

F€ËÛ^\\ ç^} &ã\ ã\ [|] [çãÇD

F€ËÛ^\\ ç^} &ã\ ã\ [|] [çãÇD

FFËÛç} åæå} ã^\\ ç^} &ã\ ã\ [å~ |ãÇD

FFËÛç} åæå} ã^\\ ç^} &ã\ ã\ [å~ |ãÇD

ÁÁÁÁFGEÛ~ ^|bÁÁç} æ[* } [{ Á\ [|ã [{ ÁÇDÁ

ÁÁÁÁFGEÛ~ ^|bÁÁç} æ[* } [{ Á\ [|ã [{ ÁÇDÁ

Zæåæ&ã



10. Sekvencijski sklopovi (1)

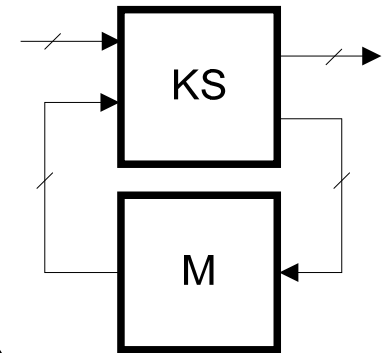


Sadržaj predavanja

- **konceptualizacija sekvencijskih sklopova**
 - sinkroni sekvencijski sklopovi
 - kanonski modeli
- projektiranje sekvencijskih sklopova
- izvedbe sekvencijskih sklopova

Sekvencijski sklopovi

- sekvencijski sklopovi:
 - digitalni sklopovi koji imaju sposobnost pamćenja
 - izlaz je funkcija:
 - trenutnog stanja ulaza
 - trenutnog *unutarnjeg* stanja sklopa
~ postoji *memorija*
 - interpretacija memorije (npr. računala)
~ pamćenje memorijskih riječi (= višebitni podaci)
→ *registri*





Sekvencijski sklopovi

- definicije:
 - n bistabila $\rightarrow 2^n$ stanja
 \sim *strojevi stanja* (engl. state machines)
 - stanja je konačno mnogo (2^n)
 \sim strojevi *s konačnim brojem stanja*
(engl. finite state machines)
 - slijed operacija u sekvencijskom sklopu
 \sim "ugrađeni" algoritam:
algoritamski stroj stanja
 - operacije se obavljaju bez čovjekove pomoći
 \sim *automat*:
digitalni automat, konačni automat

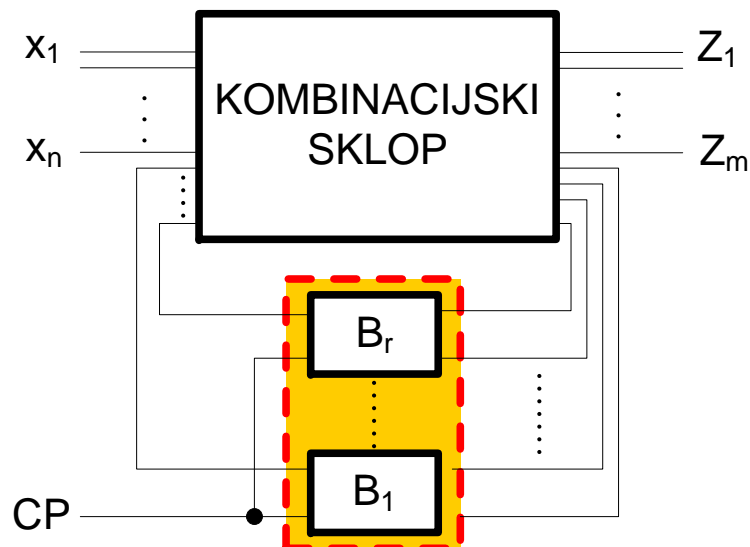
Sinkroni sekvencijski sklopovi

- ograničenje na *sinkrone* sekvencijske sklopove!
~ rad (promjena stanja) sinkroniziran s impulsima CP
 - značajno lakši postupak
~ sinkronizacija:
 - na *globalni* CP sustava
 - u odnosu na *najsporiju* stazu/element sustava
~ *sporije* od asinkronih sekvencijskih sustava
 - vrijeme je *diskretizirano*
~ *projektiranje svedeno na kombinacijsko* određivanje:
 - slijedećeg stanja sklopa
 - izlaza sklopa

na temelju *sadašnjeg* stanja sklopa i narinutih ulaza

Sinkroni sekvencijski sklopovi

- općenito strukturiranje sinkronog sekvencijskog sklopa ~ *kanonski* oblik:
 - kombinacijski (pod)sklop
 - memorija s *upravljanim* (= *sinkronim*) bistabilima ~ *registar* = (pod)sklop za pamćenje *više-bitnih* podataka



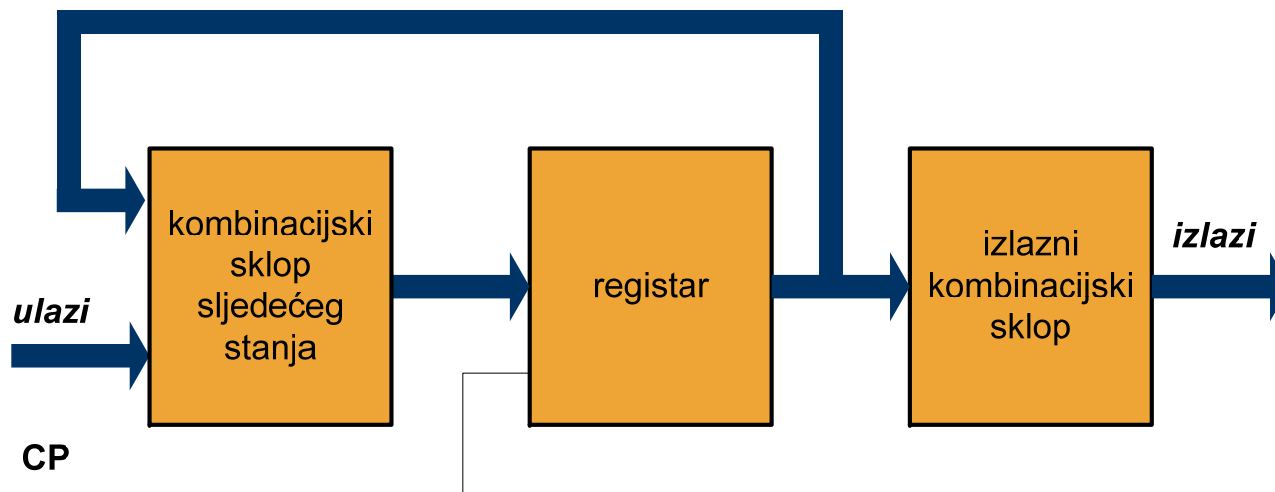


Sinkroni sekvencijski sklopovi

- nekoliko modela opće strukture sinkronog sekvencijskog sklopa:
 - Mooreov model
 - Mealyjev model
 - mješoviti model

Kanonski modeli

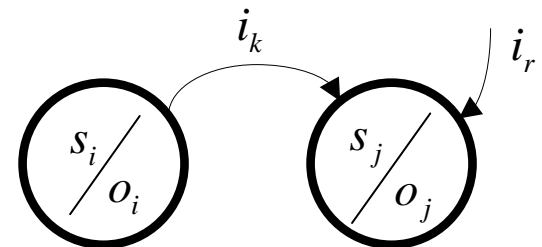
- *Mooreov model* : "automat stanja"
~ izlaz ovisi *samo o* unutarnjem stanju



$$A = \langle I, O, S, \delta, \mu \rangle$$

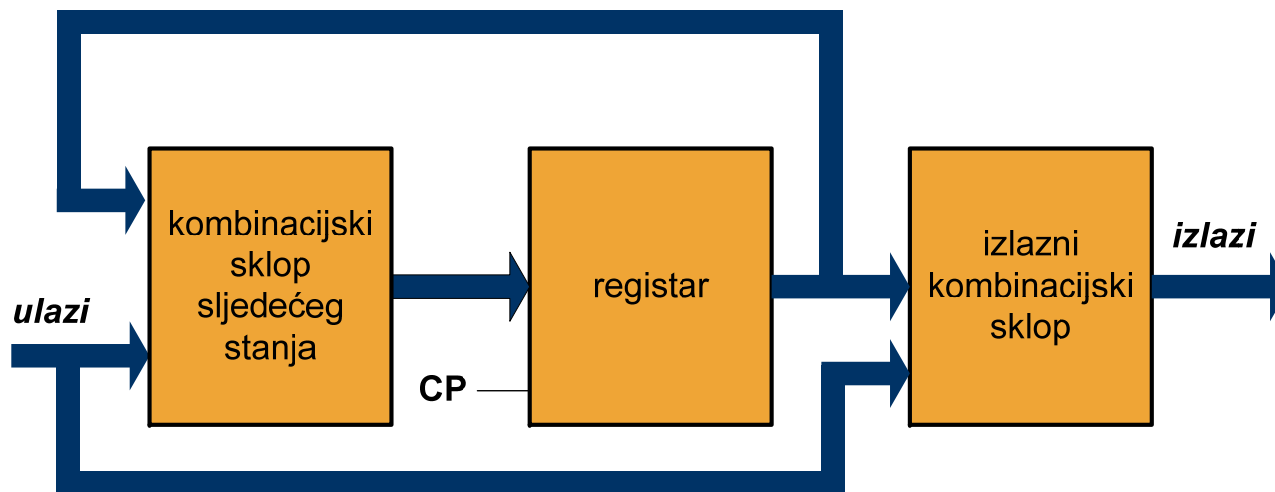
$$\delta: S \times I \rightarrow S$$

$$\mu: S \rightarrow O$$



Kanonski modeli

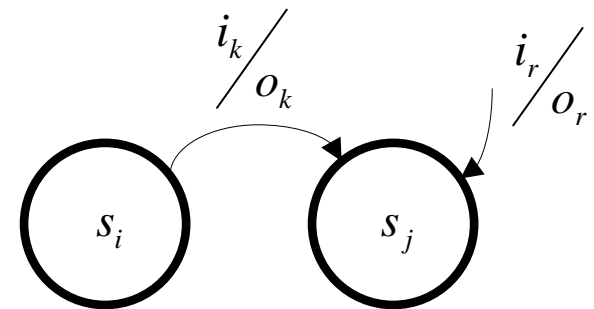
- *Mealyjev model* : "automat prijelaza"
~ izlaz ovisi o unutarnjem stanju i o ulazu



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

$$\delta: S \times I \rightarrow S$$

$$\lambda: S \times I \rightarrow O$$



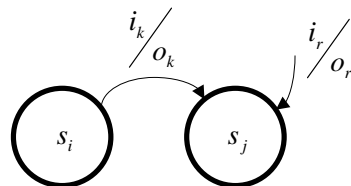
Kanonski modeli

- ekvivalencija Mooreovog i Mealyjevog automata:

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

$$\delta: S \times I \rightarrow S$$

$$\lambda: S \times I \rightarrow O$$



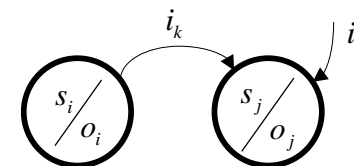
$$q^{n+1} = \delta(q^n, x^n)$$

$$z^n = \lambda(q^n, x^n)$$

$$A = \langle I, O, S, \delta, \mu \rangle$$

$$\delta: S \times I \rightarrow S$$

$$\mu: S \rightarrow O$$



$$q^{n+1} = \delta(q^n, x^n)$$

$$z^n = \mu(q^n)$$

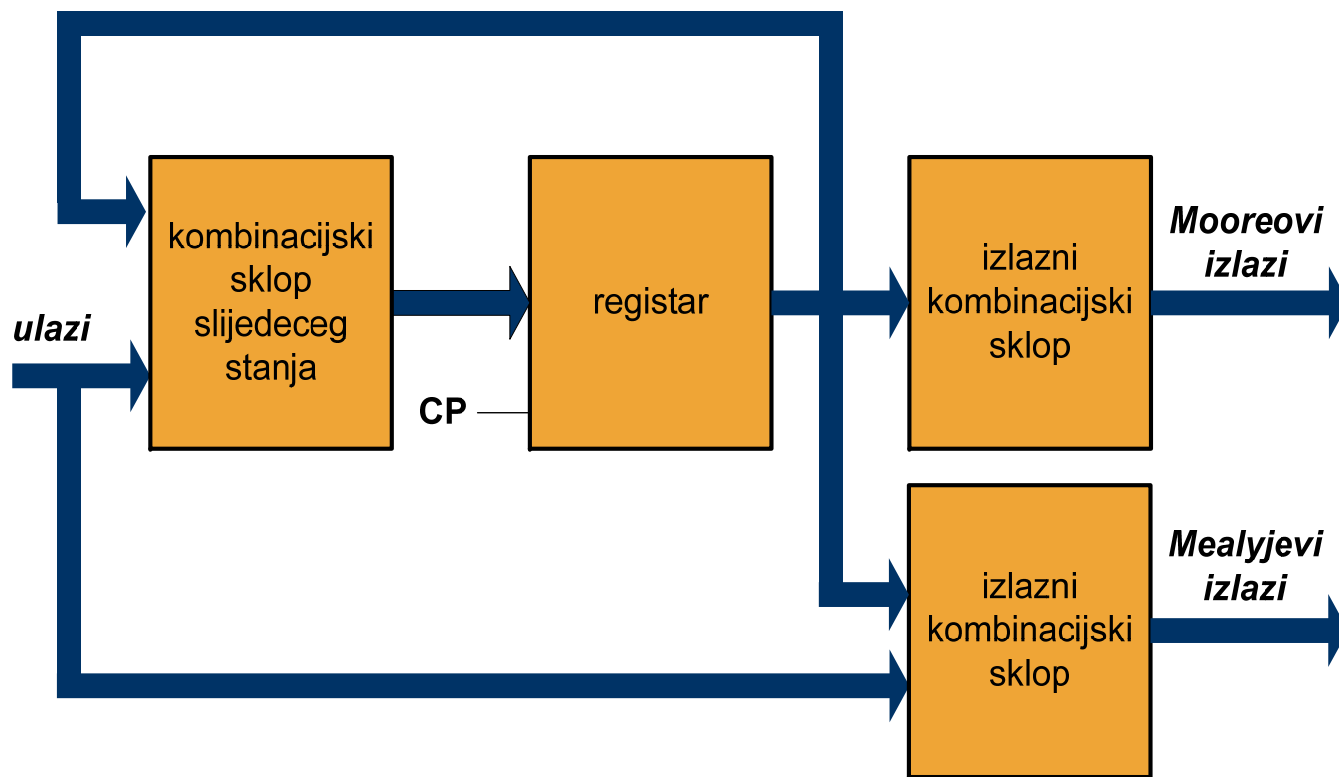
$$= \mu(\delta(q^{n-1}, x^{n-1}))$$

$$= \mu'(q^{n-1}, x^{n-1})$$

- izlaz Mooreovog automata ovisi o *prethodnom* unutarnjem stanju i ulazu!
- moguć prijelaz iz jednog u drugi

Kanonski modeli

- *mješoviti model*
~ izdvojeni izlazi Mooreovog i Mealyjevog modela





Sadržaj predavanja

- konceptualizacija sekvencijskih sklopova
- **projektiranje sekvencijskih sklopova**
 - minimiziranje memorije
 - kodiranje stanja
 - ostvarivanje sekvencijskog sklopa
- izvedbe sekvencijskih sklopova

Projektiranje sekvencijskih sklopova

- neformalni opis postupka (1):
 1. specifikacija sekvencijskog sklopa
~ ulazni jezik: verbalno, algoritamski, ...
 2. izrada dijagrama stanja ili tablice stanja:
~ m : broj stanja, n : broj bistabila $2^{n-1} < m \leq 2^n$
 3. minimiziranje broja stanja
~ smanjenje broja bistabila → minimizacija memorije!
 4. kodiranje stanja
~ dodjela binarne kodne riječi pojedinom stanju:
 - prikladno pridruživanje
~ minimiziranje kombinacijskog (pod)sklopa
 - težak kombinatorni problem



Projektiranje sekvencijskih sklopova

- neformalni opis postupka (2):
 5. izbor tipa bistabila:
 - dobivanje *ulaznih* jednažbi bistabila
~ uzbuda potrebna za odgovarajući prijelaz
(→ generiranje slijedećeg stanja)
 - dobivanje *izlaznih* jednažbi sklopa
 - minimizacija kombinacijskog (pod)sklopa
 6. formalni zapis sekvencijskog sklopa
npr. *logička shema*
~ izbor tehnologije ostvarenja (SIC, ASIC, ...)



Minimiziranje memorije

- standardni pristup
 - ~ *Huffman-Mealyjeva metoda*:
 - za *potpuno specificirane* sklopove
 - ~ \forall unutarnje stanje definirano slijedeće unutarnje stanje + izlaz
 - minimizacija broja bistabila
 - ~ redukcija broja unutarnjih stanja nalaženjem *ekvivalentnih* (nerazlučivih) stanja
 - *ekvivalentna* stanja:
 - ~ ona iz kojih se *istom* pobudom (ulazni niz simbola) dobiva *isti* izlaz (izlazni niz simbola)



Minimiziranje memorije

- ideja Huffman-Mealyjeve metode:
 - klasa ekvivalentnih stanja zamjenjuje ta stanja
~ automat s *reduciranim* brojem unutarnjih stanja
⇒ *minimizirana* memorija
 - početni (ne-minimalni) i konačni (minimalni) automat
~ *ekvivalentni* s obzirom na *izvana opazivo ponašanje*
(engl. externally observable behavior):
jednaki odziv na jednaku pobudu



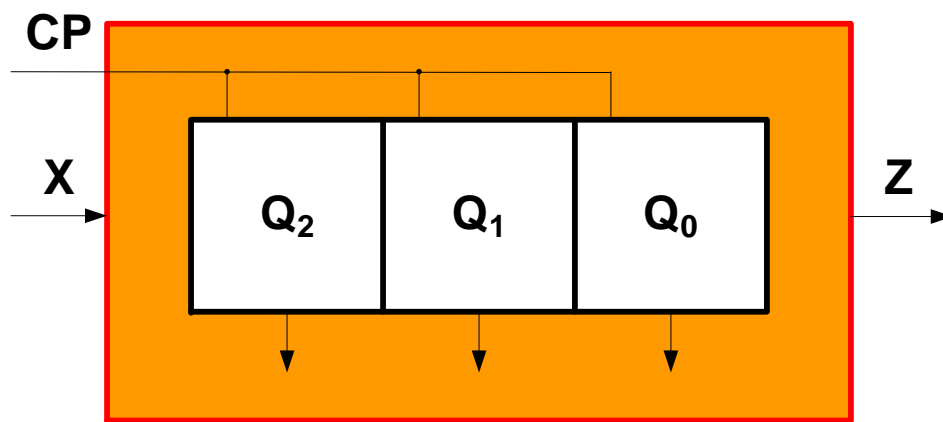
Minimiziranje memorije

- algoritam Huffman-Mealyjeve metode:
 - podjela unutarnjih stanja u *najmanji mogući broj* klasa ekvivalentnih stanja, tako da stanja u istoj klasi imaju *iste izlaze*
~ stanja grupirati s obzirom na izlaze
 - *daljnja podjela* dobivenih klasa na podklase, tako da prijelazi iz stanja jedne te iste klase vode u stanja jedne druge iste klase
 - u konačnici se prijelazi između stanja zamjenjuju prijelazima između *klasa* stanja
~ klasa naravno ima *manje* od stanja 😊

Minimiziranje memorije

Primjer: Huffman-Mealyjeva metoda

- sekvencijski sklop s jednim ulazom x i 8 stanja
- $8 = 2^3$ stanja \rightarrow 3 bistabila Q_2, Q_1, Q_0
 \sim promatraju se Q_i (promjena stanja) + *jedan* izlaz
- rad sekvencijskog sklopa
 \sim tablica stanja

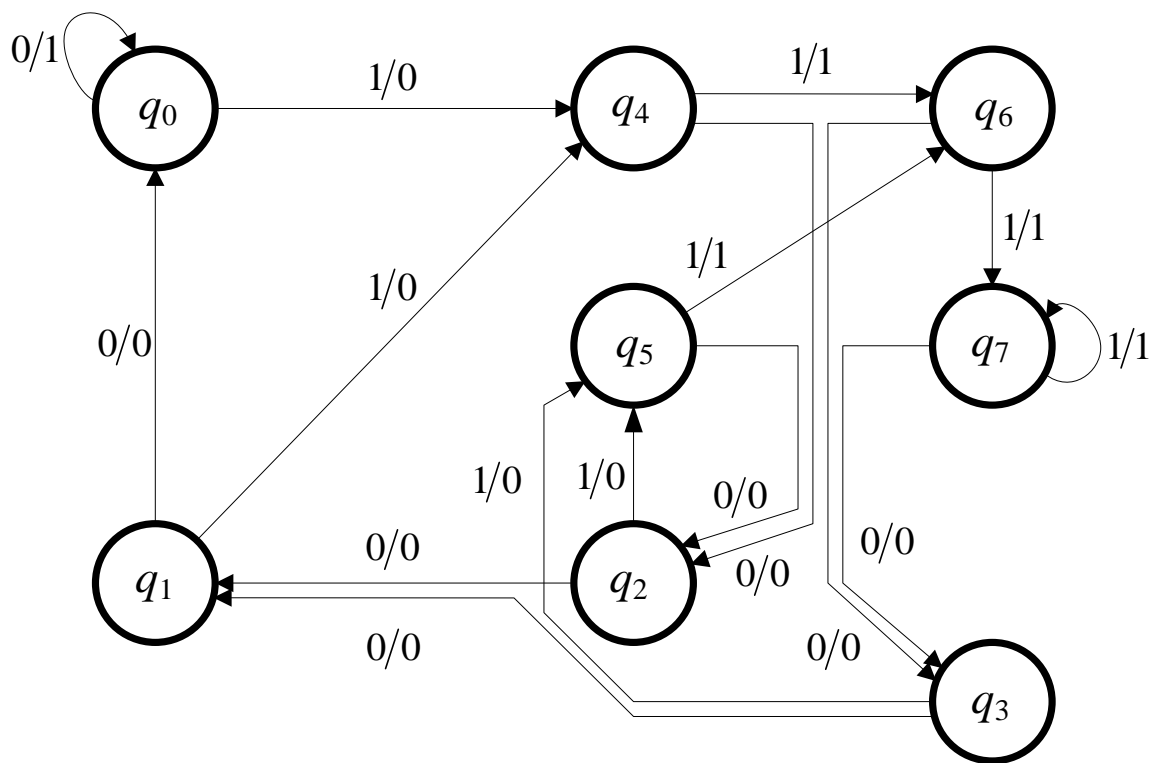


q^n	q^{n+1}, z^n	
	$x^n=0$	$x^n=1$
q_0	$q_0, 1$	$q_4, 0$
q_1	$q_0, 0$	$q_4, 0$
q_2	$q_1, 0$	$q_5, 0$
q_3	$q_1, 0$	$q_5, 0$
q_4	$q_2, 0$	$q_6, 1$
q_5	$q_2, 0$	$q_6, 1$
q_6	$q_3, 0$	$q_7, 1$
q_7	$q_3, 0$	$q_7, 1$

Minimiziranje memorije

- specifikacija automata:

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



Minimiziranje memorije

- klase ekvivalentnih stanja
prema stanju izlaza
~ (početno) 3 klase

$$a = \{q_0\}$$

$$b = \{q_1, q_2, q_3\}$$

$$c = \{q_4, q_5, q_6, q_7\}$$

- provjera prijelaza
~ 4 klase ekvivalentnih stanja

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

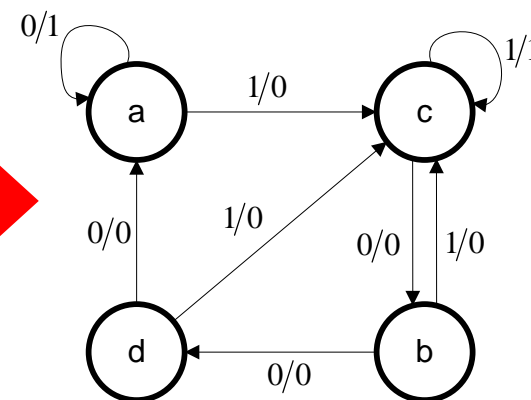
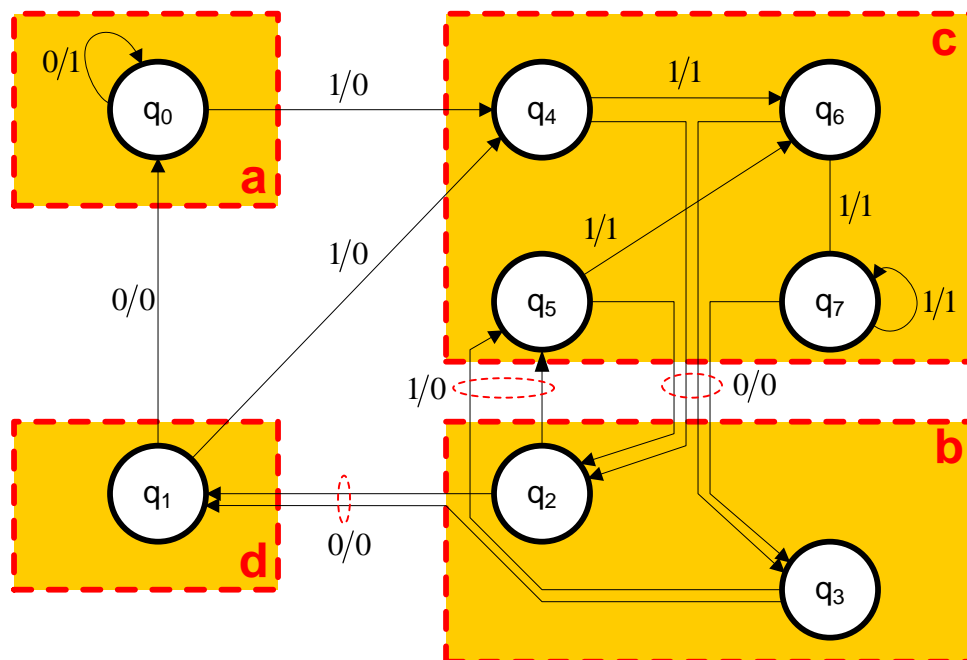
klasa	a	b			c			
stanje	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7
sl. klasa	a c	a c	b c	b c	b c	b c	b c	b c

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	d c	d c	b c	b c	b c	b c	a c



Minimiziranje memorije

- dijagram stanja:



- nova tablica stanja

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	d,0	c,0
q_4, q_5, q_6, q_7	c	b,0	c,1
q_1	d	a,0	c,0

- kodiranje stanja
 - ~ pridruživanje binarne kodne riječi pojedinom stanju:
 - utječe na veličinu kombinacijskog sklopa
 - težak kombinatorni problem
 - ~ prihvatljiva podoptimalna rješenja
 - trivijalno kodiranje
 - ~ prirodni binarni kod

Kodiranje stanja

Primjer: kodiranje stanja

- *kombinirana* tablica stanja
~ prijelaz + izlaz u ovisnosti o pobudi (ulazu)
- 4 stanja \rightarrow 2 bistabila (B_1, B_0)
- trivijalno kodiranje
~ binarni kod

q^n	q^{n+1}		z^n	
	$x^n=0$	$x^n=1$	$x^n=0$	$x^n=1$
a	a	c	1	0
b	d	c	0	0
c	b	c	0	1
d	a	c	0	0



$(B_1B_0)^n$	$(B_1B_0)^{n+1}$		z^n	
	$x=0$	$x=1$	$x=0$	$x=1$
00	00	10	1	0
01	11	10	0	0
10	01	10	0	1
11	00	10	0	0

Ostvarivanje sekvencijskog sklopa

Primjer: implementacija memorije D bistabilima

$(B_1B_0)^n$	$(B_1B_0)^{n+1}$				z^n	
	$x=0$		$x=1$		$x=0$	$x=1$
00	0	0	1	0	1	0
01	1	1	1	0	0	0
10	0	1	1	0	0	1
11	0	0	1	0	0	0



B_1^{n+1}

	B_1B_0			
	00	01	11	10
X 0		1		
1	1	1	1	1

B_0^{n+1}

	B_1B_0			
	00	01	11	10
X 0		1		1
1				

Z

	B_1B_0			
	00	01	11	10
X 0	1			
1				1

Ostvarivanje sekvencijskog sklopa

- D bistabili
~ posebno jednostavno dobivanje
ulazne jednačbe iz karakteristične :

$$B^{n+1} = D^n \Rightarrow D^n = B^{n+1}$$

$$D_1 = X + \overline{B}_1 B_0$$

$$\begin{aligned} D_0 &= \overline{X} \cdot (\overline{B}_1 B_0 + B_1 \overline{B}_0) \\ &= \overline{X} \cdot (B_1 \oplus B_0) \end{aligned}$$

$$Z = \overline{B}_1 \overline{B}_0 \overline{X} + B_1 \overline{B}_0 X$$

		$B_1 B_0$			
		00	01	11	10
X	0		1		
	1	1	1	1	1

		$B_1 B_0$			
		00	01	11	10
X	0		1		1
	1				

		$B_1 B_0$			
		00	01	11	10
X	0	1			
	1				1

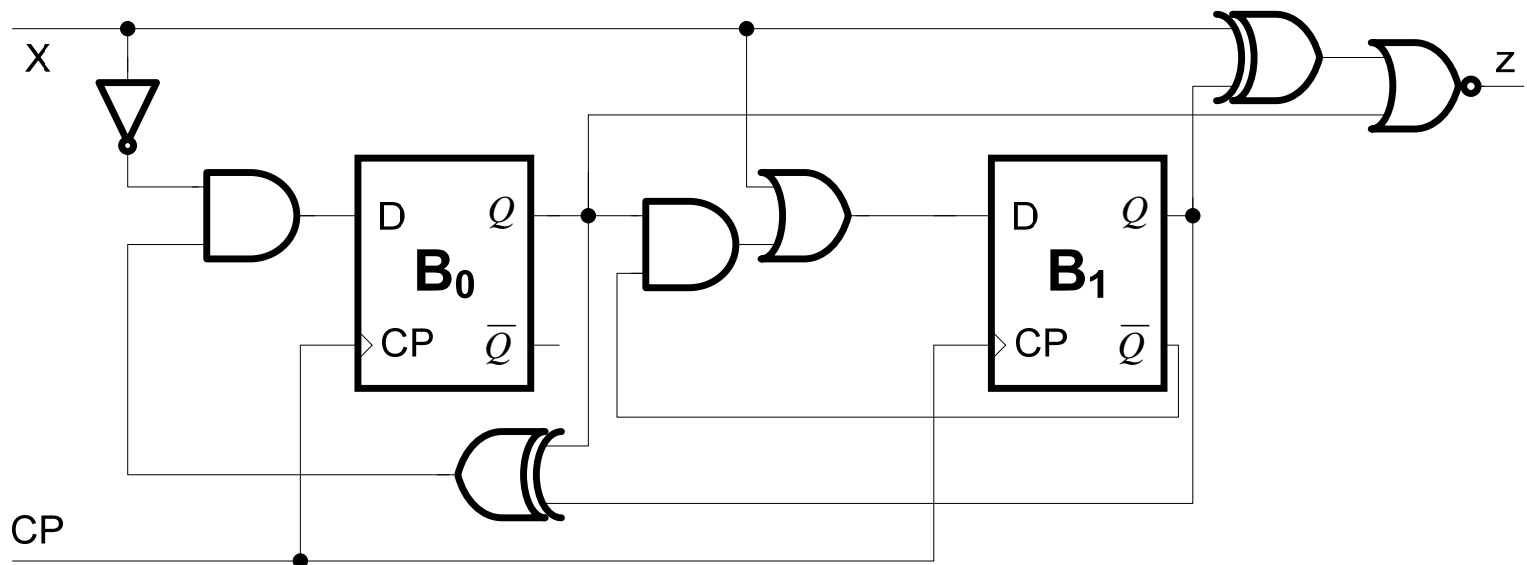
Ostvarivanje sekvencijskog sklopa

- logička shema:

$$D_1 = X + \bar{B}_1 B_0$$

$$\begin{aligned} D_0 &= \bar{X} \cdot (\bar{B}_1 B_0 + B_1 \bar{B}_0) \\ &= \bar{X} \cdot (B_1 \oplus B_0) \end{aligned}$$

$$Z = \bar{B}_1 \bar{B}_0 \bar{X} + B_1 \bar{B}_0 X = \overline{B_0 + (B_1 \oplus X)}$$



Ostvarivanje sekvencijskog sklopa

Primjer: izvedba memorije JK-bistabilima

- koristi se *uzbudna tablica*:

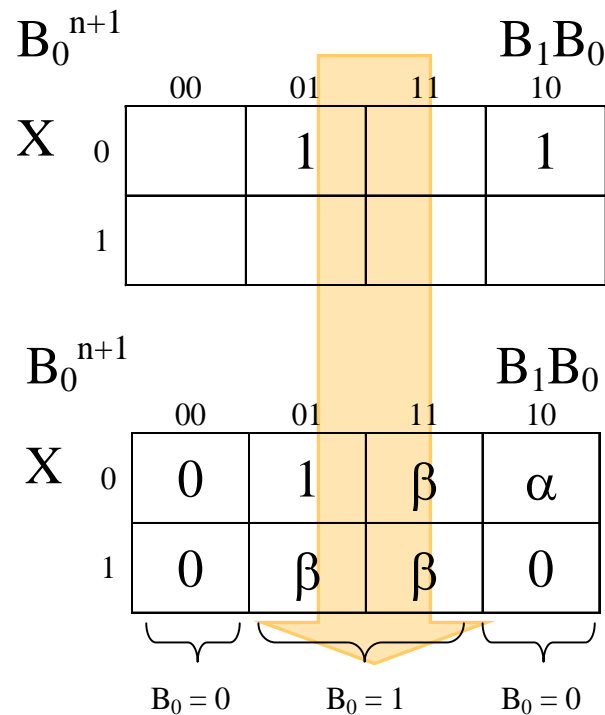
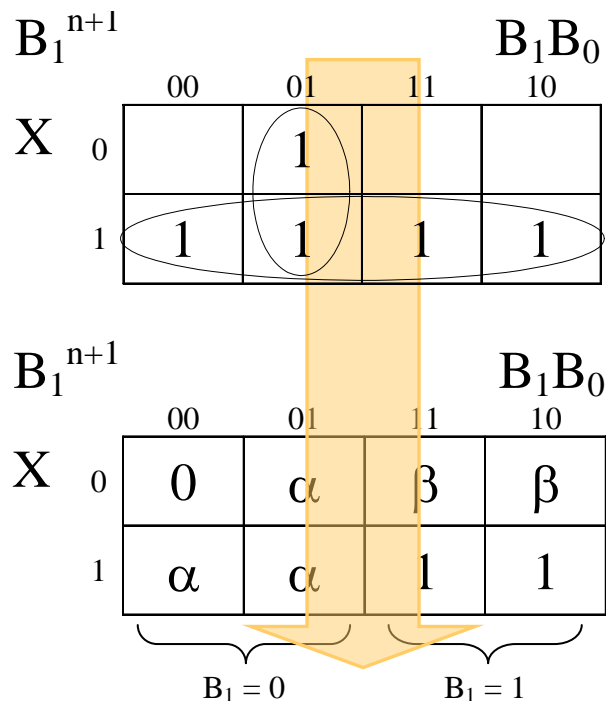
	Q^n	Q^{n+1}	J	K
$\alpha := 0 \rightarrow 1$	0	1	1	x
$\beta := 1 \rightarrow 0$	1	0	x	1
$0 := 0 \rightarrow 0$	0	0	0	x
$1 := 1 \rightarrow 1$	1	1	x	0

oznaka vrste prijelaza vrsta prijelaza potrebna uzbuda JK-bistabila

Ostvarivanje sekvencijskog sklopa

- označavanje prijelaza zapisom *uzbudne tablice*:

Q^n	Q^{n+1}	J	K
0 → 1 : α		1	x
1 → 0 : β		x	1
0 → 0 : 0		0	x
1 → 1 : 1		x	0



Ostvarivanje sekvencijskog sklopa

- izvođenje ulaznih jednažbi:

Q^n	Q^{n+1}		J	K
0	→ 1	: α	1	x
1	→ 0	: β	x	1
0	→ 0	: 0	0	x
1	→ 1	: 1	x	0

B_1^{n+1}

	00	01	11	10
X 0	0	α	β	β
1	α	α	1	1

$B_1 B_0$

$B_1 = 0$ $B_1 = 1$

B_0^{n+1}

	00	01	11	10
X 0	0	1	β	α
1	0	β	β	0

$B_1 B_0$

$B_0 = 0$ $B_0 = 1$ $B_0 = 0$

J_1

	00	01	11	10
X 0		1	x	x
1	1	1	x	x

$B_1 B_0$

J_0

	00	01	11	10
X 0		x	x	1
1		x	x	

$B_1 B_0$

K_1

	00	01	11	10
X 0	x	x	1	1
1	x	x		

$B_1 B_0$

K_0

	00	01	11	10
X 0	x		1	x
1	x	1	1	x

$B_1 B_0$

Ostvarivanje sekvencijskog sklopa

- ulazne jednačbe:

J_1

		$B_1 B_0$			
		00	01	11	10
X	0		1	x	x
	1	1	1	x	x

J_0

		$B_1 B_0$			
		00	01	11	10
X	0		x	x	1
	1		x	x	

K_1

		$B_1 B_0$			
		00	01	11	10
X	0	x	x	1	1
	1	x	x		

K_0

		$B_1 B_0$			
		00	01	11	10
X	0	x		1	x
	1	x	1	1	x

$$J_1 = B_0 + X$$

$$K_1 = \bar{X}$$

$$J_0 = B_1 \bar{X}$$

$$K_0 = X + B_1$$

Ostvarivanje sekvencijskog sklopa

- logička shema:

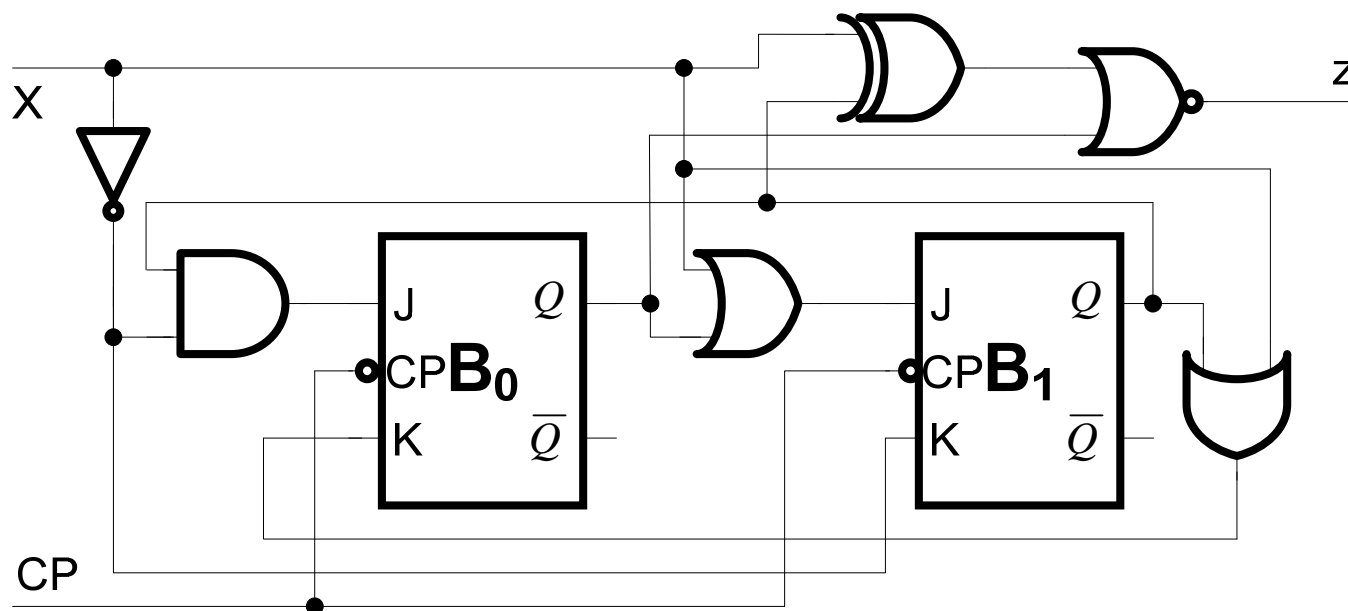
$$J_0 = B_1 \bar{X}$$

$$J_1 = B_0 + X$$

$$Z = \overline{B_0 + (B_1 \oplus X)}$$

$$K_0 = X + B_1$$

$$K_1 = \bar{X}$$





Sadržaj predavanja

- konceptualizacija sekvencijskih sklopova
- projektiranje sekvencijskih sklopova
- **izvedbe sekvencijskih sklopova**



Izvedbe sekvencijskih sklopova

- mogućnosti izvedbe
~ raspoloživa "tehnologija":
 - bistabili + kombinacijska logika (osnovni logički sklopovi)
~ kao prije
 - registar (= niz D bistabila!) + ROM
 - registar + SPLD (npr. PLA, PAL, ...)
 - druga "univerzalna logika":
 - CPLD
 - FPGA
 - sekvencijski moduli

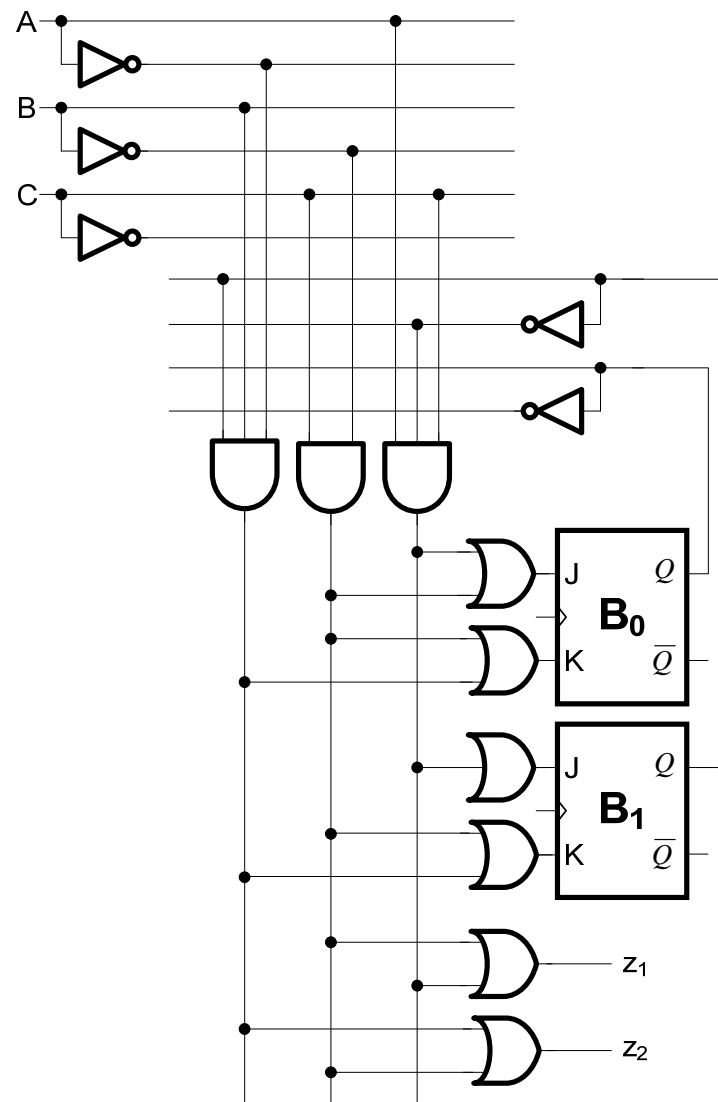


Izvedbe sekvencijskih sklopova

- *sekvencijski moduli*:
~ cjeline koje sadrže kombinacijski sklop / memoriju
(niz/skup bistabila ili registara)
- općenita klasifikacija:
 - *standardni* moduli: n -bitni
 - za funkcije tipa brojanja: npr. brojila
 - za funkcije tipa pohranjivanja podataka:
npr. registri
 - *univerzalni* moduli
~ ostvarivanje proizvoljnih sekvencijskih sklopova
(usp. generiranje Booleovih funkcija
kombinacijskim modulima)

Izvedbe sekvencijskih sklopova

- primjer univerzalnog sekvencijskog modula
~ *programirljivo sekvencijsko polje*
(engl. sequential PLA)



U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 9:
Sinkroni sekvencijski sklopovi.

- konceptualizacija sekvencijskih sklopova: str. 335-341
- projektiranje sekvencijskih sklopova: str. 354-375
- izvedbe sekvencijskih sklopova: str. 379-380



Zadaci za vježbu (1)

- U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 9: Sinkroni sekvencijski sklopovi; Poglavlje 11: Sekvencijski moduli: registri i brojila.
- projektiranje sekvencijskih sklopova: 9.5-9.9, 9.14, 9.17, 9.18, 9.22, 9.23, 9.26
 - izvedbe sekvencijskih sklopova: 9.15, 9.16; 11.25



Zadaci za vježbu (2)

M. Čupić: *Digitalna elektronika i digitalna logika. Zbirka riješenih zadataka*, Cjelina 11: Strojevi s konačnim brojem stanja.

- konceptualizacija sekvencijskih sklopova:
 - riješeni zadaci: 11.6, 11.7
- projektiranje sekvencijskih sklopova:
 - riješeni zadaci: 11.1-11.5, 11.8, 11.9, 11.12, 11.13, 11.15
 - zadaci za vježbu: 6, 7
- izvedbe sekvencijskih sklopova:
 - riješeni zadaci: 11.14
 - zadaci za vježbu: 4



10. Sekvencijski sklopovi (2)



Sadržaj predavanja

- **analiza sekvencijskih sklopova**
- vremenski odnosi
- modeliranje sekvencijskih sklopova u jeziku VHDL

Analiza sekvencijskih sklopova

- analiza *sinkronog* sekvencijskog sklopa
~ *obrnuti* postupak:
 - *ponašanje* (= rad) postojećeg sklopa?
 - *formalni* opis
- formalizmi poznati od prije:
 - tablica stanja
~ prijelazi u sljedeća stanja, izlazi
 - dijagram stanja
~ grafički prikaz tablice stanja
 - jednađbe stanja
 - iz tablice stanja ili direktno iz logičke sheme
 - opis uvjeta za promjenu stanja bistabila:

$$B_i^{n+1} = f\left(B_0^n, B_1^n, \dots, B_{r-1}^n, x_0, \dots, x_{l-1}\right)$$

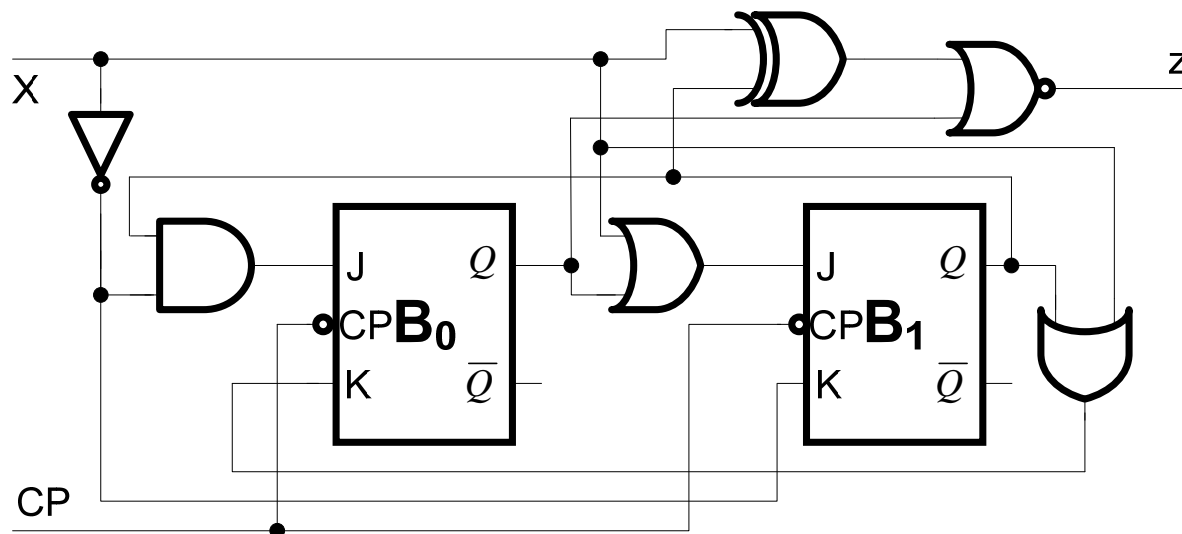


Analiza sekvencijskih sklopova

- uobičajeni postupak analize:
 - očitati *logičku shemu* iz samog sklopa
 - iz *logičke sheme* izvesti:
 - *ulazne* jednadžbe za svaki bistabil
~ Booleov izraz koji utvrđuje potrebnu pobudu za željeno ponašanje bistabila
 - *izlazne* jednadžbe za svaki izlaz
 - iz ulaznih jednadžbi bistabila i izlaznih jednadžbi ispisati *tablicu stanja*
~ početno stanje = neko karakteristično stanje
npr. kodna riječ 0 (000...0)

Analiza sekvencijskih sklopova

Primjer: analiza prethodnog sklopa



iz sheme sklopa očitano:

$$J_1 = B_0 + X \qquad J_0 = B_1 \bar{X}$$

$$K_1 = \bar{X} \qquad K_0 = X + B_1$$

$$Z = \overline{B_0 + (B_1 \oplus X)} = \overline{B_1} \overline{B_0} \overline{X} + B_1 \overline{B_0} X$$

Analiza sekvencijskih sklopova

$$J_1 = B_0 + X$$

$$J_0 = B_1 \bar{X}$$

$$Z = \bar{B}_1 \bar{B}_0 \bar{X} + B_1 \bar{B}_0 X$$

$$K_1 = \bar{X}$$

$$K_0 = X + B_1$$

- pridruživanje:

B ₁	B ₀	stanje
0	0	a
0	1	b
1	0	c
1	1	d

- tablica stanja:

	X=0	X=1
a	a,1	c,0
b	d,0	c,0
c	b,0	c,1
d	a,0	c,0



n							n+1		
B ₁	B ₀	X	J ₁	K ₁	J ₀	K ₀	B ₁	B ₀	Z
0	0	0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	1	0	0
0	1	0	1	1	0	0	1	1	0
0	1	1	1	0	0	1	1	0	0
1	0	0	0	1	1	1	0	1	0
1	0	1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0	0	0
1	1	1	1	0	0	1	1	0	0



Sadržaj predavanja

- analiza sinkronih sekvencijskih sklopova
- **vremenski odnosi**
- modeliranje sekvencijskih sklopova u jeziku VHDL



Vremenski odnosi

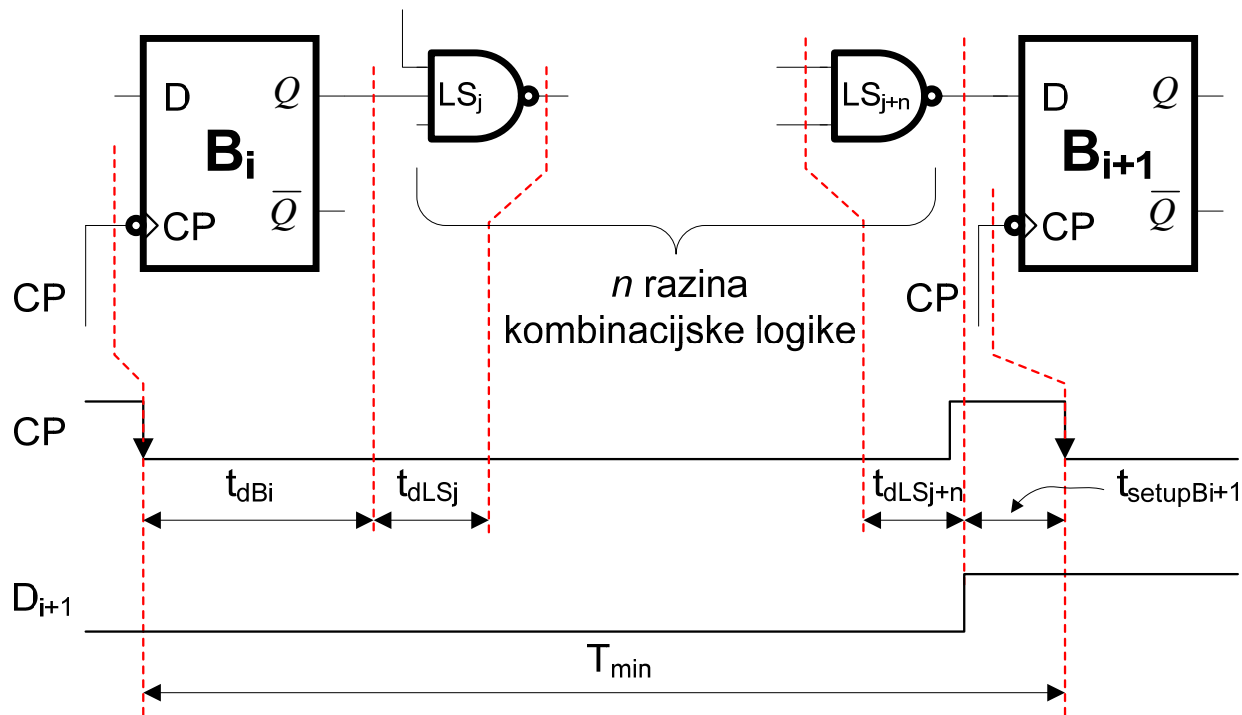
- vremenski odnosi u sekvencijskom sklopu
~ *dvije* značajne veličine:
 - maksimalna frekvencija rada sekvencijskog sklopa
~ najveća frekvencija CP,
a da sklop (= svi njegovi bistabili)
sigurno mijenja stanje kad to ulazi zahtijevaju
 - raskorak, razdešenost ritma
~ najveće dozvoljeno *kašnjenje okidanja*
bistabila u sklopu,
a da sklop *sigurno mijenja stanje*
kad to ulazi zahtijevaju

Vremenski odnosi

- *maksimalna frekvencija* rada sekvencijskog sklopa:
 - težnja
~ što viša f_{\max}
 - veći broj operacija/sek
 - veća brzina rada digitalnog sklopa
 - problem
~ kašnjenje signala
na stazi između *dva susjedna* stupnja (\rightarrow bistabila)
 - odrediti *najmanji* $T_{\min} = 1/f_{\max}$
koji osigurava ispravno okidanje
bistabila *sljedećeg stupnja*,
za *najlošiju* stazu signala
~ *najviše* kombinacijske logike između bistabila

Vremenski odnosi

- maksimalna frekvencija, f_{\max} :

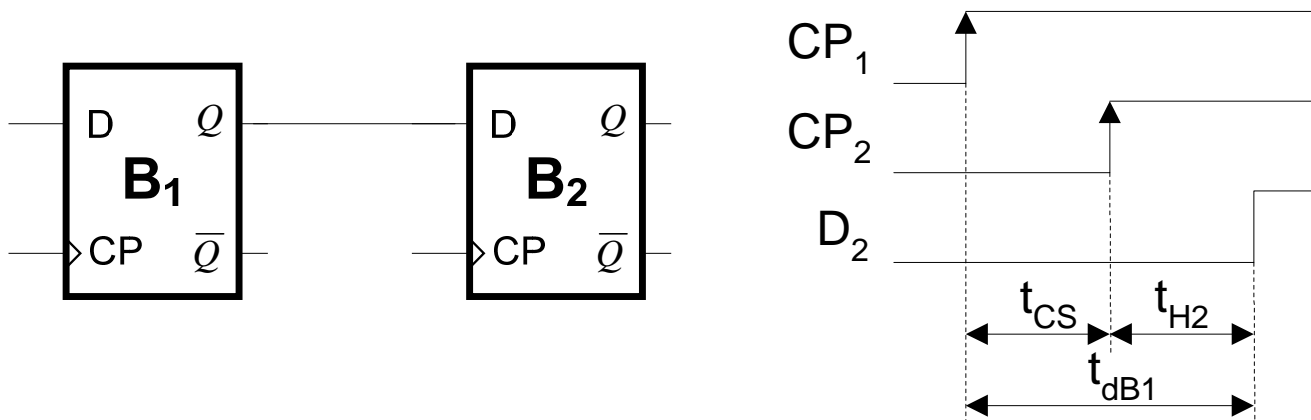


$$T_{\min} = t_{db} \Big|_{\max} + n \cdot t_{dLS} \Big|_{\max} + t_{setup} \Big|_{\max} \Rightarrow f_{\max} = \frac{1}{T_{\min}}$$

- *raskorak, razdešenost ritma* (engl. clock skew):
 - takt (pobuda radi promjene stanja): tipično iz *jedinstvenog* generatora
 - problem s *istovremenim* okidanjem bistabila
 - *različite* duljine vodova do bistabila
 - preveliko ukupno *opterećenje* pobudnih sklopova
~ signal CP se razvodi iz *više* pobudnih sklopova
 - *više* pojačala
 - *različita kašnjenja* pojedinih pojačala

Primjer: neistovremeno okidanje bistabila

- osigurati ispravan upis prethodnog stanja B_1 u B_2
- novo stanje B_1 *ne smije se pojaviti* na ulazu B_2 prije nego je B_2 ispravno prihvatio prethodno stanje B_1



Vremenski odnosi

- za najlošiji slučaj vrijedi:

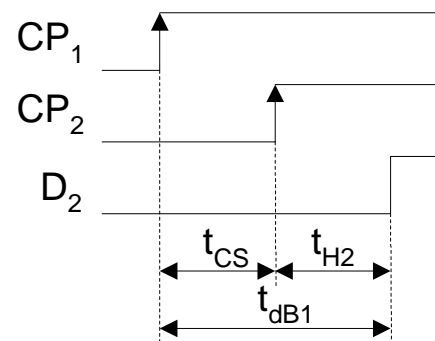
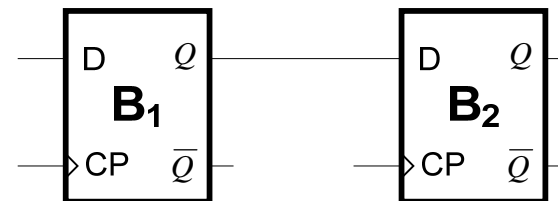
$$(t_{dB1})_{\min} \geq (t_{H2})_{\max} + (t_{CS})_{\max}$$

- osigurati ograničenje razdešenosti ritma:

$$(t_{CS})_{\max} \leq (t_{dB1})_{\min} - (t_{H2})_{\max}$$

t_{CS} : raskorak (engl. *clock skew time*)

t_H : vrijeme držanja (engl. *hold time*)



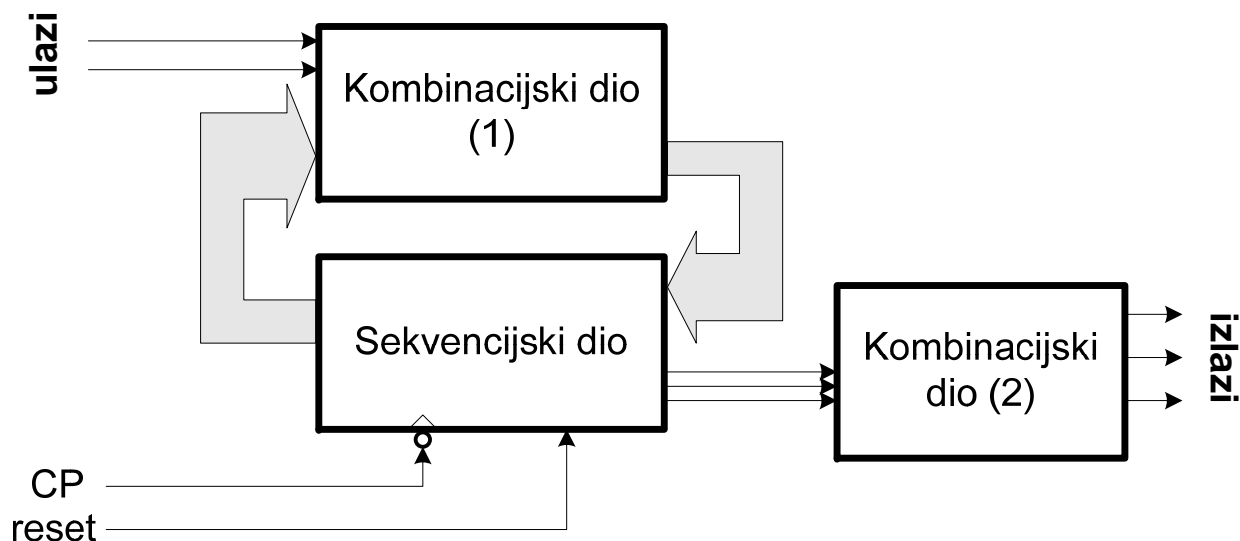


Sadržaj predavanja

- analiza sinkronih sekvencijskih sklopova
- vremenski odnosi
- **modeliranje sekvencijskih sklopova u jeziku VHDL**

Modeliranje sekvencijskih sklopova...

Primjer: Prikazati osnovnu strukturu VHDL modela kojim se opisuje Mooreov stroj s konačnim brojem stanja.
(Zbirka, zadatak 11.19)





Modeliranje sekvencijskih sklopova...

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
  
ENTITY automatMoore IS PORT (  
    ulazi:    IN std_logic_vector(N DOWNTO 0);  
    reset:    IN std_logic; -- inicijalizira početno stanje  
    izlazi:   OUT std_logic_vector(M DOWNTO 0);  
    clock:    IN std_logic  
);  
END automatMoore;
```

Modeliranje sekvencijskih sklopova...

- intuitivno kodiranje stanja
~ poredanje stanja u binarnom kodu

```
ARCHITECTURE ponasanje OF automatMoore IS
  SIGNAL state_present, state_next: std_logic_vector(K DOWNTO 0);
  CONSTANT S0: std_logic_vector(K DOWNTO 0) := "0...000";
  CONSTANT S1: std_logic_vector(K DOWNTO 0) := "0...001";
  -- ...
BEGIN
  -- Blok koji modelira Kombinatijski dio (1) na temelju ulaza
  -- i trenutnog stanja računa sljedeće stanje.
  PROCESS(ulazi, state_present)
  BEGIN
    -- na temelju signala iz liste osjetljivosti određuje se
    -- u koje bi sljedeće stanje sklop trebao prijeći.
    -- Npr. za bezuvjetan prijelaz u stanje S0:
    state_next <= S0;
  END PROCESS;
```




Modeliranje sekvencijskih sklopova...

```
-- Blok koji modelira Kombinatorski dio (2) na temelju
-- trenutnog stanja računa izlaz sklopa.
PROCESS(state_present)
BEGIN
    -- na temelju trenutnog stanja odrediti izlaz sklopa:
    CASE state_present IS
        WHEN S0 => izlazi <= ...;
        WHEN S1 => izlazi <= ...;
        -- ...
        WHEN OTHERS => izlaz <= ...;
    END CASE;
END PROCESS;
```



Modeliranje sekvencijskih sklopova...

```
-- Blok koji modelira Sekvencijski dio na temelju signala  
-- clock i asinkronih ulaza mijenja stanje
```

```
PROCESS( clock, reset )
```

```
BEGIN
```

```
-- Provjera asinkronih ulaza
```

```
IF reset = '1' THEN
```

```
    state_present <= S0;
```

```
-- Inače slijedi sinkrono djelovanje
```

```
ELSIF falling_edge(clock) THEN
```

```
    state_present <= state_next;
```

```
END IF;
```

```
END PROCESS;
```

```
END ponasanje;
```



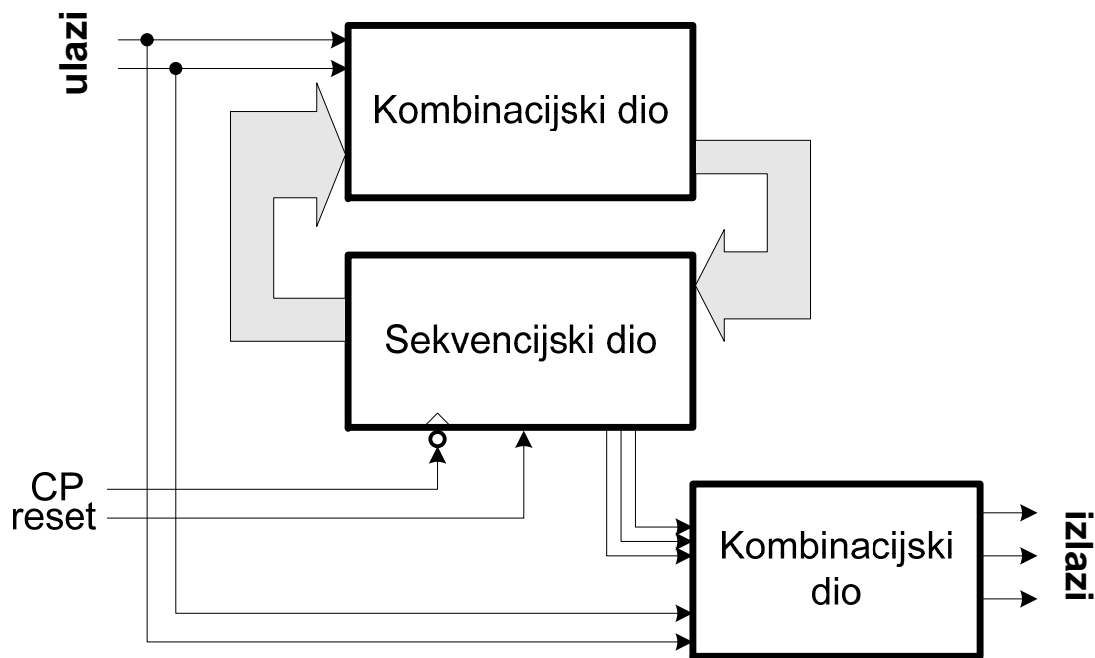
Modeliranje sekvencijskih sklopova...

- alternativni (bolji?) način rada sa stanjima:
→ navesti oznake stanja, ali ne i način kodiranja

```
TYPE stateType IS (S0, S1, S2, S3);  
SIGNAL state_present, state_next: stateType;
```

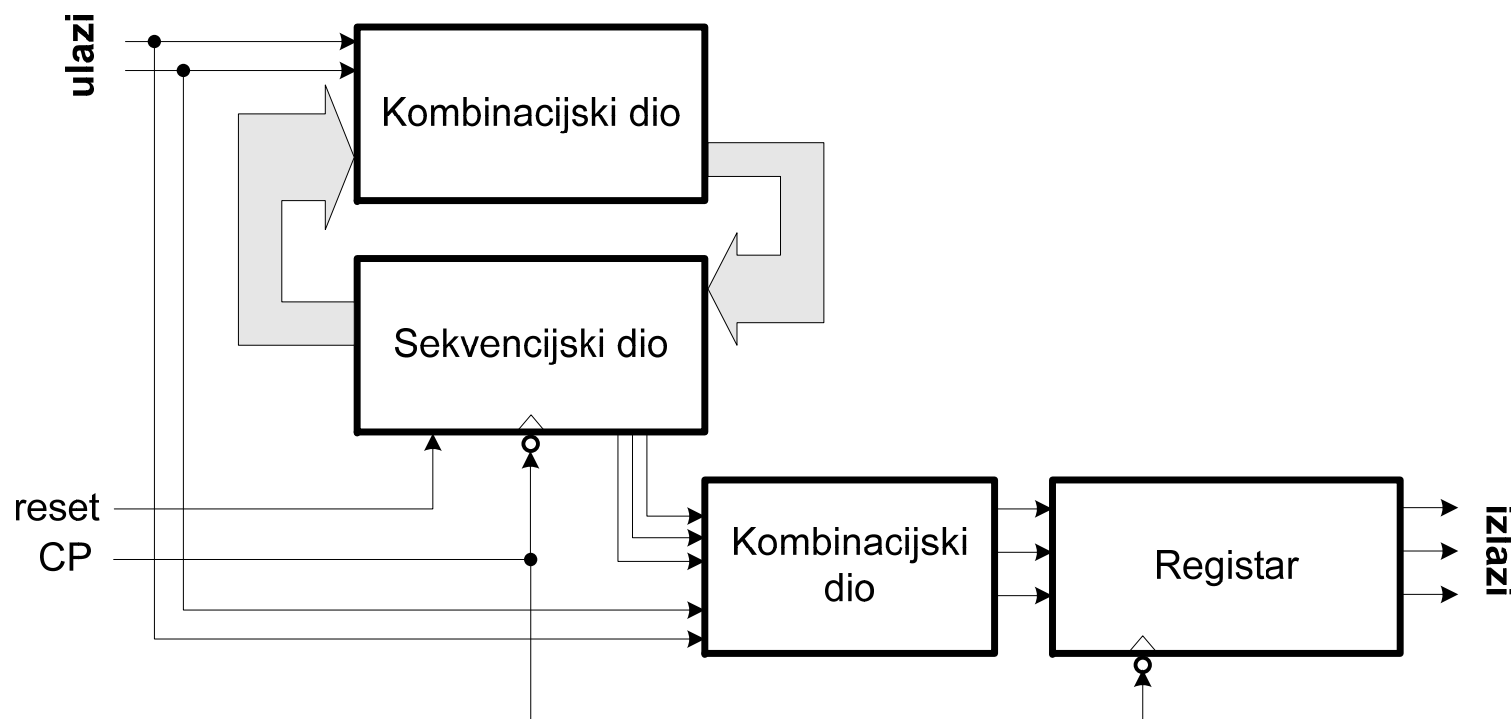
Modeliranje sekvencijskih sklopova...

Primjer: Prikazati osnovnu strukturu VHDL modela kojim se opisuje Mealyjev stroj s konačnim brojem stanja. (Zbrikanje zadatka 11.20)



Modeliranje sekvencijskih sklopova...

- verzija s registrima na izlazu
(stabilne vrijednosti izlaza između dva susjedna impulsa CP)





Modeliranje sekvencijskih sklopova...

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

ENTITY automatMealy IS PORT (
    ulazi: IN std_logic_vector(N DOWNTO 0);
    reset: IN std_logic;
    izlazi: OUT std_logic_vector(M DOWNTO 0);
    clock: IN std_logic
);
END automatMealy;
```

Modeliranje sekvencijskih sklopova...

```
ARCHITECTURE ponasanje OF automatMealy IS
  SIGNAL state_present, state_next: std_logic_vector(K DOWNTO 0);
  CONSTANT S0: std_logic_vector(K DOWNTO 0) := "0...000";
  CONSTANT S1: std_logic_vector(K DOWNTO 0) := "0...001";
  SIGNAL izlazi_next: std_logic_vector(M DOWNTO 0);
BEGIN
  -- blok koji modelira KD1 i KD2;
  -- na temelju ulaza i trenutnog stanja računaju se
  -- sljedeće stanje i sljedeći izlazi;
  PROCESS( ulazi, state_present )
  BEGIN
    -- utvrđuje se koje će biti sljedeće stanje i izlaz;
    -- odluka se donosi na temelju trenutnog stanja i ulaza;
    -- npr. za bezuvjetni prijelaz u stanje S0 i sve '0'
    -- na izlazima:
    state_next <= S0;
    izlazi_next <= "000..000";  -- niz od (K+1) nule.
  END PROCESS;
```

Modeliranje sekvencijskih sklopova...

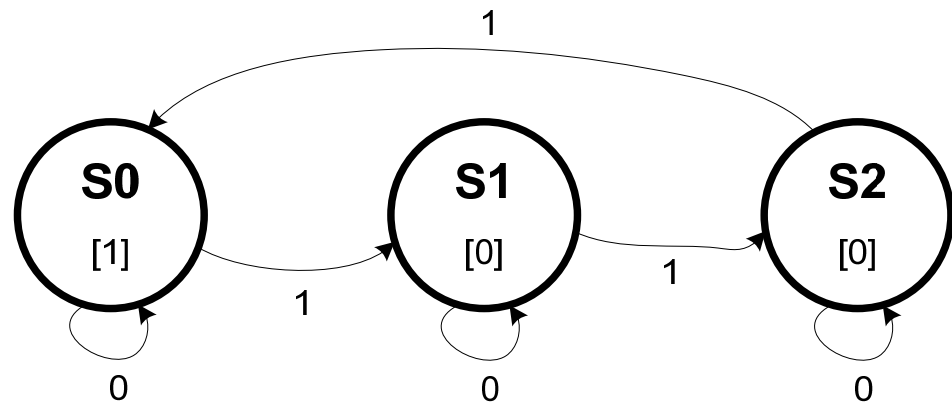
```
-- blok koji modelira Sekvencijski dio i registar;
-- na temelju signala clock i asinkronih ulaza
-- mijenja se stanje i izlaz
PROCESS( clock, reset )
BEGIN
    -- provjera asinkronih ulaza
    IF reset = '1' THEN
        state_present <= S0;  -- postaviti stanje u S0
        izlaz <= "000...00";  -- postaviti izlaze u npr. '0'
    -- inače slijedi sinkrono djelovanje
    ELSIF falling_edge(clock) THEN
        state_present <= state_next;
        izlazi <= izlazi_next;
    END IF;
END PROCESS;

END ponasanje;
```


Modeliranje sekvencijskih sklopova...

Primjer: Stroj s konačnim brojem stanja (Zbirka zadataka 11.1)
opisati VHDL-om (Zbirka zadataka 11.21)

- prikaz stroja:



```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

```
ENTITY automat1 IS PORT (  
    input0: IN std_logic;  
    y: OUT std_logic;  
    clock: IN std_logic);  
END automat1;
```

Modeliranje sekvencijskih sklopova...

```
ARCHITECTURE Behavioral OF automat1 IS
    SIGNAL state_present, state_next: std_logic_vector(1 DOWNTO 0);
    CONSTANT S0: std_logic_vector(1 DOWNTO 0) := "11";
    CONSTANT S1: std_logic_vector(1 DOWNTO 0) := "01";
    CONSTANT S2: std_logic_vector(1 DOWNTO 0) := "10";
BEGIN
    PROCESS(input0, state_present)
    BEGIN
        CASE state_present IS
            WHEN S0 =>      IF input0 = '0' THEN state_next <= S0;
                           ELSE                  state_next <= S1;
                           END IF;
            WHEN S1 =>      IF input0 = '0' THEN state_next <= S1;
                           ELSE                  state_next <= S2;
                           END IF;
            WHEN S2 =>      IF input0 = '0' THEN state_next <= S2;
                           ELSE                  state_next <= S0;
                           END IF;
            WHEN OTHERS =>      state_next <= S0;
        END CASE;
    END PROCESS;
```

Modeliranje sekvencijskih sklopova...

```
PROCESS (state_present)
BEGIN
    CASE state_present IS
        WHEN S0 => y <= '1';
        WHEN S1 => y <= '0';
        WHEN S2 => y <= '0';
        WHEN OTHERS => y <= '0';
    END CASE;
END PROCESS;

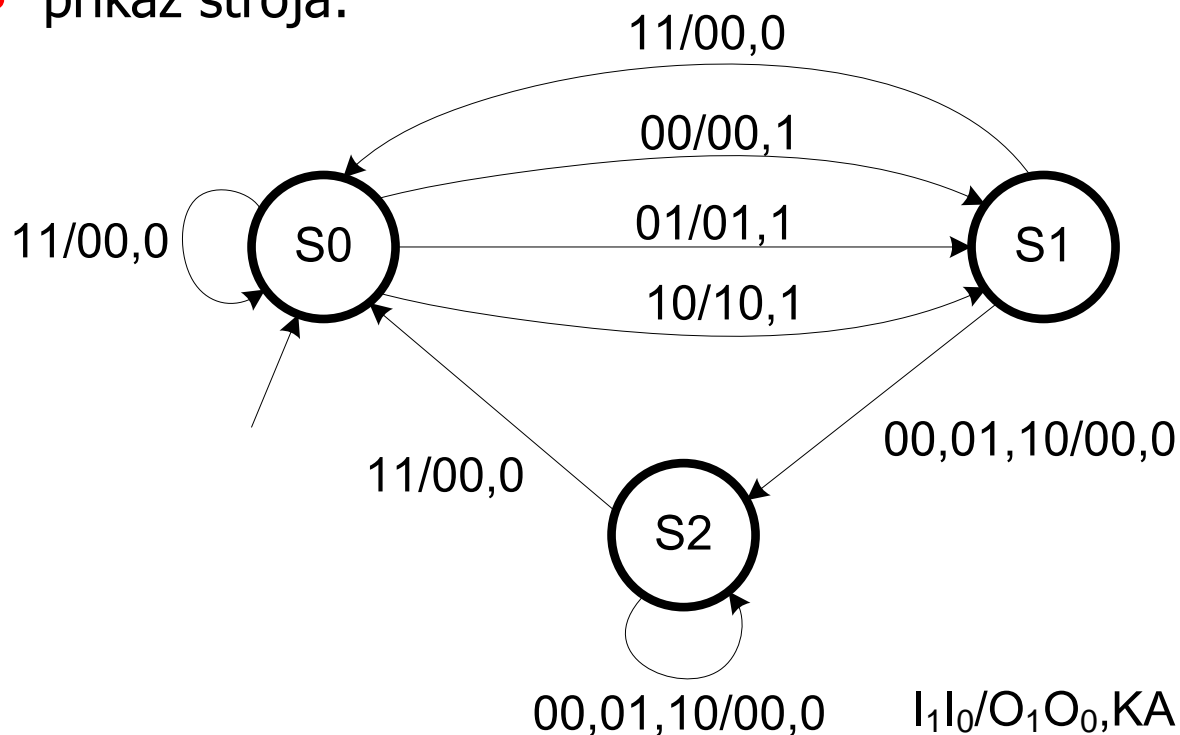
PROCESS ( clock )
BEGIN
    IF falling_edge(clock) THEN
        state_present <= state_next;
    END IF;
END PROCESS;

END BEHAVIORAL;
```

Modeliranje sekvencijskih sklopova...

Primjer: Stroj s konačnim brojem stanja (Zbirka zadataka 11.2) opisati VHDL-om. (Zbirka zadataka 11.22)

- prikaz stroja:





Modeliranje sekvencijskih sklopova...

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

ENTITY automat2 IS PORT (
    i1,i0 : IN std_logic;
    o1 : OUT std_logic; o0 : OUT std_logic;
    ka : OUT std_logic;
    clock: IN std_logic);
END automat2;

ARCHITECTURE Behavioral OF automat2 IS
    SIGNAL state_present, state_next: std_logic_vector(1 DOWNTO 0);
    SIGNAL o1_next, o0_next, ka_next: std_logic;
    CONSTANT S0: std_logic_vector(1 DOWNTO 0) := "00";
    CONSTANT S1: std_logic_vector(1 DOWNTO 0) := "01";
    CONSTANT S2: std_logic_vector(1 DOWNTO 0) := "10";
BEGIN
```

Modeliranje sekvencijskih sklopova...

```
PROCESS( i1, i0, state_present )
    VARIABLE pom: std_logic_vector(1 DOWNT0 0);
BEGIN
    pom := (i1, i0);
    CASE state_present IS
        WHEN S0 =>
            CASE pom IS
                WHEN "00" => state_next <= S1; o1_next <= '0';
                           o0_next <= '0'; ka_next <= '1';
                WHEN "01" => state_next <= S1; o1_next <= '0';
                           o0_next <= '1'; ka_next <= '1';
                WHEN "10" => state_next <= S1; o1_next <= '1';
                           o0_next <= '0'; ka_next <= '1';
                WHEN "11" => state_next <= S0; o1_next <= '0';
                           o0_next <= '0'; ka_next <= '0';
                WHEN OTHERS => state_next <= S0; o1_next <= '0';
                              o0_next <= '0'; ka_next <= '0';
            END CASE;
    END CASE;
```



Modeliranje sekvencijskih sklopova...

```
WHEN S1 =>
  CASE pom IS
    WHEN "00" => state_next <= S2; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN "01" => state_next <= S2; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN "10" => state_next <= S2; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN "11" => state_next <= S0; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN OTHERS => state_next <= S0; o1_next <= '0';
                   o0_next <= '0'; ka_next <= '0';
  END CASE;
```

Modeliranje sekvencijskih sklopova...

```
WHEN S2 =>
  CASE pom IS
    WHEN "00" => state_next <= S2; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN "01" => state_next <= S2; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN "10" => state_next <= S2; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN "11" => state_next <= S0; o1_next <= '0';
                  o0_next <= '0'; ka_next <= '0';
    WHEN OTHERS => state_next <= S0; o1_next <= '0';
                   o0_next <= '0'; ka_next <= '0';

    END CASE;
  WHEN OTHERS =>
    state_next <= S0; o1_next <= '0';
    o0_next <= '0'; ka_next <= '0';

  END CASE;
END PROCESS;
```




Modeliranje sekvencijskih sklopova...

```
PROCESS ( clock )  
  BEGIN  
    IF falling_edge(clock) THEN  
      state_present <= state_next;  
      o1 <= o1_next;  
      o0 <= o0_next;  
      ka <= ka_next;  
    END IF;  
  END PROCESS;  
  
END Behavioral;
```

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 8: Postupak projektiranja s osvrtom na jezik VHDL; Poglavlje 9: Sinkroni sekvencijski sklopovi.

- analiza sekvencijskih sklopova: str. 341-354
- vremenski odnosi: str. 376-379
- modeliranje sekvencijskih sklopova u jeziku VHDL: str. 327-330



Zadaci za vježbu (1)

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 9:
Sinkroni sekvencijski sklopovi.

- analiza sekvencijskih sklopova: 9.5-9.9, 9.14, 9.17, 9.18, 9.22, 9.23, 9.26
- modeliranje sekvencijskih sklopova u jeziku VHDL: 9.20



Zadaci za vježbu (2)

M. Čupić: *Digitalna elektronika i digitalna logika. Zbirka riješenih zadataka*, Cjelina 11: Strojevi s konačnim brojem stanja.

- modeliranje sekvencijskih sklopova u jeziku VHDL:
 - riješeni zadaci: 11.18-11.23
 - zadaci za vježbu: 2, 3, 8



11. Standardni sekvencijski moduli (1)



Sadržaj predavanja

- **registri**
 - **registri u užem smislu**
 - **posmačni registri**
- brojila



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*

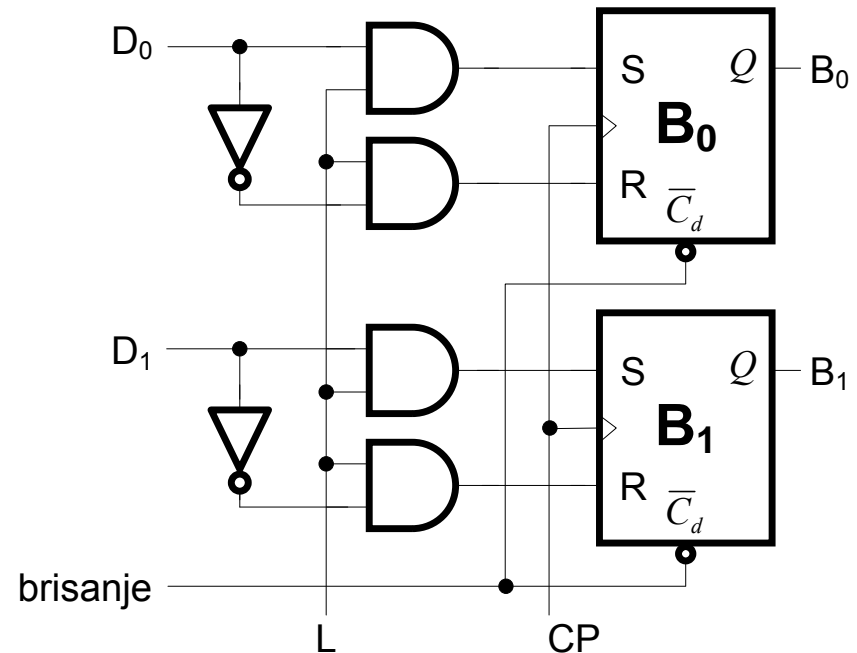


Registri

- *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
 - posmačni 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 (upravljanjih) 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}
-------	-------	-------	-----	-----------

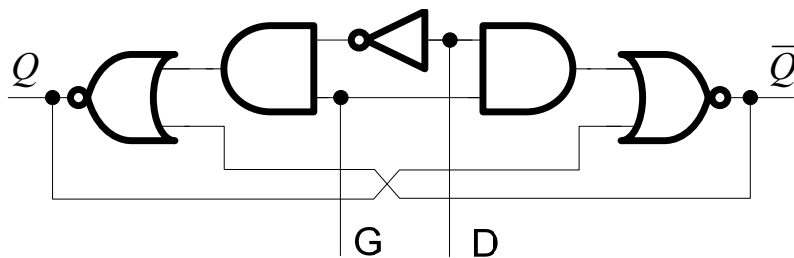
OP. KOD	ADRESA
---------	--------

- 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)

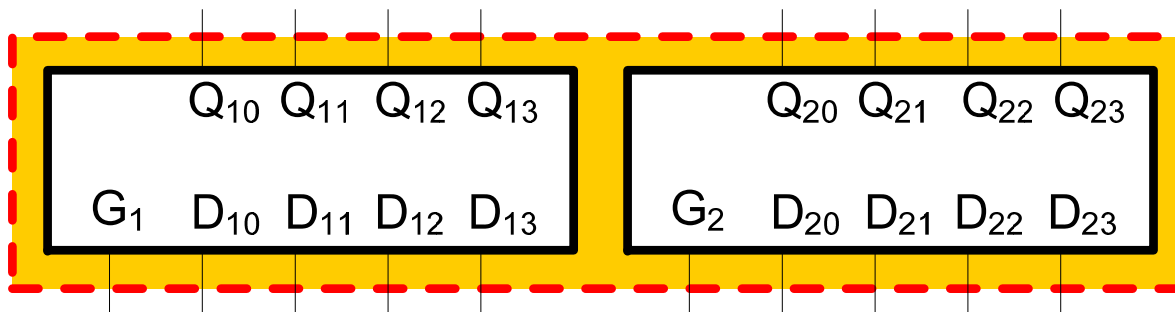
Registri u užem smislu

Primjer: "8-bit bistable latch" 74100

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

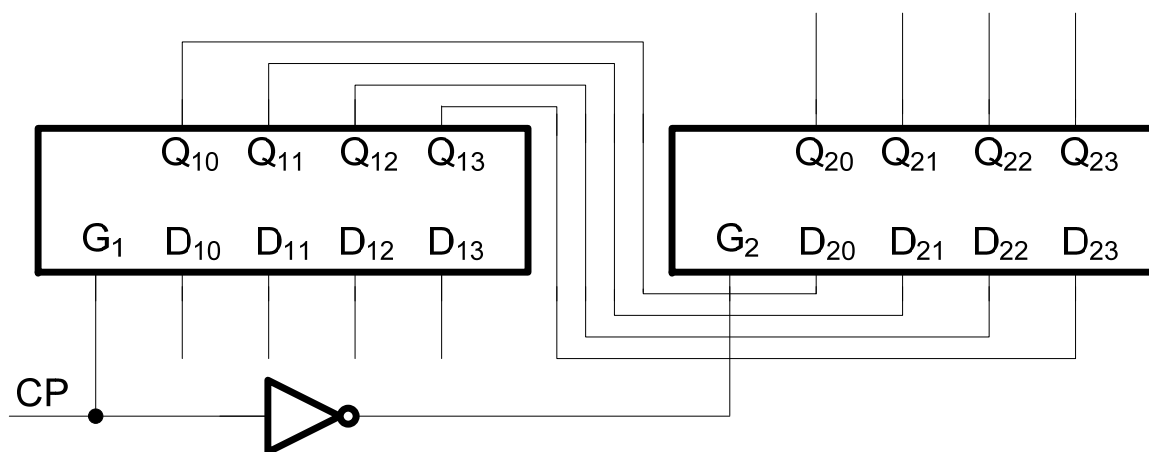


ULAZI		IZLAZI	
D	G	Q	\overline{Q}
L	H	L	H
H	H	H	L
X	L	Q^{n-1}	\overline{Q}^{n-1}



Registri u užem smislu

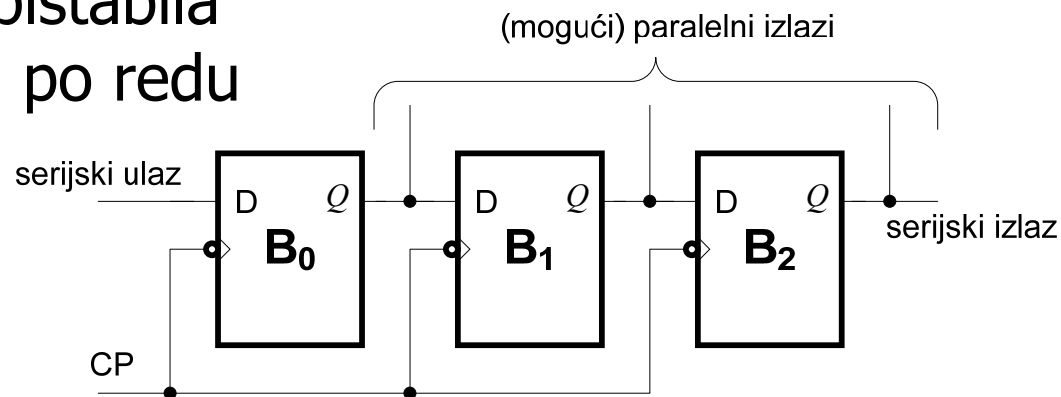
- primjena 74100:
 - privremeno pohranjivanje podataka na UI sustava ("međuspremnik", engl. buffer)
 - ostvarivanje složenijih struktura; npr. 4-bitni registar dvostrukih bistabila



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:*

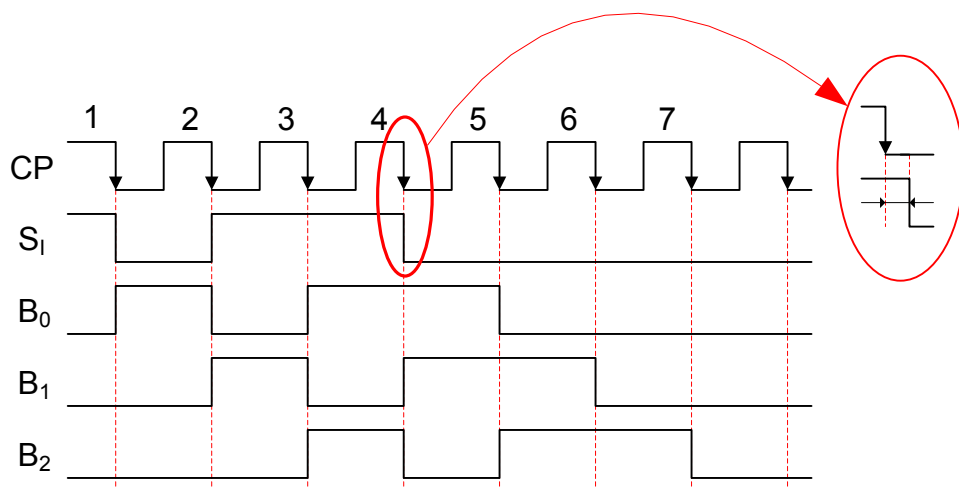
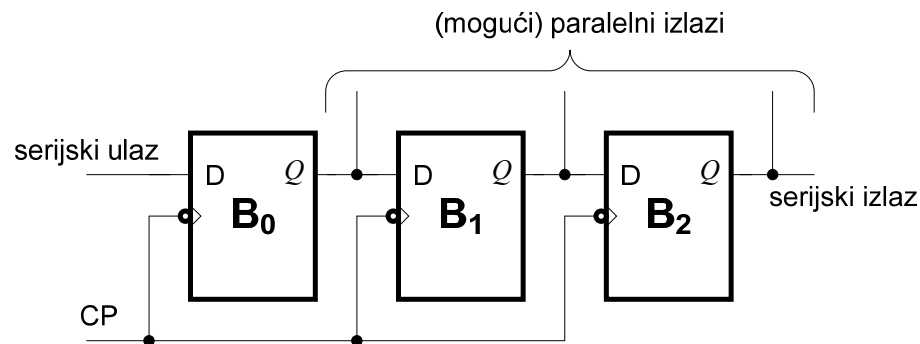
- *istovremeni upis:*

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

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

- *ispravnost upisa*

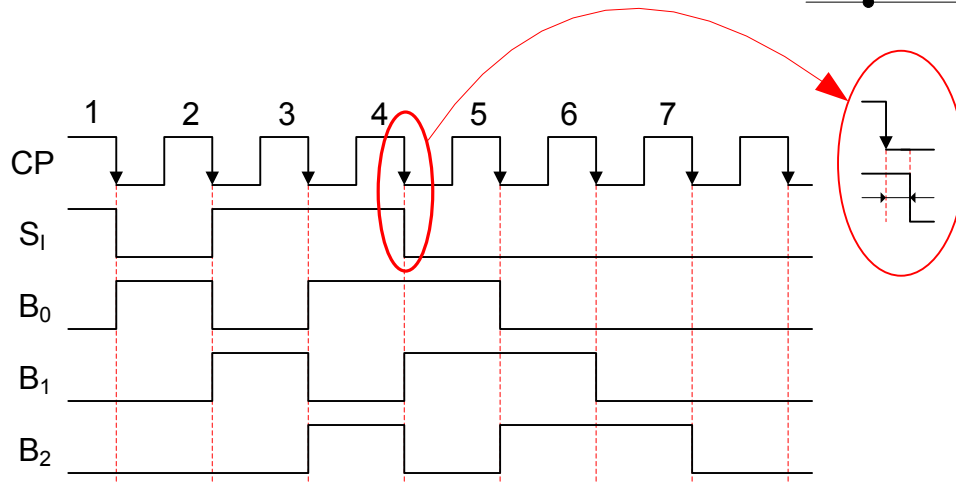
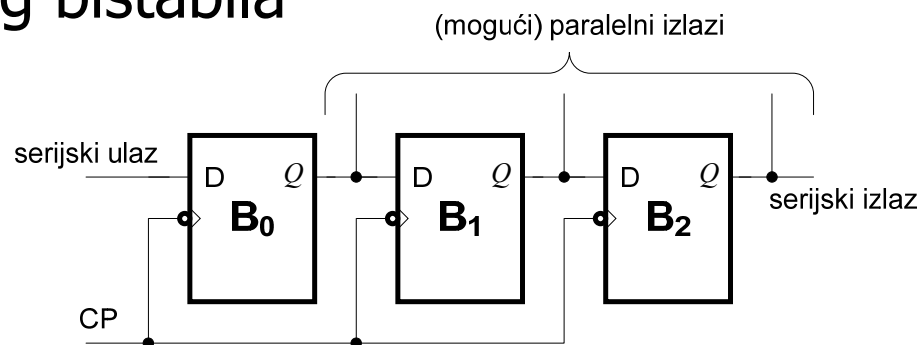
~ osigurati *kašnjenje* između bistabila



CP	S _I	S ₀ =B ₂			
		B ₀	B ₁	B ₂	S ₀
	1	0	0	0	0
1	0	1	0	0	0
2	1	0	1	0	0
3	1	1	0	1	1
4	0	1	1	0	0
5	0	0	1	1	1
6	0	0	0	1	1
7	0	0	0	0	0

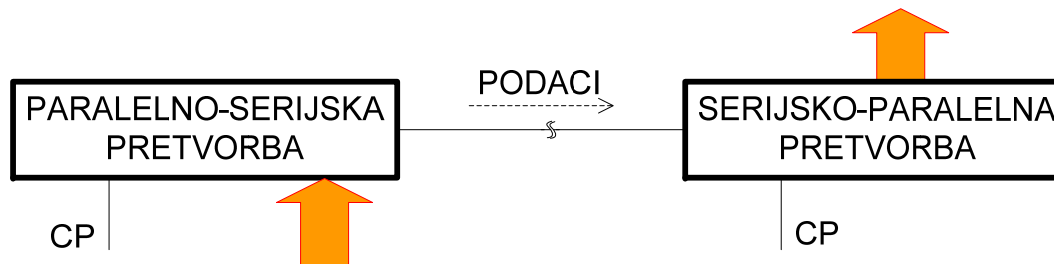
Posmačni registri

- 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*

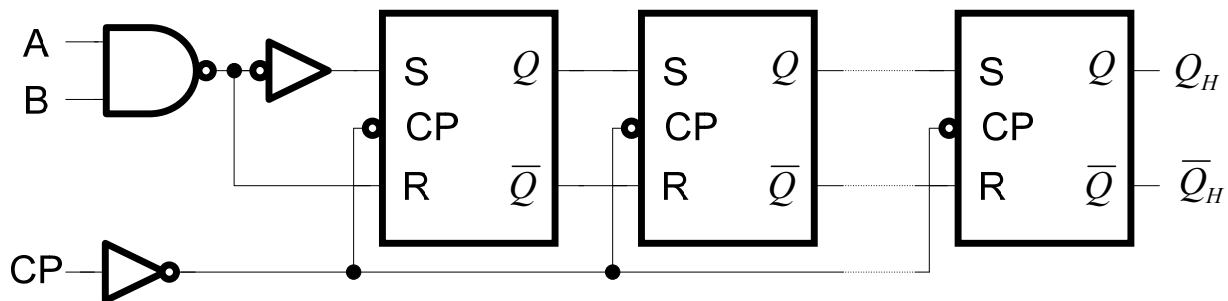
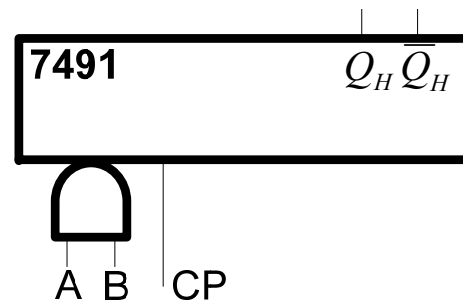


- serijski i paralelni ulaz i izlaz
~ *univerzalni* posmačni registar

Posmačni registri

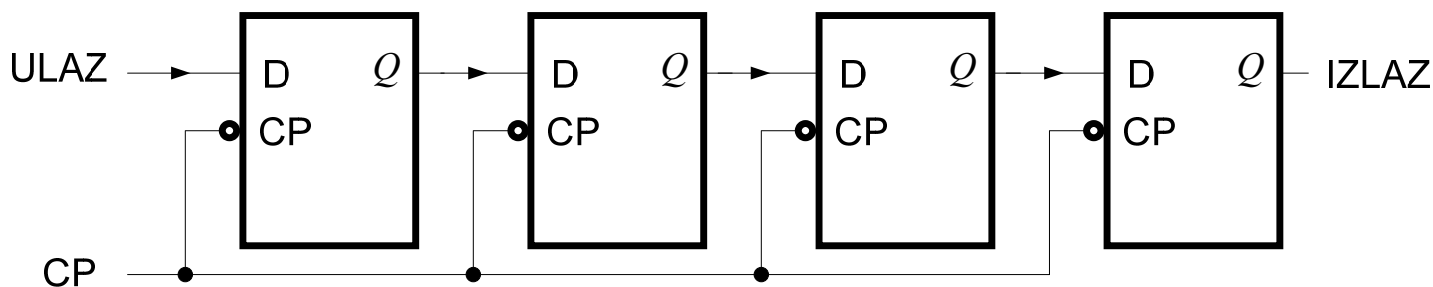
Primjer: posmačni registar 7491

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

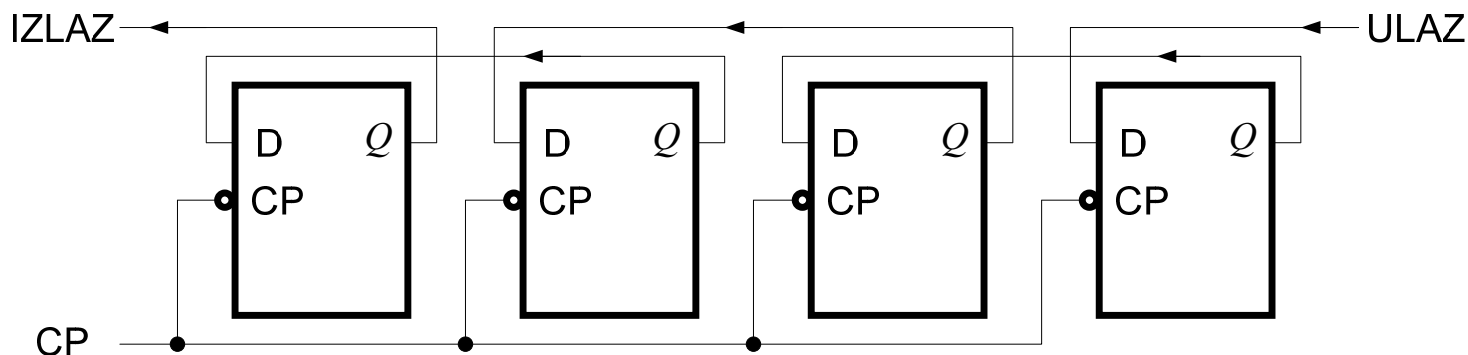


Posmačni registri

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

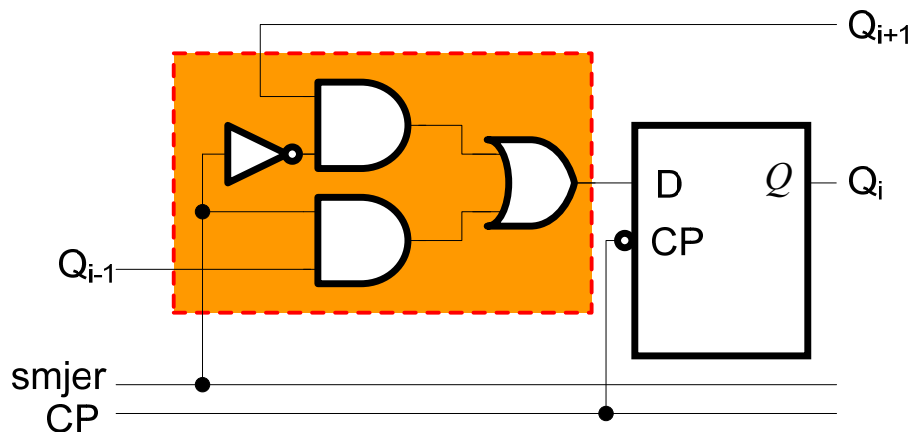


- moguće i "nalijevo", prema "normalnom" ulazu



Posmačni registri

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

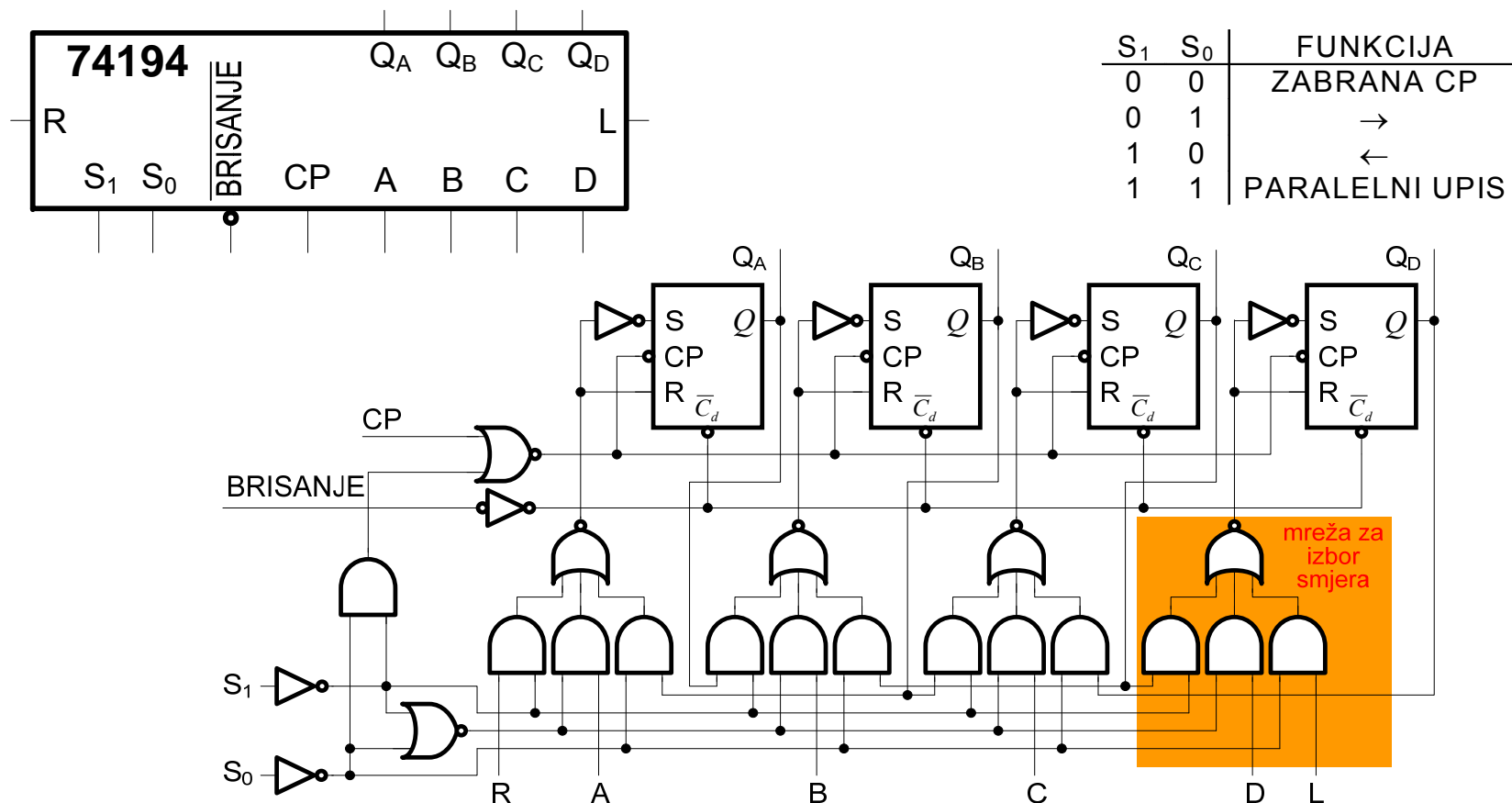


$$D_i = Q_{i-1} \cdot SMJER + Q_i \cdot \overline{SMJER}$$

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

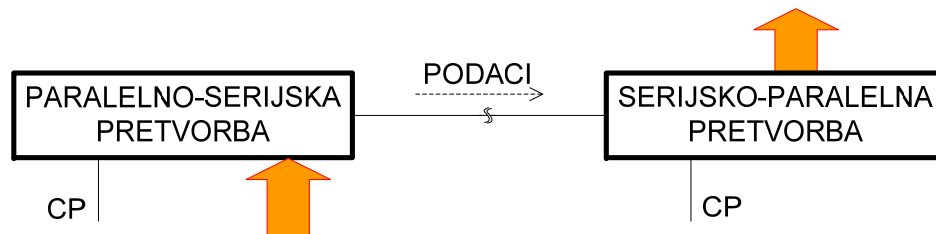
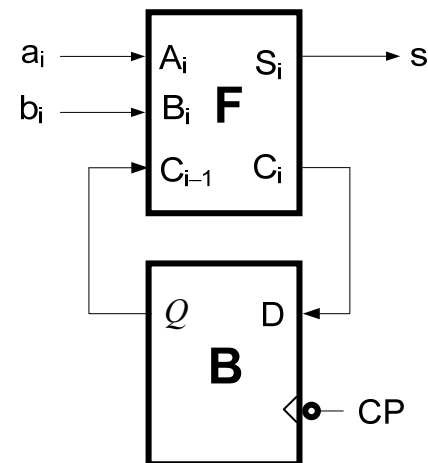
Posmačni registri

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



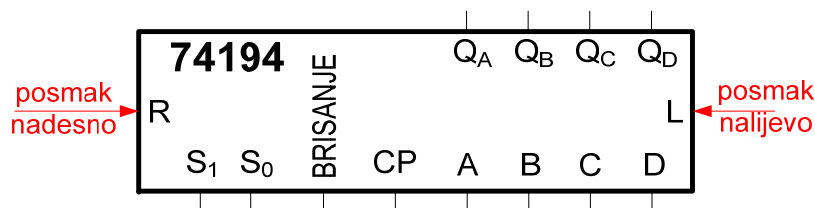
Posmačni registri

- primjene (1):
 - memoriranje podataka za *serijsko* izvršavanje (aritmetičkih) operacija; npr. *serijsko* binarno zbrajalo
 - pretvorba oblika podataka:
 - serijsko-paralelna (\exists paralelni izlazi)
 - paralelno-serijska (\exists paralelni ulazi)



Posmačni registri

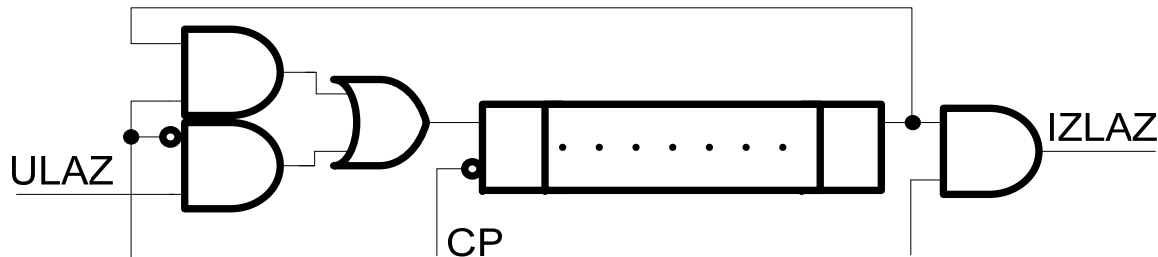
- primjene (2):
 - ostvarivanje (aritmetičkih) operacija:
 - množenje s 2: posmak nalijevo
 - dijeljenje s 2: posmak nadesno



- brojanje (\rightarrow posmačni registar u funkciji brojila)

Posmačni registri

- primjene (3):
 - sinkronizacija brzina prijenosa
~ "gladenje" prometa ($f_1 \neq f_2$)
 - upis podataka s f_1
 - ispis podataka s f_2
 - generiranje "pseudo-slučajnog" slijeda
~ *generatori sekvencije*:
linijski kodovi, kriptiranje
 - izvedbe cirkulirajućih memorija
~ npr. generatori znakova





Sadržaj predavanja

- registri
- **brojila**
 - **asinkrona brojila**
 - **binarno brojilo**
 - **reverzno i brojilo naprijed-natrag**
 - **brojilo modulo m**
 - sinkrona brojila
 - brojila na osnovi posmačnog registra

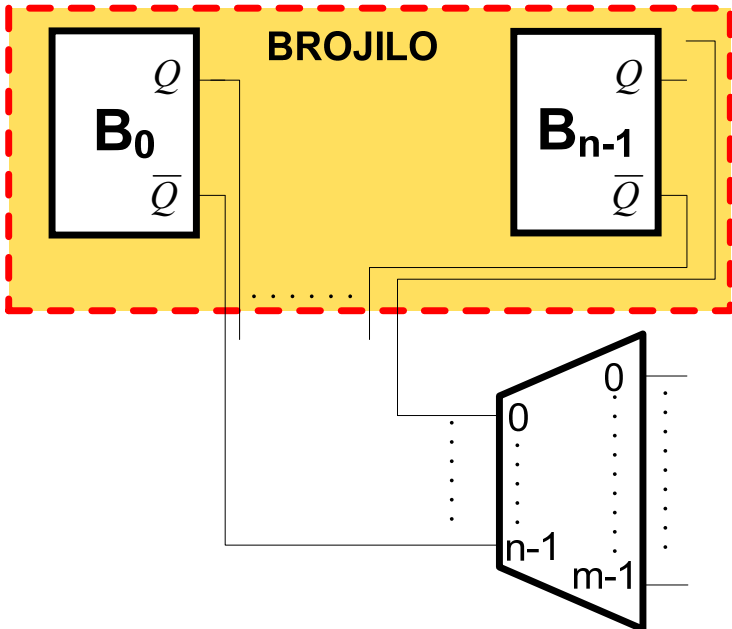
- *brojilo*:
 - ~ pod utjecajem ulaznih impulsa (obično CP) prolazi kroz *utvrđeni niz* stanja i *vraća se u početno* stanje
 - sklop "broji" ulazne impulse
 - impulsi *ne moraju* biti periodički ($f \neq \text{const.}$)
 - "autonomni" sekvencijski sklop
 - ~ samo jedan ulaz, i to obično za CP
 - definicije:
 - *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!)
 $2^{n-1} = N/2 < m \leq 2^n = N$

- 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. scalars)
 - ~ važan samo *broj* stanja,
 - ne* i redoslijed njihove izmjene

- prikladno projektiranje brojila
~ jednostavniji dekodler
- važna primjena
~ generator upravljačkih impulsa digitalnog sustava

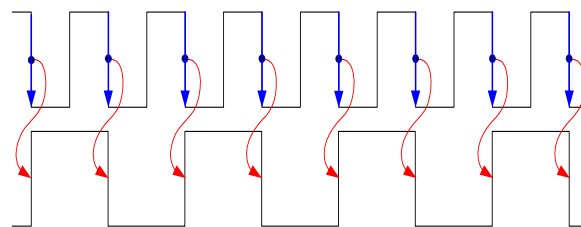
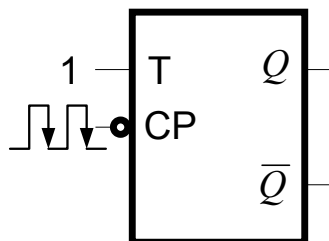


- *djelitelji frekvencije:*
 - sklop samo broji ulazne impulse
 - očitati *samo ono* stanje koje definira željeni izlazni impuls
~ nakon svakih n impulsa, od nekog početnog
 - pojednostavljivanje dekodera
~ *nepotpuno* dekodiranje
 - ubrzanje rada: f_{\max} ↗
npr. naročito za asinkrona brojila

- *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

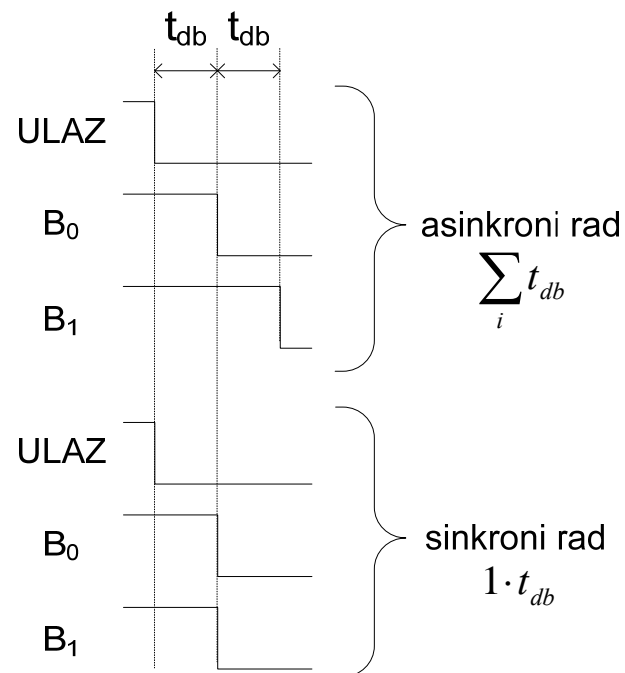
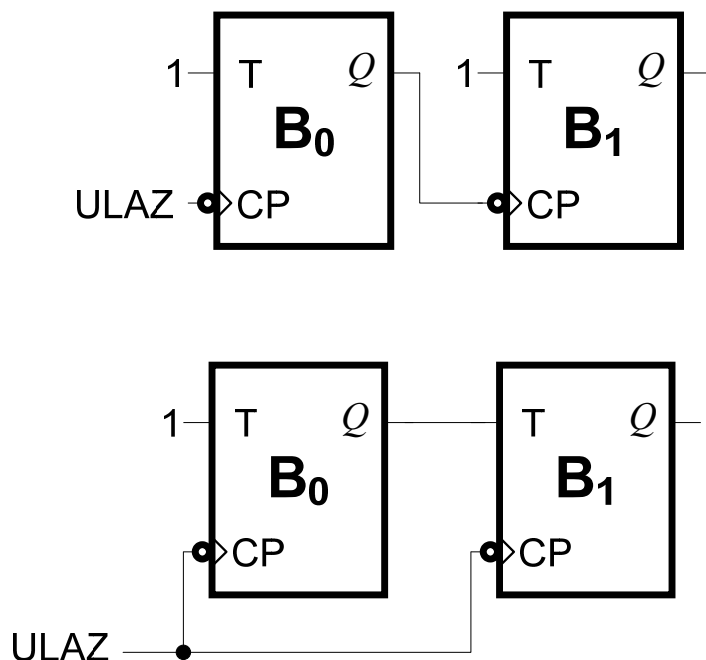


- direktna implementacija *asinkronih* brojila
 - ~ niz bistabila od kojih svaki prethodni pobuđuje naredni u nizu
- brojanje u *binarnom* brojevnom sustavu
 - ~ 2^n stanja za n bistabila:
binarno brojilo (bistabili $\sim 2^i$: težine potencije od 2)

Asinkrona brojila

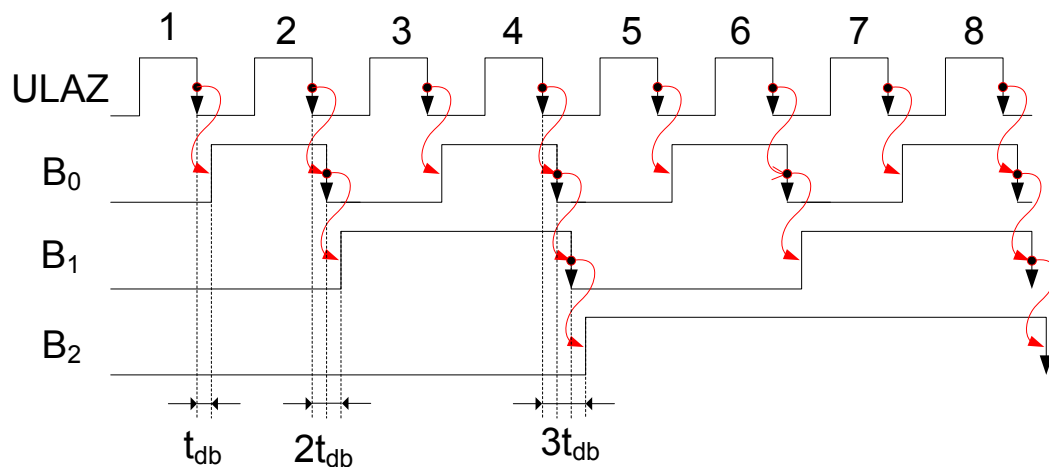
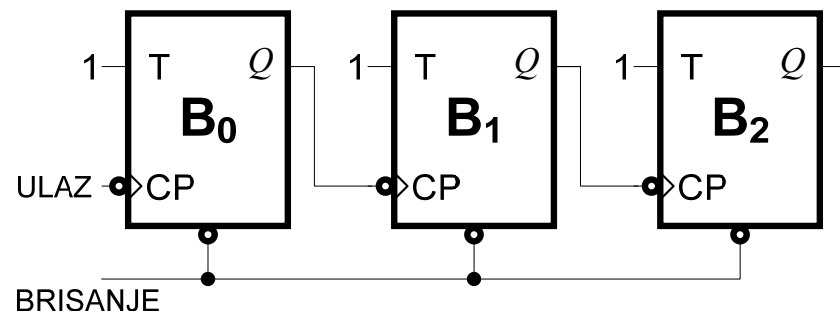
- *asinkrona brojila*

~ 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 ₂	B ₁	B ₀
↑ ↓<				

Asinkrona brojila

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

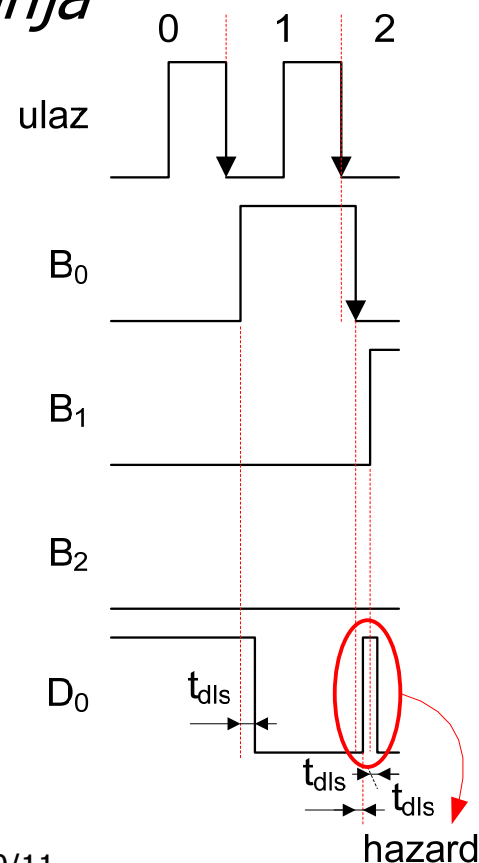
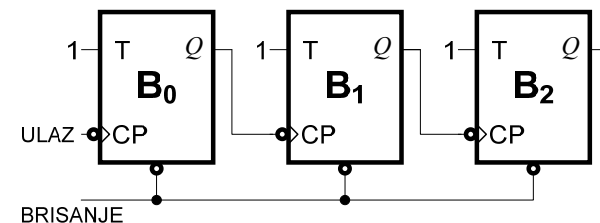
- *serijsko* okidanje bistabila:
~ *tranzijentna pogreška dekodiranja*
(→ hazard)
- dekodiranje svih 2^n stanja
~ *potpuno dekodiranje*;
npr. dekodiranje D_0

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

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

⋮

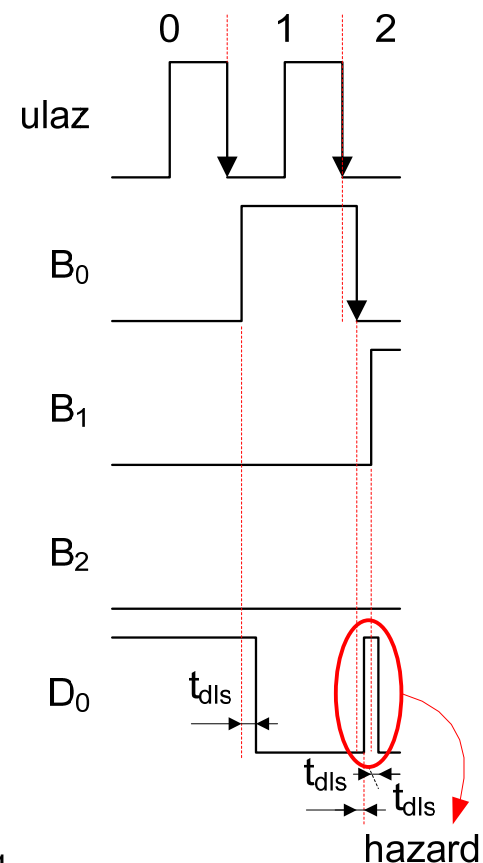
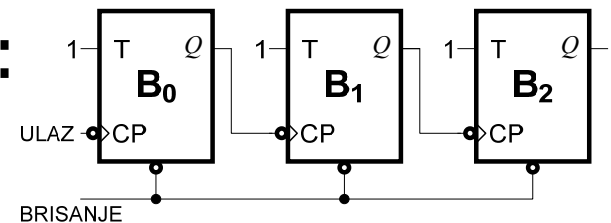
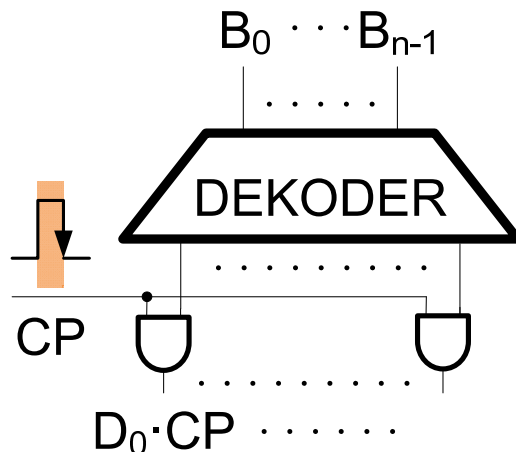
$$D_7 = B_2 B_1 B_0$$



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

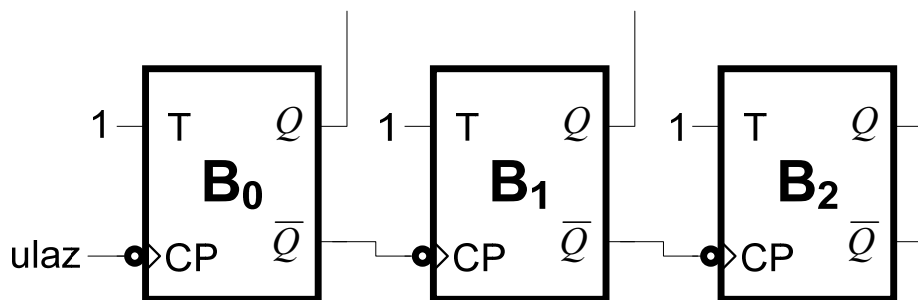
- maksimalna frekvencija *brojila u užem smislu*:
 - očitavanje (= dekodiranje) svih stanja!
 - najlošiji slučaj
~ B_0 *ne smije* promijeniti stanje sve dok B_{n-1}
ne dođe u stanje uzrokovano *prethodnim* impulsom

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

- 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
~ $Q_i: 0 \rightarrow 1$

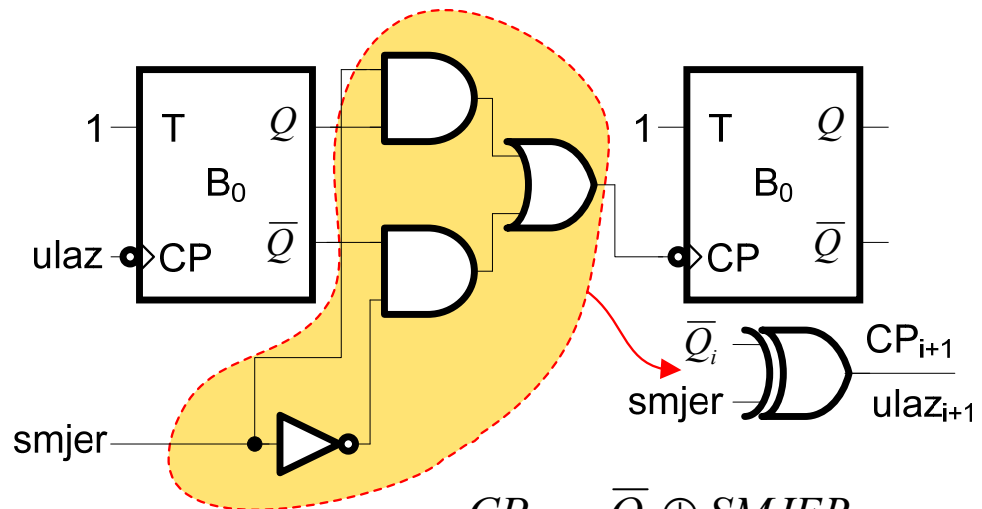


ULAZ	B ₂	B ₁	B ₀
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

Asinkrona brojila

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

- konceptualna implementacija:



- primjena:

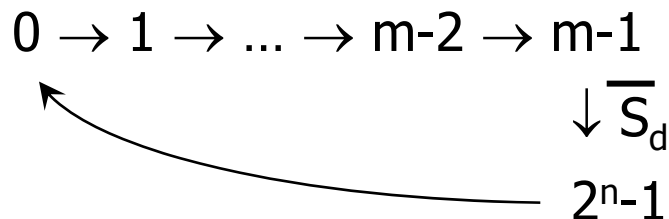
- digitalno upravljanje
- obavljanje jednostavnih aritmetičkih operacija *nad impulsima*

$$\begin{aligned} CP_{i+1} &= \bar{Q}_i \oplus SMJER \\ &= \bar{Q}_i \cdot \overline{SMJER} + Q_i \cdot SMJER \end{aligned}$$

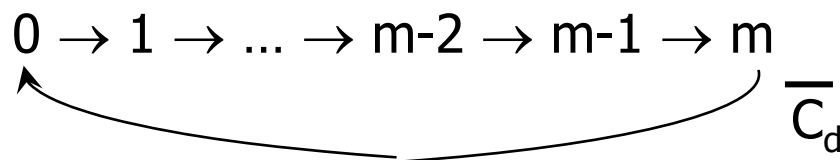
$$SMJER = \begin{cases} 0: \text{brojanje natrag} \\ 1: \text{brojanje naprijed} \end{cases}$$

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 $0 \bmod 2^n$



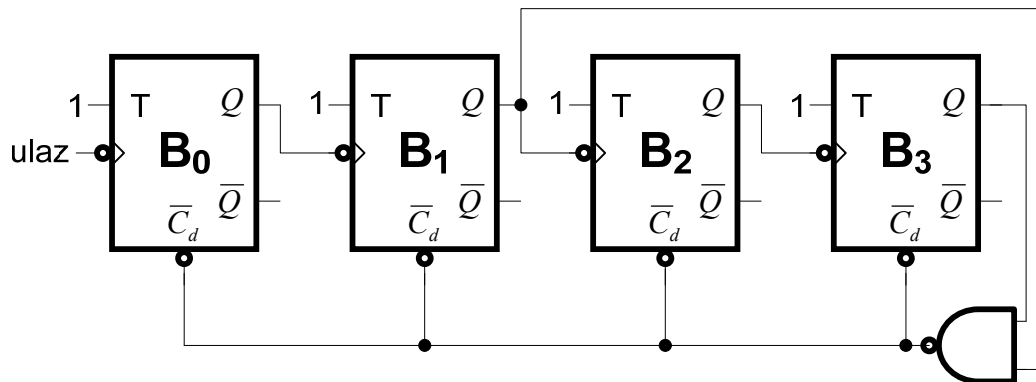
- prekid aktiviran *prvim stanjem izvan* ciklusa, m
~ \overline{C}_d prebacuje brojilo u stanje 0 :



Asinkrona brojila

Primjer: dekadsko brojilo

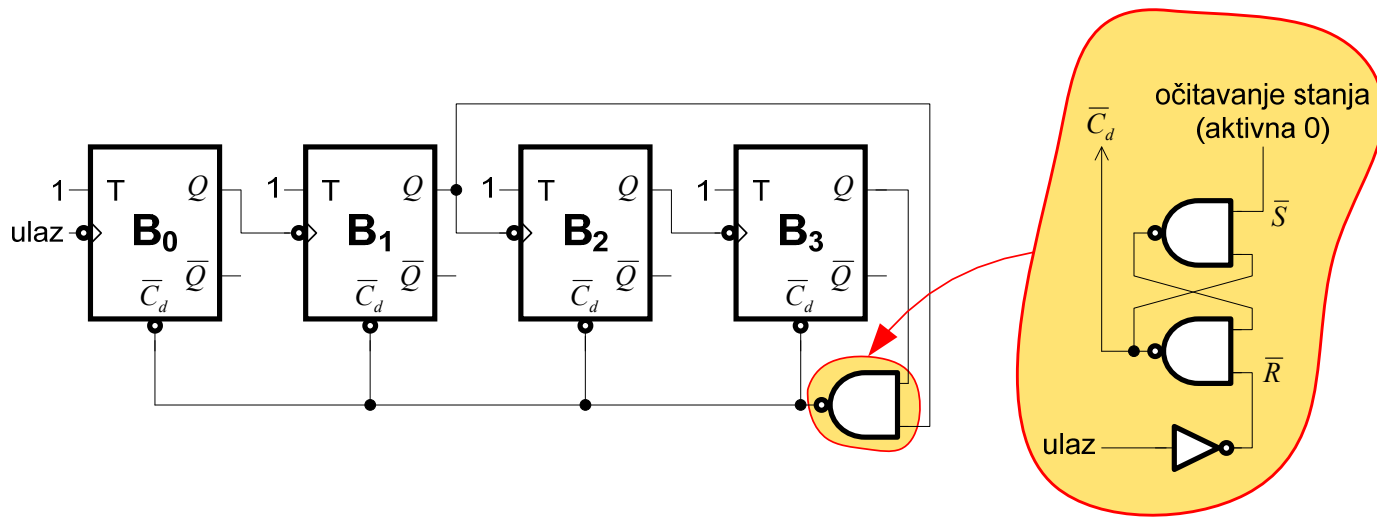
- detektirati karakterističnu pojavu $B_3B_1 = 1$
(\rightarrow brojilo broji *naprijed*: jednostavno dekodiranje)
- problem kod brisanja bistabila
 \sim *rasipanje* t_{db} :
nestanak impulsa brisanja *prije* brisanja svih bistabila!



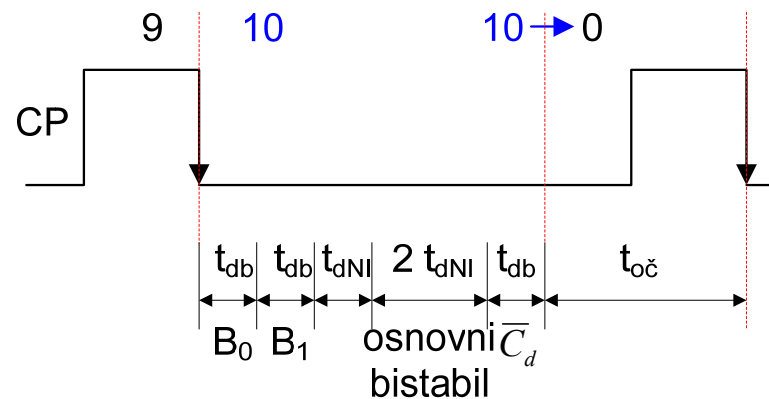
CP	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10≡0	1	0	1	0
	0	0	0	0

Asinkrona brojila

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

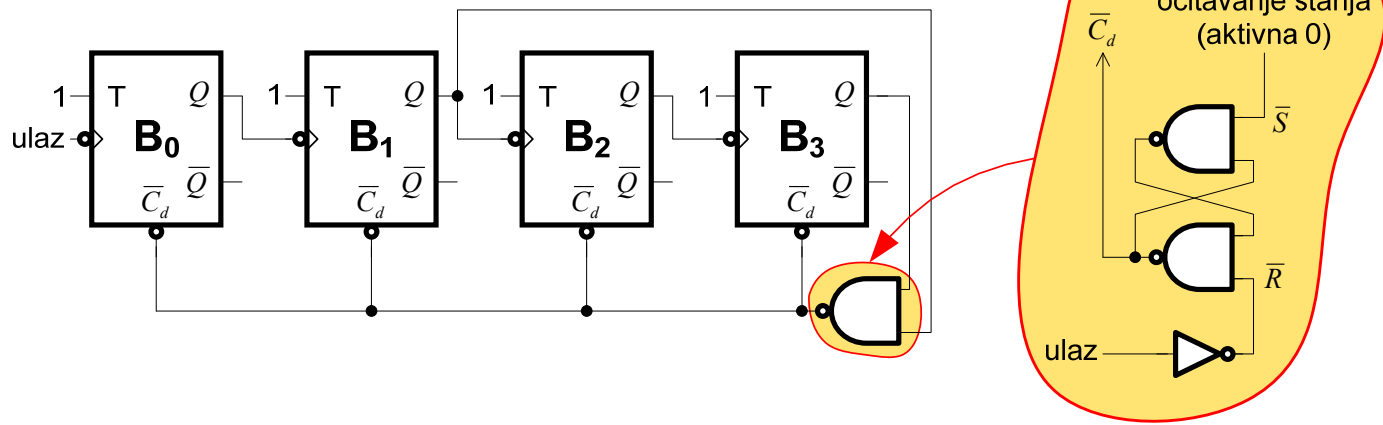


- sigurno generiranje impulsa brisanja
~ traje do slijedećeg CP = 1

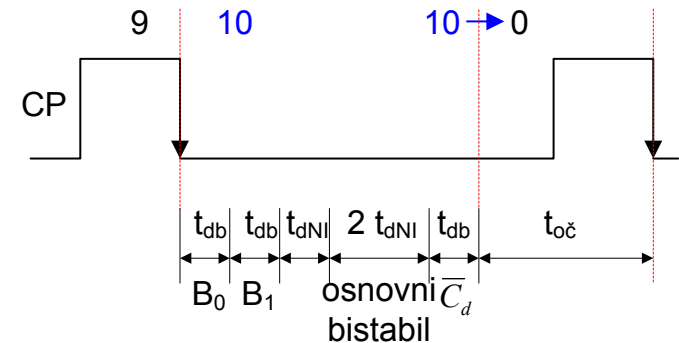


Asinkrona brojila

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



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



U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 11:
Sekvencijski moduli: registri i brojila.

- registri: str. 414-422
- asinkrona brojila: str. 435-440



Zadaci za vježbu (1)

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 11:
Sekvencijski moduli: registri i brojila.

- registri: 11.1, 11.2, 11.10, 11.11, 11.22, 11.24, 11.26, 11.27, 11.35
- modeliranje u VHDL: 11.23, 11.28, 11.32, 11.33
- asinkrona brojila: 11.36—11.41



Zadaci za vježbu (2)

M. Čupić: *Digitalna elektronika i digitalna logika.*
Zbirka riješenih zadataka, Cjelina 9: Registri; Cjelina
10: Brojila.

- registri:
 - riješeni zadaci: 9.1-9.3, 9.5-9.15
 - zadaci za vježbu: 1, 2
- asinkrona brojila:
 - riješeni zadaci: 10.1, 10.7
 - zadaci za vježbu: 1, 4



11. Standardni sekvencijski moduli (2)



Sadržaj predavanja

- **brojila**
 - **sinkrona brojila**
 - **binarno brojilo**
 - **brojilo naprijed-natrag**
 - **brojilo modulo m**
 - brojila na osnovi posmačnog registra
- generatori sekvencije

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:

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

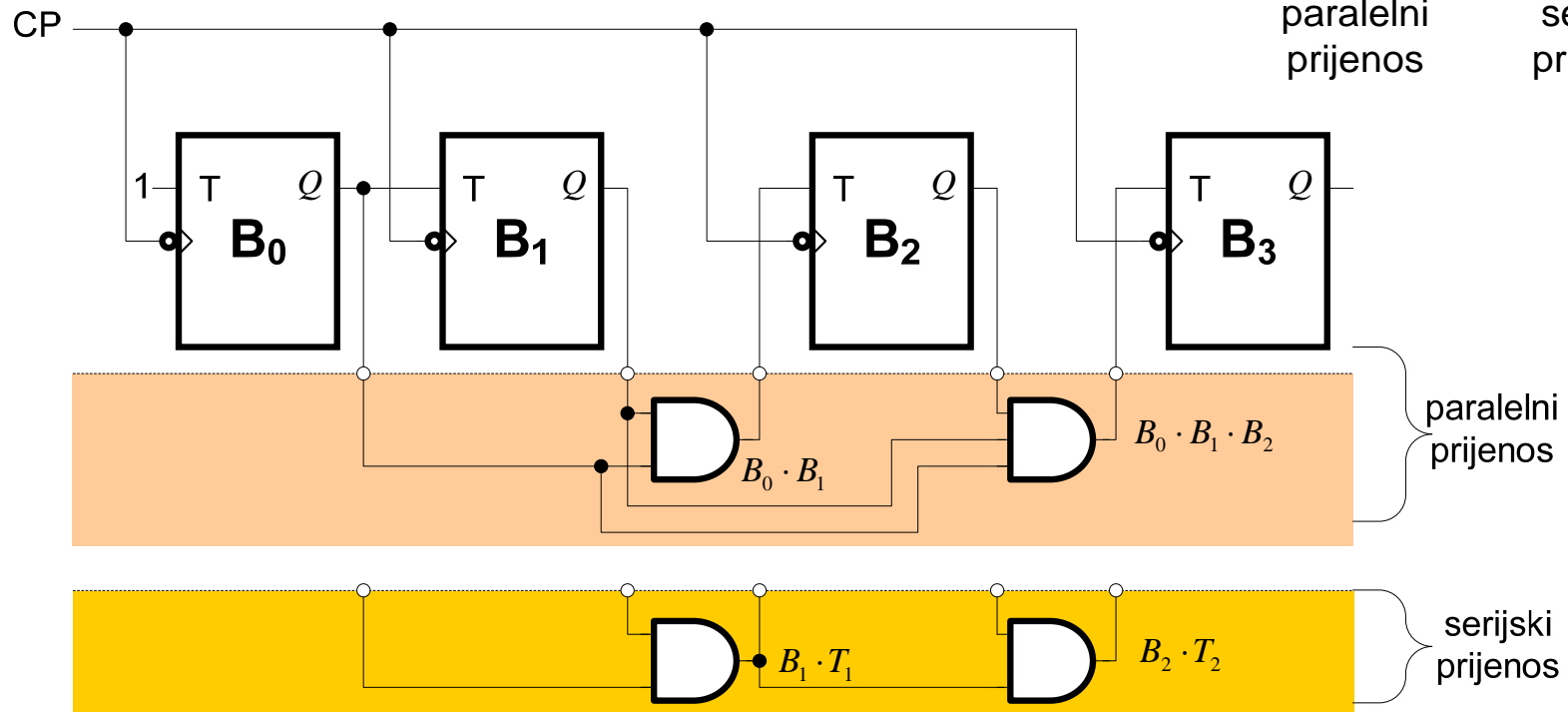
CP	B ₂	B ₁	B ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Sinkrona brojila

- struktura n -bitnog binarnog sinkronog brojila:

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

paralelni prijenos
serijski prijenos



-

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

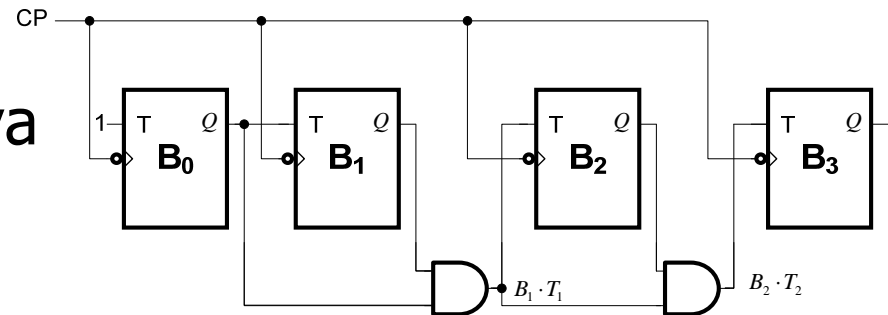
- za $n \nearrow$ izvedba je kontraproduktivna
~ teškoće pri ostvarivanju I-sklopa, $C_{\text{rasipno}} \nearrow$, itd.

Sinkrona brojila

- binarno sinkrono brojilo sa *serijskim* prijenosom:

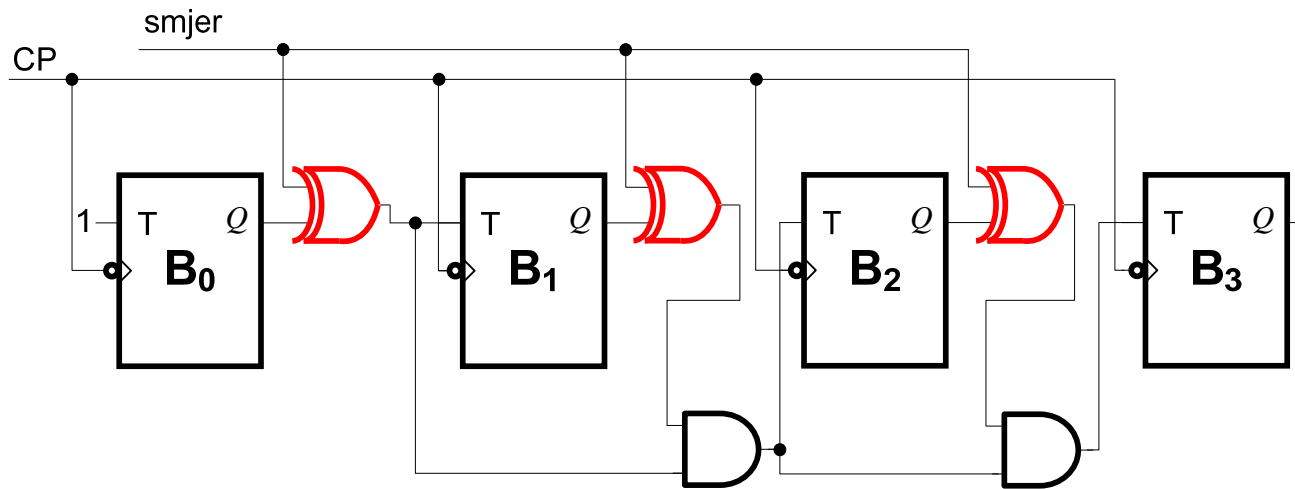
- *kaskadiranje* I-sklopova
- jeftinije rješenje
~ *istovrsni* sklopovi
s ograničenim brojem ulaza
(i to samo 2!)
- sporije rješenje:

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



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})

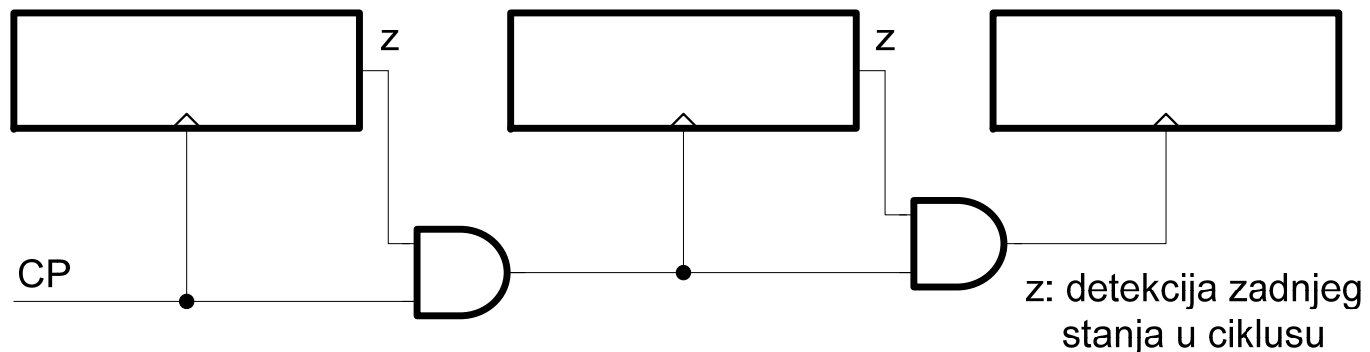


Sinkrona brojila

- *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





Sadržaj predavanja

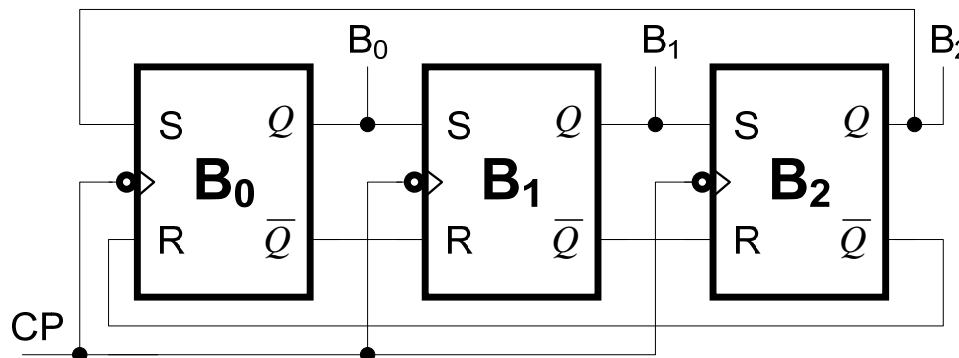
- **brojila**
 - sinkrona brojila
 - **brojila na osnovi posmačnog registra**
 - **prstenasto brojilo**
 - **Johnsonovo brojilo**
- generatori sekvencije

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



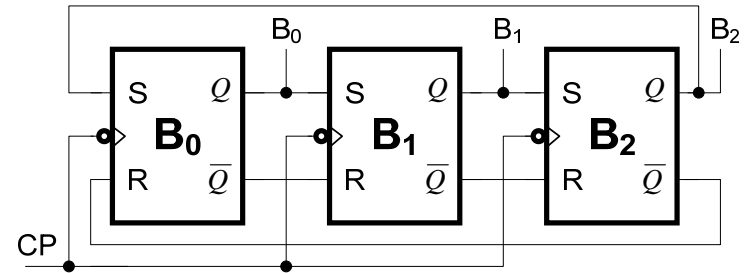
CP	B ₀	B ₁	B ₂
0	1	0	0
1	0	1	0
2	0	0	1
3	1	0	0

- 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*!



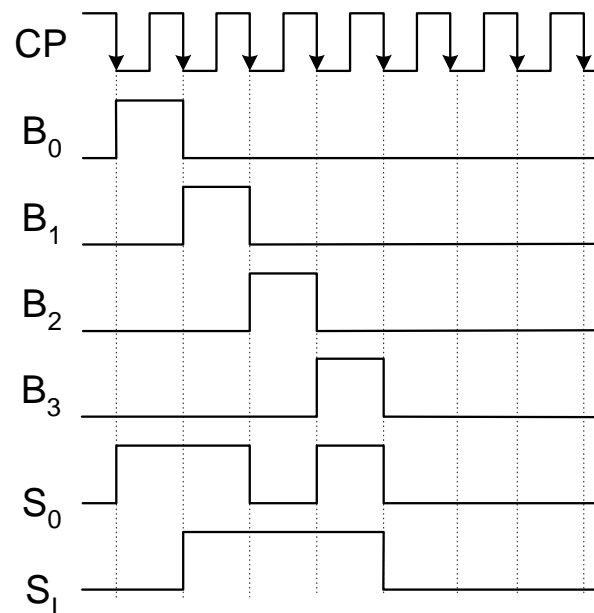
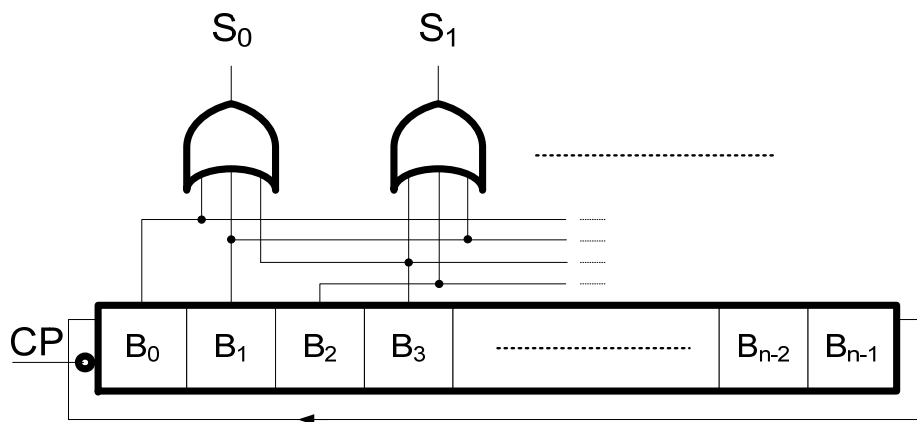
Sinkrona brojila

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

Primjer:

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

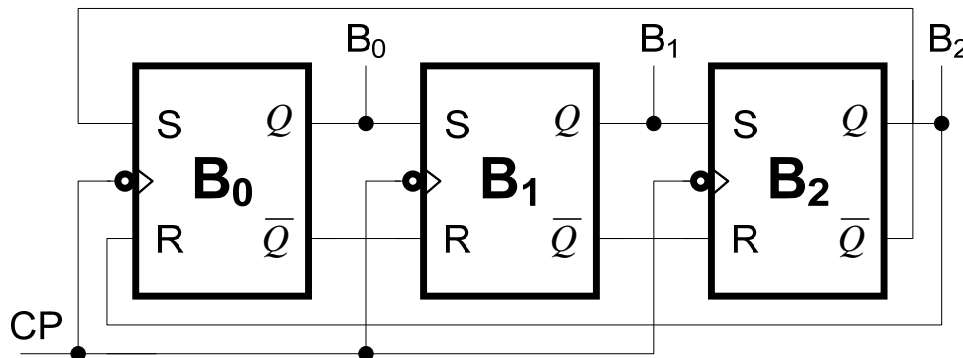
$$S_1 = B_1 + B_2 + B_3 + \dots$$



Sinkrona brojila

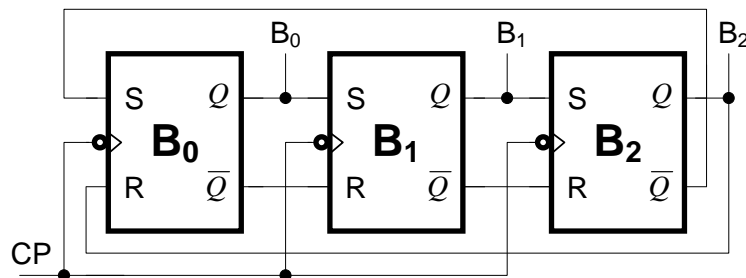
- *Johnsonovo brojilo*, brojilo s ukrštenim prstenom (engl. twisted ring counter):
 - povećanje broja stanja za dani broj bistabila: $2 \cdot n$
 - ukrstiti povratnu vezu
~ bistabili SR i JK
 - na ulaz dovesti \overline{Q}_{n-1}
~ bistabil D
 - broje u kodu s $d_{\min} = 1$
 - i dalje brže od binarnog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

CP	B 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



Sinkrona brojila

- *dekodiranje stanja* Johnsonovog brojila:
 - nije tako povoljno kao kod prstenastog brojila
 - ipak relativno jednostavno
~ 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

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

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

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

$$3: B_2 B_0$$

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

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



Sadržaj predavanja

- brojila
- **generatori sekvencije**

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:

...**0111001**01110010111...



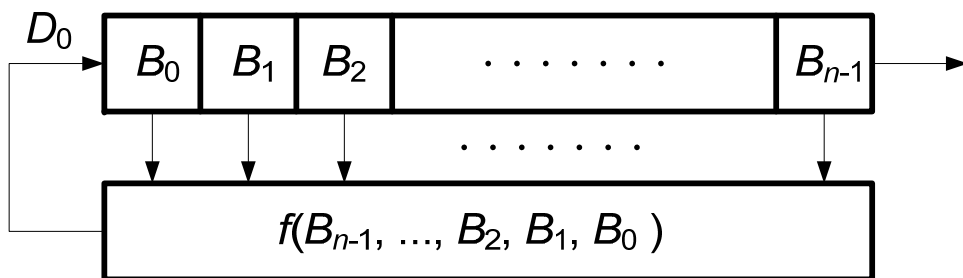
B_0	B_1	B_2	
B_{n-4}	B_{n-3}	B_{n-2}	B_{n-1}
1	1	1	0
0	1	1	1
0	0	1	1
1	0	0	1
0	1	0	0
1	0	1	0
1	1	0	1
1	1	1	0

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

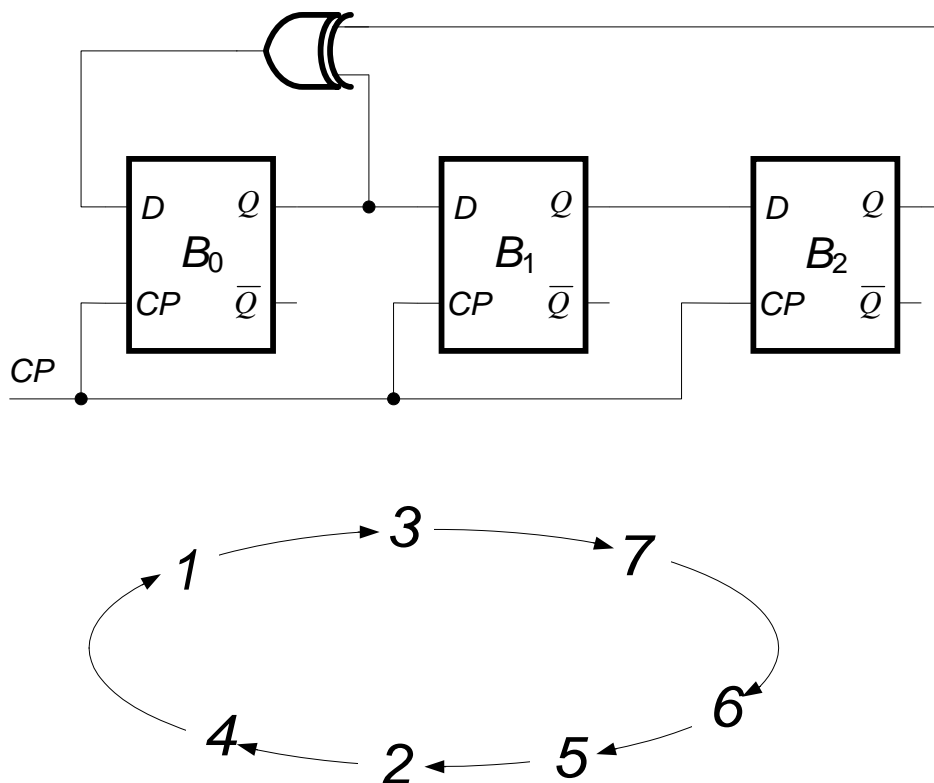
- naročito jednostavna izvedba povratne veze
~ *linearna* funkcija:

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

- *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)

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 11:
Sekvencijski moduli: registri i brojila.

- sinkrona brojila: str. 426-434
- brojila na osnovi posmačnog registra: str. 422-425
- generatori sekvencije: str. 441-451



Zadaci za vježbu (1)

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 11:
Sekvencijski moduli: registri i brojila.

- sinkrona brojila: 11.3-11.8, 11.13-11.17, 11.19-11.21, 11.29
- brojila na osnovi posmačnog registra: 11.9, 11.12, 11.26, 11.31;
- modeliranje u VHDL: 11.30
- generatori sekvencije: 11.18, 11.34



Zadaci za vježbu (2)

M. Čupić: *Digitalna elektronika i digitalna logika.*
Zbirka riješenih zadataka, Cjelina 9: Registri; Cjelina 10: Brojila; Cjelina 11: Strojevi s konačnim brojem stanja.

- sinkrona brojila:
 - riješeni zadaci: 10.3-10.6, 10.9, 10.10; 11.10, 11.11, 11.16
 - zadaci za vježbu: 2, 3
- brojila na osnovi posmačnog registra:
 - riješeni zadaci: 9.4
 - zadaci za vježbu: 3, 4
- generatori sekvencije:
 - riješeni zadaci: 11.17



11. Standardni sekvencijski moduli (3)



Sadržaj predavanja

- **osnovne karakteristike memorija**
- statičke memorije
- dinamičke memorije



Osnovne karakteristike memorija

- 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*



Osnovne karakteristike memorija

- 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

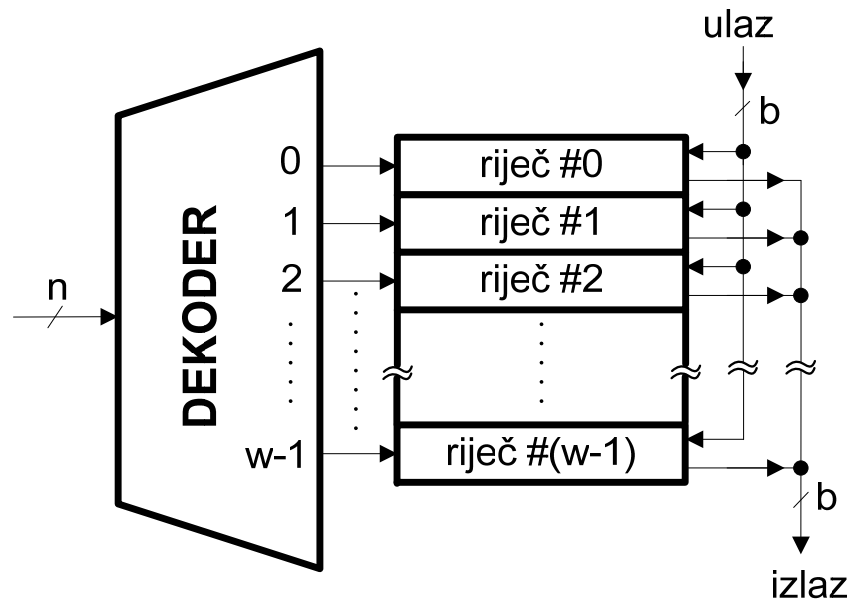


Osnovne karakteristike memorija

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

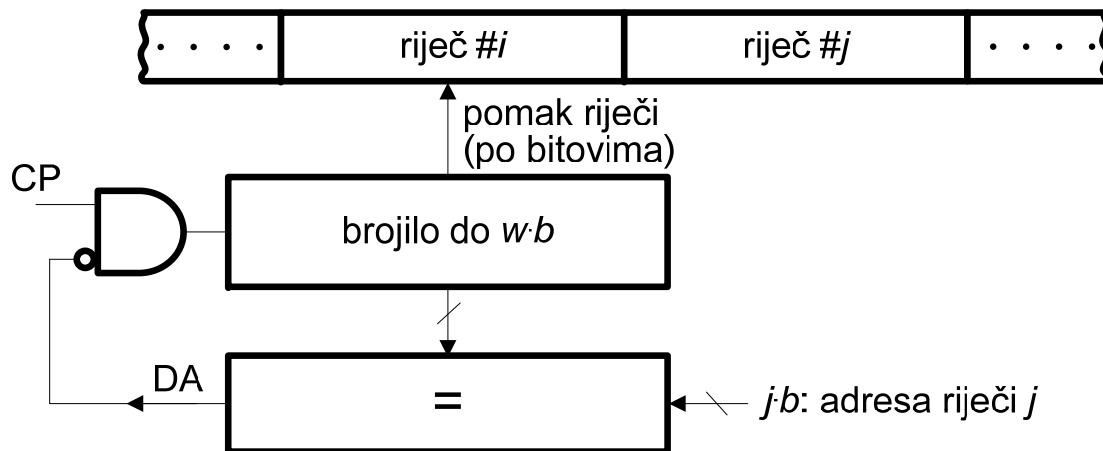
Osnovne karakteristike memorija

- *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
$$W = 2^n \rightarrow a_{n-1} \dots a_1 a_0$$



Osnovne karakteristike memorija

- *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





Osnovne karakteristike memorija

- *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

Osnovne karakteristike memorija

- *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 ($t_a \sim 5 \text{ ms}$)
 - uređaji
~ diskovi, trake



Osnovne karakteristike memorija

- 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 (kapacitet ~ 1 MB)
 - sekundarna p. m. (engl. L2 cache):
 - ~ na M/B (kapacitet ~ 16 MB)

Osnovne karakteristike memorija

- 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. $5 \text{ ns} < t_a \text{ tipično} < 70 \text{ ns}$ (suvremeni DRAM)
 - *prosječno vrijeme pristupa*, $\overline{t_a}$:
~ za *sekvencijalne* memorije
$$\overline{t_a} = (t_a)_{\max} / 2$$

Osnovne karakteristike memorija

- 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)



Sadržaj predavanja

- osnovne karakteristike memorija
- **statičke memorije**
 - **organizacija memorijskih modula**
 - **vremenski odnosi**
 - **projektiranje statičkih memorija**
- dinamičke memorije



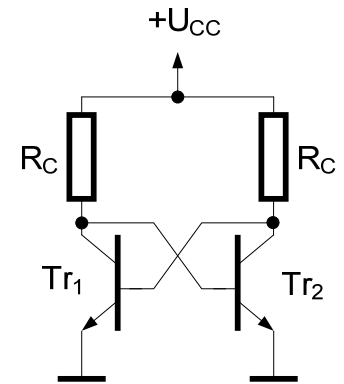
Statičke memorije

- 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

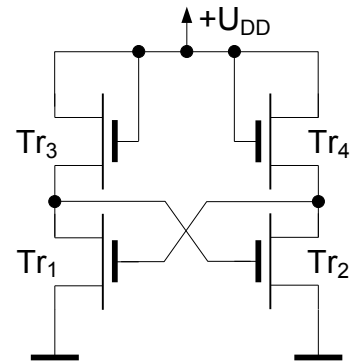
Statičke memorije

- princip izvedbe memorijskog bistabila:

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



- izvedba s MOSFETom:
~ NMOS

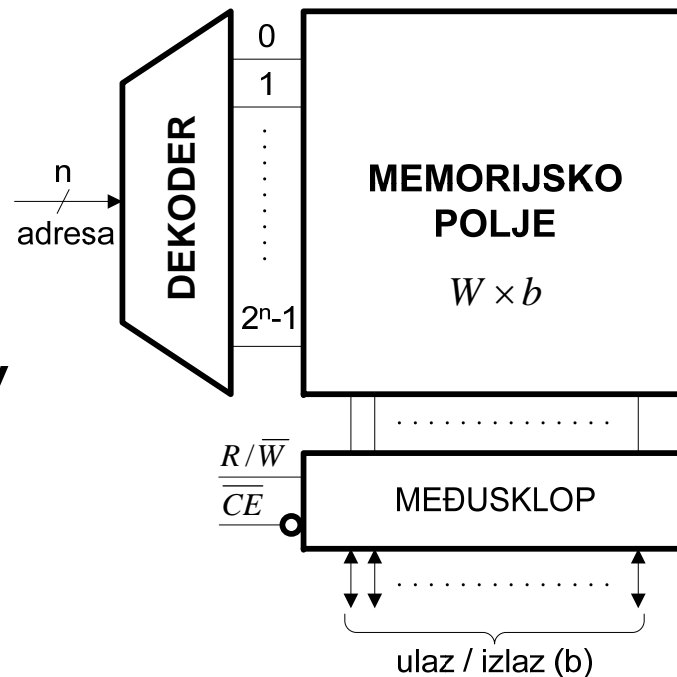
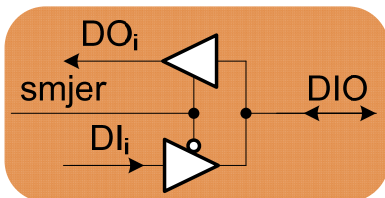
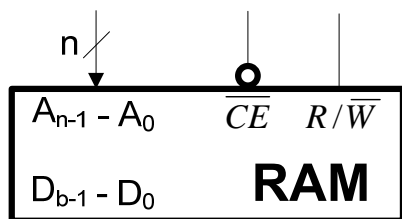


	bipolarne	MOSFET
kapacitet	manji	veći
brzina	veća	manja
cijena	veća	manja

Organizacija memorijskih modula

- *organizacija* memorijskih modula
~ karakteristični funkcijski blokovi:

- dekodeer adrese
- memorijsko polje
- međusklop:
~ pojačala za čitanje/pisanje:
konverzija električkih razina,
dvolinijskih reprezentacija



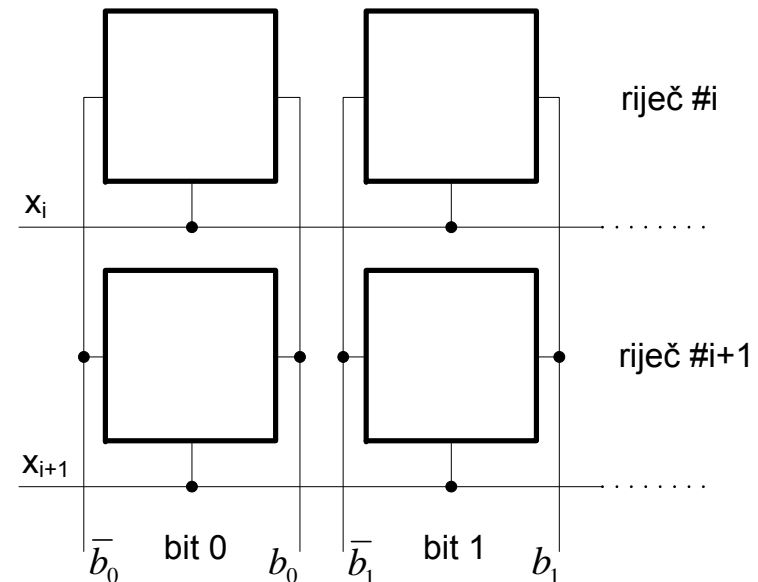


Organizacija memorijskih modula

- 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



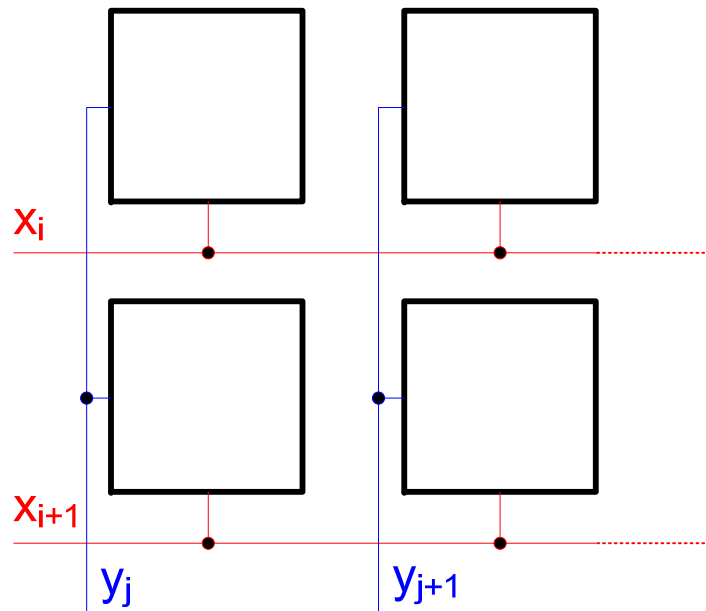


Organizacija memorijskih modula

- 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*!

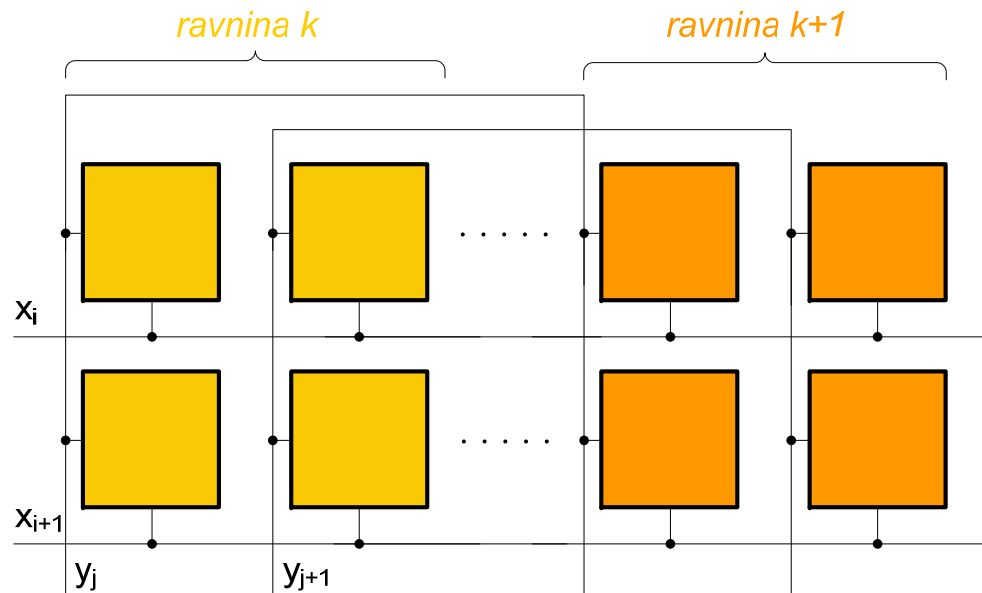
Organizacija memorijskih modula

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



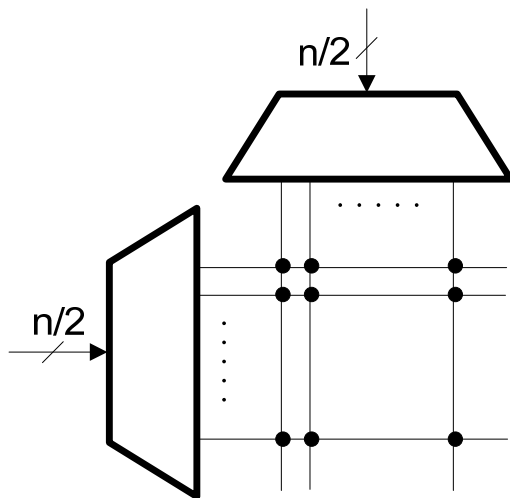
Organizacija memorijskih modula

- *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:
 $2^{n/2} + 2^{n/2} = 2 \cdot 2^{n/2} = 2\sqrt{2^n} = 2\sqrt{W}$

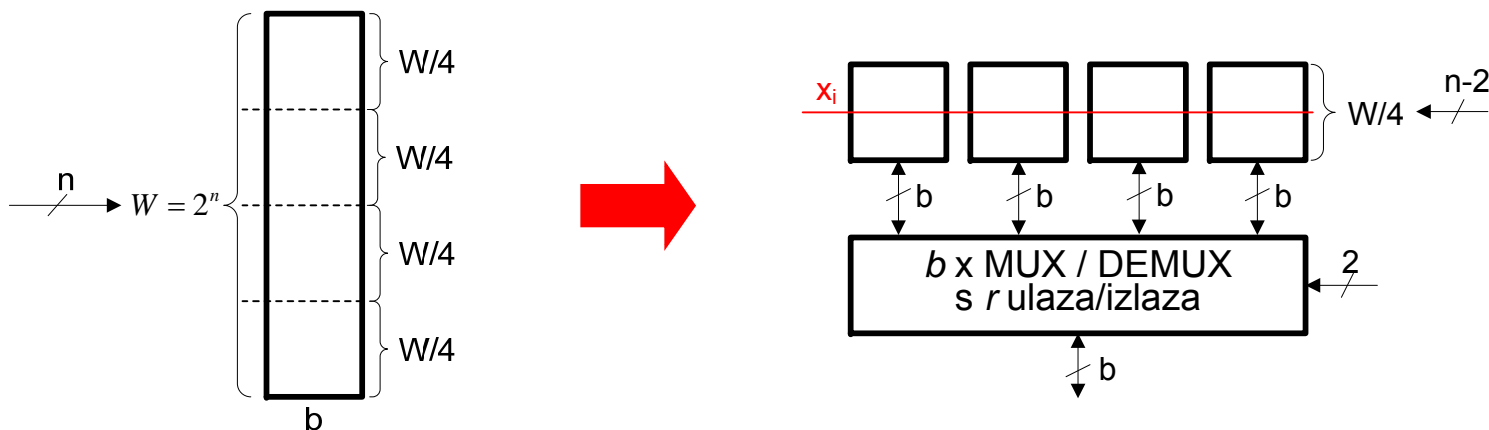


Organizacija memorijskih modula

- 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

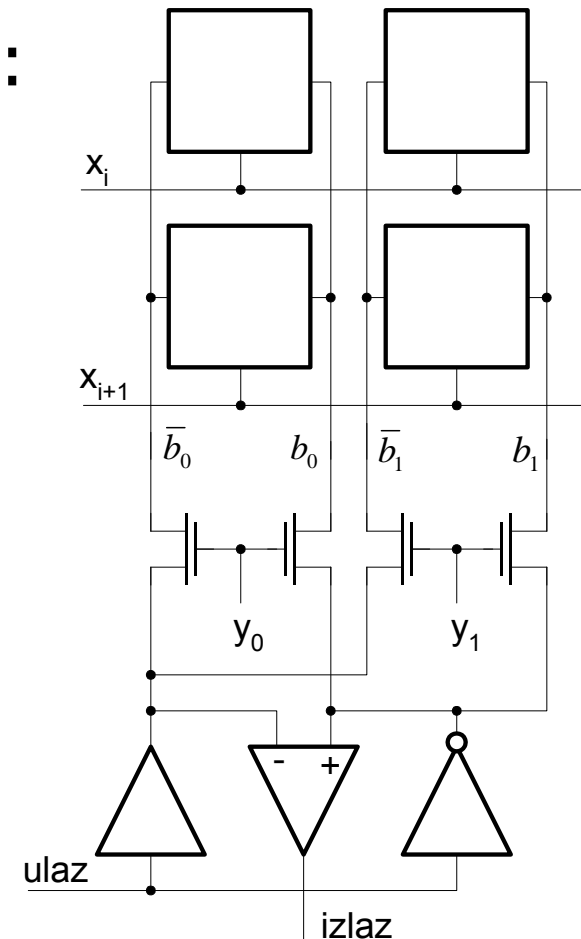
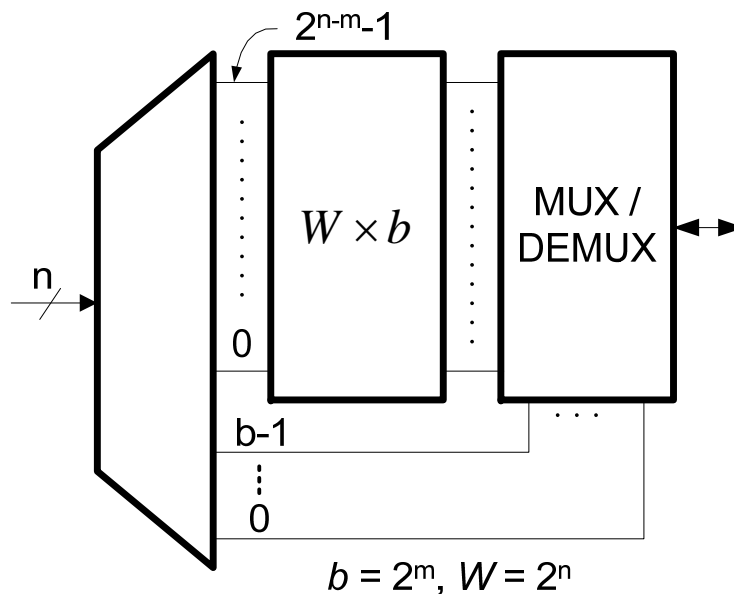
Organizacija memorijskih modula

- 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



Organizacija memorijskih modula

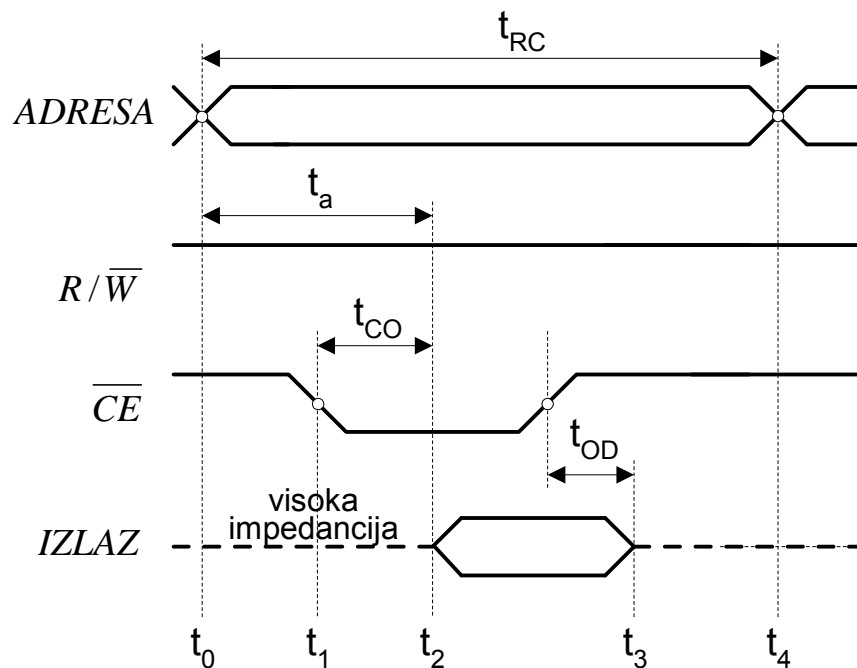
- 2 1/2 D organizacija
~ *posebno jednostavna* izvedba
pristupnog MUX/DEMUX za $b = 1$:
1-bitni memorijski moduli



Vremenski odnosi statičkih memorija

- ciklus *čitanja*:

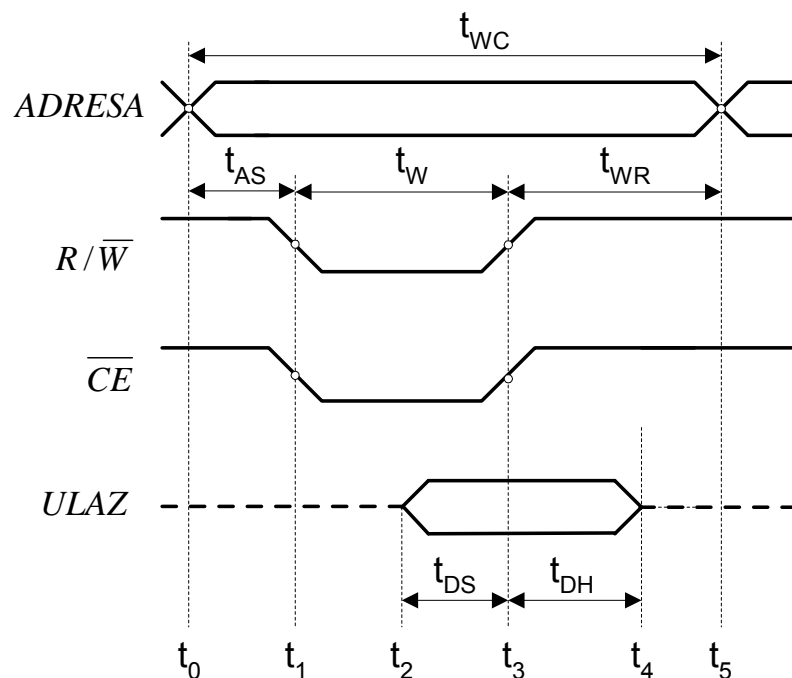
parametar	značenje
t_{RC}	trajanje ciklusa čitanja (između dvije promjene adrese)
t_a	vrijeme pristupa (pojava podataka na izlazu uz $\overline{CE} = 0$)
t_{CO}	kašnjenje u odnosu na \overline{CE} (pojava podataka): podaci su na izlazu nakon $\max(t_a, t_1 + t_{CO})$
t_{OD}	kašnjenje u odnosu na $\overline{CE} = 1$ (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 <u>adrese</u> : 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 $W' = 2^n > W = 2^m, b' > b$

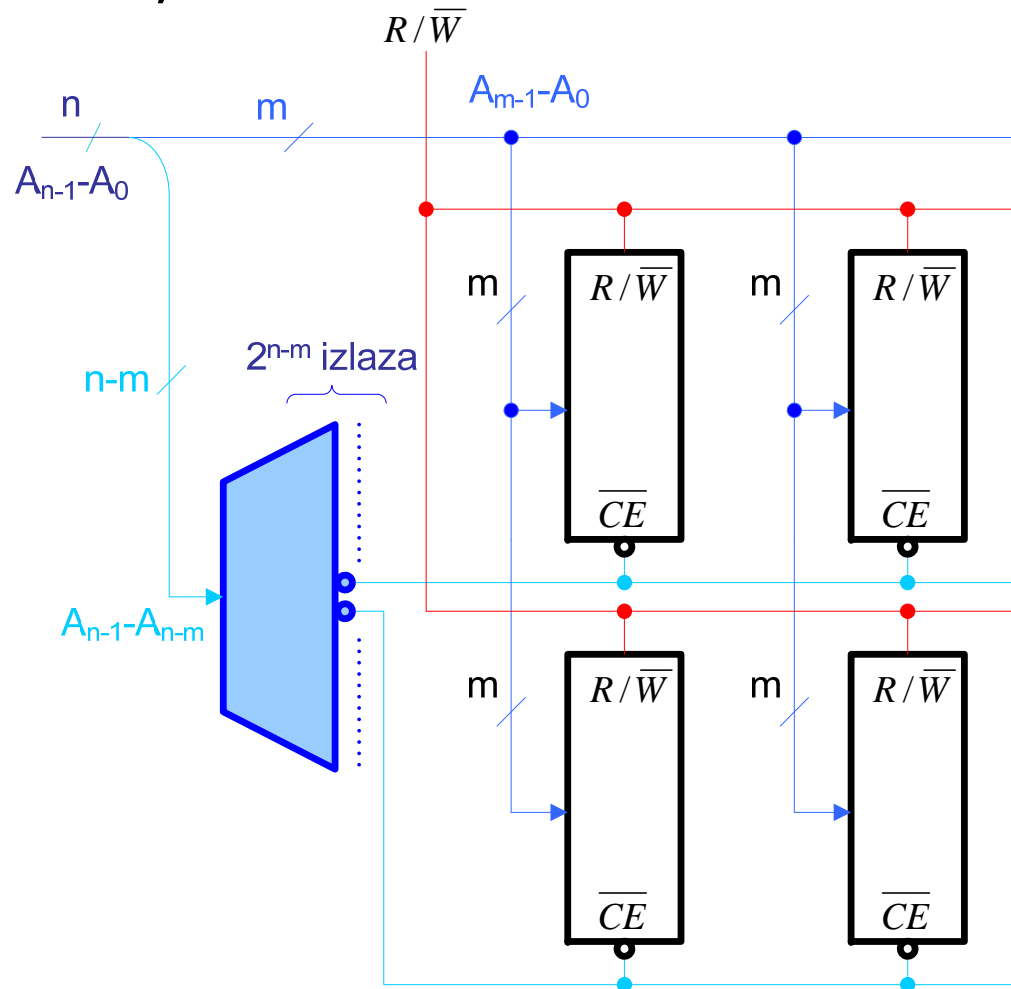


Projektiranje statičkih memorija

- struktura memorije:
 - *dekoder za viših* $(n-m)$ bitova adrese
~ putem signala \overline{CE}
 - 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
 - signal R/\overline{W} također paralelno *na sve* module

Projektiranje statičkih memorija

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



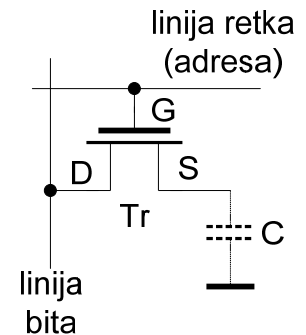


Sadržaj predavanja

- osnovne karakteristike memorija
- statičke memorije
- **dinamičke memorije**

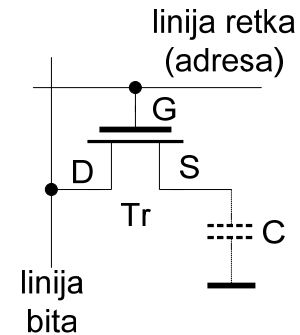
Dinamičke memorije

- 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_{par} se izbija zbog površinske vodljivosti



Dinamičke memorije

- rješenje problema gubitka naboja dinamičke memorijske ćelije:



- *periodičko* čitanje sadržaja memorijske ćelije i ponovno upisivanje (~ 2 ms):
 \sim *osvježavanje* (engl. refreshing)
- memorije s *dinamičkim* memorijskim ćelijama:
 \sim *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 Gbit/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 α -zraka
(kozmičke zrake, Sr u materijalu kućišta IC)
 - rješenje problema:
 - posebne izvedbe C
 - dodatni sklopovi za zaštitu podataka (ECC)
~ tipično Hammingovi kodovi

Dinamičke memorije

- organizacija DRAM
 - ~jedan dekodler: 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} :

$A_{13} \dots A_0 \sim$

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

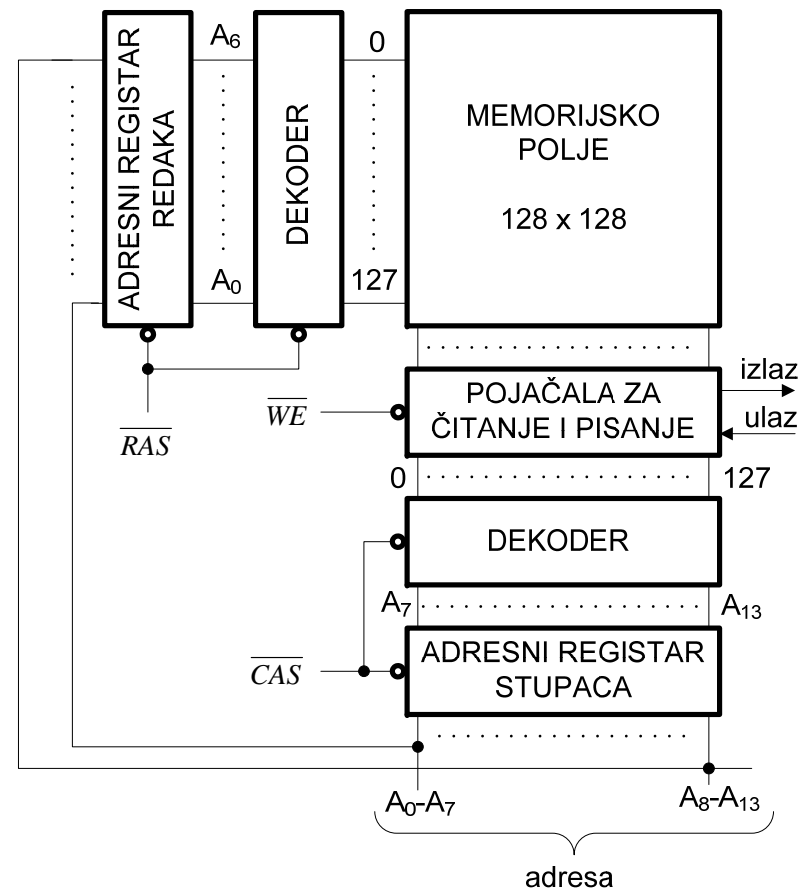
2 UI podatka

1 čitaj/piši

2 impulsa upisa adrese

4/2 napajanja

organizacija: 2 1/2 D



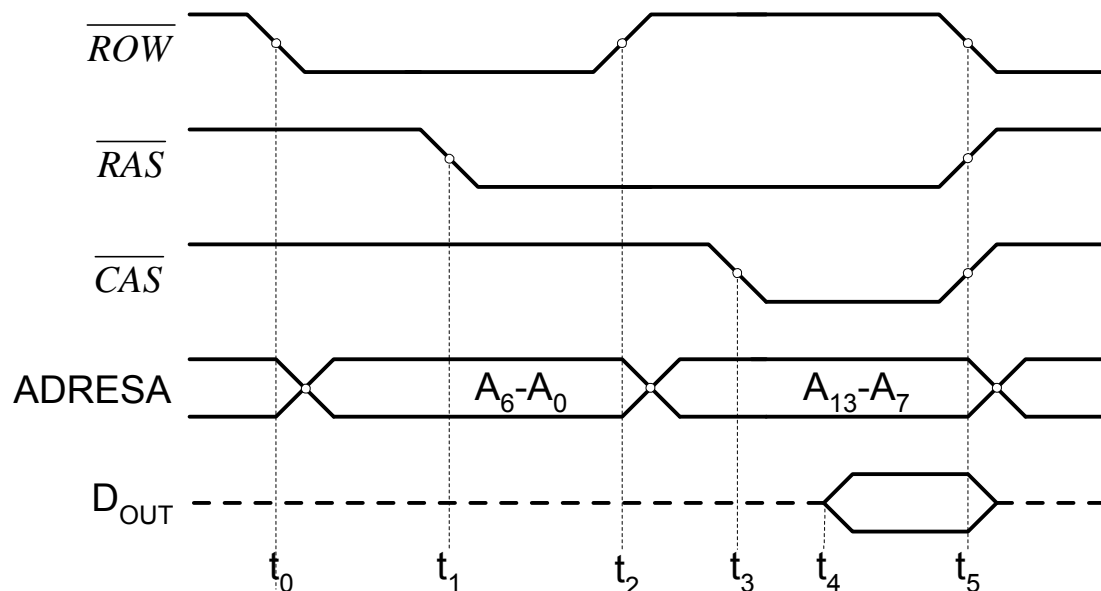
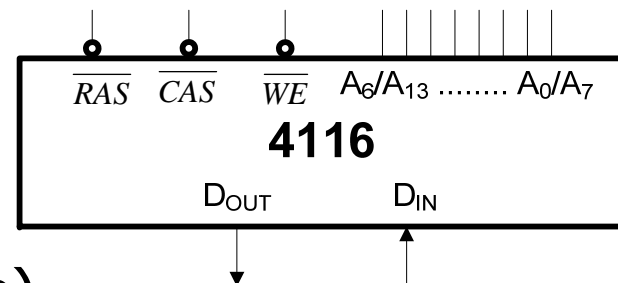
Dinamičke memorije

- vremenski dijagram čitanja/pisanja za 4116/2118:

\overline{ROW} ~ signal odabira $\overline{RAS}/\overline{CAS}$

\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
 $\rightarrow 2 \text{ ms}/128 = 15,63 \mu\text{s} \rightarrow \forall 15 \mu\text{s}$



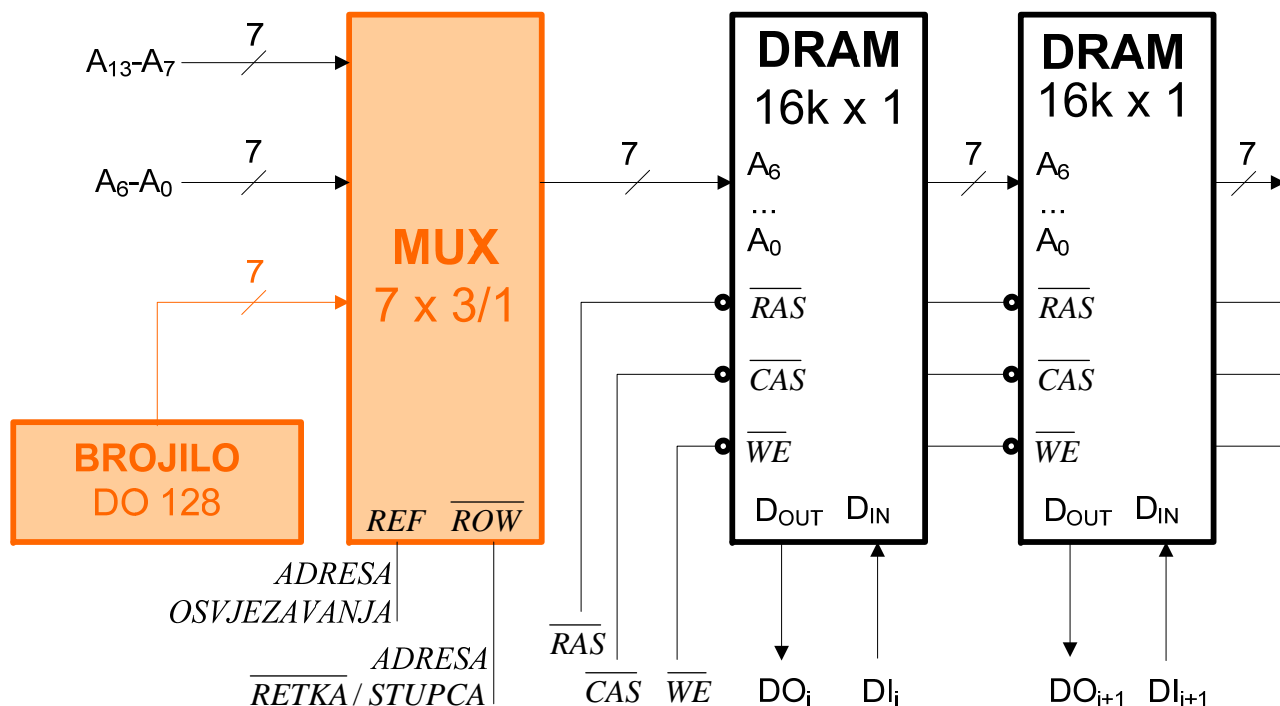
Dinamičke memorije

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

Dinamičke memorije

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

- ubaciti *sklop za osvježavanje*



U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 12: Spremnici podataka.

- osnovne karakteristike memorija: str. 459-464
- statičke memorije: str. 465-477
- dinamičke memorije: str. 477-482



Zadaci za vježbu (1)

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 12: Spremnici podataka.

- osnovne karakteristike memorija: 12.1, 12.2
- statičke memorije: 12.3-12.6, 12.9
- dinamičke memorije: 12.7



Zadaci za vježbu (2)

M. Čupić: *Digitalna elektronika i digitalna logika.*
Zbirka riješenih zadataka, Cjelina 12: Memorije.

- osnovne karakteristike memorija:
 - riješeni zadaci: 12.1
- statičke memorije:
 - riješeni zadaci: 12.6, 12.8
 - zadaci za vježbu: 1
- dinamičke memorije:
 - riješeni zadaci: 12.3, 12.4
 - zadaci za vježbu: 4



12. Sučelje s analognom okolinom (1)



Sadržaj predavanja

- **uključivanje digitalnog sustava u okolinu**
 - **konceptualizacija sučelja**
 - **parametri pretvorbe**
- digitalno-analogna pretvorba

Uključivanje digitalnog sustava u okolinu

- pretvorba na *sučelju* digitalnog sustava i stvarnog svijeta koji ga okružuje:



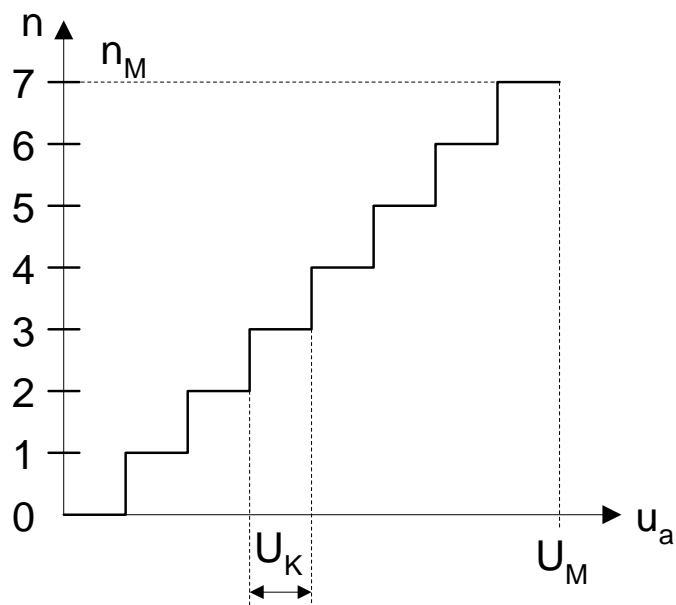
- na *ulazu* digitalnog sustava
~ (analogni) napon \Rightarrow broj:
uzorkovanje (engl. sampling) + kvantizacija
 \rightarrow analogno-digitalna pretvorba (ADC)
- na *izlazu* digitalnog sustava
~ broj \Rightarrow (analogni) napon
 \rightarrow digitalna-analogno pretvorba (DAC)

Uključivanje digitalnog sustava u okolinu

- sučelje digitalnog sustava s analognom okolinom:
 - *pretvornici, konvertori, pretvarači*
~ *sklopovi* na sučelju analognog i digitalnog, i obratno
 - na *ulazu* digitalnog sustava
~ analogno-digitalni pretvornici
(engl. analog-digital convertors, ADC)
 - na *izlazu* digitalnog sustava
~ digitalno-analogni pretvornici
(engl. digital-analog convertors, DAC)
 - "tehnologija" pretvorbe:
 - dinamički pretvornici
~ pretvorba se odvija *u vremenu!*
 - statički pretvornici

Uključivanje digitalnog sustava u okolinu

- karakteristika pretvorbe za ADC (DAC ima isti oblik):
 - U_K : kvant, naponski interval, korak, *kanal*
 - *broj koraka* (kanala)
 \sim *rezolucija* ADC





Uključivanje digitalnog sustava u okolinu

- parametri pretvorbe:
 - rezolucija, razlučivanje
 - točnost
 - pogreška kvantizacije
 - vrijeme pretvorbe

Uključivanje digitalnog sustava u okolinu

- *rezolucija, razlučivanje:*

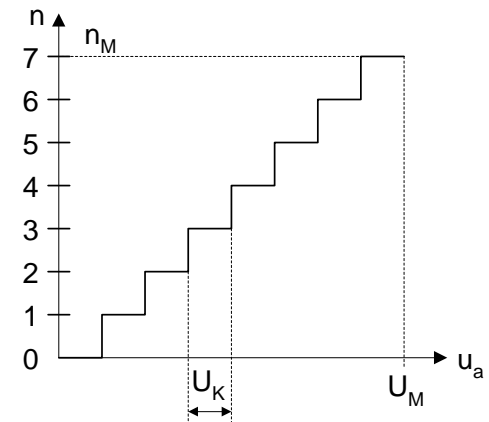
- *širina kanala* kao
% max ulaznog napona (ADC),
odnosno max izlaznog napona (DAC):

$$\text{rezolucija} = U_K / U_M$$

- izražavanje *brojem bitova* izlaznog (ADC),
odnosno ulaznog (DAC) podataka,
jer je $n_M \sim U_M$ (uz k-bitni prikaz):

$$\text{rezolucija} = 1/n_M = 1/(2^k - 1)$$

- tipične vrijednosti: ≈ 14 bitova



Uključivanje digitalnog sustava u okolinu

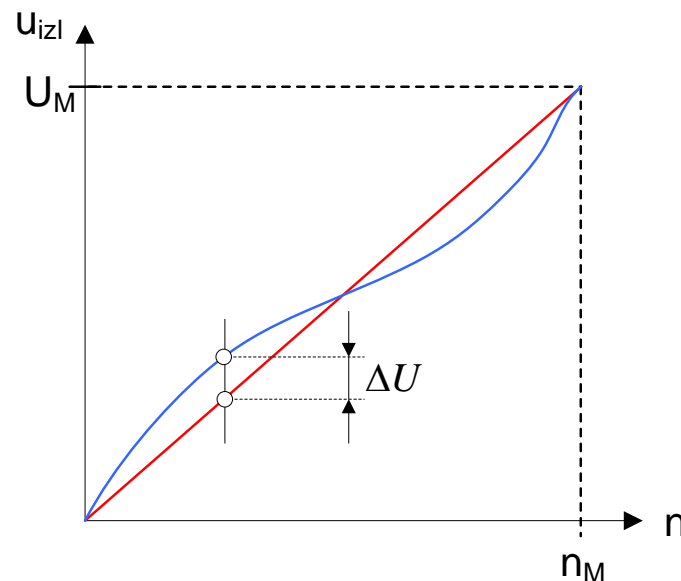
- *točnost*

~ mjera za razliku *stvarnog* analognog izlaza
i izlaza u *idealnom* slučaju

npr. realna karakteristika DAC

- *integralna* nelinearnost
~ granica pogreške:

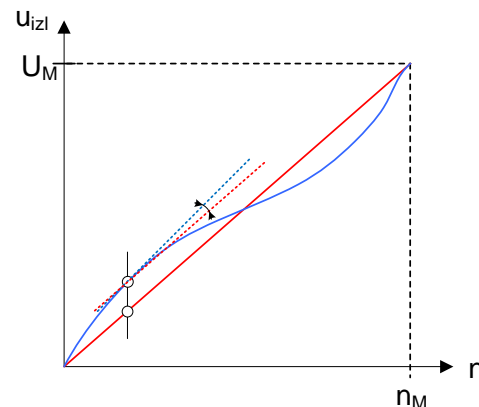
$$\varepsilon_i = \frac{\Delta U}{U_M}$$



Uključivanje digitalnog sustava u okolinu

- *diferencijalna nelinearnost*
~ granica pogreške *u kanalima*

$$\varepsilon_d = \frac{\left(\frac{du_a}{dn}\right)_{\max} - \left(\frac{du_a}{dn}\right)_{idealno}}{\left(\frac{du_a}{dn}\right)_{idealno}} = \frac{(U_K)_{\max} - U_K}{U_K} = \frac{\Delta U_K}{U_K}$$



- tipične vrijednosti za točnost: $\approx 10^{-2} \div 10^{-3}$

Uključivanje digitalnog sustava u okolinu

- *pogreška kvantizacije:*
 - rezultat diskretnog karaktera procesa konverzije
 - prava vrijednost
~ napon *u sredini kanala*
→ pogreška:

$$\varepsilon_K = \pm \frac{U_K}{2}$$

- *vrijeme pretvorbe:*
~ od početka konverzije
do pojave konačne vrijednosti na izlazu
tipične vrijednosti: ADC: $\approx 10 \text{ ns} \div 10 \text{ s}$
DAC: $\approx 100 \text{ ns}$



Sadržaj predavanja

- konceptualizacija sučelja
- **digitalno-analogna pretvorba**
 - **princip DA pretvorbe**
 - **otporna mreža s težinski raspoređenim otporima**
 - **ljestvičasta otporna mreža**
 - **DA pretvornik s brojilom**

Digitalno-analogna pretvorba

- *DA pretvorba:*

- jednostavniji sklopovi (jednostavniji od ADC)
- (u nekim rješenjima) dijelovi AD pretvornika
- cilj pretvorbe
~ broj u binarnom prikazu konvertirati
u analognu veličinu, obično napon:

$$N = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_12^1 + a_02^0$$

$$U_N = K \cdot U_{REF} \cdot N$$

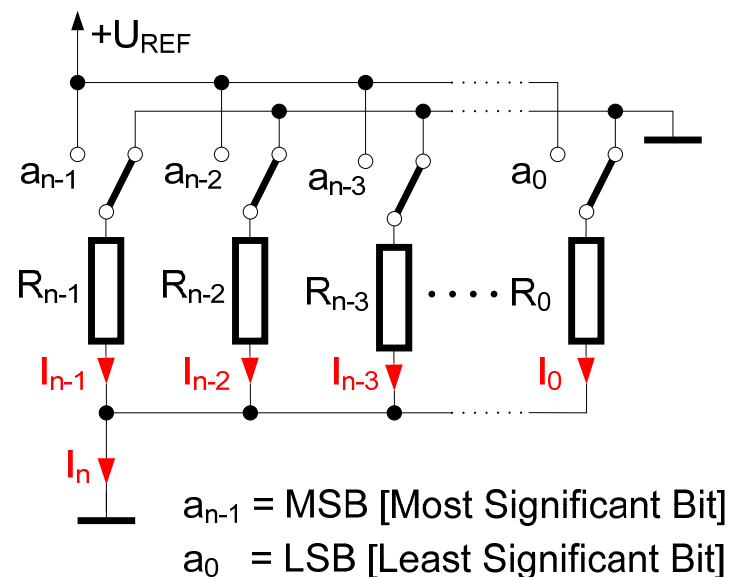
$U'_{REF} = K \cdot U_{REF}$ [V]: faktor proporcionalnosti

Digitalno-analogna pretvorba

- princip DA pretvorbe:

$$U_N = U_{REF} \cdot (a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \dots + a_1 2^1 + a_0 2^0)$$

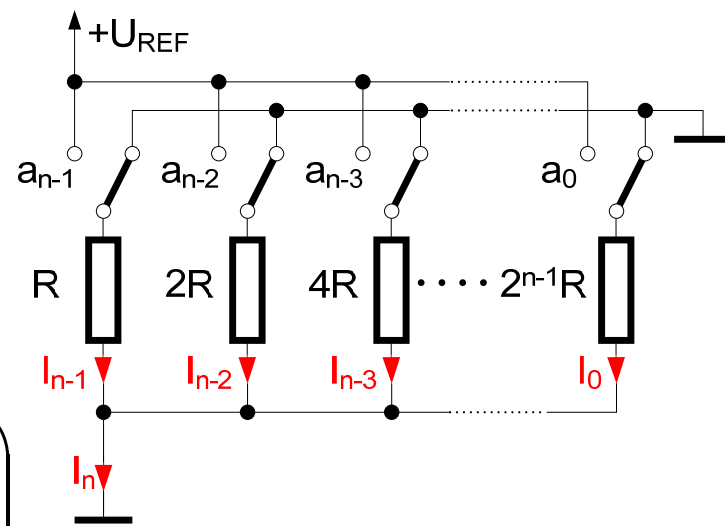
- pojedine težine binarnog broja izraziti *strujama*
- koeficijente uz odgovarajuće težine izvesti "sklopkama"
- principijelno rješenje
~ *otporna mreža*:
statički DA pretvornik



Digitalno-analogna pretvorba

- *otporna mreža s težinski raspoređenim otporima:*

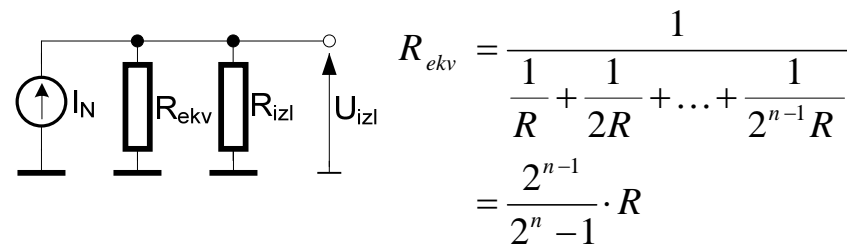
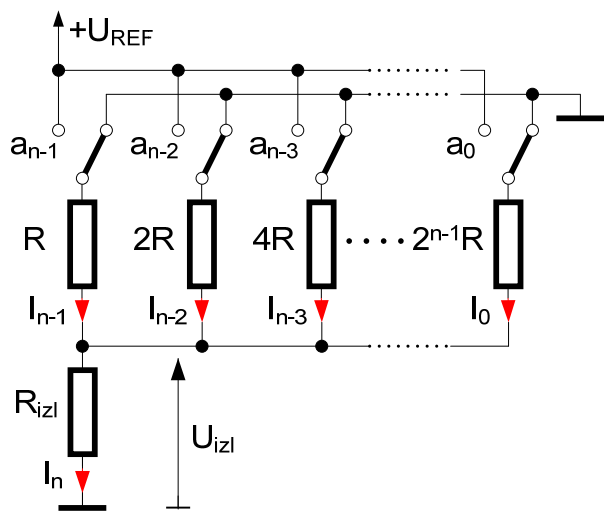
- R_i se međusobno odnose kao težine brojnih mjesta:
→ *zbrojiti struje* kroz R_i
- analogna veličina je *struja*



$$\begin{aligned} I_N &= U_{REF} \cdot \left(\frac{a_{n-1}}{R_{n-1}} + \frac{a_{n-2}}{R_{n-2}} + \dots + \frac{a_0}{R_0} \right) \\ &= \frac{U_{REF}}{2^{n-1} \cdot R} \cdot (a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \dots + a_0 \cdot 2^0) \\ &= \frac{U_{REF}}{2^{n-1} \cdot R} \cdot N \end{aligned}$$

Digitalno-analogna pretvorba

- otporna mreža s težinski raspoređenim otporima
 \sim obično se za izlaznu veličinu želi *napon*:
 \rightarrow *struja* na poznatom R



$$U_{izl} = I_N \cdot (R_{ekv} \parallel R_{izl})$$

$$= \frac{U_{REF}}{2^{n-1} \cdot R} \cdot \frac{R_{ekv} \cdot R_{izl}}{R_{ekv} + R_{izl}} \cdot (a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \dots + a_0 2^0)$$

$$= \dots =$$

$$= \frac{R_{izl} \cdot U_{REF}}{2^{n-1} \cdot R + (2^n - 1) \cdot R_{izl}} \cdot \sum_{i=0}^{n-1} a_i \cdot 2^i$$

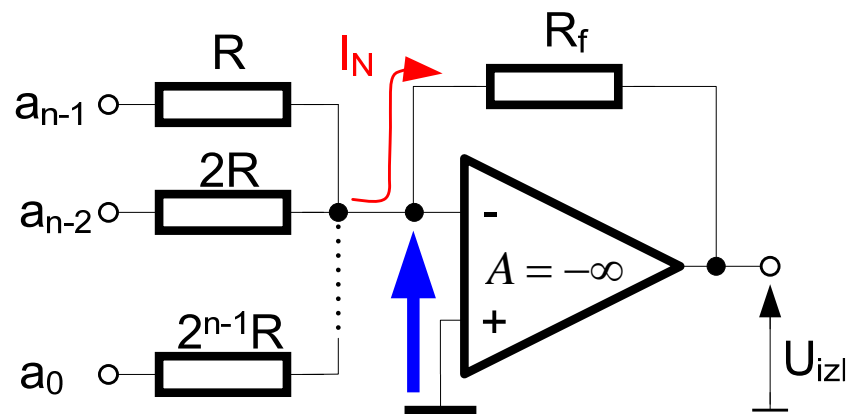
$$= K \cdot U_{REF} \cdot N$$

$$K = f(R_{izl}) \neq const.$$

Digitalno-analogna pretvorba

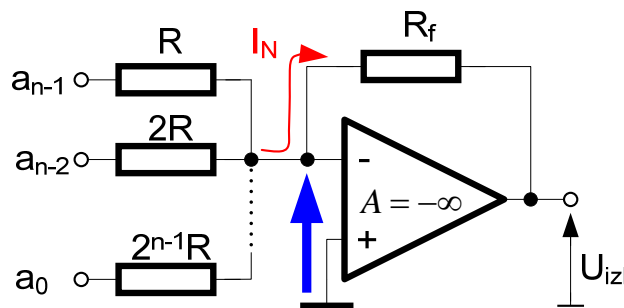
- izbjegavanje utjecaja R_{izl} u otpornoj mreži:
 - *operacijsko pojačalo*
~ "nulti otpor" između ulaza:
virtualna nula na ulazu pojačala
 - zbrajanje I_i na izlazu sklopa:

$$\begin{aligned} U_{izl} &= -I_N \cdot R_f \\ &= -\frac{U_{REF} \cdot R_f}{2^{n-1} \cdot R} \cdot \sum_{i=0}^{n-1} a_i \cdot 2^i \end{aligned}$$



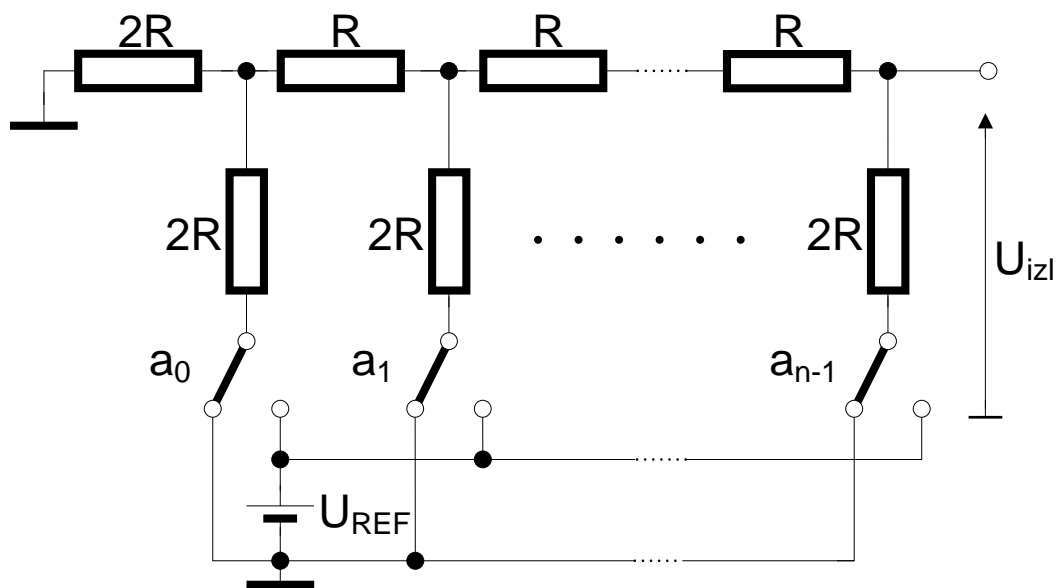
Digitalno-analogna pretvorba

- tehnološki problem izvedbe većih mreža:
 - preveliki odnos R_{n-1} (uz MSB) i R_0 (uz LSB)
~ uz zadanu pogrešku zbog tolerancija otpora
 R_{n-1} treba biti *vrlo precizan*
(također i svi R_i , za $i \nearrow$)
 - koristiti drugo rješenje, već od $n = 4$



Digitalno-analogna pretvorba

- *ljestvičasta otporna mreža* (engl. ladder network):
 - koristi *samo dvije* vrijednosti za R (R i $2 \cdot R$)
 - za isti n potrebno *dva puta više* R_i

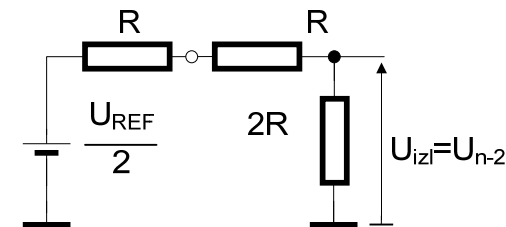
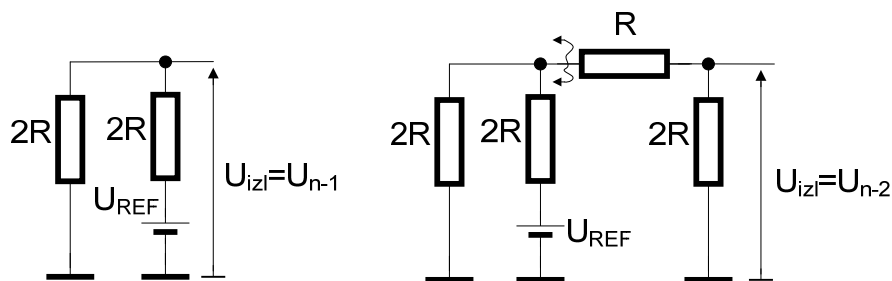
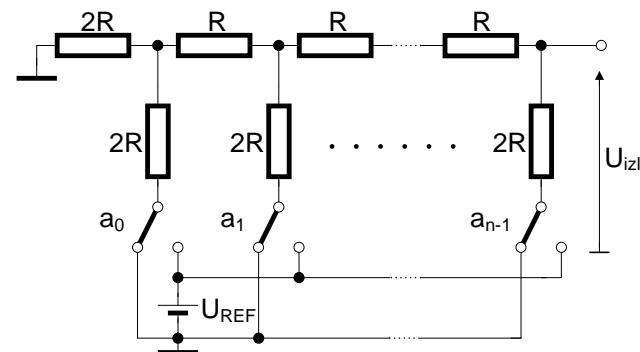


Digitalno-analogna pretvorba

- određivanje izlaznog napona ljestvičaste otporne mreže

$$U_{izl} = f(N):$$

- zaključenje $2 \cdot R$ "lijevo" od promatranog čvora a_i
- koristiti *Théveninov teorem*



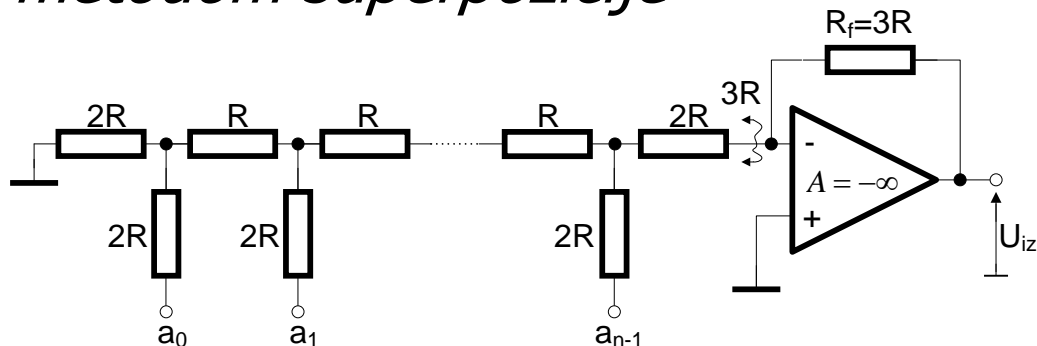
$$U_{n-1} = \frac{U_{REF}}{2}$$

$$U_{n-2} = \frac{U_{REF}}{4}$$

$$U_{izl} = \sum_{i=0}^{n-1} U_i = \frac{U_{REF}}{2^n} \cdot \sum_{i=0}^{n-1} a_i \cdot 2^i = \frac{U_{REF}}{2^n} \cdot N$$

Digitalno-analogna pretvorba

- izvedba ljestvičaste otporne mreže s *operacijskim pojačalom*:
 - zaključenje $2 \cdot R$ s obje strane čvora a_i
 - određivanja doprinosa pojedinih bitova *metodom superpozicije*



$$U_{n-1} = \frac{1}{3} \cdot \frac{U_{REF}}{2^{n-1}} \sum_{i=0}^{n-1} a_i \cdot 2^i = \frac{1}{3} \cdot \frac{U_{REF}}{2^{n-1}} \cdot N$$

$$U_{izl} = -U_{n-1} \cdot \frac{R_f}{2R} = -U_{n-1} \cdot \frac{3R}{2R} = -\frac{1}{3} \cdot \frac{3R}{2R} \cdot \frac{U_{REF}}{2^{n-1}} \cdot N = -\frac{U_{REF}}{2^n} \cdot N$$



Digitalno-analogna pretvorba

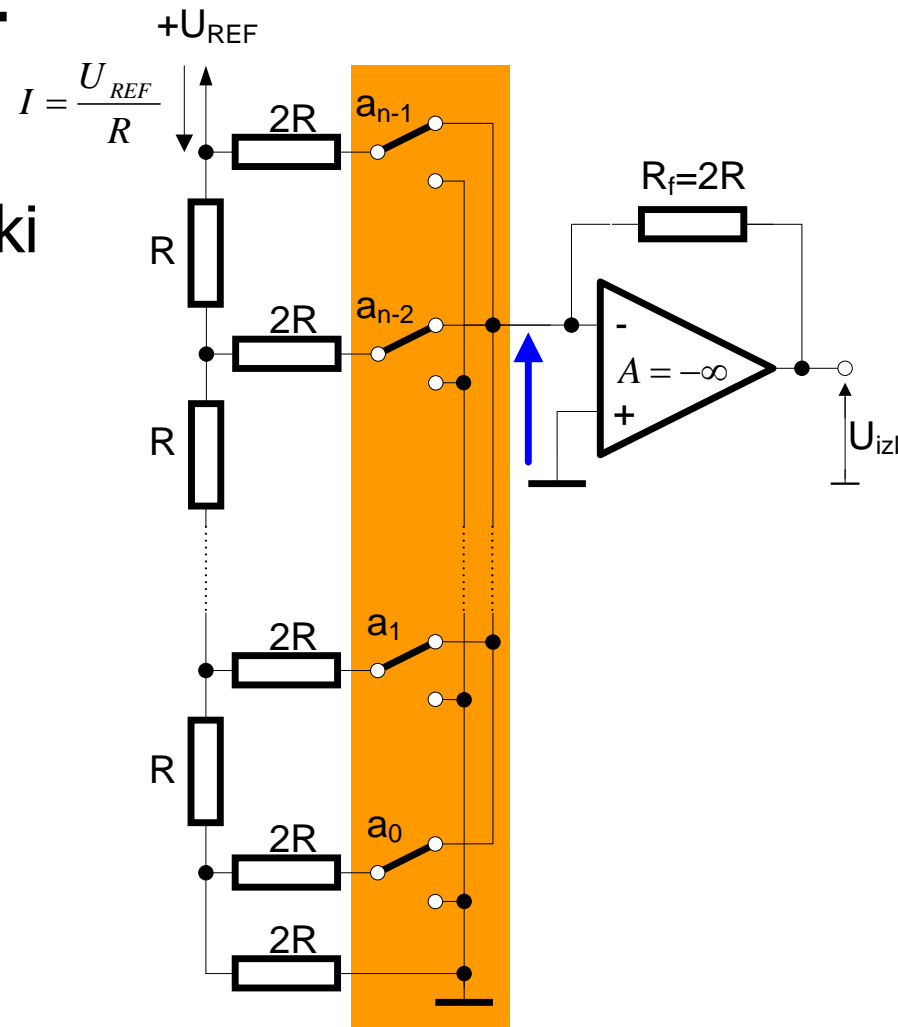
- svojstva izvedbi ljestvičastih otpornih mreža:
 - samo *dvije vrijednosti* otpora
~ moguće zadovoljiti tolerancije,
pogotovo stoga što se traži *omjer vrijednosti*
 - pogodno za *integriranu izvedbu* i za veći broj bitova
~ svi otpori i naponski izvori *jednako* opterećeni
 - *sporije* u pogledu vremenskog odziva
~ niz četveropola s C_{par}

Digitalno-analogna pretvorba

- uklanjanje problema kašnjenja ljestvičaste otporne mreže:

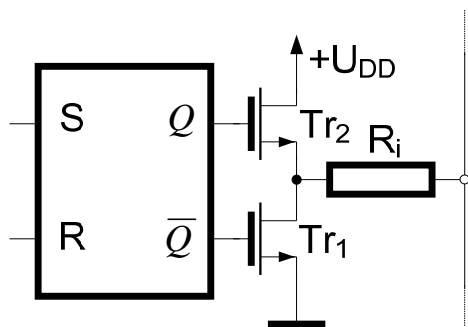
- virtualna nula:
~ nema nabijanja C_{par} kod prebacivanja sklopki
- nema promjena u opterećenju U_{REF} i R_i

$$U_{\text{izl}} = -\frac{U_{\text{REF}}}{2^{n-1}} \sum_{i=0}^{n-1} a_i \cdot 2^i$$



Digitalno-analogna pretvorba

- *sklopke za DA pretvornike:*
 - konceptualni spoj:



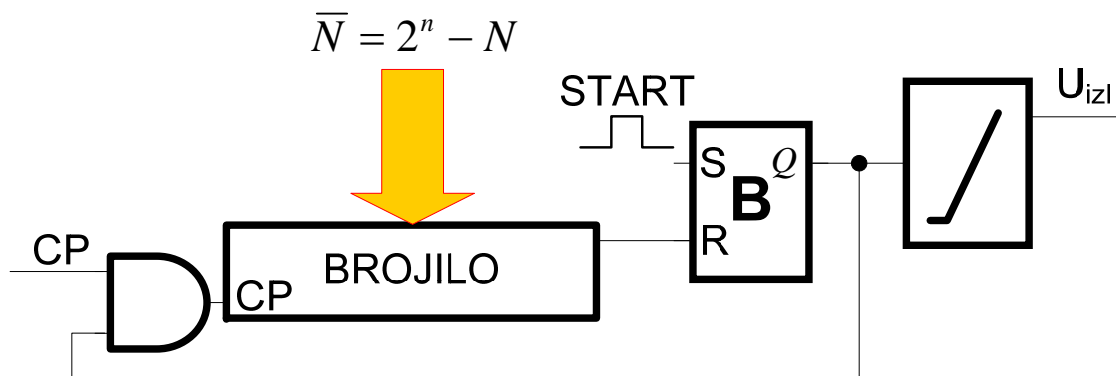
- osigurati $R_{\text{ekv}} \ll R_i$

Digitalno-analogna pretvorba

- svojstva *integriranih* otpornih mreža
~ otporne mreže integrirane u DA pretvornike:
 - ljestvičaste
 - za veći broj bitova (8÷14)
 - tipično $R = 10\text{ k}\Omega / 20\text{ k}\Omega$
 - tolerancije R : $\Delta R = \pm 5\%$
 - garantirana pogreška $U_{\text{izl}} \leq 1/4$ doprinosa LSB
u granicama $-55^\circ\text{ C} \leq T \leq 125^\circ\text{ C}$

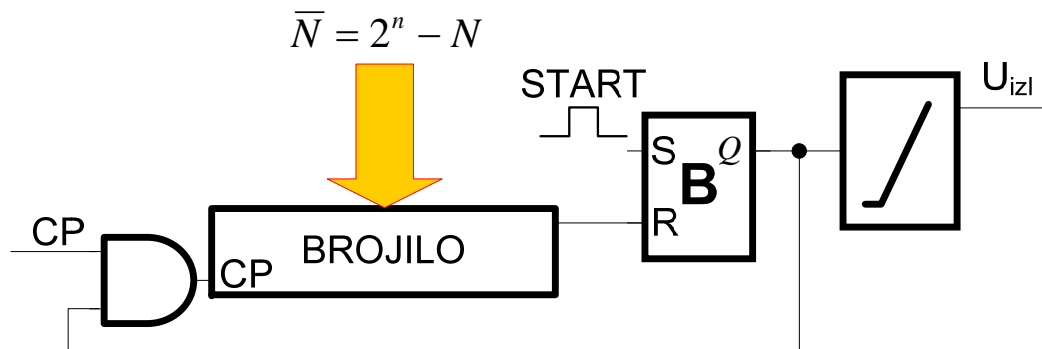
Digitalno-analogna pretvorba

- *DA pretvornik s brojilom*
~ ugrađeno brojilo s prethodnim postavljanjem:
 - u brojilo upisati *dvojni komplement* od N
 - $B = 1 \Rightarrow$ *pretvorba*: brojilo broji, generira se pilasti napon U_{izl}
 - $B = 0 \Rightarrow U_{izl} \sim N$ (n-bitni prikaz)



Digitalno-analogna pretvorba

- DA pretvornik s brojilom:



- *dinamički* DA pretvornik:
 - pretvorba *traje* izvjesno vrijeme
 - pretvorba *indirektna*
~ preko vremena
(brojilo broji $\Rightarrow B = 1 \Rightarrow$ porast U_{izl})!
- problemi pri pretvorbi
~ osigurati $f_{CP} = const.$ + *linearnost* pile

- U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 13:
Sučelje s analognom okolinom.
- uključivanje digitalnog sustava u okolinu: str. 491-494
 - digitalno-analogna pretvorba: str. 494-499



Zadaci za vježbu (1)

- U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 11:
Sekvencijski moduli: registri i brojila.
- digitalno-analogna pretvorba: 13.1-13.8, 13.15-13.19



Zadaci za vježbu (2)

M. Čupić: *Digitalna elektronika i digitalna logika. Zbirka riješenih zadataka*, Cjelina 13: AD i DA konverzija.

- digitalno-analogna pretvorba:
 - riješeni zadaci: 13.1-13.4, 13.8
 - zadaci za vježbu: 1



12. Sučelje s analognom okolinom (2)



Sadržaj predavanja

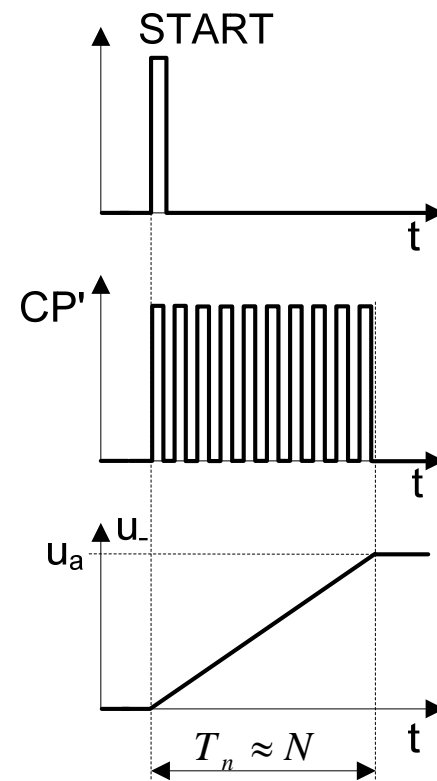
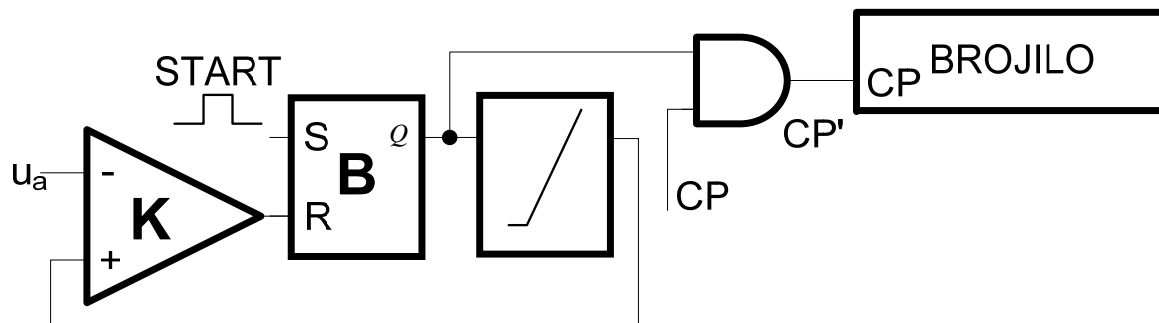
- **analogno-digitalna pretvorba**
 - **Wilkinsonov pretvornik**
 - **AD pretvornik s postepenim približavanjem**
 - **AD pretvornik sa sukcesivnom aproksimacijom**
 - **slijedni AD pretvornik**
 - **paralelni AD pretvornik**
 - **sklop za uzorkovanje**

Analogna-digitalno pretvorba

- *AD pretvorba:*
 - složeniji postupak!
 - tipično uključuje *brojilo*
~ *dinamički* postupci:
 - pretvorba *usporedbom* $U_{ul} = U_a$ i
(interno) generiranog $U_d \sim N$
 - završavanje pretvorbe za $U_d = U_a$
 - karakteristična *petlja povratne veze*
~ *naponski komparator*
 - najzahtjevnije izvedbe (stoga i najskuplje!)
~ nema brojila!

Analogna-digitalno pretvorba

- *Wilkinsonov pretvornik*
~ dinamički pretvornik:
ugrađeno brojilo
 - dok brojilo broji generira se pilasti napon
 - komparator u *petlji povratne veze*
~ usporedba pilastog napona
s $U_{ul} = U_a$



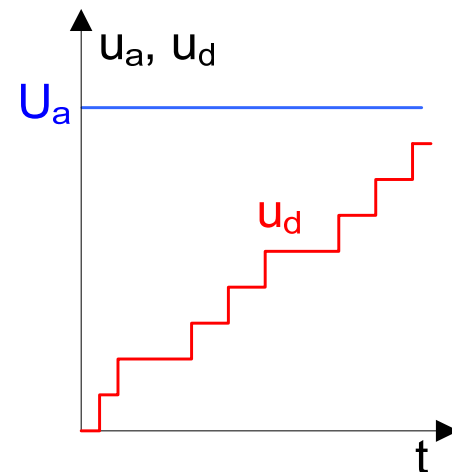
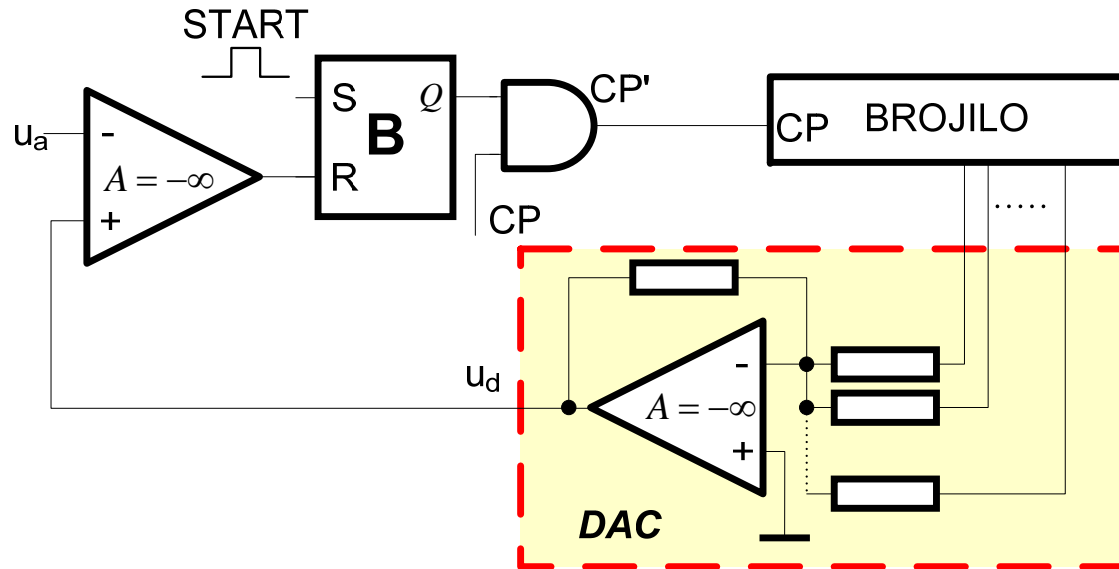


Analogna-digitalno pretvorba

- problemi Wilkinsonovog pretvornika
~ pretvorba *indirektna* – obavlja se preko vremena!
 - generirana pila obično *nelinearna*
~ pogreške pri pretvorbi ($U_d \neq U_a$)
 - potrebno osigurati $f_{CP} = \text{const.}$
 - ostvariti sinkronizaciju impulsa START i CP

Analogna-digitalno pretvorba

- *AD pretvornik s postepenim približavanjem* (s DA pretvornikom),
brojeći AD pretvornik (engl. counting ADC)
~ modifikacija Wilkinsonovog pretvornika
koja rješava njegove probleme



Analogna-digitalno pretvorba

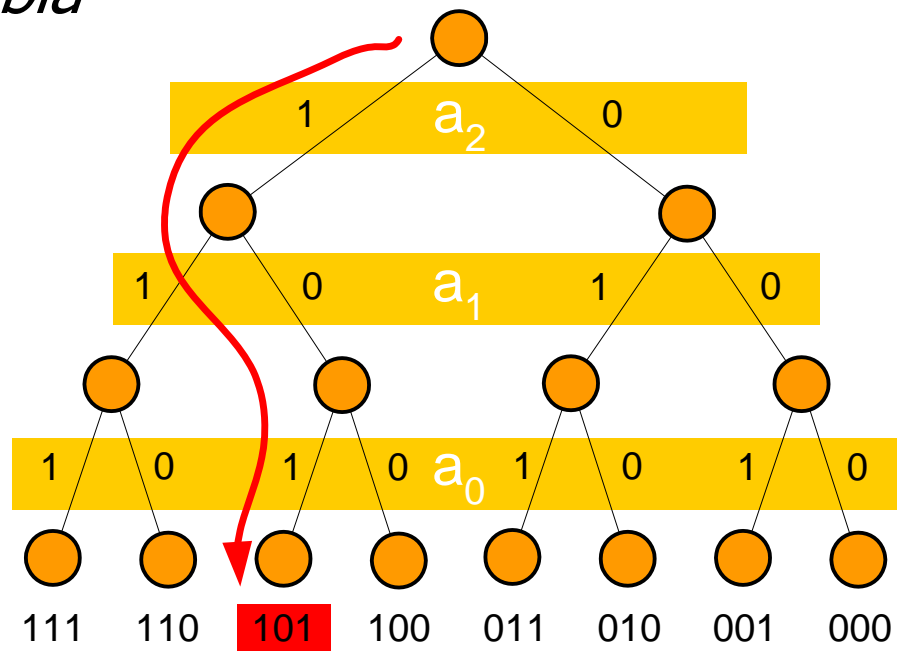
- AD pretvornik s postepenim približavanjem:
 - pretvorba je *direktna, bez* posrednika (generatora pile)
~ DA pretvornik umjesto generatora pile
 - eliminirana potreba za točnom i stabilnom f_{CP}
 - moguće dozvoliti $f_{CP} \neq \text{const.}$
~ *neperiodički* impulsi pobude brojila
 - točnost pretvorbe
~ točnost DAC:
ovisi o pasivnim komponentama (R_i),
može se dobro namjestiti

Analogna-digitalno pretvorba

- *brzi AD pretvornik,*
AD pretvornik sa sukcesivnom aproksimacijom
(engl. successive approximation ADC):
 - jako popularni AD pretvornik
~ široko korišten samostalno ili
u kombinacijama s paralelnim AD pretvornikom
 - pretvorba *direktna*
~ DA pretvornik u petlji povratne veze
 - cilj:
 - ubrzanje pretvorbe, $T_{\max} \ll N_{\max} \cdot T_{\text{CP}}$
~ *smanjenje* broja koraka
 - konstantno vrijeme pretvorbe, $T \neq f(U_a)$
~ *konstantan* broj koraka

Analogna-digitalno pretvorba

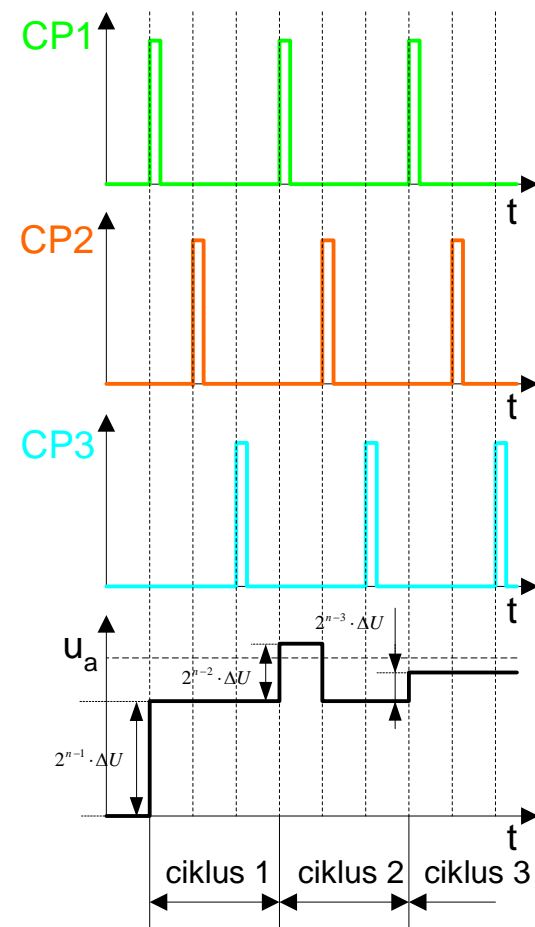
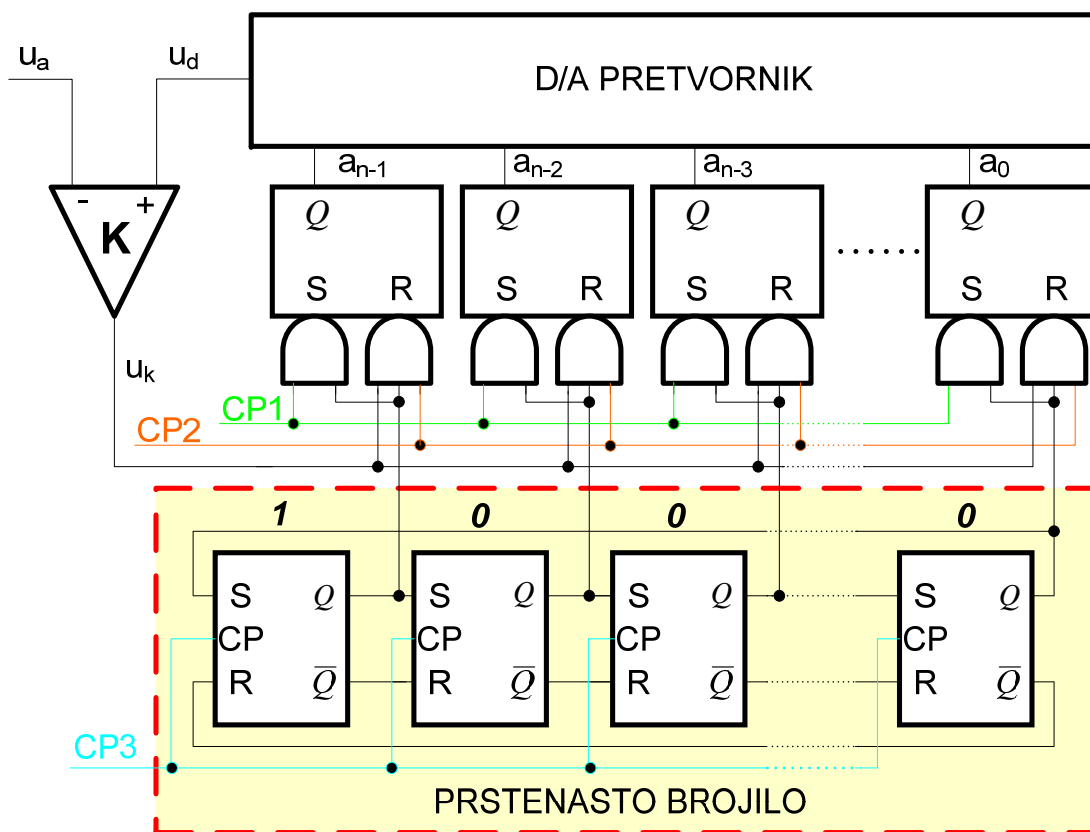
- princip rada brzog AD pretvornika:
 - usporedba u_a i u_d :
~ postepeno formiranje u_d počev od MSB (a_{n-1}):
if $u_a > u_{d,i}$ **then** $u_k = 0$
else $u_k = 1$ {brisanje doprinosa a_i }
 - *obilazak binarnog stabla po dubini*
(engl. preorder tree traversal)



Analogna-digitalno pretvorba

- princip rada brzog AD pretvornika:
 - broj koraka za proizvoljni u_a
 \sim broj bitova zapisa n
 - izvedba sklopom za *slijedno ispitivanje* bitova:
 - prstenasto brojilo (početno 1 kod a_{n-1})
 - binarno brojilo s dekoderom
 - mreža logičkih sklopova

- izvedba brzog AD pretvornika *prstenastim brojilom*



Analogna-digitalno pretvorba

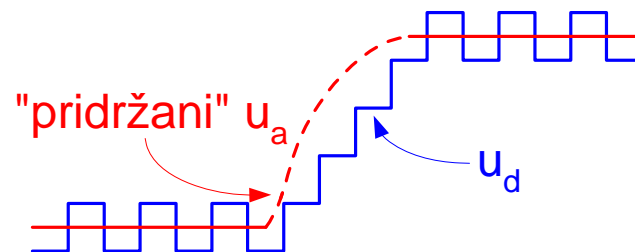
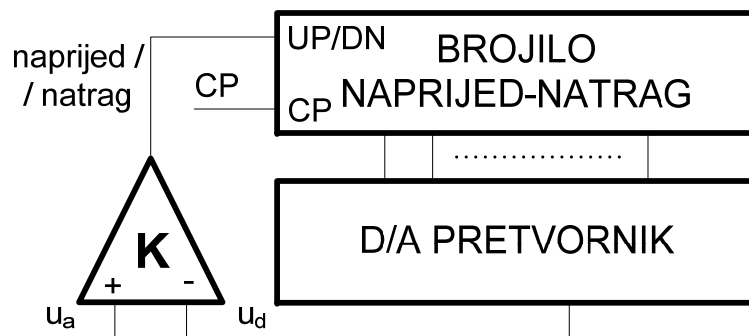
- *slijedni AD pretvornik*
(engl. tracking converter, servo converter),
kontinuirano brojeći AD pretvornik
(engl. continuous-digital-ramp converter):
 - (dodatno) *ubrzanja* pretvorbe
~ modifikacija osnovnog AD pretvornika s brojiлом
(s postepenim približavanjem)
 - zamjena binarnog brojila (broji $0 \rightarrow N \sim U_{a,i+1}$)
brojiлом naprijed-natrag

Analogna-digitalno pretvorba

- slijedni AD pretvornik
~ ubrzanje pretvorbe
korištenjem brojila naprijed-natrag:
 - smjer brojanja: $\nearrow/\searrow = f(\Delta U_a) = f(U_{a,i+1} - U_{a,i})$
~ u_d "slijedi" u_a
 - naročito pogodno
kad se u_a mijenja "relativno sporo"
 - u prosjeku $T_{\text{naprijed-natrag}} \approx 1/2 \cdot T_{\text{binarno}}$
 $\Rightarrow f_{\text{naprijed-natrag}} \approx 2 \cdot f_{\text{binarno, max}}$

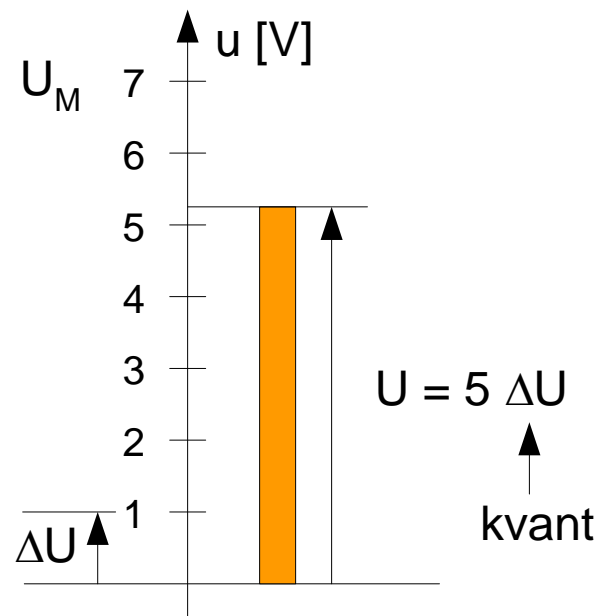
Analogna-digitalno pretvorba

- slijedni AD pretvornik
~ smjer brojanja: ↗/↘ = $f(\Delta U_a) = f(U_{a,i+1} - U_{a,i})$



Analogna-digitalno pretvorba

- *paralelni AD pretvornik, komparatorski AD pretvornik* (engl. parallel-comparator ADC):
 - naročito rješenje za postizanje *najbrže moguće* ADC, također i najskuplje
 - usporediti u_a s *nizom referentnih napona*, \forall naponsku razinu u intervalu $0 \div U_M$, s razmakom ΔU



Analogna-digitalno pretvorba

- paralelni AD pretvornik, komparatorski AD pretvornik
~ usporedba u_a s *nizom referentnih napona*:
 - najprikladnija izvedba "niza referentnih napona":
~ otporni djelitelj
 - *niz* naponskih komparatora
~ $l = (2^n - 1)$ za n -bitnu pretvorbu
 - \forall dodatni bit *dva puta više* komparatora
~ cijena!

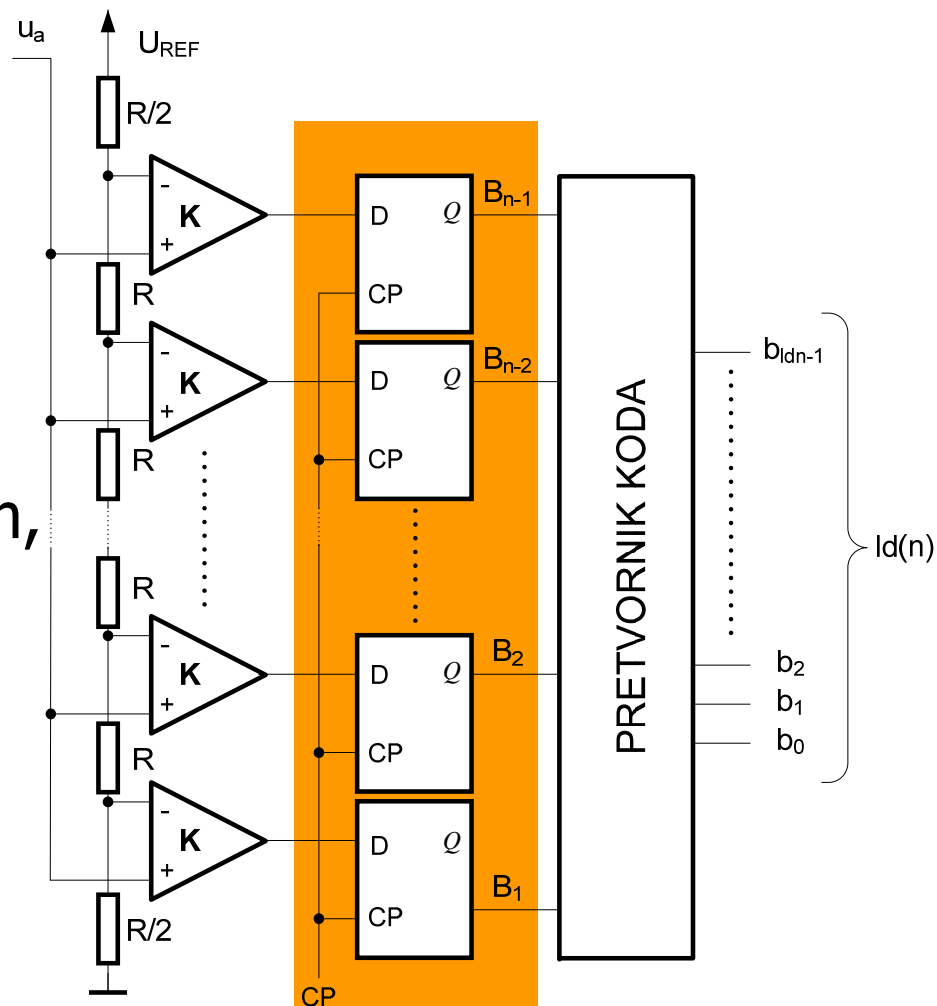
Analogna-digitalno pretvorba

- paralelni AD pretvornik:
 - usporedba u_a s nizom referentnih napona
 \sim *paralelni* rad:
 ne treba brojati!
 - izvorni kod na izlazu komparatora
 \sim "termometarski niz":
 N u *binarnom* obliku tek nakon *pretvorbe koda*
 (Id n = $\log_2 n$: broj bitova ADC)

B_{n-1}	B_{n-2}	...	B_3	B_2	B_1	B_0	$m_{l(l+1)}$...	m_2	m_1	m_0
0	0	...	0	0	0	0	0	...	0	0	0
0	0	...	0	0	0	1	0	...	0	0	1
0	0	...	0	0	1	1	0	...	0	1	0
0	0	...	0	1	1	1	0	...	0	1	1
0	0	...	1	1	1	1	0	...	1	0	0
.....										
1	1	...	1	1	1	1	1	...	1	1	1

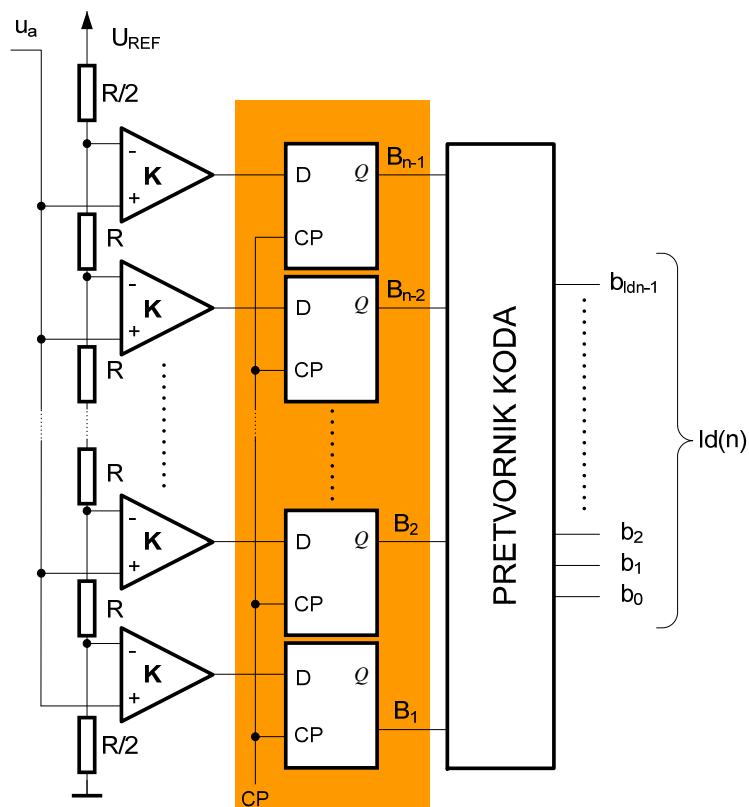
Analogna-digitalno pretvorba

- paralelni AD pretvornik:
 - registar
~ pohranjivanje rezultata pretvorbe; nije nužan
 - bistabili i pretvornik koda
~ najbržom tehnologijom, npr. ECL



Analogna-digitalno pretvorba

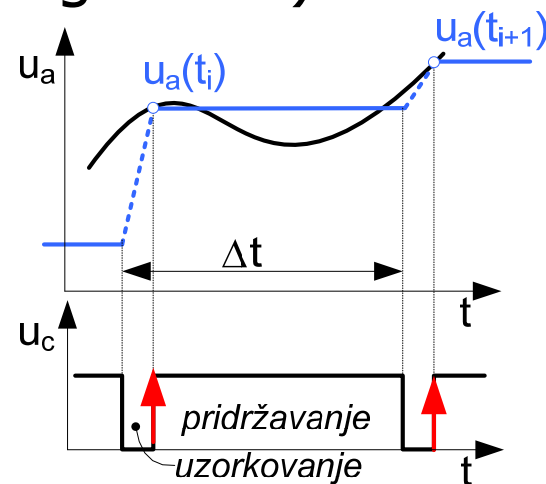
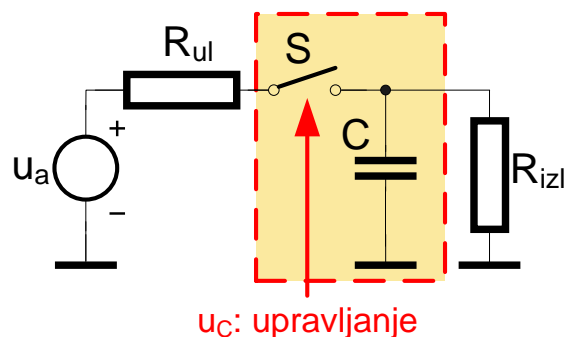
- paralelni AD pretvornik
 \sim ujednačavanje ε_K posebnom raspodjelom referentnih napona: $\varepsilon_K = U_{REF} / (2n - 2)$



$$\begin{array}{lcl}
 u_{n-1} = \frac{2n-3}{2(n-1)} U_{REF} & \left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} 11\dots111 & \text{---} U_{REF} \\
 u_{n-2} = \frac{2n-5}{2(n-1)} U_{REF} & \left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} 01\dots111 & \text{---} \frac{n-2}{n-1} U_{REF} \\
 u_{n-3} = \frac{2n-7}{2(n-1)} U_{REF} & \left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} 00\dots111 & \text{---} \frac{n-3}{n-1} U_{REF} \\
 \vdots & & \vdots \\
 u_3 = \frac{5}{2(n-1)} U_{REF} & \left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} 00\dots011 & \text{---} \frac{2}{n-1} U_{REF} \\
 u_2 = \frac{3}{2(n-1)} U_{REF} & \left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} 00\dots001 & \text{---} \frac{1}{n-1} U_{REF} \\
 u_1 = \frac{1}{2(n-1)} U_{REF} & \left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} 00\dots000 & \text{---} 0
 \end{array}$$

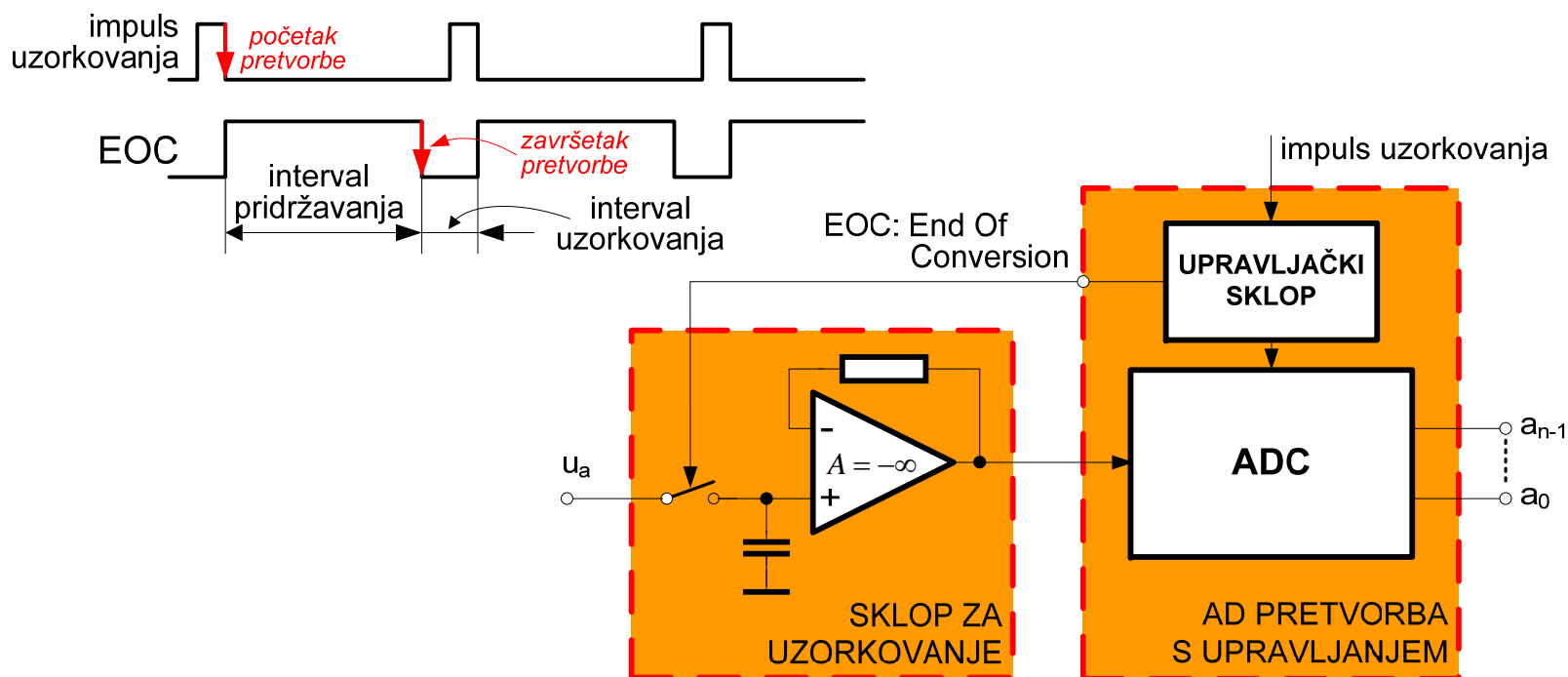
Sklop za uzorkovanje

- *sklop za uzorkovanje*
(engl. sample-and-hold, S/H circuit):
 - uzimanje uzoraka ("uzorkovanje")
~ nabijanje C zatvaranjem prikladno izvedene sklopke
 - otvorena sklopka
~ *pridržavanje* uzorka U_a za vrijeme pretvorbe
 - sklopka
~ "analogna sklopka" (engl. analog switch):
propušta *analogni* napon



Sklop za uzorkovanje

- povezivanje sklopa za uzorkovanje i ADC:
 - uzorkovanje u intervalima (između) pretvorbe
 - pridržavanje uzorka u_a za vrijeme pretvorbe
 - podešavanje opsega u_a



U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 13:
Sučelje s analognom okolinom.

- analogno-digitalna pretvorba: str. 499-509



Zadaci za vježbu (1)

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 11:
Sekvencijski moduli: registri i brojila.

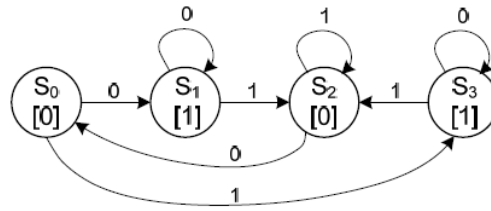
- analogno-digitalna pretvorba: 13.9-13.14



Zadaci za vježbu (2)

M. Čupić: *Digitalna elektronika i digitalna logika. Zbirka riješenih zadataka*, Cjelina 13: AD i DA konverzija.

- analogno-digitalna pretvorba:
 - riješeni zadaci: 13.5-13.7
 - zadaci za vježbu: 2



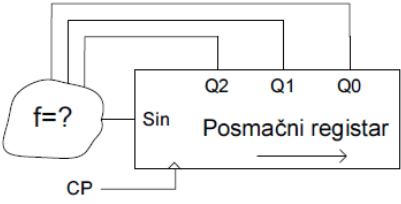
Slika 1: zajednička za zadatke 13-16.

13.	Zadan je Mooreov automat (slika 1). Koja su od stanja ekvivalentna?		
	a) S0 i S3	c) S0 i S2	e) S1 i S3
	b) nema ekvivalentnih stanja	d) S1 i S2	f) ništa od navedenoga

14.	Zadan je Mooreov automat (slika 1). Automat je potrebno ostvariti uporabom dva bistabila T, pri čemu je stanje S_i kodirano kao binarno zapisan broj i . Kako glasi minimalni zapis funkcije ulaza T bistabila B_1 koji čuva bit stanja veće težine? Ulaz automata označen je A.		
	a) $\overline{Q_1} + A$	c) $A\overline{Q_1} + Q_1\overline{Q_0}\overline{A}$	e) $Q_1A + \overline{Q_1}\overline{Q_0}\overline{A}$
	b) Q_0A	d) $\overline{A} + Q_1\overline{Q_0}$	f) ništa od navedenoga

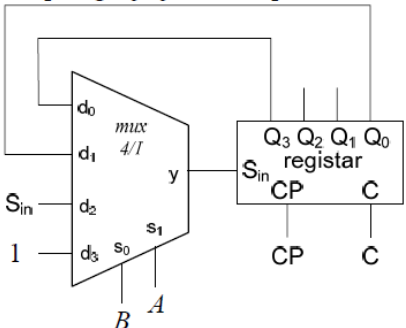
15.	Zadan je Mooreov automat (slika 1). Pretpostavite da se po uključanju automat nalazi u stanju S_0 . Na ulaz se potom dovodi niz 0,0,1,0,1. U što će se generirati na izlazu automata i u kojem će se stanju on nalaziti na kraju?		
	a) 1, 0, 1, 0, 0, 1, stanje S_0	c) 0, 0, 1, 0, 1, 1, stanje S_1	e) 0, 1, 0, 0, 0, 1, stanje S_3
	b) 0, 1, 1, 0, 0, 1, stanje S_3	d) 0, 1, 1, 0, 0, 0, stanje S_2	f) ništa od navedenoga

13. Posmačnim registrom sa slike potrebno je ostvariti brojilo koje broji u ciklusu 0,4,2,5,6,3,1. Sklop mora imati sigurni start. Koji od sljedećih izraza opisuje minimalni oblik funkcije f ? Prilikom očitavanja stanja izlaz Q_2 predstavlja bit najveće težine. Posmak se obavlja u smjeru strelice.



a) $f = Q_1\overline{Q_0} + \overline{Q_1}Q_0 + \overline{Q_2}\overline{Q_1}$
 b) $f = Q_1\overline{Q_0} + \overline{Q_2}Q_1$
 c) $f = \overline{Q_1}\overline{Q_0} + \overline{Q_2}Q_1Q_0$
 d) $f = \overline{Q_2}Q_0 + Q_2\overline{Q_0} + \overline{Q_1}\overline{Q_0}$
 e) $f = \overline{Q_2}\overline{Q_0} + Q_2\overline{Q_1}Q_0$
 f) ništa od navedenoga

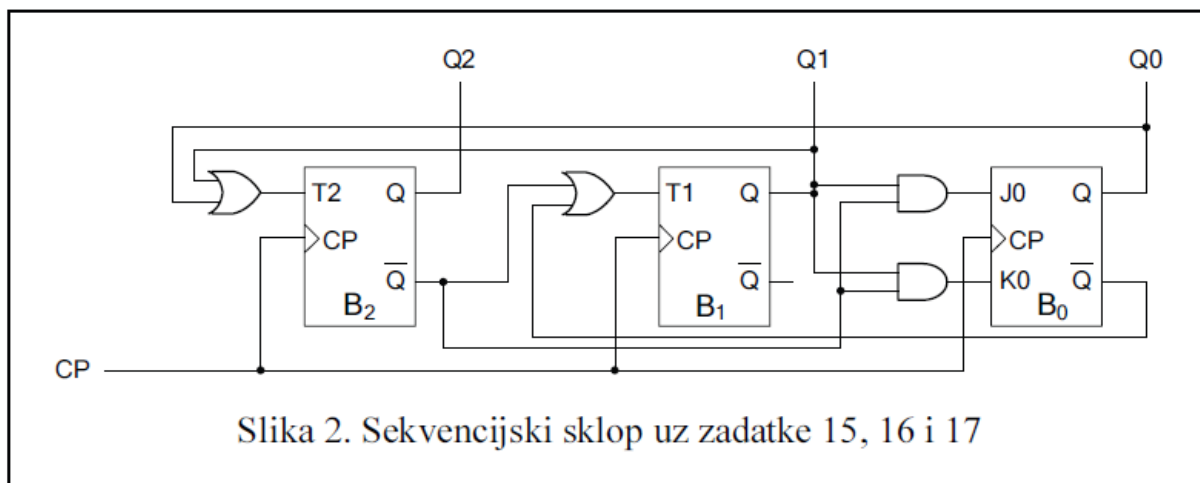
24. Na raspolaganju je 4-bitni posmačni registar i multipleksor 4/1 koji su spojeni prema slici. Ako je $C=0$, registar posmiče podatak u lijevo, a ako je $C=1$, posmiče ga u desno. Na upražnjeno mjesto uvijek se upisuje S_{in} . Ovim sklopom želimo napraviti registar s upravljačkim ulazima X i Y , koji će redefinirati funkciju registra na sljedeći način: za $XY=00$ obavlja se posmak u lijevo (punjenje sa S_{in}), za $XY=01$ posmak u lijevo (punjenje sa 1), za $XY=10$ posmak u desno (punjenje sa S_{in}), a za $XY=11$ aritmetički posmak u desno. Što treba dovesti na ulaz A ?



a) $\overline{X} + \overline{Y}$
 b) $\overline{X}Y$
 c) $X + Y$
 d) $X\overline{Y}$
 e) $X \oplus Y$
 f) ništa od navedenoga

19. Uporabom posmačnog registra sa serijskim ulazom i paralelnim izlazima potrebno je ostvariti brojilo koje broji u ciklusu 0, 4, 2, 5, 6, 7, 3, 1. Označimo izlaze registra $Q_2Q_1Q_0$, pri čemu je Q_2 izlaz najveće težine. Smjer posmaka je takav da se izlaz Q_2 upisuje u Q_1 . Kombinacijski sklop koji će upravljati ulazom S_{in} želimo ostvariti uporabom sklopa PLA tipa Ni-NI minimalnih dimenzija. Kakav nam sklop treba?

a) 3x3x1
 b) 3x3x2
 c) 3x4x1
 d) 3x4x2
 e) 4x3x1
 f) ništa od navedenoga



- | | |
|-----|--|
| 15. | <p>Pogledajte sklop sa slike 2. Poznati sljedeći parametri: $t_{dls}=10\text{ns}$, $t_{hold}=10\text{ns}$, $t_{setup}=10\text{ns}$, $t_{db}=30\text{ns}$. Kolika je maksimalna frekvencija signala takta uz koju će sklop još raditi ispravno?</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>a) 50 MHz</p> <p>b) 40 MHz</p> <p>c) 25 MHz</p> </div> <div style="width: 48%;"> <p>d) 20 MHz</p> <p>e) 10 MHz</p> <p>f) ništa od navedenog</p> </div> </div> |
| 16. | <p>Pogledajte sklop sa slike 2. U kojem ciklusu broji to brojilo?</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>a) 0, 3, 5, 2, 1, 7, 6, 4</p> <p>b) 0, 2, 5, 1, 7, 3, 4, 6</p> <p>c) 0, 5, 1, 7, 3, 4, 2, 6</p> </div> <div style="width: 48%;"> <p>d) 0, 2, 3, 7, 5, 4, 6, 1</p> <p>e) 0, 1, 3, 4, 2, 6, 5, 7</p> <p>f) ništa od navedenog</p> </div> </div> |
| 17. | <p>Pogledajte sklop sa slike 2. Ako bistabil B0 zamijenimo s D bistabilom, što bi tada trebalo dovoditi na njegov ulaz D, kako ne bi promijenili rad sklopa?</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>a) $D = Q_2 \oplus Q_1$</p> <p>b) $D = Q_2 \oplus Q_0$</p> <p>c) $D = Q_2 Q_0 + \overline{Q_1} Q_0 + \overline{Q_2} Q_1 \overline{Q_0}$</p> </div> <div style="width: 48%;"> <p>d) $D = Q_1 Q_0 + \overline{Q_1} \overline{Q_0}$</p> <p>e) $D = \overline{Q_2} Q_1 Q_0$</p> <p>f) ništa od navedenog</p> </div> </div> |
| 11. | <p>Koliko iznosi maksimalna frekvencija rada 5-bitnog sinkronog binarnog brojila sa serijskim prijenosom (prijenos je izveden sklopovima I), ako je poznato $t_{db} = 20\text{ ns}$, $t_{setup} = 15\text{ ns}$, $t_{dls} = 5\text{ ns}$?</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>a) 10 MHz</p> <p>b) 25 MHz</p> <p>c) 50 MHz</p> </div> <div style="width: 48%;"> <p>d) 20 MHz</p> <p>e) 100 MHz</p> <p>f) ništa od navedenog</p> </div> </div> |

4.	Asinkrono binarno brojilo unaprijed ostvareno je uporabom bistabila T. Bistabili imaju dodatne ulaze za postavljanje S_d koji su svi spojeni zajedno, i koriste se za skraćivanje ciklusa brojila. Ako ciklus sadrži 20 stanja a ulazi za postavljanje se aktiviraju logičkom razinom 1, koju funkciju treba ostvarivati sklop koji upravlja ulazima za postavljanje?				
a)	$Q_4\overline{Q}_3Q_2\overline{Q}_1\overline{Q}_0$	c)	$\overline{Q}_4\overline{Q}_3\overline{Q}_2Q_1Q_0$	e)	$Q_4\overline{Q}_3\overline{Q}_2Q_1Q_0$
b)	$Q_4\overline{Q}_3\overline{Q}_2Q_1\overline{Q}_0$	d)	$\overline{Q}_4Q_3Q_2Q_1Q_0$	f)	ništa od navedenog

5.	Bistabilima tipa T koji su okidani padajućim bridom signala takta ostvareno je 4-bitno asinkrono binarno brojilo unaprijed. Parametri bistabila su: $t_{db}=20ns$, $t_{setup}=10ns$, $t_{hold}=10ns$. Neposredno prije trenutka $t=100ns$ brojilo se nalazi u stabilnom stanju 7. U trenutku $t=100ns$ nastupa padajući brid signala takta. Frekvencija signala takta je 10MHz. Što ćemo očitati na izlazima brojila u trenutku $t=145ns$?					
	a) 4	b) 0	c) 12	d) 10	e) 8	f) ništa od navedenog

11.	Asinkrono binarno brojilo u ciklusu s 13 stanja ostvareno je uporabom bistabila T s asinkronim ulazom za brisanje. Ako je poznato $t_{db}=20ns$, $t_{hold}=10ns$, $t_{setup}=20ns$, $t_{dls}=5ns$, $t_{očitanja}=20ns$, izračunajte period signala takta za maksimalnu frekvenciju na kojoj će sklop i dalje raditi ispravno.		
	a) 90 ns	c) 100 ns	e) 120 ns
	b) 25 ns	d) 65 ns	f) ništa od navedenoga

18. Uporabom logičkih blokova (CLB) sklopa FPGA potrebno je ostvariti sklop koji na izlazima ciklički generira slijed 3, 7, 2, 1. CLB-ovi su temeljeni na dvoulaznom LUT-u, D bistabilu i multipleksoru čiji je adresni ulaz označen sa s . Kada je $s=0$, na izlaz CLB-a propušta se izlaz LUT-a, dok se za $s=1$ na izlaz CLB-a propušta izlaz bistabila. Djelomično programiranje sklopa već je napravljeno. Što treba upisati u LUT CLB-a označenog s L4?

The diagram shows a chain of five CLB blocks labeled L1 through L5. Each CLB contains a 2-to-1 multiplexer with inputs a_1 and a_0 and a select input s . The connections are as follows:

- L1:** $a_1 = 0$, $a_0 = 1$, $s = 1$. Its output is connected to the a_1 input of L2.
- L2:** $a_1 = 1$, $a_0 = 0$, $s = 1$. Its output is connected to the a_1 input of L3.
- L3:** $a_1 = ?$, $a_0 = 1$, $s = 0$. Its output is connected to the a_1 input of L4 and also to output O_2 .
- L4:** $a_1 = ?$, $a_0 = ?$, $s = 0$. Its output is connected to the a_1 input of L5 and also to output O_1 .
- L5:** $a_1 = ?$, $a_0 = ?$, $s = 0$. Its output is connected to the a_0 input of L1 and also to output O_0 .

a) 1101

b) 0111

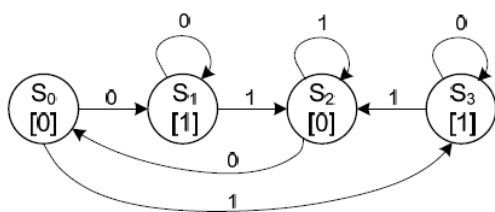
c) 1011

d) 0011

e) 1110

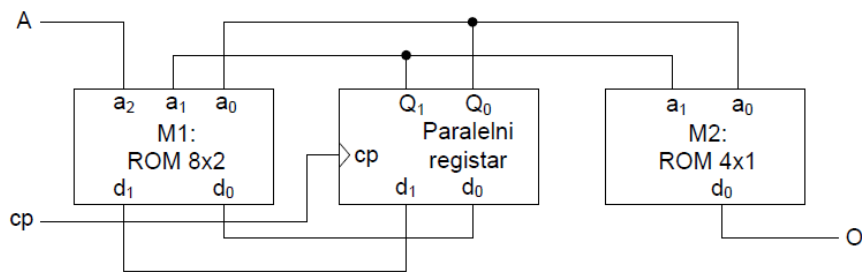
f) ništa od navedenog

10.	Memorija 256×2 bita ima 2 ½ D organizaciju. Koliko logičkih riječi u tom slučaju sadrži jedna fizička riječ, ako se na adresni dekođer retka dovodi 5 bitova adrese?
a) jednu logičku riječ	d) šesnaest logičkih riječi
b) četiri logičke riječi	e) trideset i dvije logičke riječi
c) osam logičkih riječi	f) ništa od navedenog



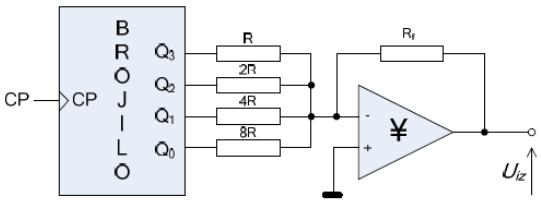
Slika 1

16. Zadan je Mooreov automat (slika 1). Potrebno ga je ostvariti sklopom prikazanim u nastavku, koji se sastoji od sinkronog registra s paralelnim ulazima i paralelnim izlazima, te dvije memorije; M1 kapaciteta 8x2 bita te M2 kapaciteta 4x2 bita. Koristi se kodiranje stanja koje stanju S_i pridjeljuje kodnu riječ koja odgovara binarnom zapisu od i . Što će u M1 pisati na lokaciji 4 a što u M2 na lokaciji 3? U odgovorima su ponuđeni dekadski zapisane vrijednosti.



- a) 3, 0
b) 0, 1
c) 3, 1
d) 1, 0
e) 2, 1
f) ništa od navedenoga

7.	Na raspolaganju je težinski 4-bitni D/A pretvornik s operacijskim pojačalom (za kod 8421). Ako je najveći otpor u težinskom dijelu $8\text{ k}\Omega$, otpor R_F u povratnoj vezi operacijskog pojačala $4\text{ k}\Omega$, a referentni napon napajanja $U_{REF}=5\text{ V}$, koliki će se napon dobiti na izlazu pretvornika kada na ulaz dovedemo podatak 4?
	a) -2 V b) -4 V c) -5 V d) -8 V e) -10 V f) ništa od navedenog

19.	<p>Na sinkrono binarno brojilo spojen je D/A pretvornik. Signal takta je simetrični, poluperiode $500\text{ }\mu\text{s}$. U trenutku $t=0$, brojilo se nalazi u stanju 0, i započinje poluperioda signala takta vrijednosti 0. Koji će se napon pojaviti na izlazu pretvornika u trenutku $t = 2,2\text{ ms}$? Za pretvornik je poznato: $R = 1\text{ k}\Omega$, $R_F = 2\text{ k}\Omega$. Pretpostavite da izlazi brojila imaju otpor $0\text{ }\Omega$, te da na njegovu izlazu logička 0 odgovara naponu 0 V a logička 1 naponu $+5\text{ V}$.</p> 
	a) $-3,125\text{ V}$ b) $-3,75\text{ V}$ c) $-2,5\text{ V}$ d) $-1,25\text{ V}$ e) $-3,3\text{ V}$ f) ništa od navedenog

19.	Neki 5-bitni A/D pretvornik sa sukcesivnom aproksimacijom ulazni napon od 12 V pretvara 200 ns . Koliko vremena će mu trebati za pretvorbu upola manjeg ulaznog napona?
	a) 50 ns b) 100 ns c) 200 ns d) 400 ns e) 800 ns f) ništa od navedenog