



Osnovi računarstva I

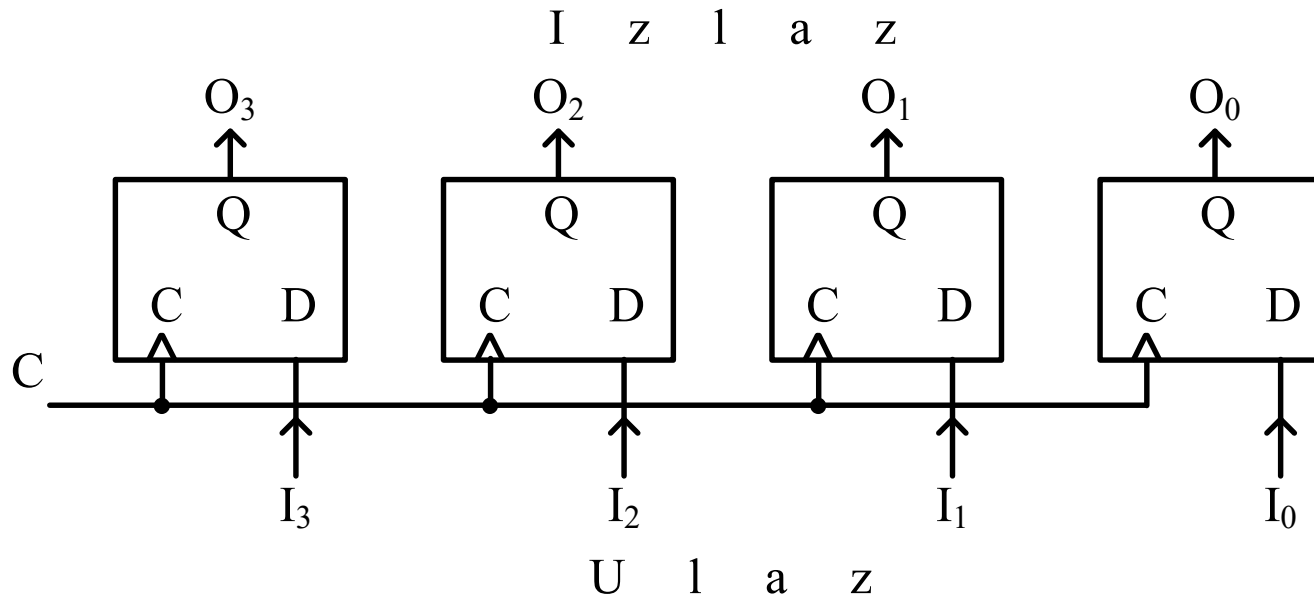
Osnovni digitalni sistemi (nastavak)



REGISTRI

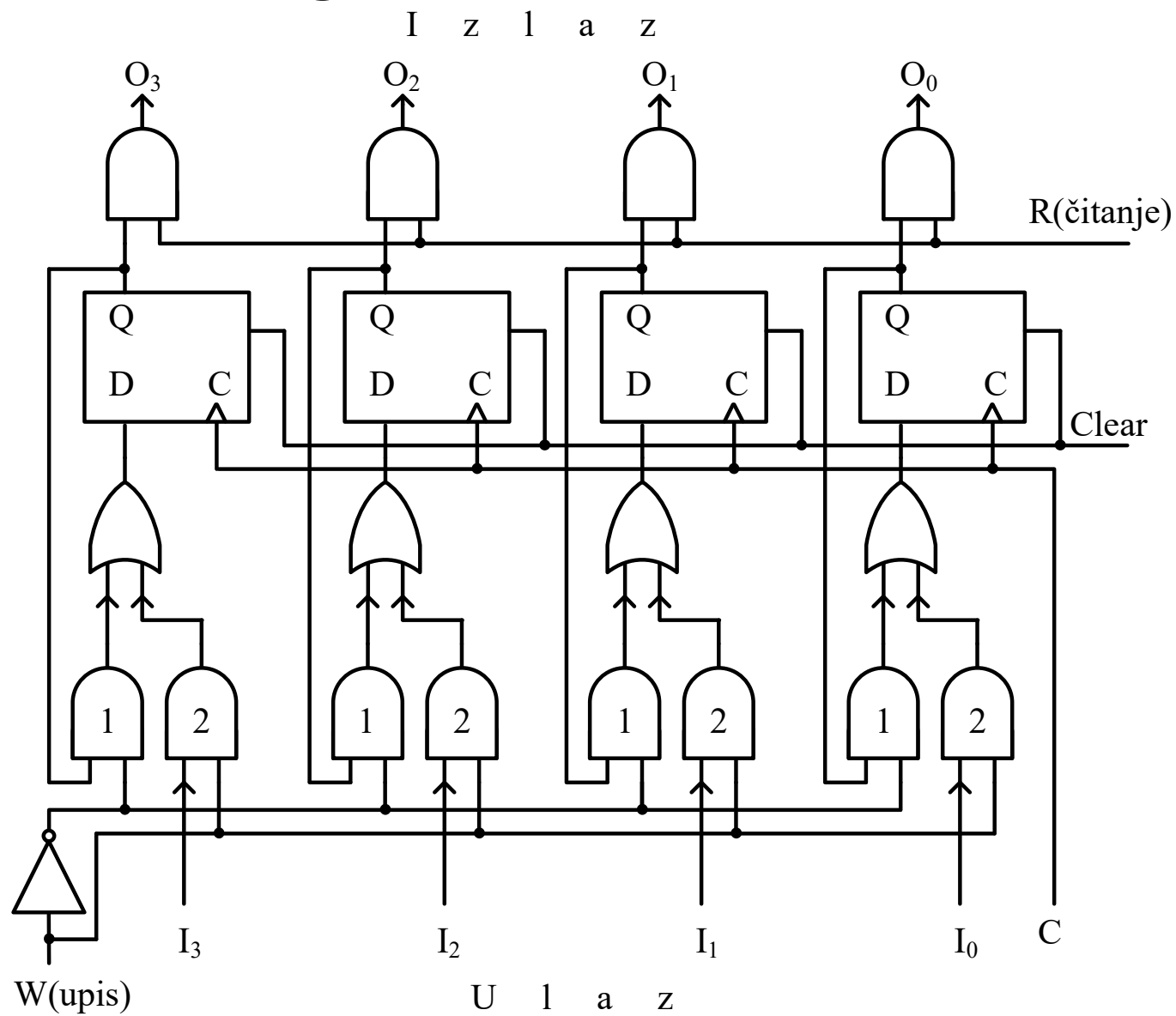
- Registri su veoma brze memorijske lokacije, malog kapaciteta, koje se upotrebljavaju za privremeno smještanje podataka, sastavljenih od niza binarnih cifara.
- Registar je digitalni sistem koji se sastoji od elementarnih memorijskih elemenata (flip-flopora), koji se upotrebljavaju za smještanje odgovarajućih bitova memorijske riječi (u svakom flip-flopu čuva se po jedan bit posmatrane riječi).
- U registru koji se sastoji od n flip-flopora može se smjestiti n -tobitni binarni podatak.
- Upis binarnog podatka u registar može se izvršiti na dva načina:
 - Istovremenim (paralelnim) dovođenjem svih bitova podatka na ulaze odgovarajućih flip-flopora – *paralelni registri*
 - Sekvencijalnim (rednim – bit-po-bit) dovođenjem bitova na ulaz registra - *pomjerački (shift) registri*

PARALELNI REGISTRI



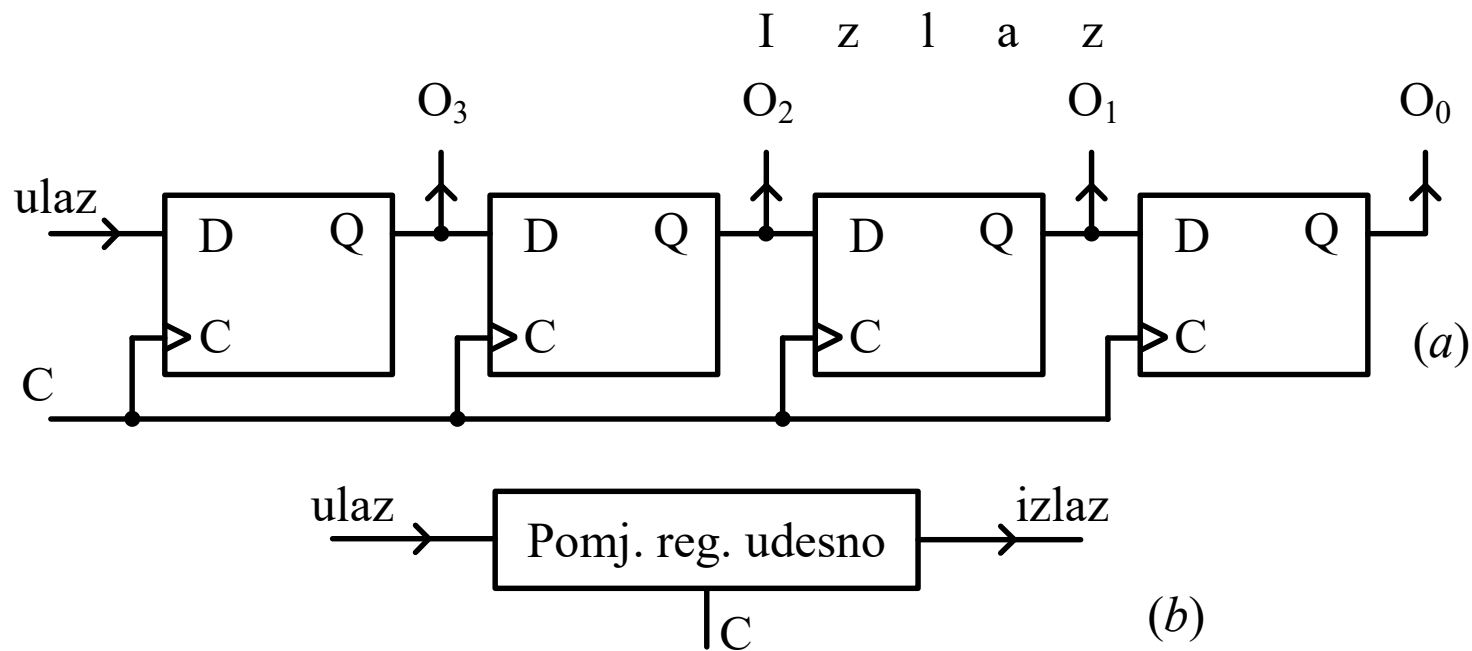
- D flip-flovi, koji čine registar, reaguju na jednu od dvije ivice takta (na slici su prikazani flip-flovi koji reaguju na uzlaznu ivicu)
- Upisivanjem novog podatka nepovratno se gubi sadržaj prethodno zapisan u registru, tako da se ovaj proces mora striktno kontrolisati
- Struktura registra prikazana na slici ne posjeduje mogućnost kontrole protoka podataka

Paralelni registri sa kontrolom upisa/čitanja



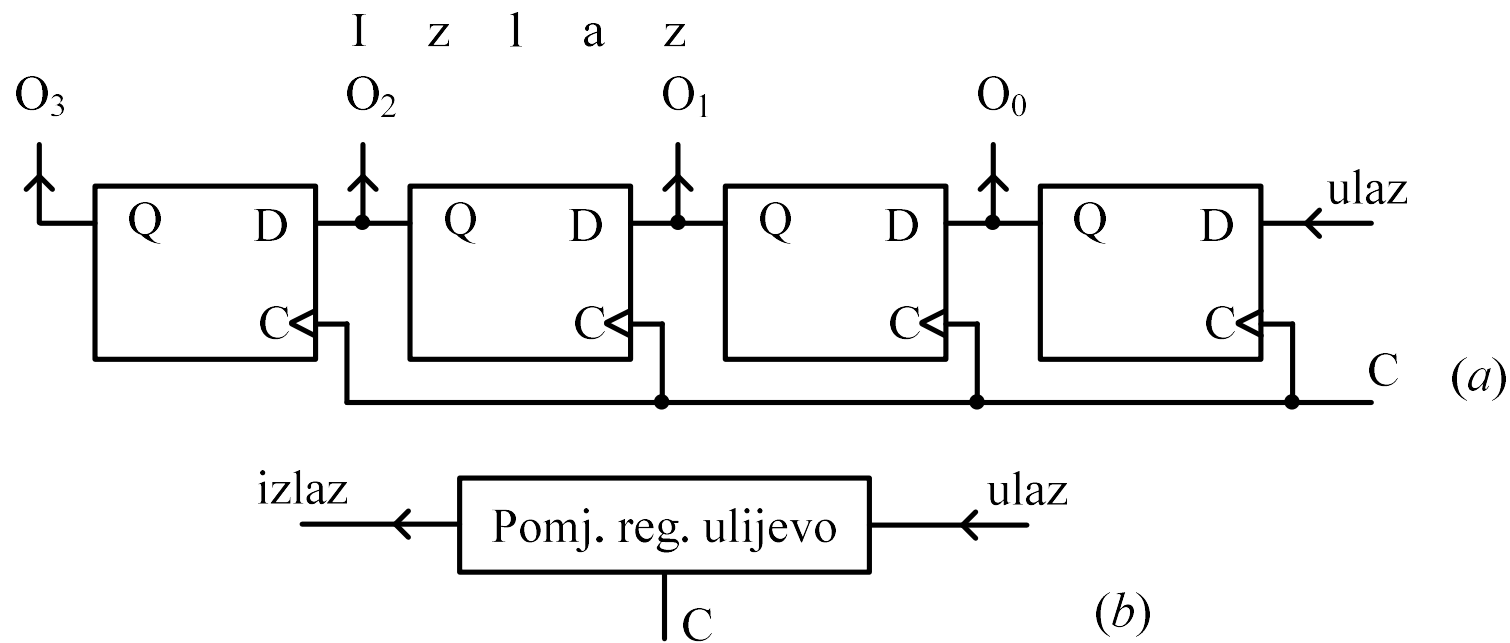
Pomjerački registri

- Pomjerački (shift) registar može se realizovati u dva oblika:
 - pomjerački registar udesno (eng. *shift right register*)
 - pomjerački registar ulijevo (eng. *shift left register*)



Pomjerački registri (nastavak)

- pomjerački registar ulijevo



- Potrebno je vrijeme od n taktih impulsa za upisivanje n -tobitnog podatka u pomjerački registar



Pomjerački registri (nastavak)

- Pomjeranjem **za jedno mjesto u lijevu stranu**, numerička vrijednost podatka koji se nalazi u registru se **udvostručuje** (množi se osnovom binarnog sistema $\rightarrow 2$)
- Pomjeranjem za **jedno mjesto u desnu stranu**, numerička vrijednost podatka koji se nalazi u registru se **dijeli sa 2**
- Na ovaj način bitno se pojednostavljuje implementacija računskih operacija množenja i dijeljenja sa 2
- Pomjeranjem **za n mjesta u lijevu stranu**, numerička vrijednost podatka koji se nalazi u registru se **množi sa 2^n**
- Pomjeranjem **za n mjesta u desnu stranu**, numerička vrijednost podatka koji se nalazi u registru se **dijeli sa 2^n**



Pomjerački registri (nastavak)

- *Primjer:* Ako je 0000 početno stanje 4-bitnog pomjeračkog registra udesno i ako se na njegov ulaz dovodi podatak 1011, odrediti sadržaj registra poslije prvog, drugog, trećeg i četvrtog taktnog impulsa

- *Rješenje:*

Početno stanje:	0000
Stanje nakon 1. taktnog impulsa:	1000
Stanje nakon 2. taktnog impulsa:	1100
Stanje nakon 3. taktnog impulsa:	0110
Stanje nakon 4. taktnog impulsa:	1011

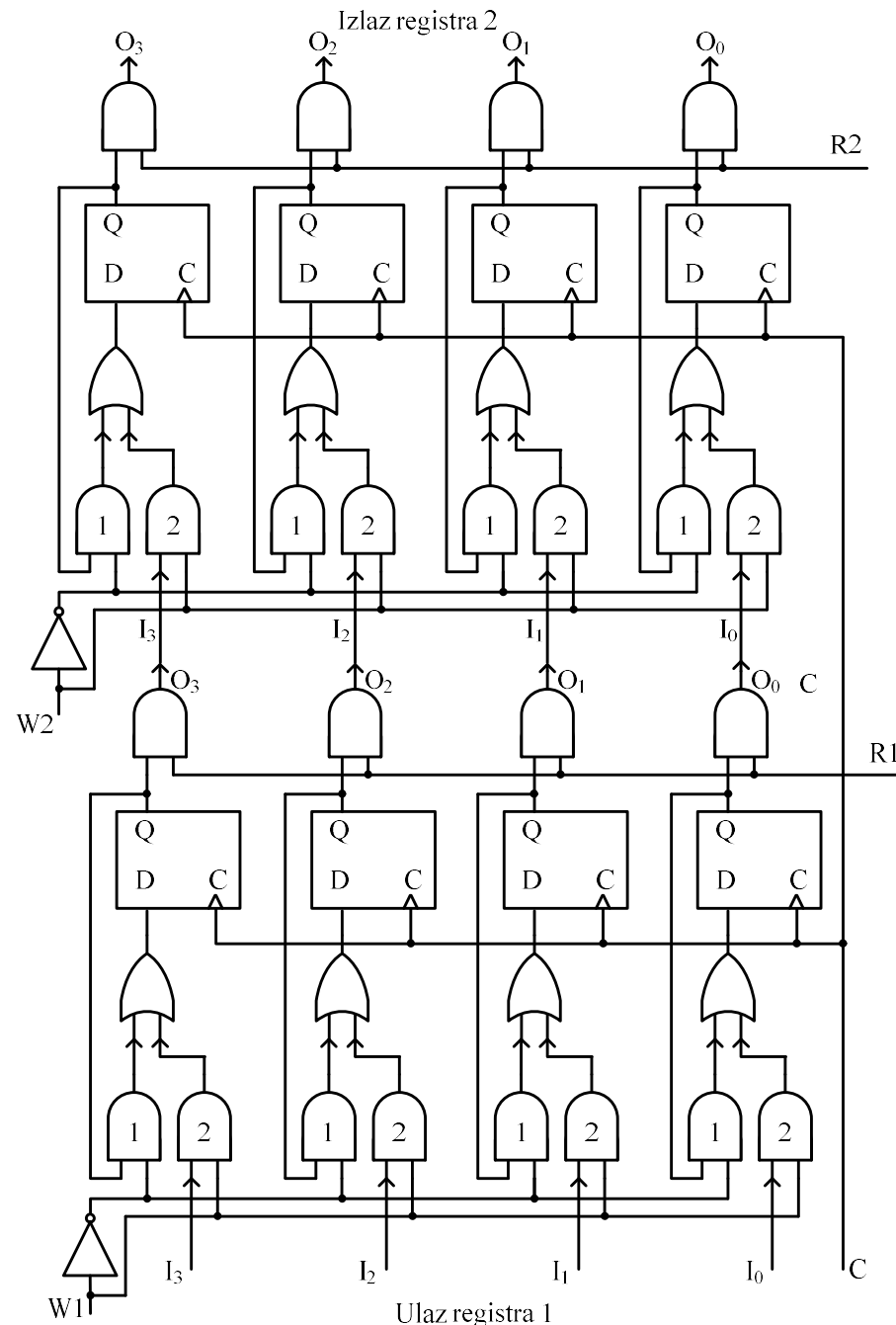


Prenos podataka iz registra u registar

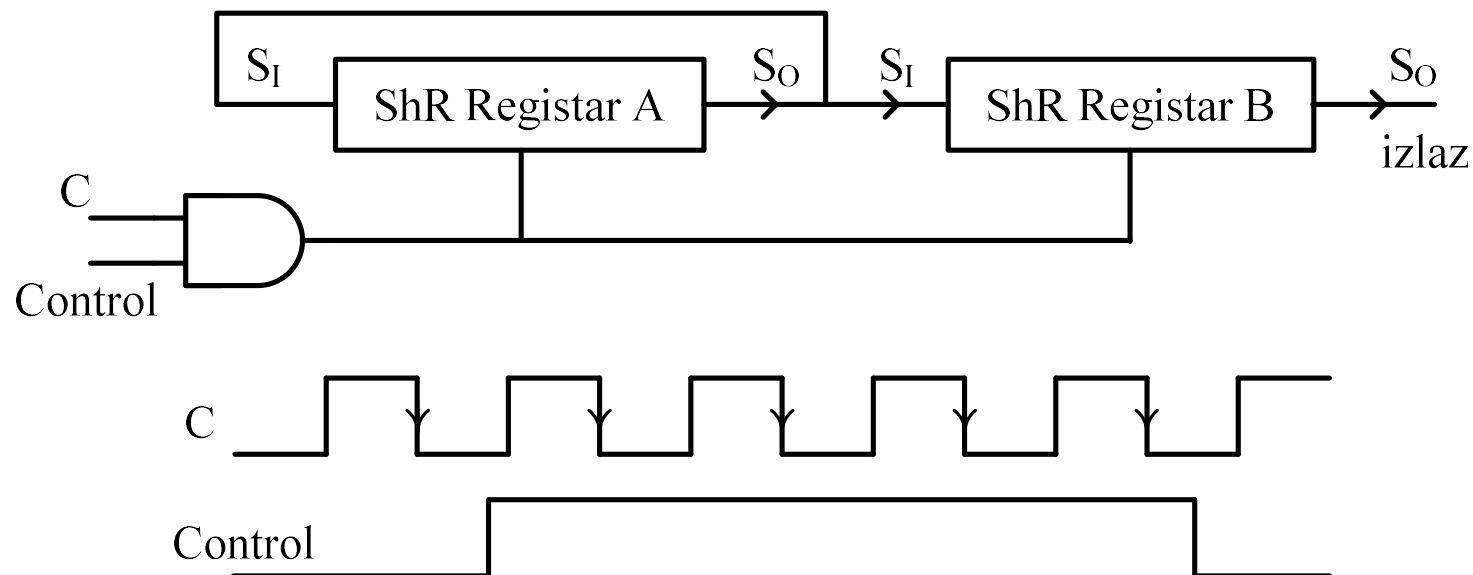
- Prenos podataka između registara može se ostvariti na dva načina: paralelno i serijski
- **Paralelni** prenos podatka iz registra u registar ostvaruje se direktnim povezivanjem odgovarajućih ulaza i izlaza dva registra (njihovih flip-flova)
- Paralelni prenos podataka može se izvršiti samo između paralelnih registara
- **Serijski** prenos podataka izvršava se između pomjeračkih registara
- Pošto se čitanje i upisivanje podataka u n -tobitnim pomjeračkim registrima izvršava u toku n taktnih intervala, serijski prenos podataka zahtijeva isti broj taktnih intervala
- Serijski prenos podataka između registara značajno je sporiji od paralelnog prenosa, koji zahtijeva samo jedan taktni interval

Paralelni prenos podataka iz registra u registar

- Prenosom podataka upravljaju kontrolni signal R1, (kontrola čitanja podataka iz registra iz kojeg se podatak prenosi), i kontrolni signal W2 (kontrola upisivanja podatka u registar u koji se podatak prenosi)
- U tom smislu, ova dva signala moraju imati iste logičke vrijednosti u trenutku kada se vrši prenos podataka iz registra 1 u registar 2



Serijski prenos podataka iz registra u registar

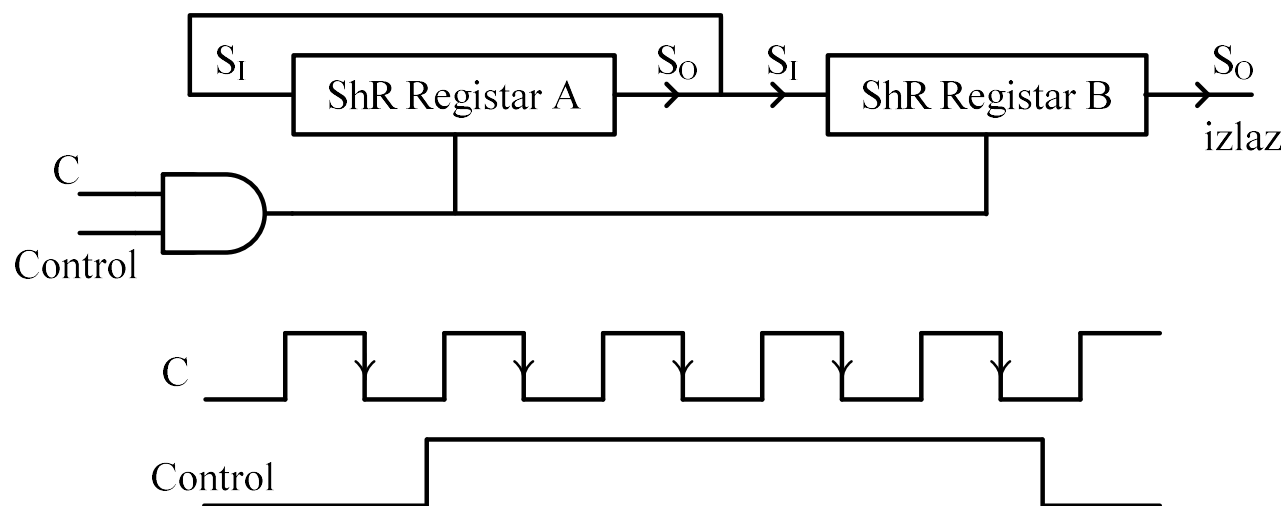


- Prikazanom konfiguracijom, u toku četiri taktna intervala, podatak koji se nalazi u registru A upisuje se u registar B
- Istovremeno, povratna veza sa izlaza registra A na njegov ulaz obezbeđuje da registar A zadrži prethodni sadržaj
- Serijski prenos podataka iz jednog registra u drugi kontroliše se signalom *Control* (uzima vrijednost logičke jedinice u toku prenosa podatka)

- Primjer: Pretpostavimo da je 1001 početno stanje pomjeračkog registra A, a da je 1110 početno stanje pomjeračkog registra B, sa donje slike. Uzimajući da su A i B pomjerački registri udesno, odrediti njihove sadržaje za vrijeme dok kontrolni signal Control ima vrijednost logičke jedinice.

- Rješenje:

	A	B
Početno stanje	1001	1110
Stanje nakon 1. taktnog impusa	1100	1111
Stanje nakon 2. taktnog impusa	0110	0111
Stanje nakon 3. taktnog impusa	0011	0011
Stanje nakon 4. taktnog impusa	1001	1001



Binarni polusabirač

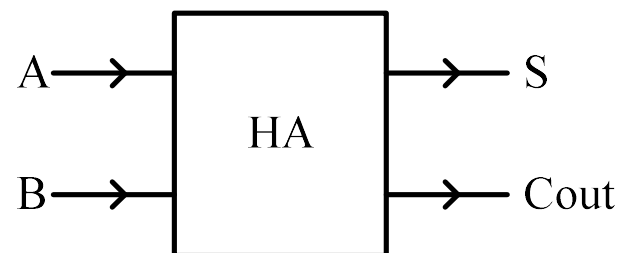
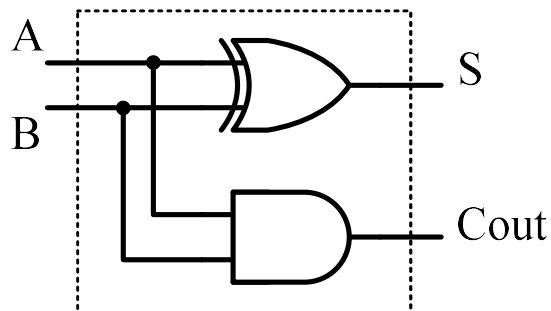
- Polusabirač (*Half-Adder – HA*):

A	B	S	C _{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = A \oplus B = \bar{A} \cdot B + A \cdot \bar{B},$$

- Prenos na više težinsko mjesto (*output carry*) ima vrijednost 1 samo u slučaju logičke jedinice kod oba binarna sabirka:

$$C_{out} = A \cdot B.$$



Binarni sabirač

- Sabirač koji pored binarnih cifara A i B uzima u obzir i prenos C_{in} sa nižeg težinskog mjesta, tzv. *ulazni prenos (input carry)*, naziva se **potpuni binarni sabirač** (*Full-Adder – FA*).

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

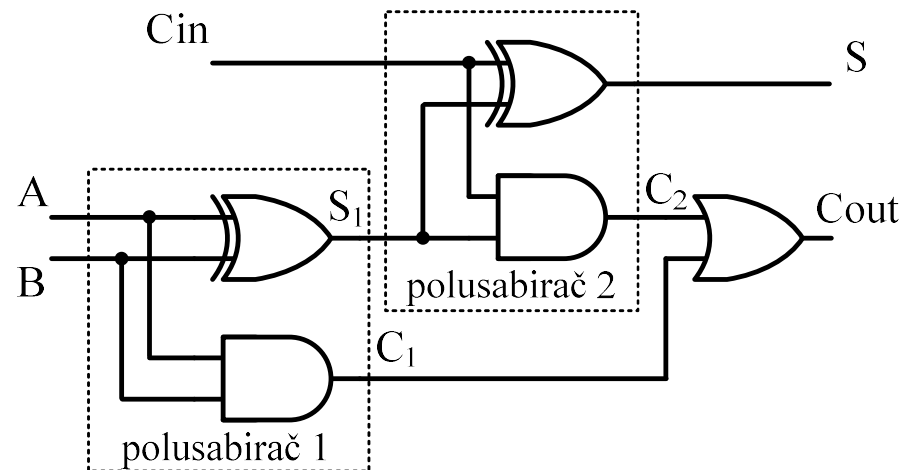
$$\begin{aligned}
 S &= \bar{A} \cdot \bar{B} \cdot C_{in} + \bar{A} \cdot B \cdot \bar{C}_{in} + A \cdot \bar{B} \cdot \bar{C}_{in} + A \cdot B \cdot C_{in} = \\
 &= \bar{A} \cdot (\bar{B} \cdot C_{in} + B \cdot \bar{C}_{in}) + A \cdot (\bar{B} \cdot \bar{C}_{in} + B \cdot C_{in}) = \\
 &= \bar{A} \cdot (B \oplus C_{in}) + A \cdot (\overline{B \oplus C_{in}}) = \\
 &= A \oplus B \oplus C_{in}
 \end{aligned}$$

$$\begin{aligned}
 C_{out} &= \bar{A} \cdot B \cdot C_{in} + A \cdot \bar{B} \cdot C_{in} + A \cdot B \cdot \bar{C}_{in} + A \cdot B \cdot C_{in} = \\
 &= (\bar{A} \cdot B + A \cdot \bar{B}) \cdot C_{in} + A \cdot B \cdot (\bar{C}_{in} + C_{in}) = \\
 &= (A \oplus B) \cdot C_{in} + A \cdot B.
 \end{aligned}$$

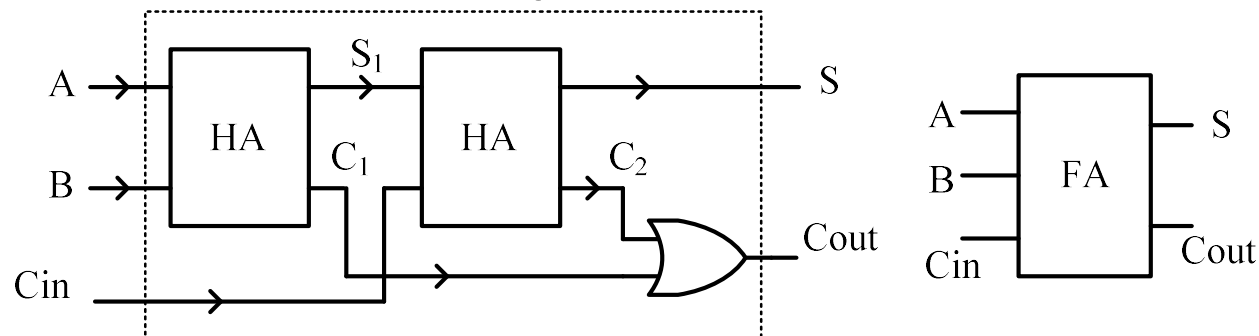
Binarni sabirač (nastavak)

$$S = A \oplus B \oplus C_{in} \quad C_{out} = (A \oplus B) \cdot C_{in} + A \cdot B.$$

- Realizacijom gornjih logičkih izraza implementira se digitalno kolo:

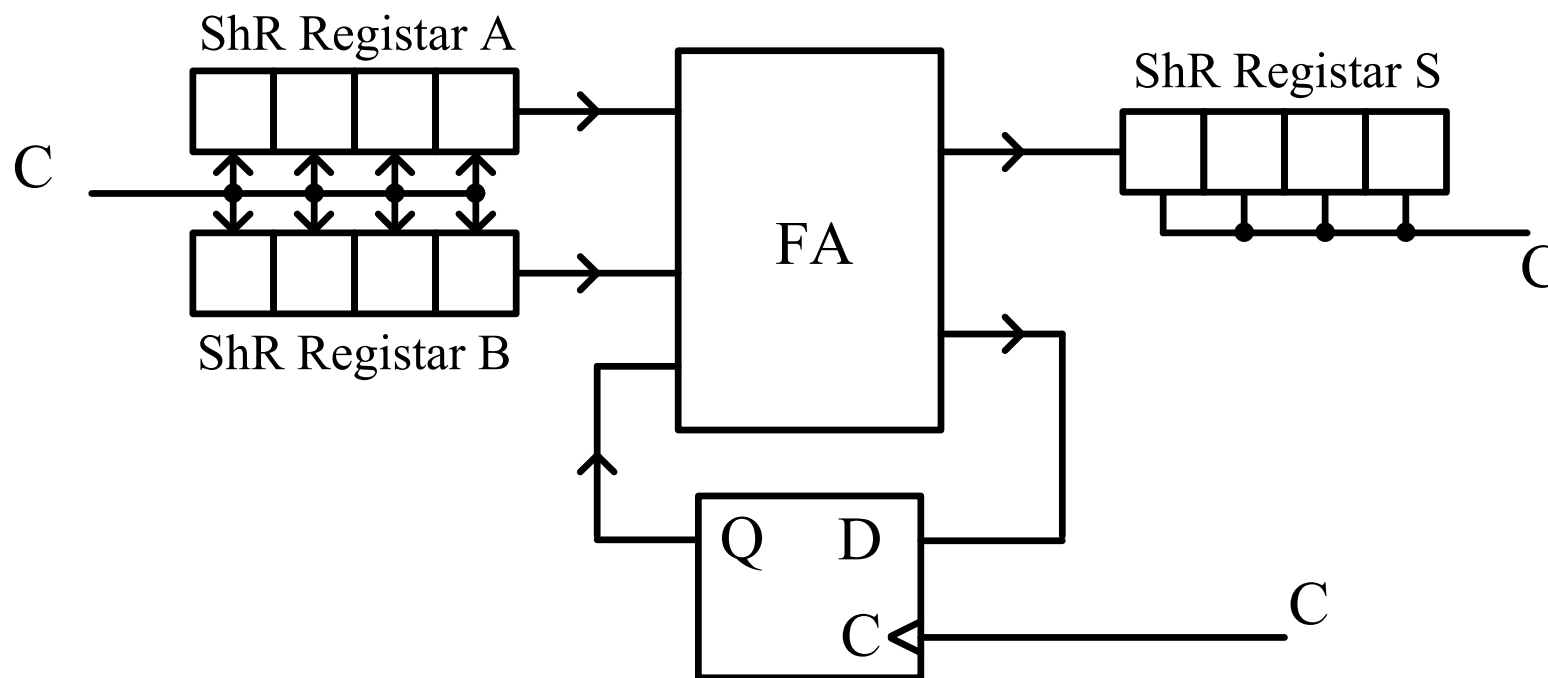


- Ako se ovo kolo uporedi sa implementacijom binarnog polusabirača dolazi se do zaključka da se potpuni binarni sabirač može realizovati korišćenjem dva binarna polusabirača, kao što je prikazano na slici:



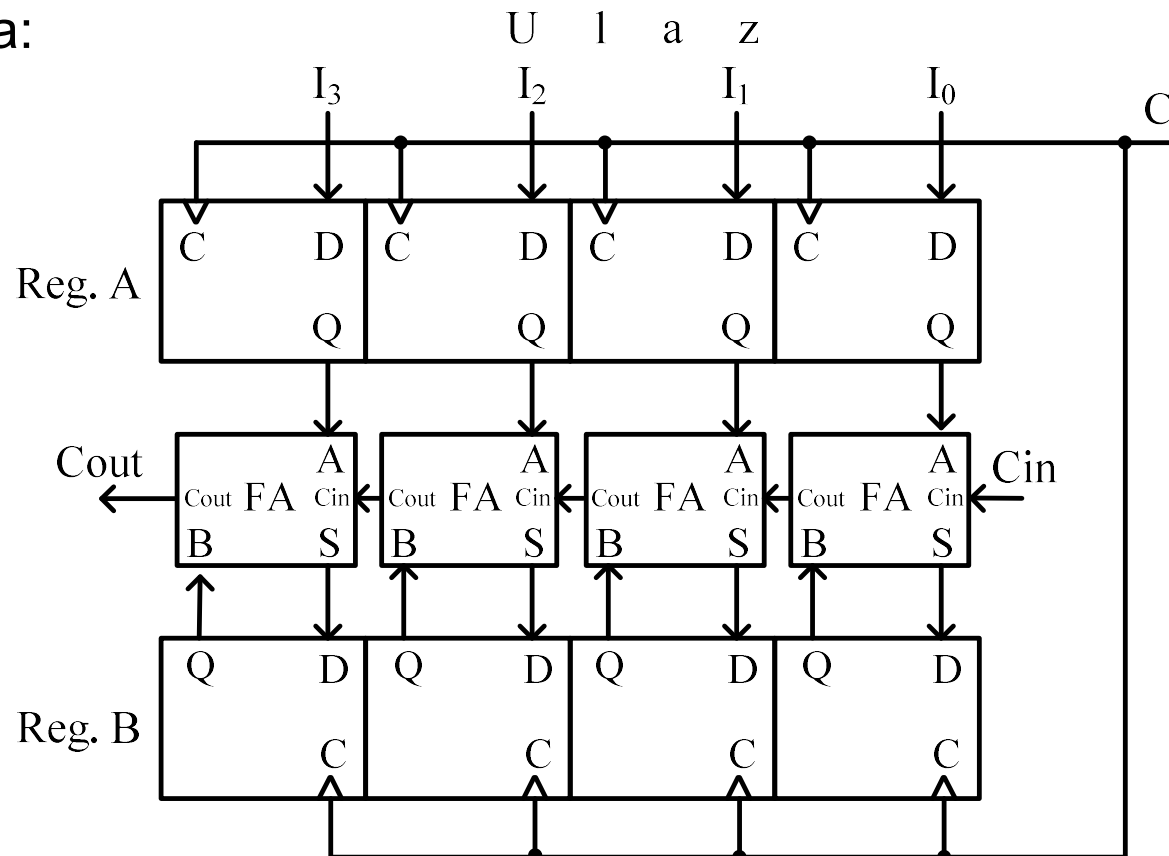
Redno sabiranje dva višecifrena binarna broja

- Prikazani polusabirač i potpuni binarni sabirač omogućavaju sabiranje jednocifrenih binarnih brojeva
- Redno sabiranje dva višecifrena binarna broja odgovara ručnom načinu izračunavanja: niz 1-cifrenih sabiranja odgovarajućih bitova sabiraka, počev od bita najmanjeg značaja (LSB) do najznačajnijeg bita (MSB).



Paralelno sabiranje dva višecifrena binarna broja

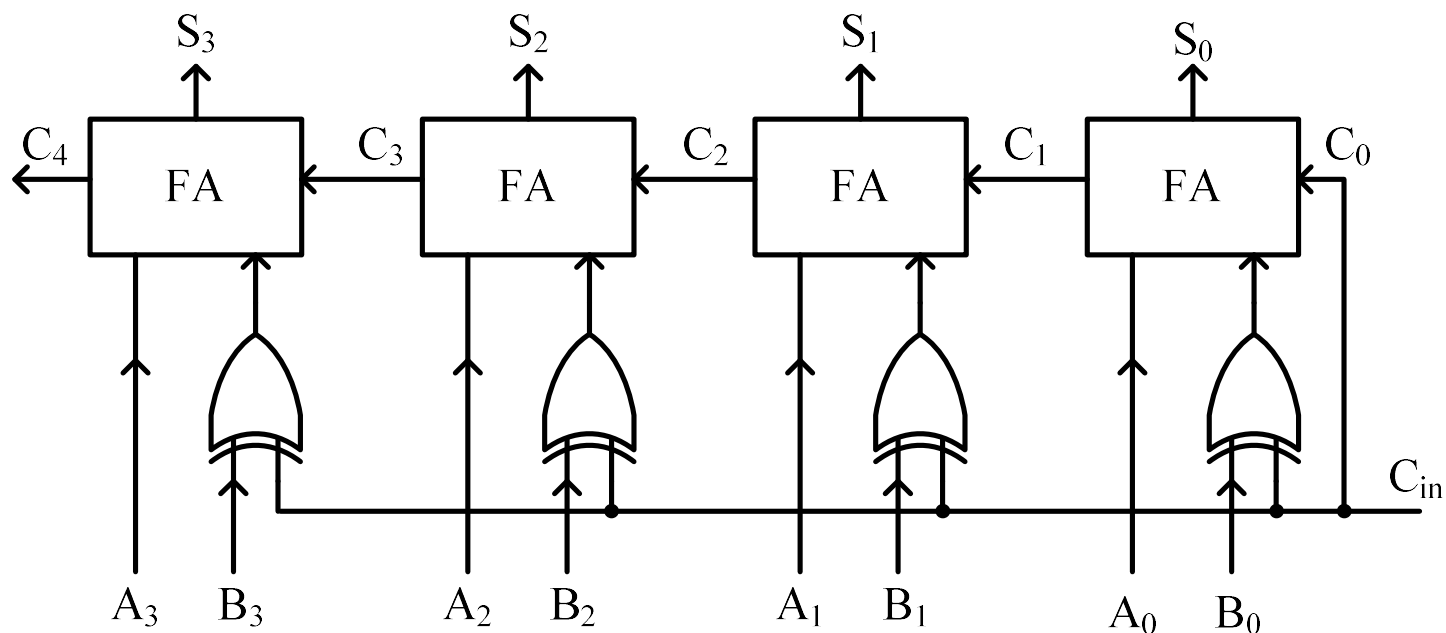
- Sva 1-cifrena sabiranja izvršavaju se istovremeno, pa je za paralelno sabiranje dva n -tocifrena binarna broja potrebno n potpunih binarnih sabirača:



- Registar B upotrebljava se i za smještanje drugog sabirka i za smještanje rezultata sabiranja

BINARNI SABIRAČ/ODUZIMAČ

- Operacija oduzimanja cijelih binarnih brojeva svodi se na logičko komplementiranje umanjioaca, njegovo sabiranje sa umanjenikom i dodavanje jedinice dobijenom rezultatu



- $C_{in}=0$ \rightarrow binarno sabiranje
- $C_{in}=1$ \rightarrow binarno oduzimanje

C_{in}	B	$C_{in} \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

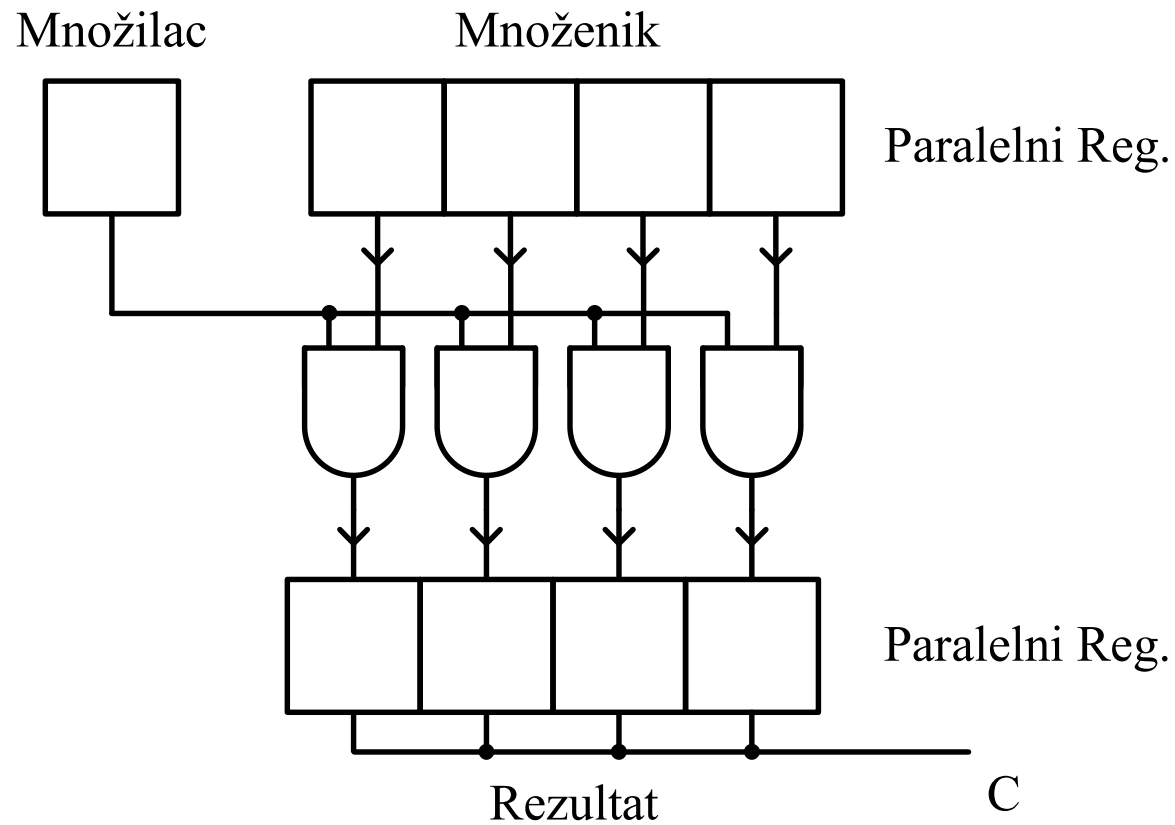


BINARNI MNOŽAČ

- Suštinski, množenje dva višecifrena binarna broja svodi se na množenje jednog od njih (nazovimo ga množenikom A) sa svim bitovima drugog broja (nazovimo ga množiocem B) ponaosob
- Najprije se višecifreni broj A pomnoži sa LSB broja B (bit B_0), potom sa sljedećim značajnijim bitom broja B (bit B_1), i tako redom ka MSB broja B
- Dobijeni rezultat svakog navedenog parcijalnog proizvoda pomjera se za po jedno mjesto u lijevu stranu u odnosu na prethodno izračunati parcijalni proizvod
- Potom se parcijalni proizvodi međusobno saberu
- Postupak se završava nakon uključivanja u sabiranje posljednjeg izračunatog parcijalnog proizvoda
- Proizvod dva jednobitna broja praktično se implementira logičkim I kolom

BINARNI MNOŽAČ (nastavak)

- Sistem za množenje 4-robitnog množenika 1-bitnim množiocem:



BINARNI MNOŽAČ (nastavak)

- Sistem za množenje dva 4-cifrena binarna broja:

