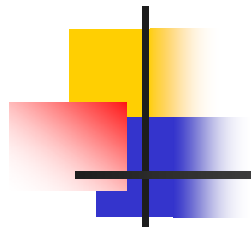




## 12. Standardni sekvencijski moduli

---

- sekvencijski moduli
- registri
- posmačni registri
- asinkrona brojila
- sinkrona brojila
- generatori sekvencije
- memorije



# Sekvencijski moduli

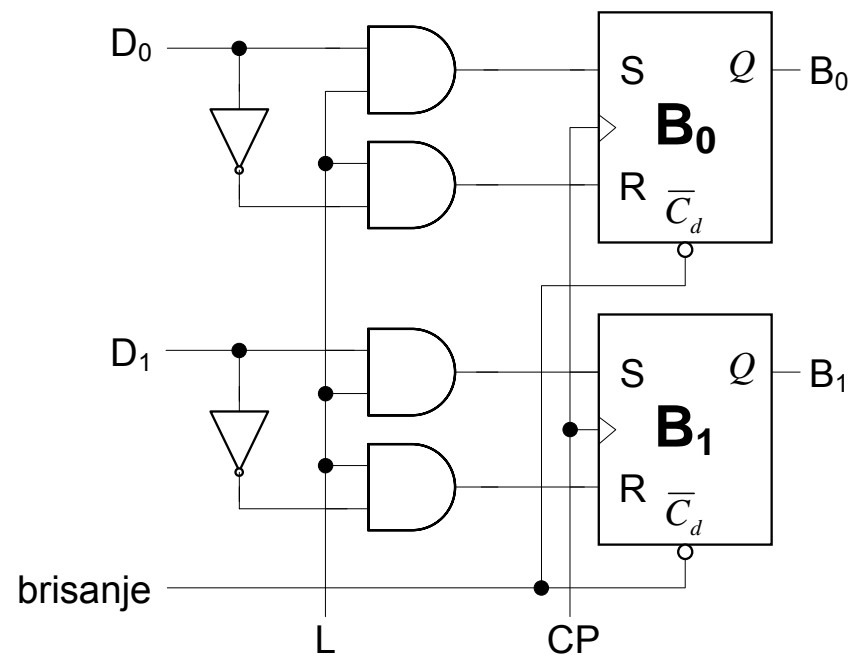
---

- *sekvencijski moduli*:
  - ~ cjeline koje sadrže kombinacijski sklop / memoriju (niz/skup bistabila ili registara)
- naročito zanimljivi *standardni* moduli:
  - n-bitni moduli
    - ~ n bistabila
  - pohranjivanje podataka
    - ~ *registri*
  - brojanje
    - ~ *brojila*

- *registri*
  - ~ pamćenje *višebitnih* podataka (= "registriranje"):
    - obično jedna riječ/znak
      - ~ standardna jedinica podataka za digitalni sustav
    - mogućnost upisa i ispisa/čitanja:
      - registri u užem smislu
        - ~ *paralelni* upis i ispis
      - posmaćni registri
        - ~ *serijski* upis i ispis
      - kombinacije upisa/ispisa
        - ~ druge primjene
  - izvedbe:
    - *svi* tipovi bistabila (osim T)
    - MSI i LSI moduli

# Registri u užem smislu

- osnovna struktura *registra u užem smislu*:  
~ *uređeni skup nepovezanih bistabila*
  - paralelni upis podatka
  - paralelno čitanje pohranjenog podatka
- način upisa:
  - sinkroni  
~ uobičajeni, bolji (→ upravljani!)
  - "asinkroni"  
~ registri (upravljanih) osnovnih bistabila:  
problem transparentnosti





# Registri u užem smislu

- prikaz (tipično)
  - ~ blok-simbol za cijeli registar:
    - (svi) bistabili
    - grupe bistabila
      - ~ format pohranjene riječi

$B_0$	$B_1$	$B_2$	...	$B_{n-1}$
-------	-------	-------	-----	-----------

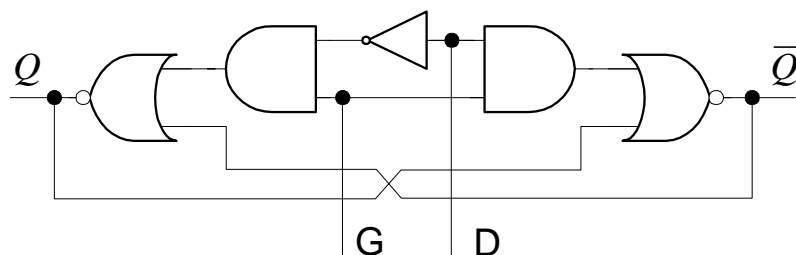
OP. KOD	ADRESA
---------	--------

- značajni elementi arhitekture i organizacije sustava:
  - protok podataka:
    - ~ registri i *staze* (engl. registers & data paths)  
*između* procesnih elemenata
  - viša razina razmatranja/opisivanja sustava
    - ~ RTL (engl. Register Transfer Level)

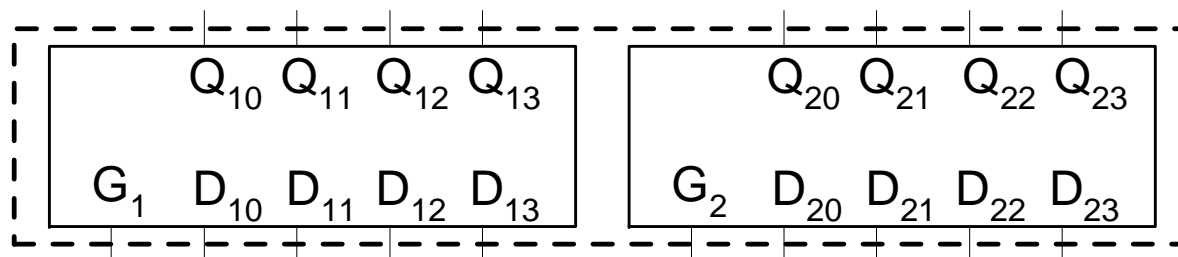
# Registri u užem smislu

*Primjer:* "8-bit bistable latch" 74100

- dvostruki 4-bitni registar (upravljanih osnovnih) D bistabila

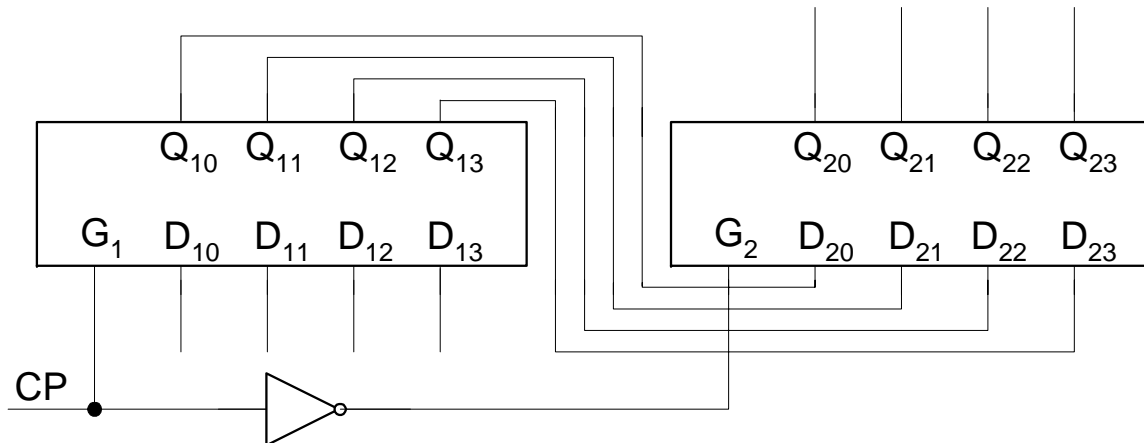


ULAZI		IZLAZI	
D	G	$Q$	$\bar{Q}$
L	H	L	H
H	H	H	L
X	L	$Q^{n-1}$	$\bar{Q}^{n-1}$



# Registri u užem smislu

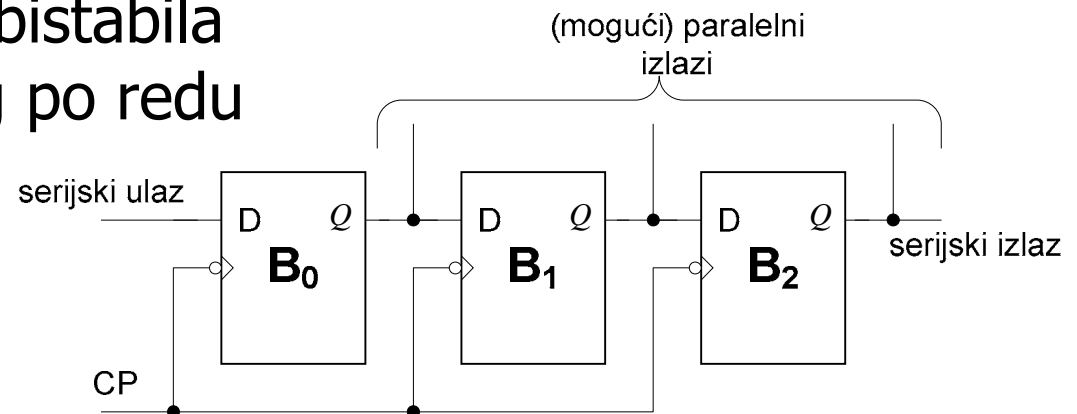
- primjena 74100:
  - privremeno pohranjivanje podataka na UI sustava ("međuspremnik", engl. buffer)
  - ostvarivanje složenijih struktura; npr. 4-bitni registar dvostrukih bistabila



ULAZI		IZLAZI	
D	G	$Q$	$\overline{Q}$
L	H	L	H
H	H	H	L
X	L	$Q^{n-1}$	$\overline{Q}^{n-1}$

# Posmačni registri

- *posmačni registar* (engl. shift register):
  - funkcijski pogled:
    - registar sa *serijskim* upisom i ispisom  
~ svojstveni mehanizam pomicanja (bitova) podatka od ulaza prema izlazu
    - analogija s tokarskim strojem  
~ "posmak" (engl. shift)
  - karakteristična struktura  
~ izlaz prethodnog bistabila na ulaz slijedećeg po redu





# Posmačni registri

- *posmak podataka:*

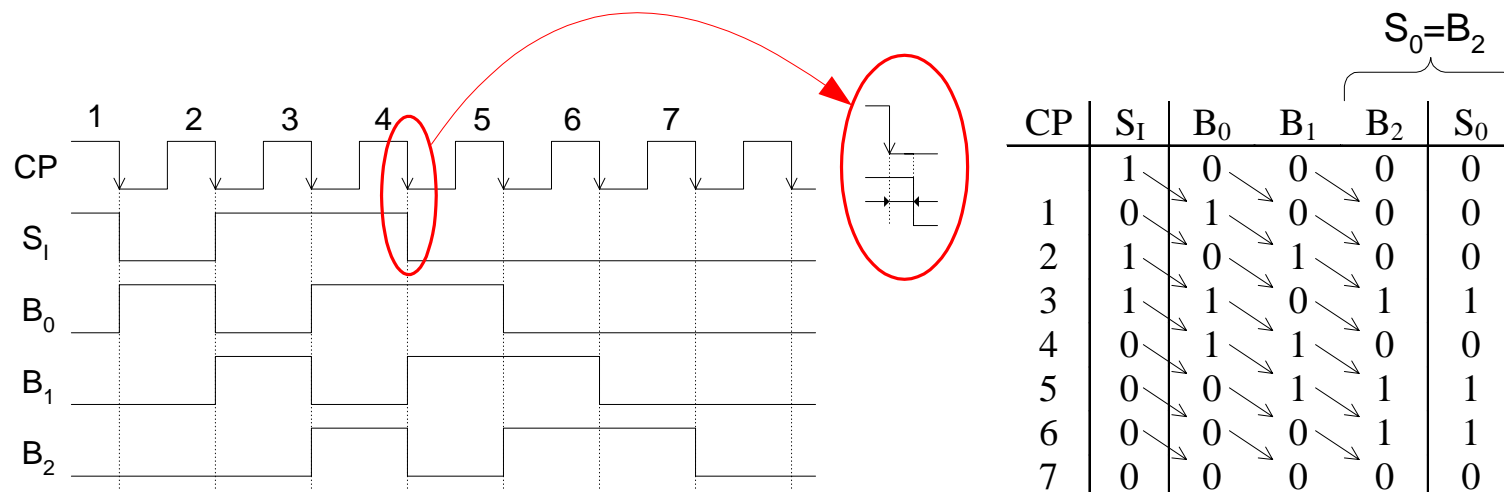
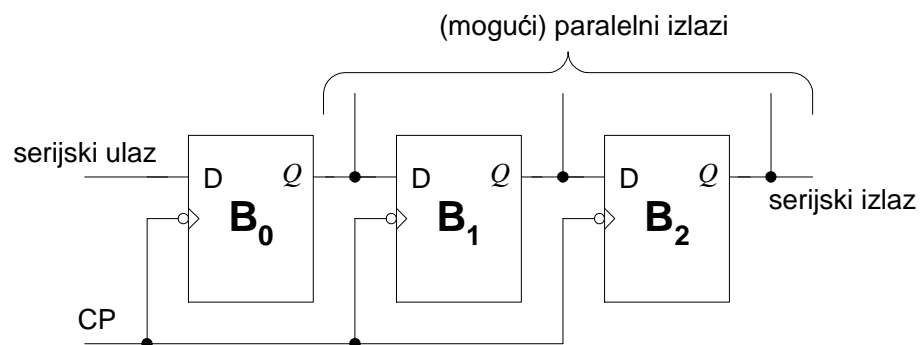
- *istovremeni upis:*

$$B_{i-1} \rightarrow B_i$$

$$B_i \rightarrow B_{i+1}$$

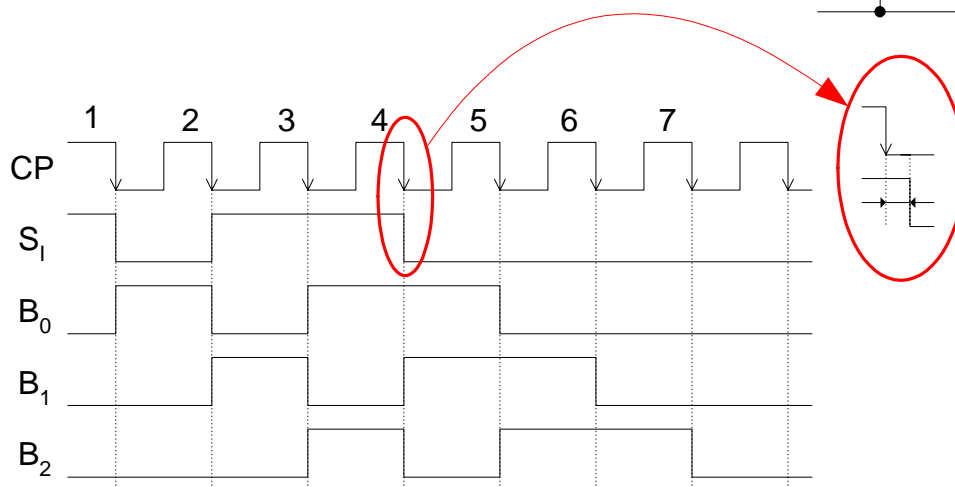
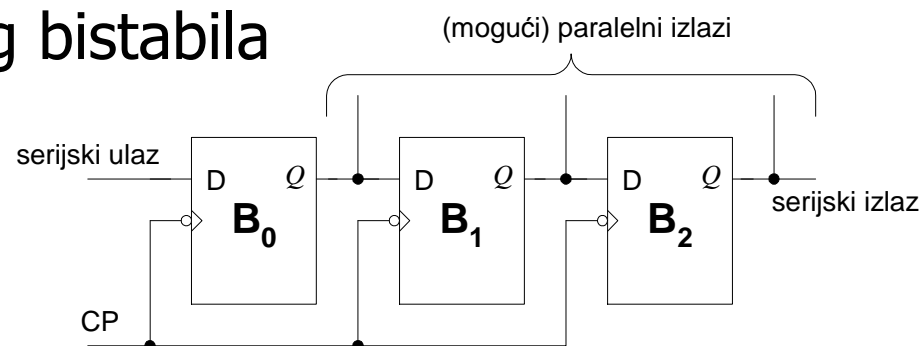
- *ispravnost upisa*

~ osigurati *kašnjenje* između bistabila



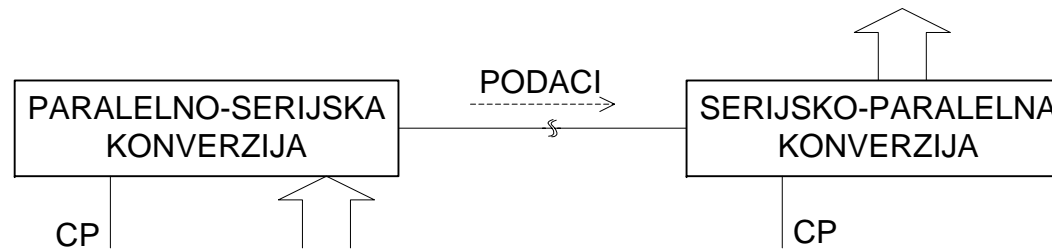
# Posmačni registri

- izvedbe *kašnjenja* između bistabila:
  - dvostruki bistabil
  - dva bistabila po bitu  
~ "simulacija" dvostrukog bistabila
  - bridom okidani bistabil



# Posmačni registri

- zapažanje:  
serijski upisani bitovi "putuju" kroz posmačni registar  
~ paralelni ispis  $n$ -bitnog serijskog podatka:  
*serijsko-paralelna pretvorba* (konverzija)
- kombinacije ~ tip pretvorbe:
  - serijski ulaz-paralelni izlaz ~ *serijsko-paralelna*
  - paralelni ulaz-serijski izlaz ~ *paralelno-serijska*

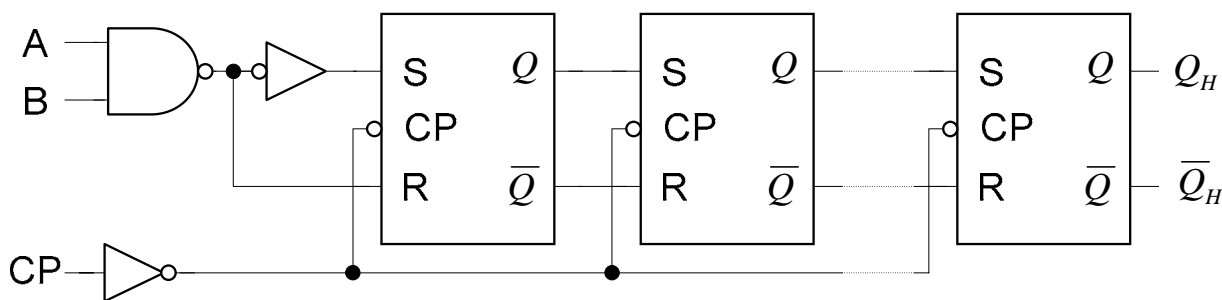
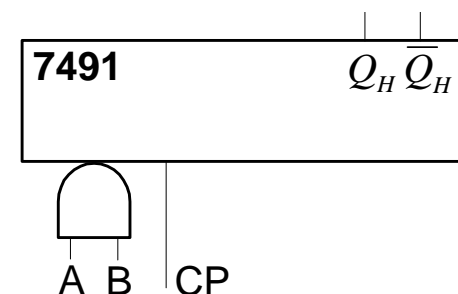


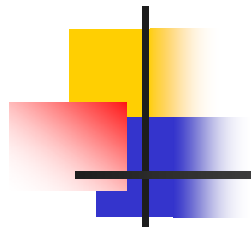
- serijski i paralelni ulaz i izlaz  
~ *univerzalni* posmačni registar

# Posmačni registri

*Primjer:* posmačni registar 7491

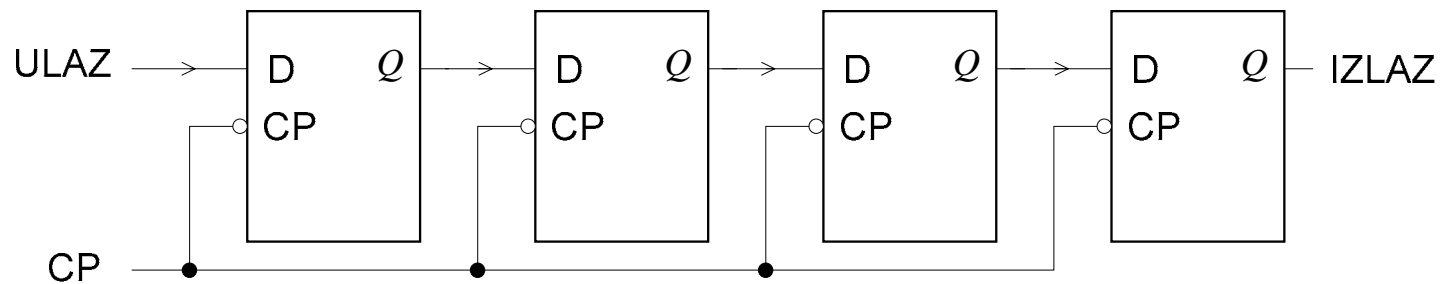
- 8-bitni MSI modul
- dvostruki SR bistabili
- serijski ulaz-serijski izlaz



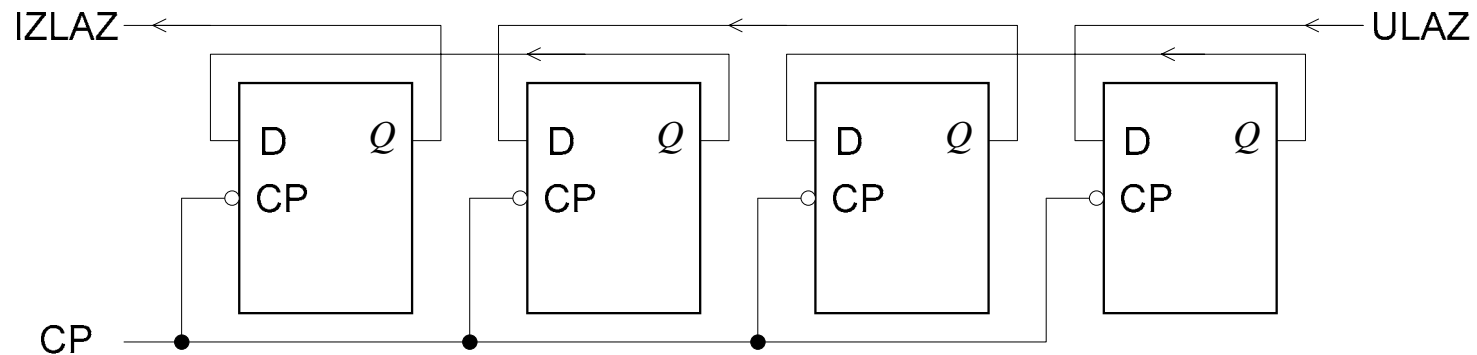


# Posmačni registri

- "smjer" posmaka:
  - uobičajeno "nadesno" (prema "normalnom" izlazu)

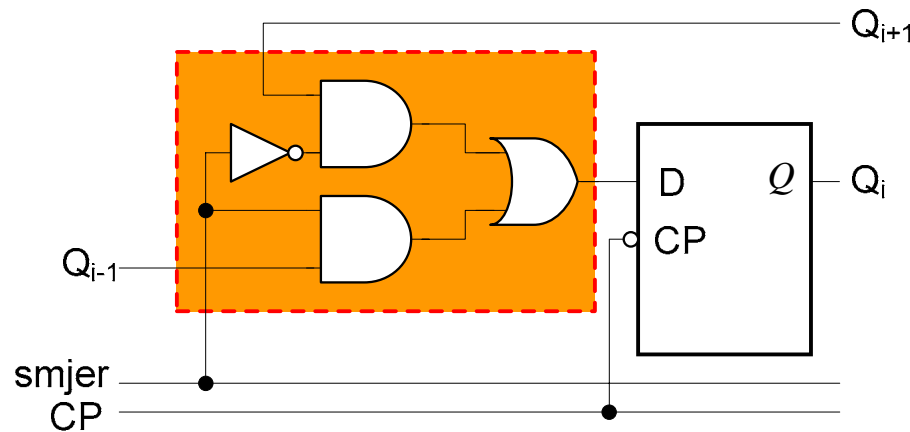


- moguće i "nalijevo", prema "normalnom" ulazu



# Posmačni registri

- kombiniranje smjera posmaka  
~ *dvosmjerni* (engl. bidirectional) posmačni registar

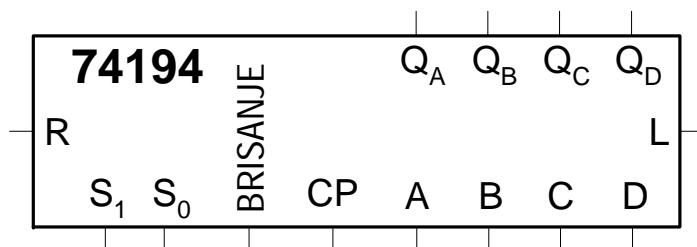


$$D_i = Q_{i-1} \cdot SMJER + Q_i \cdot \overline{SMJER}$$

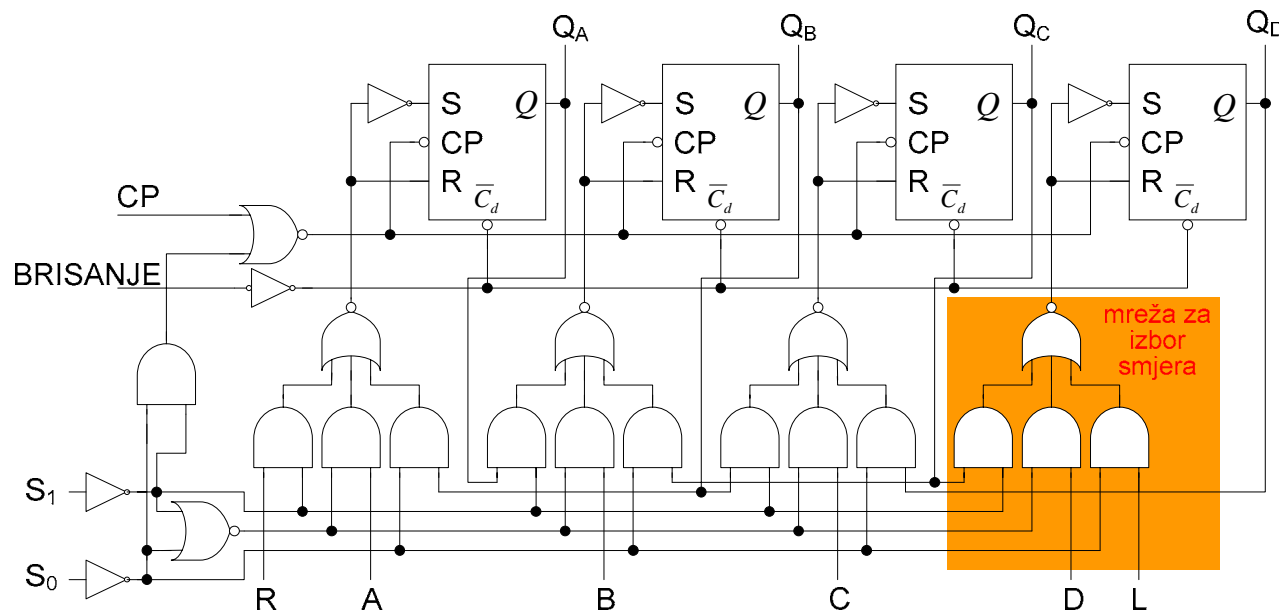
- značajne primjene:
  - efikasno obavljanje aritmetičkih operacija;  
npr. množenje/dijeljenje s  $2^n$  *posmakom* za n bitova
  - sklop za posmak (engl. shifter) na izlazu ALU

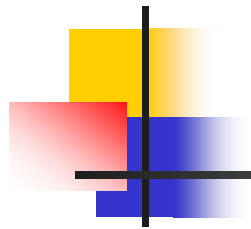
# Posmačni registri

*Primjer:* MSI dvosmjerni univerzalni posmačni registar s asinkronim brisanjem  
(4-bitni: 74194, 8-bitni: 74198)



S <sub>1</sub>	S <sub>0</sub>	FUNKCIJA
0	0	ZABRANA CP
0	1	→
1	0	←
1	1	PARALELNI UPIS





# Posmačni registri

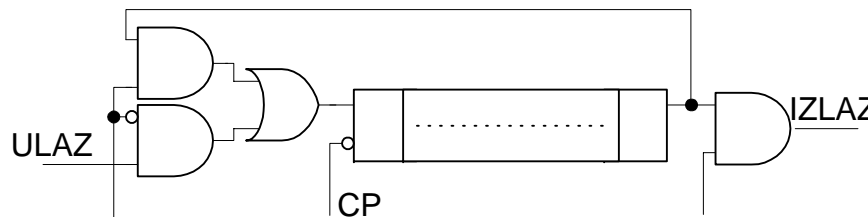
---

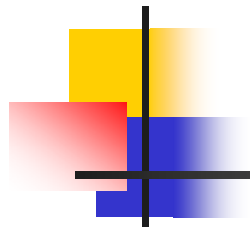
- primjene (1):
  - memoriranje podataka za serijsko izvršavanje (aritmetičkih) operacija
  - pretvorba oblika podataka:
    - serijsko-paralelna ( $\exists$  paralelni izlazi)
    - paralelno-serijska ( $\exists$  paralelni ulazi)
  - ostvarivanje (aritmetičkih) operacija:
    - množenje s 2: posmak nalijevo
    - dijeljenje s 2: posmak nadesno
  - brojanje ( $\rightarrow$  posmačni registar u funkciji brojila)



# Posmačni registri

- primjene (2):
  - sinkronizacija brzina prijenosa  
~ "glađenje" prometa ( $f_1 \neq f_2$ )
    - upis podataka s  $f_1$
    - upis podataka s  $f_2$
  - generiranje "pseudo-slučajnog" slijeda  
~ *generatori sekvencije*:  
linijski kodovi, kriptiranje
  - izvedbe cirkulirajućih memorija  
~ npr. generatori znakova

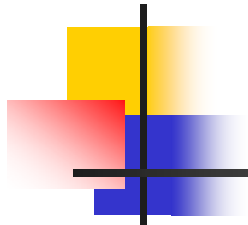




# Brojila

---

- *brojilo*:
  - ~ pod utjecajem ulaznih impulsa (obično CP) prolazi kroz *utvrđeni niz* stanja i *vraća u početno* stanje
  - sklop "broji" ulazne impulse
  - impulsi *ne moraju* biti periodički ( $f \neq \text{const.}$ )
  - "autonomni" sekvencijski sklop
    - ~ samo jedan ulaz, i to obično za CP
  - definicije:
    - *ciklus brojanja*
      - ~ niz stanja kroz koja brojilo prolazi
    - *baza brojanja*
      - ~ baza brojevnog sustava u kojem brojilo broji: broj stanja u ciklusu brojanja



# Brojila

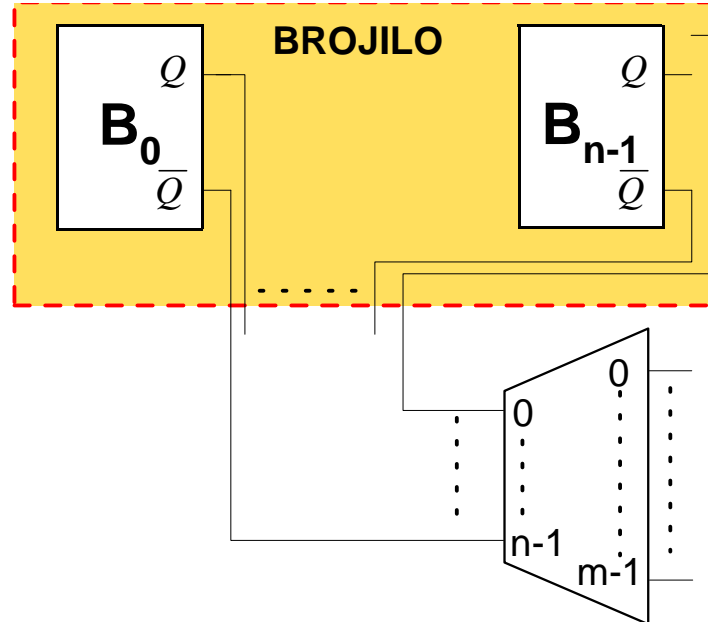
---

- baza brojanja  
~ brojanje u "modulu":
  - stanje brojila = ostatak cjelobrojnog dijeljenja bazom (modulom)
  - *brojilo modulo  $m$  ( $m = B$ )*
    - $l$  impulsa  $\rightarrow l = k \cdot m + j$ ,  $j$ : sadržaj brojila ~ stanje
    - $n$  bistabila  $\rightarrow N = 2^n$  : max broj stanja
    - $W = 2^n - 1$  : max broj (binarni kod!)
    - $2^{n-1} = N/2 < m \leq 2^n$

- osnovna *funkcijska* podjela:
  - brojila u užem smislu (engl. counters)
    - ~ važan je *redoslijed* izmjene stanja u ciklusu  
i *mogućnost ispravnog* očitavanja (→ dekodiranja!)  
svakog stanja
  - djelitelji frekvencije (engl. scalers)
    - ~ važan samo broj stanja,  
*ne* i redoslijed njihove izmjene

# Brojila

- *brojila u užem smislu:*
  - prikladno projektiranje brojila  
~ jednostavniji dekodler
  - važna primjena  
~ generator upravljačkih impulsa digitalnog sustava

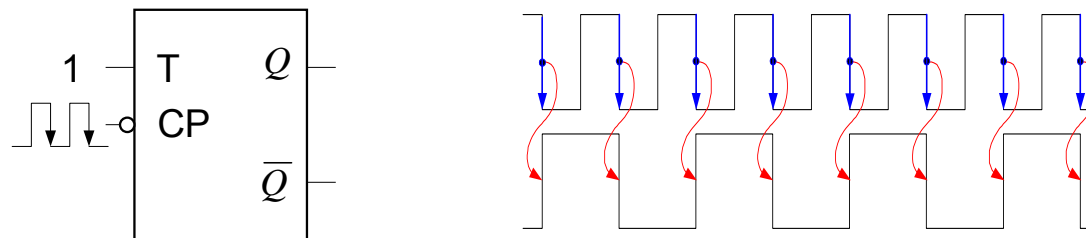


- *djelitelji frekvencije:*
  - sklop samo broji ulazne impulse
  - očitati *samo ono* stanje koje definira željeni izlazni impuls  
~ nakon svakih  $n$  impulsa, od nekog početnog
  - pojednostavljivanje dekodera  
~ *nepotpuno* dekodiranje
  - ubrzanje rada:  $f_{\max}$  ↗  
npr. naročito za asinkrona brojila

- *vremenski odnosi* prilikom promjene stanja:
  - *sinkrona* brojila:
    - (svi) bistabili mijenjaju stanja *sinkrono* s nailaskom ulaznih impulsa (takta)
    - složenija, skuplja, brža
  - *asinkrona* (engl. ripple) brojila:
    - promjena stanja *prvog* bistabila uzrokuje *serijsku* promjenu stanja *sljedećih u nizu*
    - prostiranje promjene stanja  
~ izlaz prethodnog pobuđuje sljedeći bistabil  
(engl. ripple: mrežkanje, talasanje)
    - jednostavnija, jeftinija, sporija

# Brojila

- bistabil u brojilima:
  - ~ konceptualno T, ali izveden od JK ili RS
- $T = 1 \rightarrow$  promjena stanja
  - ~ dijeli frekvenciju ulaznih impulsa s 2

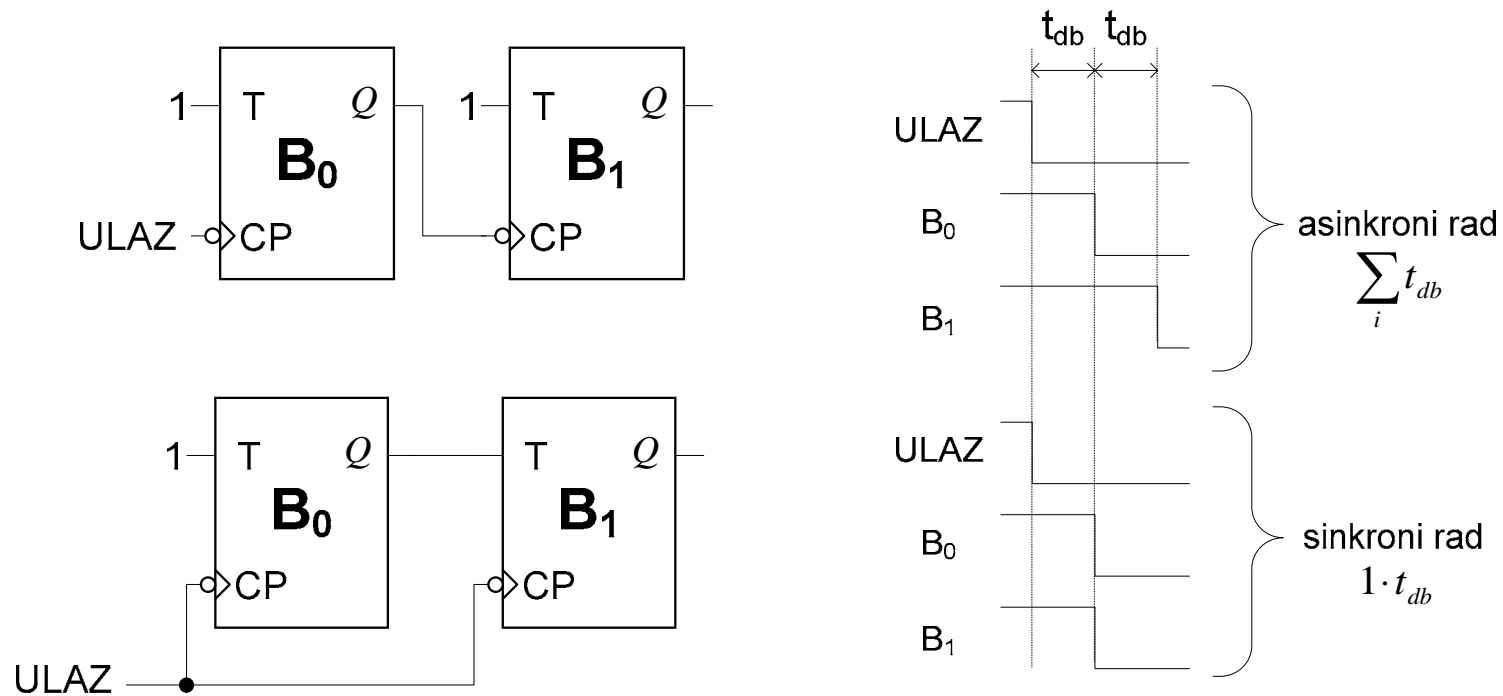


- direktna implementacija *asinkronih* brojila
  - ~ niz bistabila od kojih svaki prethodni pobuđuje naredni u nizu
- brojanje u *binarnom* brojevnom sustavu
  - ~  $2^n$  stanja za  $n$  bistabila:
    - binarno brojilo (bistabili  $\sim 2^i$  : težine potencije od 2)



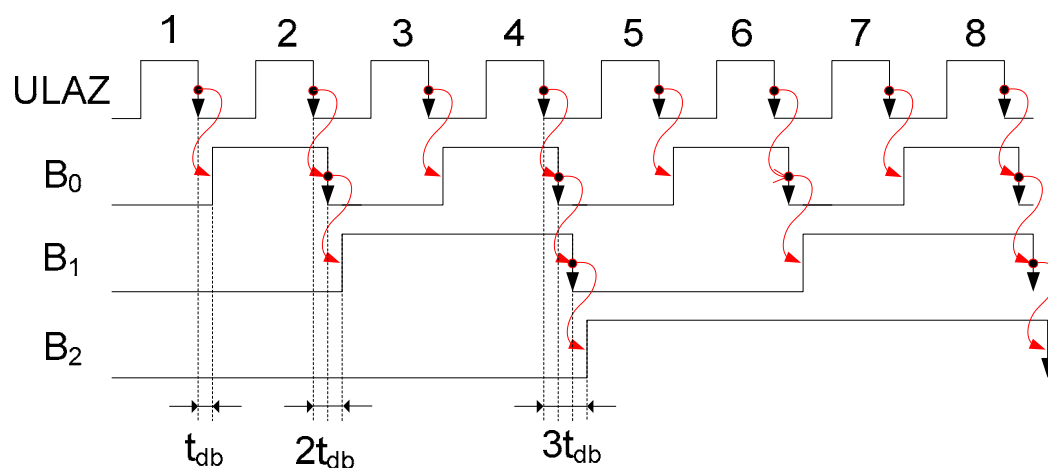
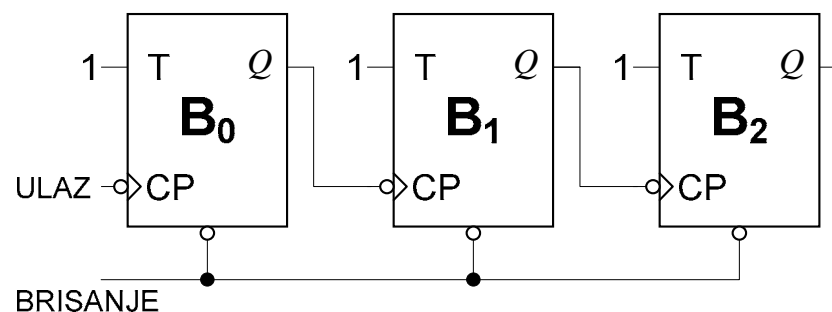
# Asinkrona brojila

- *asinkrona brojila*  
~ bistabili *ne* mijenjaju stanje u *sinkronizmu*  
sa zajedničkom pobudom:  
sporiji rad!



# Asinkrona brojila

- *binarno* brojilo:
  - brojilo broji u *binarnom* brojevnom sustavu
  - $2^n$  stanja za  $n$  bistabila;  
npr.  $n = 3 \rightarrow m = 2^n = 8$



BROJ ULAZNIH IMPULSA	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

# Asinkrona brojila

- *očitanje* (dekodiranje) stanja  
~ tipični problem:

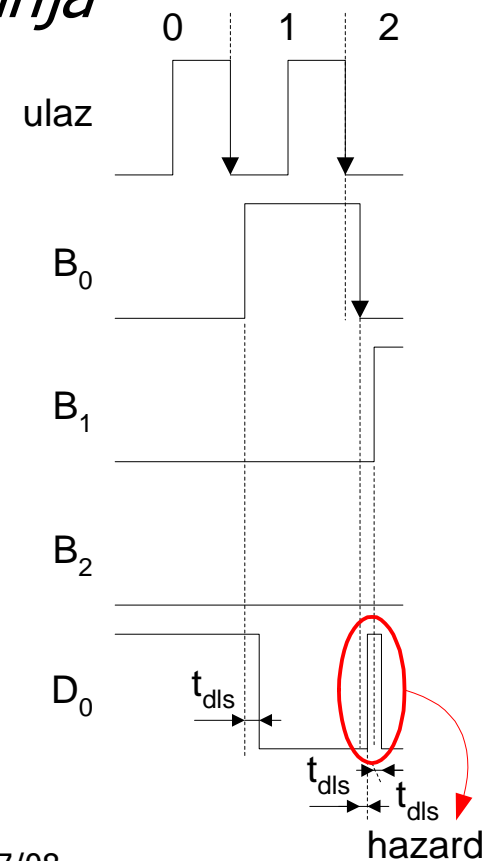
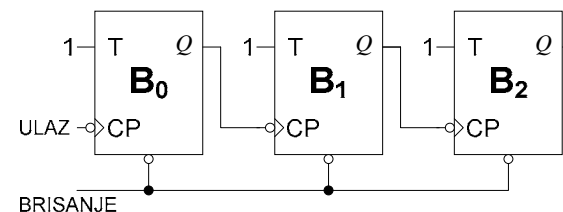
- *serijsko* okidanje bistabila:  
~ *tranzijentna pogreška dekodiranja*  
(→ hazard)
- dekodiranje svih  $2^n$  stanja  
~ *potpuno dekodiranje*;  
npr. dekodiranje  $D_0$

$$D_0 = \overline{B_2} \overline{B_1} \overline{B_0}$$

$$D_1 = \overline{B_2} \overline{B_1} B_0$$

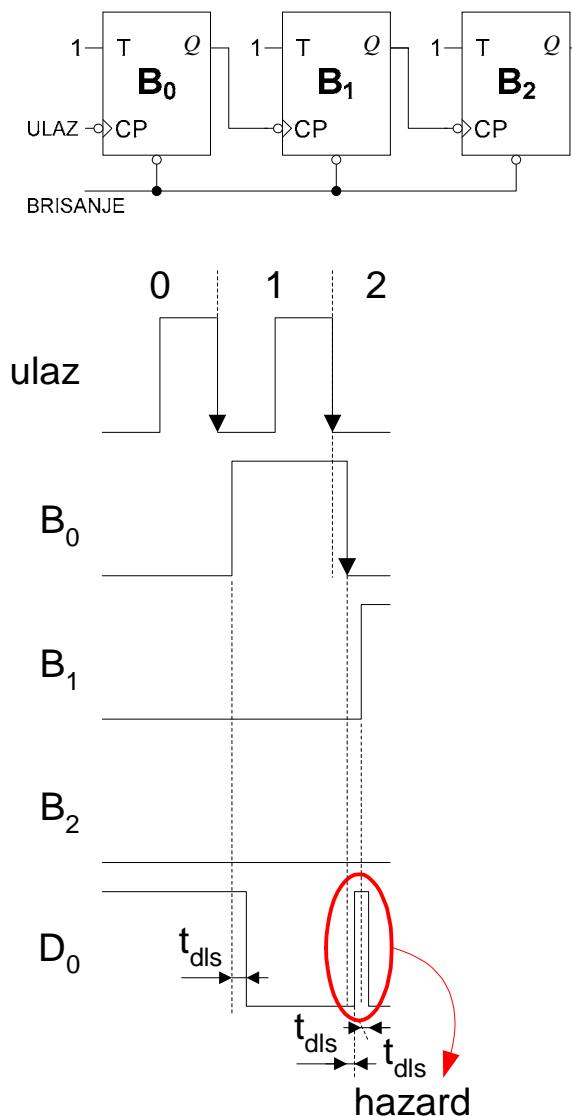
⋮

$$D_7 = B_2 B_1 B_0$$



# Asinkrona brojila

- *tranzijentna pogreška dekodiranja*:  
→ pojava hazarda
  - moguće rješenje
    - ~ *zakasniti* očitavanje tako da prijelazna pojava ne djeluje
  - praktična implementacija
    - ~ *kombinirati* očitavanje s ulaznim impulsima





# Asinkrona brojila

---

- vremenski odnosi:
  - *vrijeme kašnjenja* (cijelog) brojila  
~ najduže vrijeme odziva:  
promjena stanja *svih*  $n$  bistabila  $T_d = n \cdot t_{db}$
  - *vrijeme razlučivanja* (*rezolucije*) ulaznih impulsa  
~ svojstvo prvog bistabila  $T_{\min} = t_{db}$
  - *maksimalna frekvencija*  
~ *različita* za brojila u užem smislu i za djelitelja



# Asinkrona brojila

---

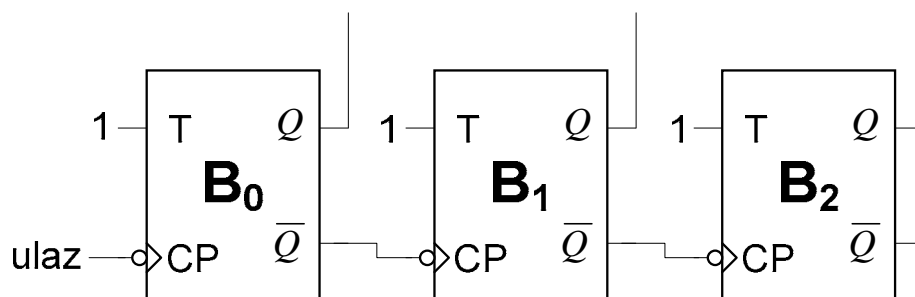
- maksimalna frekvencija *brojila u užem smislu*:
  - očitanje (= dekodiranje) svih stanja!
  - najlošiji slučaj
    - ~  $B_0$  *ne smije* promijeniti stanje sve dok  $B_{n-1}$  ne dođe u stanje uzrokovano *prethodnim* impulsom

$$f_{\max} = \frac{1}{n \cdot t_{db} + t_{oč}}$$

- maksimalna frekvencija *djelitelja*:
  - odabrati "prikladno" stanje koje će se očitati
    - ~ min broj bistabila mijenja stanje
  - $f_{\max}$  slijedi iz analize *prijelaza u to stanje*

# Asinkrona brojila

- *reverzno* (binarno) brojilo  
 ~ brojilo unatrag:
  - "smanjivanje" sadržaja brojila  
 ~ odbijanje impulsa
  - pobuda s  $\bar{Q}_{i-1}$  prethodnog bistabila  
 ~  $Q_i: 0 \rightarrow 1$

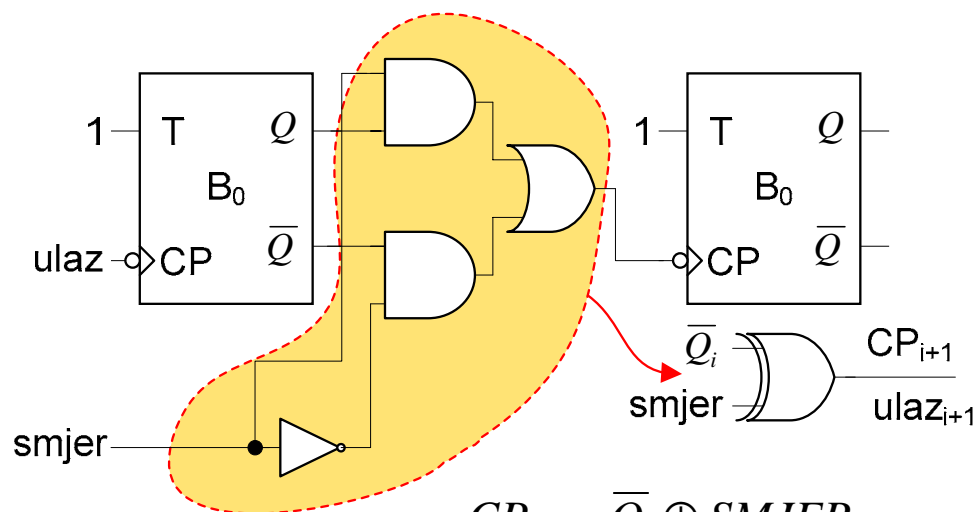


ULAZ	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

# Asinkrona brojila

- *brojilo naprijed-natrag* (engl. up-down counter)  
~ kombiniranje brojanja naprijed i natrag:  
veća fleksibilnost

- konceptualna implementacija:



$$CP_{i+1} = \bar{Q}_i \oplus SMJER$$

$$= \bar{Q}_i \cdot \overline{SMJER} + Q_i \cdot SMJER$$

- primjena:

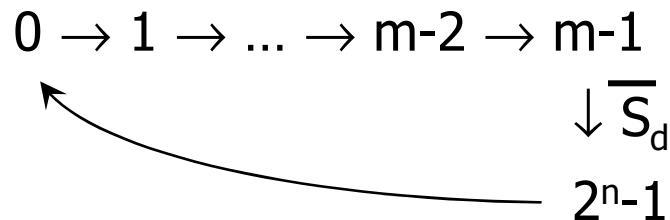
- digitalno upravljanje
- obavljanje jednostavnih aritmetičkih operacija *nad impulsima*

$$SMJER = \begin{cases} 0: \text{brojanje natrag} \\ 1: \text{brojanje naprijed} \end{cases}$$

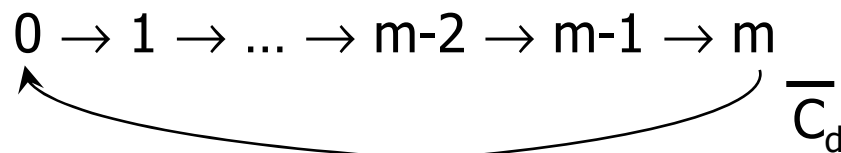


# Asinkrona brojila

- *brojilo modulo m,  $m \neq 2^n$*   
~ prekid ciklusa binarnog brojanja  
korištenjem *asinkronih* ulaza bistabila
- prekid aktiviran *zadnjim stanjem* u ciklusu,  $m-1$   
~  $\overline{S_d}$  prebacuje brojilo u stanje  $2^n-1 = W^n$  :  
slijedeći ga impuls prebacuje u  $0 \bmod 2^n$



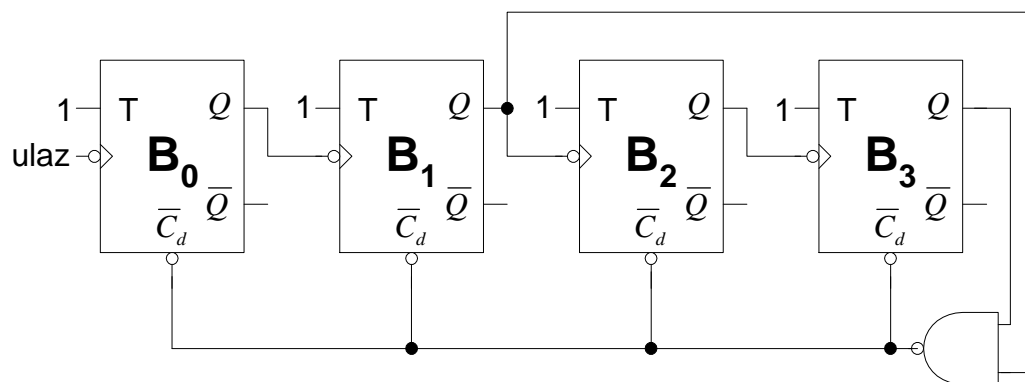
- prekid aktiviran *prvim stanjem izvan ciklusa*,  $m$   
~  $\overline{C_d}$  prebacuje brojilo u stanje 0:



# Asinkrona brojila

## Primjer: dekadsko brojilo

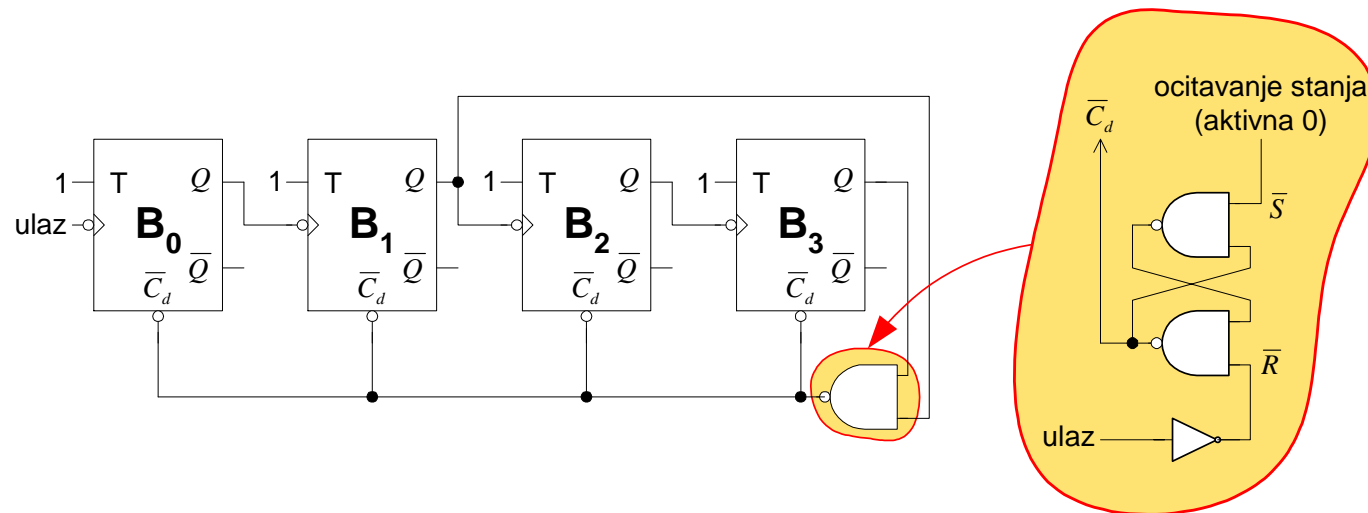
- detektirati karakterističnu pojavu  $B_3B_1 = 1$   
( $\rightarrow$  brojilo broji *naprijed*: jednostavno dekodiranje)
- problem kod brisanja bistabila  
~ *rasipanje*  $t_{db}$ :  
nestanak impulsa brisanja *prije* brisanja svih bistabila!



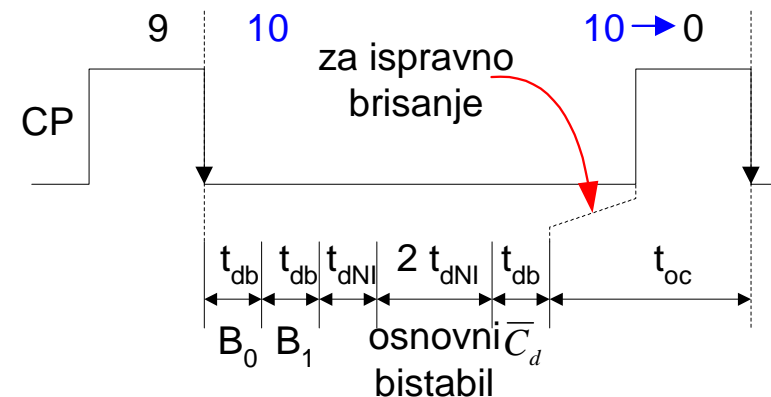
CP	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 $\equiv$ 0	1	0	1	0
	0	0	0	0

# Asinkrona brojila

- rješenje problema brisanja:  
~ osnovni bistabil u "petlju povratne veze"

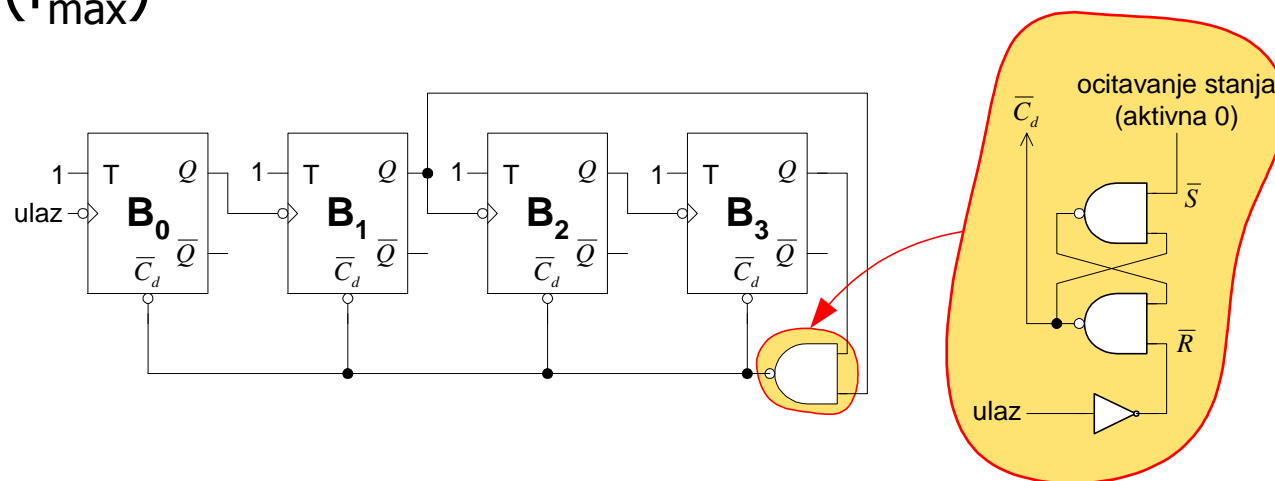


- sigurno generiranje impulsa brisanja  
~ traje do slijedećeg CP = 1

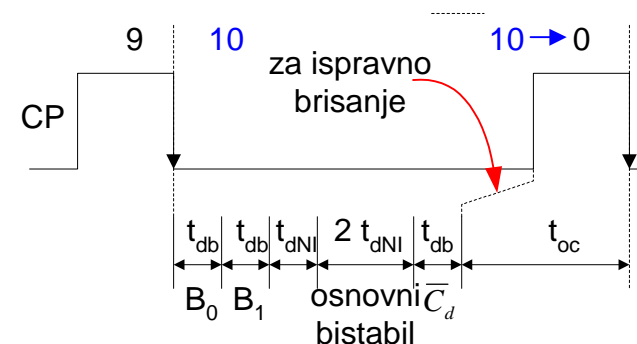


# Asinkrona brojila

- računanje  $f_{\max}$  za očitavanje stanja 0
- uzeti  $\min(f_{\max})$



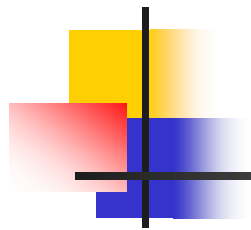
$$f_{\max} = \left\{ \frac{1}{4 \cdot t_{db} + t_{oč}} \right. \\ \left. \frac{1}{2 \cdot t_{db} + t_{db} + t_{dNI} \cdot (+2 \cdot t_{dNI}) + t_{oč}} \right\}$$



# Sinkrona brojila

- *binarno sinkrono brojilo*:
  - ~ struktura brojila iz *rekurzivne* definicije mehanizma promjene stanja
  - prvi bistabil  $B_0$  mijenja stanje uvijek:  $T_0 = 1$
  - i-ti bistabil  $B_i$  mijenja stanje kad su svi prethodni bistabili u 1:  
 $T_i = B_0 \cdot B_1 \cdot \dots \cdot B_{i-1}$

CP	$B_2$	$B_1$	$B_0$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



# Sinkrona brojila

- izvođenje *strukture*  
*n*-bitnog binarnog sinkronog brojila:

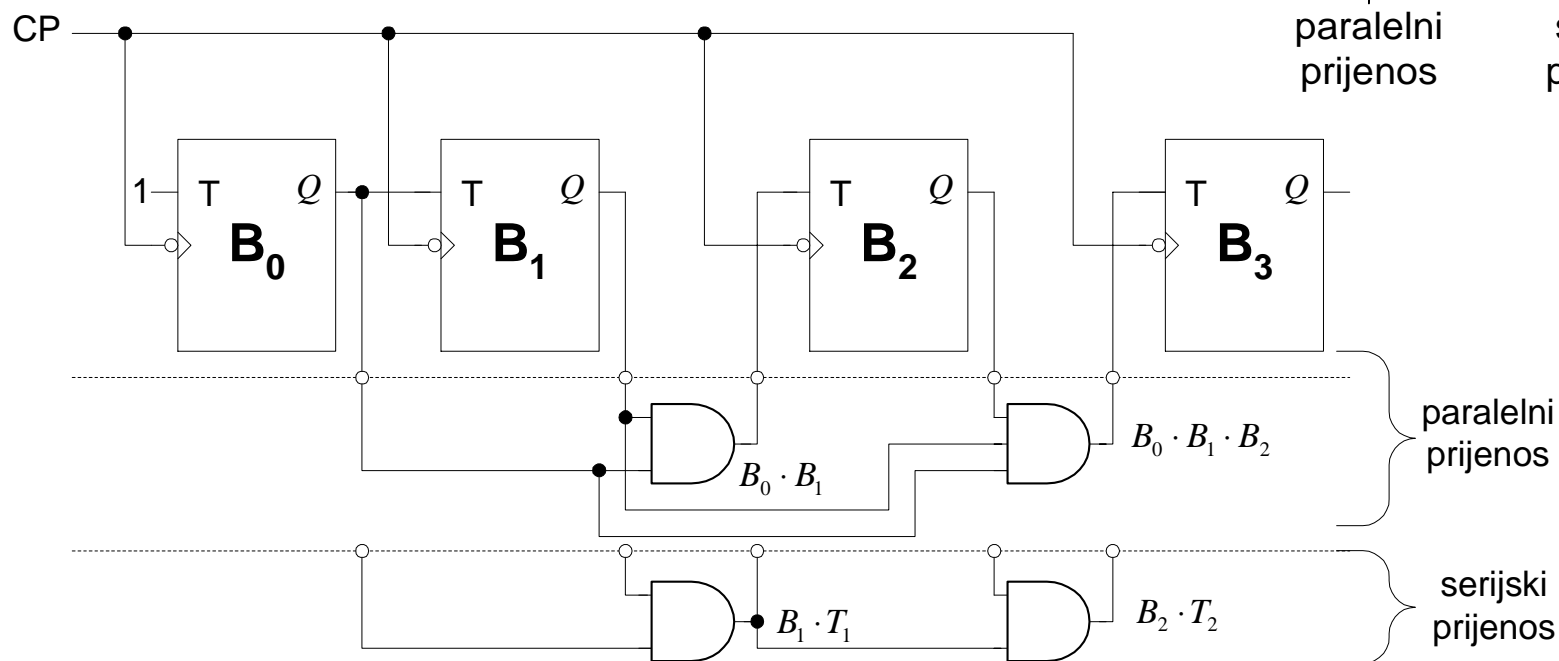
$$\begin{array}{llll} T_0 & = & 1 & = & 1 \\ T_1 & = & B_0 & = & B_0 \\ T_2 & = & B_1 \cdot B_0 & = & B_1 \cdot (B_0) & = & B_1 \cdot T_1 \\ T_3 & = & B_2 \cdot B_1 \cdot B_0 & = & B_2 \cdot (B_1 \cdot B_0) & = & B_2 \cdot T_2 \\ & \dots & & & \dots & & \\ T_i & = & B_i \cdot \dots \cdot B_0 & = & B_i \cdot (B_{i-2} \cdot \dots \cdot B_0) & = & B_{i-1} \cdot T_{i-1} \\ & \dots & & & \dots & & \\ T_{n-1} & = & \underbrace{B_{n-2} \cdot \dots \cdot B_0}_{\text{paralelni prijenos}} & = & B_{n-2} \cdot (B_{n-3} \cdot \dots \cdot B_0) & = & \underbrace{B_{n-2} \cdot T_{n-2}}_{\text{serijski prijenos}} \end{array}$$

CP	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

# Sinkrona brojila

- struktura  $n$ -bitnog binarnog sinkronog brojila:

$$\begin{array}{lll}
 T_0 & = 1 & = 1 \\
 T_1 & = B_0 & = B_0 \\
 T_2 & = B_1 \cdot B_0 & = B_1 \cdot T_1 \\
 T_3 & = B_2 \cdot B_1 \cdot B_0 & = B_2 \cdot T_2 \\
 \dots & & \dots \\
 T_i & = B_i \cdot \dots \cdot B_0 & = B_{i-1} \cdot T_{i-1} \\
 \dots & & \dots \\
 T_{n-1} & = \underbrace{B_{n-2} \cdot \dots \cdot B_0}_{\text{paralelni prijenos}} & = \underbrace{B_{n-2} \cdot T_{n-2}}_{\text{serijski prijenos}}
 \end{array}$$



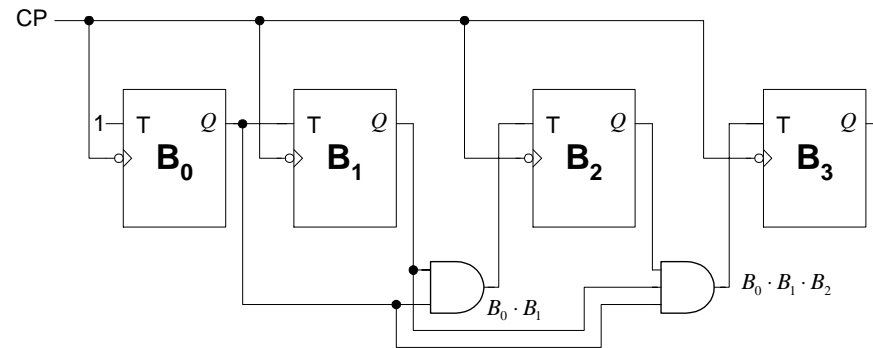
# Sinkrona brojila

- binarno sinkrono brojilo s *paralelnim* prijenosom:

- posebni I-sklop za svaki  $T_i$
- brže rješenje  
~ *samo jedan* I-sklop:

$$f_{\max} = \frac{1}{t_{\text{setup}} + t_{\text{db}} + t_{\text{dI}}}$$

- za n ↗ izvedba je kontraproduktivna  
~ teškoće pri ostvarivanju I-sklopa,  $C_{\text{rasipno}}$  ↗, itd.



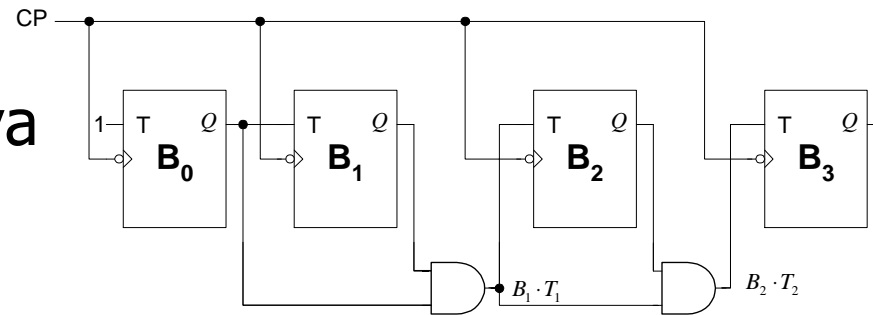


# Sinkrona brojila

- binarno sinkrono brojilo sa *serijskim* prijenosom:

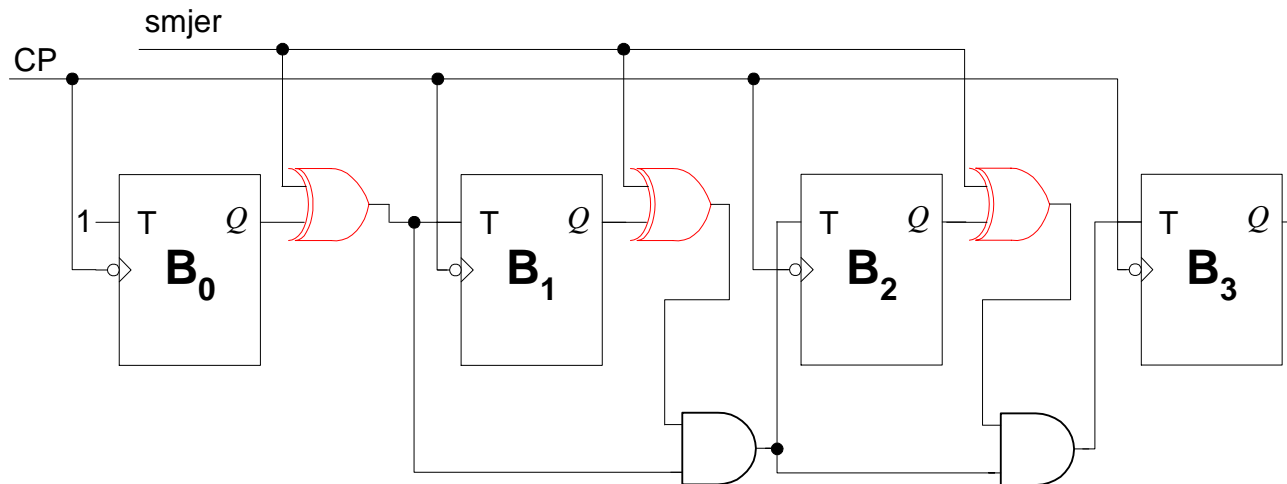
- *kaskadiranje* I-sklopova
- jeftinije rješenje  
~ *istovrsni* sklopovi  
s ograničenim brojem ulaza  
(i to samo 2!)
- sporije rješenje:

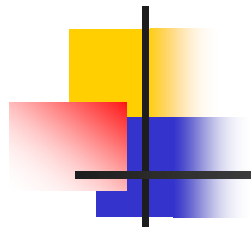
$$f_{\max} = \frac{1}{t_{\text{setup}} + t_{\text{db}} + (n-2) \cdot t_{\text{dI}}}$$



# Sinkrona brojila

- *brojilo naprijed-natrag:*
  - mreža za izbor "smjera brojanja"  
~ MUX za prenošenje  $Q_i$  ili  $\bar{Q}_i$
  - tipična izvedba: EX-ILI  
(uzeti u obzir kod računanja  $f_{\max}$ )



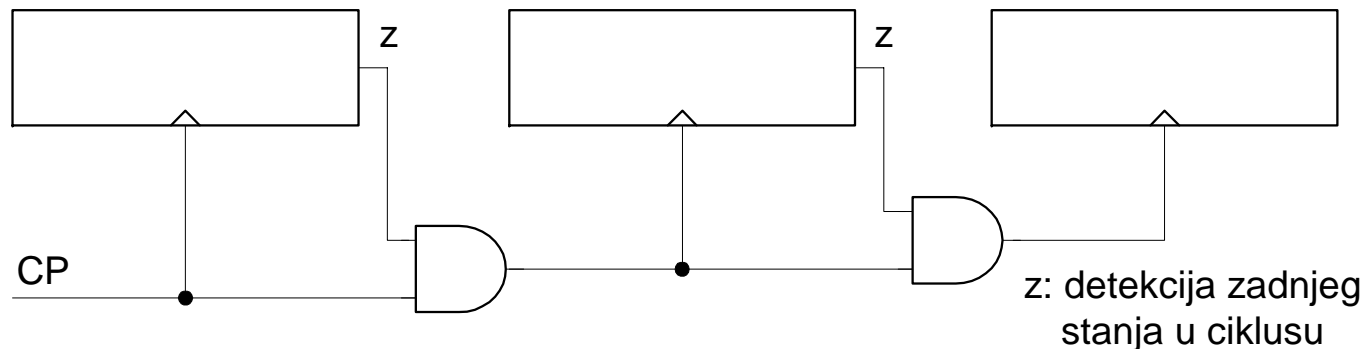


# Sinkrona brojila

- *brojilo modulo m,  $m \neq 2^n$ :*
  - projektiranje kao proizvoljni sekvencijski sklop  
~ mogućnost izbora koda:
    - jednostavniji dekodler
    - ugradnja "sigurnog starta"
  - posebno za sekvencijske module  
~ integrirana brojila:  
broje u *binarnom* sustavu
    - prethodno postavljanje (engl. presetting):
      - početno stanje: 2-komplement baze m
      - m-ti impuls:  $(2^n - 1) \rightarrow \overline{m}^2$
    - detekcija maksimalnog broja:  $W = m - 1$ 
      - m-ti impuls:  $(m - 1) \rightarrow 0$

# Sinkrona brojila

- *integrirana brojila:*
  - uglavnom 4-bitni MSI moduli: npr. serija 74
    - asinkrono: 7493
    - sinkrono binarno: 74163
    - sinkrono naprijed-natrag: 74191
  - proširivanje broja bitova  
~ veći broj bitova:  
*kaskadiranje*





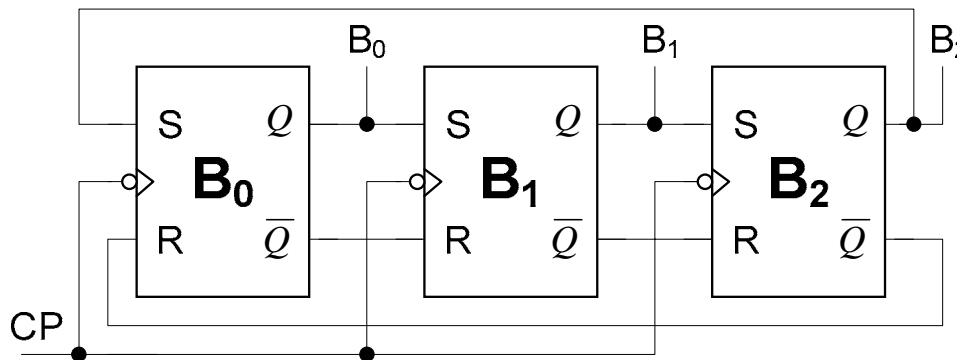
# Sinkrona brojila

---

- *brojila na osnovi posmačnog registra:*
  - struktura:
    - ~ povratna veza s izlaza posmačnog registra na njegov ulaz
  - *dvije* mogućnosti:
    - prstenasto brojilo
      - ~ povratna veza ( $D_0 = Q_{n-1}$ )
      - + početno samo jedna 1 u posmačnom registru
    - Johnsonovo brojilo:
      - $D_0 = \overline{Q}_{n-1}$

# Sinkrona brojila

- *prstenasto brojilo* (engl. ring counter)
  - ~ brojanje impulsa na "ulazu" CP posmakom 1:  
brojilo modulo broj bistabila

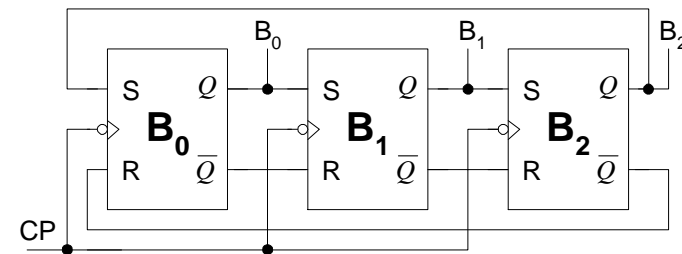


CP	B <sub>0</sub>	B <sub>1</sub>	B <sub>2</sub>
0	1	0	0
1	0	1	0
2	0	0	1
3	1	0	0

- brojilo u užem smislu
  - ~ u posmačnom registru cirkulira *samo jedna 1*
- djelitelj frekvencije:
  - ~ početno upisati uzorak *različit* od  
"sve 0" = 0, i "sve 1" =  $(2^n - 1)$

# Sinkrona brojila

- *prstenasto brojilo*:
  - baza (modul) = broj bistabila  
~ neefikasno, ali *brže* od binarnog brojila!
  - direktno očitavanje stanja  
~ stanje  $\sim (B_i = 1)$ :  
vrlo povoljno  $\rightarrow$  *ne treba* dekodera!
  - osigurati *sigurni start*!



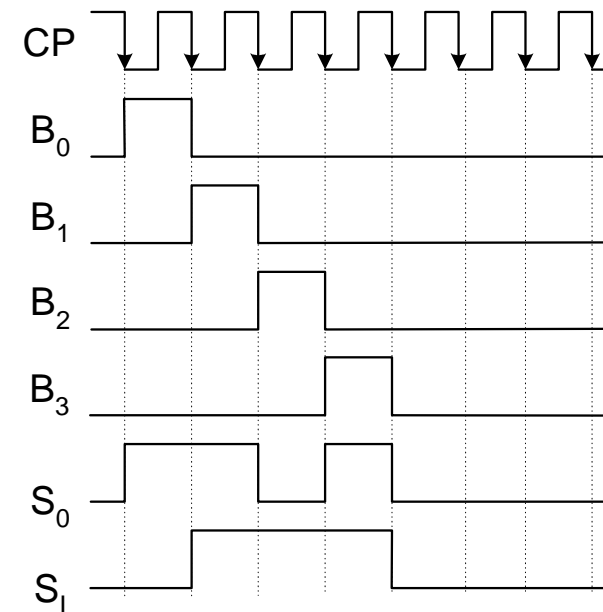
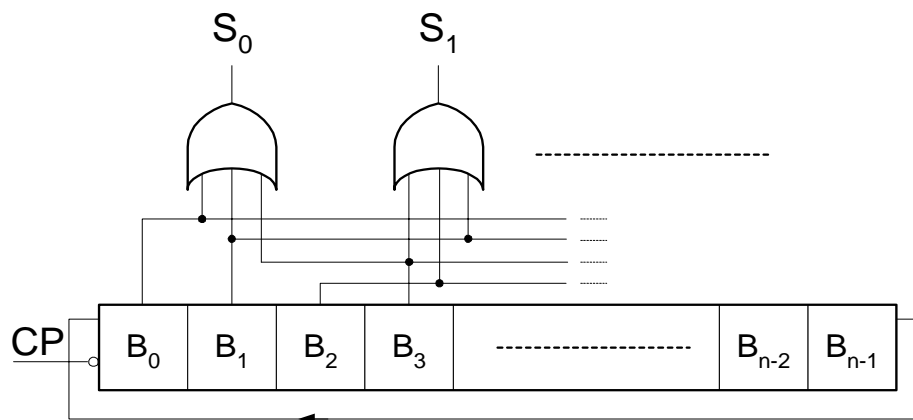
# Sinkrona brojila

- popularne izvedbe *upravljačkih jedinica* računala:
  - prstenasto brojilo
  - proizvoljni valni oblik  
~ kombiniranje (funkcija ILI) izlaza pojedinih bistabila

*Primjer:*

$$S_0 = B_0 + B_1 + B_3 + \dots$$

$$S_1 = B_1 + B_2 + B_3 + \dots$$

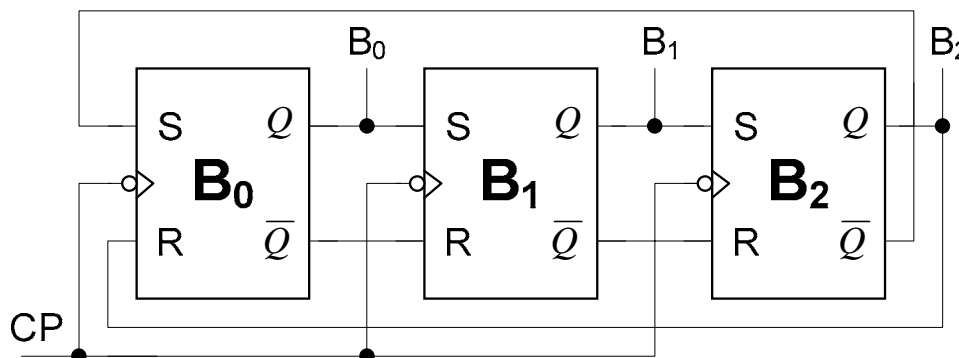




# Sinkrona brojila

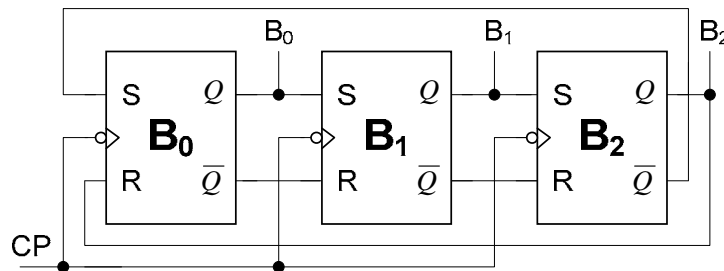
- *Johnsonovo brojilo*, brojilo s ukrštenim prstenom (engl. twisted ring counter):
  - povećanje broja stanja za dani broj bistabila:  $2 \cdot n$ 
    - ukrstiti povratnu vezu  
~ bistabili SR i JK
    - na ulaz dovesti  $\overline{Q}_{n-1}$   
~ bistabil D
    - broje u kodu s  $d_{\min} = 1$
  - i dalje brže od binarnog brojila!

CP	B <sub>0</sub>	B <sub>1</sub>	B <sub>2</sub>
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0



# Sinkrona brojila

- *dekodiranje stanja* Johnsonovog brojila:
  - nije tako povoljno kao kod prstenastog brojila
  - ipak relativno jednostavno  
 $\sim$  konjunkcija dva *susjedna* izlaza  $B_i$  i  $\overline{B}_i$



CP	$B_0$	$B_1$	$B_2$
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0

$$0: \overline{B}_2 \overline{B}_0$$

$$1: \overline{B}_1 B_0$$

$$2: \overline{B}_2 B_1$$

$$3: B_2 B_0$$

$$4: B_1 \overline{B}_0$$

$$5: B_2 \overline{B}_1$$

# Generatori sekvencije

- *generator sekvencije* (engl. sequence generator):
  - generiranje *propisane* sekvenc(ij)e bitova  
~ ponavlja se!
  - duljina sekvencije  
~ broj uzastopnih bitova  
koji se ponavljaju
  - sekvencija  
~ *izlaz* posmačnog registra

*Primjer:*

...011100101110010111...



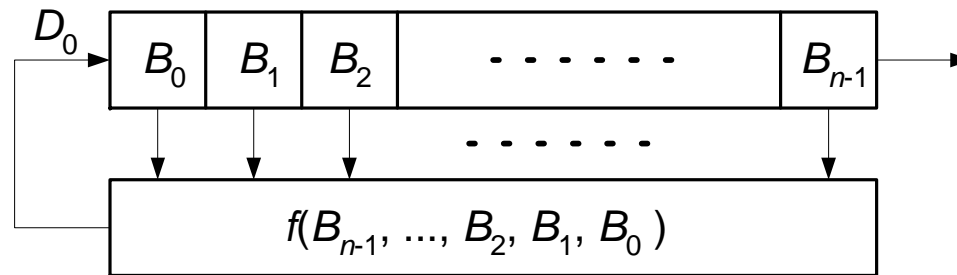
$B_0$	$B_1$	$B_2$	
$B_{n-4}$	$B_{n-3}$	$B_{n-2}$	$B_{n-1}$
1	1	1	0
0	1	1	1
0	0	1	1
1	0	0	1
0	1	0	0
1	0	1	0
1	1	0	1
1	1	1	0

sekvencija

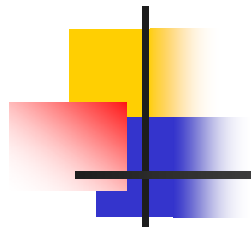
# Generatori sekvencije

- izvedba generatora sekvencije:
  - poopćenje povratne veze posmačnog registra:

$$D_0 = f(B_{n-1}, \dots, B_1, B_0)$$



- specijalni slučaj:
  - prstenasto brojilo:  $D_0 = B_{n-1}$
  - Johnsonovo brojilo:  $D_0 = \overline{B_{n-1}}$



# Generatori sekvencije

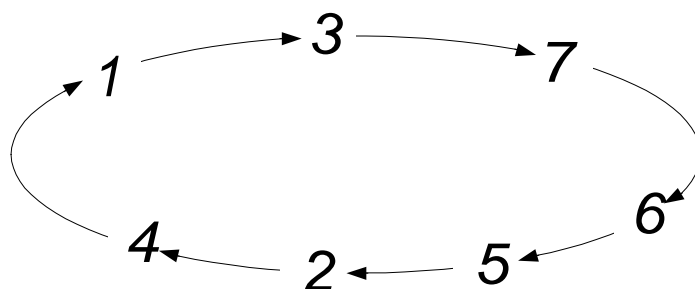
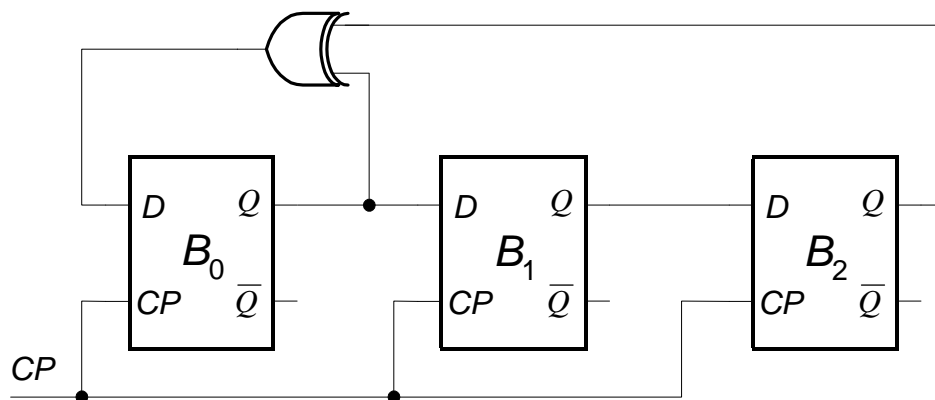
- naročito jednostavna izvedba povratne veze  
~ *linearna* funkcija:

$$f(x_{n-1}, \dots, x_1, x_0) = c_{n-1}x_{n-1} \oplus \dots \oplus c_1x_1 \oplus c_0x_0, \quad c_0 \in \{0,1\}$$

- *posmačni registar s linearnom povratnom vezom*  
(engl. Linear Feedback Shift Register, LFSR):
  - jednostavna struktura sklopa  
~ samo sklopovi EX-ILI
  - najveća moguća *duljina sekvencije* (za n bistabila)  
~  $2^n - 1$
  - zabranjeno stanje 00..00  
~ izbjeći to stanje:  
sklop za sigurni start

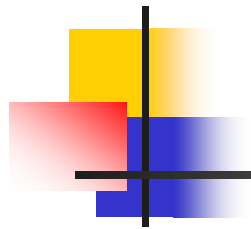
# Generatori sekvencije

*Primjer:*  $D_0 = f(B_2, B_1, B_0) = B_2 \oplus B_0$



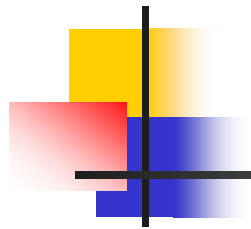
$B_0$	$B_1$	$B_2$	$D_0$
1	0	0	1
1	1	0	1
1	1	1	0
0	1	1	1
1	0	1	0
0	1	0	0
0	0	1	1
1	0	0	1

sekvencija



# Generatori sekvencije

- primjena generatora sekvencije:
  - generiranje *pseudoslučajne* sekvencije bitova  
~ "vrlo duga" sekvencija (ali se ponavlja!),  
*generator pseudoslučajne sekvencije*  
(engl. Pseudo-Random Sequence Generator):
    - "randomizacija" bitovnih nizova (engl. scrambling)
    - zaštitni bitovi prilikom prijenosa
    - tajni ključevi za kriptiranje
    - *ispitni vektori* za ispitivanje digitalnih sklopova
  - očitavanje *stanja* posmačnog registra  
~ *generator pseudoslučajnih brojeva*  
(engl. Pseudo-Random Number Generator, PRNG)



# Memorije

---

- memorija
  - ~ digitalni (pod)sustav za pamćenje *većeg broja* podataka:
  - nadogradnja koncepta pamćenja
    - ~ osnovna interpretacija:  
*skup* registara
    - bistabil
      - ~ 1 bit
    - registar
      - ~ 1 podatak ("riječ")
    - memorija
      - ~ više riječi
  - osnovna jedinica informacije
    - ~ *memorijska riječ*

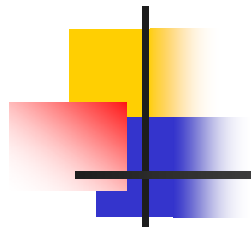




# Memorije

---

- funkcija pamćenja  
~ primitivne (pod)funkcije:
  - pamćenje grupa bitova
  - pristup podacima
  - *čitanje* iz memorije, *pisanje* u memoriju
- organizacija memorije:
  - memorijsko polje  
~ skup memorijskih ćelija
  - sklopovi za pristup podacima
  - sklopovi za čitanje i pisanje



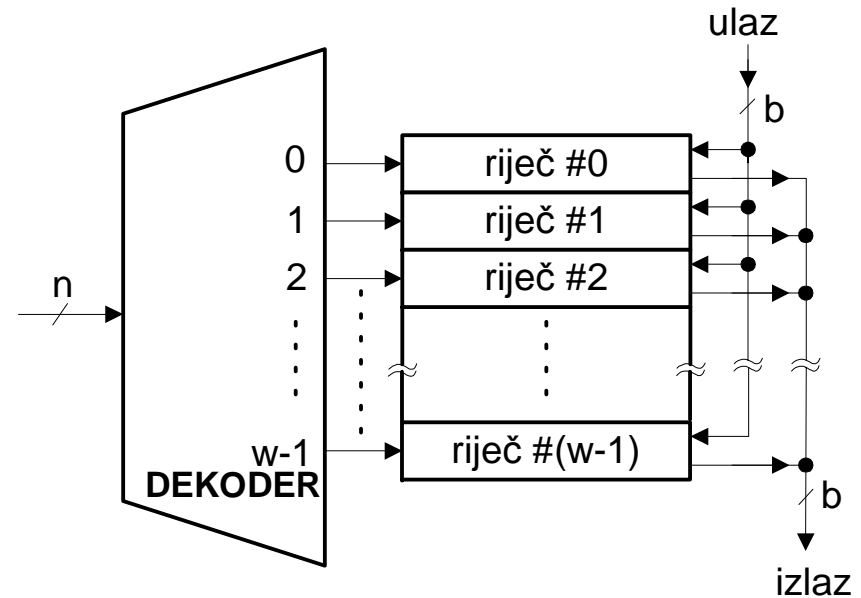
# Memorije

---

- svojstva *memorijskih ćelija*:
  - *razlučiva* i lako prepoznatljiva stanja
  - održavanje stabilnog stanja:
    - postojeane ćelije/memorije  
~ bez utroška energije proizvoljno dugo ostaju u *stabilnom* stanju;  
npr. magnetske memorije
    - nepostojane ćelije/memorije  
~ troše energiju za pamćenje,  
gube informaciju kod prestanka napajanja;  
npr. poluvodičke memorije

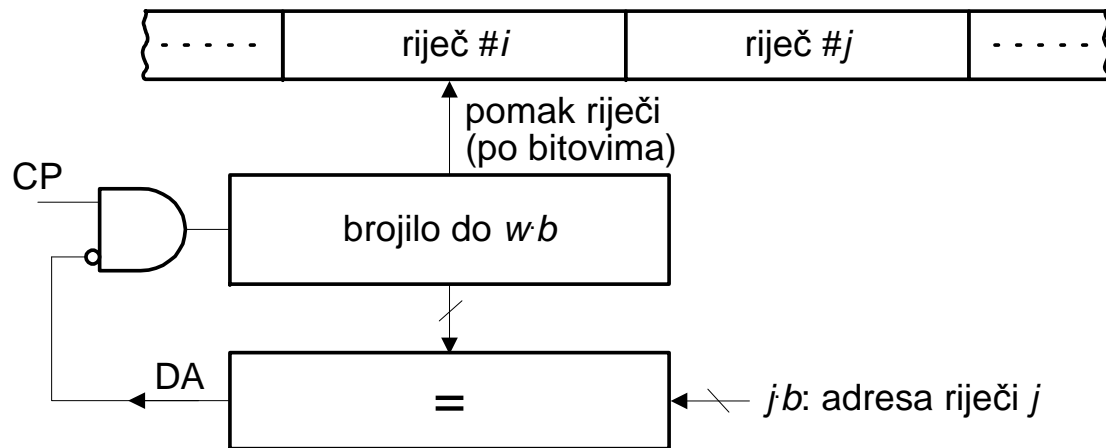
# Memorije

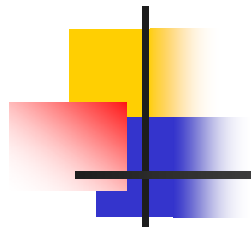
- *paralelni pristup* podacima  
~ paralelno *svakoj* riječi
  - vrijeme pristupa jednako za sve riječi  
~ *paralelne* memorije
- nasumični pristup:  
~ pristup pojedinoj riječi *ne ovisi* o prethodnim pristupima  
(engl. Random Access Memory, RAM)
- *adresa*:  
~ "redni broj" riječi unutar memorije
$$W = 2^n \rightarrow a_{n-1} \dots a_1 a_0$$



# Memorije

- *serijski pristup* podacima  
~ serijski po riječima (i bitovima):
  - pristup adresiranoj riječi  
~ od *prve* ili od *prethodno adresirane* riječi:  
*serijske (sekvencijalne)* memorije;  
npr. mg. trake, mg. diskovi, "veliki posmačni registri"
  - *bitno različito* vrijeme pristupa pojedinim riječima

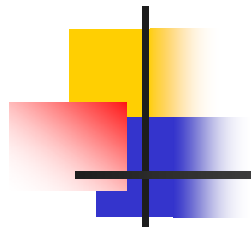




# Memorije

---

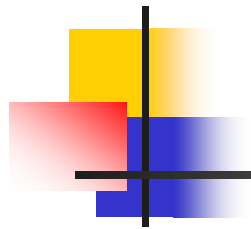
- *čitanje i pisanje:*
  - "upisno-ispisne memorije"
    - ~ memorije promjenjivog sadržaja,  
"memorije" u užem smislu;  
(paralelne upisno-ispisne memorije: RAM)
  - ispisne memorije = *permanentne* memorije (ROM)
    - ~ u odnosu na čitanje, zanemarivo mali broj pisanja



# Memorije

---

- *funkcijska podjela*
  - ~ smještaj u odnosu na digitalni sustav/procesor:
    - *unutarnje (interne)*, "operativne"
      - ~ paralelne, velikih brzina (~ brzina CPU)
    - *vanjske (eksterne)*
      - ~ CPU *ne* komunicira s njima neposredno:
        - metoda pristupa podacima
          - ~ putem glavne memorije
        - znatno veći kapacitet,  
znatno manja brzina ( $t_a \sim 10 \text{ ms}$ )
        - uređaji
          - ~ diskovi, trake



# Memorije

---

- unutarnje, "operativne" memorije:
  - *glavna memorija*:
    - ~ glavni spremnik za programe, podatke, operacijski sustav, itd.
    - veličina (kapacitet) nije tako brza kao CPU
  - *priručna memorija* (engl. cache)
    - ~ pohranjivanje manje količine upravo korištenih instrukcija (odsječak programa) i podataka
    - manja memorija, brzina ~ CPU, skupa!
  - primarna p. m. (engl. L1 cache):
    - ~ u sklopu CPU
  - sekundarna p. m. (engl. L2 cache):
    - ~ na M/B



# Memorije

---

- karakteristični *parametri memorija*:
  - *kapacitet* memorije:  $C = W \cdot b$   
~ ukupna količina bitova, okteta/riječi (bajtova)
  - *vrijeme pristupa*,  $t_a$ :  
~ vrijeme potrebno za dohvat podatka,  
od zahtjeva za čitanje do pojave podatka  
na izlazu, uz prethodno postavljenu adresu  
npr.  $50 \text{ ns} < t_a \text{ tipično} < 500 \text{ ns}$
  - *prosječno vrijeme pristupa*,  $\overline{t_a}$ :  
~ za *sekvencijalne* memorije  
$$\overline{t_a} = (t_a)_{\max} / 2$$

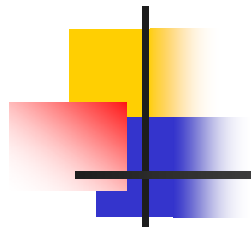




# Memorije

---

- karakteristični *parametri memorija*:
  - *vrijeme ciklusa,  $t_c$* :  
~ minimalno moguće vrijeme između dva čitanja ili pisanja;  
za *paralelne* memorije  
općenito  $t_c > t_a$
  - *maksimalna frekvencija memorije:  $f_M = t_c^{-1}$*   
~ maksimalni broj čitanja/pisanja u jedinici vremena [sec];  
*širina* (frekvencijskog) *pojasa* memorije,  
u analogiji s elektroničkim pojačalima,  
jer  $f_{\min} = 0$  (kad se *ne* čita/piše)



# Statičke memorije

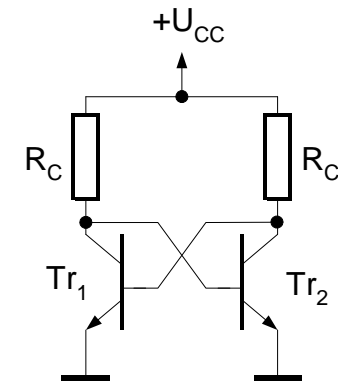
---

- statičke memorije
  - ~ memorijska ćelija *bistabil*
    - nepostojana memorija
    - elektronskim cijevima već u prvim digitalnim računalima (ENIAC, 1945)
    - danas poluvodička tehnologija
      - ~ posebno pojednostavljeni sklopovi:
        - što manja površina na čipu
          - ~ veća gustoća pakiranja,
          - manja vjerojatnost defekata
        - što manja disipacija

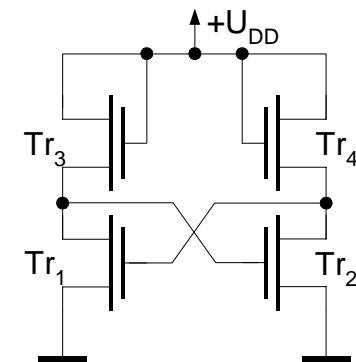
# Statičke memorije

- princip izvedbe memorijskog bistabila:

- bipolarna izvedba  
 ~ manje elemenata, mali hod  $u_{izl}$   
 (pobuda logičkih sklopova?)



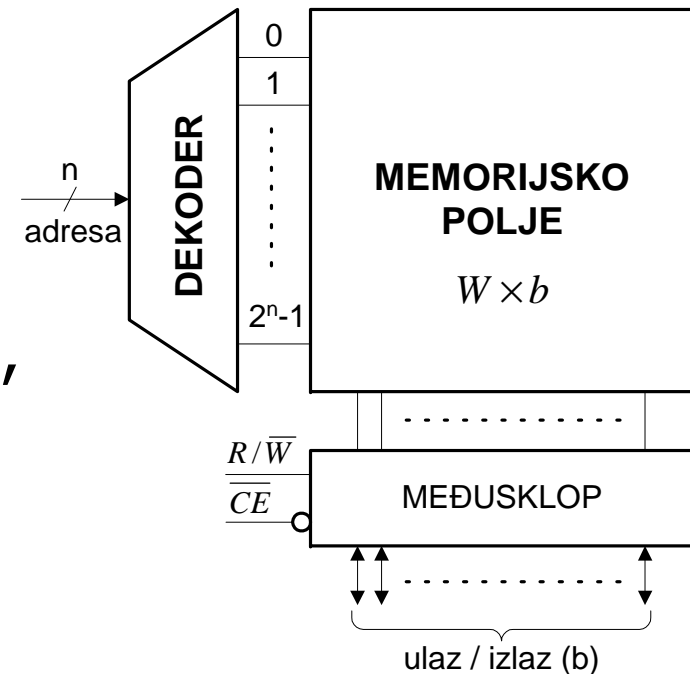
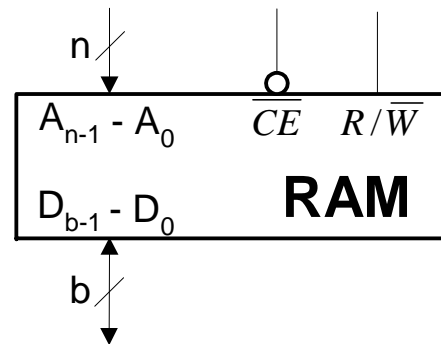
- izvedba s MOSFETom:  
 ~ NMOS

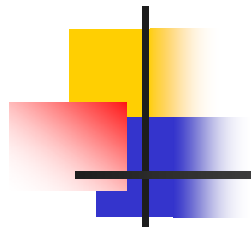


	bipolarne	MOSFET
kapacitet	manji	veći
brzina	veća	manja
cijena	veća	manja

# Organizacija memorijskih modula

- *organizacija* memorijskih modula  
~ karakteristični funkcijski blokovi:
  - dekodeer adrese
  - memorijsko polje
  - međusklop:  
~ pojačala za čitanje/pisanje:  
konverzija električkih razina,  
dvolinijskih reprezentacija





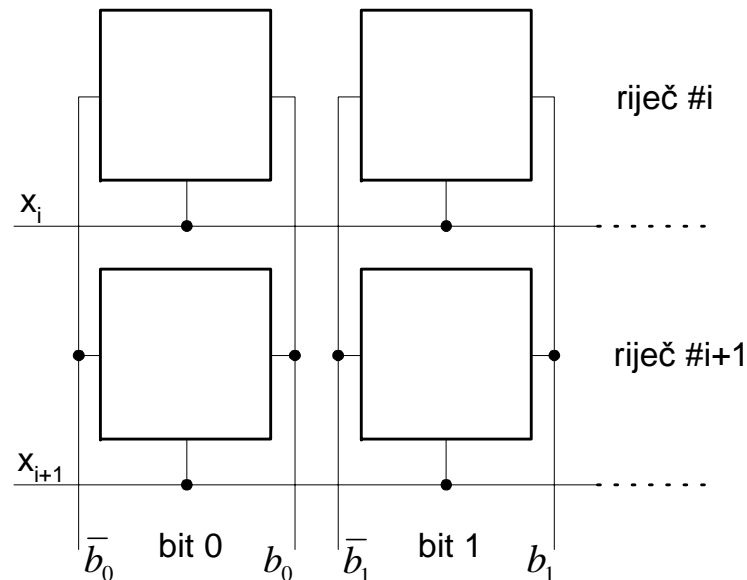
# Organizacija memorijskih modula

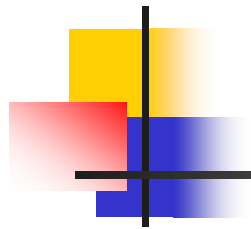
---

- organizacija *memorijskog polja*:
  - 2 D - "dvodimenzijska"  
~ linearno adresiranje
  - 3 D - "trodimenzijska"  
~ koincidentno adresiranje
  - 2 1/2 D  
~ implementacijsko poboljšanje 2 D

# Organizacija memorijskih modula

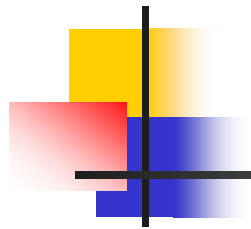
- 2 D - "dvodimenzijska" organizacija:
  - $b$  ćelija jedne riječi  
~ *samo jedna* adresna linija:  
*linijsko* adresiranje
  - 1 linija retka  
~ 1 izlaz dekodera





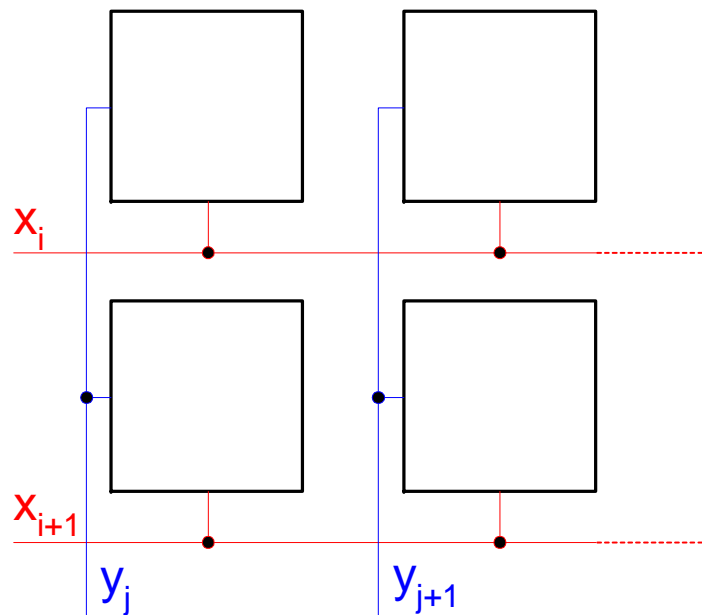
# Organizacija memorijskih modula

- 3 D - "trodimenzijska" organizacija:
  - smanjiti veličinu dekodera za  $W \gg :$ 
    - preveliki i presloženi dekodер
    - preveliki broj izlaza iz dekodera
  - *koincidentno* adresiranje:
    - ~ memorijska se ćelija adresira *koincidiranjem dvije* adresne linije (retka i stupca)
      - koordinatni razmještaj ćelija
      - ćelije s 2 adresne linije
      - 2 dekodera koji su *upola manji*!



# Organizacija memorijskih modula

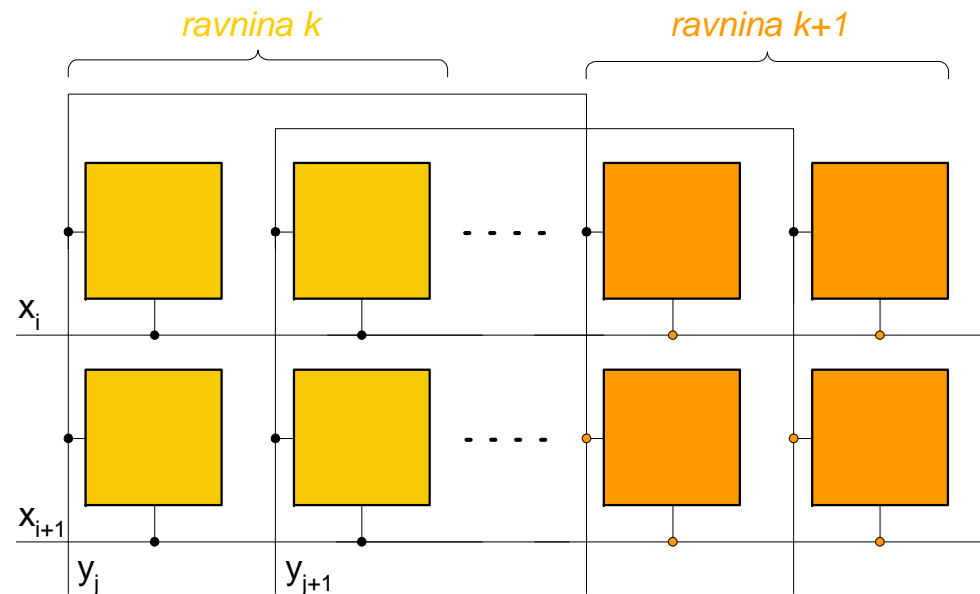
- *koincidentno* adresiranje  
~ memorijska polja *jednobitnih* riječi





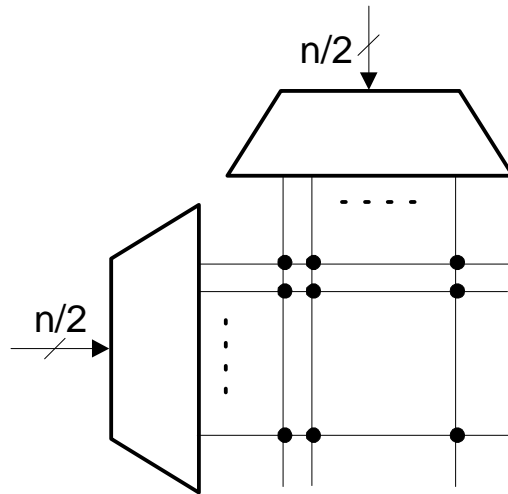
# Organizacija memorijskih modula

- *koincidentno* adresiranje  
~ memorijska polja *višebitne* riječi:  
*više ravnina* (~ memorijska polja)  
koja se "slažu" u "trećoj dimenziji"



# Organizacija memorijskih modula

- *koincidentno* adresiranje  
~ *dekodiranje* s dva dekodera:
  - broj riječi  $W$  = broj presjecišta:  
 $n/2 \rightarrow 2^{n/2} \cdot 2^{n/2} = 2^n = W$
  - broj izlaza iz dekodera:  
 $2^{n/2} + 2^{n/2} = 2 \cdot 2^{n/2} = 2\sqrt{2} = 2\sqrt{W}$



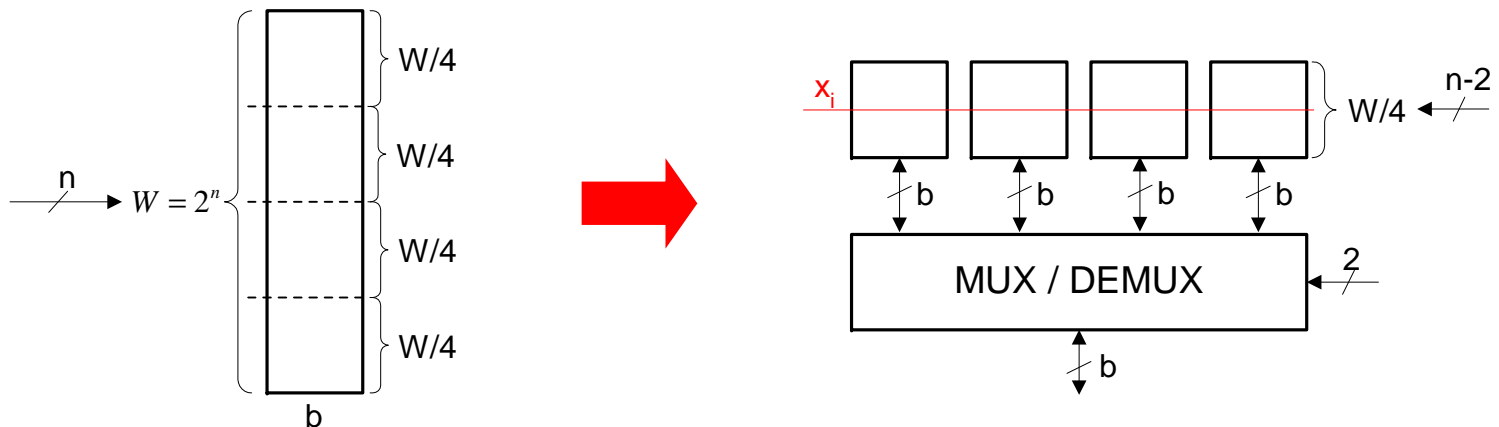


# Organizacija memorijskih modula

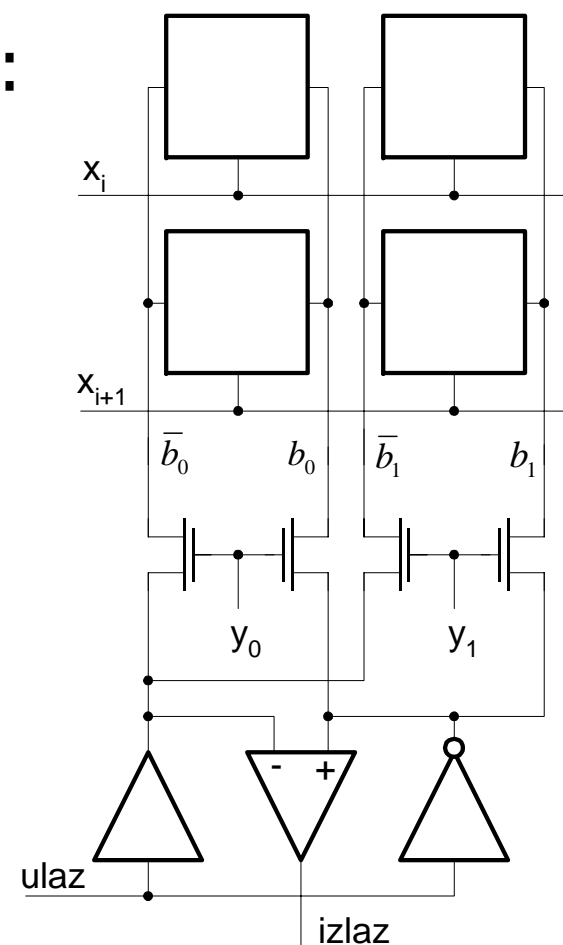
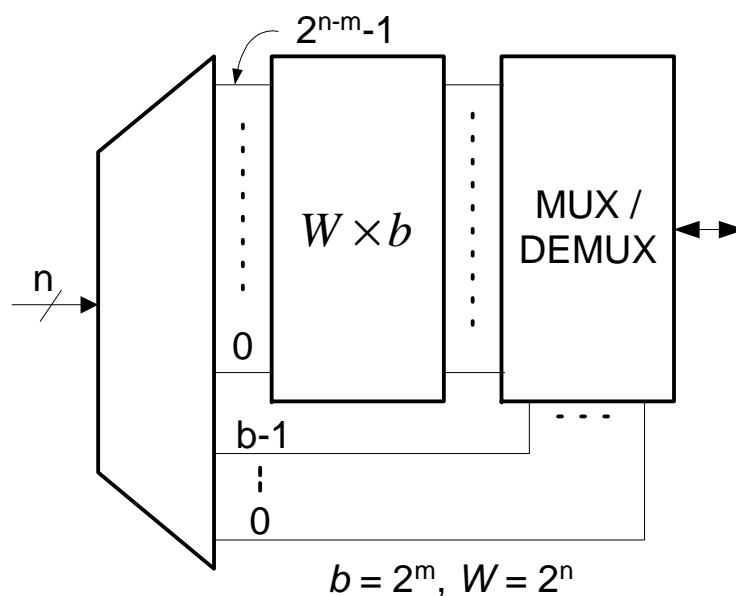
- 2 1/2 D organizacija:
  - ~ poboljšanje ponašanja 2 D organizacije:  
duljina (fizičke) linije bita =  $W \times \text{"duljina bita"}$ :
    - $C_{\text{par}} \nearrow \Rightarrow$  veća kašnjenja
    - ponašanje prijenosne linije
    - rješenje:
      - ~ podjela memorije na  
*podpolja s manjim brojem riječi*  
*povećanog broja bitova*:  
 $r$  podpolja  $\rightarrow$  1 *fizička* riječ od  $r$  *logičkih* riječi

# Organizacija memorijskih modula

- 2 1/2 D organizacija
  - ~ podjela memorije na podpolja s *manjim brojem riječi povećanog broja bitova*:
    - r podpolja → 1 *fizička* riječ od r *logičkih* riječi
    - dva dekodera:
      - za fizičku riječ
      - "pristupni" MUX/DEMUX



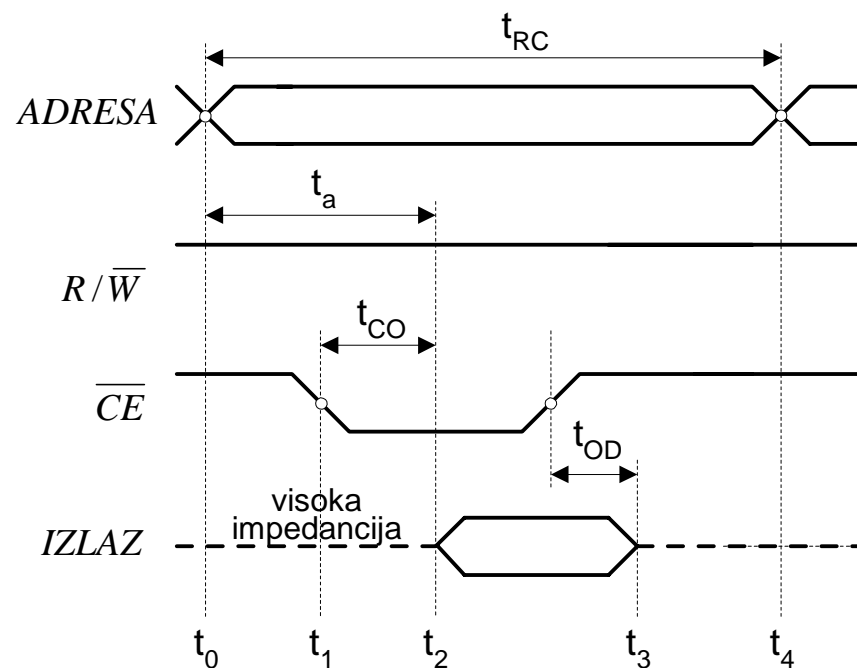
- 2 1/2 D organizacija  
~ *posebno jednostavna* izvedba pristupnog MUX/DEMUX za  $b = 1$ :  
*1-bitni* memorijski moduli



# Vremenski odnosi statičkih memorija

- ciklus čitanja:

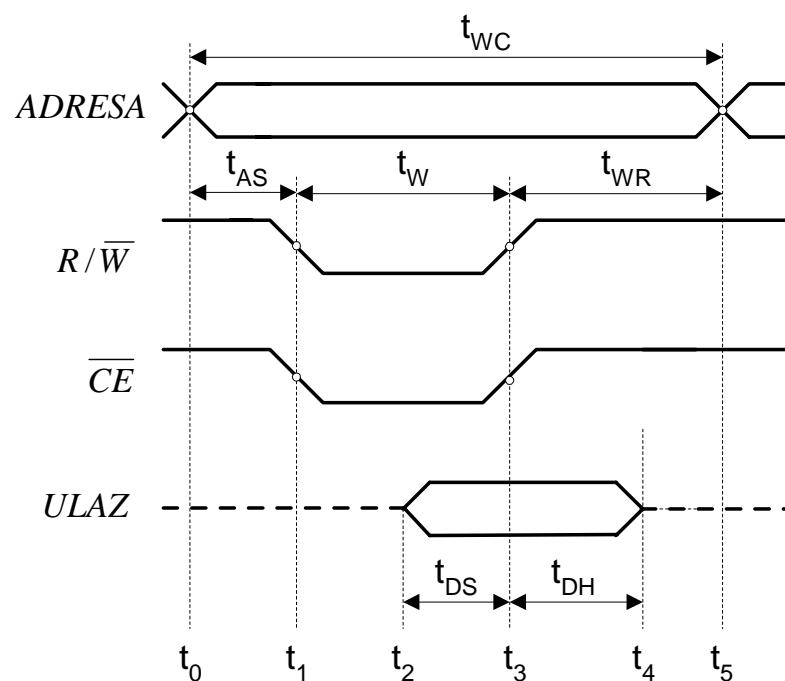
parametar	značenje
$t_{RC}$	trajanje ciklusa čitanja (između dvije promjene adrese)
$t_a$	vrijeme pristupa (pojava podataka na izlazu uz $\overline{CE} = 0$ )
$t_{CO}$	kašnjenje u odnosu na $\overline{CE}$ (pojava podataka): podaci su na izlazu nakon $\max(t_a, t_1 + t_{CO})$
$t_{OD}$	kašnjenje u odnosu na $\overline{CE} = 1$ (uklanjanje podataka na izlazu)

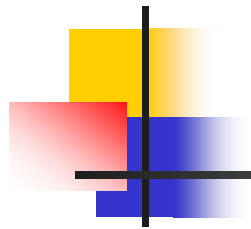


# Vremenski odnosi statičkih memorija

- ciklus *pisanja*:

parametar	značenje
$t_{WC}$	trajanje ciklusa pisanja (između dvije promjene adrese)
$t_{AS}$	vrijeme postavljanja adrese: min kašnjenje $R/\overline{W}$ i $\overline{CE}$ u odnosu na početak ciklusa
$t_W$	vrijeme pisanja; min trajanje $R/\overline{W}$ i $\overline{CE}$
$t_{WR}$	vrijeme otpuštanja (pri pisanju) ~ min vrijeme oporavka
$t_{DS}$	vrijeme postavljanja podatka
$t_{DH}$	vrijeme održavanja podatka

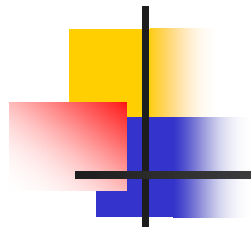




# Projektiranje statičkih memorija

- "projektiranje" memorije  
~ izgradnja memorije većeg kapaciteta  
od modula manjeg kapaciteta
  - memorija  $W' \times b'$
  - memorijski moduli  $W \times b$
  - vrijedi  $W' = 2^n > W = 2^m, b' > b$



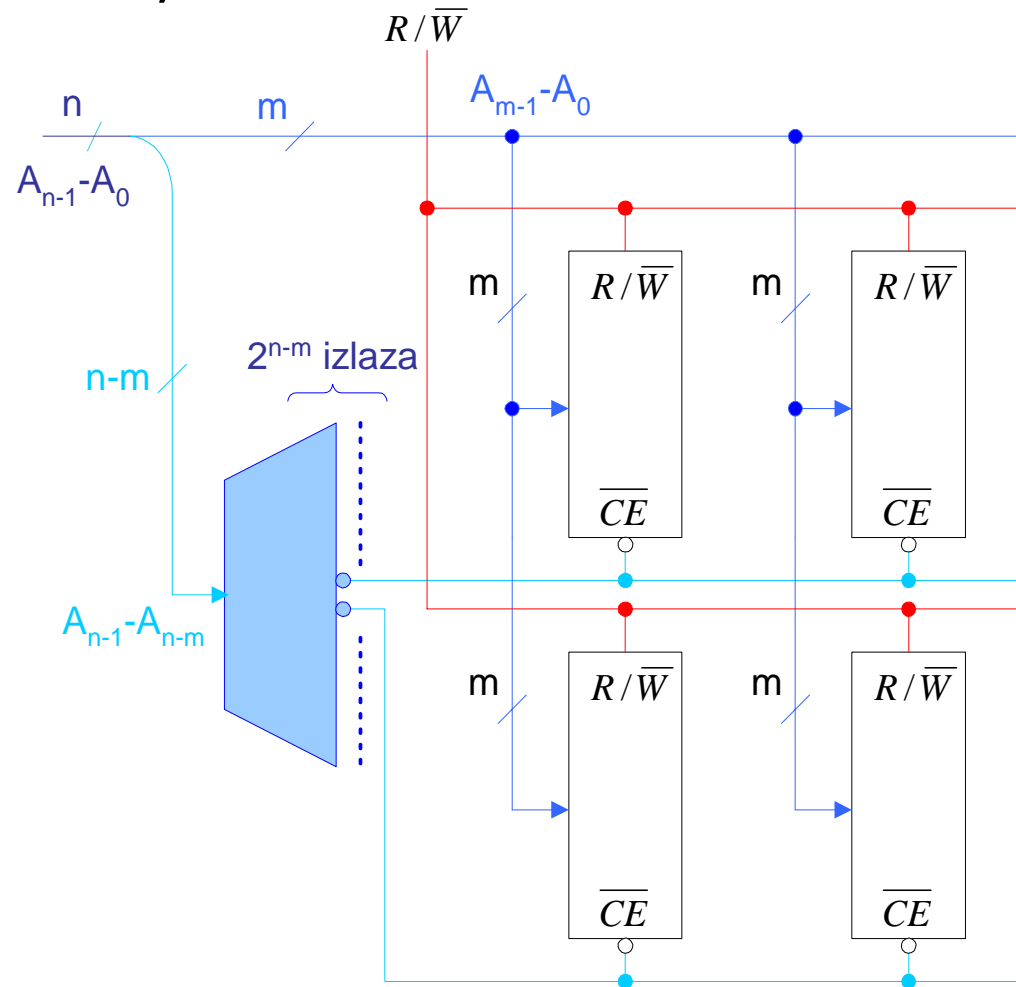


# Projektiranje statičkih memorija

- struktura memorije:
  - *dekoder za viših* (n-m) bitova adrese
  - ulazi bitova podatka  $b_i$  spajaju se skupa
  - izlazi bitova podatka  $b_i$  spajaju se skupa;  
izvedbe izlaza:
    - sklopovi s tri stanja
    - sklopovi sa slobodnim kolektorom
    - kombinacija ulaz-izlaz)
  - niži adresni bitovi: paralelno *na sve* module
  - signali  $\overline{CE}$  i  $R/\overline{W}$ : također paralelno *na sve* module

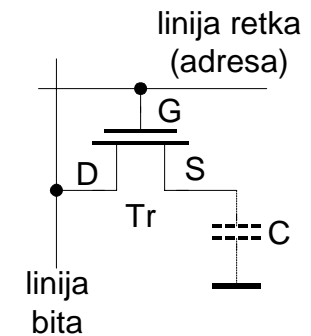
# Projektiranje statičkih memorija

*Primjer:* memorija  $W' \times b'$ , moduli  $W \times b$ ,  
 $W' = 2^n > W = 2^m$ ,  $b' > b$



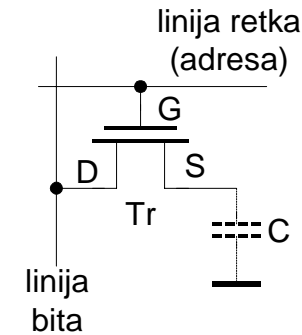
# Dinamičke memorije

- povećanje kapaciteta poluvodičke memorije  
~ smanjenje veličinu memorijske ćelije:
  - pohranjivanje bita podatka:
    - ~ naboj na C
      - dovoljan jedan tranzistor
      - $C = C_{\text{par}}$ : međuelektrodni kapacitet
  - pamćenje informacije  
~  $C_{\text{par}}$  se izbija zbog površinske vodljivosti



# Dinamičke memorije

- rješenje problema gubitka naboja dinamičke memorijske ćelije:



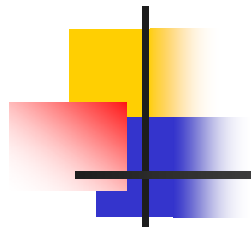
- *periodičko* čitanje sadržaja memorijske ćelije i ponovno upisivanje ( $\sim 2$  ms):  
~ *osvježavanje* (engl. refreshing)
- memorije s *dinamičkim* memorijskim ćelijama:  
~ *dinamičke memorije*, DRAM (engl. Dynamic RAM)



# Dinamičke memorije

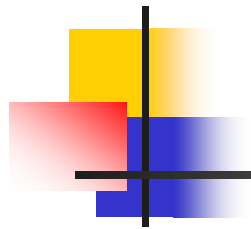
---

- *čitanje* informacije  
~ *adresirati* ćeliju:
  - očitavanje napona na C:  
~ dobro pojačalo za čitanje (engl. sense amplifier)
$$U_b \propto \frac{C}{C_b + C} \approx \frac{C}{C_b}$$
  - zbog  $C_b \gg C$  (duga linija bita!) čitanje je *destruktivno*!  
~ nakon čitanja *ponovni* upis podataka
- *pisanje* informacije  
~ adresirati ćeliju i upisati podatak s linije bita



# Dinamičke memorije

- suvremeni DRAM:
  - ~ jednotranzistorske ćelije:  
kapaciteti ~ više 100 Mbit/modul
  - drastično smanjena površina memorijske ćelije:  
npr. 64 K:  $250 \mu\text{m}^2 \rightarrow 4 \text{ M: } 8,9 \mu\text{m}^2$  (CMOS)  
~ C također smanjen!
  - povećana osjetljivost na "meke" pogreške  
~ gubitak informacije zbog  $\alpha$ -zraka  
(kozmičke zrake, Sr u materijalu kućišta IC)
  - rješenje problema:
    - posebne izvedbe C
    - dodatni sklopovi za zaštitu podatka (ECC)



# Dinamičke memorije

- organizacija DRAM:
  - ~ jedna adresna linija: 2 D ili 2 1/2 D (povoljnije)
- gustoća pakiranja bitova ↗
  - veličina chipa ↘
  - kućište ↘
  - broj izvoda (broj adresnih linija ↗)?
- *smanjiti* broj izvoda za adrese *multipleksiranjem*:
  - veličina IC ↘
    - broj IC na PCB ↗
    - kapacitet memorije ↗
  - posebno sklopovlje za multipleksiranje
  - usporavanje rada memorije

# Dinamičke memorije

*Primjer:* DRAM modul 4116/2118 (16 K x 1)

kućište: DIP sa 16 izvoda

7 adresa za 16 K =  $2^{14}$ :

$A_{13} \dots A_0 \sim$

$\text{MUX}(A_6 \dots A_0,$   
 $A_{13} \dots A_7)$

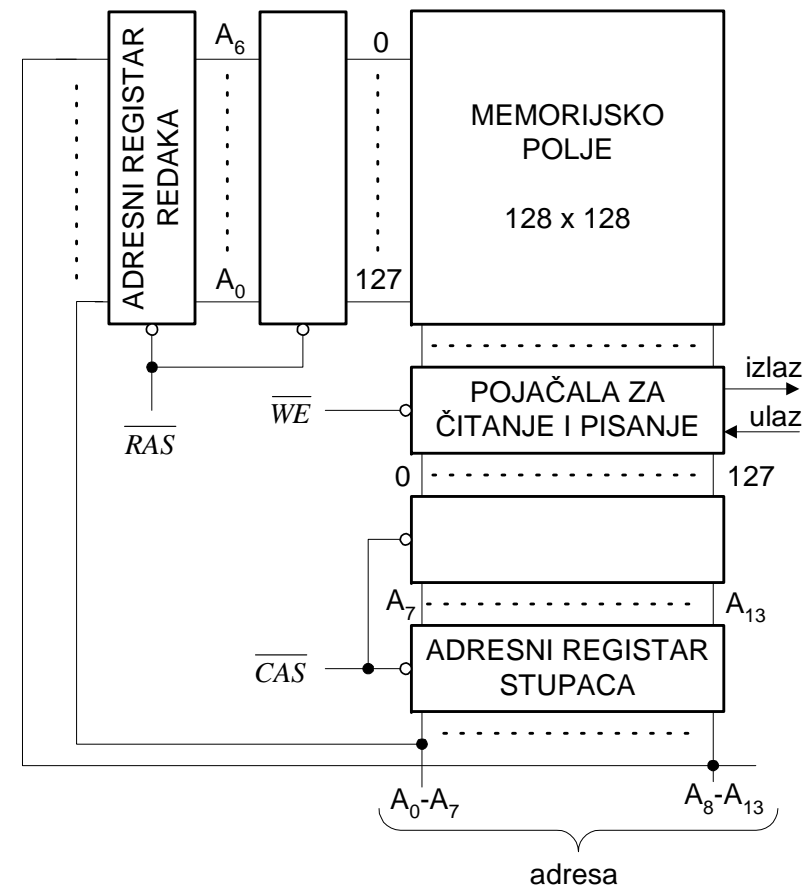
2 UI podatka

1 čitaj/piši

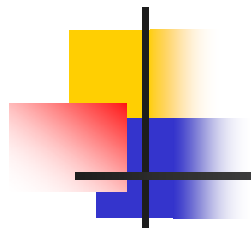
2 impulsa upisa adrese

4/2 napajanja

organizacija: 2 1/2 D







# Dinamičke memorije

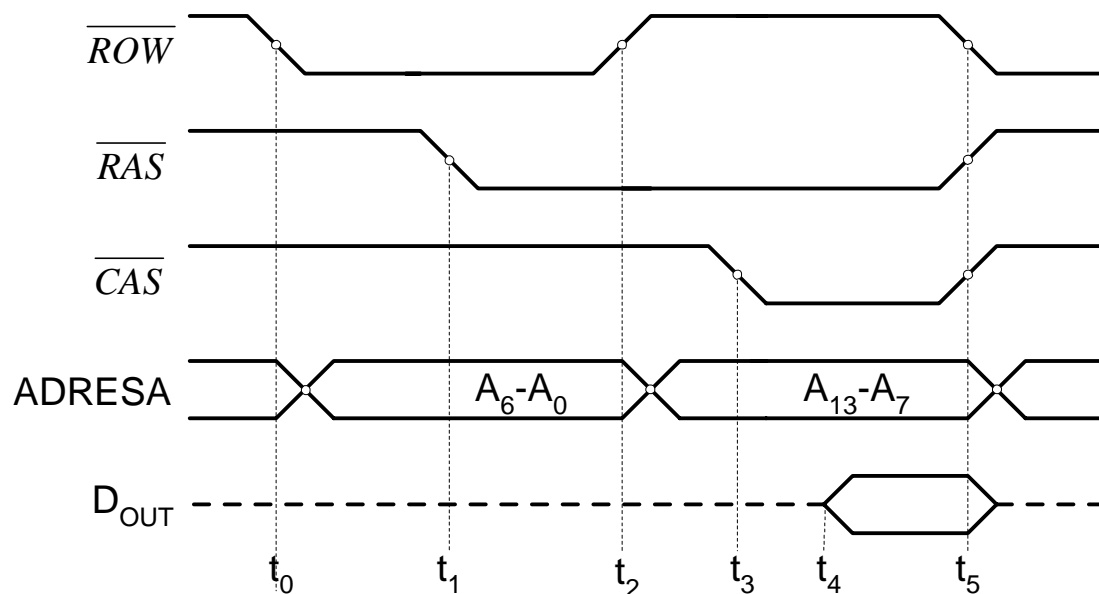
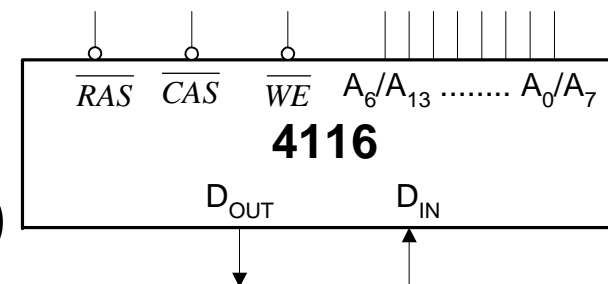
- vremenski dijagram čitanja/pisanja za 4116/2118:

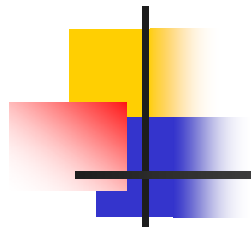
$\overline{RAS}$  (engl. Row Address Strobe)

~ impuls upisa adrese retka

$\overline{CAS}$  (engl. Column Address Strobe)

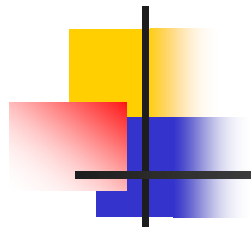
~ upis adrese stupca





# Dinamičke memorije

- *osvježavanje*
    - ~ adresiranje *samo* retka ili stupca:  
osvježavaju se *sve* ćelije retka/stupca
  - vrste osvježavanja:
    - distribuirano:
      - ~  $\forall$  2 ms/broj\_redaka (stupaca) ubaciti ciklus osvježavanja u sustavske cikluse
    - usnopljeno (engl. burst refreshing)
      - ~  $\forall$  2 ms osvježiti *sve* retke (stupce):  
ubaciti *odgovarajući broj* ciklusa osvježavanja
- npr. DRAM 16 K x 1 (4116/2118)  
16 K = 128 x 128  
 $\rightarrow 2 \text{ ms}/128 = 15,63 \mu\text{s} \rightarrow \forall 15 \text{ ms}$



# Dinamičke memorije

---

- izvedba osvježavanja:
  - posebna sklopovska podrška (MSI)
    - ~ MUX za adresu, s ugrađenim brojilom osvježavanja: male memorije
  - sklopovska podrška ugrađena u  $\mu$ P
    - ~ generiranje adrese osvježavanja  
*za vrijeme dekodiranja instrukcije:*  
npr. 8-bitni  $\mu$ P (npr. Zilog Z.80)
  - *pseudostatičke* memorije
    - ~ ugrađeno sklopovlje za osvježavanje:  
*transparentno za sustav!*

*Primjer:* memorija izvedena DRAM modulima 16 K x 1 (4116/2118)

- ubaciti *sklop za osvježavanje*

