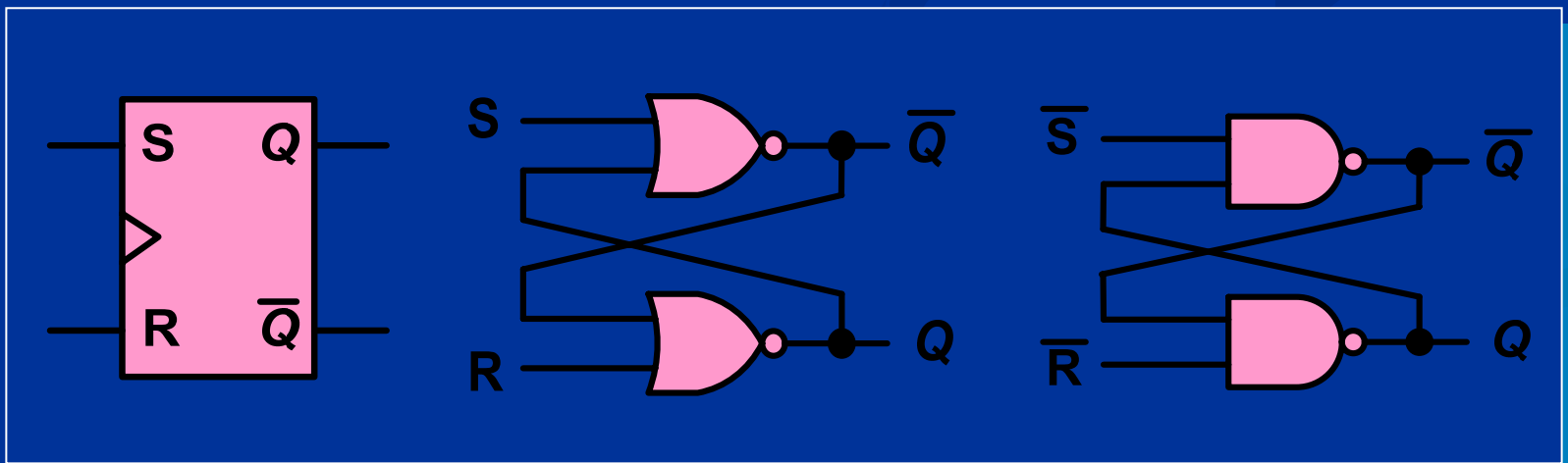


Elementarna memorijska kola

- Memorijska kola mogu da zapamte prethodno stanje
- Flip-flop je logička mreža koja može da zapamti samo jedan bit podatka (jednu binarnu cifru)
 - flip - flop je kolo sa dva stabilna stanja
- Rad flip - flopa može da se opiše:
 - tabelom stanja
 - analitički logičkom funkcijom
- Rad flip - flopa može da se sinhronizuje sa signalom takta (*clock*)

■ RS-FF je memorijsko kolo koje ima priključke za **S**et i **R**eset

- Realizuje se pomoću logičkih kola
- Sastoji se od ukrštene veze izlaza i ulaza dva NILI ili dva NI kola

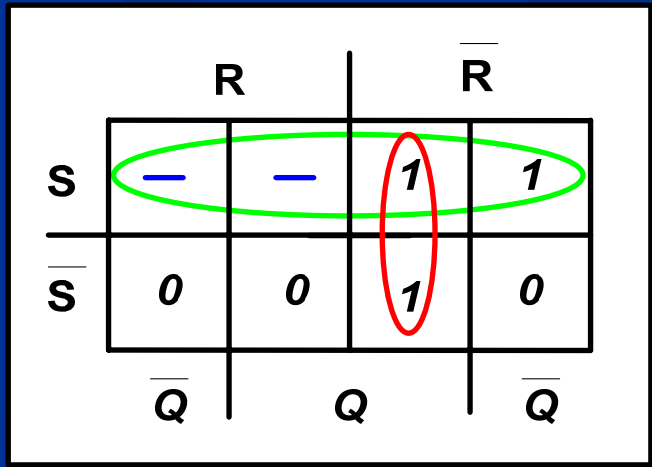


ULAZNE PROMENLJIVE			IZLAZNE PROMENLJIVE
$t = t_n$			$t = t_{n+1}$
R	S	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	ND
1	1	1	ND

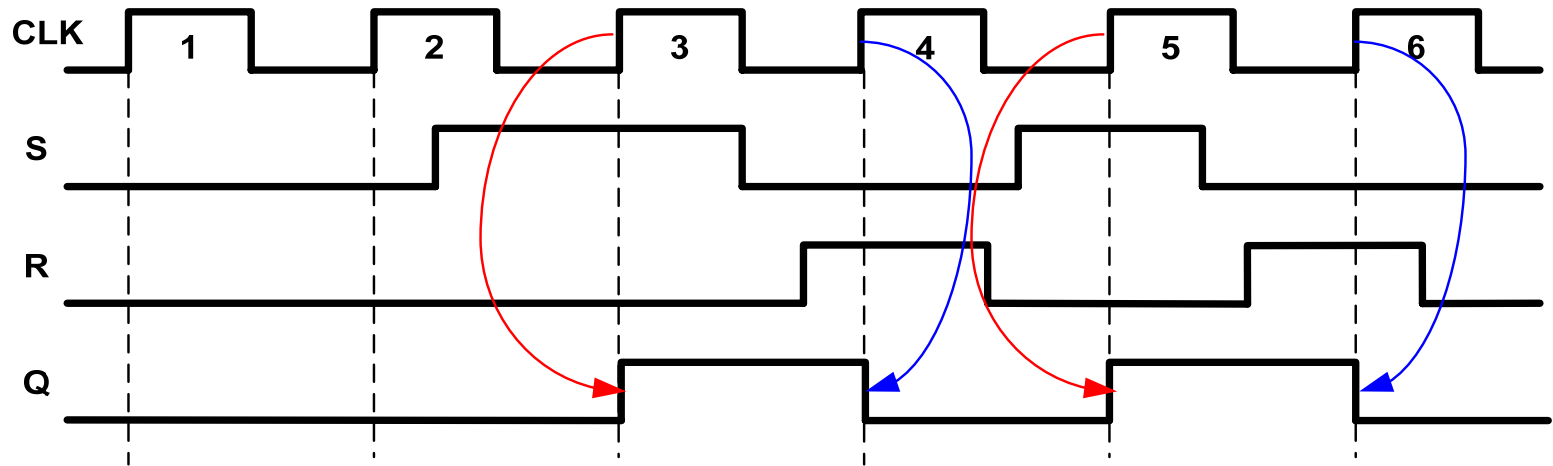
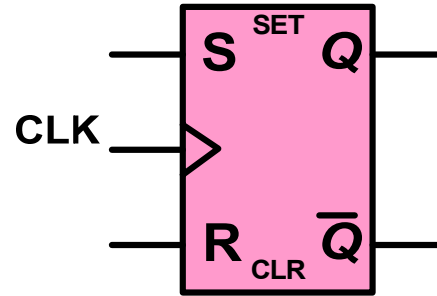
FUNKCIONALNA TABELA ZA RS-FF
SA NILI LOGIČKIM KOLIMA

$$Q_{n+1} = S + \bar{R}Q_n$$

$t = t_n$		$t = t_{n+1}$
R	S	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	ND



■ Sinhroni RS-FF

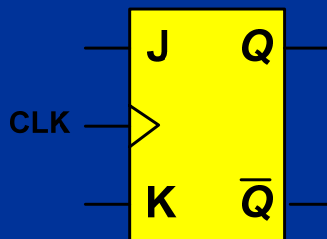


JK-FF nema nedefinisana stanja

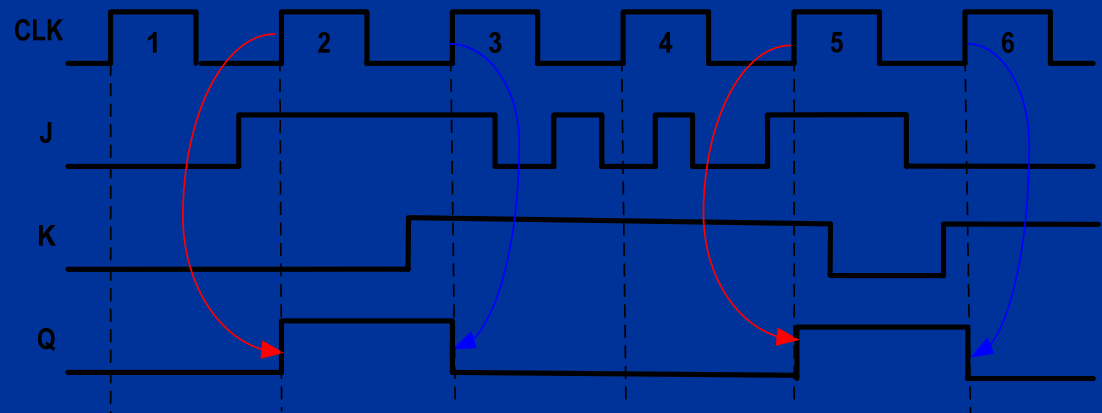
ULAZNE PROMENLJIVE			IZLAZNE PROMENLJIVE
$t = t_n$			$t = t_{n+1}$
J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

FUNKCIONALNA TABELA

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

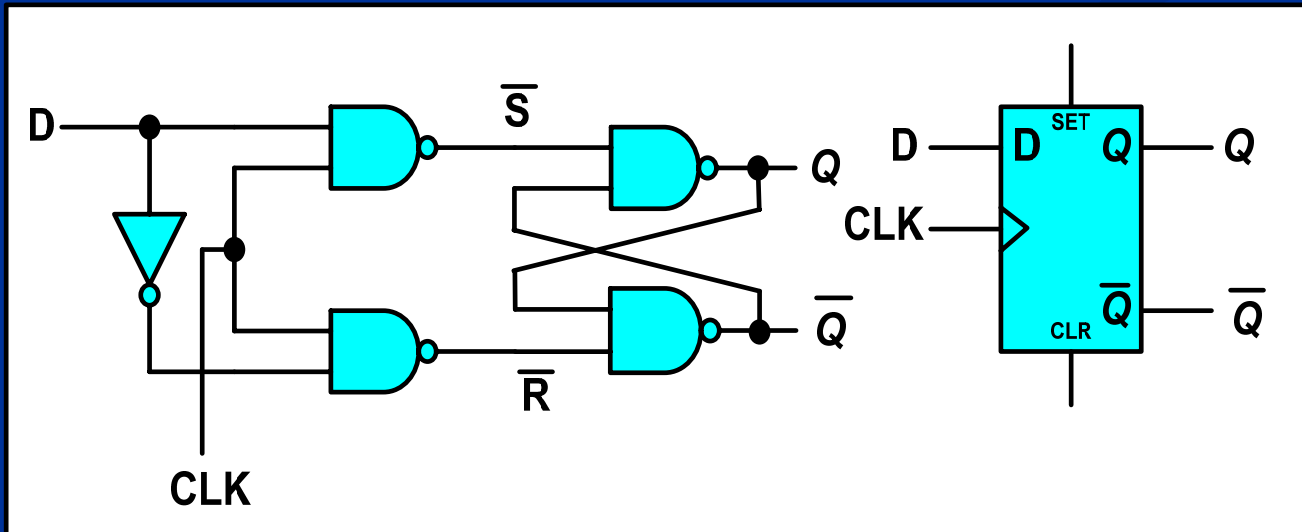


$t = t_n$		$t = t_{n+1}$
J	K	Q_{n+1}
0	0	Q_n
0	1	
1	0	
1	1	



■ D-FF ima samo jedan ulazni priključak

- Pogodniji je za prihvatanje i prenos podataka od flip-flopora sa dva ulaza
- Koristi se kao osnovna komponenta za realizaciju drugih memorijskih kola

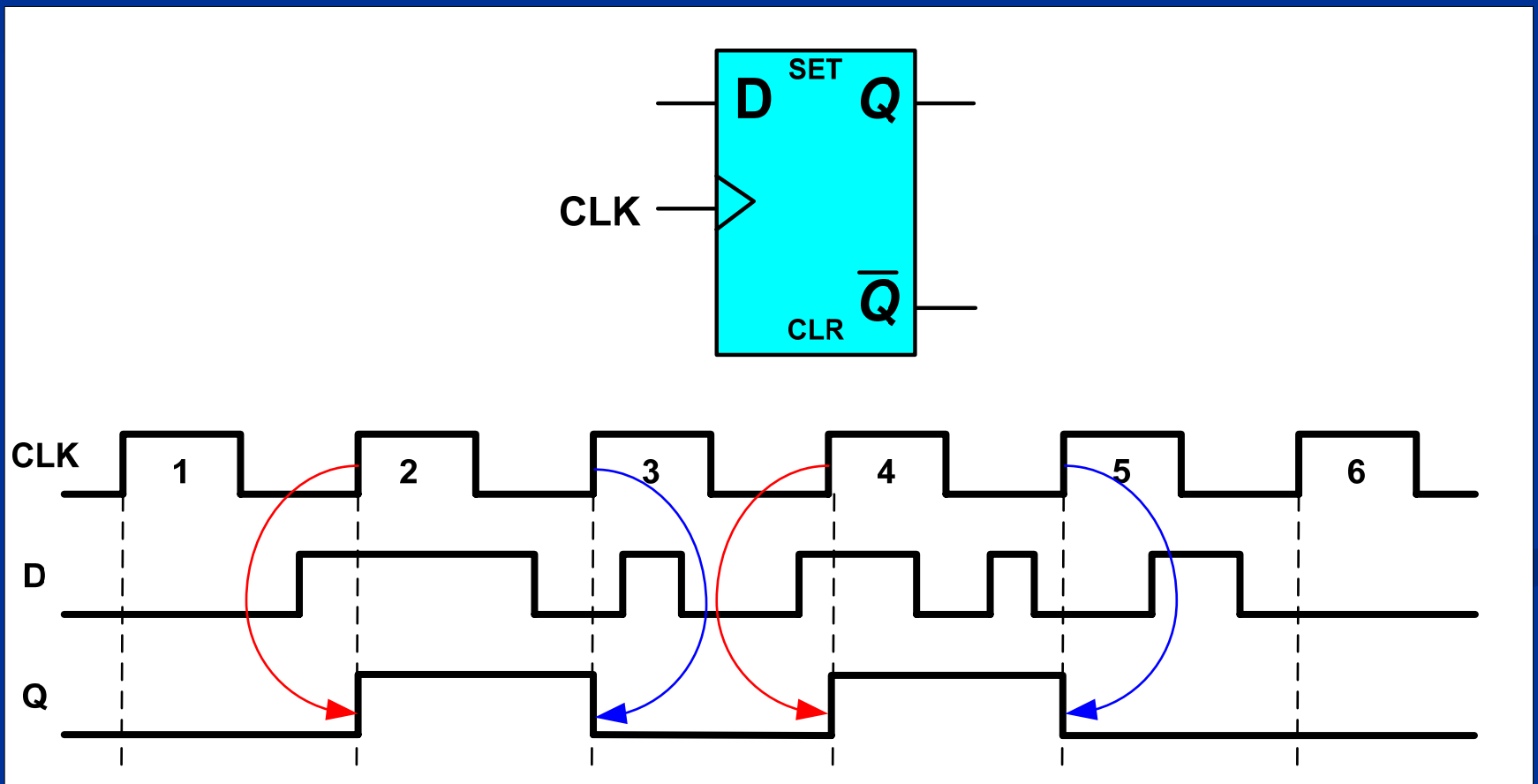


t = t _n		t = t _{n+1}
D	Q _n	Q _{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

FUNKCIONALNA TABELA

$$Q_{n+1} = D\bar{Q}_n + DQ_n = D$$

■ Sinhroni D-FF

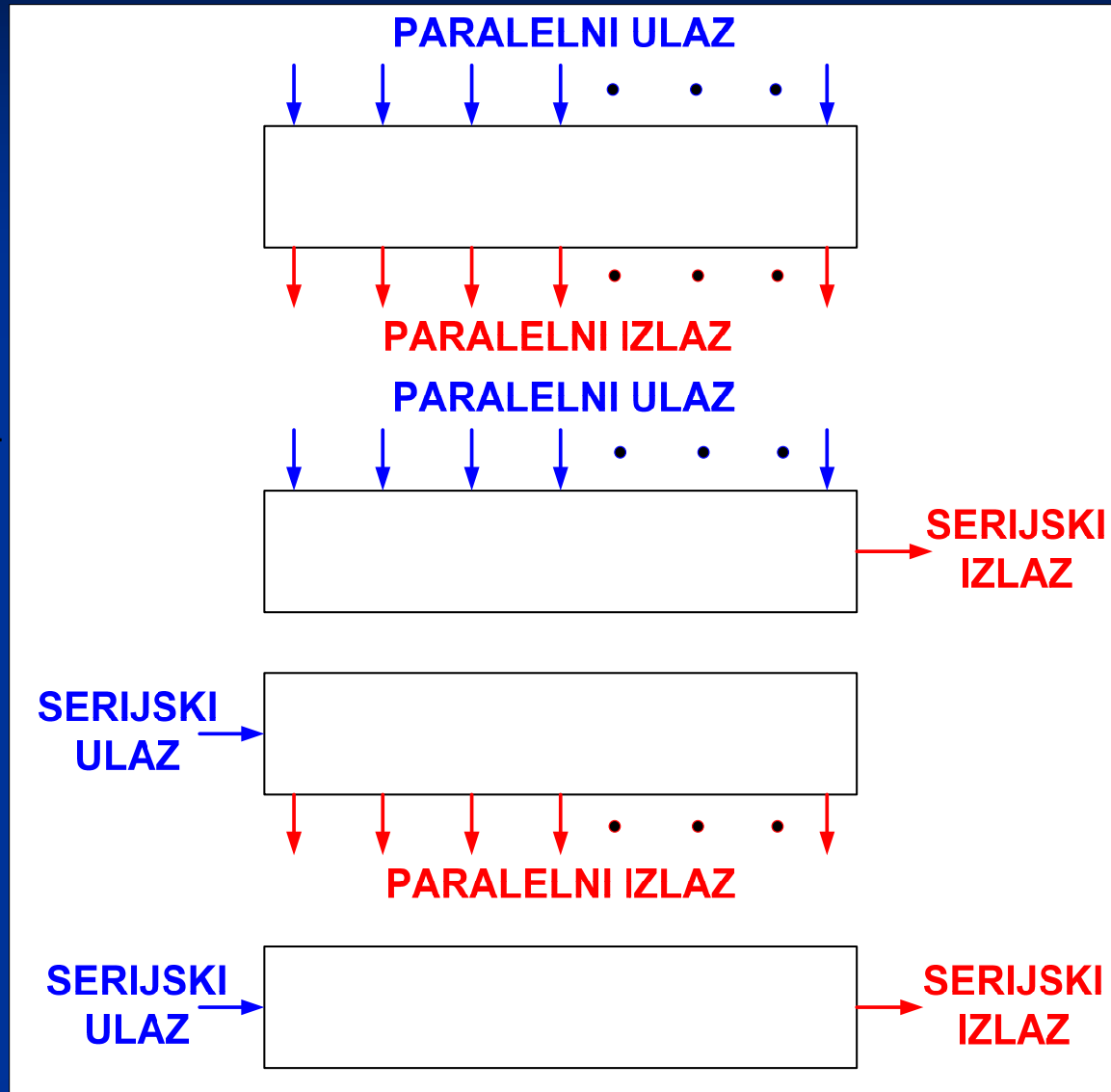


Registri

- **Registar je element koji služi za čuvanje proizvoljnog binarnog broja ograničene dužine**
 - Za binarni broj od n cifara potrebno je n memorijskih ćelija
 - Binarni broj u registru je sadržaj registra
- **Primena registara**
 - Veza između blokova sa različitim brzinama
 - Pri realizovanju aritmetičkih operacija
 - Pretvaranje serijskog u paralelni kod i obrnuto...

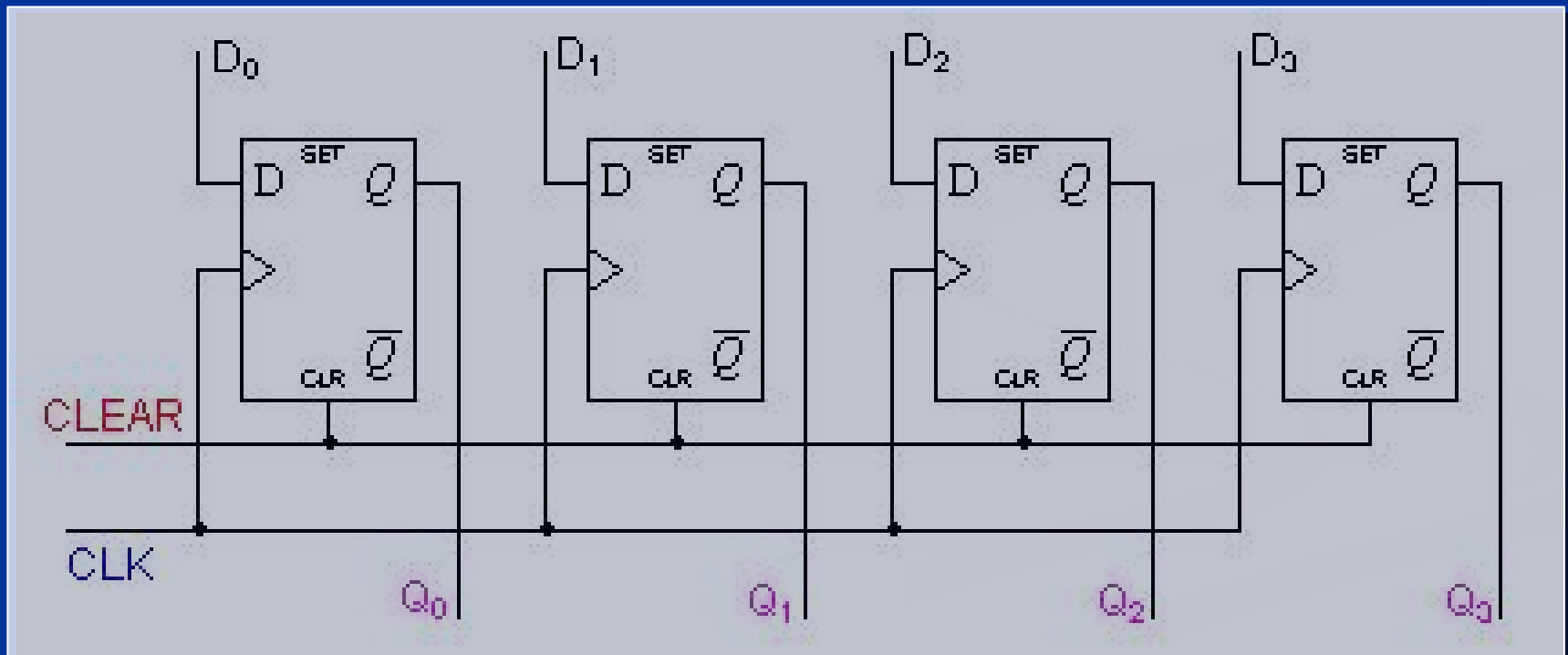
Registri

- **Upis i čitanje sadržaja registra**
 - Paralelno
 - Serijski
- **Svi registri u CPU dele se na:**
 - Registre opšte namene
 - Registre specijalne namene



Registri

- Realizacija 4-bitnog registra sa paralelnim ulazom i paralelnim izlazom
 - Podatak se privremeno čuva (baferiše)

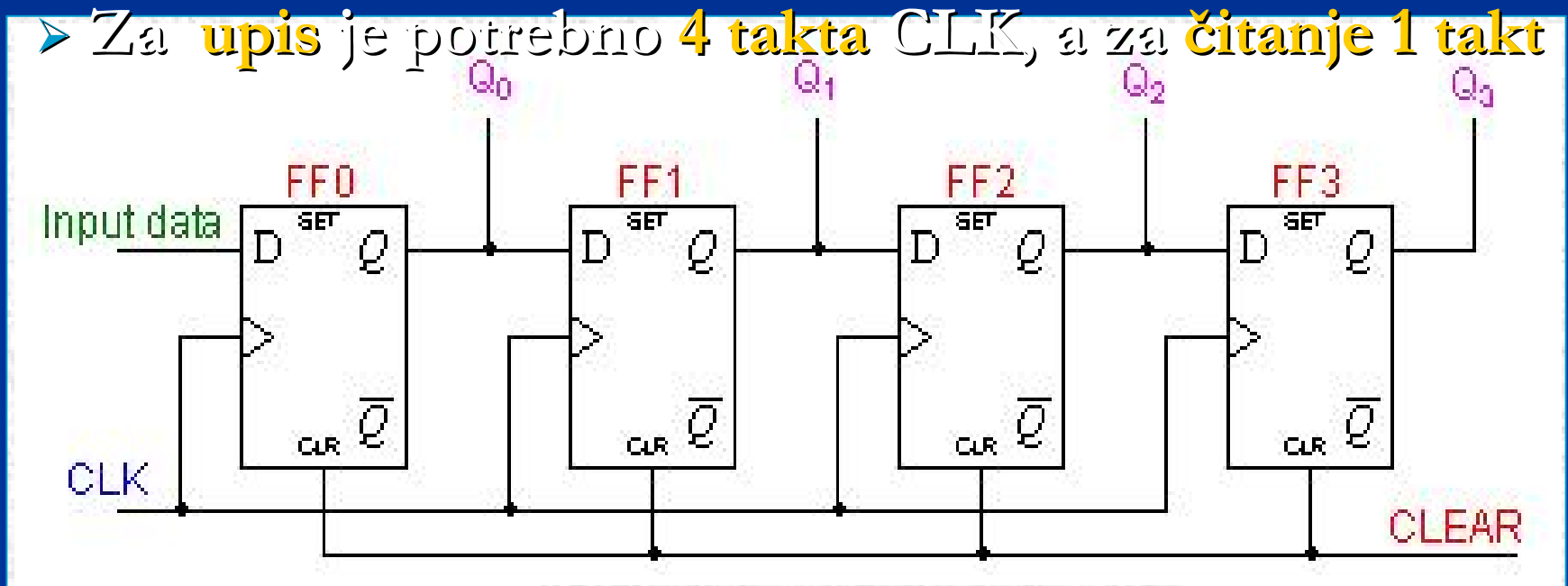


Registri

Realizacija 4-bitnog registra sa serijskim ulazom i paralelnim izlazom

Serijska informacije se pretvara u paralelnu

Za **upis** je potrebno **4 takta** CLK, a za **čitanje** 1 takt



Upis podatka: 1001

CLEAR
1001

Q0	Q1	Q2	Q3
0	0	0	0

Q0	Q1	Q2	Q3
1	0	0	0

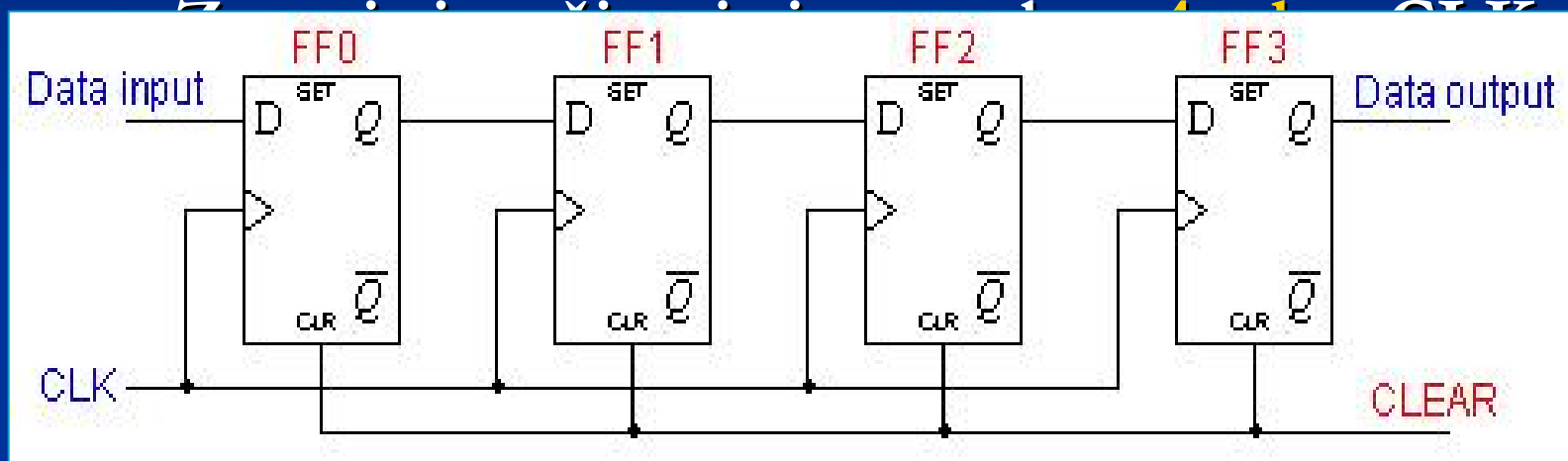
Q0	Q1	Q2	Q3
0	1	0	0

Q0	Q1	Q2	Q3
0	0	1	0

Q0	Q1	Q2	Q3
1	0	0	1

Registri

- Realizacija 4-bitnog registra sa serijskim ulazom i serijskim izlazom



Upis podatka: 1001

FF0	FF1	FF2	FF3
0	0	0	0

FF0	FF1	FF2	FF3
0	0	0	0

FF0	FF1	FF2	FF3
1	0	0	0

FF0	FF1	FF2	FF3
0	1	0	0

FF0	FF1	FF2	FF3
0	0	1	0

FF0	FF1	FF2	FF3
1	0	0	1

Čitanje podatka: 1001

FF0	FF1	FF2	FF3
1	0	0	1

FF0	FF1	FF2	FF3
0	1	0	0

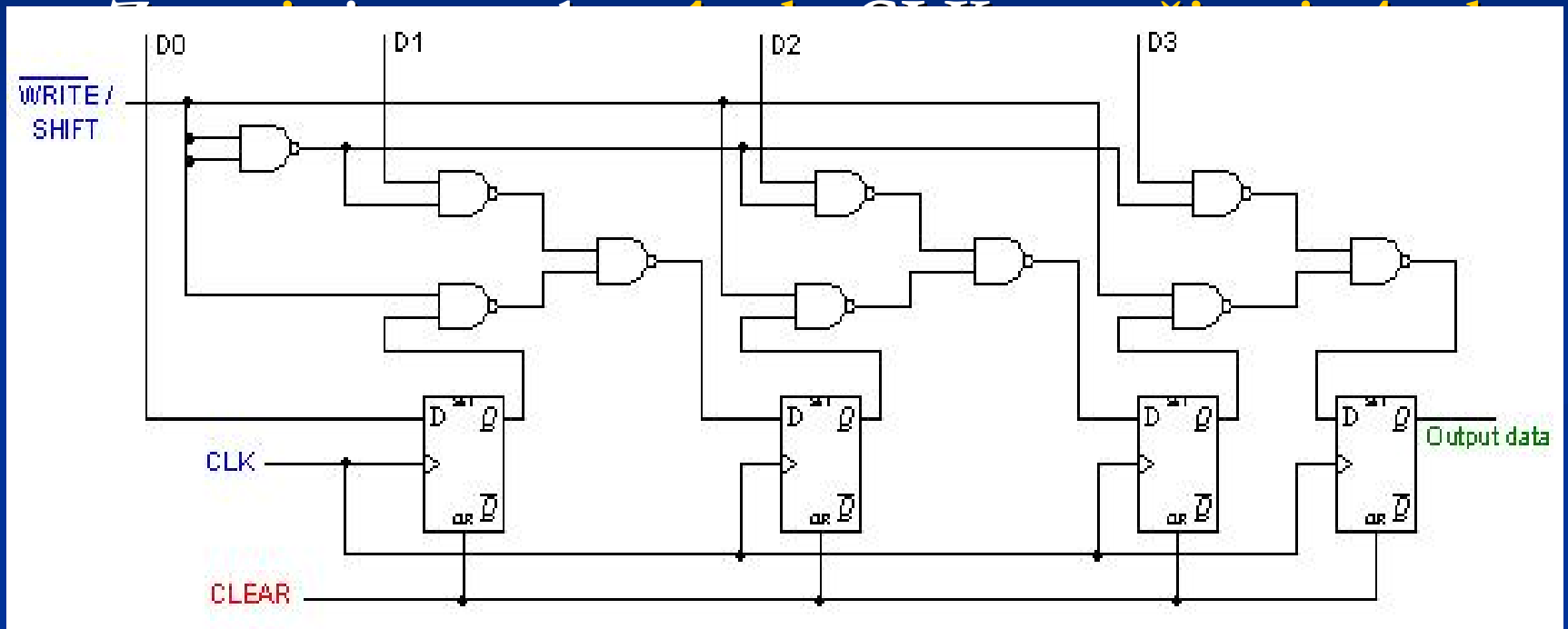
FF0	FF1	FF2	FF3
0	0	1	0

FF0	FF1	FF2	FF3
0	0	0	1

FF0	FF1	FF2	FF3
0	0	0	0

Registri

- Realizacija 4-bitnog registra sa paralelnim ulazom i serijskim izlazom



CLEAR = 1

Q0	Q1	Q2	Q3
0	0	0	0

WRITE = 0

Q0	Q1	Q2	Q3
1	0	0	1

WRITE = 0

Q0	Q1	Q2	Q3
1	0	0	1

SHIFT = 1

Q0	Q1	Q2	Q3
1	1	0	0
1	0	0	1

SHIFT = 1

Q0	Q1	Q2	Q3
1	1	1	0
1	0	0	1

SHIFT = 1

Q0	Q1	Q2	Q3
1	1	1	1
1	0	0	1

SHIFT = 1

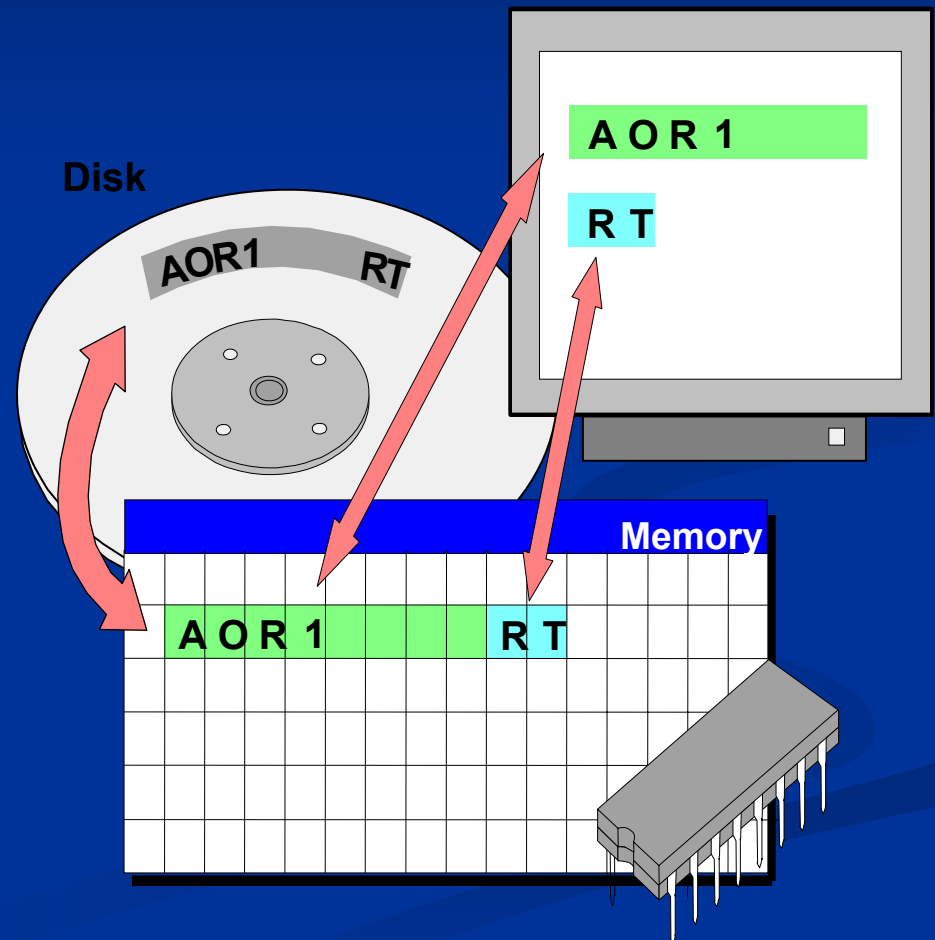
Q0	Q1	Q2	Q3
1	1	1	1
1	0	0	1

Podela registara prema nameni

- Registri opšte namene ($R_0 - R_N$)
 - Privremeno čuvanje podataka različite namene
- Registri specijalne namene:
 - Akumulatori (*accumulator*)
 - Čuvaju rezultate operacija sa binarnim brojevima
 - Međuregistri (*buffers*)
 - Pomoćni registri za prihvatanje podataka iz akumulatora, sa magistrale, iz memorije, za čuvanje memorijske adrese...
 - Registar stanja (*status register, SR*) ili registar uslova (*condition code register, CCR*)
 - Niz nezavisnih FF-ova (zastavica, flag-ova) za prikazivanje različitih stanja nastalih tokom obrade podataka
 - Adresni registri (*address registers*)
 - čuvaju adrese memorijskih lokacija u kojima se nalaze podaci ili instrukcije (brojač instrukcija)

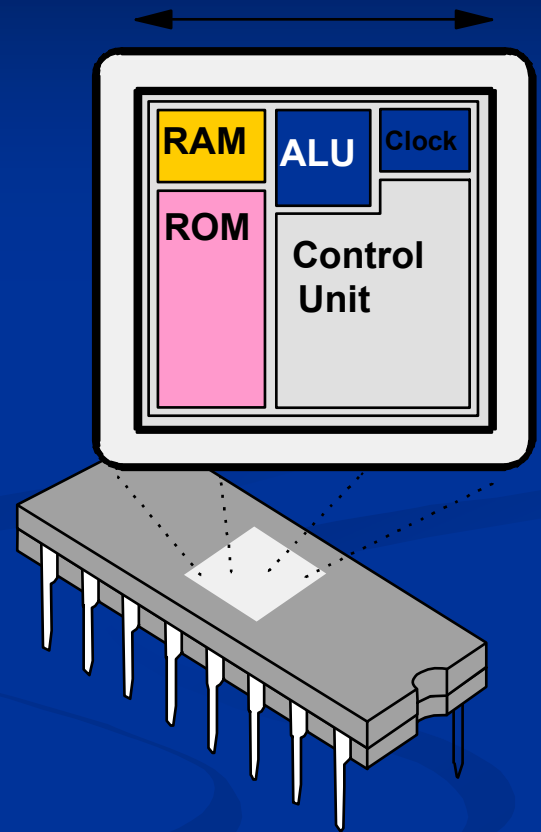
Memorije

- U okviru CPU nalaze se primarne memorije RAM i ROM tipa i registri
- Sekundarne memorije (za čuvanje velike količine podataka) nalaze se izvan CPU



Memorije

- Veliki broj registara povezanih u celinu čini memoriju
- Memorije služe za smeštanje binarnih podataka i programskih instrukcija
- **Kapacitet memorije** je broj adresabilnih lokacija na kojima mogu da se čuvaju podaci
 - $1\text{MB} = 2^{20} = 1,048,576$ lokacija
 - $1\text{GB} = 2^{30} = 1,073,700,000$ lokacija



Memorije

- **Pristup memoriji** (*memory access*) je čitanje ili upis podataka
 - Što je kraće vreme pristupa memorija je brža
- **Prema vremenu pristupa razlikuju se :**
 - Sekvencijalne memorije (magnetne trake, CCD, ...)
 - Memorije sa direktnim pristupom (RAM, ROM)

Memorije

■ Organizacija memorije:

REDNI BROJ
VRSTE

PODATAK

00	
01	
02	
03	
...	
FD	
FE	
FF	

STRANICA
00

REDNI BROJ
VRSTE

PODATAK

00	
01	
02	
03	
...	
FD	
FE	
FF	

STRANICA
01

REDNI BROJ
VRSTE

PODATAK

00	
01	
02	
03	
...	
FD	
FE	
FF	

STRANICA
FF

Memorije

- Adresa lokacije podatka prikazana u obliku rednog broja vrste je **fizička adresa** podatka
- Podacima i adresama mogu da se dodele simbolička imena
- Svaki računarski sistem ima razne vrste memorijskih uređaja

FIZIČKA
ADRESA

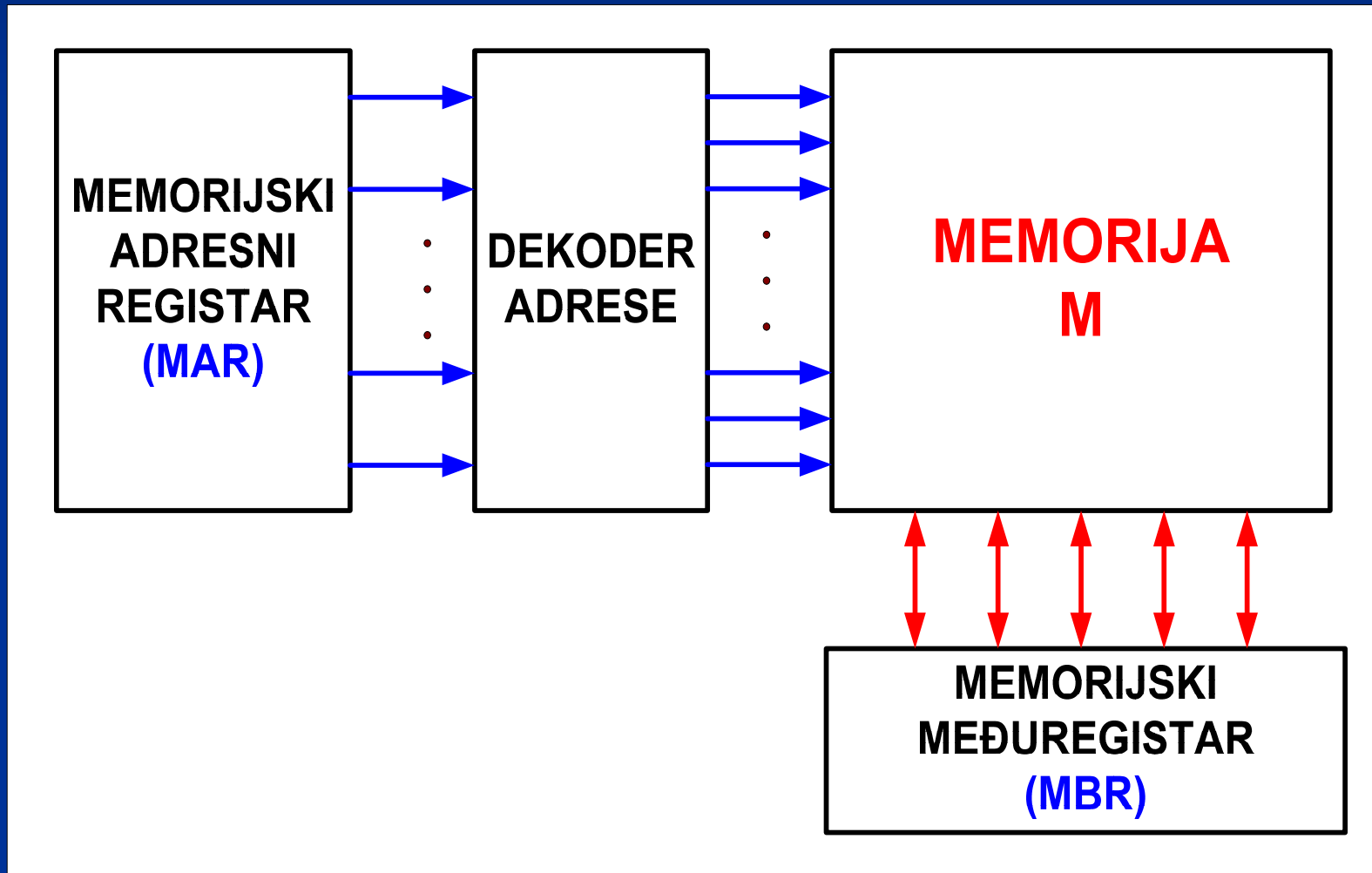
PODATAK

0000	
0001	
0002	
0003	
. . .	
00FE	
00FF	
0100	
. . .	
01FE	
01FF	
0200	
. . .	
FFFE	
FFFF	

GLAVNA MEMORIJA
RAČUNARA

Memorije

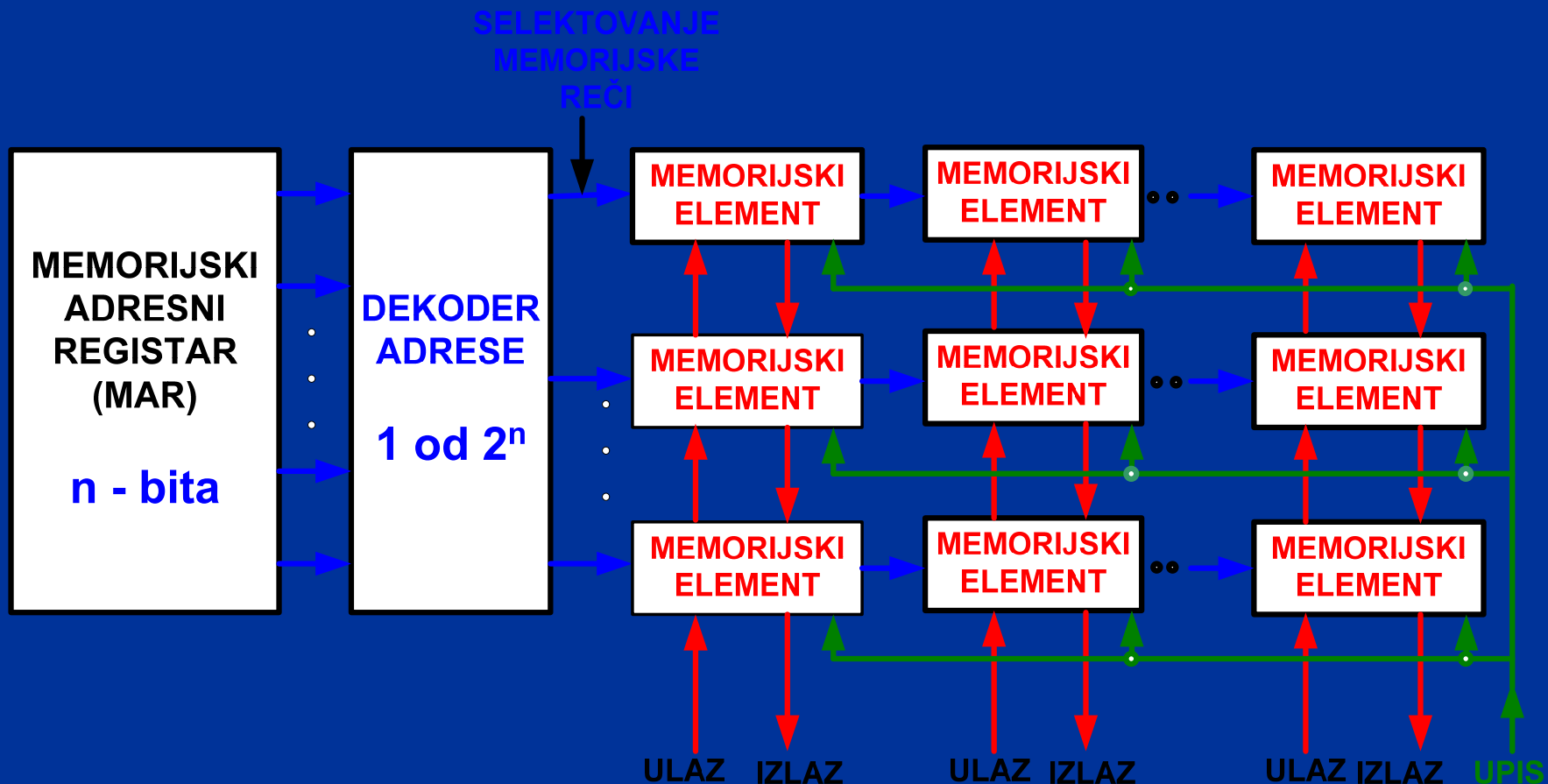
- Opšta blok-šema memorije sa dekoderom adrese i registrima



Memorije

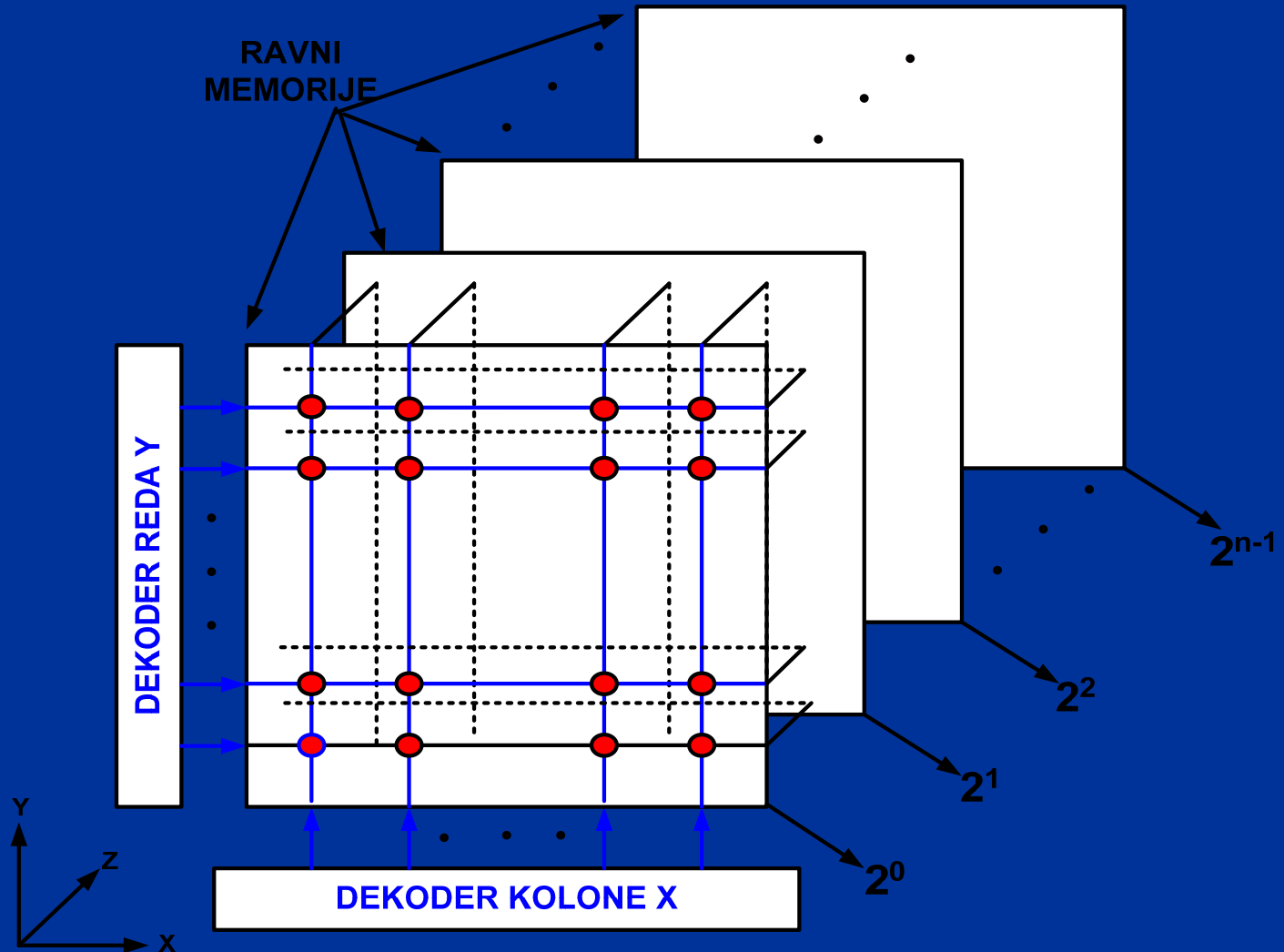
■ Dvodimenzionalna organizacija memorije

- Od n adresnih bita iz adresnog registra dobija se 2^n adresnih linija sa dekodera adrese



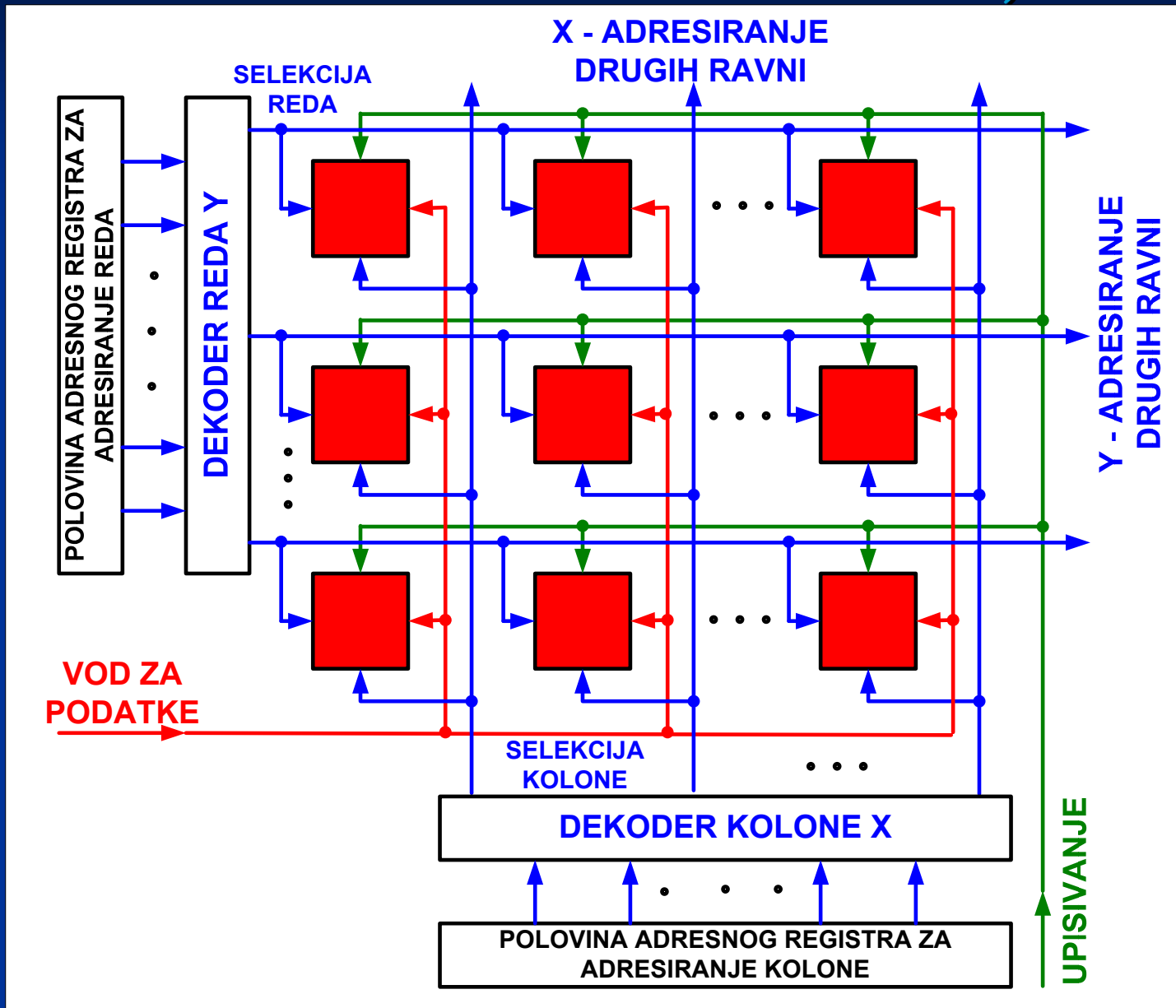
Memorije

- Trodimenzionalna organizacija memorije
 - Adrese čine 2 dimenzije, a dužina reči treću



Memorije

■ Jedna ravan trodimenzionalne memorije



Memorije

REGISTRI U
CPU

(*REGISTERS*)

SKRIVENA
MEMORIJA

(*CACHE STORAGE*)

OPERATIVNA (GLAVNA,
PRIMARNA) MEMORIJA

(*PRIMARY STORAGE*)

PROŠIRENA MEMORIJA

(*EXTENDED STORAGE*)

SKRIVENI DISK

(*CACHE DISK*)

STALNO AKTIVNE SEKUNDARNE MEMORIJE

(*ON-LINE SECONDARY STORAGE*)

POVREMENO AKTIVNE SEKUNDARNE MEMORIJE

(*OFF-LINE SECONDARY STORAGE*)

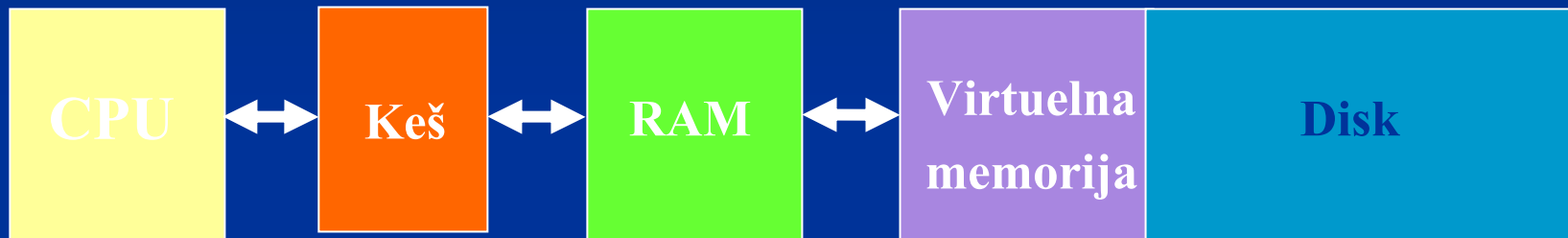
Memorije

■ Po uključanju napajanja:

- Računar učitava podatke iz **ROM** memorije i vrši početno testiranje rada osnovnih delova sistema
- Iz ROM memorije se učitava BIOS (Basic Input/Output System)
- Operativni sistem se učitava sa hard diska u **RAM**
- Sve aplikacije i fajlovi koji se koriste u aplikaciji nalaze se u RAM-u
- Nakon zatvaranja aplikacije, rezultat obrade može da se sačuva na hard disku, a aplikacija se briše iz RAM-a

Memorije

■ Povezanost memorija u računarskom sistemu



- Virtuelna memorija je deo hard diska koji čuva kopiju sadržaja RAM memorije koji se ređe koristi i tako oslobađa RAM za nove aplikacije
- Keš memorija ubrzava rad CPU sa RAM memorijom

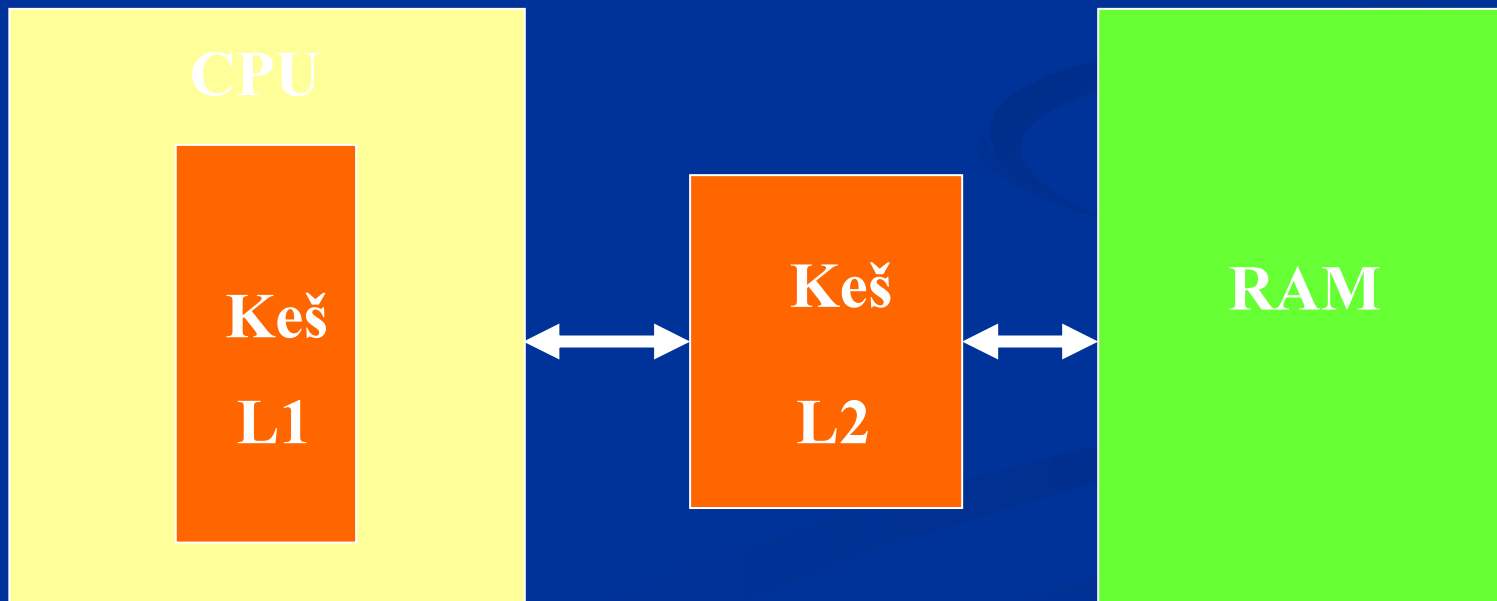
Memorije

- Unutrašnja (osnovna) memorija je veoma brza, ali je sporija od mikroprocesora
 - Keš-memorija je brža i služi kao posrednik između mikroprocesora i unutrašnje memorije
 - Može se nalaziti unutar samog mikroprocesora ili izvan njega



Memorije

- Kod savremenih računarskih sistema često je jedan deo keš memorije (prvi nivo, L1) u okviru mikroprocesora (CPU) i on je povezan sa kešom koji je izvan CPU (drugi nivo L2)



Memorije

- Operativna memorija je RAM (random access) tipa
 - Podaci u toku rada mogu da se upisuju u memoriju i da se čitaju iz nje sa proizvoljnih adresa
 - Poluprovodnička RAM memorija gubi sadržaj po isključenju napajanja



Memorije

- **SRAM (statički RAM) su nedestruktivne**
 - Zadržavaju svoj sadržaj i posle čitanja
 - Realizuju se pomoću FF-ova
- **DRAM (dinamički RAM) su destruktivne**
 - Posle čitanja se gubi podatak
 - podatak posle čitanja mora da se ponovo upiše
 - Realizuju se kao kapacitivnost MOS tranzistora
 - sadržaj memorije mora povremeno da se osvežava

Memorije

■ Tipovi DRAM memorije:

➤ SDRAM (*Synchronous DRAM*)

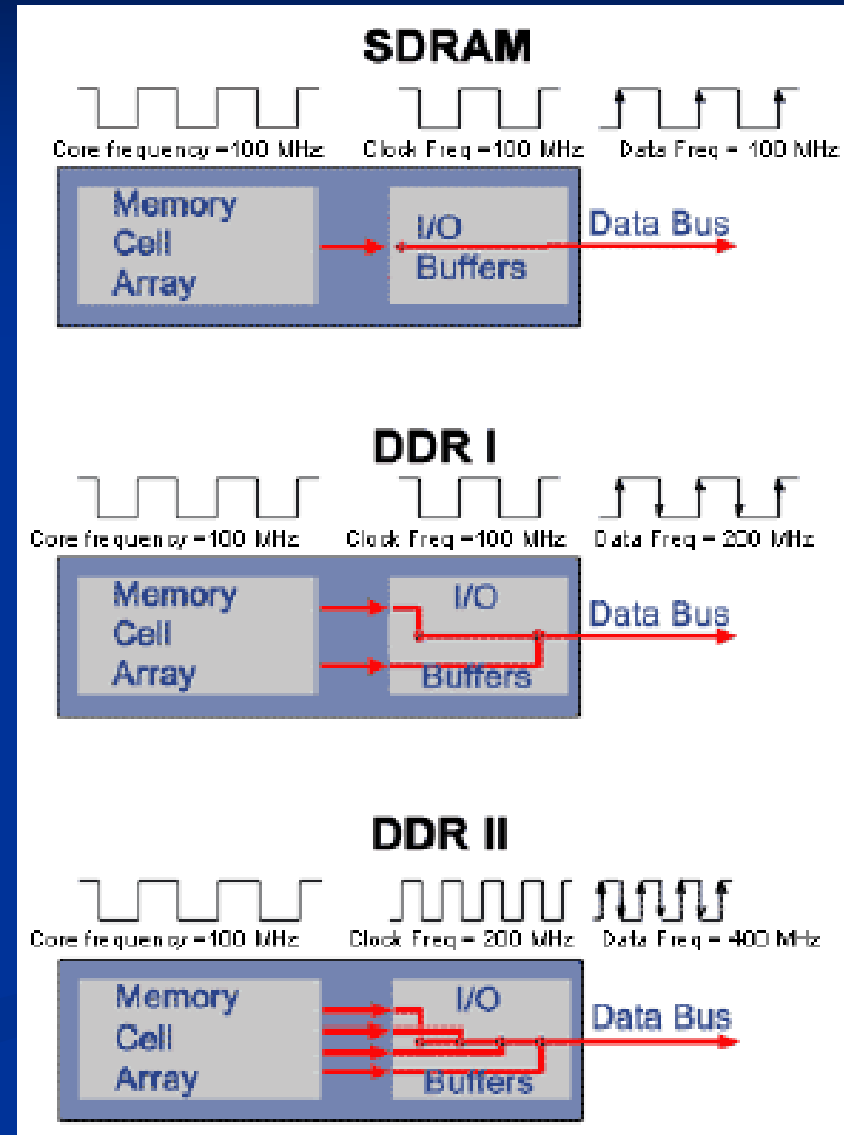
- ❑ Takt memorijskih čipova i CPU-a su uzajamno sinhronisani

➤ DDR1 SDRAM (*Double Data Rate SDRAM*)

- ❑ Podaci se prenose duplo većom brzinom u odnosu na SDRAM, jer je pristup na uzlaznoj i silaznoj ivici takta

➤ DDR2, DDR3 SDRAM

- ❑ Rade na većim učestanostima takta od DDR1 SDRAM-a



Memorije

■ Memorije ROM (read only) tipa

- Neizbrisive (**non - volatile**)
- Nedestruktivne (**non - destructive**)
- Koriste se za čuvanje stalnih programa

■ ROM

- Sadržaj je upisan u toku izrade čipa
- Isplativo za količinu preko 1000 komada
- Vreme pristupa 500 - 850ns

■ PROM

- Korisnik može da programira samo jednom pomoću uređaja za programiranje
- Konfiguracija sa bipolarnim poljem dioda ili sa bipolarnim tranzistorima ima
- Programiranje impulsima koji tope pregorljive metalizovane veze u PN-spoju (u polju dioda) ili vezu B-E (u tranzistoru)
- Vreme pristupa < 100ns

Memorije

■ EPROM

- Može da se programira i briše više puta
- Briše se UV zracima (RPPROM se briše električno)
- Izrađuju se primenom MOS tehnologije
- Kapacitet reda 64, 128, 256, 512KB
- Vreme pristupa reda 150 - 1200ns

■ EEPROM

- Moguće je čitanje i upis
- Operacija upisa je reda ms
- Kapacitet nekoliko desetina KB
- Operacije čitanja su reda μ s

Memorije

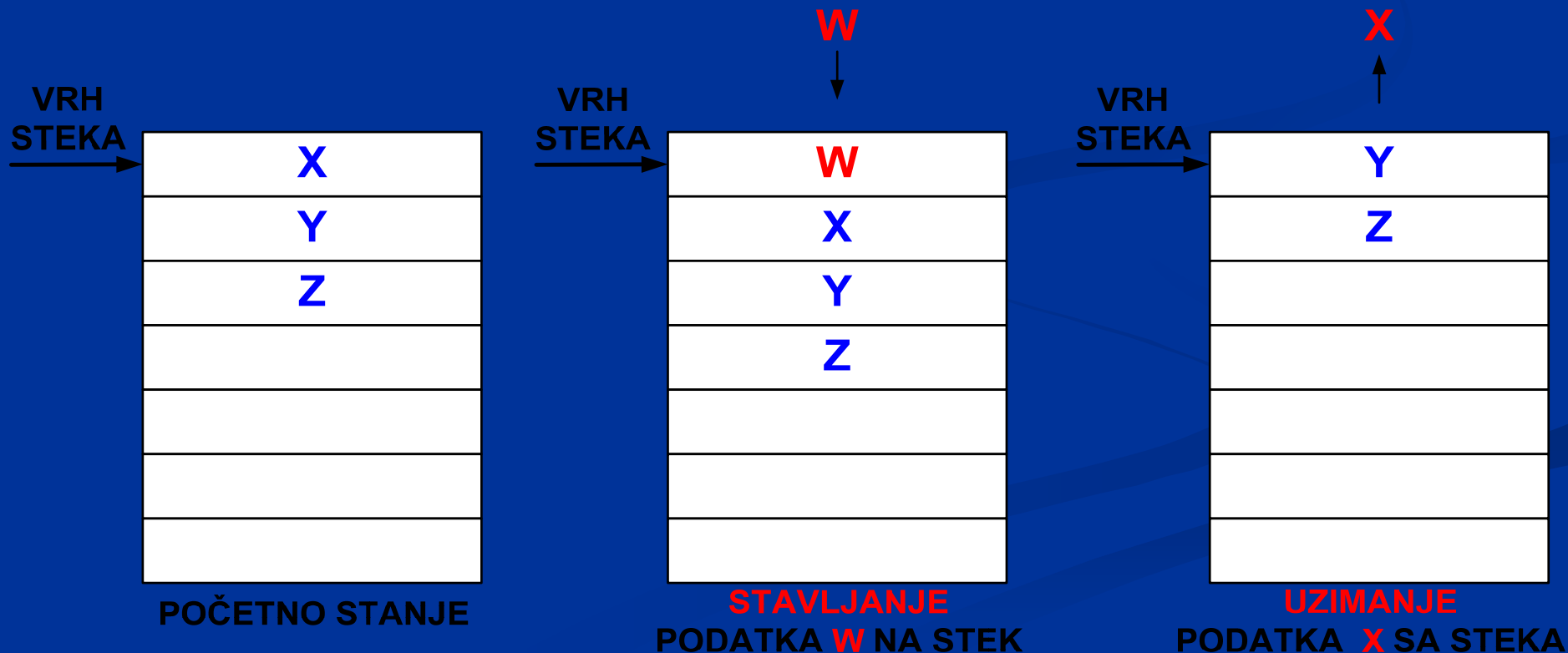
■ STEK memorija

- Sastoji se od niza registara koji su složeni jedan na drugi
- Podaci mogu da se upisuju ili čitaju samo po nekom definisanom redu
- Može da bude realizovana kao:
 - Softverski stek
 - Hardverski stek

Memorije

■ Hardverski realizovan stek

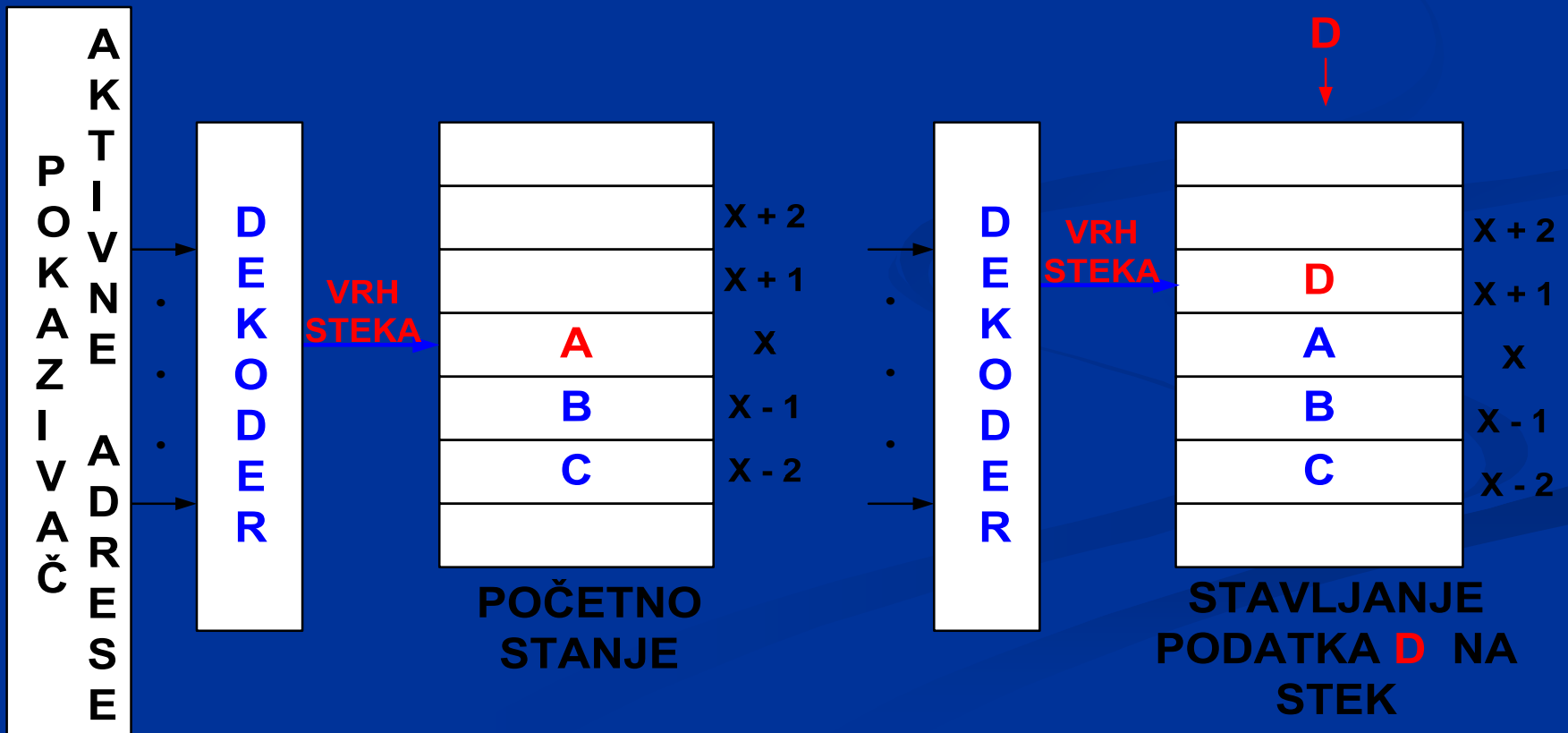
- Za upis ili čitanje dostupan je samo registar koji se nalazi na vrhu (**L**ast **I**n **F**irst **O**ut)
- Podatak se fizički pomera pri upisu i čitanju



Memorije

■ Softverski realizovan stek

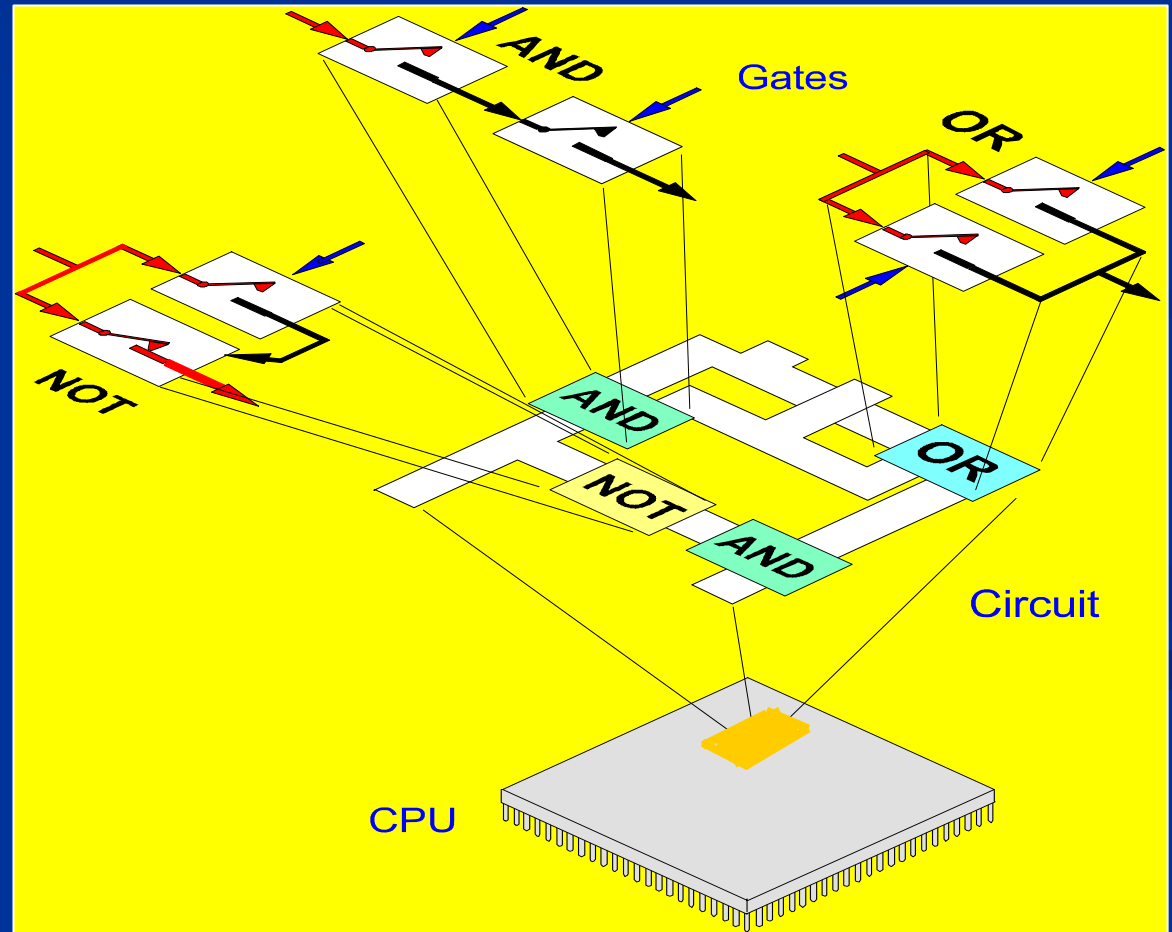
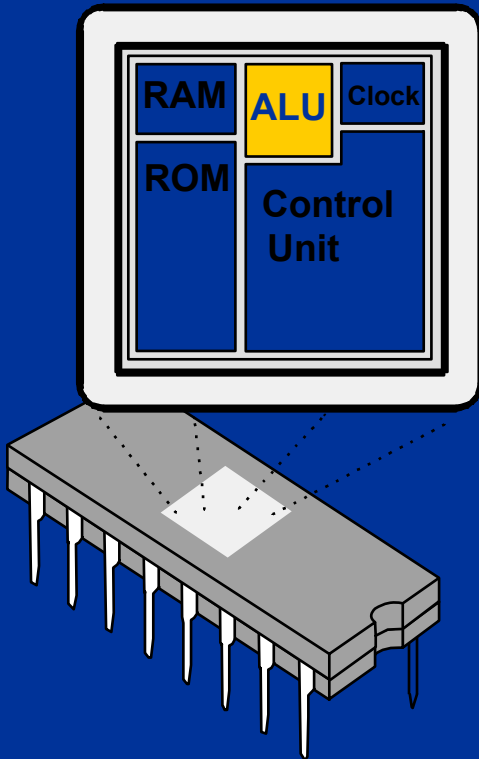
- Menja se adresa koja odgovara vrhu steka
 - Podaci u toku čitanja i upisa ne menjaju mesto
- mesto



Aritmetičko-logička jedinica (ALU)

■ Aritmetičko-logička jedinica (ALU)

- Izvršava aritmetičke i logičke instrukcije nad binarnim brojevima



Aritmetičko-logička jedinica (ALU)

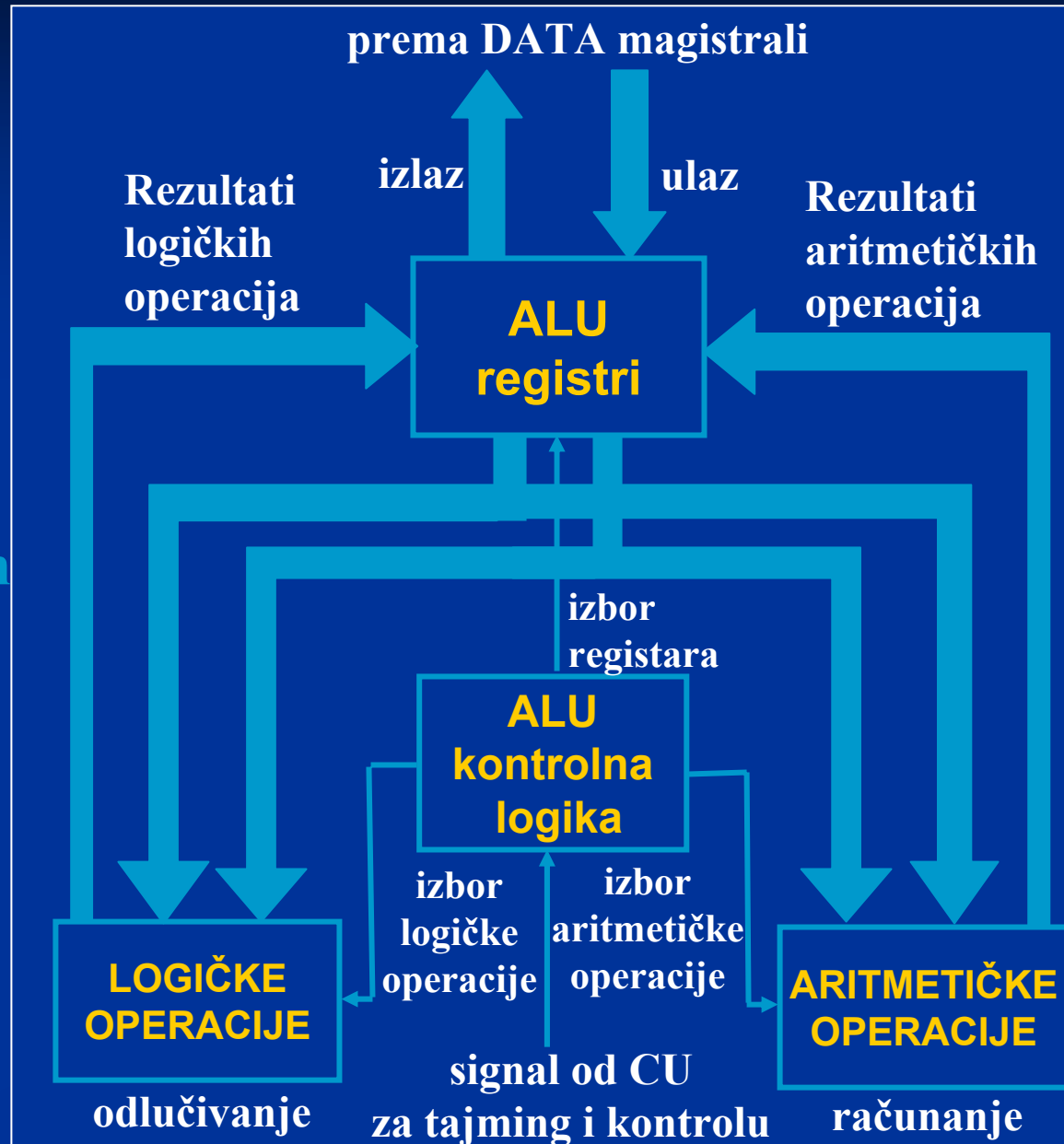
■ Aritmetičko-logička jedinica (ALU)

➤ Aritmetičke operacije

- Sabiranje, oduzimanje, množenje i deljenje binarnih i BCD brojeva sa pokretnim i nepokretnim zarezom

➤ Logičke operacije

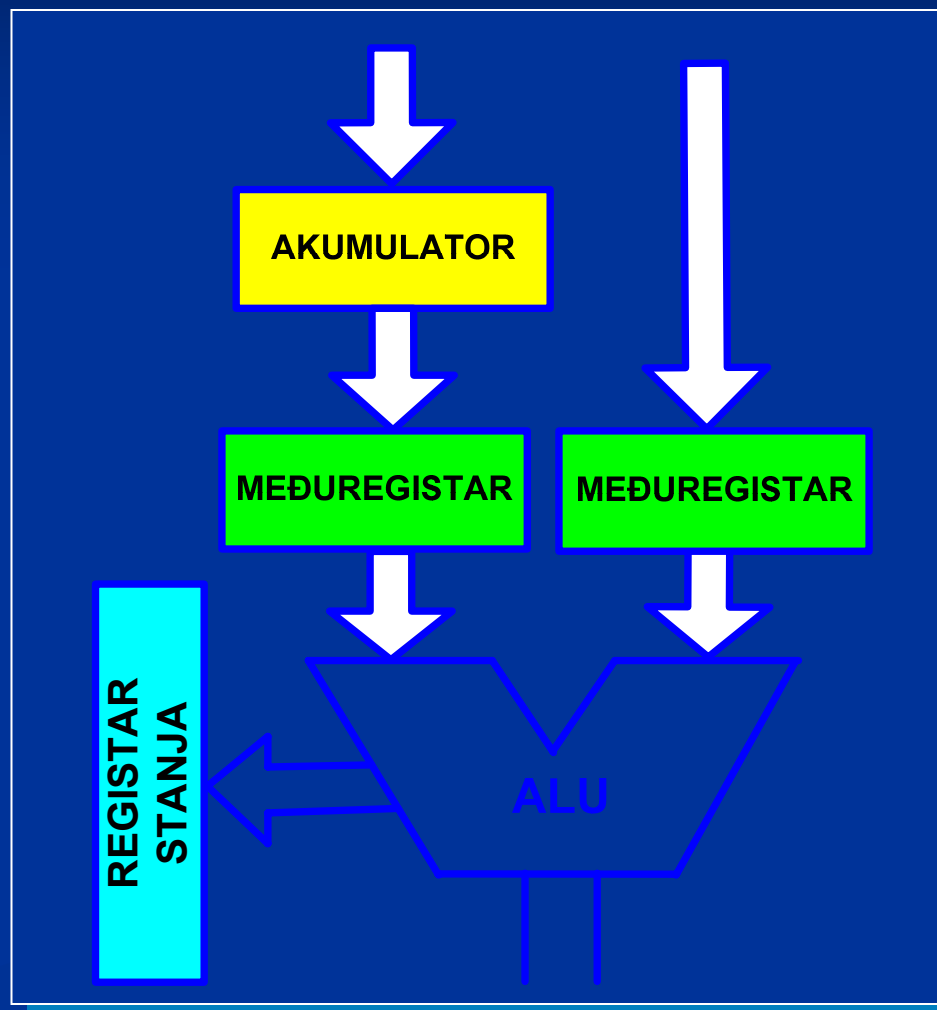
- Osnovne logičke operacije I, ILI, NE...
- Pomeranje, rotacija, komplementiranje,...



Aritmetičko-logička jedinica (ALU)

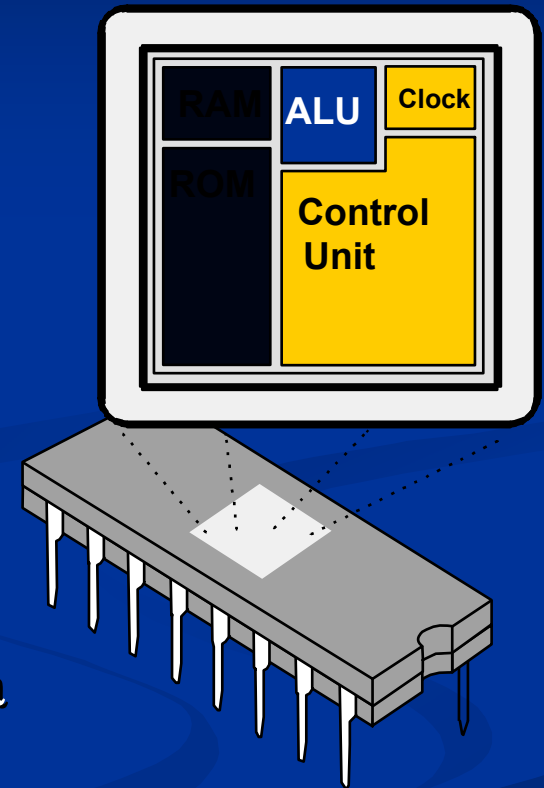
■ U sklopu ALU nalaze se i registri u koje se smeštaju operandi i rezultat operacije:

- Akumulatori
- Privremeni registri (međuregistri)
- Registar stanja...

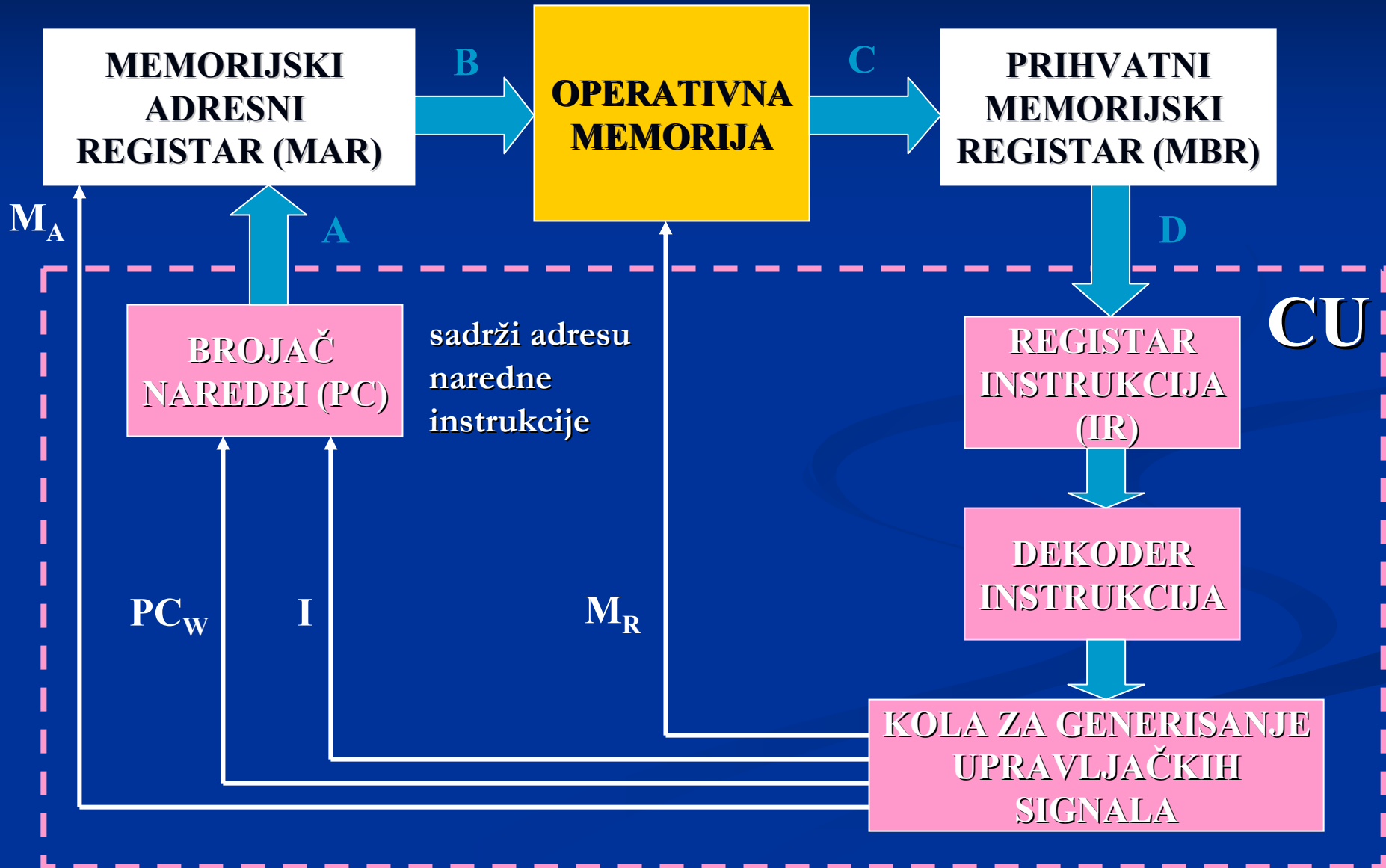


Realizacija upravljačkih jedinica (CU)

- U hardverskoj realizaciji CU upravljački signali se generišu pomoću posebnih digitalnih mreža
 - Ovakve upravljačke jedinice su brze
 - Često su vrlo složene
 - Nisu fleksibilne
 - Nisu dostupne korisniku da ih modifikuje prema svojim potrebama
 - Ovakvu organizaciju imaju **RISC** procesori (**R**educed **I**nstruction **S**et **C**omputers)



Hardverska realizacija CU



Upravljačkih jedinica (CU)

■ U mikroprogramskoj (firmverskoj) realizaciji CU upravljački signali su memorisani u mikroprogramskoj memoriji (ROM-tipa)

➤ CU su sporiji od hardverskih

❖ Generisanje upravljačkih signala se obavlja pomoću mikroprograma koji se sastoji od mikroinstrukcija

❖ Za svaku mašinsku instrukciju postoji niz mikroinstrukcija pomoću kojih se generišu upravljački signali.

❖ U nekim sistemima korisnik može da kreira mikroprogram prema svojim potrebama

➤ Ovakvu organizaciju imaju **CISC** procesori (**C**omplete **I**nstruction **S**et **C**omputers)

