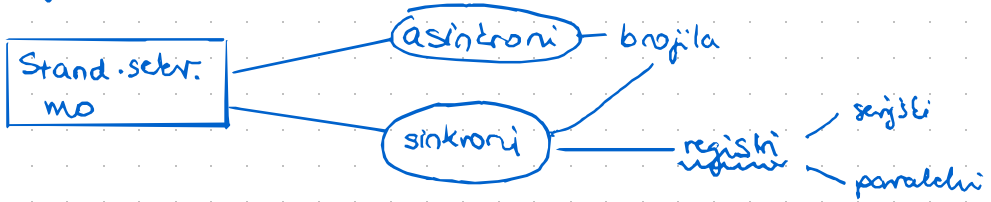


REGISTRI I BROJILA

Registar

- sebvencijski modul ostvoren kristalima

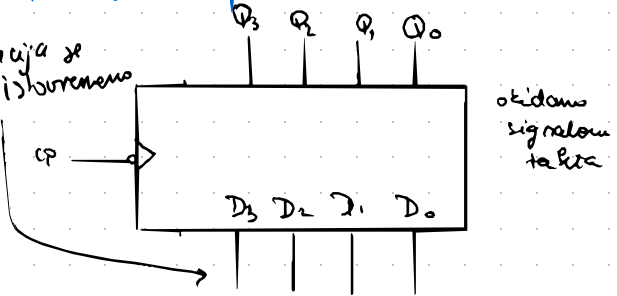


1) N-BITOVNI PARALELNI REGISTAR

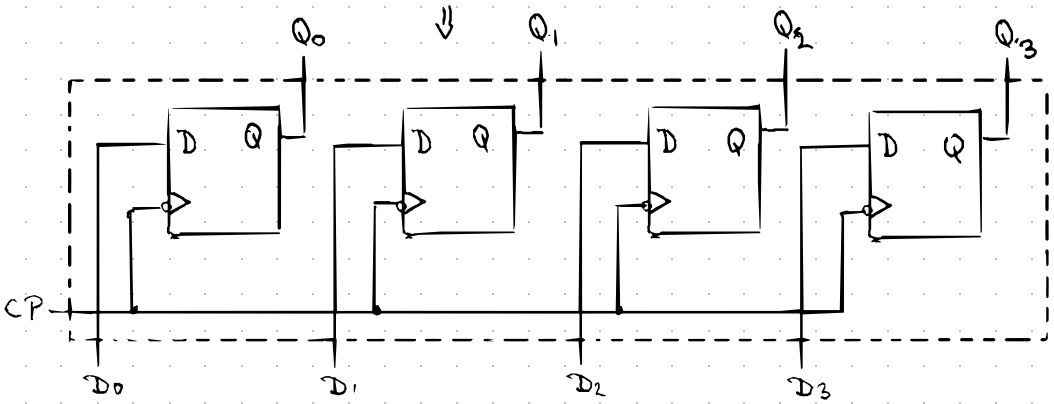
→ n bitova i pamti n-bitne binarne

⇒ skup povezanih kristala koji u gelini pamte određen broj bitova

- paralelni ulazi → informacija se upisuje istovremeno
- paralelni izlazi
- koliko kristala koliko bita ima riječ koju treba - registrirati, spremih



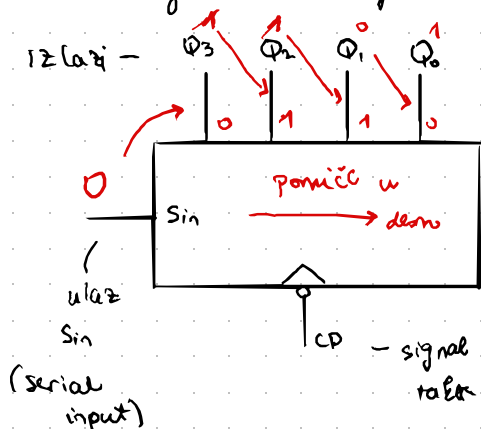
⇒ treba nam još ovaj kristal koji pamti ⇒ D (data)



2) N-BITOVNI POSMAČNI REGISTAR

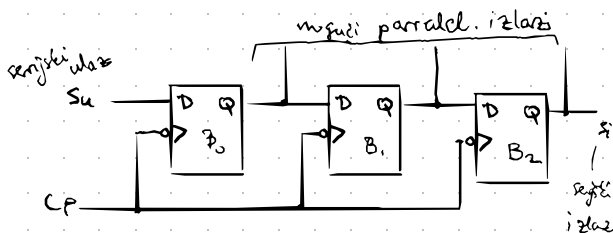
- Sastoji se od niza bistabilnih i određene kombinaijske logike
- D bistabilni
- razlika u načinu povezivanja

→ uobičajeno ima serijski ulaz i paralelne izlaze



- da nema serial inputa Q_3 bi ostao prazan

- podaci se serijski pomiču iz bistabila u bistabil

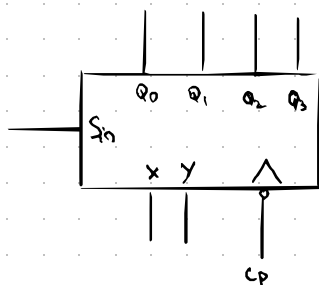


BITNO - da u bistabilu postoji dovoljno veliko kašnjenje

→ na sledeći bistabil djeluje samo izlaz iz prethodnog bistabila (dok je on pod utjecajem CP-a)

↓
JER AKO NE - sustav ne bi funkcionirao (nema serijskog povezivanja)

ZADATAK: Nacrtati implementaciju 4-bit registra čiji je simbol:



pri čemu onoliko ulazima x y register

obavlja sljedeće

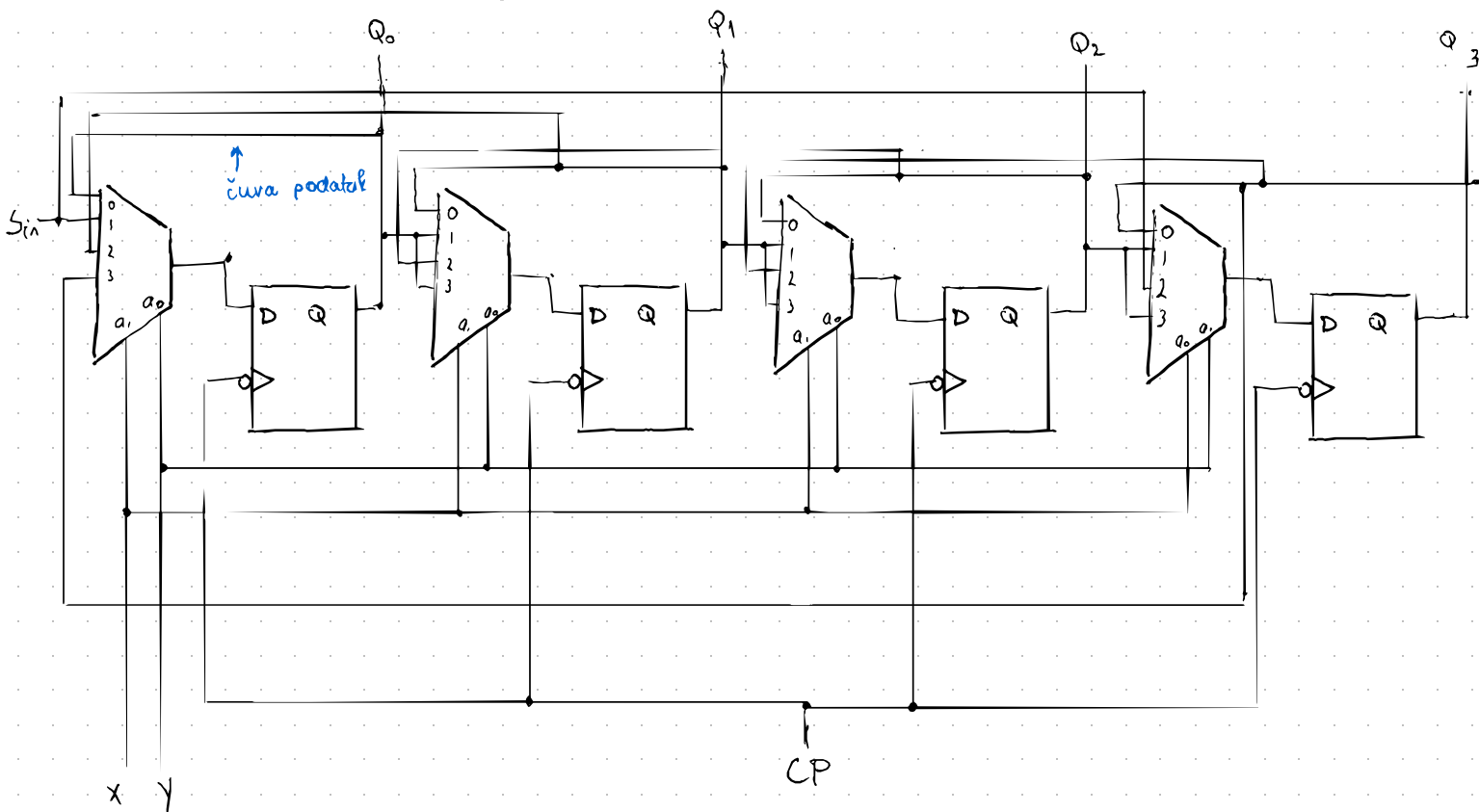
operacije:

x	y	operacija
0	0	čuva upisani pod.
0	1	pomakni udesno
1	0	pomakni ulijevo
1	1	rotacija udesno

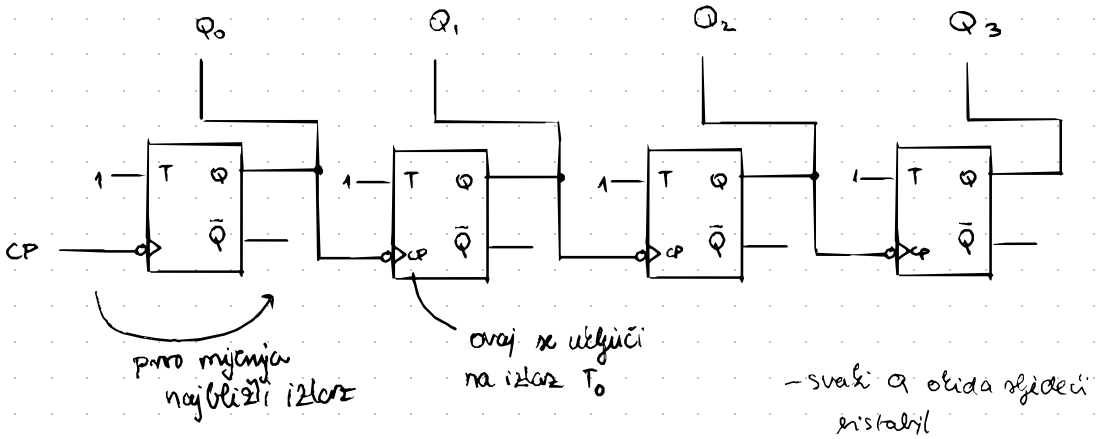
Treba 4 bistabila

→ za svaki bistabil 1 mux da "struklun" di ide

→ Sa 2 ulaza

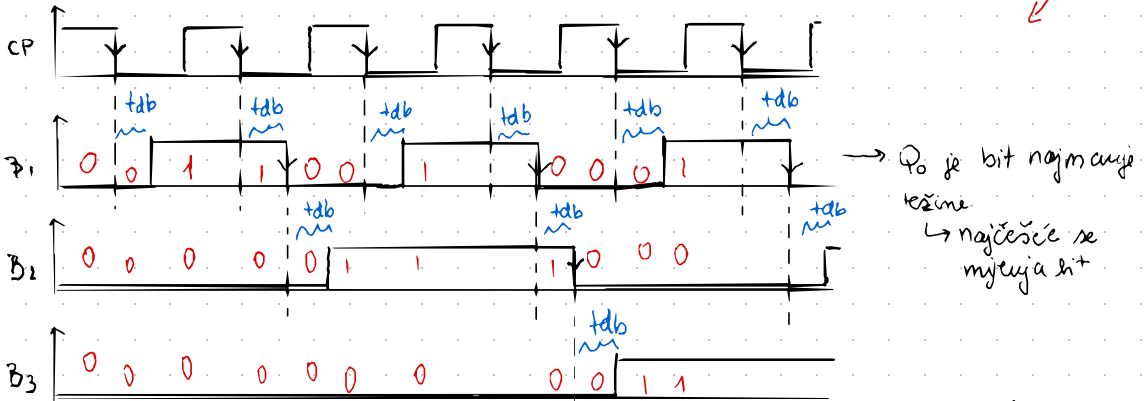
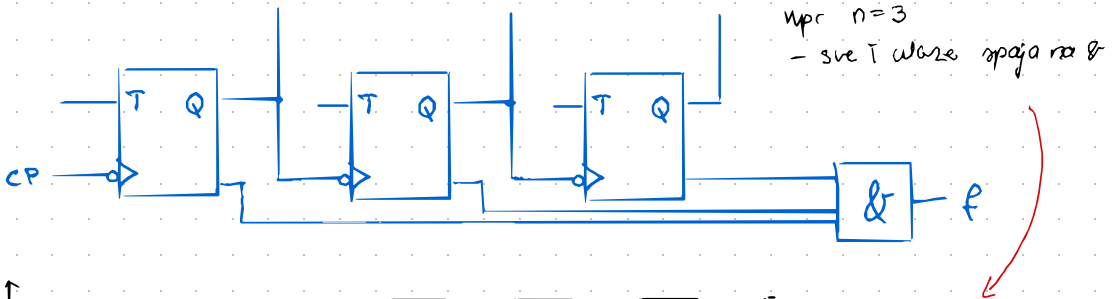


Asinkrona binarna brojila



POSLEDICA OVAKVOG POVEZIVANJA:

→ listaois Tcc na svaki padajuci broj signala tabeta promijeniti stavu



0 → 1 → (0) → 2 → 3 → (2) → (0) → 4 → 5. ↑ ide do 7 jer je III

() stabilna → stanja → L L također generira izlaz - POGRESKE TRANZICIJNE DEKODIRANJA

→ trošak selop koji u vrijednosti 000 stavlja u 1

→ NAND $\bar{Q} \cdot \bar{Q} \cdot \bar{Q} = 1$

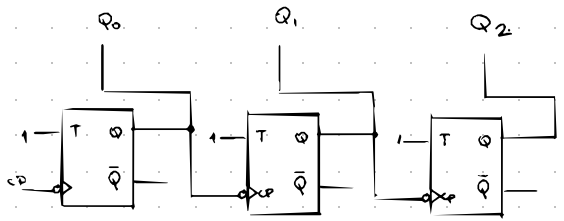
Kaskadni evalgov step

- brojilo od 4 bita

- kaskadni pojedini bita

je 1s → najgori slučaj: moramo čekat sve prethodne bita da promijene stanje

$$T_d = 4 \cdot t_{db} = 4s$$



OPĆENITO za asinkrono n-bit brojilo

$$T_d = n \cdot t_{db}$$

Maksimalna brzina kojemu evalgov step može raditi

→ koliko brzo smijemo postavljati novi padajući brd tako da pritom ne izgubimo prethodno stanje broja

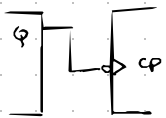
↳ vrijeme očitavanja

- vrijeme potrebno da pročitamo dekodirano stanje izlaza

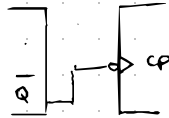
— najgori slučaj: treba pričekati vrijeme kaskadiranja broja i vrijeme očitavanja

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

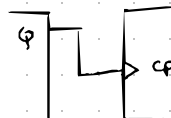
- postoje 4 mogućnosti spajanja Q/Q-bar x ↑/↓



Brojilo unaprijed



unaprijed

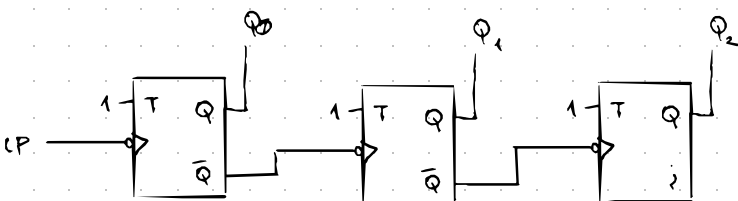


Brojila unazad



* ako promijenimo i jedno i drugo kao da se ponovi pa ostane u prvom unaprijed

- 3-bitno asinkrono binarno brojilo unatrag



- asinkrono binarna brojila izgrađeno od n listabla
 prirodno broje u ciklusu dužine 2^n stanja npr. za
 $n=3$ brojilo broji u ciklusu dužine 8

pitanje: kako realizirati ABB koji broji u ciklusu
 dužine $< 2^n$?

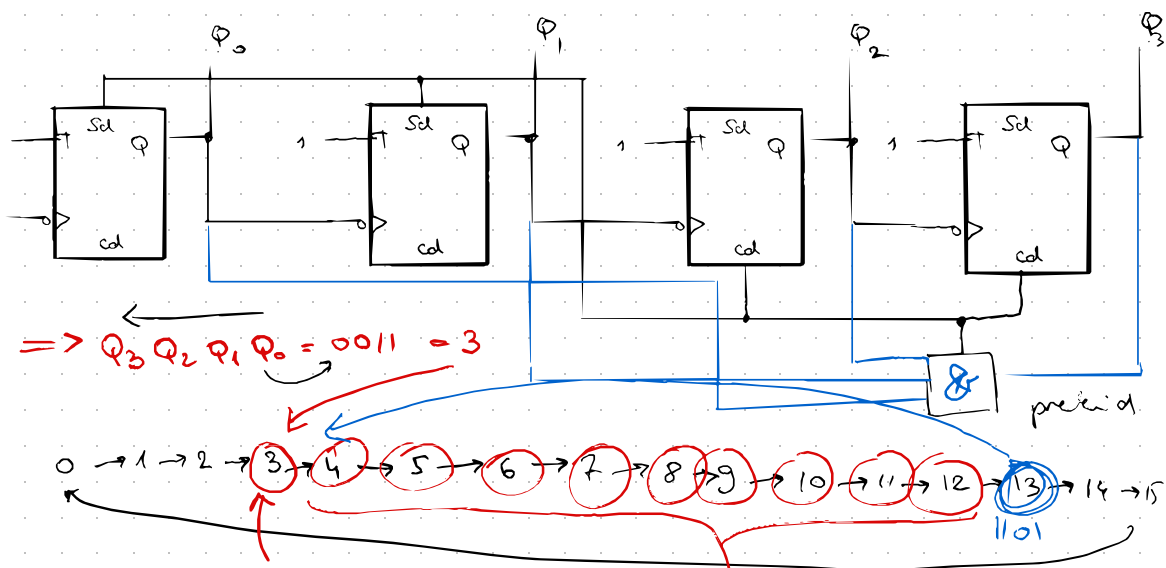
→ trebaju dodatne asinkrone ulaze za brisanje/
 postavljanje na listabilima T

⇒ generalna ideja: trebamo kombinacijski sklop koji će
 generirati signal za prekid ciklusa

↳ njega ćemo razvesti po nekoj selekciji
 asinkronih ulaza za postavljanje/brisanje

PRIMER: želimo dekadsko ABB (dužina ciklusa je 10)

- minimalni $n=4$ (i njega biram)



kada se dogodi prekid ciklusa
 ode u stanje 3

želimo da
 radi u onih
 10 stanja

* kada kliknemo prekid,
 ono što je spojeno na cd se cleara (Q_2 i Q_3), a
 Sd postaje 1 (Q_0 i Q_1)

* ne znam zašto 8 promijeniti to

sljedeće pitanje - koliko brzo možemo okidati CP

Da bismo odredili f_{max} , moramo odrediti koji je najgori slučaj:

- ① u dijelu ćelusa koji je "prirodan"
- ② u slučaju uređaja ćelusa

čitanje
↑

→ ① moramo čekati $4 \times t_{db} + t_{oc}$ (npr. 7 → 8)

② moramo čekati $1 \times t_{db} + t_{des} + 1 \times t_{db} + t_{oc}$
 $12 \rightarrow 13$

za slop &
koji generira
prekid

da se
ni bistabilu
stiraju/resetiraju

$$T_{cp} = \max(①, ②)$$

$$= \max(4 t_{db} + t_{oc}, 2 t_{db} + t_{des} + t_{oc})$$

$$f_{max} = \frac{1}{T_{cp}}$$

Sinkrono n-bitovno brojilo

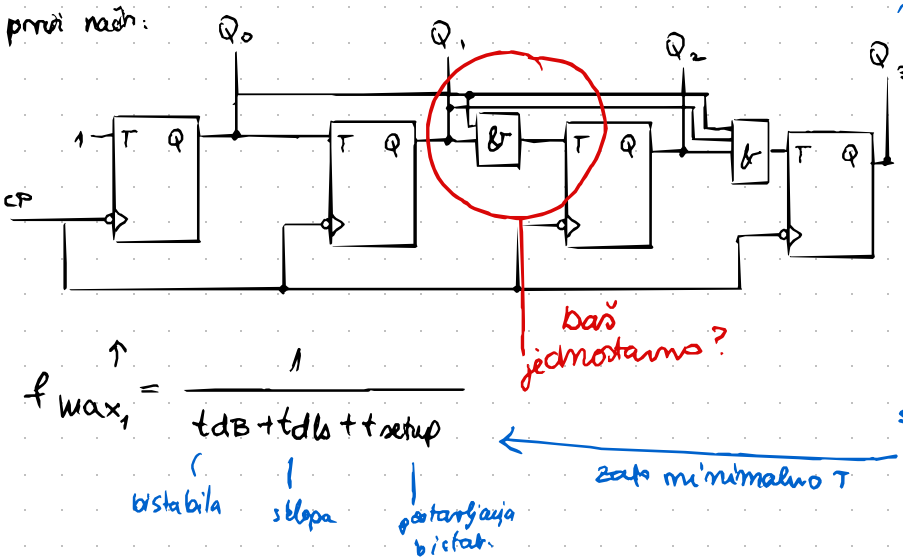
- svi CP spojeni zajedno
- endito brojača koliko i bitova

► T prvog bistabla je fiksna jedinica (1)

↳ T svih ostalih brojača je log & izlaza prethodnih bistabla

Praktični primjer: $n=4$

prvi način:



PARALELNO

- svi & sklopovi spojeni paralelno upravljaju varijable

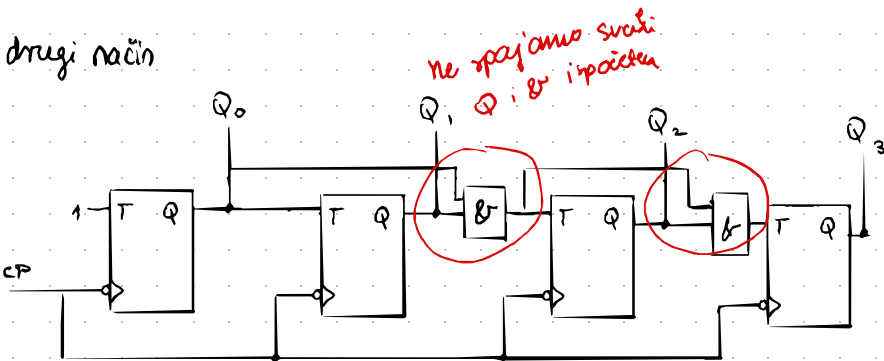
↓
pri svakoj promjeni mijenja se ulaz samo jednog sklopa

$$f_{max,1} = \frac{1}{t_{dB} + t_{dl} + t_{setup}}$$

(bistabila) (sklopa) (potraživanja bistaba)

zato minimalno T

drugi način



SERIJSKI

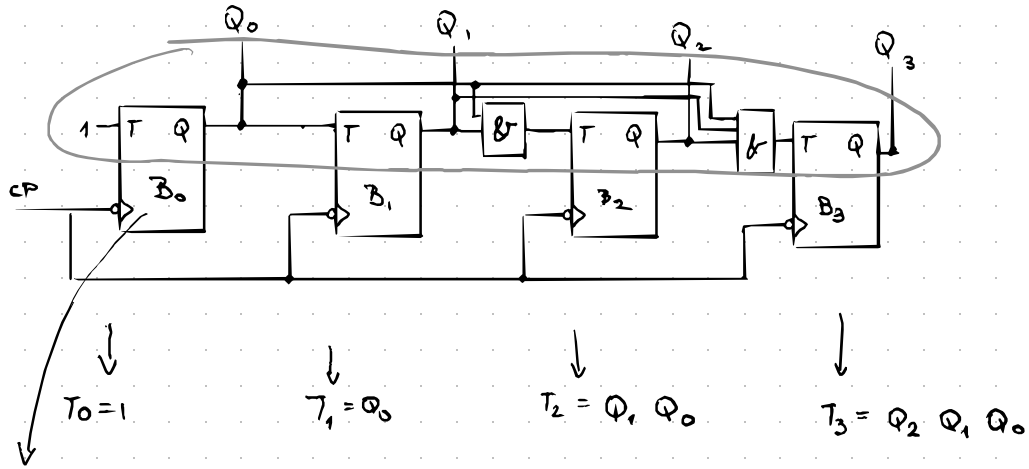
- signal prolazi kroz $n=2$ log sklopa; novi ulaz ne smije doći prije nego što zadnji bistabil ima ispravne log vrijednosti na ulazu

$f_{max,2}$ je različit zbog različitog spajanja &

$$f_{max} = \frac{1}{t_{dB} + (n-2)t_{dl} + t_{setup}}$$

↳ u prvom i drugom bistabilu nije potreban logički sklop &

PROMJENA STANJA BISTABILA



* B_0 mijenja stanje svakim dolaznim impulsom

B_i - 11- na svaku promjenu...

OPĆENITO :

$T_0 = 1$ → $T_i = B_0 B_1 \dots B_{i-1}$
 osnovni korak korak rekursije

unaprijed

-svaki sljedeći T jednak je log & n-ih prethodnih Q

unazad

-svaki sljedeći T jednak je log & n-ih prethodnih Q

• dvosmjerno brojeb

$T = 1$

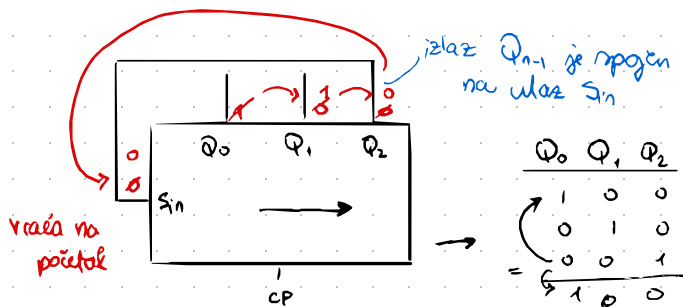
$\left\{ \begin{array}{l} x=0 \text{ unaprijed} \\ x=1 \text{ unazad} \end{array} \right.$

$T_1 = (Q_0 \oplus x)$

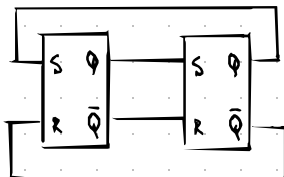
$T_2 = (Q_1 \oplus x)(Q_0 \oplus x) \dots$

Brojila temeljena na posmaćnim registrima

Prstenasto brojilo



log. implementacija

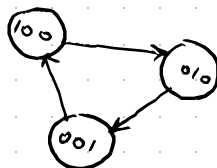


posmaćni registar koji rotira zapisani podatak

- po uključanju na napajanje ne možemo dopustiti da se bistabilni namučno inicijaliziraju

► najmanja važnost u 1, ostali u 0

• duljina ciklusa iznosi n za n bitovni registar



Johnsonovo brojilo (broji sa ukrštenim prstenom)

- isto kao u prstenastom,

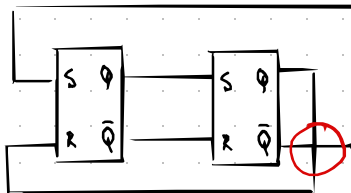
! ali vraćamo komplement Q_{n-1} bita

- dvostruko efikasnije od prstenastog

↳ ciklus je duljine

$2 \times n$

Q_0	Q_1	Q_2
0	0	0
1	0	0
1	1	0
1	1	1
0	1	1
0	0	1
0	0	0



→ nema lag sklopova, nema serijski/paralel. uključivanja

→ max. frekv. rada — svi bistabilni u istoj situaciji