SEKVENCIJSKI SKLOPOVI

* kombinacijsko sklopovlje → radi kao funkcija (preslikava)
  (koder, dekoder...)
  – ulazne xy kombinacije uvijek preslikava u iste vrijednost

Sekvencijsko sklopovlje

- za određeni ulaz u različitim vremenskim trenutcima daje različit izlaz

↳ nije isključivo funkcija trenutnog ulaza (pobude)
  · ovisi i o unutarnjem stanju

Primjer: Pokušajmo realizirati sklop koji:

· na ulaz dobiva jedno po jedno slovo iz skupa $\{A,B\}$
- ima jedan izlaz - na njega ide simbol D/N (da/Ne)
  ↳ D ako na ulazu ide niz slova A duljine barem 2
  (i ostaje D dok dolaze slova A

ulaz	izlaz
B	N
A	N
B	N
A	N
A	D

Ograničenja:

• projektiramo „crnu kutiju"
• sklop može primiti po samo jedan od simbola
· izlaz sklopa funkcija je zapamćenog simbola (definiramo tablicom)
· nakon svakog čitanja slova na ulazu možemo na temelju pročitanog slova i zapamćenog simbola odrediti koji se simbol dalje pamti

ulaz = $\{A,B\}$
izlaz = $\{D,N\}$

ulaz $\{A,B\}$ => memorija => izlaz $\{D,N\}$

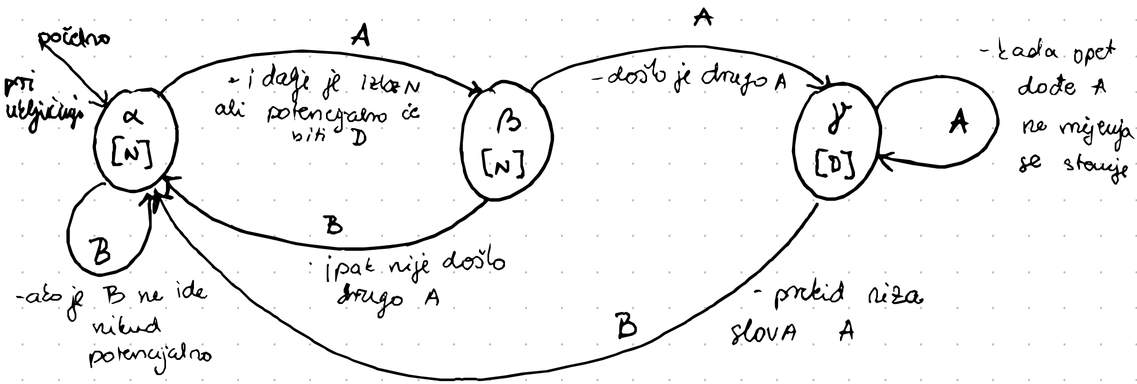
*po uključenju izlaz je na N

$$N \xrightarrow{B} N \xrightarrow{B} N \xrightarrow{A} N \xrightarrow{B} N \underbrace{\xrightarrow{A} N \xrightarrow{A}}_{\text{jer su dva uzastopna}} D$$

pobuda

→ Stanje memorije = simbol koji se pamti u memoriji
(stanje sustava)

Primjer 2.) U memoriji pamtimo slova grčkog alfabeta
→ nastavak na prošli



α - situacija kada nema slova A

β - vidi prvo slovo A, možda će doći do promjene u D

γ - započeo je niz slova A najmanje duljine A

⇓

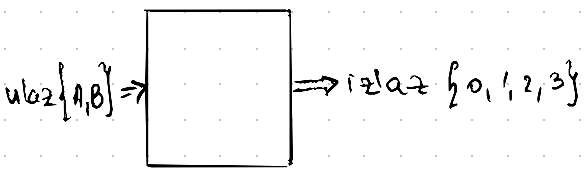
Stanje	izlaz
α	N
β	N
γ	D

=>

Stanje	ulaz	sljedeće stanje
α	A	β
α	B	α
β	A	γ
β	B	α
γ	A	γ
γ	B	α

ZAD 2)

Konstruirati automat koji na ulaz dobiva slovo iz $\{A, B\}$. Automat na izlaz postavlja binarno zapisanu vrijednost $N \% 4$ gdje N odgovara ukupno dotada viđenih slova A.

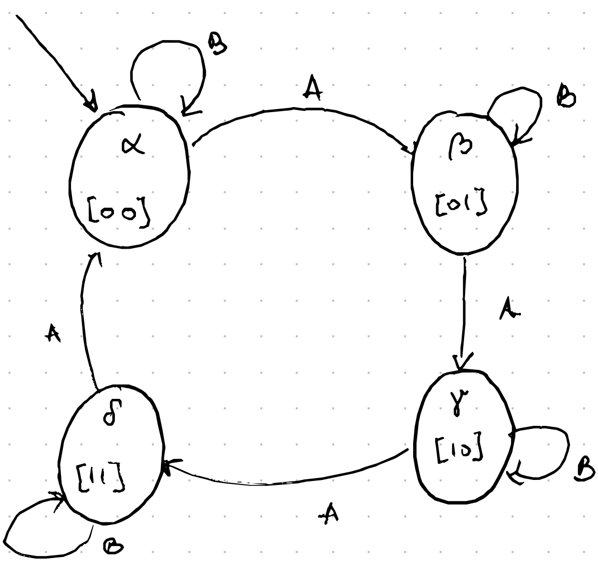


po uključenju: $0 \xrightarrow{B} 0 \xrightarrow{B} 0 \xrightarrow{A} 1 \xrightarrow{B} 1 \xrightarrow{B} 1 \xrightarrow{A} 2 \xrightarrow{B} 2$
$\xrightarrow{A} 3 \xrightarrow{A} 0 \xrightarrow{A} 1 \xrightarrow{B} 1$

- trebat ćemo 4 stanja koja redom reprezentiraju svaki od ostataka dijeljenja N sa 4: $\alpha, \beta, \gamma, \delta$

$\alpha = N \% 4 = 0$ $\qquad$ $\gamma = N \% 4 = 2$

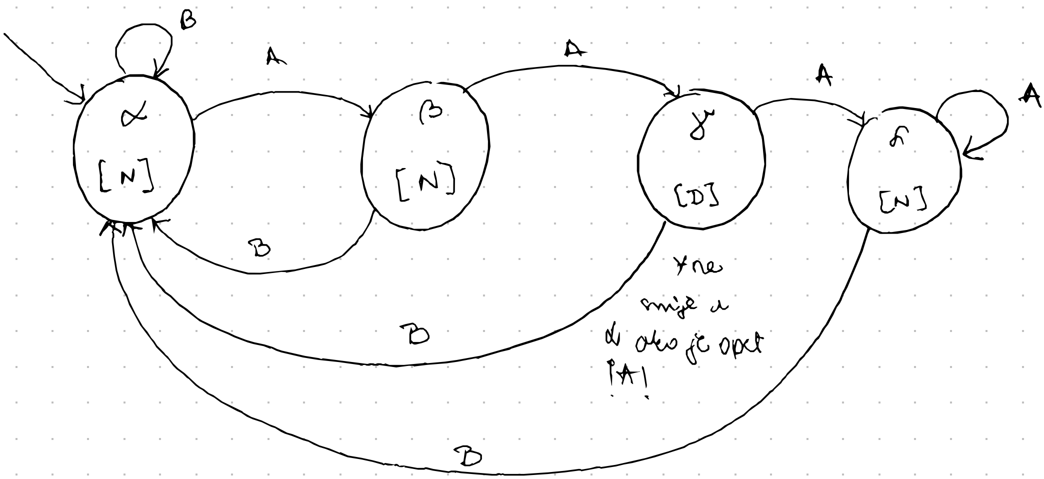
$\beta = N \% 4 = 1$ $\qquad$ $\delta = N \% 4 = 3$



ZAD 3.)

Slično kao <u>1. zad</u>, uz razliku da izlaz postaje D samo u trenutku kad je detektirana sekvenca od 2A, a za sva sljedeća slova A ponovno je N

$N \xrightarrow{B} N \xrightarrow{A} N \xrightarrow{A} D \xrightarrow{A} N \xrightarrow{A} N \xrightarrow{A} N \cdots \xrightarrow{B} N \xrightarrow{A} N \xrightarrow{A} D$



Model sekvencijskog sklopa koji smo upravo koristili:

- sljedeće stanje je funkcija od trenutnog stanja i pobude (ulazi)
- izlaz je f-ja TRENUTNOG STANJA

Zove se <u>Moorov automat</u> ili <u>Moorov stroj s konačnim brojem stanja</u>

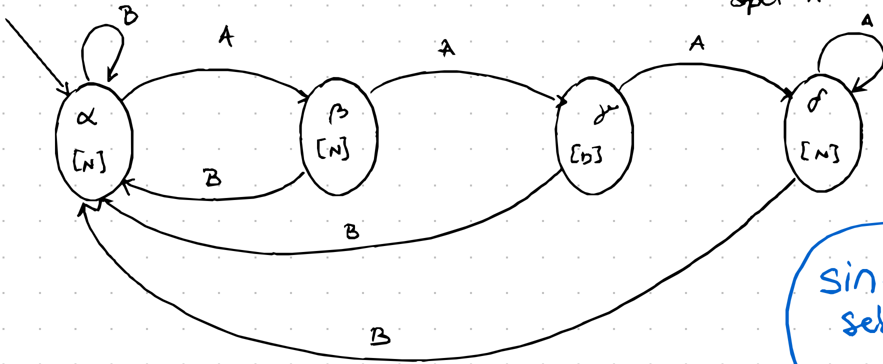
Kod Mealyjevog automata, i sljedeće stanje i izlaz su funkcija trenutnog stanja i trenutne pobude.

→ Moorov automat je stroj stanja

→ Mealyjev je stroj prijelaza

Mooreov automat

* ne smije u α ako je opet A



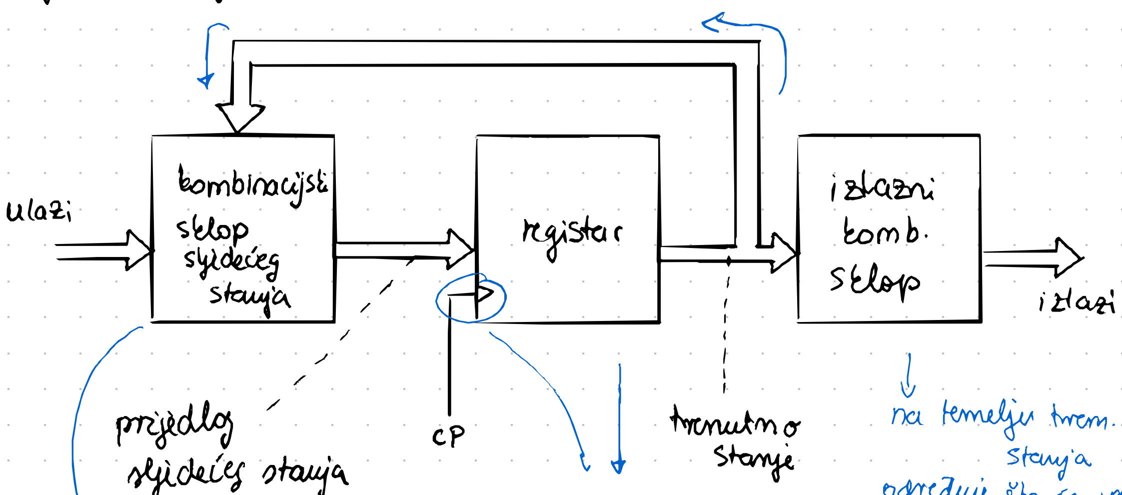
sinkroni sekvencijski sklop

- stroj s konačnim brojem stanja

• sklop sadrži internu memoriju u kojoj može pamtiti jedan simbol (trenutno stanje)

• izlaz je funkcija trenutnog stanja

• sljedeće stanje: FUNKCIJA TRENUTNOG STANJA I TRENUTNE POBUDE

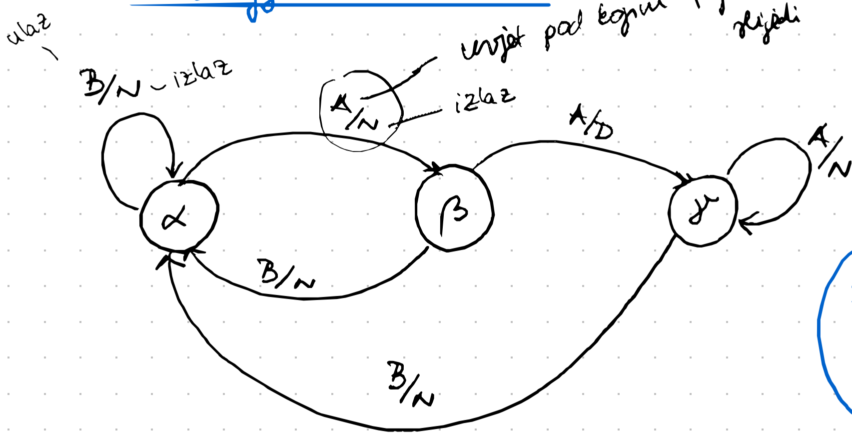


na temelju trem. stanja određuje što će van

na temelju trenutnog stanja i trenutne pobude priprema prijedlog sljedećeg stanja → tj se prijedlog usvaja na na rastući brid signala takta

⇒ postaje novo stanje

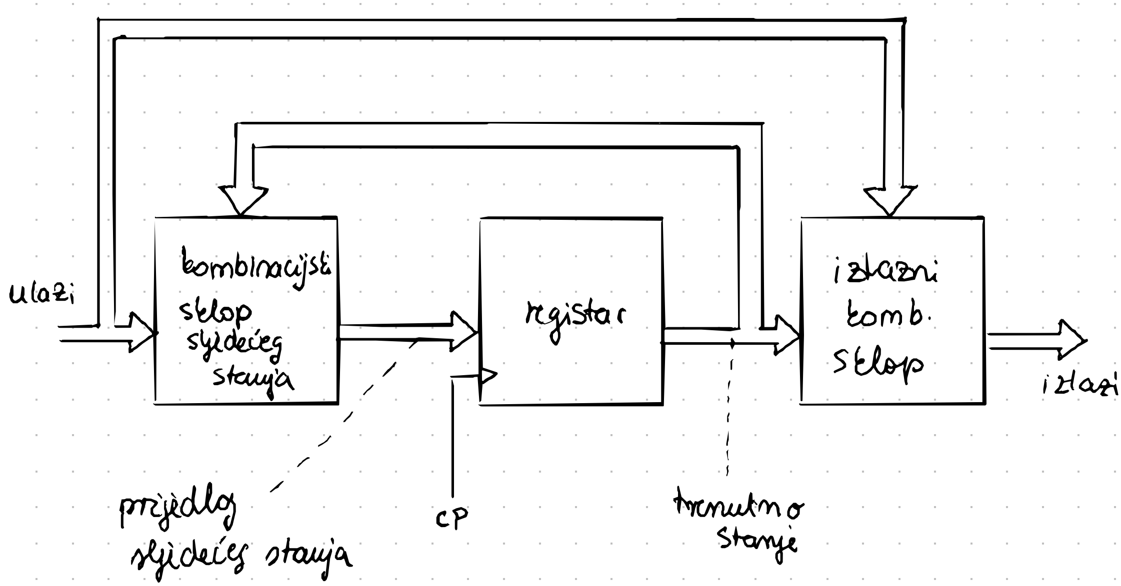
Mealyjev automat



sinkroni sekvencijski sklop

- ako izlazi osim o stanjima bistabila ovise još o unutarnjim stanjima

→ ne veže se izravno za stanje već uz prijelaz



Projektiranje Mooreovog automata

Korak 1.)

· sve simbole potrebno je prikladno kodirati
→ ulaze
→ simbole stanja
→ izlaze

Korak 2.)

· Napraviti tablicu koja za sva moguća stanja i ulaze određuje što je sljedeće stanje i izlaz

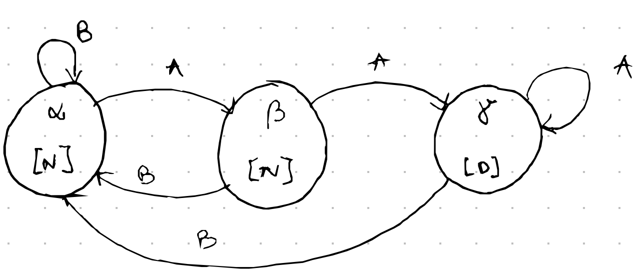
Korak 3.)

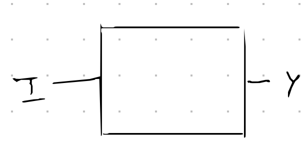
· Odabrati bistabile kojima će se ostvariti „registar", proširiti tablicu iz prethodnog koraka ulazima tih bistabila i popuniti te stupce tako da se ostvare prethodno definirane promjene stanja

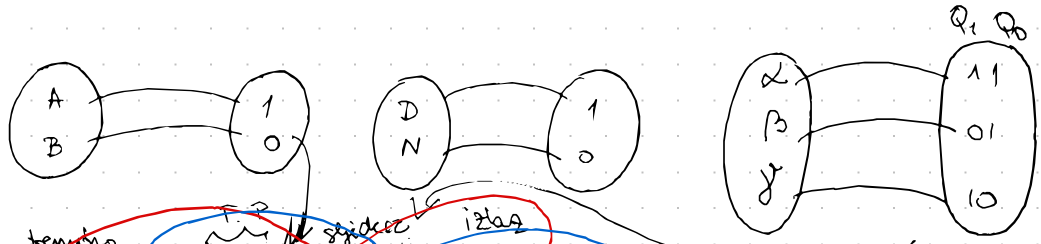
Korak 4.)

· minimizirati Booleove funkcije ulaza bistabila te izlaze automata

Radni primjer iz zad (1)







trenutno stanje / sljedeće stanje / izlaz

	Q_1	Q_0	Y			Q_1	Q_0		Y	D_1	$J_0 K_0$
	0	0	0			X	X		X	X	XX
	0	0	1			X	X		X	X	XX
β	0	1	0	B	α	1	1	N	1	1	X0
	0	1	1	A	γ	1	0	N	1	1	X1
γ	1	0	0	B	α	1	1	D	0	1	1X
	1	0	1	A	γ	1	0	D	0	1	0X
α	1	1	0	B	α	1	1	N	1	0	X0
	1	1	1	A	β	0	1	N	1	0	X0

*Moore grede

trenutna stanja stoga se ovaj izlaz ravna prema $Q_1 Q_0$ iz trenutnog

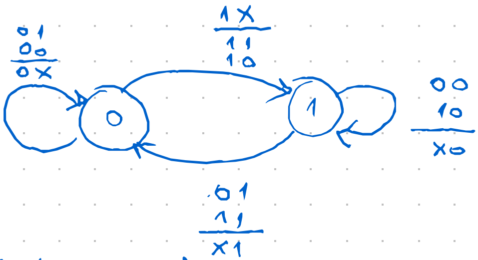
$$y(Q_1, Q_0, I) = \sum (2,3,6,7) + \sum d(0,1)$$

* triba nestat I kada bismo to raspisali

*za Mealyjev automat koristi istu tablicu, ali neće se nužno potratit I_0

B_1 ..... D-bistabil

B_2 ..... JK-bistabil



$$J_0(Q_1, Q_0, I) = \sum m(4) + \sum d(0,1,2,3,6,7)$$

$$K_0[Q_1, Q_0, I) = \sum m(3) + \sum d(0,1,4,5)$$

$$D_1(Q_1, Q_0, I) = \sum m(2,3,4,5,6) + \sum d(0,1)$$

$Q_0 Q_1$, I

X	1	1	
X	1	1	

$Y = Q_0$

$Q_0 Q_1$, I, Q_0, Q_1

X	X	X	1
X	X	X	

$J_0 = \bar{I}$

$Q_0 Q_1$, I

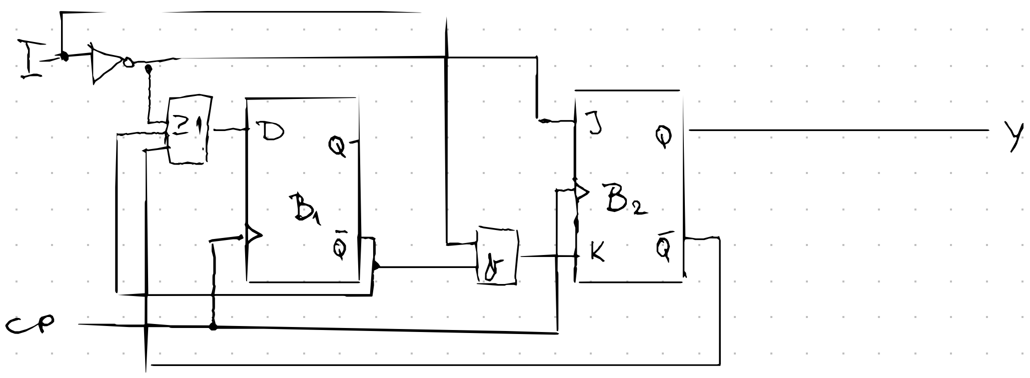
X			X
X	1		X

$K_0 = \bar{Q}_1 \cdot I$

$Q_0 Q_1$, I

I \ Q_0Q_1	00	01	11	10
0	X	1	1	1
1	X	1		1

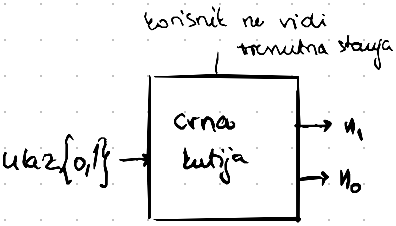
$D_1 = \bar{I} + \bar{Q}_0 + \bar{Q}_1$

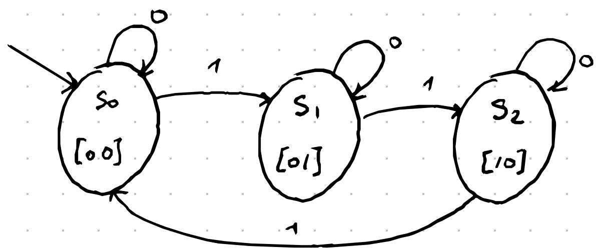


Ekvivalentna stanja i minimizacija broja stanja automata

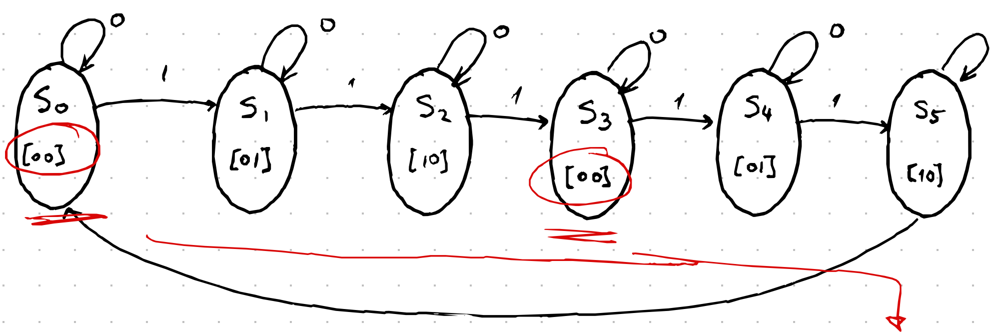
Radni primjer:

Automat koji na ulaz dobiva {0,1}, na izlazu u binarnom zapisu generira broj tada viđenih (modulo 3 → % 3)





→ možemo proširiti



Pojam ekvivalentna stanja – oni S za koje su izlazi isti za sve moguće pobude da se ponašaju identično

- dva stanja S_i i S_j su ekvivalentna <u>AKKO</u>
  - ↳ su identična ($i = j$)
    - ako imaju identične izlaze te za sve moguće slijedove ulaza generiraju identične izlaze

ili

↳ ako imaju jednake izlaze te za sve pobude idu u isto ili ekvivalentna stanja

Reduciranje broja stanja

pronalaženje i eliminiranje ekvivalentnih stanja

→ VDŽ. Dig. Sustavi U. Peruško, V. Glavinić, str. 364.

Sadašnje stanje	Sljedeće stanje x = 0	Sljedeće stanje x = 1	Izlaz x = 0	Izlaz x = 1
A	C	D	1	0
~~B~~	~~C~~	~~D~~	~~1~~	~~0~~
C	~~B~~ A	D	0	1
D	F	~~E~~ D	0	1
~~E~~	~~F~~	~~D~~	~~0~~	~~1~~
F	~~B~~ A	C	1	1

→ ovisno o pobudi

– dva stanja se međusobno uspoređuju

• izlaz z im je jednak za sve vrijednosti ulazne varijable x
+ oba prijelaze u ista stanja C i D

⟹ ta dva stanja su ekvivalentna i jedno možemo ELIMINIRATI

1) ↳ eliminacija stanja B

2) zamjena svih stanja B sa stanjem A u tablici (jer su ekvivalentna)

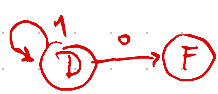
3) nastaviti uspoređivanje

↳ stanje A uspoređujemo s ostalim stanjima

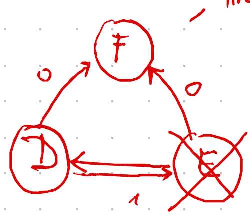
↳ zatim stanje C uspoređujemo ... zatim D

	x = 0	x = 1	Izlaz x = 0	Izlaz x = 1
D	F	~~E~~ D	0	1
~~E~~	~~F~~	~~D~~	~~0~~	~~1~~

idu jedno u drugo →



trebamo 1 za kad je 1 i 1 za kad je 0



↳ ne trebamo sva 3

Tablica prije reduciranja

Sada. stanje	Slijed. stanje X = 0	Slijed. stanje X = 1	Izlaz z x = 0	Izlaz z x = 1
A	C	D	1	0
B	C	D	1	0
C	B	D	0	1
D	F	E	0	1
E	F	D	0	1
F	B	C	1	1

$\Longrightarrow$

Tablica poslije reduciranja

Sada. stanje	Slijed. stanje X = 0	Slijed. stanje X = 1	Izlaz z x = 0	Izlaz z x = 1
A	C	D	1	0
C	A	D	0	1
D	F	D	0	1
F	A	C	1	1

- kod reduciranja broja stanja većih sustava, višestruko pronalaženje kroz tablicu stanja može dovesti do pogreške ili previda

Huffman Mealyjeva metoda reduciranja

1) podjela unutarnjih stanja u najmanji mogući broj klasa ekvivalentnih stanja
   → stanja u klasi imaju iste izlaze
   ⇒ grupirati s obzirom na izlaze

2) podjela na podklase – prijelazi iz stanja jedne klase vode u stanje druge klase

3) prijelazi između stanja zamjenjuju prijelazima između klasa stanja

Primjer.) sekv. sklop sa ulazom x i izlazom z, 8 stanja

	q^n	q^{n+1}, z^n ($x^n=0$)	q^{n+1}, z^n ($x^n=1$)
a	q_0	q_0, 1	q_4, 0
b	q_1	q_0, 0	q_4 0
b	q_2	q_1, 0	q_5, 0
b	q_3	q_1, 0	q_5, 0
c	q_4	q_2, 0	q_6, 1
c	q_5	q_2, 0	q_6, 1
c	q_6	q_3, 0	q_7, 1
c	q_7	q_3, 0	q_7, 1

* nije Mooreov automat jer se izlaz mijenja ovisno o ulazu → Mealy

→ boje – grupirano s obzirom na izlaze
   → a, b i c grupe

klasa	a	b			c			
Stanje	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7
Sl. klasa	a c	a c	b c	b c	b c	b c	b c	b c

x=0 → q_0 pripada a

x=1 → q_4 pripada c

q_1 nije kao ostala b stanja jer ima ac a ne bc
   → novu grupu napraviti (d)

klasa	a	b		c				d
Stanje	q_0	q_2	q_3	q_4	q_5	q_6	q_7	q_1
Sl. klasa	a c	b c	b c	b c	b c	b c	b c	a c

⇒ želimo da grupe budu homogene

→ da neko od stanja ima $q^{n+1} = q_1$ onda bi sad umjesto pripadnog slova klase trebali staviti d i ponovno homogenizirati tablicu

⇒ ovdje ne moramo dalje

Klasa	a	b		c				d
Stanje	q_0	q_2	q_3	q_4	q_5	q_6	q_7	q_1
Sl. klasa	a c	b c	b c	b c	b c	b c	b c	a c

- nakon zamjene slova grupe našli bismo duplikate
  ↳ ponovljene retke mičemo

=> dobili bismo reducirani automat

→ umjesto polaznih 8 redaka imamo 4

Stanja u klasi ekvivalencije	q^n	q^{n+1}, z^n	
		$x^n = 0$	$x^n = 1$
q_0	a	a , 1	c , 0
q_2, q_3	b	b , 0	c , 0
q_4, q_5, q_6, q_7	c	b , 0	c , 1
q_1	d	a , 0	c , 0

→ izlazi ostaju isti
↳ zato smo ih i grupirali po izlazima

→ umjesto 8 bistabila imamo 4

* u zadacima ne traže minimizaciju memorije, ali pitat će koja su stanja međusobno ekvivalentna

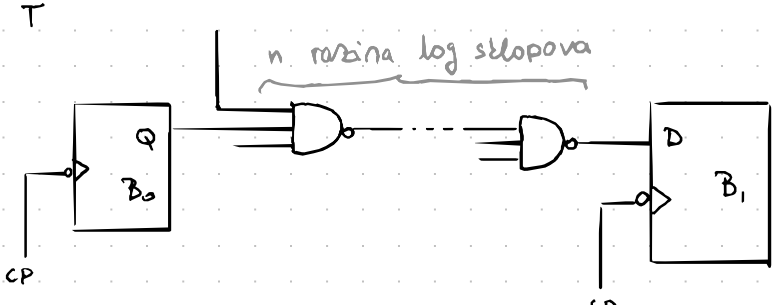
VREMENSKI ODNOSI

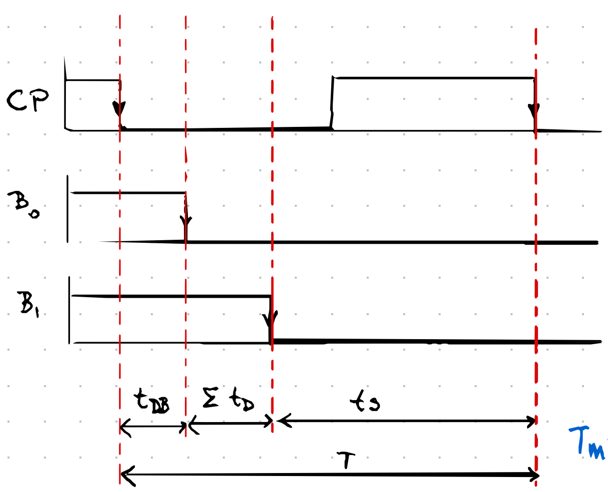
Maksimalna frekvencija rada (takta) - najveća frekvencija CP pri kojoj sklop (svi njegovi bistabili) sigurno mijenjaju stanje kad to ulazi zahtijevaju

→ želimo da je što viša f_{max} (da sklop radi brzo)
  ↳ veći broj operacija u sekundi

→ Problem = između susjednih bistabila (i kod samih bistabila) postoji nekakvo kašnjenje signala

$f = \frac{1}{T}$ → da bi f bio max, T mora biti min





t_{DB} - Bistabil delay

t_D - kašnjenje jednog sklopa
  ↳ Σt_d - n kašnjenja log. sklopova

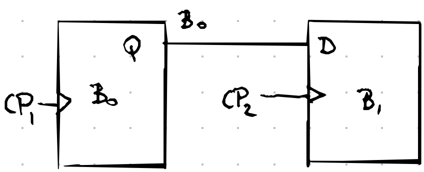
t_s - set time (postavljanja)

$$T_{min} = (t_{DB})_{max} + (\Sigma t_D)_{max} + (t_s)_{max}$$

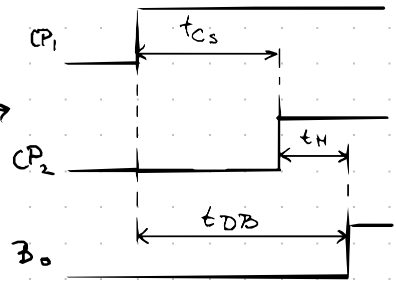
Rastorak

- svi bistabili se ne pobuđuju odmah iz jednog izvora impulsa takta
- između dva CP impulsa (koji su nominalno jednaki) dođe do vremenskog rastoraka takta

$t_{dB} = t_{CS} + t_H$



→



↳ najgora moguća situacija

- B_0 je pod utjecajem CP_1, a B_1 je pod utjecajem CP_2 ⟹ između CP_1 i CP_2 postoji vremenski rastorak t_{CS}

— da se osigura normalan rad sklopa:

↳ B_1 mora reagirati na CP_2 pod utjecajem stanja B_0 (Q) koje on ima prije promjene koja nastupa pod utjecajem CP_1

↳ CP_2 mora doći prije tog najmanjeg vremena za iznos vremena zadržavanja t_H (hold)

⟹ treba osigurati da maksimalan rastorak ne bude veći od vrijednosti $(t_{CS})_{max} = (t_{dB})_{max} - (t_H)_{max}$