reg1	reg2	des	shift	funct
Immediate	op	reg1	reg2	16-bit constant		
Jump	op	26-bit constant				

Formati instrukcija MIPS R2000

- Ako je op polje 0, tada se radi o instrukciji registra (engl. Register) koja općenito izvršava zadanu aritmetičku ili logičku operaciju
- Polje *funct* definira operaciju koja se izvodi, dok polja *reg1* and *reg2* predstavljaju adrese registara nad kojim se izvršava zadana operacija
- Polje *des* predstavlja adresu registra u kojem se pohranjuje rezultat

	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
Register	op	reg1	reg2	des	shift	funct

Primjer instrukcije MIPS 2000

- Zbrajanje vrijednosti registara s adresama 20 i 17 i spremanje rezultata u registar na heksadekadskoj adresi 16*

- Instrukcija binarno:

0000 0010 1001 0001 1000 0000 0010 0000

Field	op	reg1	reg2	des	shift	funct
Width	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
Values	0	20	17	16	0	add
Binary	000000	10100	10001	10000	00000	100000

- Instrukcija heksadekadski:

0x02918020