DIGITALNA I MIKROPROCESORSKA TEHNIKA

1. UVOD
2. SINTEZA KOMBINACIJSKIH LOGIČKIH STRUKTURA
3. SINTEZA SEKVENCIJALNIH SKLOPOVA
4. OSNOVE ARHITEKTURE MIKRORAČUNALA

3. SINTEZA SEKVENCIJALNIH SKLOPOVA

- 3.1. SEKVENCIJALNI SKLOPOVI
- 3.2. ELEMENTARNI SKLOPOVI ZA PAMĆENJE - BISTABILI
- 3.3. SLOŽENE STRUKTURE S BISTABILIMA
- 3.4. DIGITALNI AUTOMATI I SINTEZA AUTOMATA
- 3.5. PROGRAMABILNI AUTOMATI I ALGORITMI

3.1. SEKVENCIJALNI SKLOPOVI

- KOMBINACIJSKI SKLOPOVI (kombinacijske logičke strukture, KLS)

  - ovise samo o trenutnom ulazu

  - ne pamte prethodne događaje

  - nemaju memoriju, nemaju stanja (stateless)

- SEKVENCIJALNI SKLOPOVI (bistabili, automati, algoritmi)

  - ovise o sadašnjem i prošlim ulazima - događajima

  - pamte prethodne događaje

  - imaju memoriju i stanja (statefull)

KAŠNJENJE I PAMĆENJE

- KAŠNJENJE

  zadržavanje informacije u vremenu (nepoželjno)

- PAMĆENJE

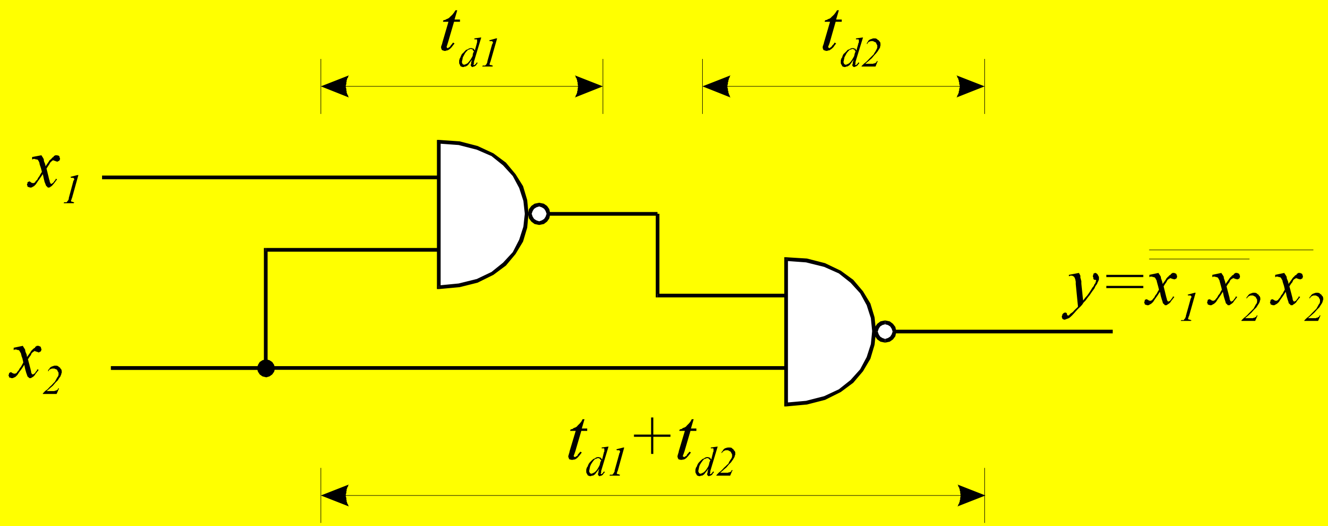
  zadržavanje informacije u vremenu (poželjno)

⇒KAŠNJENJE = PAMĆENJE

ostvaruje se promjenom i zadržavanjem strukture materije i/ili energije

DISKRETNO VRIJEME

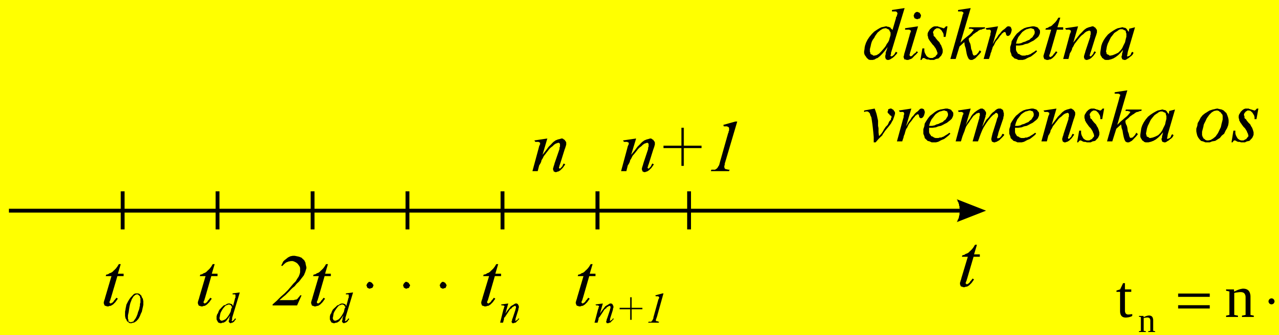
- Promatrajmo sklop:



vidimo da ulazi djeluju na izlaz u trenucima određenim kašnjenjem na logičkim vratima

DISKRETNO VRIJEME

- Definirajmo diskretno vrijeme:

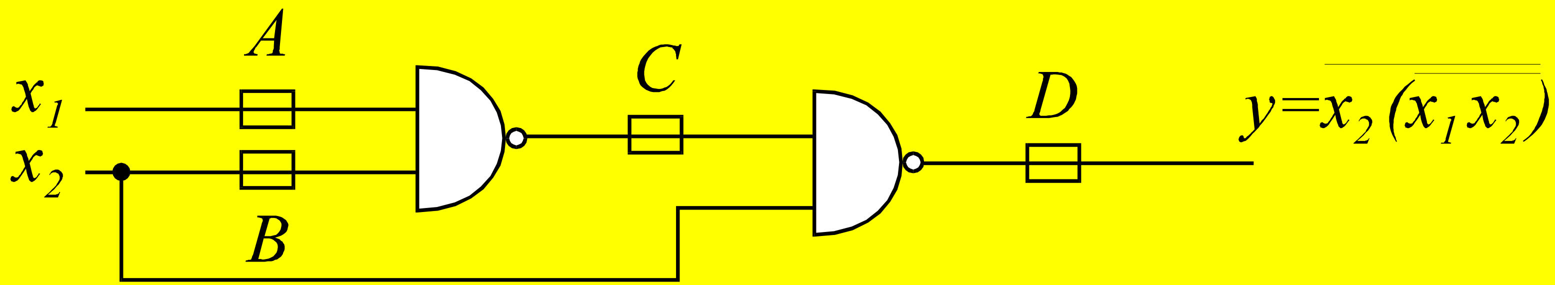


$$t_n = n \cdot t_d$$

- promjene se dešavaju u trenucima t
- unutar perioda nema promjena
- t_n = sadašnji trenutak, t_{n+1} slijedeći trenutak
- n = sadašnji period, n+1 slijedeći period

POSLJEDICE KAŠNJENJA

- Pogledajmo ponašanje sklopa:



- mjerimo signale u mjernim točkama A, B, C i D
- u trenutku t_n kodna riječ ABCD čini **stanje** sklopa
- u nizu trenutaka sklop prolazi kroz **niz stanja**
- neka stanja su **stabilna**, a neka **nestabilna**

POSLJEDICE KAŠNJENJA

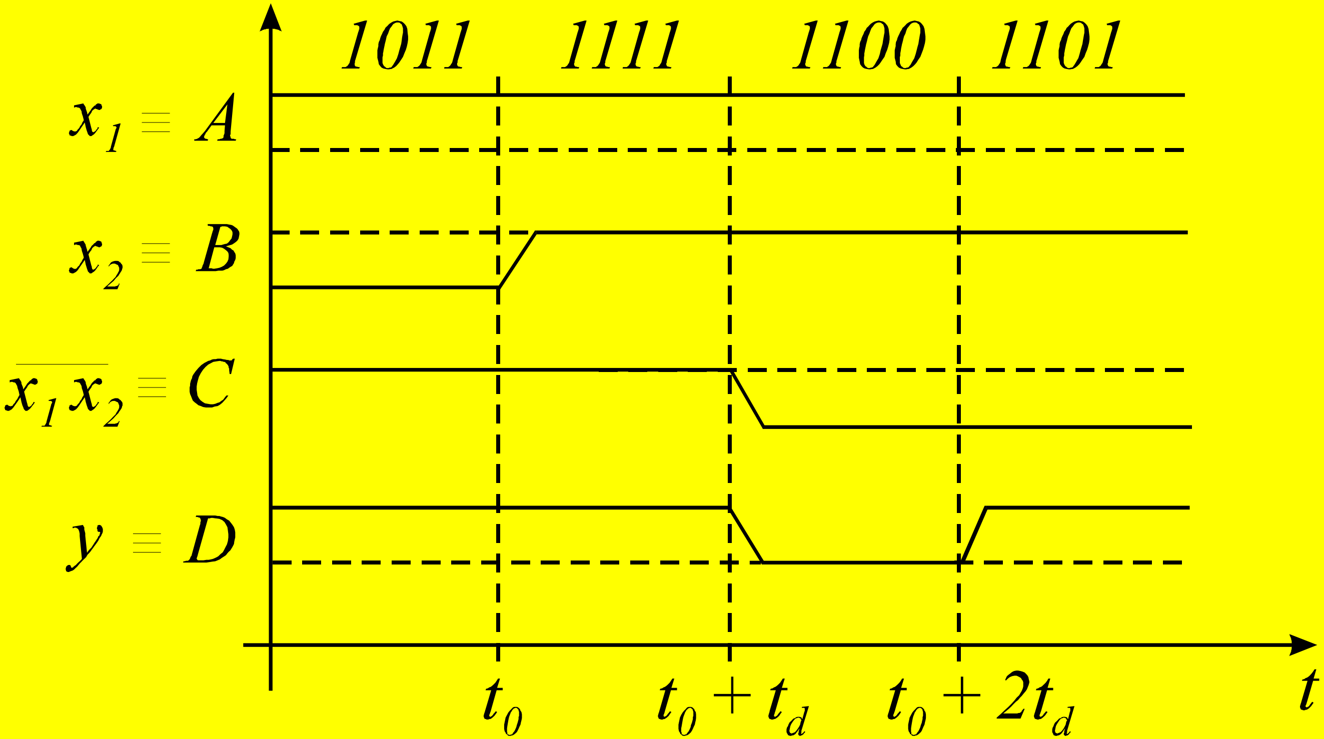
- Za neki konkretni vremenski niz na ulazu:

x_1	x_2	A	B	C	D	
1	0	1	0	1	1	
1	1	1	1	1	1	t_0
1	1	1	1	0	0	t_0+t_d
1	1	1	1	0	1	t_0+2t_d

- stabilno stanje 1011, promjena na ulazu iz 10 u 11
- nestabilna stanja 1111 i 1100, impuls u trajanju t_d
- novo stabilno stanje 1101

POSLJEDICE KAŠNJENJA

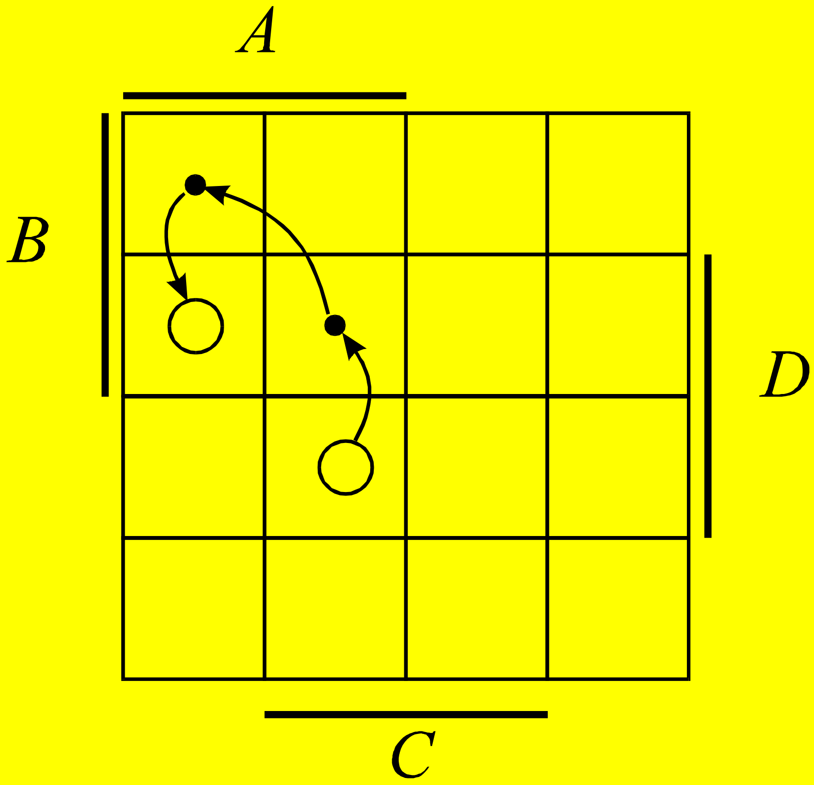
- Isto, u vremenskom dijagramu:



- neželjeni impuls 0 u trajanju t_d

POSLJEDICE KAŠNJENJA

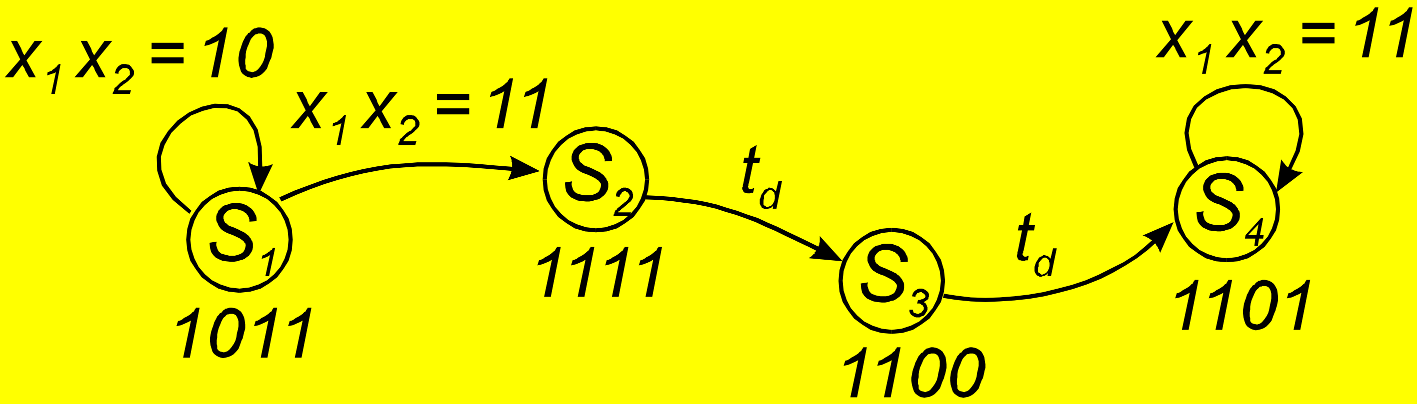
- Isto, u Veitchevom dijagramu:



- (O) stabilno stanje
- (.)   nestabilno stanje
- put kroz stanja
  = trajektorija

POSLJEDICE KAŠNJENJA

- Isto, u usmjerenim grafom:



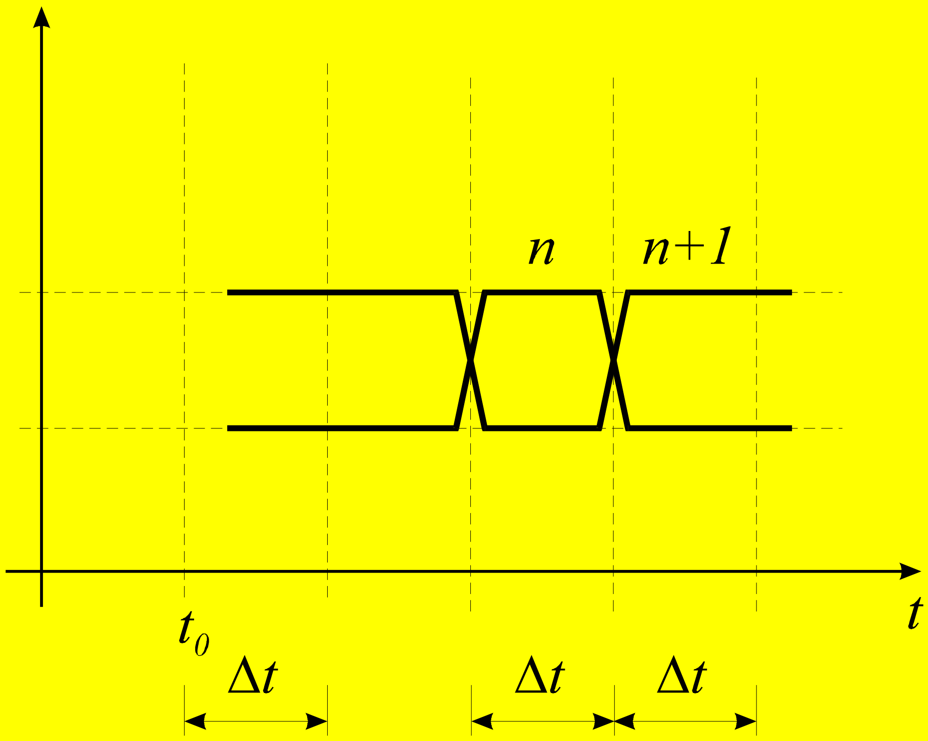
- krugovi: predstavljaju stanja

  stabilna: dok je ulaz isti

  nestabilna: traju t_d
- usmjerene dužine: prijelazi, uz njih pišemo uzrok

PAMĆENJE

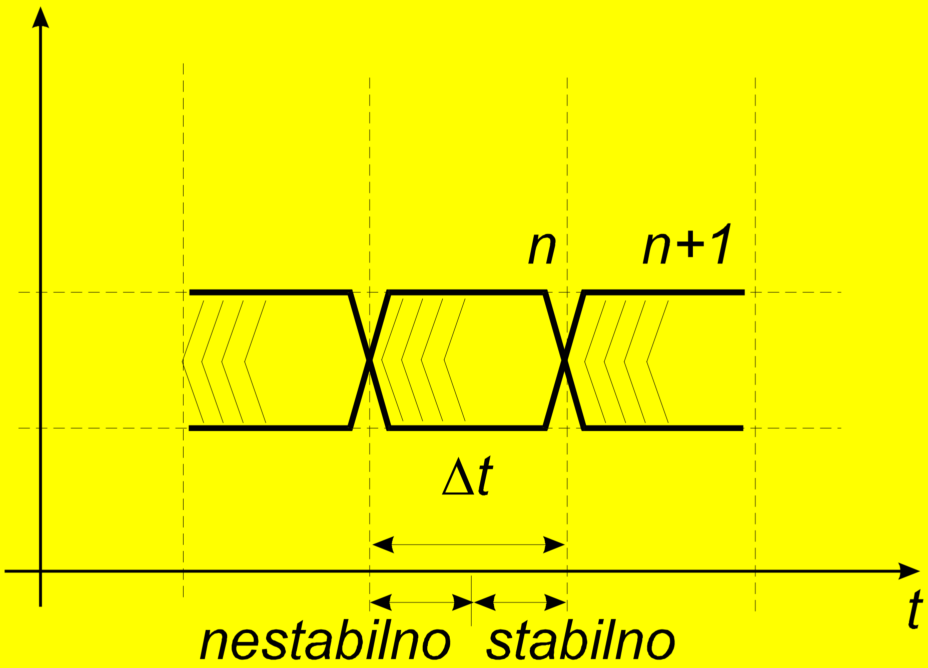
- Želimo pamtiti informaciju neko vrijeme:



- proširimo Δt na proizvoljno trajanje!

PAMĆENJE

- Time kompenziramo prijelaznu pojavu:



- Gradimo **sinkrone** sustave!

3.2. ELEMENTARNI SKLOPOVI ZA PAMĆENJE - BISTABILI

- U DIGITALNOJ TEHNICI

  - koristimo binarni brojevni sustav

  - pamtimo vrijednost Booleove varijable

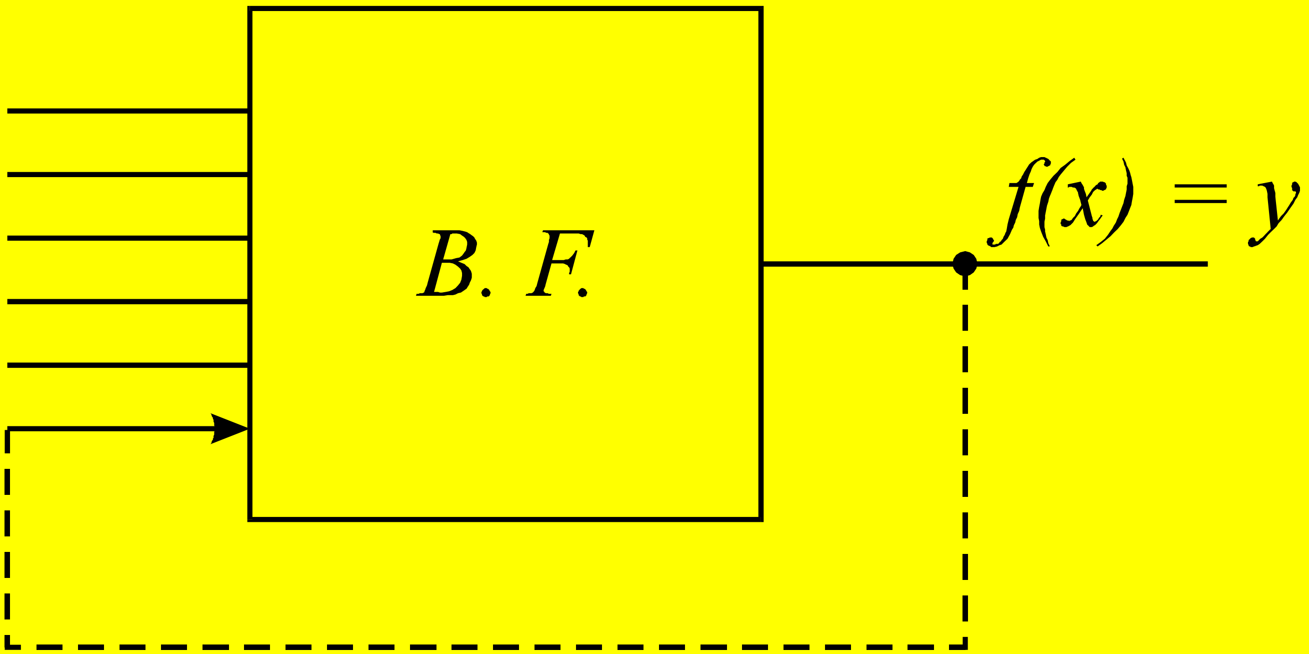
  - dakle, pamtimo konstante 0 i 1

  - trebaju nam dva stabilna stanja

  ⇒ koristimo BISTABIL (flip-flop, latch)

BISTABIL

- Uvedimo povratnu vezu:

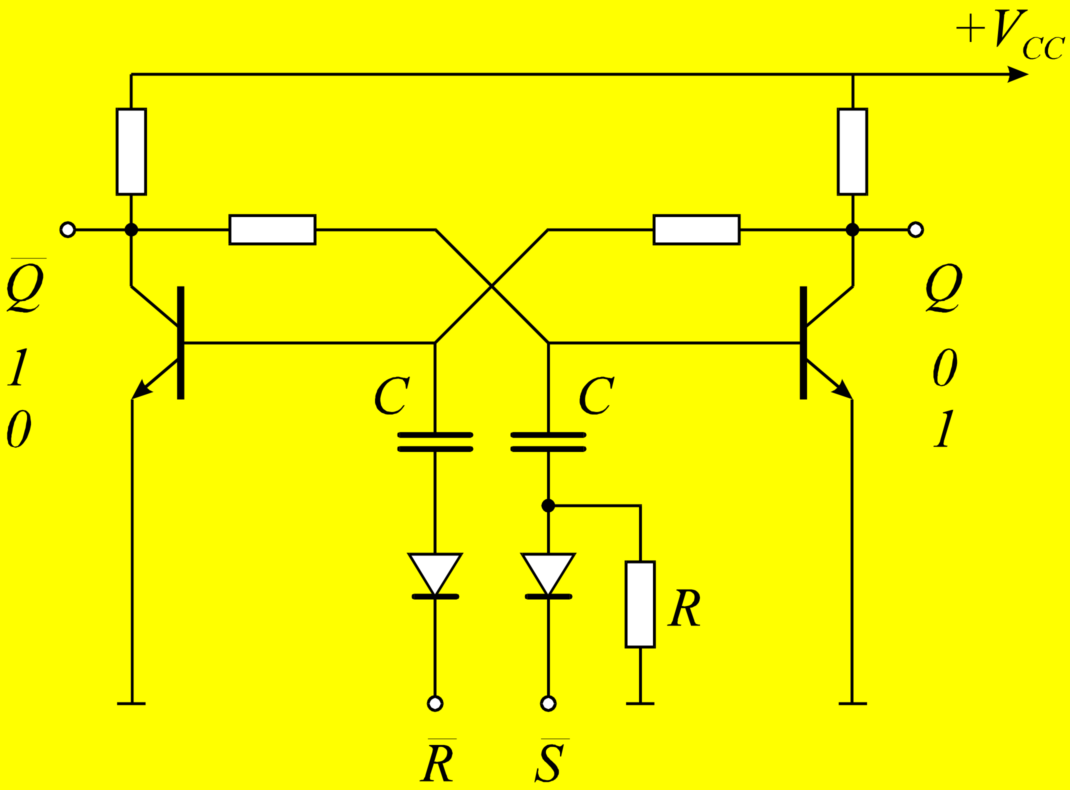


- izlaznu vrijednost dovodimo na ulaz

- izlaz podržava sam sebe, nakon $2*t_d$

BISTABIL

- Korištenjem dva tranzistora:

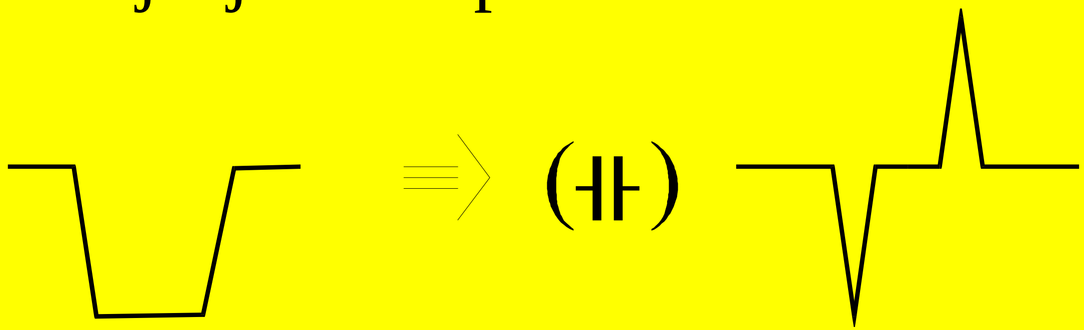


- snažna pozitivna povratna veza

- sprječava oscilacije

BISTABIL

- Stanje mijenjamo impulsima:



- kondenzatorom deriviramo signal

- diodom izdvojimo negativni impuls

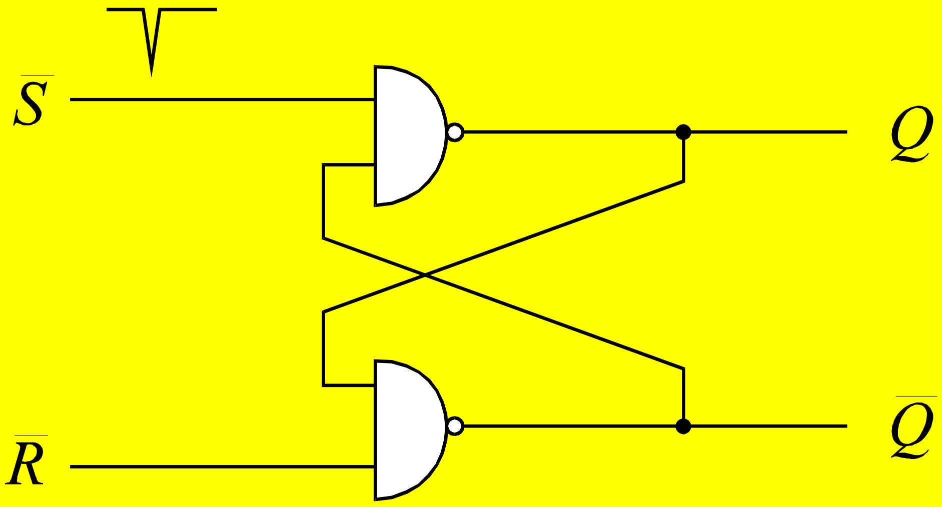
- bistabil je osjetljiv na silazni brid signala

- RS bistabil je osnovni bistabil

- taktom definiramo DISKRETNO VRIJEME

RS BISTABIL

- Koristimo dvoja NI logička vrata:



- impulsom 0 na gornjem ulazu upisujemo 1

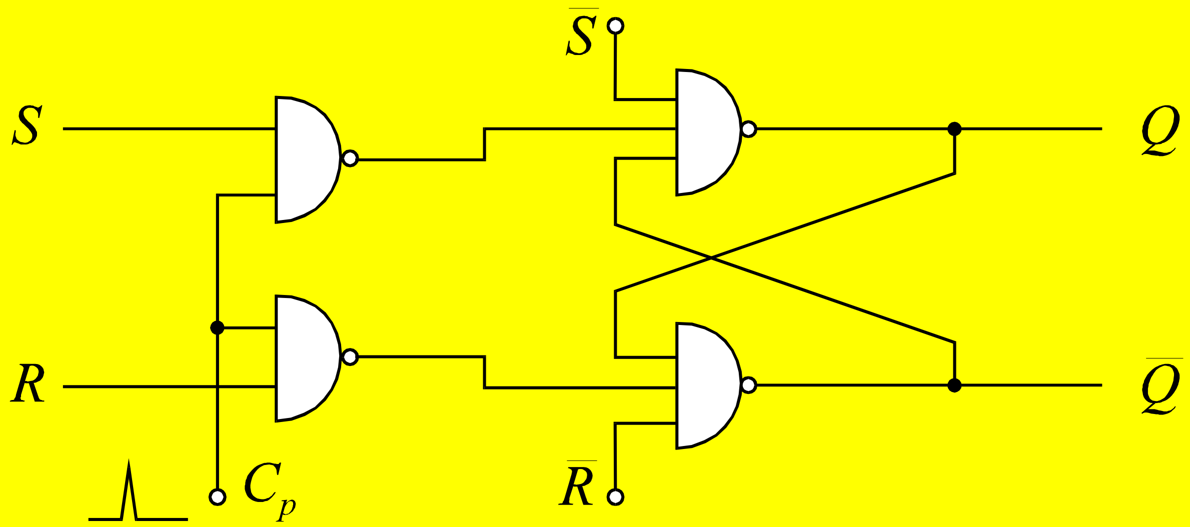
- to je $\overline{\text{S}}$ (Set, postavi 1)

- impulsom 0 na donjem ulazu upisujemo 0

- to je $\overline{\text{R}}$ (Reset, vrati 0)

RS BISTABIL

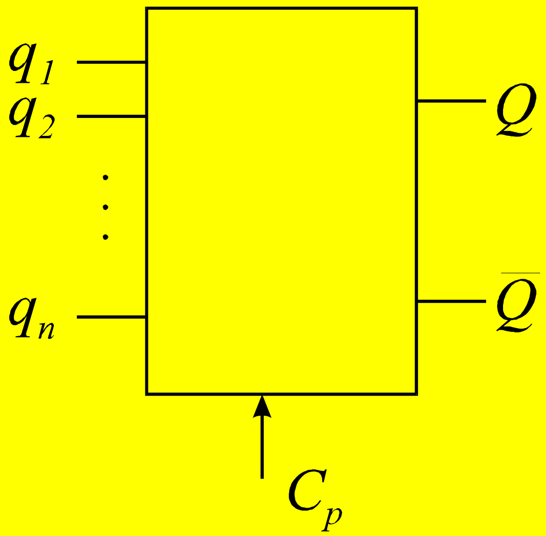
- Kontrolirajmo trenutak djelovanja ulaza:



- impulsom 1 na taktnom ulazu omogućavamo djelovanje S i R ulaza
- S i R ne smiju istovremeno biti u jedinici
- $\overline{\text{R}}$ i $\overline{\text{S}}$ su asinkroni ulazi (postavi početno stanje)

OPĆI BISTABIL

- Promatrajmo neki proizvoljni bistabil:



- u trenutku nastupa taktnog signala
  bistabil mijenja stanje na osnovu:

  * vlastitog trenutnog stanja

  * vrijednosti na ulazima

ZAPISIVANJE BISTABILA

- Tablica prijelaza, potpuna:

$(q_1$	q_2	$\ldots$	q_n	$Q)^n$	Q^{n+1}	Q^{n+1}
0	0	. . .	0	0	0	Q^n
0	0	. . .	0	1	1	
				0	1	$\overline{Q}^n$
		⋮		1	0	
				0	0	0
				1	0	
1	1	. . .	1	0	1	1
1	1	. . .	1	1	1	

- sliči tablici istine, ali ima vremenski odnos

- s lijeve strane su ulazne varijable i stanje u “n”

- s desne strane je buduće stanje, 0 ili 1

ZAPISIVANJE BISTABILA

- Skraćena tablica prijelaza:

$(q_1\ q_2\ \ldots\ q_n)^n$	Q^{n+1}
0 0 . . . 0 0	Q^n
0 0 . . . 0 1	$\overline{Q}^n$
⋮	
1 1 . . . 1 0	0
1 1 . . . 1 1	1

- kao da je razbijena na parcijalne funkcije

- s lijeve strane su ulazne varijable u "n"

- s desne strane je buduće stanje, funkcija od Q^n

ZAPISIVANJE BISTABILA

- Bistabil zapisujemo i funkcijom prijelaza:

$$Q^{n+1} = f\left(Q, q_1, \ldots, q_m\right)^n = \left(G_1\, Q \vee G_2\, \overline{Q}\right)^n$$

- sliči Booleovoj funkciji
- ima vremenski odnos
- Q^{n+1} je funkcija $(Q, q_1 \ldots q_n)^n$
- pišemo u kanonskom obliku
- G_1 opisuje ponašanje bistabila kad je u 1
- G_2 opisuje ponašanje bistabila kad je u 0

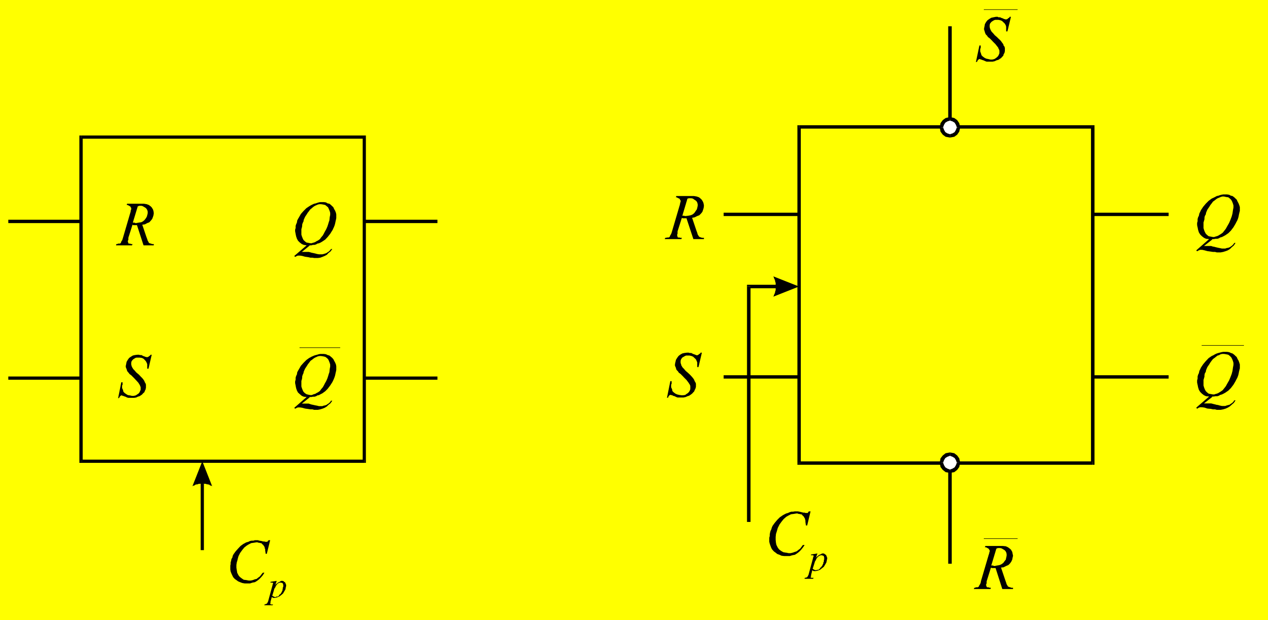
STANDARDNI BISTABILI

- Neki bistabili se proizvode u integriranoj tehnologiji, pa ih zovemo standardnima:

  - RS
  - JK
  - T
  - D

STANDARDNI BISTABILI

- RS bistabil: simbol



STANDARDNI BISTABILI

- RS bistabil (osnovni bistabil): tablice

$(R$	$S)^n$	Q^{n+1}
0	0	Q^n
0	1	1
1	0	0
1	1	X

$(R$	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	X
1	1	1	X

- X neodređen prijelaz

- uvjet RS=0

STANDARDNI BISTABILI

- RS bistabil: funkcija prijelaza

$$Q^{n+1} = \left(\overline{R}\,Q \vee S\,\overline{Q}\right)^n$$

Q^{n+1}:

R^n (columns 1–2), Q^n (columns 2–3), S^n (row 1):

x	x	*1*	*1*
		1	

$$G_1 = \overline{R} \qquad G_2 = S$$

$$Q^{n+1} = \left(\overline{R}\,Q \vee \overline{R}\,S\right)^n$$

$$Q^{n+1} = \left(\overline{R}\,(Q \vee S)\right)^n$$

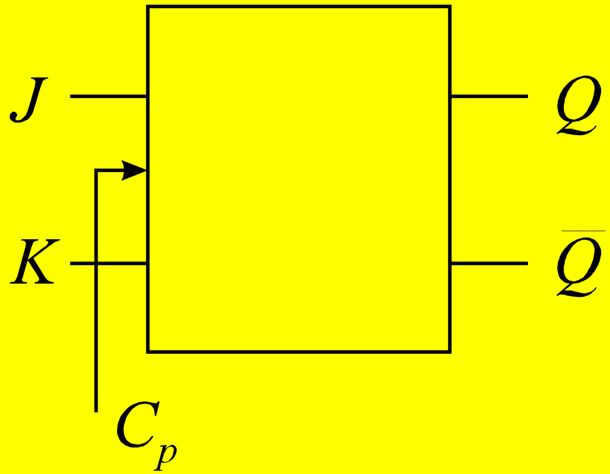
uvjet RS=0

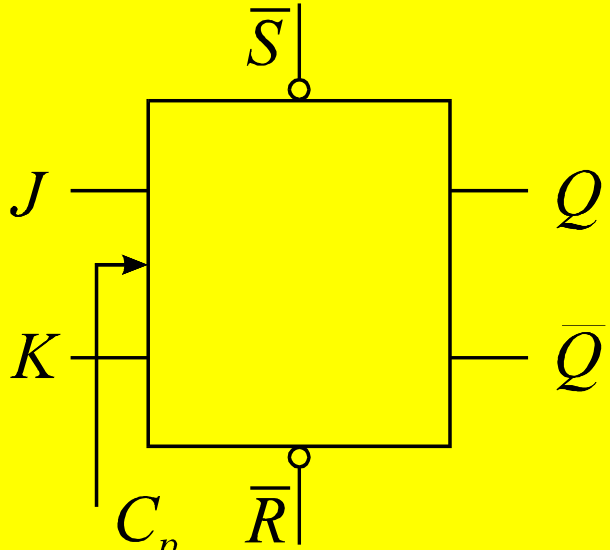
osiguravamo izvana!

$$Q^{n+1} = \left(\overline{R}\,Q \vee S\right)^n$$

STANDARDNI BISTABILI

- JK bistabil: simbol





STANDARDNI BISTABILI

- JK bistabil (univerzalni bistabil): tablice

$(J$	$K)^n$	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	$\overline{Q}^n$

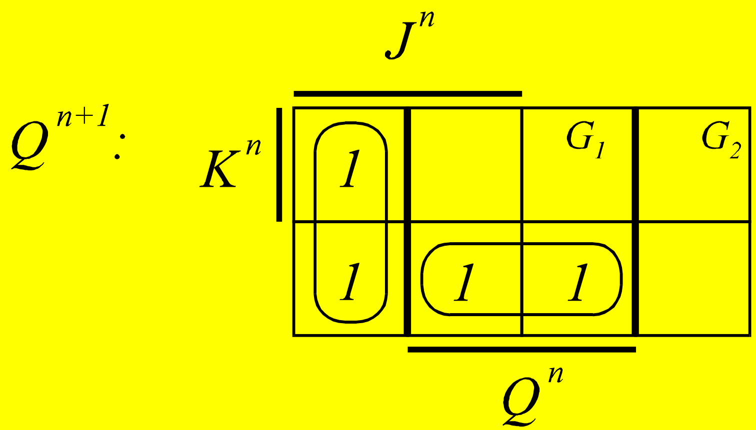
$(J$	K	$Q)^n$	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

potpuno upravljiv

sa ulaza

STANDARDNI BISTABILI

- JK bistabil: funkcija prijelaza

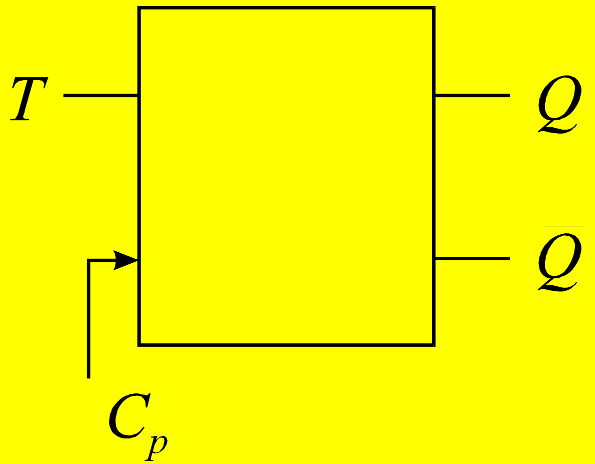


$$Q^{n+1} = \left(\overline{K}\,Q \vee J\,\overline{Q}\right)^n$$

$$G_1 = \overline{K} \qquad G_2 = J$$

STANDARDNI BISTABILI

- T bistabil (brojila): tablice



T^n	Q^{n+1}
0	Q^n
1	$\overline{Q}^n$

STANDARDNI BISTABILI

- T bistabil: simbol i funkcija prijelaza

$(T$	$Q)^n$	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q^{n+1}:

		T
Q	G_1	1
	G_2 1	

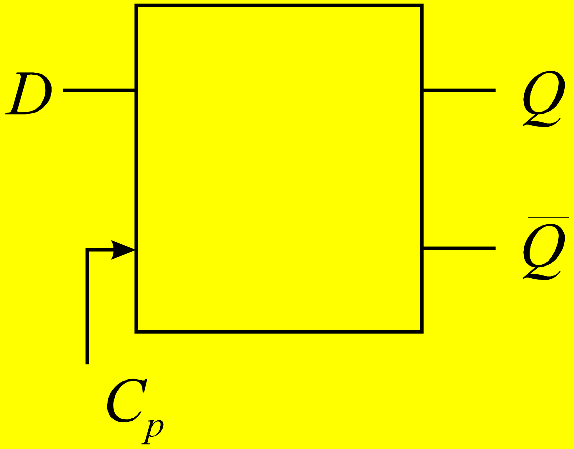
$$Q^{n+1} = \left(\overline{T}\,Q \vee T\,\overline{Q}\right)^n$$

$$G_1 = \overline{T} \qquad G_2 = T$$

$$G_1 = \overline{G}_2$$

STANDARDNI BISTABILI

- D bistabil (registri): tablice



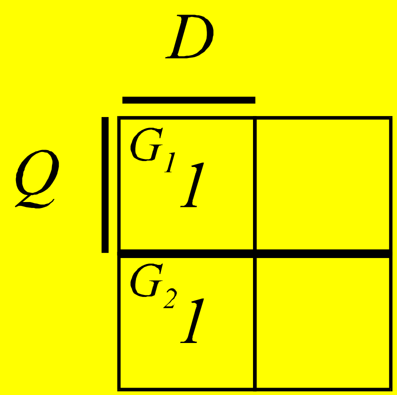
D^n	Q^{n+1}
0	0
1	1

STANDARDNI BISTABILI

- D bistabil: simbol i funkcija prijelaza

(D	$Q)^n$	Q^{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Q^{n+1}:



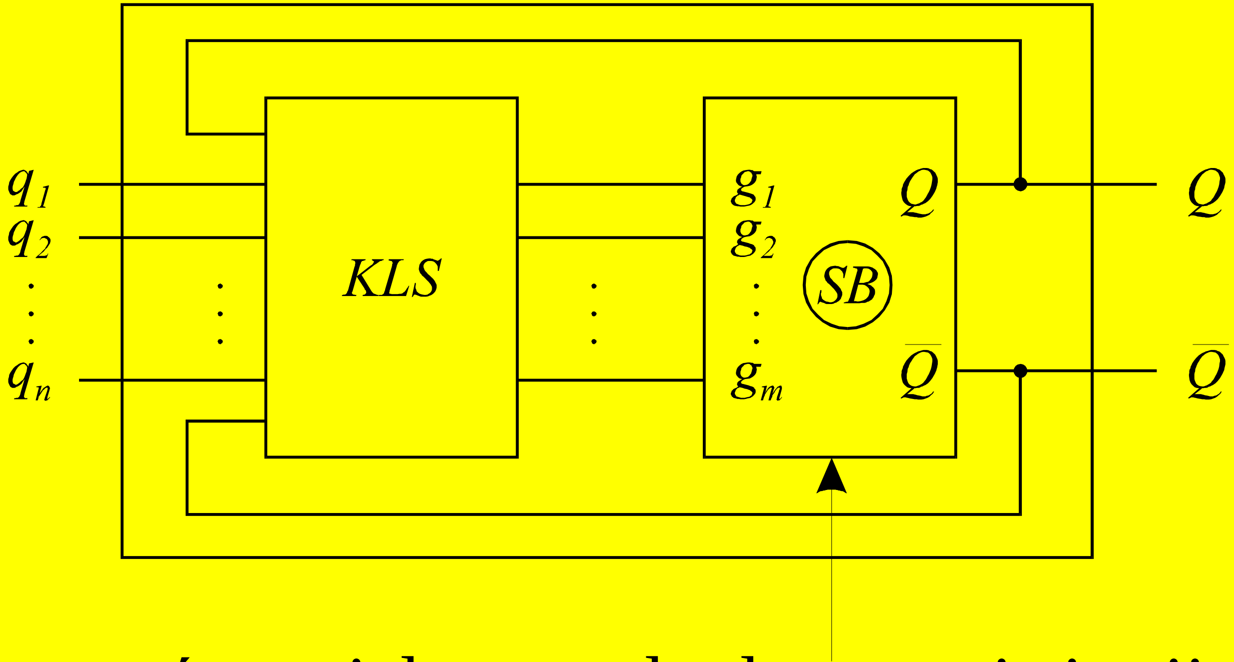
$$Q^{n+1} = \left(D\,Q \vee D\,\overline{Q}\right)^n \qquad Q^{n+1} = D^n$$

$$G_1 = D \qquad G_2 = D$$

$$G_1 = G_2$$

SINTEZA OPĆIH BISTABILA

- Koristimo standardne bistabile i MODEL:



- izlaz općeg = izlaz standardnog => isti prijelazi

- KLS modificira ulaze da bi standardni obavljao iste prijelaze $Q_{OB}^{n} = Q_{SB}^{n}$ $Q_{OB}^{n+1} = Q_{SB}^{n+1}$

SINTEZA OPĆIH BISTABILA

- METODA REKONSTRUKCIJE (za RS i T):

$(q_1$	$q_2 \ldots q_n$	$Q)^n$		Q^{n+1}	$(g_1$	$g_2 \ldots$	$g_m)^n$
		0	$\rightarrow$	0		…	
		0	$\rightarrow$	1		…	
		1	$\rightarrow$	0		…	
		1	$\rightarrow$	1		…	

- potpunu tablicu prijelaza nadopunimo potrebnim ulazima u standardni bistabil da bi radio iste prijelaze

- lijeva i dodana desna strana čine TABLICU ISTINE za KLS

SINTEZA OPĆIH BISTABILA

- Rekonstrukciju obavljamo prema tablici:

Q^n		Q^{n+1}	R	S	J	K	T	D
0	$\rightarrow$	0	r	0	0	r	0	0
0	$\rightarrow$	1	0	1	1	r	1	1
1	$\rightarrow$	0	1	0	r	1	1	0
1	$\rightarrow$	1	0	r	r	0	0	1

- dobivena na osnovu potpunih tablica standardnih bistabila

SINTEZA OPĆIH BISTABILA

- Primjer: RS, NI realizirati JK bistabil

  - rekonstruirajmo potrebne ulaze u RS bistabil

(J	K	$Q)^n$	Q^{n+1}	(R	$S)^n$
0	0	0	0	r	0
0	0	1	1	0	r
0	1	0	0	r	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	0	r
1	1	0	1	0	1
1	1	1	0	1	0

  - jednostavna kontrola uvjeta RS=0

SINTEZA OPĆIH BISTABILA

- Primjer: RS, NI realizirati JK bistabil
  - minimizacija:

R:

	J	J		
K		*1*	*1*	*r*
				r
		Q	Q	

S:

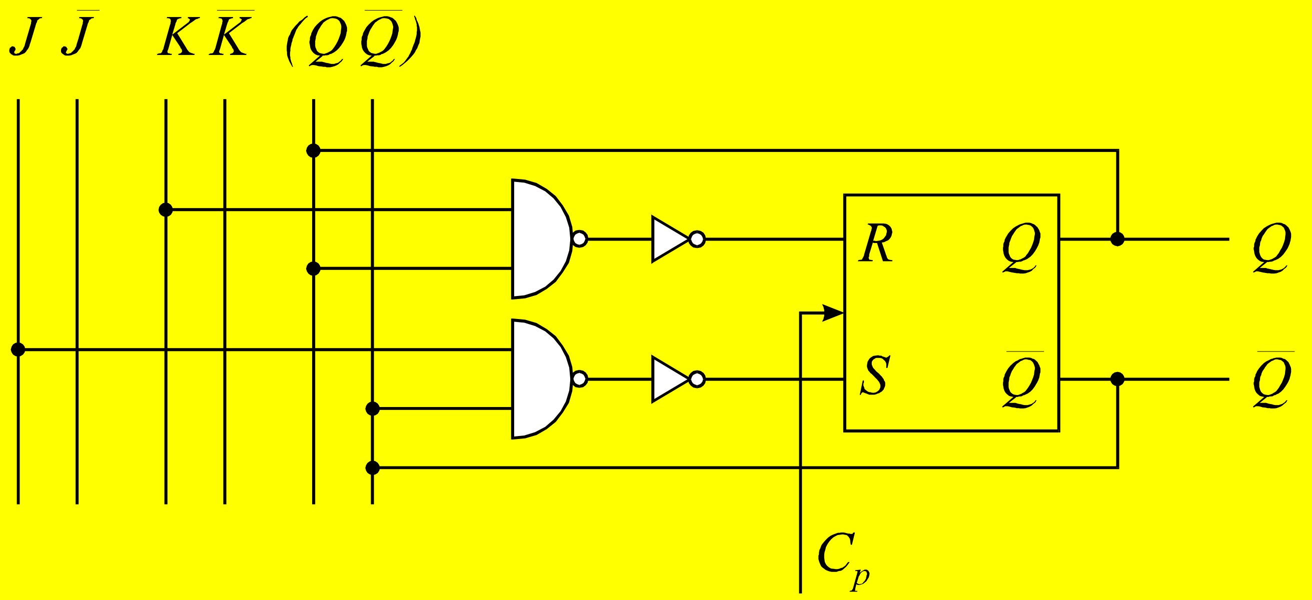
	J	J		
K	*1*			
	1	*r*	*r*	
		Q	Q	

$$R = K\,Q = \overline{\overline{K\,Q}}$$

$$S = J\,\overline{Q} = \overline{\overline{J\,\overline{Q}}}$$

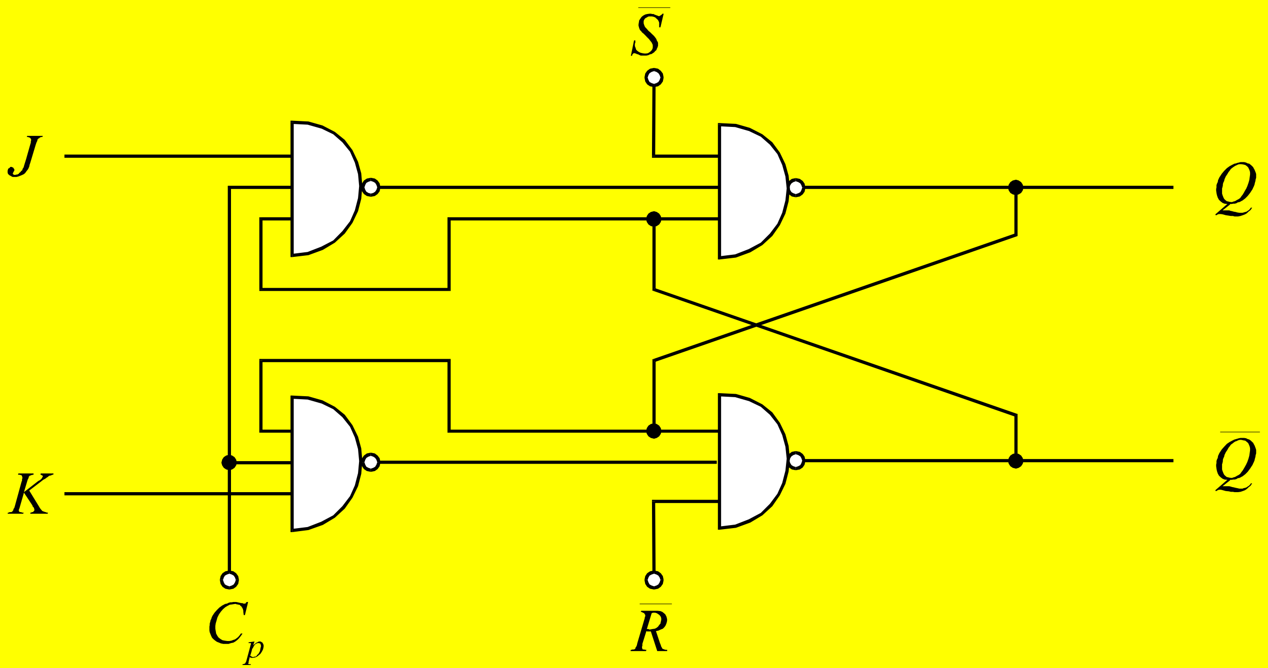
SINTEZA OPĆIH BISTABILA

- Primjer: RS, NI realizirati JK bistabil
  - shema:



SINTEZA OPĆIH BISTABILA

- Primjer: RS, NI realizirati JK bistabil
  - shema stvarnog sklopa:



SINTEZA OPĆIH BISTABILA

- METODA IZJEDNAČAVANJA (za JK bistabil):
  - izjednačimo jednadžbe prijelaza
    aplikacionu (općeg bistabila)
    i karakterističnu (standardnog bistabila)

$$Q_{SB}^{n+1} = Q_{OB}^{n+1} \qquad \left(H_1 Q \vee H_2 \overline{Q}\right)^n = \left(G_1 Q \vee G_2 \overline{Q}\right)^n$$

$$H_1 = G_1 \qquad H_2 = G_2$$

- za JK bistabil

$$\overline{K} Q \vee J \overline{Q} = G_1 Q \vee G_2 \overline{Q} = f\left(q_1, q_2, \ldots, q_m\right)^n$$

$$K = \overline{G}_1 \qquad J = G_2$$

SINTEZA OPĆIH BISTABILA

- METODA ZA D BISTABIL:
  - istovremeno i metoda rekonstrukcije
    i metoda izjednačavanja
  - zasniva se na:

$$Q^{n+1} = D^n$$

  - tablica prijelaza je ujedno tablica istine za KLS:

$(q_1 \quad q_2 \; \dots \; q_m \quad Q)^n$	$Q^{n+1}=D^n$

PRIMJERI SINTEZE OPĆIH BISTABILA

- REALIZIRATI BISTABIL AB:
  zadan skraćenom tablicom

A	B	Q^{n+1}
0	0	0
0	1	Q^n
1	0	$\overline{Q}^n$
1	1	1

PRIMJERI SINTEZE OPĆIH BISTABILA

- ZA BISTABIL AB:
  napišemo potpunu tablicu i rekonstruiramo:

(A	B	Q)^n	Q^{n+1}	R	S	T	$\overline{T}$	D
0	0	0	0	r	0	0	1	0
0	0	1	0	1	0	1	0	0
0	1	0	0	r	0	0	1	0
0	1	1	1	0	r	0	1	1
1	0	0	1	0	1	1	0	1
1	0	1	0	1	0	1	0	0
1	1	0	1	0	1	1	0	1
1	1	1	1	0	r	0	1	1

PRIMJERI SINTEZE OPĆIH BISTABILA

- Realizirati bistabil AB korištenjem RS i NI:
  - minimiziramo KLS prema rekonstruiranim R i S

R:

	A	A		
B				*r*
		1	*1*	*r*
		Q	Q	

S:

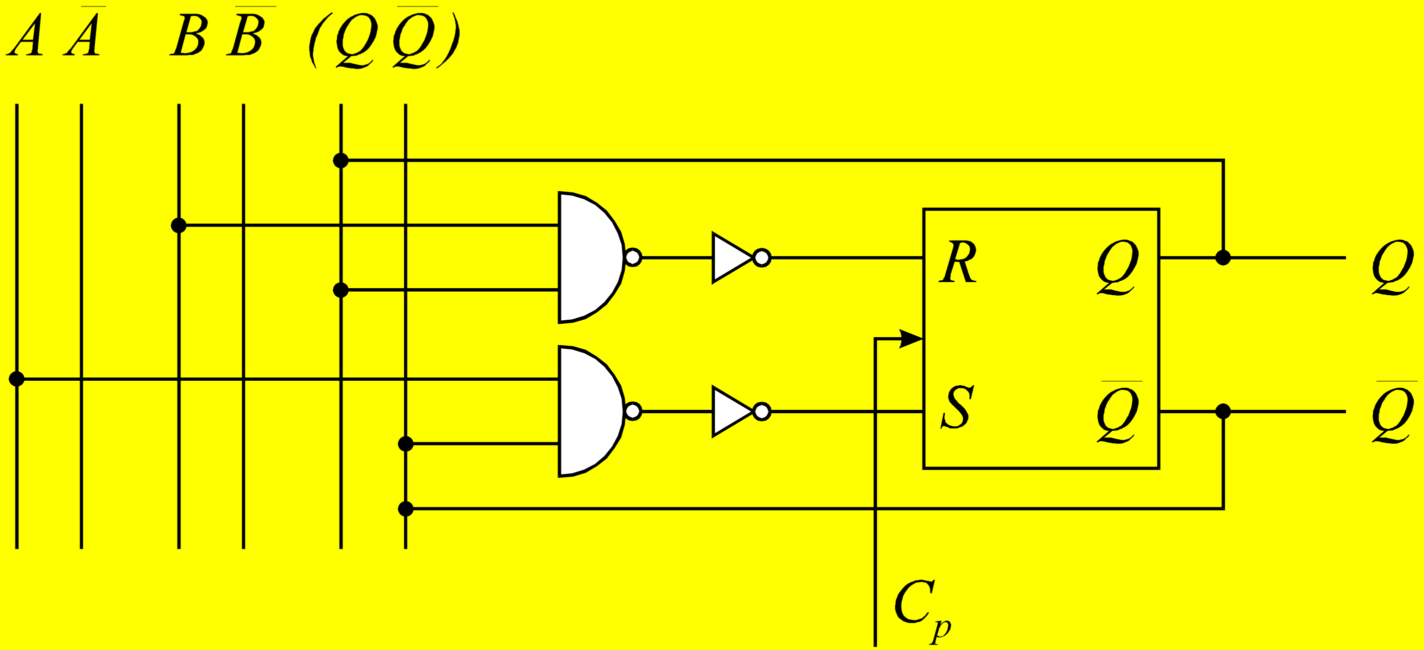
	A	A		
B	*1*	*r*	*r*	
	1			
		Q	Q	

$$R = \bar{B}\,Q = \overline{\overline{\bar{B}\,Q}}$$

$$S = A\,\bar{Q} = \overline{\overline{A\,\bar{Q}}}$$

PRIMJERI SINTEZE OPĆIH BISTABILA

- Realizirati bistabil AB korištenjem RS i NI:
  - nacrtamo shemu



PRIMJERI SINTEZE OPĆIH BISTABILA

- Realizirati bistabil AB korištenjem T i NILI:
  - minimiziramo KLS prema rekonstruiranom $\overline{T}$

$\overline{T}$:

	A	A		
B		1	1	1
				1
		Q	Q	

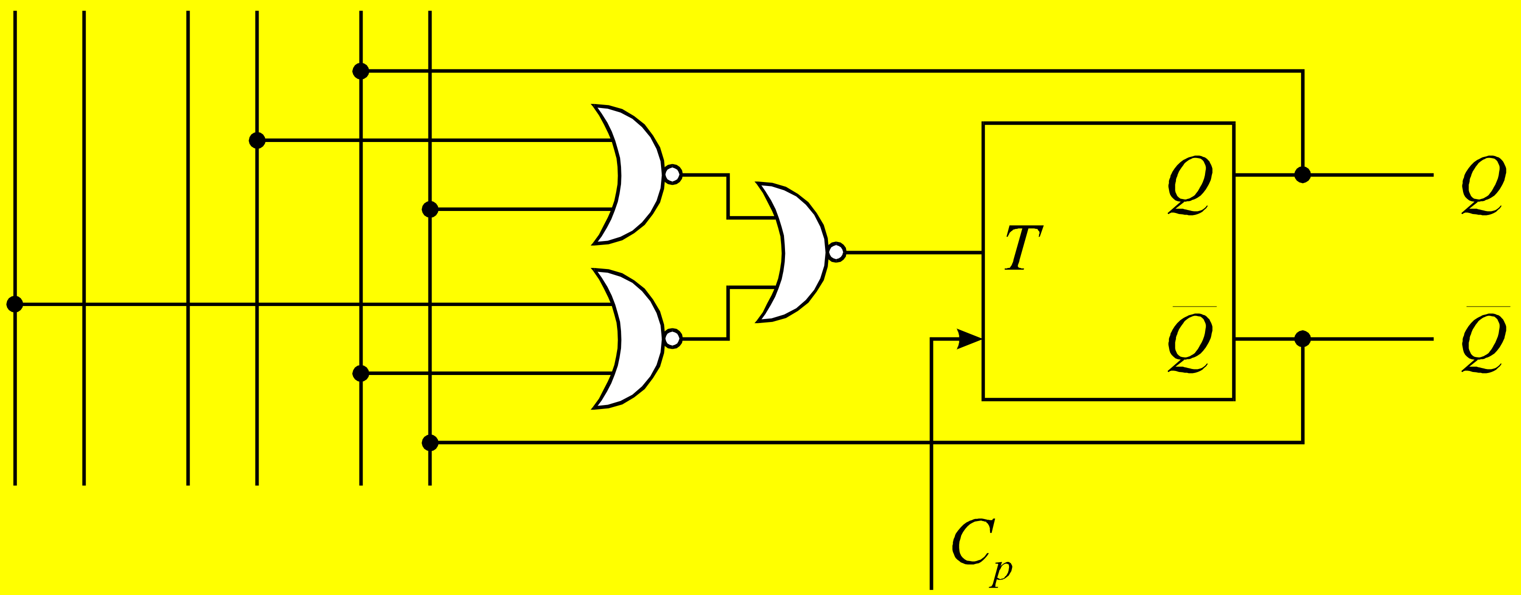
$$\overline{T} = \overline{\overline{BQ}} \vee \overline{\overline{\overline{A}\,\overline{Q}}} \qquad /^{-}$$

$$T = \overline{\overline{\overline{B} \vee \overline{Q}} \vee \overline{A \vee Q}}$$

PRIMJERI SINTEZE OPĆIH BISTABILA

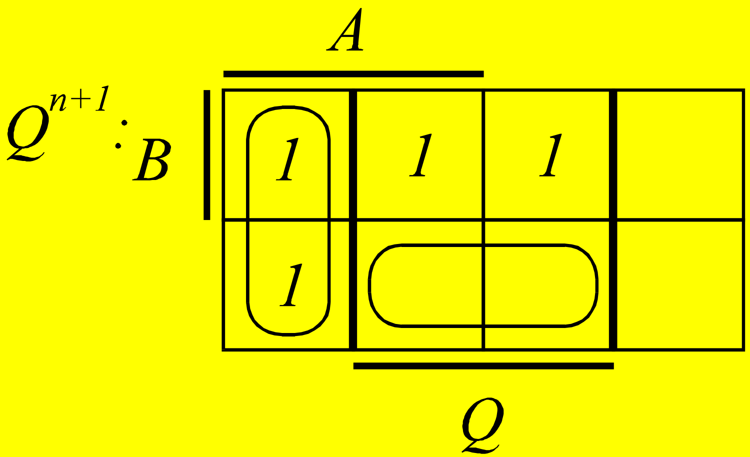
- Realizirati bistabil AB korištenjem T i NILI:
  - nacrtamo shemu

$A \ \bar{A} \quad B \ \bar{B} \quad (Q \ \bar{Q})$



PRIMJERI SINTEZE OPĆIH BISTABILA

- Realizirati bistabil AB korištenjem JK i NI ili NILI:
  - metoda izjednačavanja
  - upišemo Q^{n+1}, minimiziramo $\overline{G}_1$ i G_2

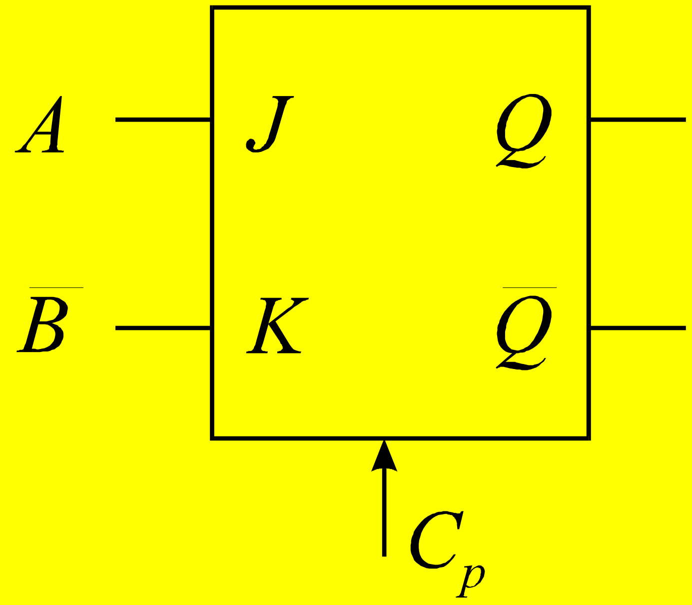


NI	NILI
$K = \overline{G}_1$	$\overline{K} = G_1$
$J = G_2$	$\overline{J} = \overline{G}_2$

$$Q^{n+1} = G_1 Q \vee G_2 \overline{Q} = \overline{K} Q \vee J \overline{Q} \quad \Rightarrow \quad K = \overline{B} \quad J = A$$

PRIMJERI SINTEZE OPĆIH BISTABILA

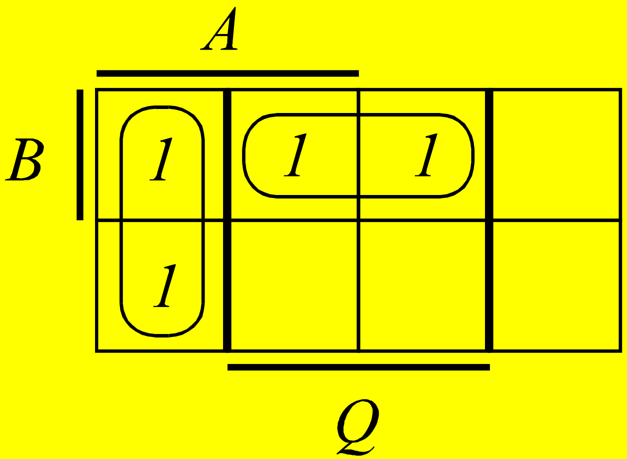
- Realizirati bistabil AB korištenjem JK i NI ili NILI:
  - nacrtamo shemu



PRIMJERI SINTEZE OPĆIH BISTABILA

- Realizirati bistabil AB korištenjem D i NI:
  - minimiziramo D:

$Q^{n+1} = D^n$:

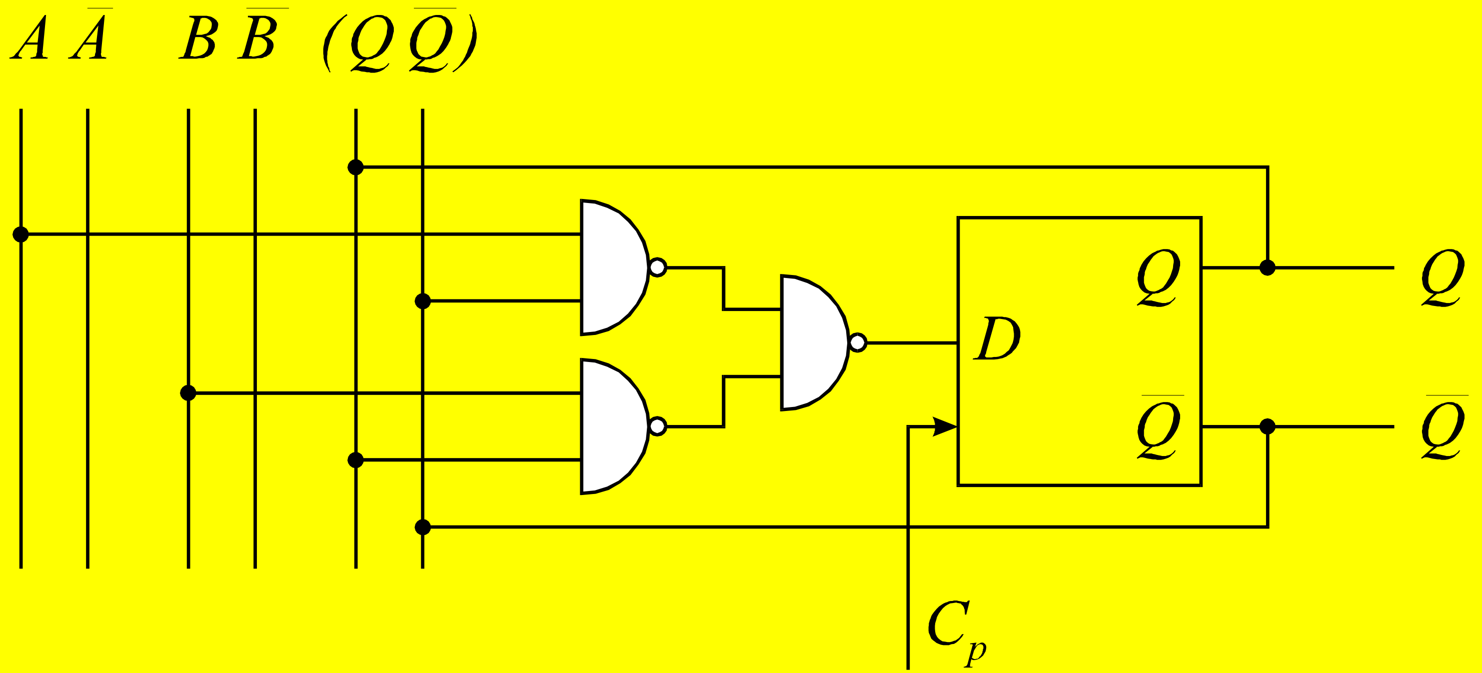


$$D = A\overline{Q} \vee BQ \quad / =$$

$$D = \overline{\overline{A\overline{Q}}\;\overline{BQ}}$$

PRIMJERI SINTEZE OPĆIH BISTABILA

- Realizirati bistabil AB korištenjem D i NI:
  - nacrtamo shemu

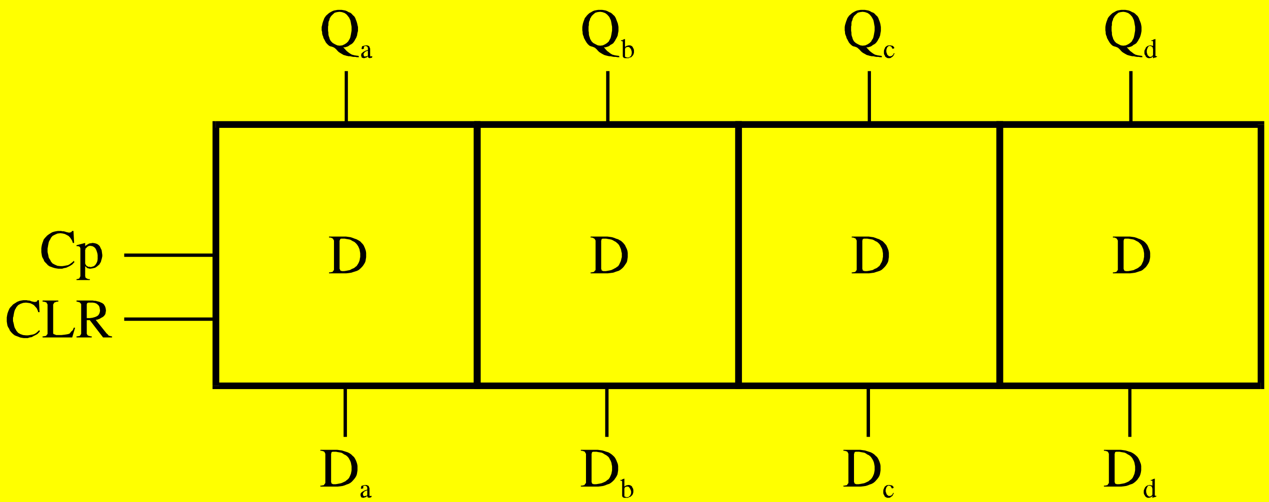


3.3. SLOŽENE STRUKTURE S BISTABILIMA

- REGISTAR (pamtilo, buffer)
- POMAČNI REGISTAR (shift registar)
- BROJILO (counter)

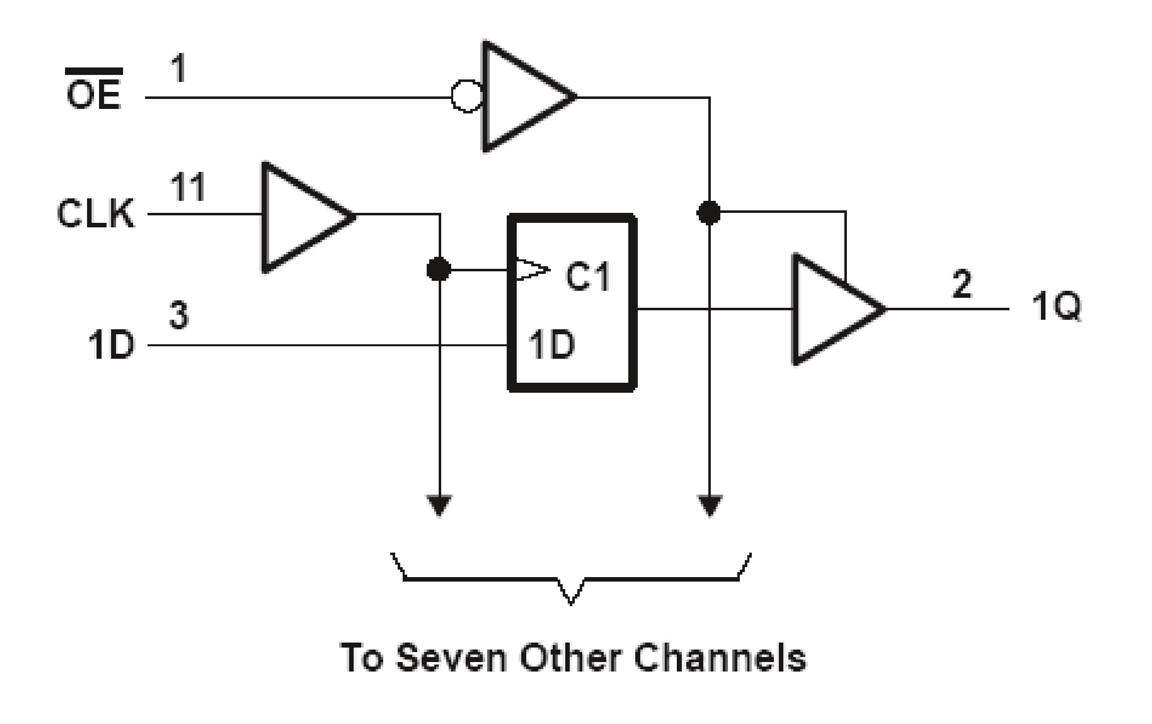
SLOŽENE STRUKTURE

- REGISTAR:
  - više D bistabila sa zajedničkim taktnim ulazom
  - pamti kodnu riječ sa ulaza kao cjelinu
  - nekad zajednički R ulaz (početno stanje 0)



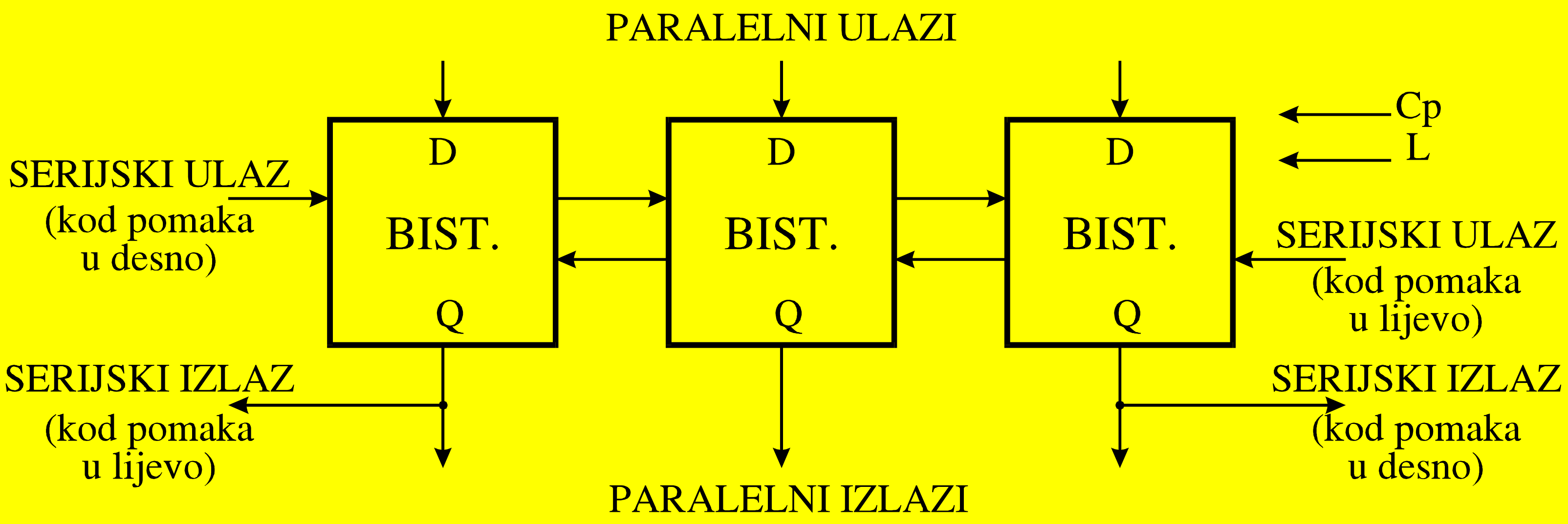
SLOŽENE STRUKTURE

- REGISTAR: 74374



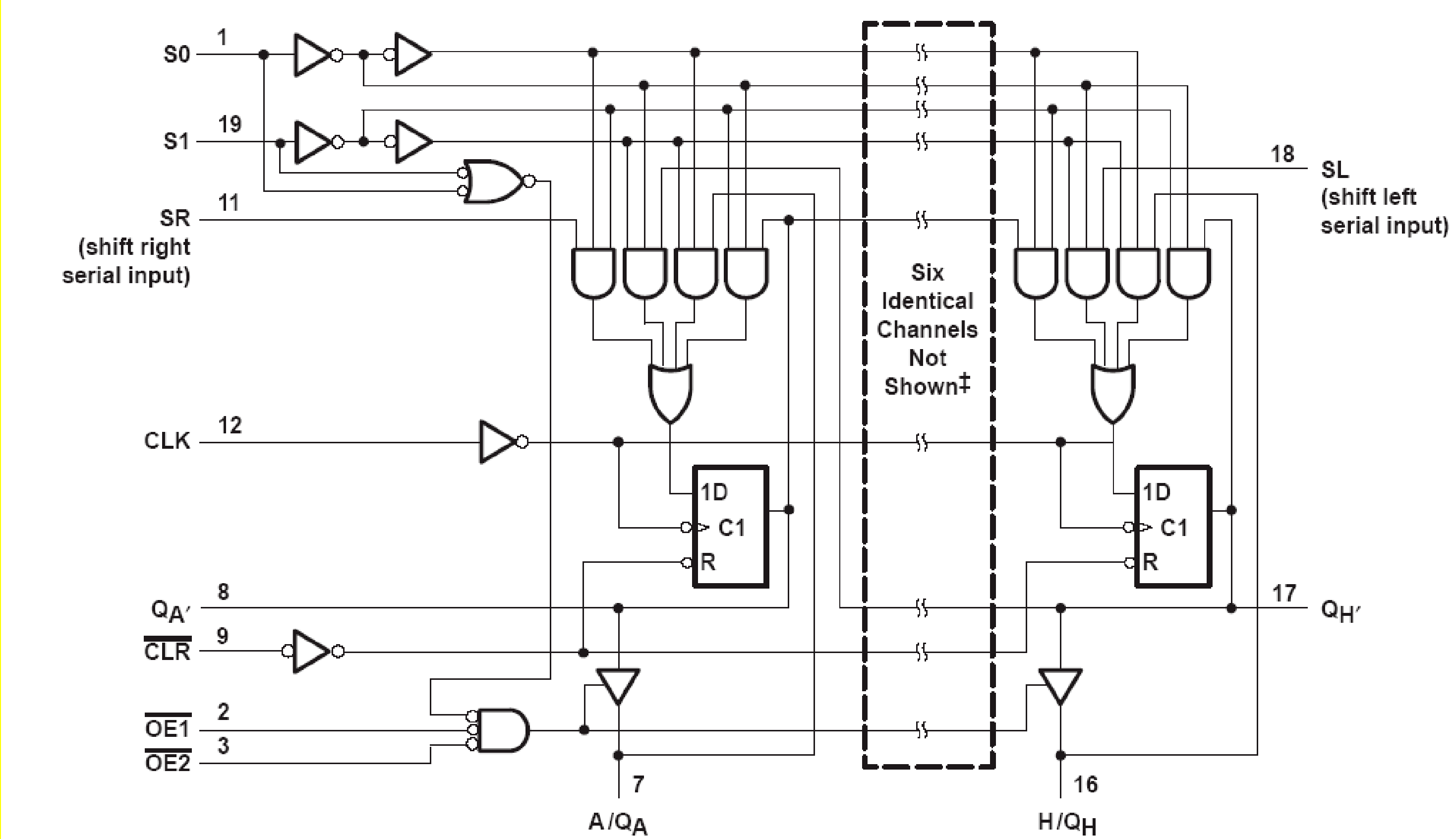
SLOŽENE STRUKTURE

- POMAČNI REGISTAR:
  - ulaz spojen na izlaz
  - kodnu riječ pomiče u desno ili lijevo
  - množenje/dijeljenje s 2, paralelno-serijska pretv.



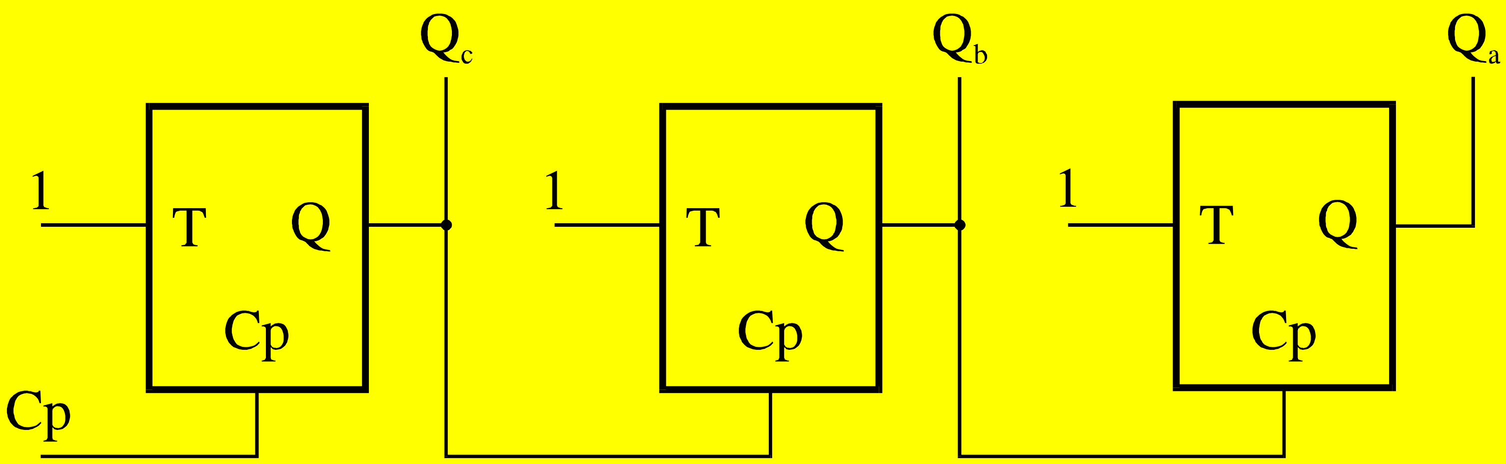
SLOŽENE STRUKTURE

- POMAČNI REGISTAR: 74299



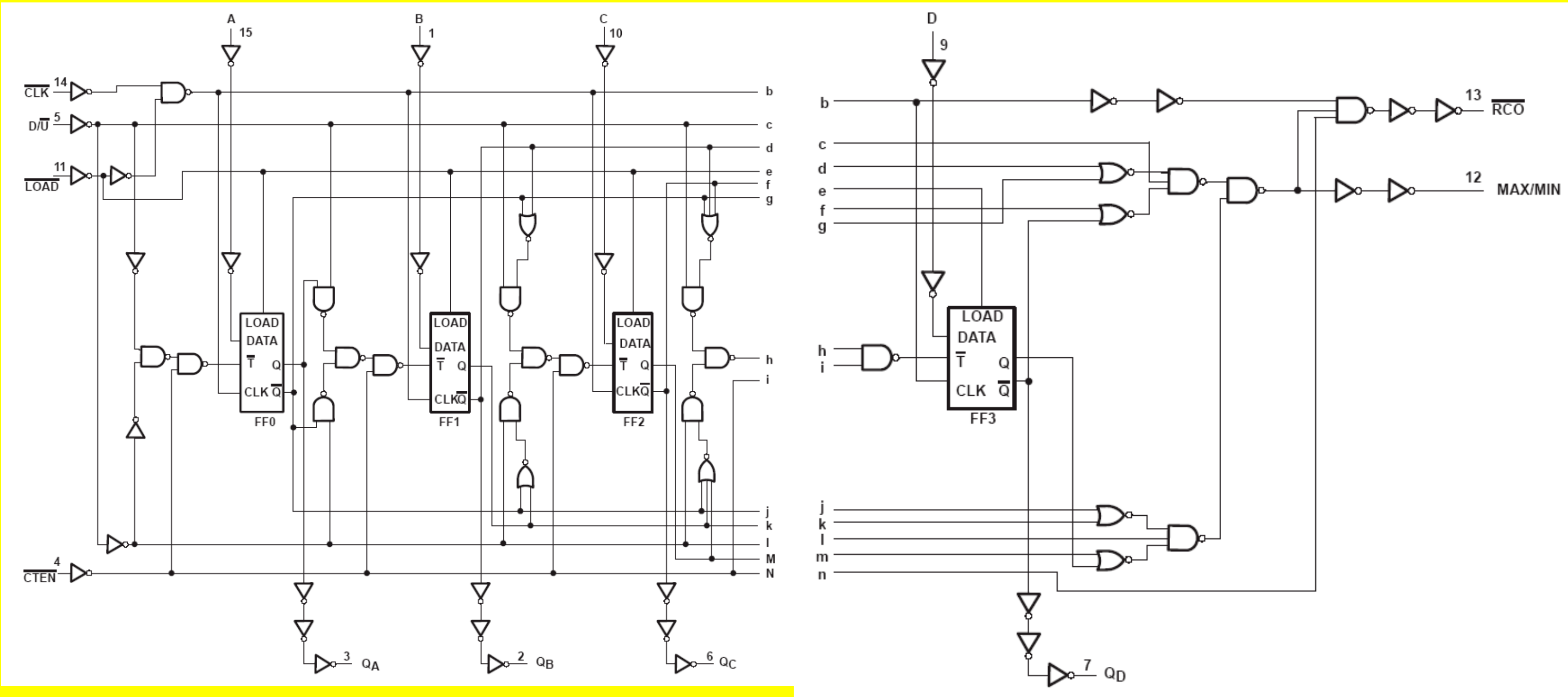
SLOŽENE STRUKTURE

- BROJILO (asinkrono):
  - taktni ulaz spojen na izlaz (asinkroni rad)
  - generira niz kodnih riječi
  - binarno brojilo ($0..2^n-1$), dekadsko brojilo (0..9)



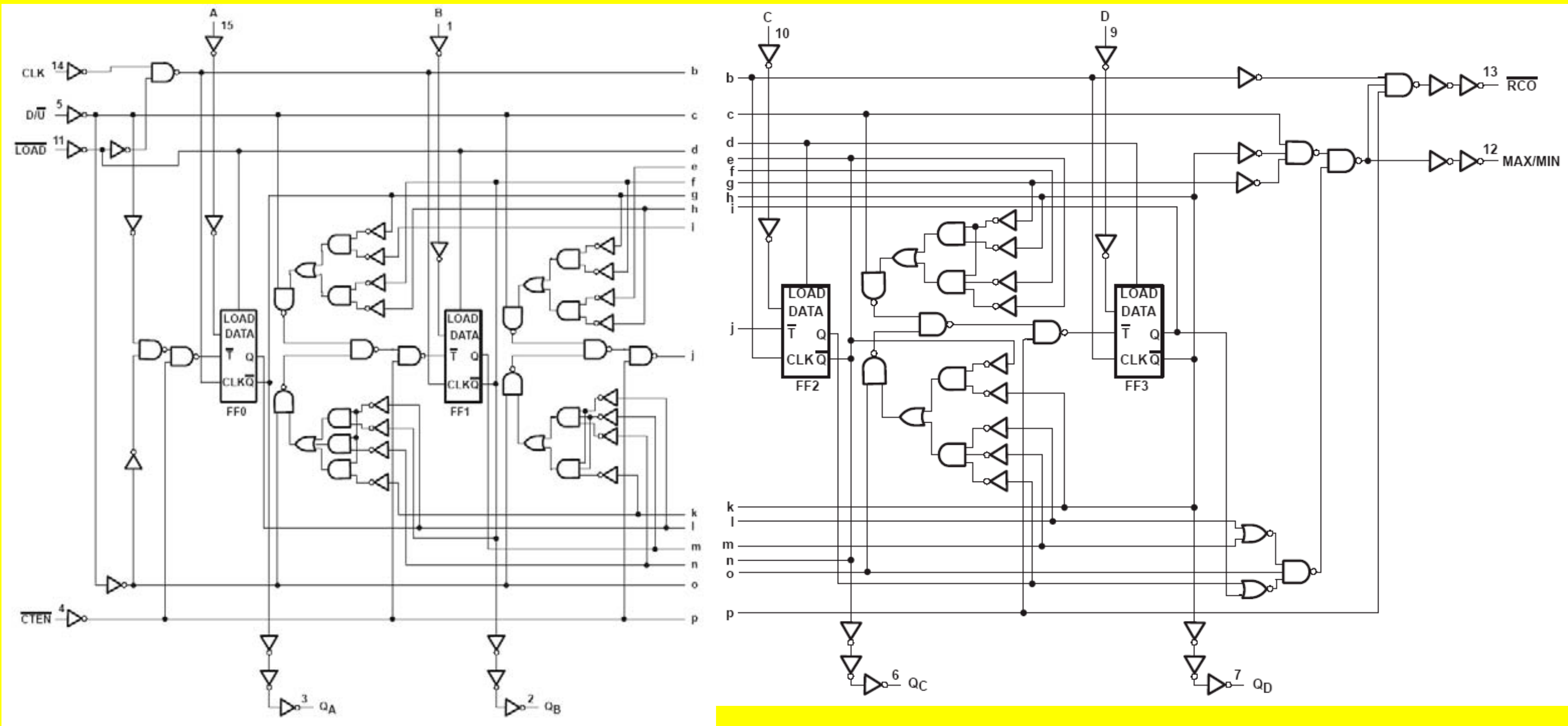
SLOŽENE STRUKTURE

- BROJILO (sinkrono, binarno): 74193



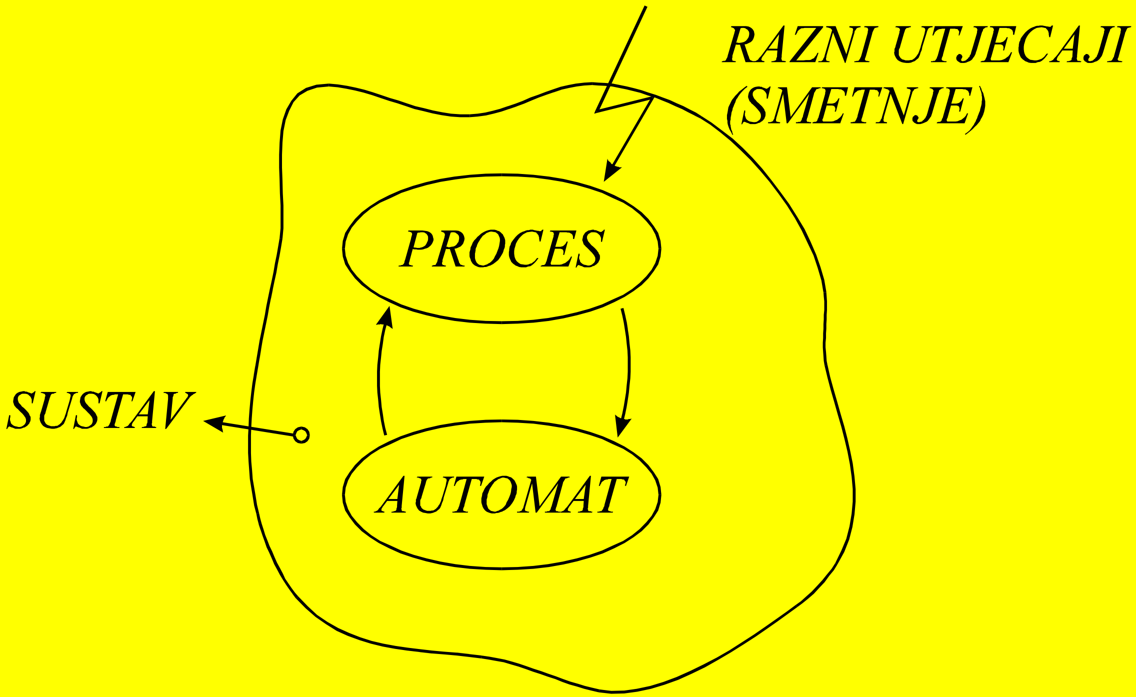
SLOŽENE STRUKTURE

- BROJILO (sinkrono, dekadsko): 74190



3.4. DIGITALNI AUTOMATI

- TEORIJA SUSTAVA



- sustav: proces na kojeg utječe okolina i automat koji regulira stanje procesa

TEORIJA SUSTAVA

- PROCES
  - treba biti u optimalnom režimu
  - po zadanoj funkciji cilja
- AUTOMAT
  - mjeri stanje procesa
  - razlučuje sva bitna stanja procesa (mjerljivost)
  - poznaje svojstva procesa (ugrađeno znanje)
  - želi kompenzirati utjecaj okoline
  - generira izlaze (naredbe)
  - raspolaže dovoljnim brojem izlaza (upravljivost)

DIGITALNI AUTOMAT

- DISKRETAN
  - radi u diskretnom vremenu
- KONAČAN
  - ima konačan broj stanja, konačnu memoriju
- DIGITALAN
  - raspolaže digitalnim ulazima i izlazima

DIGITALNI AUTOMAT

- DETERMINIRAN
  - jednoznačno obavlja svoju funkciju
- SPECIFICIRAN
  - **potpuno**: očekuje proizvoljni niz ulaza
  - **nepotpuno**: mogući su samo neki nizovi ulaza
- SINKRON
  - diskretno vrijeme određeno taktnim signalom

DIGITALNI AUTOMAT

- DEFINIRAN PETORKOM

$$A = \langle U, I, S, \delta, \lambda \rangle$$

- U: skup ulaznih simbola
  kodiranih kodnim riječima varijabli X:

$$U = \{u_1, u_2, \ldots, u_p\}$$

$$X = \{x_1, x_2, \ldots, x_e\}$$

$$2^e \geq p$$

DIGITALNI AUTOMAT

- I: skup izlaznih simbola
  kodiranih kodnim riječima varijabli Y:

$$I = \{i_1, i_2, \ldots, i_q\}$$
$$Y = \{y_1, y_2, \ldots, y_m\}$$
$$2^m \geq q$$

- S: skup unutrašnjih stanja
  kodiranih kodnim riječima varijabli Z
  (a to su kodne riječi stanja bistabila memorije)

$$S = \{s_1, s_2, \ldots, s_n\}$$
$$Z = \{z_1, z_2, \ldots, z_k\}$$
$$2^k \geq n$$

DIGITALNI AUTOMAT

- δ: FUNKCIJA PRIJELAZA
  određuje slijedeće stanje
  na osnovu ulaza i sadašnjeg stanja:

$$\delta: \qquad s^{n+1} = \delta(s,u)^n \qquad S \times U \rightarrow S$$

- λ: FUNKCIJA IZLAZA
  određuje sadašnji izlaz
  - MEALY: na osnovu ulaza i sadašnjeg stanja
  - MOORE: na osnovu sadašnjeg stanja

$$\lambda: \qquad i^n = \begin{cases} \lambda(s,u)^n & \text{Mealy} \quad S \times U \rightarrow I \\ \lambda(s)^n & \text{Moore} \quad S \rightarrow I \end{cases}$$

ZAPISIVANJE AUTOMATA

- TABLICA PRIJELAZA I IZLAZA
  za Mealyev model automata:

	s^{n+1}	i^n
	$u_1, u_2, \ldots, u_p$	$u_1, u_2, \ldots, u_p$
s_1		
s_2	s_j	i_k
$\vdots$		
s_n		

svako mjesto u tablici odgovara jednom paru s,u

ZAPISIVANJE AUTOMATA

- TABLICA PRIJELAZA I IZLAZA

  za Mooreov model automata:

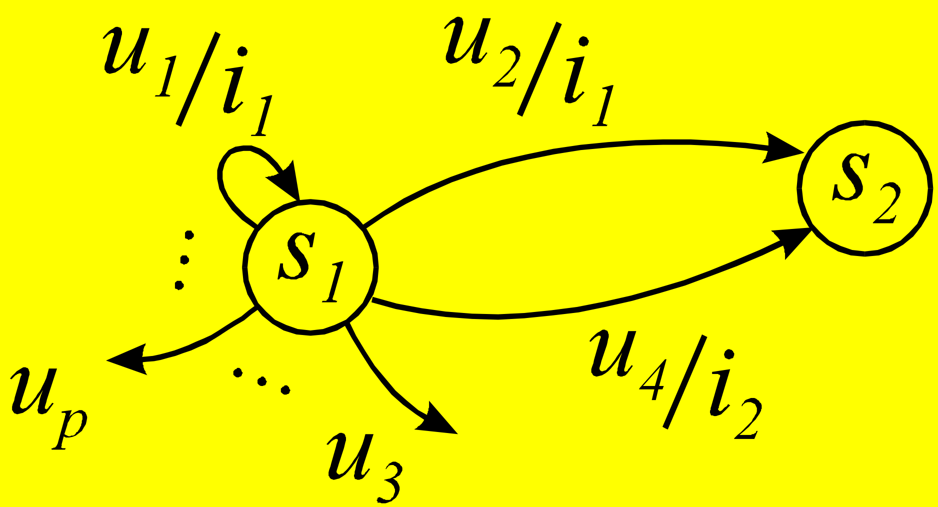
	s^{n+1}	i^{n}
	$u_1, u_2, \ldots, u_p$	
s_1		
s_2	s_j	i_k
$\vdots$		
s_n		

izlazi ovise samo o stanju s

ZAPISIVANJE AUTOMATA

- USMJERENIM GRAFOM

  za Mealyev model automata:

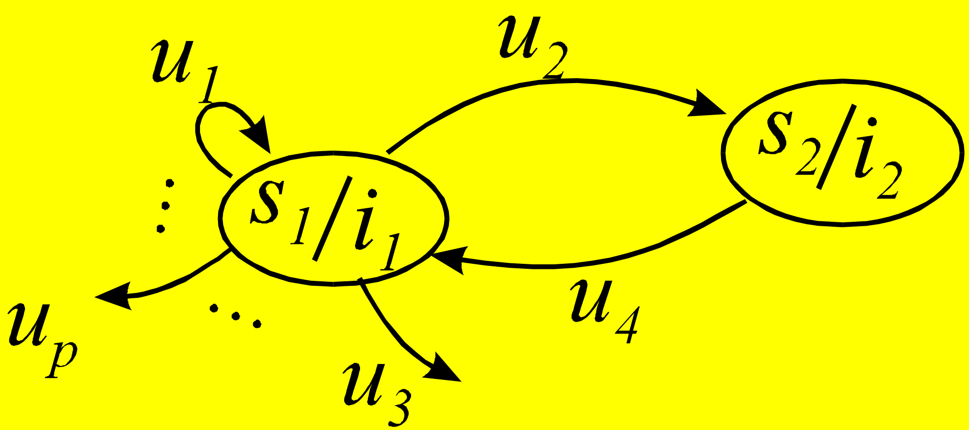


- čvorovi su stanja, usmjerene duljine su prijelazi
- uz duljine pišemo izlaze jer ovise o stanju i ulazu

ZAPISIVANJE AUTOMATA

- USMJERENIM GRAFOM

  za Mooreov model automata:



- čvorovi su stanja, usmjerene duljine su prijelazi

- uz stanja pišemo izlaze jer ovise samo o stanju

SINTEZA AUTOMATA

- APSTRAKTNA SINTEZA
  - zadavanje automata
  - minimizacija
- STRUKTURNA SINTEZA
  - kodiranje stanja, ulaza i izlaza
  - uvrštavanje kodova, prepoznavanje:
    * tablica prijelaza za pojedine bistabile
    * tablica istine za izlazne varijable
  - realizacija automata
    * općim bistabilima i logičkim vratima
    * mux-demux strukturom i D bistabilima (MDD)

APSTRAKTNA SINTEZA AUTOMATA

- ZADAVANJE AUTOMATA: tri pristupa

  - transformator sekvence

    pravila: ulazna sekvenca →izlazna sekvenca

    (matematičke gramatike)

  - akceptor sekvence

    pravila: ulazna sekvenca →izlazni simbol

    (jezik regularnih izraza)

  - korak po korak

    pravila: ulazni simbol →izlazni simbol

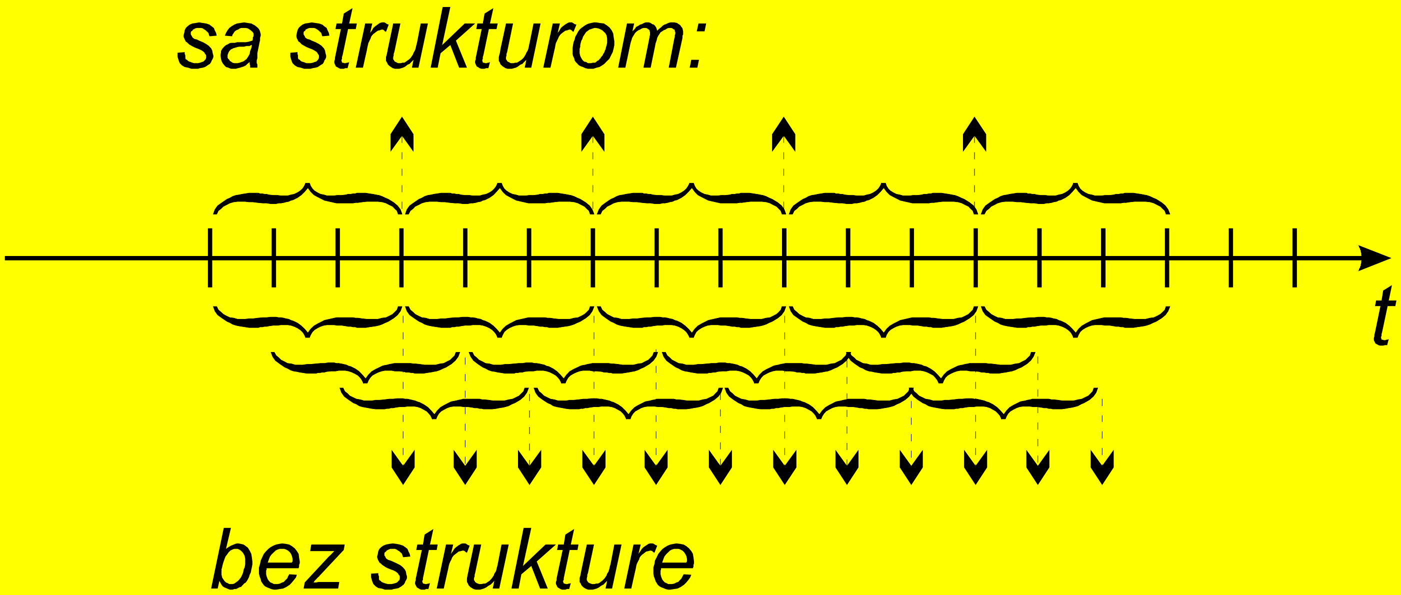
    (metoda potpunog stabla)

ULAZNA SEKVENCA

- BESKONAČNA BEZ STRUKTURE
  - niz nezavisnih ulaznih simbola
  - tražena sekvenca započinje bilo kada
- BESKONAČNA SA STRUKTUROM
  - niz konačnih sekvenci
  - konačne sekvence iste duljine:
    sve sekvence moguće
  - konačne sekvence različite duljine:
    kraće ne smiju biti početak duljih
  - tražena sekvenca započinje kad prethodna završi
    (automat održava sinkronizaciju)

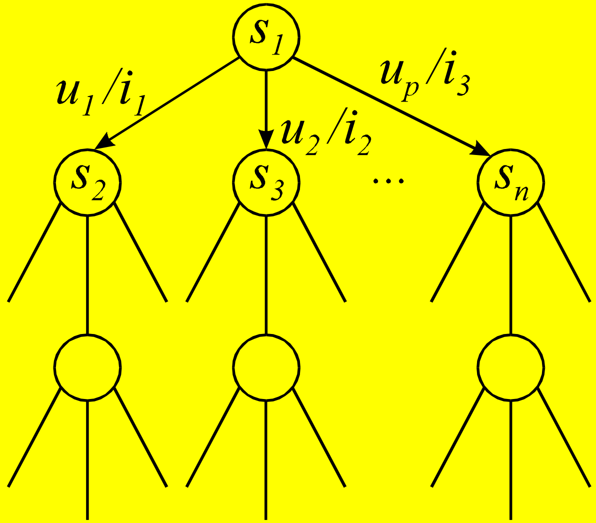
ULAZNA SEKVENCA

Odluke sa i bez strukture:



ZADAVANJE KORAK PO KORAK

- CRTAMO GRAF
  (potpuno stablo)



- ISPISUJEMO TABLICU PRIJELAZA I IZLAZA
  (primitivna, početna tablica)

ZADAVANJE KORAK PO KORAK

- PRIMJER:

  zadati automat koji ima skupove U i I:

$$U = \{u_1, u_2\} \qquad I = \{i_1, i_2\}$$

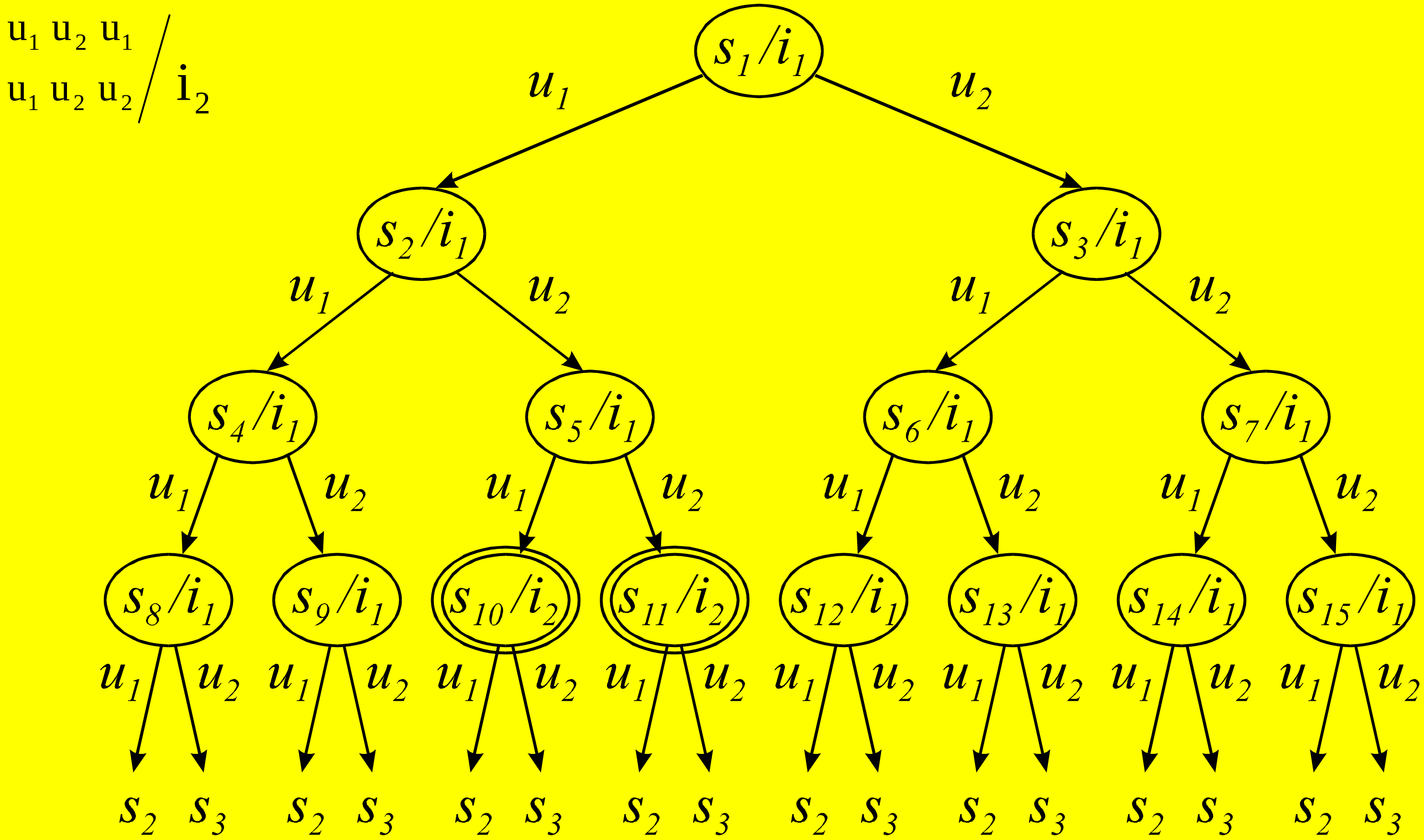
i koji traži ulazne sekvence

$$u_1\ u_2\ u_1 \qquad u_1\ u_2\ u_2$$

kad nađe neku od njih, na izlazu daje i_2

ZADAVANJE KORAK PO KORAK

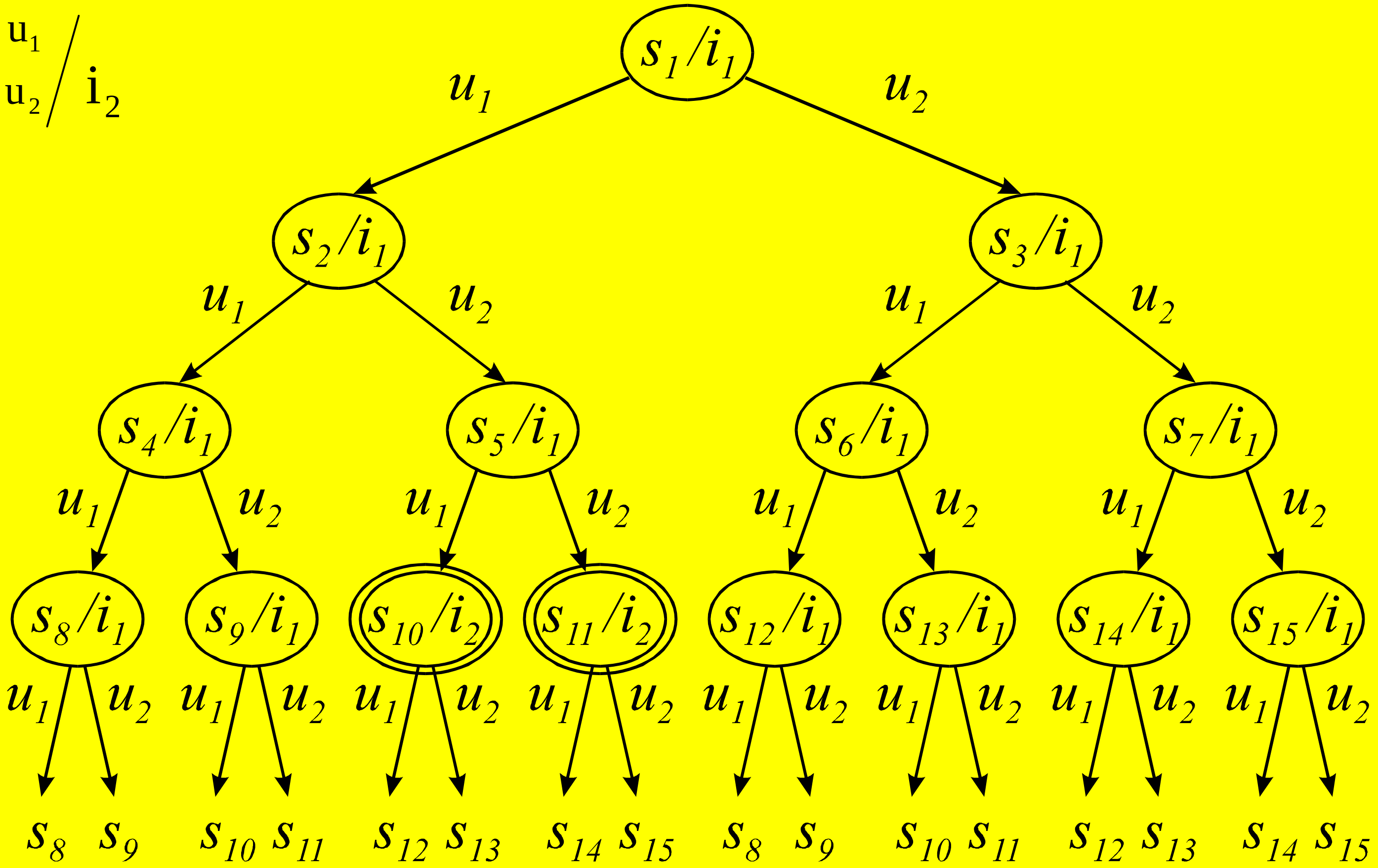
- MOORE automat, sekvenca SA strukturom:



ZADAVANJE KORAK PO KORAK

- MOORE automat, sekvenca BEZ strukture:

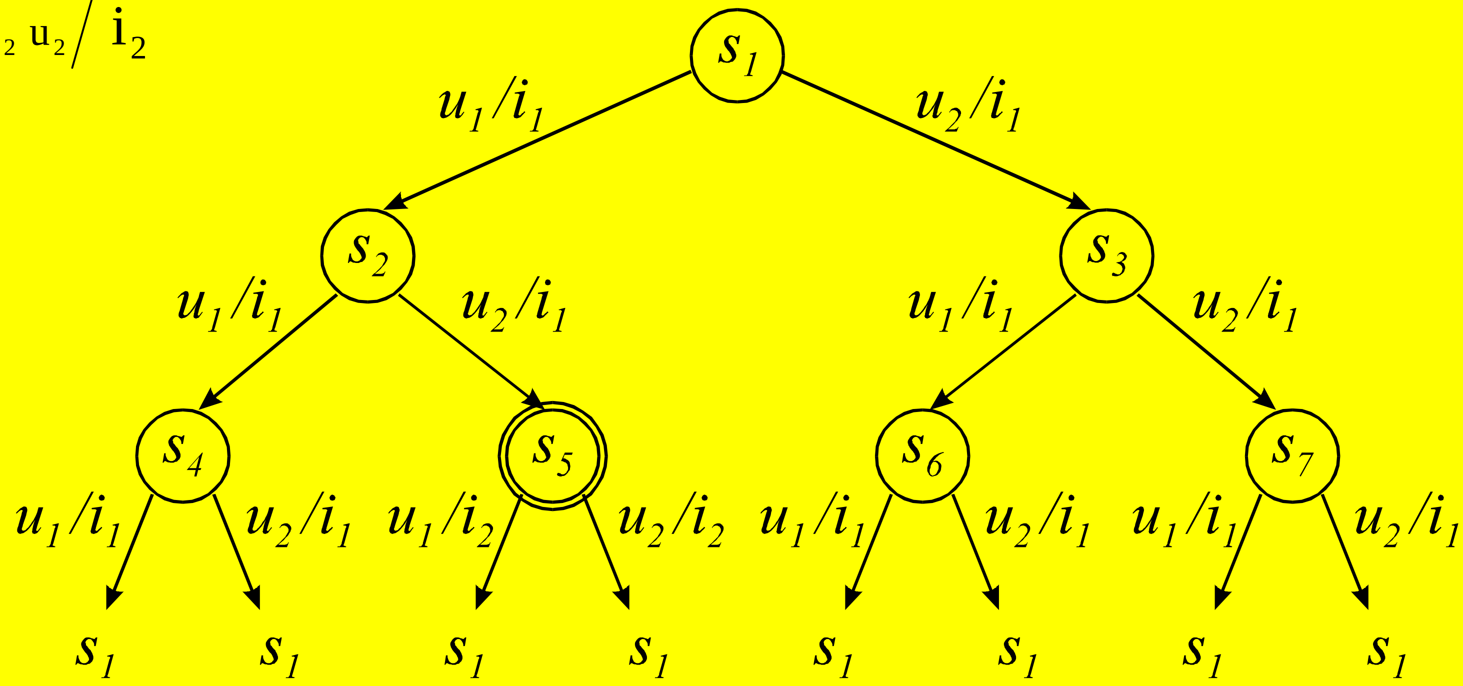
$$\left.\begin{matrix} u_1\, u_2\, u_1 \\ u_1\, u_2\, u_2 \end{matrix}\right/ i_2$$



ZADAVANJE KORAK PO KORAK

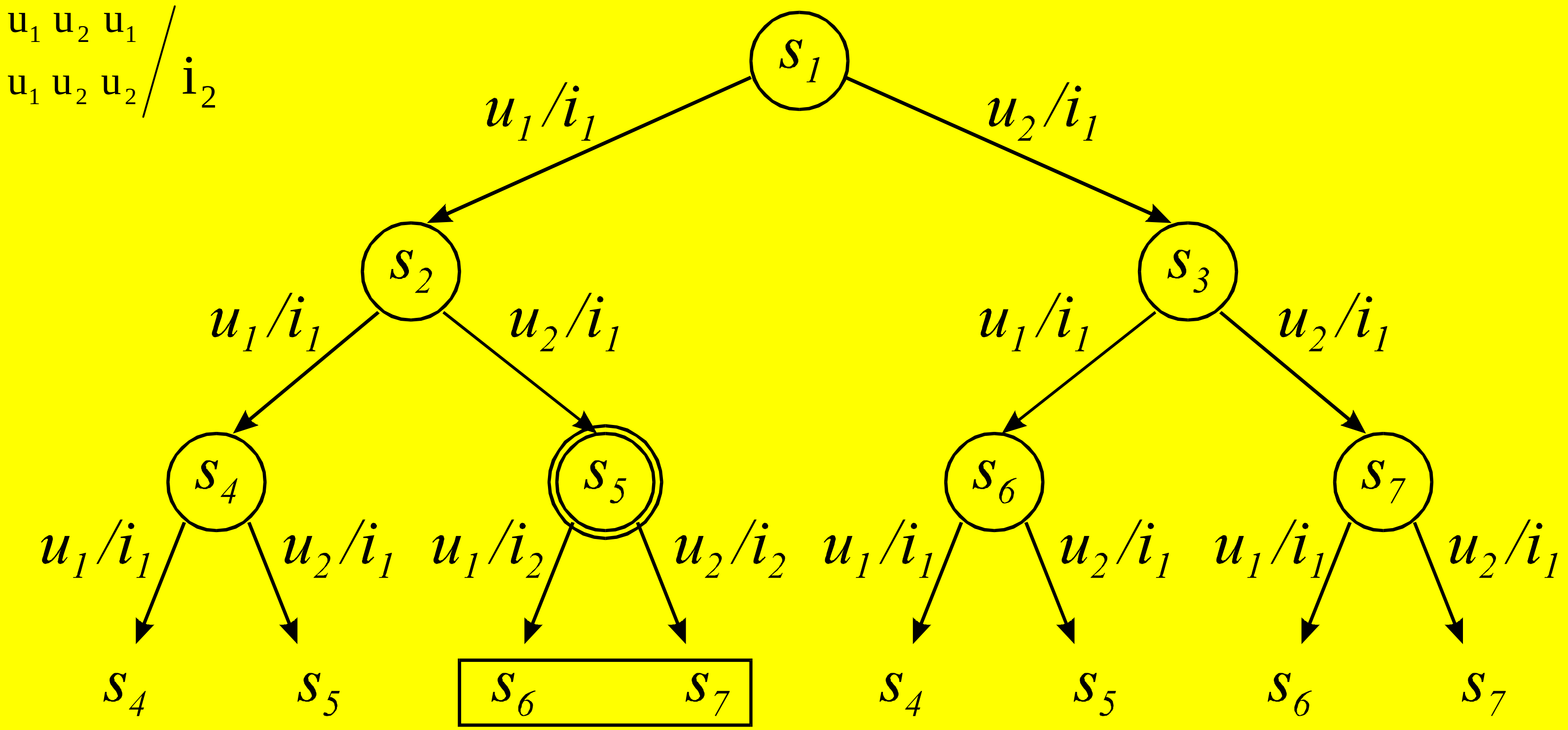
- MEALY automat, sekvenca SA strukturom:

$$\left.\begin{matrix} u_1\, u_2\, u_1 \\ u_1\, u_2\, u_2 \end{matrix}\right/ i_2$$



ZADAVANJE KORAK PO KORAK

- MEALY automat, sekvenca BEZ strukture:



prijelaz akceptorskih stanja ovisi o preklapanju sekvenci

ZADAVANJE KORAK PO KORAK

- PRIMITIVNA (početna) TABLICA
  MEALY automat, sekvenca BEZ strukture:

$\dfrac{u_1 u_2 u_1}{u_1 u_2 u_2} / i_2$

	s^{n+1} u_1	u_2	i^n u_1	u_2
s_1	s_2	3	1	1
s_2	4	5	1	1
s_3	6	7	1	1
s_4	4	5	1	1
s_5	6	7	2	2
s_6	4	5	1	1
s_7	6	7	1	1

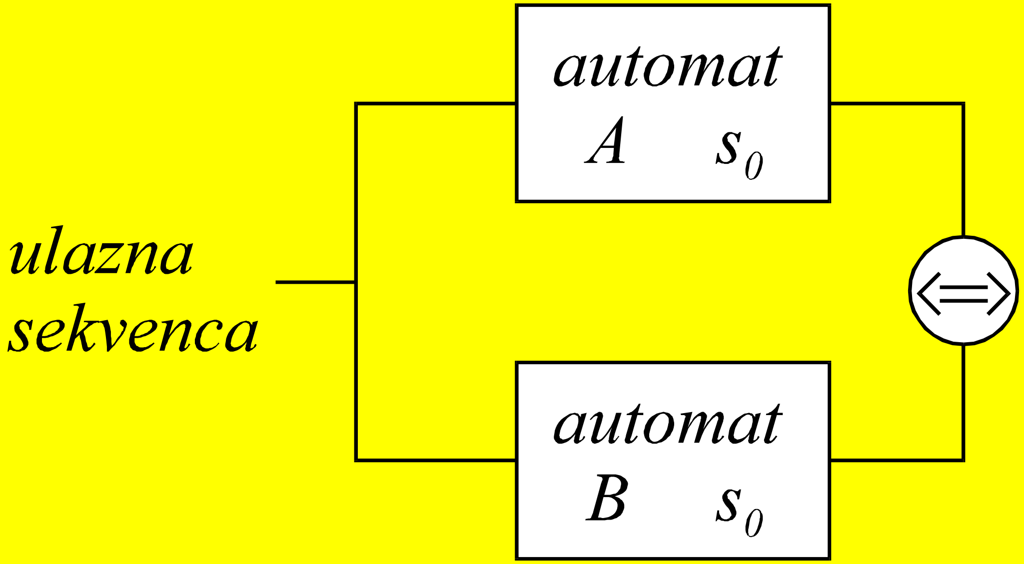
preklapanje
dozvoljeno

EKVIVALENTNOST AUTOMATA

- DVA AUTOMATA
  - isti automati: ista funkcija, isti po strukturi
  - različiti automati: različite funkcija i struktura
  - ekvivalentni automati:
    ista funkcija a različita struktura!
- MINIMIZACIJA AUTOMATA
  - postoje ekvivalentni automati
  - pronaći među njima minimalan
  - ekvivalentni se razlikuju po **skupu stanja**
  - ne mogu se razlikovati po skupovima
    ulaznih i izlaznih simbola

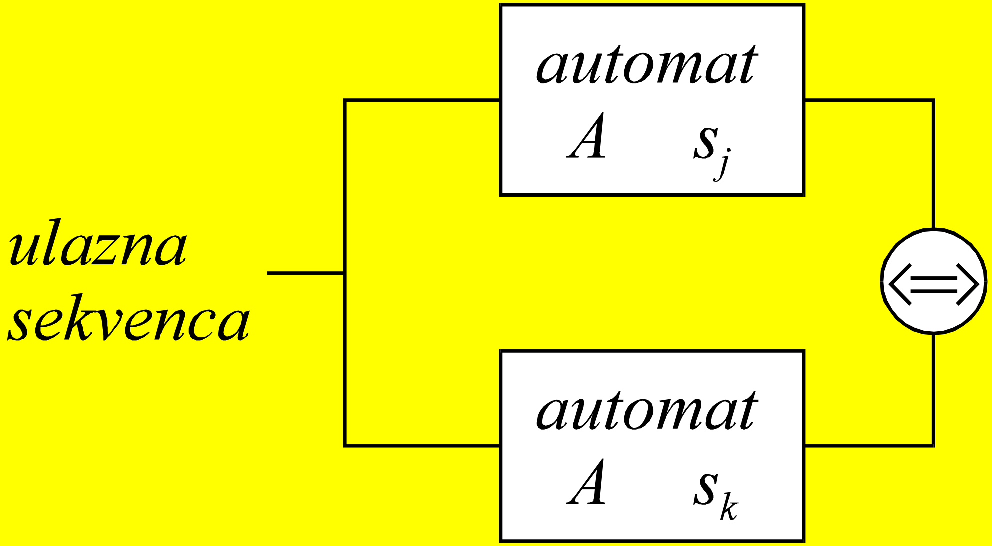
MINIMIZACIJA AUTOMATA

- DEFINICIJA EKVIVALENT. AUTOMATA
  dva automata su ekvivalentna ako
  - počnu raditi iz početnog stanja
  - za istu **proizvoljnu** ulaznu sekvencu
  - dadu istu izlaznu sekvencu



MINIMIZACIJA AUTOMATA

- DEFINICIJA EKVIVALENTNOSTI STANJA
  dva stanja istog automata su ekvivalentna ako
  - automat počne raditi iz jednog ili drugog
  - za istu **proizvoljnu** ulaznu sekvencu
  - dade u oba testa istu izlaznu sekvencu



KRITERIJ EKVIVALENCIJE STANJA

- NUŽAN UVJET EKVIVALENCIJE
  dva stanja istog automata
  mogu biti ekvivalentna ako
  - imaju iste retke u tablici izlaza
  - tada je sigurno prvi simbol izlazne sekvence u oba testa isti

- DOVOLJAN UVJET EKVIVALENCIJE
  dva stanja istog automata **jesu** ekvivalentna ako
  - je zadovoljen nužan uvjet
  - imaju iste retke u tablici prijelaza
  - tada je sigurno i ostatak izlazne sekvence isti

METODE MINIMIZACIJE

- MINIMIZACIJA PRIMITIVNE TABLICE
  - pretpostavlja da su sva stanja neekvivalentna
- HUFMANN-MEALY ALGORITAM
  - pretpostavlja da su sva stanja ekvivalentna
- PAUL-UNGEROV ALGORITAM
  - polazi od implikacije
  - koristi tablicu implikanata

METODE MINIMIZACIJE

- MINIMIZACIJA PRIMITIVNE TABLICE
  - provodi se nad primitivnom tablicom
  - neposredno primjenjuje kriterij ekvivalencije
  - tražimo stanja sa istim recima
    u tablici prijelaza i izlaza
  - prekrižimo sva ekvivalentna stanja osim jednog
  - zamijenimo oznake prekriženih stanja
    s oznakom stanja koje nije prekriženo
  - ponovimo postupak radi otkrivanja
    novih grupa ekvivalentnih stanja

METODE MINIMIZACIJE

- MINIMIZACIJA PRIMITIVNE TABLICE
  za gornji primjer

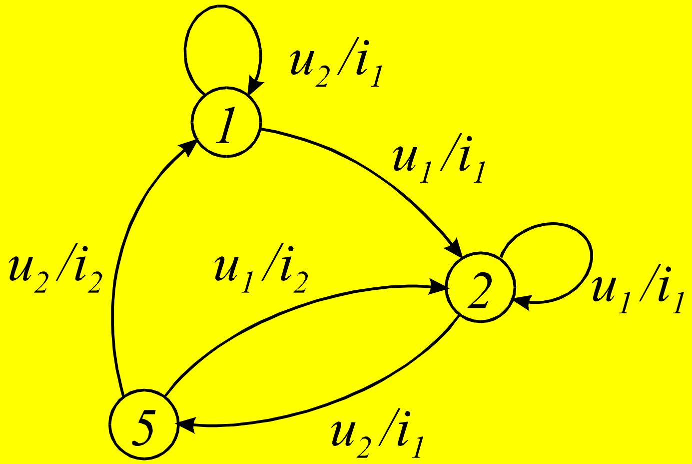
s^n	s^{n+1} u_1	s^{n+1} u_2	i^n u_1	i^n u_2
1	2	3	1	1
2	2 ~~4~~	5	1	1
3	2 ~~6~~	~~7~~ 3	1	1
~~4~~	~~4~~	~~5~~	~~1~~	~~1~~
5	2 ~~6~~	~~7~~ 3	2	2
~~6~~	~~4~~	~~5~~	~~1~~	~~1~~
~~7~~	~~6~~	~~7~~	~~1~~	~~1~~

s^n	s^{n+1} u_1	s^{n+1} u_2	i^n u_1	i^n u_2
1	2	~~3~~ 1	1	1
2	2 ~~4~~	5	1	1
~~3~~	2 ~~6~~	~~7~~ 3	~~1~~	~~1~~
~~4~~	~~4~~	~~5~~	~~1~~	~~1~~
5	2 ~~6~~	~~7~~ ~~3~~ 1	2	2
~~6~~	~~4~~	~~5~~	~~1~~	~~1~~
~~7~~	~~6~~	~~7~~	~~1~~	~~1~~

METODE MINIMIZACIJE

- MINIMIZACIJA PRIMITIVNE TABLICE
  ispisujemo tablicu automata i crtamo graf:

	s^{n+1}		i^n	
s^n	u_1	u_2	u_1	u_2
1	2	1	1	1
2	2	5	1	1
5	2	1	2	2



METODE MINIMIZACIJE

- MINIMIZACIJA PRIMITIVNE TABLICE

mane minimizacije primitivne tablice:

- ne otkriva sve ekvivalentnosti

- na primjer:

s	1	2	3	i
1	1	2	3	1
2	2	1	3	1
3				

stanja 1 i 2 su zapravo ekvivalentna!

METODE MINIMIZACIJE

- HUFMANN-MEALY ALGORITAM
  - pretpostavlja da su stanja za koja je zadovoljen **nužan uvjet** ekvivalentna
  - **klasa** je podskup stanja iz skupa S za koja pretpostavljamo da su ekvivalentna
  - klasa je **zatvorena** ako sva stanja klase
    * imaju iste prijelaze u klase
    * sadrži samo jedno stanje
  - kada su sve klase zatvorene, svaku zamijenimo s jednim stanjem

METODE MINIMIZACIJE

- HUFMANN-MEALY ALGORITAM
  1. Definiramo primarne klase
     na osnovu nužnog uvjeta ekvivalentnosti
  2. Za sva stanja unutar klase
     odredimo prijelaze u klase
  3. Kontroliramo zatvorenost klasa
     (isti prijelazi u klase ili samo jedno stanje)
  4. Razbijamo otvorene klase i ponavljamo (2)
     (prema istim prijelazima u klase)
  5. Ispisujemo tablicu minimalnog automata

METODE MINIMIZACIJE

- HUFMANN-MEALY ALGORITAM
  - primjer:

	s^{n+1}		i^n	
	u_1	u_2	u_1	u_2
s_1	s_2	3	1	1
s_2	4	5	1	1
s_3	6	7	1	1
s_4	4	5	1	1
s_5	6	7	2	2
s_6	4	5	1	1
s_7	6	7	1	1

A (1 1)	u_1	u_2
1	A	A
2	A	B
3	A	A
4	A	B
6	A	B
7	A	A

B (2 2)	u_1	u_2
5	A	A

zatvorena

METODE MINIMIZACIJE

- HUFMANN-MEALY ALGORITAM
  - razbijemo klasu A:

$A_1\ (1\ 1)$	u_1	u_2
1	A_2	A_1
3	A_2	A_1
7	A_2	A_1

$B\ (2\ 2)$	u_1	u_2
5	A_2	A_1

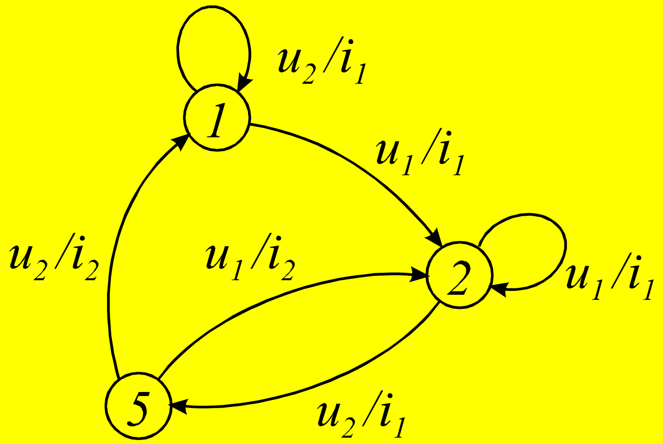
$A_2\ (1\ 1)$	u_1	u_2
2	A_2	B
4	A_2	B
6	A_2	B

zatvorena

METODE MINIMIZACIJE

- HUFMANN-MEALY ALGORITAM
  - crtamo tablicu i graf minimalnog automata:

s^n	s^{n+1} u_1	s^{n+1} u_2	i^n u_1	i^n u_2
1	2	1	1	1
2	2	5	1	1
5	2	1	2	2



$\Rightarrow$ isto kao prije!

METODE MINIMIZACIJE

- PAUL-UNGER ALGORITAM

1. Implikacija među skupovima:
   skup S_p impliciran je skupom S_{ri}
   ako za S_{ri} sadrži sva stanja u koja prelaze stanja
   iz S_p za promatrani ulaz u_i
2. Skupova S_{ri} ima “p”, koliko ima ulaznih simbola
3. Ekvivalentnost:
   S_p sadrži ekvivalentna stanja
   * ako je zadovoljen nužan uvjet ekvivalencije
   * ako su svi njemu pripadni S_{ri} ekvivalentni
   (svodi se na zadovoljavanje nužnog uvjeta)

METODE MINIMIZACIJE

- PAUL-UNGER ALGORITAM

S_{R_1} S_{R_2} S_P

s^n	s^{n+1} u_1	s^{n+1} u_2	i^n u_1	i^n u_2
1	2	3	1	1
2	4	5	1	1
3	6	7	1	1
4	4	5	1	1
5	6	7	2	2
6	4	5	1	1
7	6	7	1	1

METODE MINIMIZACIJE

- PAUL-UNGER ALGORITAM

  - skupove S_p formiramo sistematski 2 po 2 stanja
    (ispitujemo ekvivalentnost svakog sa svakim)

  - koristimo tablicu implikanata

  - to je trokutasta matrica bez dijagonale
    * $s_i \Leftrightarrow s_i$, stanje ekvivalentno samo sebi
    * $s_i \Leftrightarrow s_j \Rightarrow s_j \Leftrightarrow s_i$ (“komutativnost”)

  - tablica ima n-1 redaka i stupaca

  - svako mjesto odgovara jednom paru s_i, s_j

  - u mjesta tablice upisujemo implikante,
    a to su skupovi S_{ri}

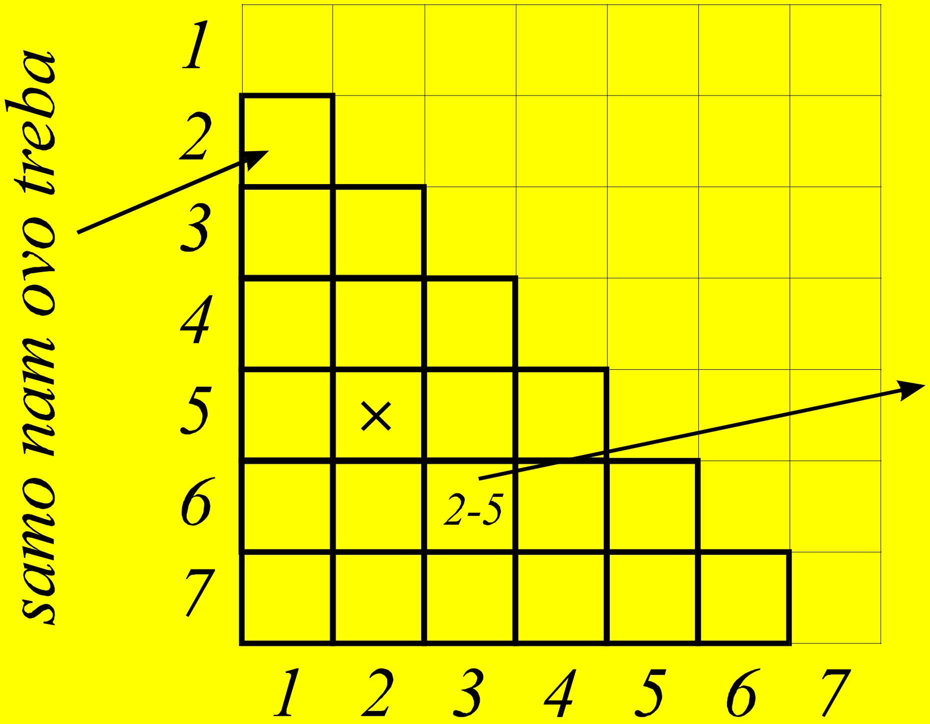
METODE MINIMIZACIJE

- PAUL-UNGER ALGORITAM

  1. Formiramo trokutastu matricu n-1 x n-1
  2. Upišemo implikante
     Sri $\Rightarrow$ zadovoljen nužan, ne i dovoljan uvjet,
     X $\Rightarrow$ nužan uvjet nije zadovoljen (neekv.)
     V $\Rightarrow$ zadovoljeni nužan i dovoljan uvjet (ekv.)
  3. Ispitujemo kontradikcije,
     upisujemo X za otkrivene neekvivalentnosti
  4. Ponavljamo (3) ako ima novih neekvivalentnosti
  5. Ispisujemo tablicu minimalnog automata
     neprekrižena polja znače ekvivalentna stanja

METODE MINIMIZACIJE

- PAUL-UNGER ALGORITAM



$S_P = \{3, 6\}$

$S_{ri} = \{2, 5\}$

METODE MINIMIZACIJE

- PAUL-UNGER ALGORITAM
  - primjer: popunimo tablicu implikanata

	s^{n+1}		i^n	
	u_1	u_2	u_1	u_2
s_1	s_2	3	1	1
s_2	4	5	1	1
s_3	6	7	1	1
s_4	4	5	1	1
s_5	6	7	2	2
s_6	4	5	1	1
s_7	6	7	1	1

2	*2-4* *3-5*					
3	*2-6* *3-7*	*4-6* *5-7*				
4	*2-4* *3-5*	✓	*4-6* *5-7*			
5	×	×	×	×		
6	*2-4* *3-5*	✓	*4-6* *5-7*	✓	×	
7	*2-6* *3-7*	*4-6* *5-7*	✓	*4-6* *5-7*	×	*4-6* *5-7*
	1	*2*	*3*	*4*	*5*	*6*

METODE MINIMIZACIJE

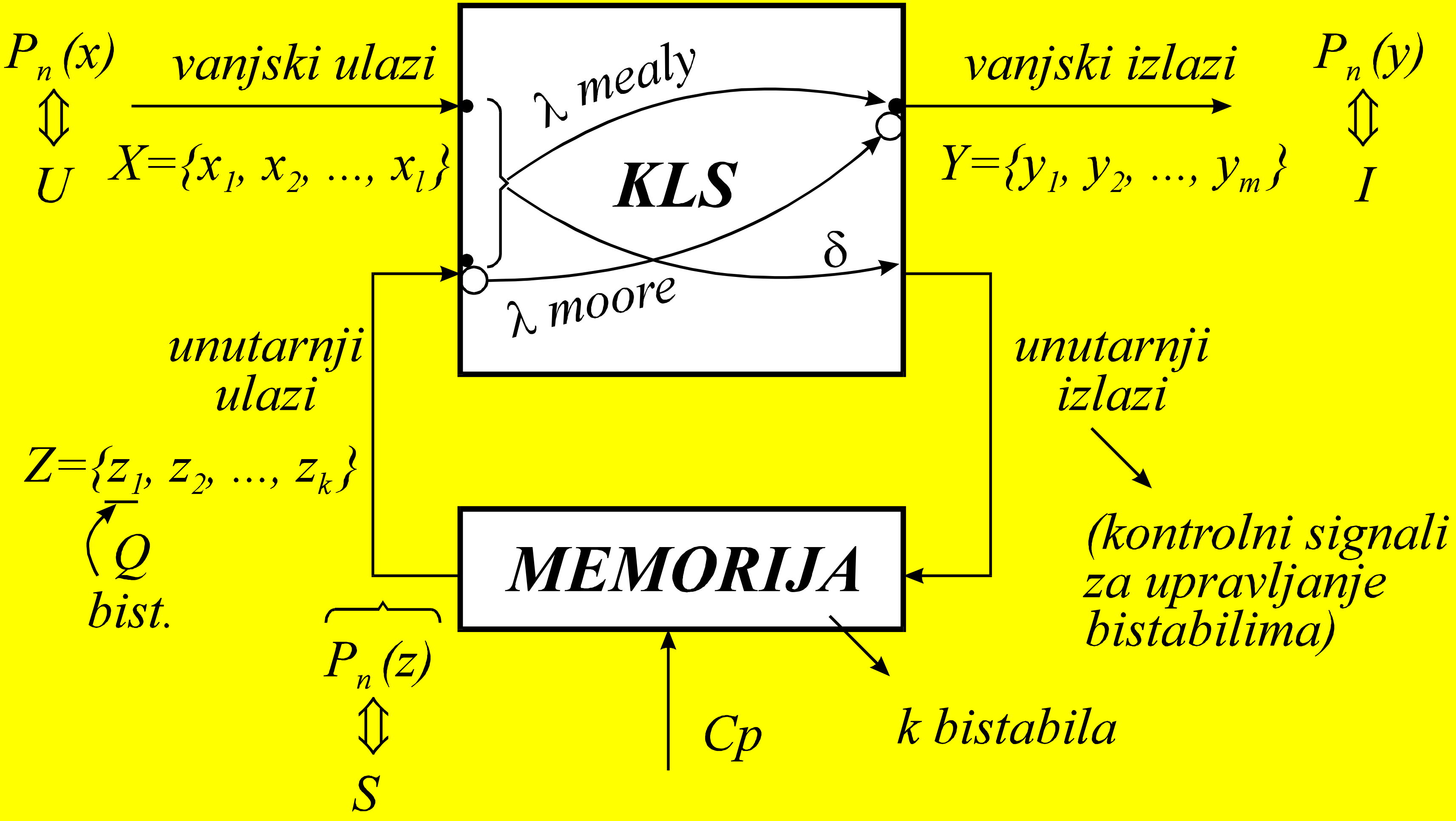
- PAUL-UNGER ALGORITAM
  - provjerimo kontradikcije

2	~~2-4 3-5~~					
3	2-6 3-7	~~4-6 5-7~~				
4	~~2-4 3-5~~	√	~~4-6 5-7~~			
5	×	×	×	×		
6	~~2-4 3-5~~	√	~~4-6 5-7~~	√	×	
7	2-6 3-7	~~4-6 5-7~~	√	~~4-6 5-7~~	×	~~4-6 5-7~~
	1	2	3	4	5	6

	s^{n+1}		i^n	
s^n	u_1	u_2	u_1	u_2
1	2	1	1	1
2	2	5	1	1
5	2	1	2	2

STRUKTURNA SINTEZA

- MODEL REALIZACIJE



STRUKTURNA SINTEZA

- KODIRANJE ULAZA I IZLAZA
  - najčešće ovisno o okolini (izvorištu i odredištu)
- KODIRANJE STANJA
  - unutrašnja stvar automata
  - o kodiranu ovisi kompleksnost strukture
  - nema egzaktnog postupka
  - koristimo strategiju susjednosti
  - za stanja među kojima postoji prijelaz nastojimo dodijeliti susjedne kompleksije
  - kodiramo Veitchevim dijagramom

STRUKTURNA SINTEZA

- PRIMJER KODIRANJA

	s^{n+1}		i^{n}	
	u_1	u_2	u_1	u_2
1	2	1	1	1
2	2	5	1	1
5	2	1	2	2

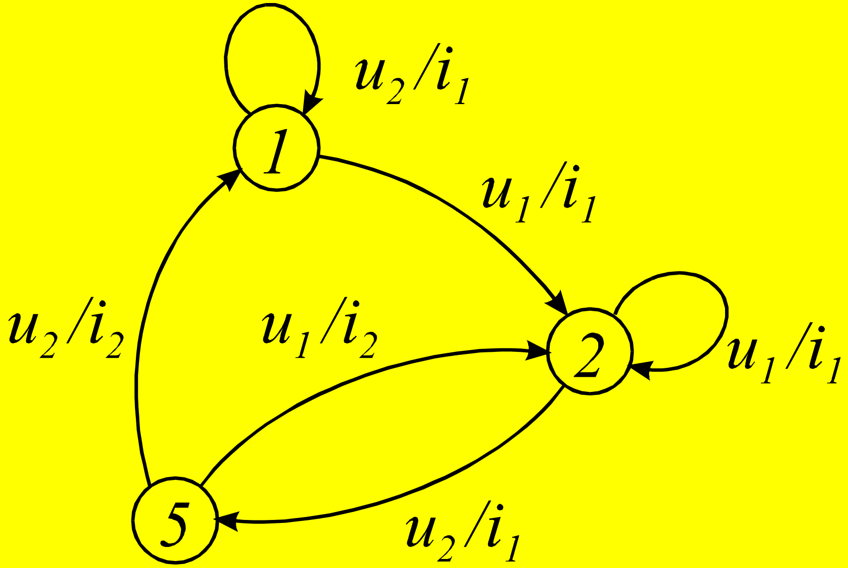
- KODIRANJE ULAZA I IZLAZA

U	x_1
u_1	0
u_2	1

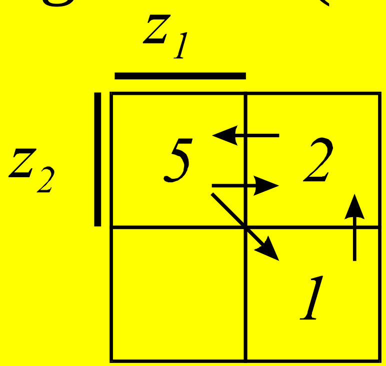
I	y_1
i_1	0
i_2	1

STRUKTURNA SINTEZA

- KODIRANJE STANJA
  prijelazi među stanjima očiti su na grafu automata



kodiramo Veitchevim dijagramom (susjednost)



početno stanje: kompleksija 0 (reset)

STRUKTURNA SINTEZA

- UPISIVANJE KODOVA U TABLICU

  - standardna tablica je dvodimenzionalna:

	s^{n+1}	i^{n}
	$u_1, u_2, \ldots, u_p$	$u_1, u_2, \ldots, u_p$
s_1		
s_2		
$\vdots$		
s_n		

STRUKTURNA SINTEZA

- UPISIVANJE KODOVA U TABLICU
  - predimo na jednodimenzionalnu tablicu:

S	U	s^{n+1}	i^n
s_1	u_1		
	u_2		
	$\vdots$		
	u_p		
s_2	u_1		
	u_2		
	$\vdots$		
	u_p		
$\vdots$			

STRUKTURNA SINTEZA

- UPISIVANJE KODOVA U TABLICU
  - kodove upisujemo u jednodimenzionalnu tablicu:
    (tako da kodne riječi budu u binarnom nizu)

s^n				u^n				s^{n+1}				i^n			
z_1	z_2	…	z_k	x_1	x_2	…	x_ℓ	z_1	z_2	…	z_k	y_1	y_2	…	y_m
0	0	…	0	0	0	…	0								
0	0	…	0	0	0	…	1								
		⋮				⋮		0	0	…	0	0	0	…	0
1	1	…	1	1	1	…	0								
1	1	…	1	1	1	…	1								

STRUKTURNA SINTEZA

- UPISIVANJE KODOVA U TABLICU
  - za gornji primjer:

	s^{n+1}		i^{n}	
	u_1	u_2	u_1	u_2
1	2	1	1	1
2	2	5	1	1
5	2	1	2	2

s	z_1	z_2
1	0	0
2	0	1
5	1	1

U	x_1
u_1	0
u_2	1

I	y_1
i_1	0
i_2	1

	z_1	z_2	x_1	z_1	z_2	y
s_1	0	0	0	0	1	0
			1	0	0	0
s_2	0	1	0	0	1	0
			1	1	1	0
R	1	0	0	R		R
			1			
s_5	1	1	0	0	1	1
			1	0	0	1

prepoznamo: tablice prijelaza bistabila i tablice istine!

STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA

  - bistabilima i logičkim vratima:

  * koristimo metode minimizacije BF
  * realiziramo NI i NILI vratima
  * koristimo metode sinteze općih bistabila

STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA

  primjer: RS bistabilima i NI logičkim vratima:

  - metoda rekonstrukcije

z_1	z_2	x_1	z_1	z_2	y	R_1	S_1	R_2	S_2
0	0	0	0	1	0	r	0	0	1
		1	0	0	0	r	0	r	0
0	1	0	0	1	0	r	0	0	r
		1	1	1	0	0	1	0	r
1	0	0	R		R	r	r	r	r
		1				r	r	r	r
1	1	0	0	1	1	1	0	0	r
		1	0	0	1	1	0	1	0

STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA

  primjer: RS bistabilima i NI logičkim vratima:

  - minimizacija R_1, S_1

S_1: (z_1 above columns 1–2, z_2 beside row 1, x_1 below columns 2–3)

		1	
r	*r*		

R_1: (z_1 above columns 1–2, z_2 beside row 1, x_1 below columns 2–3)

1	*1*		*r*
r	*r*	*r*	*r*

$$S_1 = \overline{\overline{\bar{z}_1 \, z_2 \, x_1}}$$

$$R_1 = z_1$$

STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA

  primjer: RS bistabilima i NI logičkim vratima:

  \- minimizacija R_2, S_2

S_2:

	z_1	z_1		
z_2	*r*		*r*	*r*
	r	*r*		*1*
		x_1	x_1	

R_2:

	z_1	z_1		
z_2		*1*		
	r	*r*	*r*	
		x_1	x_1	

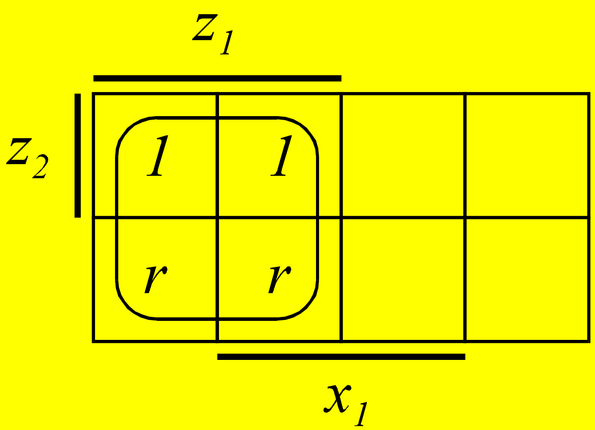
$$S_2 = \bar{x}_1$$

$$R_2 = \overline{\overline{z_1 \, x_1}}$$

STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA
  primjer: RS bistabilima i NI logičkim vratima:
  - minimizacija y

y:

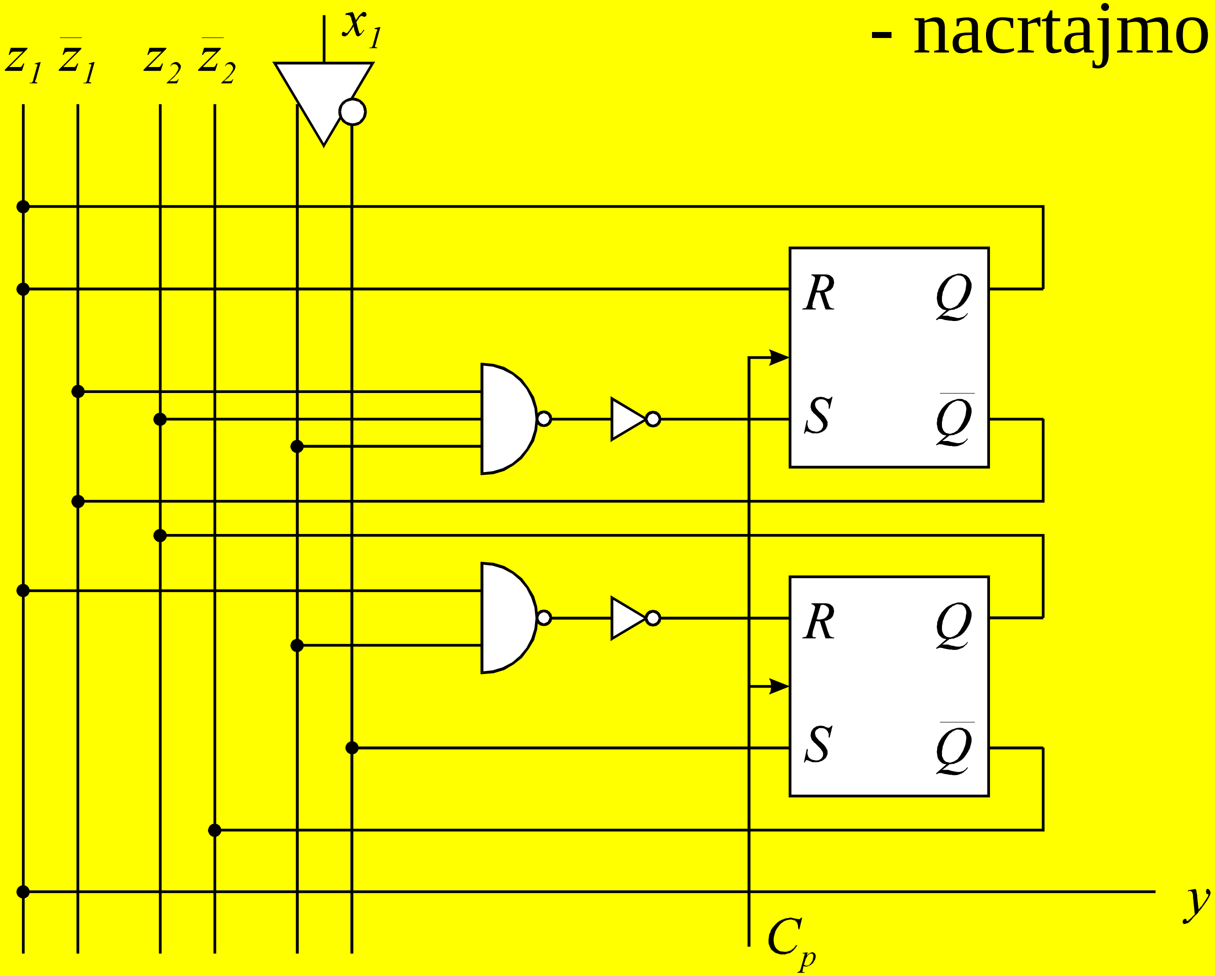


$$y = z_1$$

STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA

  - nacrtajmo shemu



STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA

  - multipleksersko demultiplekserskom strukturom
    i D bistabilima (registar), MDD
      * odrediti broj multipleksera M=m+k
      * odrediti veličinu multipleksera i demultipleksera
        (po kriteriju kvadratičnosti matrice)
      * koristimo metodu za D bistabil:

$$Q^{n+1} = D^n$$

      * tablica prijelaza numerički identična tablici istine
      * popunjavamo matricu

STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA
  primjer: MDD struktura
  - kvadratičnost matrice

z_1	z_2	x_1	z_1	z_2	y
0	0	0	0	1	0
		1	0	0	0
0	1	0	0	1	0
		1	1	1	0
1	0	0	R		R
		1			
1	1	0	0	1	1
		1	0	0	1

$$2^d \approx M \cdot 2^m$$

$$d + m = k + \ell$$

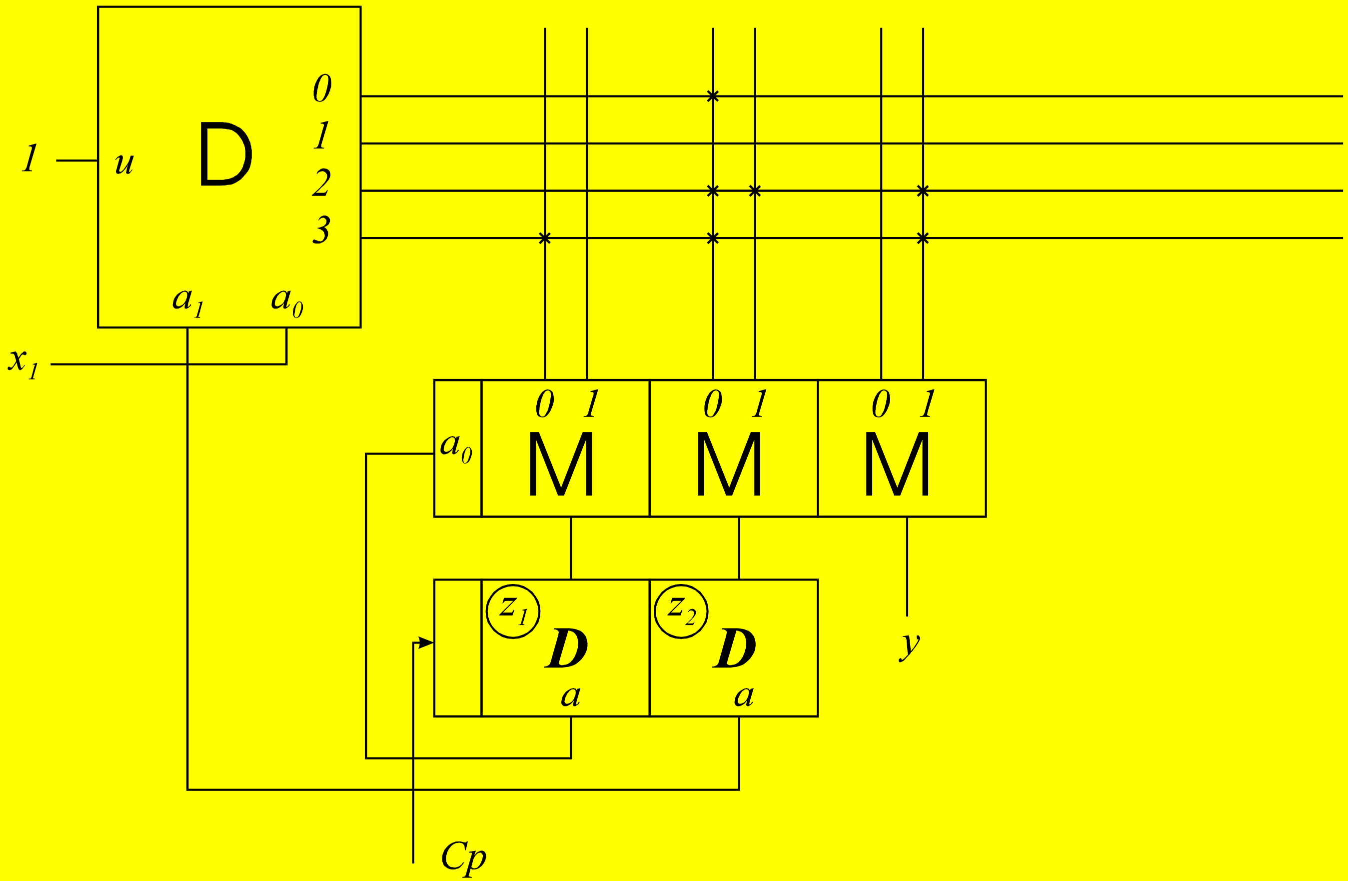
$$M = 3$$

$$d + m = 3$$

d	m	2^d	$M \cdot 2^m$
1	2	2	12
2	1	4	6

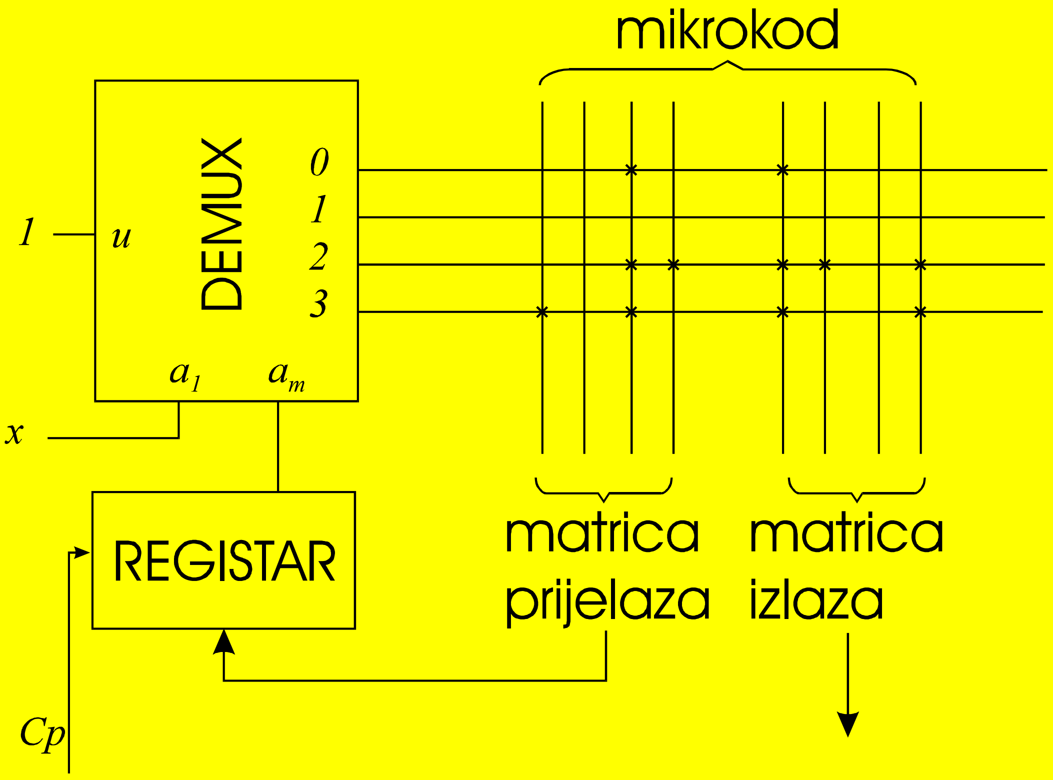
STRUKTURNA SINTEZA

- REALIZACIJA AUTOMATA
  - shema, pazimo na raspored varijabli



3.5. PROGRAMABILNI AUTOMATI I ALGORITMI

- WILKIESOV MODEL (MDD struktura)



sadržaj matrice je **mikroprogram!**

AUTOMAT I PROCESOR RAČUNALA

- ULOGA AUTOMATA
  - upravlja izvršenjem naredbi računala
    (upravlja radom sabirnice i ALU jedinicom)
- IZVEDBA AUTOMATA
  - mikroprogramirani
    * jedna instrukcija računala
      više mikroinstrukcija
    * sporiji, lakše konstruira i modificira
  - klasični (bistabili, logička vrata)
    * jedna instrukcija računala više stanja
    * brži, teže se konstruira i modificira

ALGORITMI

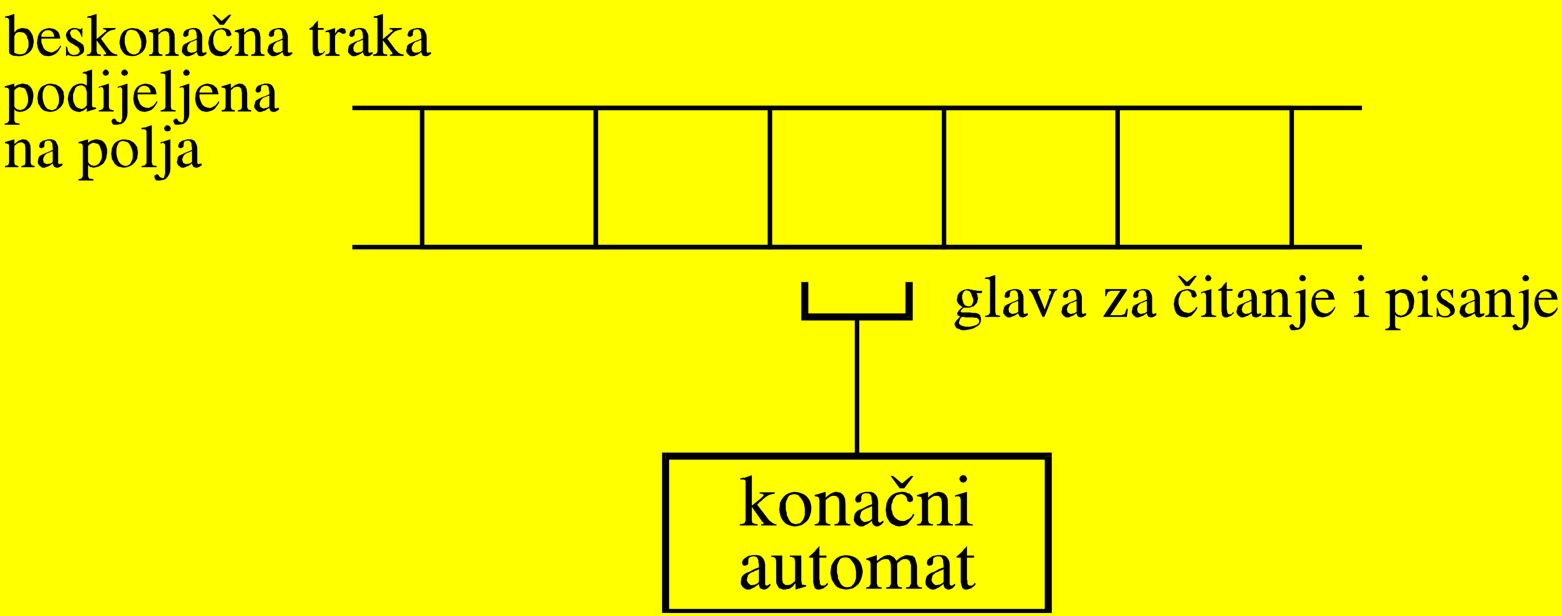
- ALGORITAM
  - skup transformacija ili instrukcija
  - u konačnom broju koraka
  - postiže rješenje problema
  - iz neke klase problema

- RJEŠENJE
  - za konkretan problem nižemo instrukcije
  - odgovara pisanju programa

- ALGORITAM i AUTOMAT
  - algoritmu odgovara automat - stroj
  - npr. Turingov stroj

TURINGOV STROJ

- Alan Turing
  - najsnažniji stroj
  - raspolaže beskonačnom trakom
  - automat upravlja glavom za čitanje/pisanje



TURINGOV STROJ

- ZADAVANJE TURINGOVOG STROJA
  - rad automata definiramo petorkama
  - kao da se radi o mikroprogramiranom automatu
  - ulazni i izlazni alfabet su isti (T)
  - dodana je naredba za pomak glave
    (lijevo, desno ili stop)

$$\left\langle S_i, T_i, T_j, \left|\begin{matrix} S \\ L \\ R \end{matrix}\right|, S_j \right\rangle$$

TURINGOV STROJ

- VRSTE TURINGOVOG STROJA
  - dozvolimo rad bez pomaka glave (N automat)
  - istovremeno ili upis ili pomak (F automat)
  - traka konačna s jedne strane
  - automat s dvije trake

    svi su jednako sposobni!

TURINGOV STROJ

- PRIMJER
  množenje binarnog broja s 2 (pomak u lijevo)

b 0 1 0 0 0 1 0 1 b

$< S_{00}, 0, 0, L, S_{00} >$	$< S_{01}, 0, 1, L, S_{00} >$
$< S_{00}, 1, 0, L, S_{01} >$	$< S_{01}, 1, 1, L, S_{01} >$
$< S_{00}, b, b, S, S_{00} >$	$< S_{01}, b, 1, S, S_{00} >$