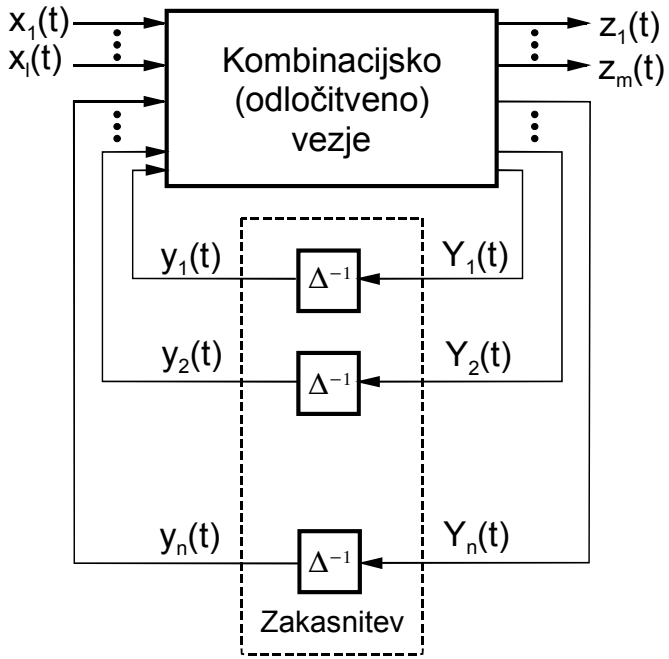
9 ASINHRONSKA SEKVENČNA VEZJA

9.1 Modeli asinhronskih sekvenčnih vezij

9.1.1 Kanalski model asinhronskega sekvenčnega vezja



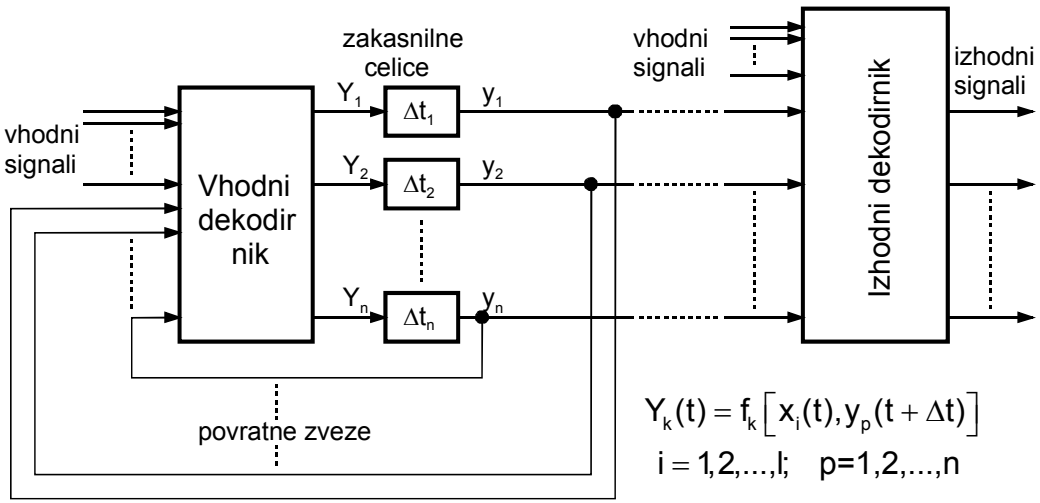
x = $(x_1(t), x_2(t),\ldots,x_i(t),\ldots,x_l(t))$ – primarni vhodni vektor

y = $(y_1(t), y_2(t),\ldots y_k(t),\ldots, y_n(t))$ – sekundarni vhodni vektor

z = $(z_1(t), z_2(t),\ldots, z_j(t),\ldots,z_m(t))$ – vektor izhodnih spremenljivk

Y = $(Y_1(t), Y_2(t),\ldots,Y_k(t),\ldots,Y_n(t))$ – vektor vzbujalnih spremenljivk

9.1.2 Serijski model asinhronskega sekvenčnega vezja

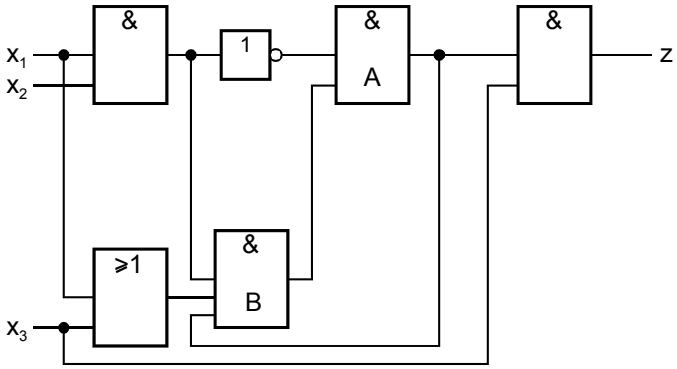


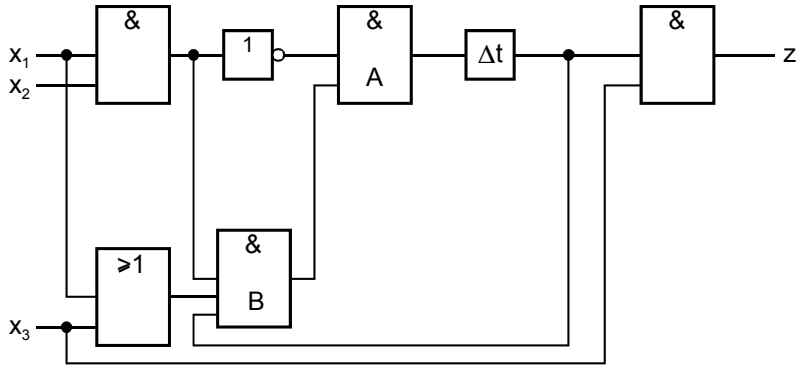
$$Y_k(t) = f_k\left[x_i(t), y_p(t+\Delta t)\right]$$

$i = 1,2,\ldots,l;\quad p=1,2,\ldots,n$

$$z_j(t) = \lambda_j\left[x_i(t), y_p(t)\right]$$

$j = 1,2,\ldots,m$



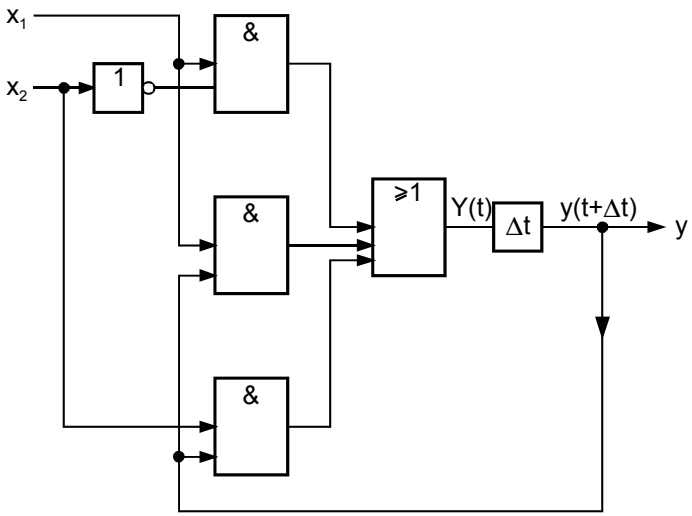


9. 2 Analiza asinhronskih sekvenčnih vezij

Obnašanje asinhronskih sekvenčnih vezij podamo lahko na več načinov:

- vzbujalna tabela (vzbujalni diagram)
- diagram prehajanja stanj – osnovnih ali združenih
- tabela prehajanja stanj – osnovnih ali združenih

9. 2. 1 Osnovna oblika asinhronskega vezja (Fundamental mode)



$Y(t) = y(t + \Delta t)$.

ali pa

$Y(t) \neq y(t + \Delta t)$.

Y(t) notranje stanje ali vzbujalna spremenljivka

$y(t+ \Delta t)$ pa signal povratne zveze ali sekundarna vhodna spremenljivka.

Če velja prvi pogoj, je vezje v stabilnem stanju

Za drugi pogoj pa pravimo, da je vezje na prehodu z enega stabilnega stanja v drugo. To stanje je torej nestabilno.

$Y(t) = f [x_1 (t), x_2(t) , y(t+ \Delta t)]$

Za opazovano vezje je to:

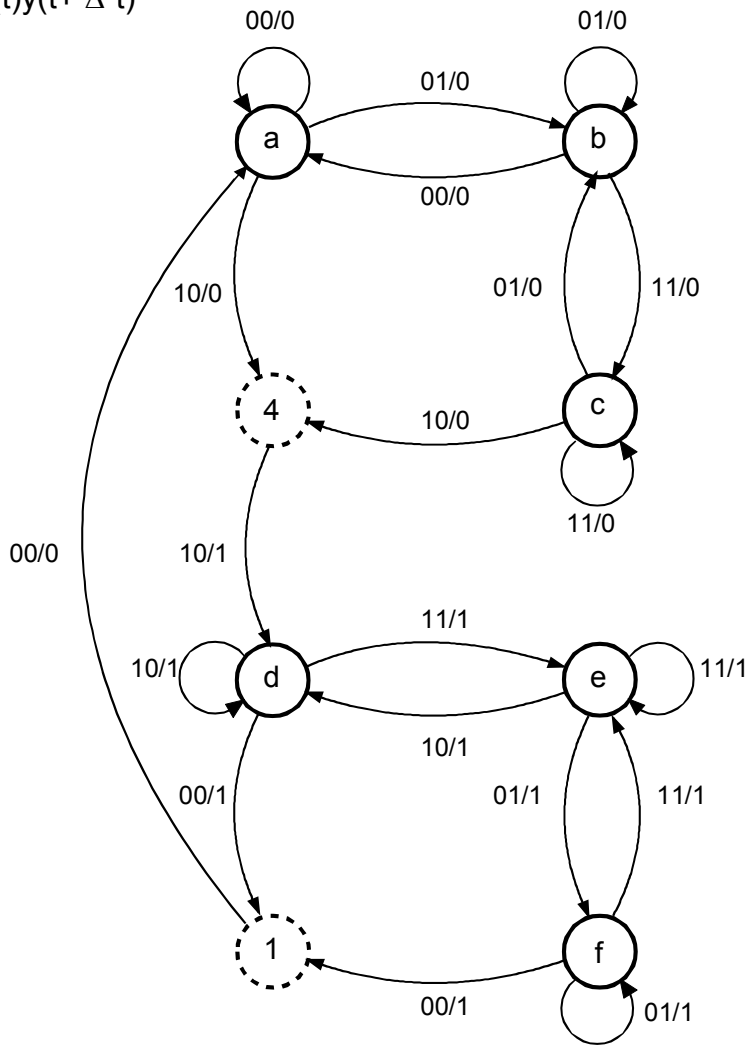
$Y(t) = x_1(t)\bar{x}_2(t)+ x_1(t)y(t+ \Delta t) + x_2(t)y(t+ \Delta t)$

Spremembe vhodnih signalov

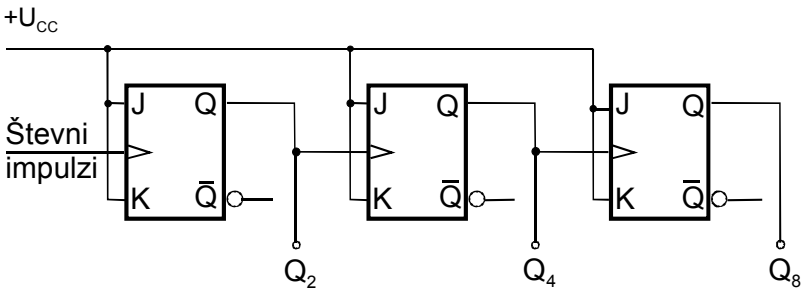
x_1x_2 / y	00	01	11	10
0	(0) 0	(0) 2	(0) 6	1 4
1	0 1	(1) 3	(1) 7	(1) 5

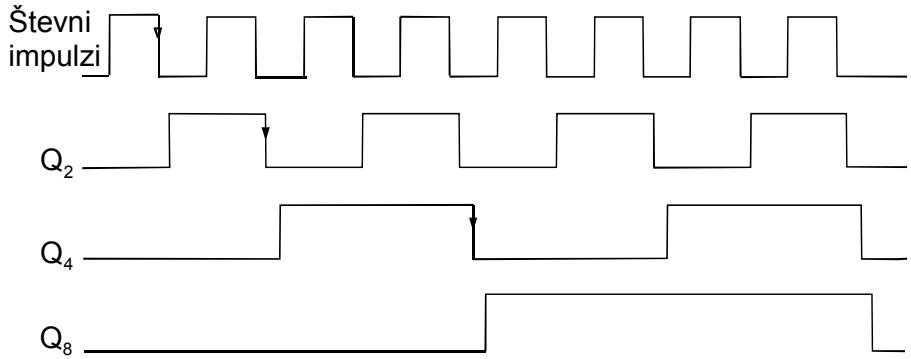
Spremembe signalov povratnih zvez

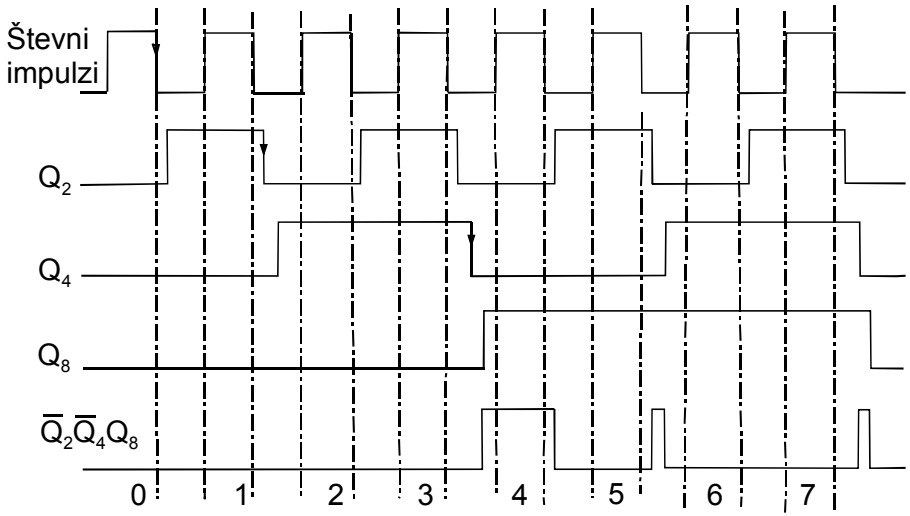
x_1x_2 / y	00	01	11	10
0	(a) 0	(b) 2	(c) 6	d 4
1	a 1	(f) 3	(e) 7	(d) 5

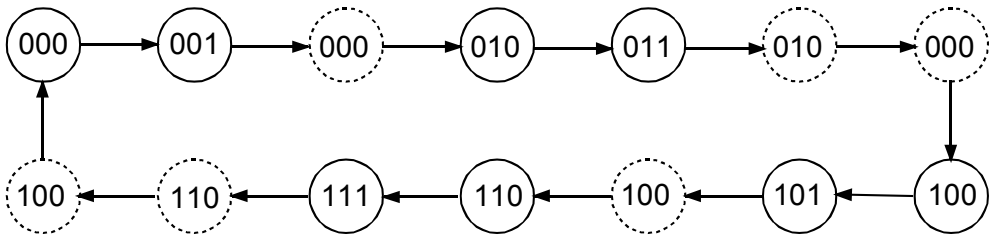


9. 2. 2 Analiza ripple števca







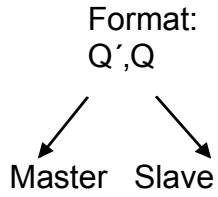


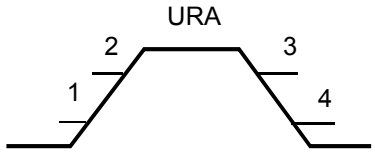
$$t_z = \sum_{i=1}^{n} t_i, \qquad t_z = n\Delta t,$$

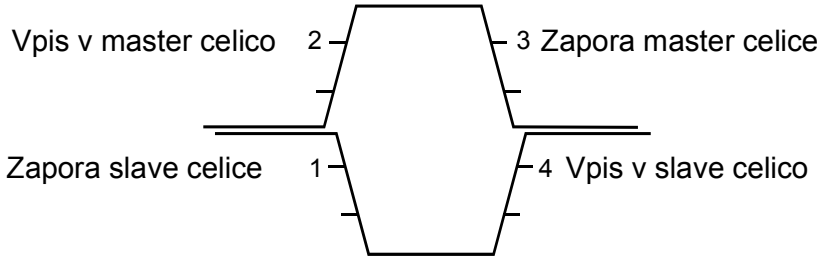
$$\Delta t = 10ns; \quad n = 8; \quad t_z = 80ns$$

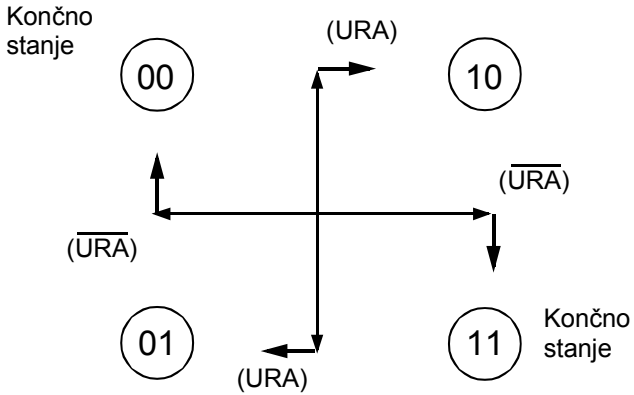
$$f_{max} = \frac{1}{t_z} = \frac{1}{80ns} = 12,5MHz$$

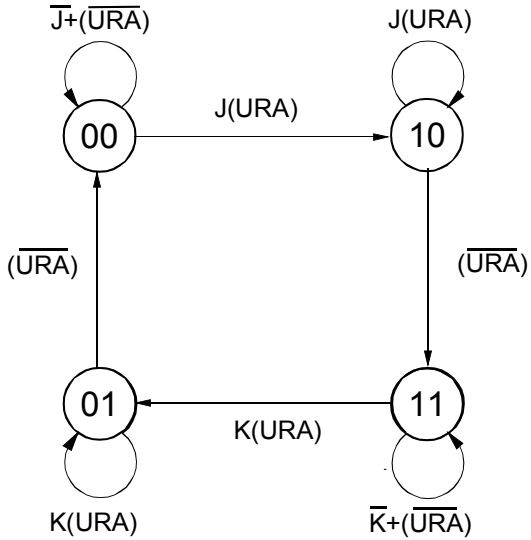
9. 2. 3 Asinhronska analiza spominskih celic s predpomnjenjem











9. 3 Asinhronski sekvenčni avtomati

9. 3. 1 Uvod in osnovne karakteristike

Kadar je širina “urinih” impulzov primerljiva z zakasnitvami posameznih sklopov vezja moramo delovanje vezja obravnavati v luči asinhronske analize. Urin impulz, če obstoja, postane tako neodvisen vhod.

V veliki večini primerov ni vhodnih sprememb, ki bi se dogajale ob ekvidistantnih časovnih intervalih.

Spremembe stanj bodo torej nastopale v spremenljivih diskretnih časih naključno razmaknjenih med seboj.

Da bomo lahko ta vezja podredili dvovrednostni logiki, bomo morali privzeti določene predpostavke in postaviti nekatere neizbežne omejitve. Ena izmed teh je Huffmanov pogoj !

Še več – omejitev moramo postaviti tudi na frekvenco ponavljanja vhodnih sprememb.

Spremembe vhodov se smejo dogajati samo v časovnih intervalih v katerih se celotno vezje zadržuje v stabilnem stanju, povzročenem s prejšnjo spremembo na vhodu.

9. 3. 1. 1 Jezik avtomata in tabela prehajanja stanj

Jezik avtomata, kot sredstvo za interpretacijo algoritma, mora biti metodološko prirejen sintezi in analizi teh vezij podobno, kot smo imeli pri sinhronih.
Kljub temu pa nastopajo nekatere pomembne razlike. Ena teh je:

Totalno stanje avtomata – kombinacija vhodnega in notranjega stanja.

Tu je potrebno strogo ločiti med totalnim, notranjim in vhodnim stanjem.

Za vsako totalno stanje tabela prehajanja stanj določa naslednje notranje stanje in izhodno stanje v obliki kombinacije izhodnih spremenljivk.
Množico totalnih stanj bomo označevali takole:

$$T^A = \{\delta_1, \delta_2, \delta_3, ..., \delta_r\}$$

pri čemer je na primer:

$$\delta_i = \delta_j^k = (s_j, x^k)$$

kombinacija notranjega stanja $s_j \in \mathbf{S}$ in vhodnega stanja $x^k \in \mathbf{X}$

Analogno vpeljimo tudi oznake izhodnih stanj:

$z_m^n \in \mathbf{Z}$

9. 3. 1. 2 Tabela posamičnih izhodnih prehodov (PIP tabela):

x	0	1
1	① , 00	2 , 00
2	② , 11	3 , 01
3	4 , 10	③ , 01
4	④ , 10	3 , 01

a)

x	0	1
1	① , 00	3 , 00
2	② , 11	3 , 01
3	4 , 10	③ , 01
4	④ , 10	3 , 01

b)

x: $0 \rightarrow 1$

Očitno je torej, da je trajanje spremembe izhodnega stanja različno.

Vendar, če to ne povzroča nobenih posledic, lahko obe tabeli prehajanja stanj jemljemo kot ekvivalentni, glede na to, da se pri obeh za enake vhodne sekvence generirajo na izhodu enake izhodne sekvence.
V primeru, da po PIP tabeli vsako prehajanje pelje direktno v stabilno stanje, pravimo, da asinhronski avtomat deluje v normalnem načinu.
Vsako PIP tabelo lahko transformiramo v tabelo normalnega načina delovanja, če je le trajanje spremembe izhoda nepomembno.

9. 3. 1. 3 Tabela večkratnih izhodnih prehodov (VIP tabela)

x	0	1
1	2 , 10	①, 11
2	3 , 00	1 , 10
3	③, 01	2 , 01

a)

x_1x_2	x^1 00	x^2 01	x^3 11	x^4 10
1	①, 00	2 , 10	①, 11	-,-
2	3 , 11	②, 10	3 , 00	-,-
3	③, 11	4 , 01	4 , 01	-,-
4	1 , 00	④, 01	2 , 10	-,-

b)

Če totalni stanji (2,x^2) ali (4,x^2) vzbudimo z x^3, avtomat ne more več priti v stabilno stanje!

Iz teh primerov zaključimo, da vse tabele, ki podajajo normalen način delovanja, podajajo hkrati tudi osnovni način delovanja, toda tabele osnovnega načina delovanja ne podajajo vedno tudi normalni način delovanja asinhronskega avtomata.

10. 3. 1. 4 Impulzno vzbujanje asinhronskih avtomatov

x_1x_2	00	01	11	10
1	①, 0	3 , 1	①,	4 , 0
2	②, 0	4 , 1		4 , 0
3	③, 1	③, 1	1 , 0	4 , 0
4	3 , 1	④, 0	1 , 0	④, 0

	x^1	x^2	x^3	x^4	x^5
1	①, S	1 , 0	3 , 1	1 , 0	4 , 0
2	②, S	2 , 0	4 , 1	-,-	4 , 0
3	③, S	3 , 1	3 , 1	1 , 0	4 , 0
4	④, S	3 , 1	4 , 0	1 , 0	4 , 0

Stolpci x^2 do x^5 ustrezajo stolpcem leve tabele. Vsak stolpec x^i, kjer je $i \neq 1$ predstavlja impulzni vhod. Stolpec x^1 pa predstavlja odsotnost kakršnega koli vzbujanja.

9. 3. 2 Metode minimizacije asinhronskih avtomatov

9. 3. 2. 1 Primitivna tabela prehajanja stanj

Najbolj občutljiva faza sinteze se nanaša na pravilno formuliranje, oziroma na natančno jezikovno interpretacijo delovanja avtomata. Nabor vseh funkcionalnih karakteristik in definicij končnega algoritma je večinsko osnovan na intuiciji, zato nobena metodologija sinteze ne more v nadaljnjih korakih sinteze izločiti netočnosti ali nepravilnosti algoritma.

Izhodiščna osnova za natančno metodološko sintezo je tabela prehajanja osnovnih stanj.

Določitev te tabele se izvrši na osnovi verbalnega, matričnega opisa ali pa grafičnega v obliki diagrama prehajanja stanj.

Diagram prehajanja nam močno olajša kontrolo in morebitno potrebno korekcijo funkcionalnih karakteristik, ki svojo kanonično obliko dobijo prav z tabelo prehajanja osnovnih stanj.

Pretvorba jezikovnih izrazov v tabelo prehajanja osnovnih stanj je v principu enaka, kot pri sinhronih avtomatih.

Kljub temu pa določene posebnosti, ki nastopajo pri asinhronih avtomatih zahtevajo nekoliko več pazljivosti, kar se odraža tudi pri strukturi te tabele prehajanja stanj.

Začetno obliko te tabele imenujemo primitivna tabela prehajanja stanj – tudi tabela prehajanja osnovnih stanj; skladno z osnovnim načinom delovanja asinhronskega avtomata.

Izhajamo namreč iz predpostavke, da za vsako vhodno vzbujanje obstoja stabilno stanje v avtomatu, kateremu je pridruženo odgovarjajoče izhodno stanje. Če ta predpostavka nebi veljala, potem celoten poskus, da z avtomatom realiziramo vnaprej definirane in programirane pretvorbe informacije postane absurden – avtomat nujno nebi sploh moral priti v stabilno stanje.

S tem zavestno vnašamo redundanco, saj tako vsakemu stabilnemu (totalnemu) stanju dodelimo svojo vrstico v tabeli (primitivni) prehajanja stanj.

Vendar moramo še enkrat poudariti, da to redundantnost uvajamo zaradi maksimalne zanesljivosti glede točnosti in vernosti algoritemskega prikaza.

Z metodami reduciranja stanj nato zmanjšamo število stanj na minimalno potrebno, čeprav so te metode pogosto težavne in dolgotrajne.

9. 3. 2. 2 Reduciranje primitivne tabele prehajanja stanj.

Redukcija primitivne tabele zajema, poleg že znanih metod iskanja ekvivalentnih razredov stanj, tudi postopek spajanja in nekatere posebnosti, ki so vezane na specifični način delovanja.
Reduciranje primitivne tabele vključuje torej v sebi tudi zmanjšanje števila notranjih stanj asinhronskega avtomata.

Spomnimo se, da je stabilno stanje definirano z vhodnim in notranjim stanjem.

Zato ni razloga, da ista koda notranjega stanja nebi bila pridružena različnim notranjim stanjem primitivne tabele stanj.

To pomeni, da se lahko v isti vrstici nahajata dve oziroma več (totalnih) stabilnih stanj, do katerih spremembe prihaja izključno zaradi sprememb vhodnih stanj.

S tem avtomatično zmanjšujemo število vrstic primitivne tabele.

Zavedati pa se moramo, da s tem zmanjšujemo tudi število sekundarnih vhodnih spremenljivk s katerimi kodiramo notranja stanja.
Kasneje bomo spoznali, da za spajanje ni neobhodno potrebna identičnost izhodnih stanj.

Dalje, omejitev, ki se nanaša na posamično spremembo stanja vhodnih spremenljivk, pogosto omogoča reduciranje tabele prehajanja stanj.

To možnost se lahko enostavno modelira z nespecificiranim stanjem v primitivni tabeli.

Toda, če v tabeli obstojajo vrstice z dvema ali več stabilnimi stanji, se lahko zgodi, da za nekatere stolpce znotraj te vrstice ni več možno skupno sekundarno vzbujanje.

Opazujmo naslednjo tabelo prehajanja:

x_1x_2	00	01	11	10
1	①, 0	2 , 0	- , -	3 , 0
2	1 , 0	②, 0	②, 0	4 , 1
3	4 , 1	2 , 0	③, 1	③, 0
4	④, 1	2 , 0	2 , 0	4 , 1

Pripadajoča primitivna tabela prehajanja:

x_1x_2	00	01	11	10
1	①, 0	2 , 0	- , -	3 , 0
2	1 , 0	②, 0	2´, 0	- , -
2´	- , -	2 , 0	②´, 0	4 , 1
3	4´, 1	- , -	3´, 1	③, 0
3´		2 , 0	③´, 1	3 , 0
4	4´, 1	- , -	2´, 0	④, 1
4´	④, 1	2 , 0	- , -	4 , 1

x_1x_2	00	01	11	10
1	①, 0	2, 0	-, -	3, 0
2	1, 0	②, 0	2´, 0	-, -
2´	-, -	2, 0	②´, 0	4, 1
3	4´, 1	-, -	3´, 1	③, 0
3´	-, -	2, 0	③´, 1	3, 0
4	4´, 1	-, -	2´, 0	④, 1
4´	④´, 1	2, 0	-, -	4, 1

A: (1,2)

B: (2´, 4, 4´)

C: (3, 3´)

x_1x_2	00	01	11	10
A	Ⓐ, 0	Ⓐ, 0	B, 0	C, 0
B	Ⓑ, 1	A, 0	Ⓑ, 0	Ⓑ, 1
C	B, 1	A, 0	Ⓒ, 1	Ⓒ, 0

Razumljivo je, da vsako PIP tabelo lahko pretvorimo nazaj v primitivno tabelo prehajanja stanj, ki je izhodiščna v postopku minimizacije.

V tabelah prehajanja v katerih obstoja večkratno prehajanje izhodnih spremenljivk (VIP) mora biti izhod, ki je pridružen nestabilnemu stanju, enak izhodu, ki je pridružen stabilnemu stanju in to na začetku ali na koncu prehoda.

Če isti izhod pridružimo nestabilnim stanjem in končnemu stanju to samo skrajša čas prehajanja.

Če te učinke zanemarimo, lahko obe tabeli obravnavamo kot ekvivalentni.

Torej, če je neko stanje nestabilno v posameznem stolpcu, je lahko njegovo naslednje stanje in pridružen izhod enako stabilnemu stanju in njegovemu pridruženemu izhodu v tem stolpcu.

Namreč, če je s_i nestabilno stanje v stolpcu x^k, a s_j stabilno, ki ga bo to nestabilno stanje doseglo, potem lahko izenačimo:

$$s_j = S(s_i, x^k) \text{ in } Z(s_i, x^k) = Z(s_j, x^k)$$

S tem postopkom pretvarjamo PIP tabelo v tabelo prehajanja za normalni način delovanja.

Oglejmo si primer avtomata, ki nima ekvivalentnih parov stanj.

x_1x_2	00	01	11	10
1	①, 0	2, 0	2, 0	①, 0
2	4, 1	3, 0	②, 1	②, 1
3	1, 0	③, 0	4, 1	-, -
4	④, 1	3, 1	④, 1	④, 1

x_1x_2	00	01	11	10
1	①, 0	3, 0	2, 1	①, 0
2	4, 1	3, 0	②, 1	②, 1
3	1, 0	③, 0	4, 1	-, -
4	④, 1	3, 0	④, 1	④, 1

x_1x_2	00	01	11	10
1	①, 0	①, 0	2, 1	①, 0
2	②, 1	①, 0	②, 1	②, 1

9. 3. 2. 3 Spajanje vrstic

	000	001	010	100	$z_1z_2z_3$
1	①	2	4	6	000
2	3	②	-	-	001
3	③	2	4	6	001
4	5	-	④	-	010
5	⑤	2	4	6	010
6	7	-	-	⑥	100
7	⑦	2	4	6	100

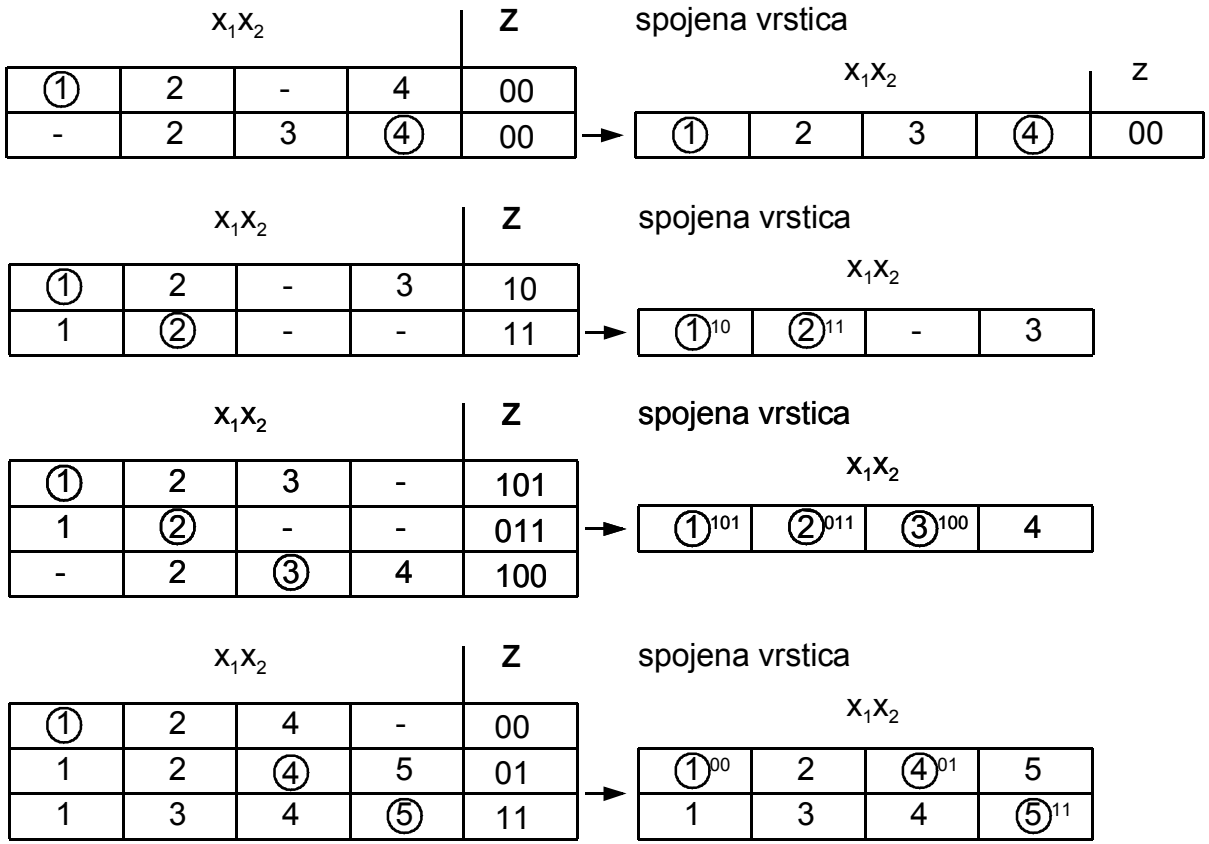
	00	01	11	10	z
α	4	①	2	3	0
β	4	1	②	3	1

	00	01	11	10
(α,β)	4	① 0	② 1	3

Pri spajanju posameznih vrstic se moramo držati naslednjih splošnih pravil:

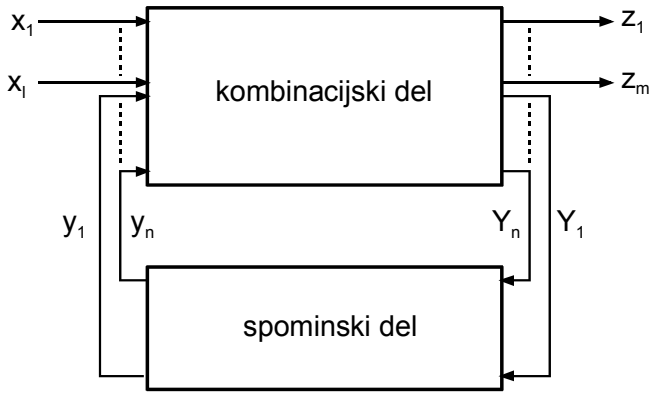
Dve ali več vrstic lahko spojimo samo v primeru, da imata oziroma imajo v istoležnih stolpcih identična stanja (stabilna ali nestabilna); ne glede na njihove izhode.

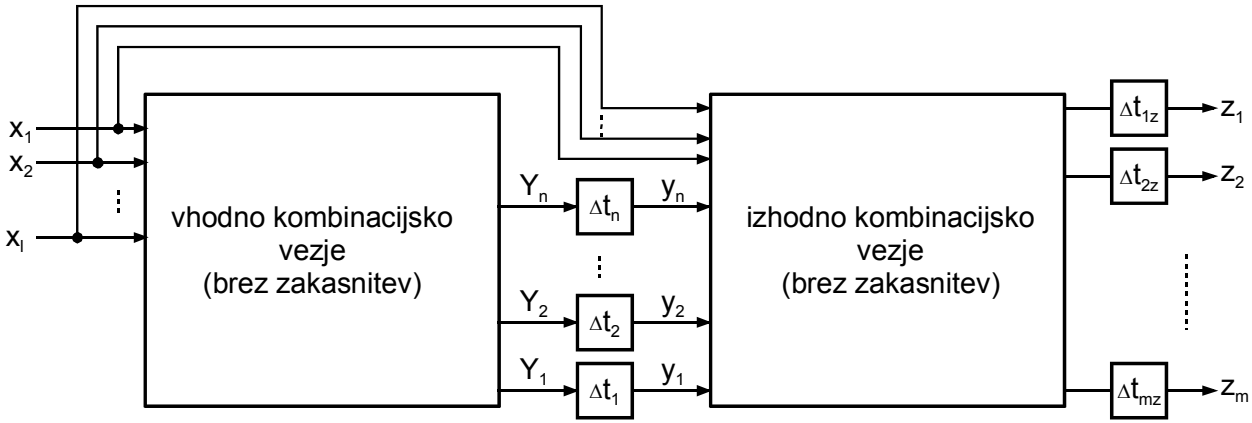
Katerokoli stanje, ki je obkroženo ali neobkroženo se lahko spoji z redundatnim stanjem, če s tem ni kršeno pravilo identičnosti v ostalih stolpcih vseh spojenih vrstic.



- Vsak par simbolov (lahko tudi več), ki ni obkrožen v stolpcu moramo nadomestiti z istim neobkroženim simbolom v spojeni vrstici
- Če je eden izmed simbolov obkrožen, ga je potrebno nadomestiti z istim obkroženim simbolom v spojeni vrstici
- Če se v nekem stolpcu nahajata obkrožen ali neobkrožen simbol in redundatno stanje, moramo v spojeni vrstici v tem stolpcu vnesti obkrožen simbol.
- Če so v istem stolpcu vseh vrstic, ki se spajajo sama redundatna stanja, potem je v tem stolpcu spojene vrstice tudi redundatno stanje.
- Izhode v spojeni vrstici lahko izpustimo, če se spajajo vrstice z različnimi izhodi. To informacijo še vedno lahko dobimo iz primitivne tabela prehajanja stanj. Priporočljivo pa je , da jih pripišemo k stabilnim (totalnim stanjem) znotraj odgovarjajočega polja spojene vrstice.

9. 4 Zakasnitve v asinhronskih vezjih





Izhodi so tako izraženi z naslednjo funkcionalno odvisnostjo:

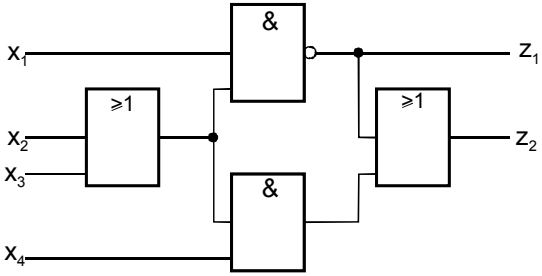
$$z_j(t+\Delta t_{j0}) = \Phi\left[x_1(t), x_2(t), \ldots, x_l(t); y_1(t+\Delta t_1), y_2(t+\Delta t_2), \ldots, y_n(t+\Delta t_n)\right]$$

Δt_{j0} - časovni interval v katerem se stabilizira j-ti izhod

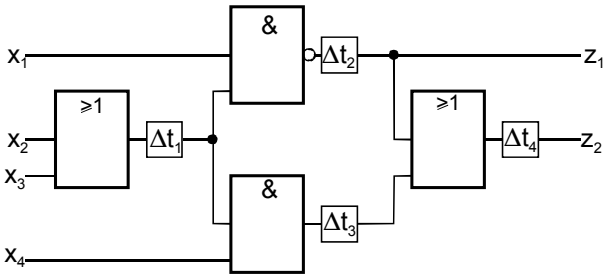
Končno izhodno stanje vektorja $\mathbf{z^1}$ lahko opazujemo šele po Δt_{1min} za vzbujanjem $\mathbf{x^i}$, pri čemer mora veljati:

$$(\Delta t_1)_{min} \geq (\Delta t_{10})_{max}$$

Primer koncentriranja zakasnitev:

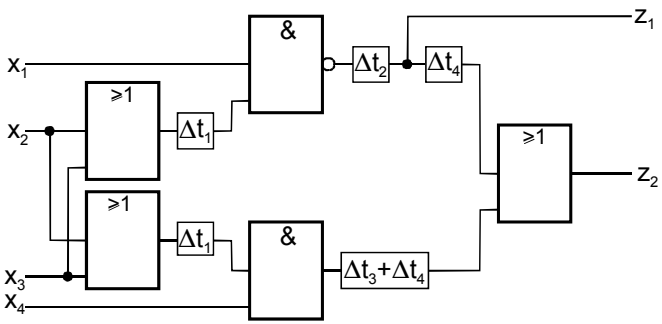


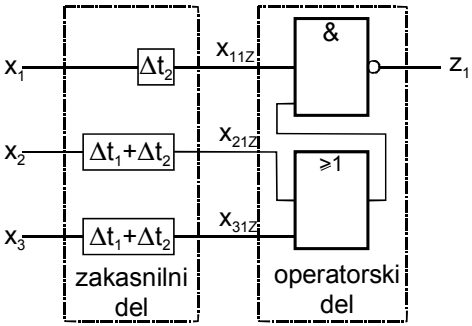
V tem vezju nastopajo sledeče zakasnitve:

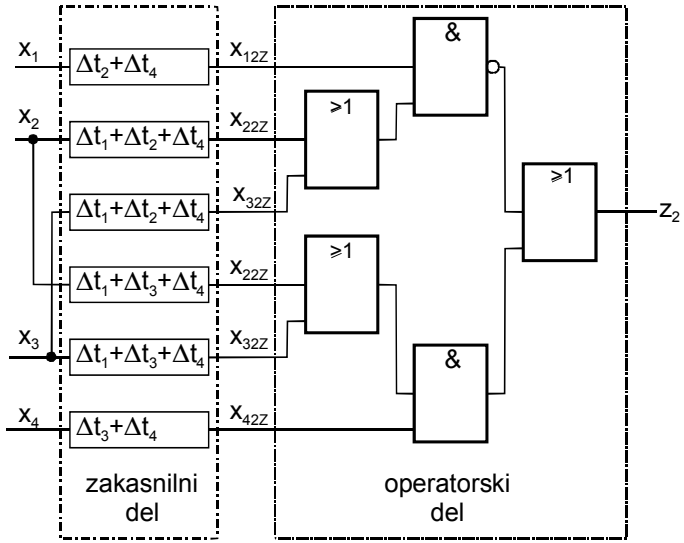


V zakasnitve je lahko vključen tudi čas preleta signala po spojnih vodih.

Prvi korak koncentriranja zakasnitev je sledeč:



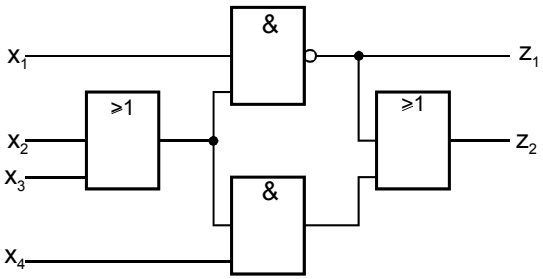




V prvem koraku analize časovnih dogodkov predpostavimo:

$$\Delta t_1 = \Delta t_2 = \Delta t_3 = \Delta t_4 = \Delta t_0$$

Operatorski del določimo iz osnovnega vezja, ki ga tako obravnavamo idealno:



$$z_1 = \overline{x}_1 + \overline{x}_2\overline{x}_3$$

$$z_2 = z_1 + x_2x_4 + x_3x_4$$

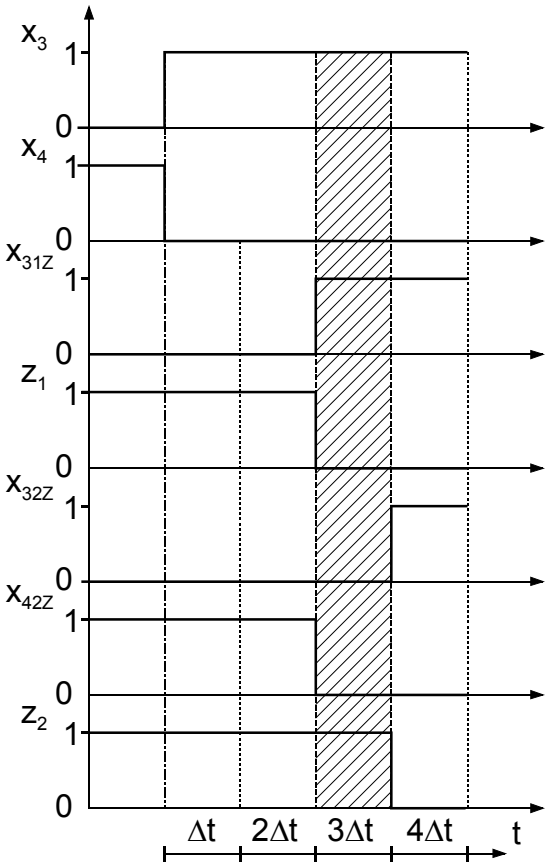
Vektor **x** lahko zavzame 16 (2^4) različnih vrednosti, kar predstavlja 16 različnih vhodnih dogodkov.
Vsako kombinacijo predstavimo s posebnim vektorjem, ki ga po dogovoru indeksiramo z zgornjim indeksom.

$$\mathbf{x}^0 \triangleq \begin{Bmatrix} \overline{x}_1 & \overline{x}_2 & \overline{x}_3 & \overline{x}_4 \\ 0 & 0 & 0 & 0 \end{Bmatrix}$$

Iz celotne množice: **X = {x^0 x^1,…,x^{15}}** opazujemo tiste kombinacije vektorjev, ki spreminjajo izhoda z_1 in z_2 ugotavljamo nastala prehodna stanja**.**

$$\mathbf{x}^9 = \begin{Bmatrix} x_1 & \overline{x}_2 & \overline{x}_3 & x_4 \\ 1 & 0 & 0 & 1 \end{Bmatrix};\ z_1(\mathbf{x}^9) = 1,\ \ z_1(\mathbf{x}^{10}) = 0$$

$$\mathbf{x}^{10} = \begin{Bmatrix} x_1 & \overline{x}_2 & x_3 & \overline{x}_4 \\ 1 & 0 & 1 & 0 \end{Bmatrix};\ z_2(\mathbf{x}^9) = 1,\ \ z_2(\mathbf{x}^{10}) = 0$$



Pomembno je opozoriti, da je v času 3Δt izhod v “nedefiniranem” stanju.

V tem času je namreč $z_1 = 0$; $z_2 = 1$.

9. 5 Kodiranje notranjih stanj in njihovo prehajanje

9. 5. 1 Kodiranje notranjih stanj

Kodiranje notranjih stanj (KNS) je eno izmed najbolj kritičnih opravil v postopku sinteze asinhronskega avtomata.

Poudariti je potrebno, da se ta problem v temelju razlikuje od kodiranja pri sinhronskih vezjih.

Pri sinhronskih je bilo dovolj, da smo zadovoljili enačbo: $k = l_d r$; oziroma $r = 2^k$.

Pri asinhronskih pa nastopi nova dimenzija kodiranja, od katere ni odvisna samo kompleksnost vezja, temveč tudi zanesljivost delovanja.

{x} / {s}	x = 0	x = 1
1	①	2
2	3	②
3	③	4
4	1	④

Koda A

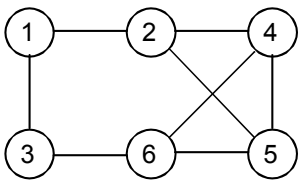
y_1	y_2
0	0
0	1
1	0
1	1

Koda B

y_1	y_2
0	0
0	1
1	1
1	0

Pri kodiranju notranjih stanj nam je lahko v veliko pomoč graf sosednih prehodov.

X / S	x_1x_2 = 00	01	11	10
1	①	2	3	①
2	4	②	②	-
3	1	③	③	-
4	④	5	④	-
5	6	⑤	2	-
6	⑥	3	4	⑥



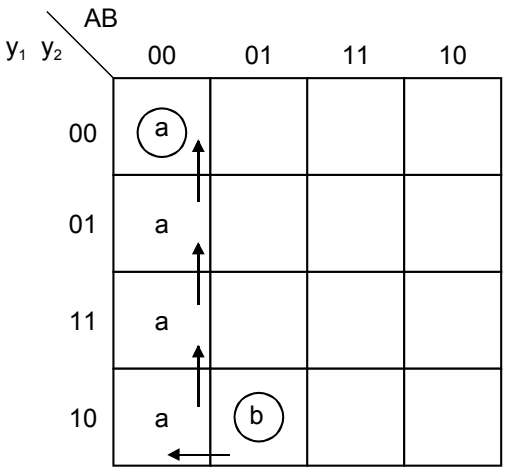
Stanja 2, 4, 5 morajo biti medsebojno sosedna!

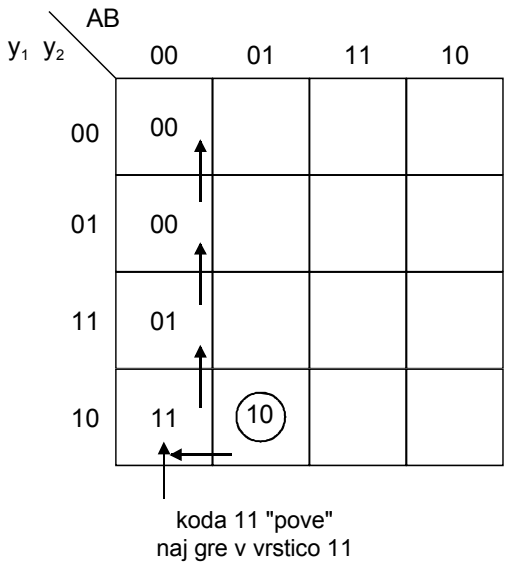
Isto velja za stanja 4, 5 in 6!

9. 5. 2 Prehajanje stanj v asinhronskih avtomatih

- ciklično prehajanje notranjih stanj
- sočasno prehajanje notranjih stanj

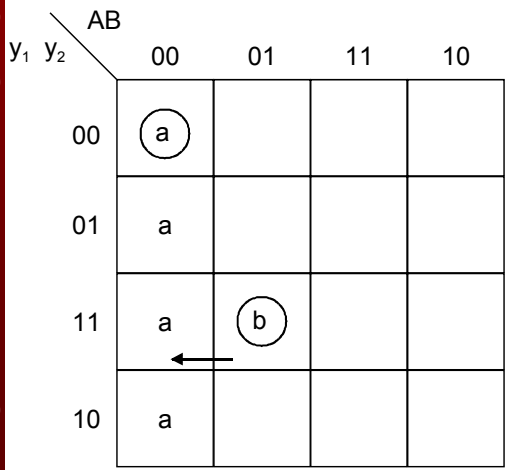
Pri cikličnem prehajanju stanj se istočasno spreminja samo ena sekundarna spremenljivka oziroma stanje povratne zveze.

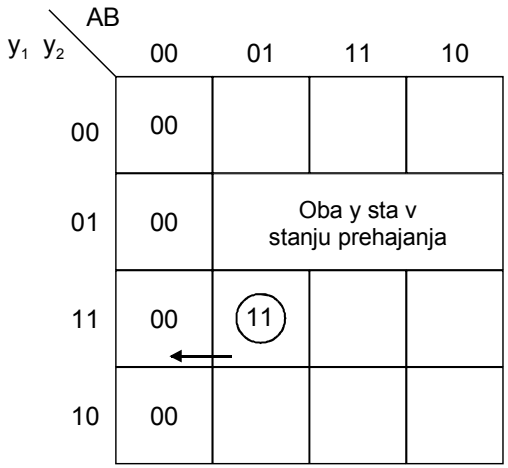




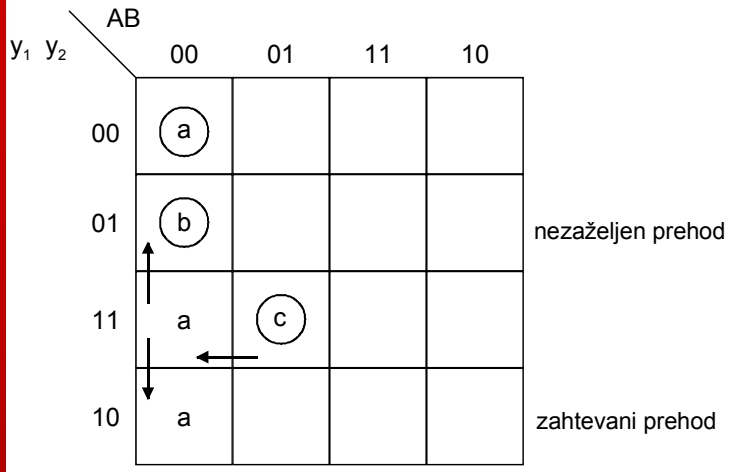
Pri sočasnem prehajanju nastopata dve možnosti tega prehajanja – kritično in nekritično sočasno prehajanje.

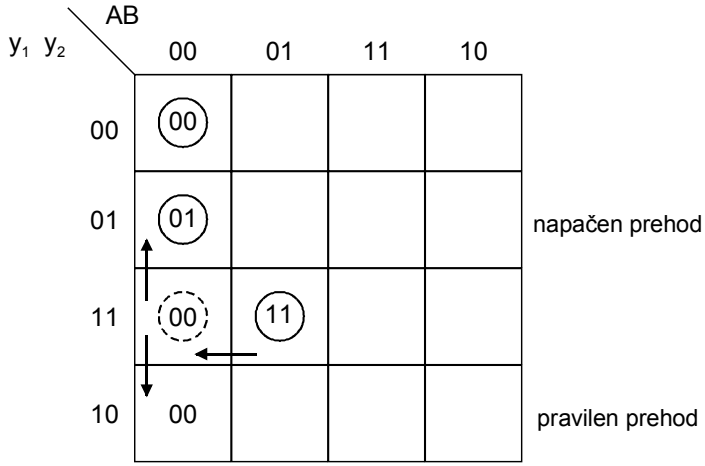
Nekritičen sočasni prehod notranjega stanja:



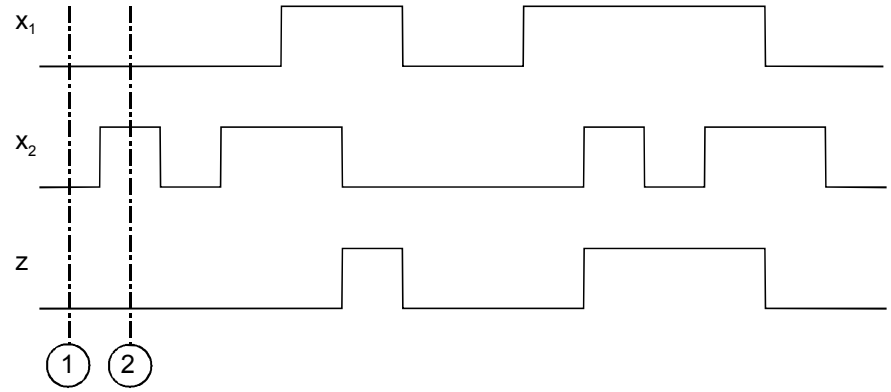


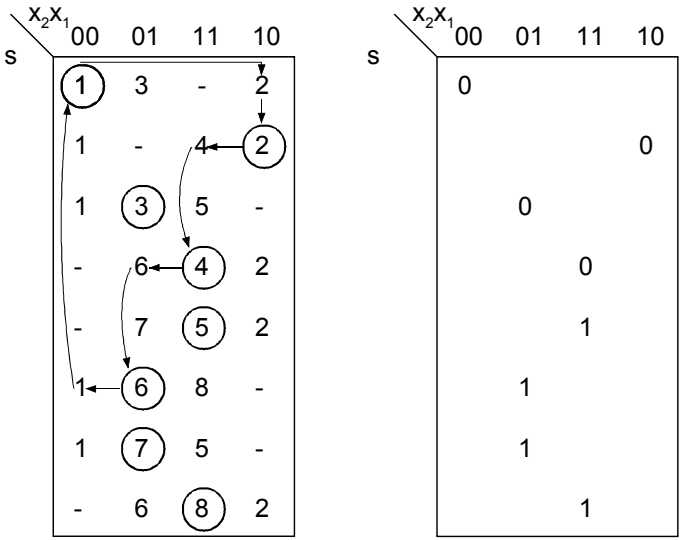
Kritičen sočasni prehod notranjega stanja:





Primer sinteze asinhronskega avtomata:





Določitev maksimalnih ekvivalentnosti:

(12), (13), (24), (5678)

(13), (24), (5678)

q		Δ^1q				z			
		00	01	11	10	00	01	11	10
(1,3)	(a)	(a)	(a)	c	b	0	0	-	0
(2,4)	(b)	a	c	(b)	(b)	0	-	0	0
(5,6,7,8)	(c)	a	(c)	(c)	b	-	1	1	-

Kodiranje notranjih stanj:

Obstojajo prehodi med vsemi tremi stanji: a – b, a – c in b – c !

Potrebujemo torej dve notranji spremenljivki: y_1 in y_2.

Vendar teh dveh ni mogoče kodirati tako, da nebi nastopilo sočasno prehajanje pri enem izmed prehodov.

Problem lahko rešimo na dva načina; z dodelitvijo dveh kod istemu stanju ali s proglasitvijo nespecificiranega stanja za nestabilno.

y_2y_1	q	$\Delta^1 q$			
		00	01	11	10
00	(a)	(a)	(a)	c	b
01	(b)	a	c	(b)	(b)
11	(c)	a	(c)	(c)	b
10	(c)	a	(c)	(c)	b

y_2y_1	q	$\Delta^1 q$			
		00	01	11	10
00	(a)	(a)	(a)	c	b
01	(b)	a	c	(b)	(b)
11	(c)	a	(c)	(c)	b
10	c	a	-	c	-

Sedaj lahko pristopimo k določitvi vzbujalnih funkcij in izhodne funkcije.

Če izberemo drugi način kodiranja dobimo za Y_1 in Y_2 naslednji Karnaughjev diagram:

y_2y_1 \ x_2x_1	00	01	11	10
00	(00)	(00)	10	11
01	00	11	(01)	(01)
11	00	(11)	(11)	01
10	00	-	11	-

y_2y_1 \ x_2x_1	00	01	11	10
00	0	0		0
01	0		0	0
11		1	1	
10				

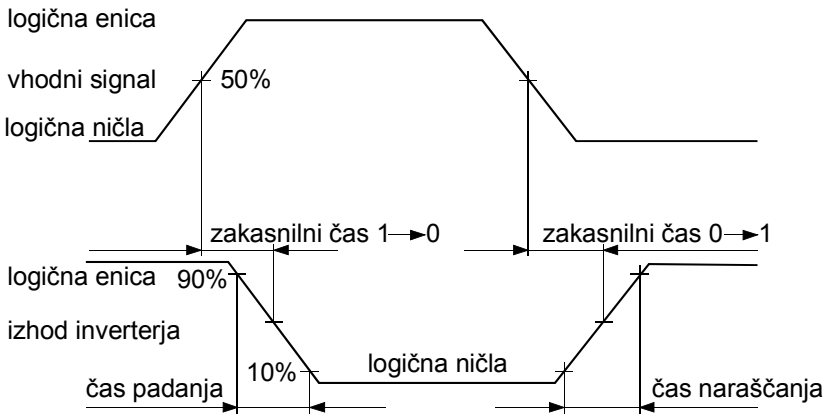
Iz njiju pa določimo vzbujalni funkciji in izhodno funkcijo.

Vaja: določi vzbujalni in izhodno funkcijo in nariši simbolni diagram z NAND elementi.

9. 6 Hazardni prehodi

9. 6. 1 Hazardni prehodi v kombinacijskih vezjih

V fizikalnem svetu vedno nastopajo zakasnitve med trenutkom, ko se spremeni vhod v nek element in trenutkom, ko se pojavi ustrezna sprememba tudi na izhodu.



Zakasnitev lahko izvira tudi iz povezav med elementi, če so te daljše ali hitrosti velike.

Čas prehodnega pojava ni vedno le preprost seštevek posameznih zakasnilnih časov.

Obstoja lahko več vzporednih poti, zato se prehodni pojav lahko odvija po več poteh hkrati.

Vsi logični izrazi, funkcijske odvisnosti, izjavnostne tabele in podobno veljajo le za stacionarno stanje!!!

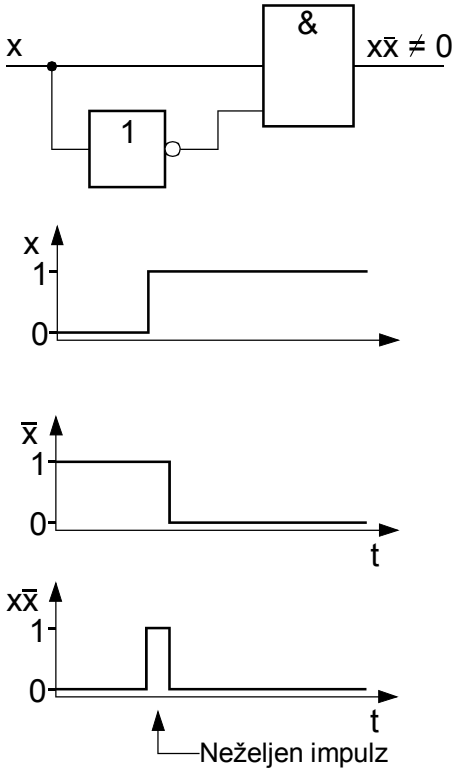
Zakasnitve elementov se lahko močno razlikujejo tudi znotraj iste družine

Samo logična analiza in sinteza ne zadoščata več, potrebna je tudi analiza prehodnih pojavov!

Definicija hazarda:

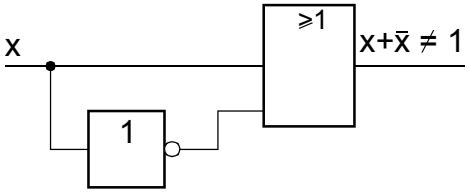
Hazard je dejansko ali potencialno nepravilno delovanje vezja zaradi prisotnosti zakasnitev v vezju

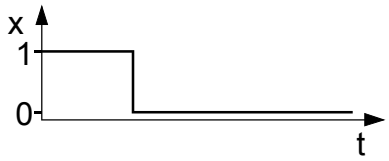
Da do hazarda pride zelo hitro lahko pokažemo na zelo enostavnem primeru invertiranja neodvisne spremenljivke:

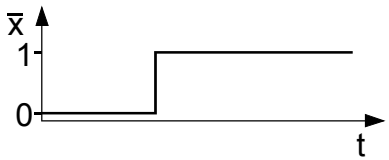


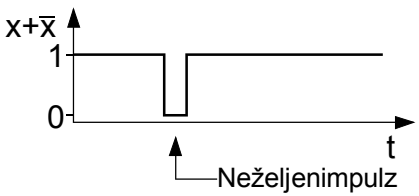
V času prehodnega pojava tako ne veljata niti postulata Booleove algebre p_4 in p_4'

Naslednje vezje bi moralo imeti enico na izhodu ne glede na vhod,









Zato je potrebno še enkrat poudariti, da logični zapisi Booleovih funkcij ne zajamejo časovnega dogajanja.

Umestno pa je vprašanje ali hazarden prehod pomeni nepravilno delovanje?

Podrobnejše opazovanje teh pojavov nam odkrije, da obstoja več vrst hazarda:

Statični hazard – dva prehoda brez logične spremembe izhoda

Glede na stacionarno stanje nastopata:

Statični hazard enice

Statični hazard ničle

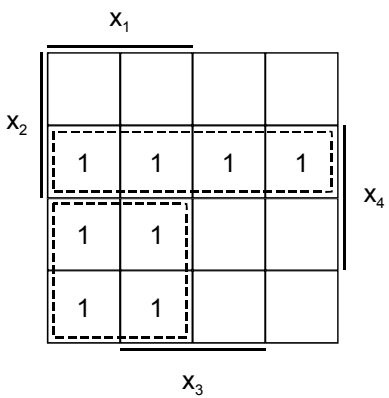
Katerega koli od teh hazardov lahko povzroči sprememba ene same vhodne spremenljivke ali pa več spremenljivk hkrati.

$$f = x_1\bar{x}_2 + x_2x_4$$

$x_1 = x_4 = 1$ je $f = 1$ ne glede na x_2

V prehodnem času nastopa:

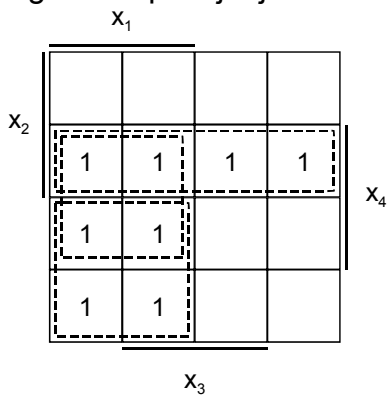
$$x_2 + \bar{x}_2 \neq 1$$



Za odpravljanje statičnega hazarda imamo dve možnosti

- logično odpravljanje
- odpravljanje z dodajanjem zakasnitev

Logično odpravljanje hazarda bi v prejšnjem primeru pomenilo dodajanje nove konjunkcije:



V splošnem torej velja, da minimizacija ustvarja potencialne možnosti različnih hazardov.

Da jih preprečimo moramo narediti vse konjunkcije oz. minterme medsebojno odvisne.

To pa je neka nova minimalna oblika – tista, ki ne ustvarja hazardnih prehodov!!

V literaturi lahko najdemo metode za takšno minimizacijo – Mc Cluskey.

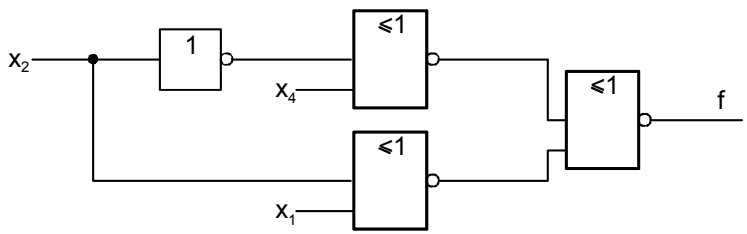
Vendar je potrebno opozoriti, da vse te metode temeljijo na disjunktivni obliki, ta pa v primeru uporabe NOR elementov ni optimalna.

$$f = x_1\bar{x}_2 + x_2x_4 + x_1x_4$$

Pripadajoča konjunktivna oblika je:

$$f = (\bar{x}_2 + x_4)(x_1 + x_2)$$

Realizacija z NOR pa naslednja:



Iz analize simboličnega diagrama lahko ugotovimo, da bo sedaj mogoč hazardni prehod pri stanju vhoda $x_1 = x_4 = 0$.

Prehod x_2 iz nič na ena bo povzročil kratkotrajno enico na izhodu.

V tem primeru gre za hazardni prehod pri ničli !!

V nasprotju s statičnim hazardom, pri katerem se stacionarno stanje izhoda ne spreminja, se pri dinamičnem stacionarno stanje zamenja.

Pri tej vrsti hazarda nastopajo zato najmanj tri spremembe izhoda.

Sprememb je lahko tudi več, vendar njihovo število je vedno liho.

Sproži ga lahko sprememba ene same spremenljivke, ali pa sočasna sprememba več vhodnih spremenljivk.

Posledica takšnih hazardnih prehodov je lahko sprožitev ali preklop v vezju, ki je nanj priključeno.

Posebej nevarno je če med tema vezjema obstoja povratna zveza.

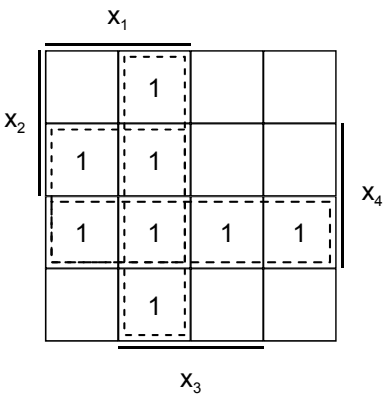
Poglobljena analiza dogajanja med dinamičnimi hazardnimi prehodi nas pripelje do spoznanja, da so za nastanek dinamičnega hazarda potrebne najmanj tri različne poti z različnim zakasnilnim časom za isto vhodno spremenljivko.

Dinamični hazard je v splošnem posledica:

- konjunktivne oblike realizacije
- predolgih povezav pri ekstremno hitrih vezjih

Za dinamični hazard so še zlasti občutljive NOR/NAND izvedbe – inverter na izhodu.

$$f = \overline{x}_2 x_4 + x_1 x_3 + x_1 x_4$$



Če ni na razpolago tri vhodnega elementa, moramo funkcijo faktorizirati.

Takšna realizacija sicer ni dobra, se pa v praksi podobne izvedbe, ki so potem viri dinamičnega hazarda rade pojavljajo.

Zato si je vredno primer podrobno analizirati !

$$f = \overline{\left\{\left[\overline{\overline{(x_1\overline{x}_4)}x_2}\right]\left[\overline{(x_1\overline{x}_4)}\right]\left[\overline{(\overline{x}_2\overline{x}_4)}\right]\right\}\left\{\left[\overline{\overline{(\overline{x}_3\overline{x}_4)}x_1}\right]\right\}}$$

$$f = \left[\overline{\overline{(x_1\overline{x}_4)}x_2}\right]\left[\overline{(x_1\overline{x}_4)}\right]\left[\overline{(\overline{x}_2\overline{x}_4)}\right] + \left[\overline{\overline{(\overline{x}_3\overline{x}_4)}x_1}\right]$$

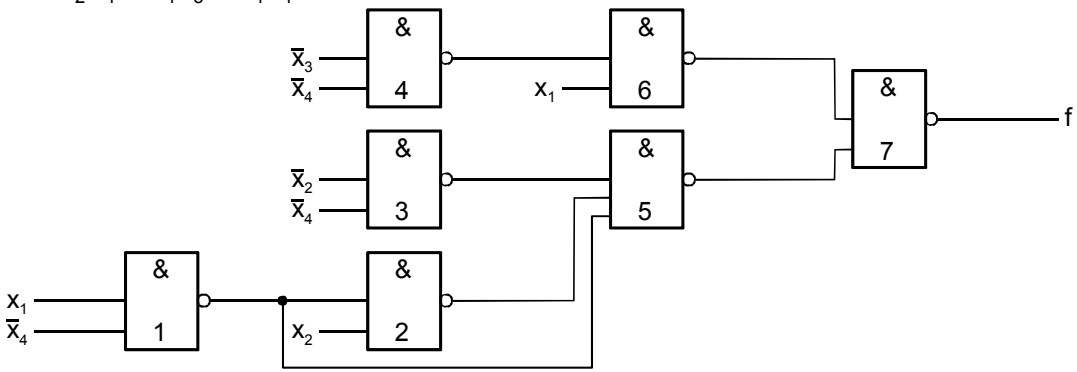
$$f = \left[(x_1\overline{x}_4) + \overline{x}_2\right]\ \left[\overline{x}_1 + x_4\right]\ \left[x_2 + x_4\right] + x_1x_4 + x_1x_3$$

$$f = (x_1\overline{x}_4x_4 + x_1\overline{x}_1x_4 + \overline{x}_2x_4 + \overline{x}_1\overline{x}_2)(x_2 + x_4) + x_1x_3 + x_1x_4$$

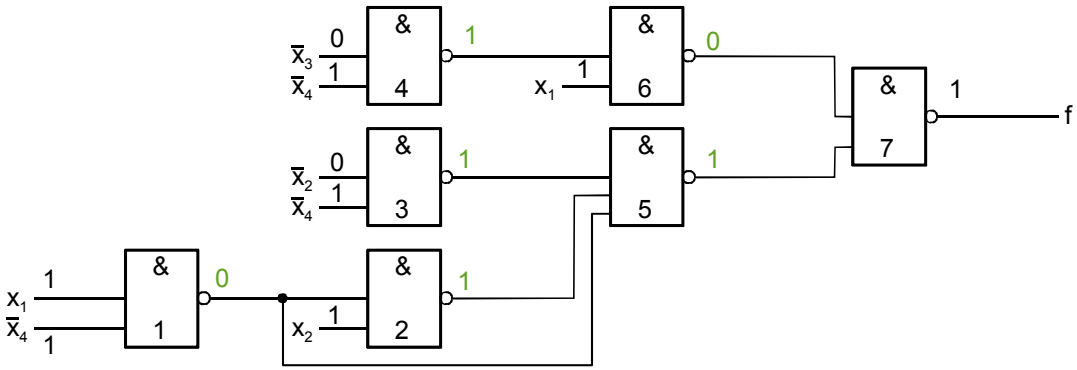
$$f = \overline{x}_2x_4 + x_2\overline{x}_2x_4 + \overline{x}_1\overline{x}_2x_4 + \overline{x}_1\overline{x}_2x_2 + x_1x_4 + x_1x_3$$

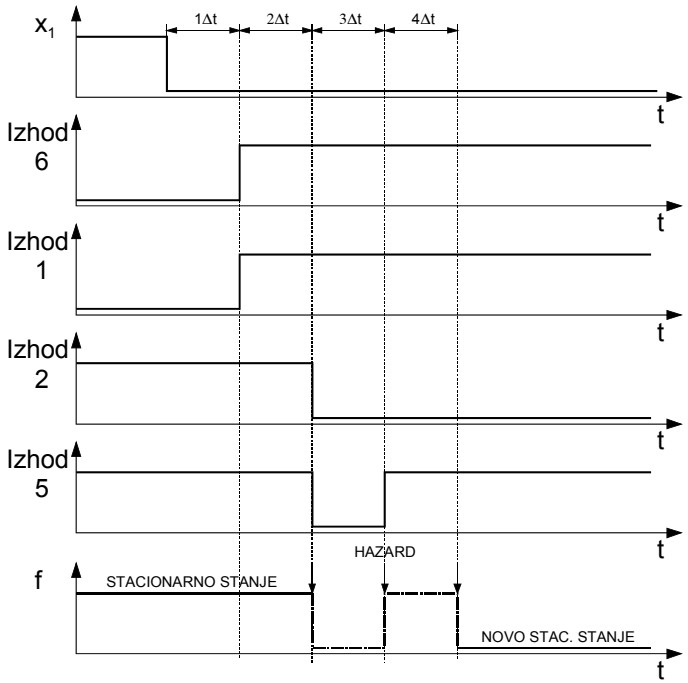
Če to še dalje poenostavimo pridemo do izhodiščne oblike:

$$f = \overline{x}_2x_4 + x_1x_3 + x_1x_4$$



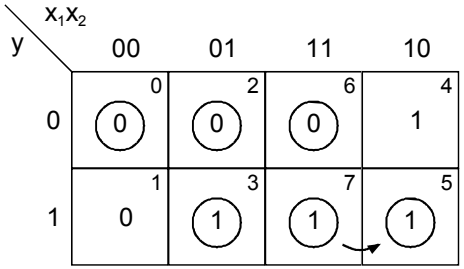
Predpostavimo stacionarno stanje: $x_1 = x_2 = x_3 = 1$ in $x_4 = 0$





9. 6. 2 Hazard v asinhronskih vezjih

$Y = x_1\bar{x}_2 + x_2y$



Δt – zakasnitev posameznega elementa

Začetno stanje:

$x_1 = 1$

$x_2 = 1$

$y = 1$

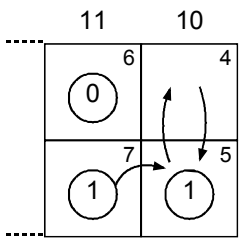
Predpostavimo: $t = 0^-$ x_1x_2y : 1,1,1 – minterm 7 v vzbujalni tabeli

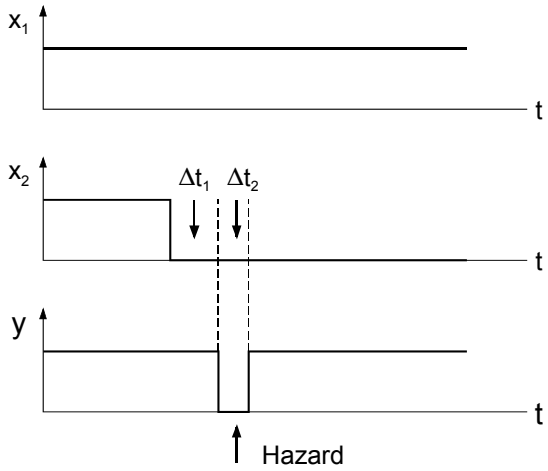
Zato : $Y = x_1\bar{x}_2 + x_2y = 1 \cdot 0 + 1 \cdot 1 = 1$

Potem: $t = 0$ – x_2 preide iz $1 \rightarrow 0$ – pomik na minterm 5

Potem: $t = \Delta t_1$ – $Y = 1 \cdot 0 + 0 \cdot 1 = 0$ – HAZARD - preskok na 4

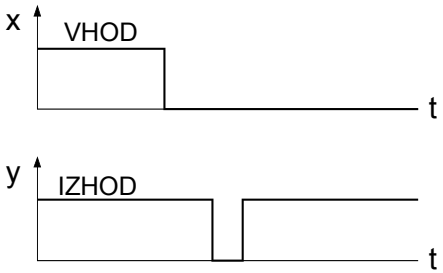
Potem: $t = \Delta t_1 + \Delta t_2$ – $Y = 1 \cdot 1 + 0 \cdot 1 = 1$



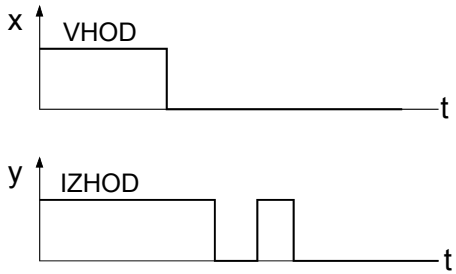


Statični hazard – enkraten trenuten pojav na izhodu, čeprav bi ta moral obdržati prvotno stanje glede na vhodno spremembo.

Dinamični hazard – večkraten prehod stanja izhoda, ki se odvije samo enkrat na vsako spremembo vhoda



a) Statični hazard



b) Dinamični hazard

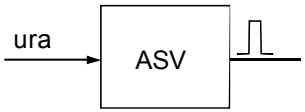
Statični in dinamični hazard je mogoče eliminirati z isto metodo dodajanja novih elementov; to je konjunkcij v disjunktivni obliki Booleove funkcije.

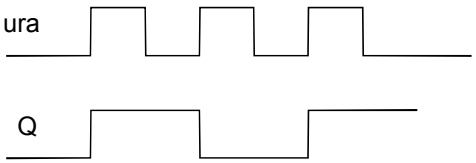
Bistven ali vgrajen hazard (Essential hazard)

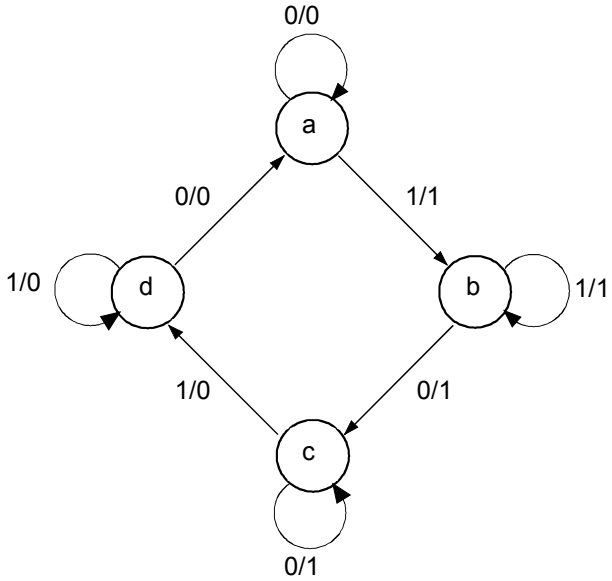
- Pojavlja se le v vezjih z dvema ali več povratnimi zvezami.
- Če obstoja je vezan na oboje; to je na zakasnitve in na vhodne specifikacije o delovanju vezja.

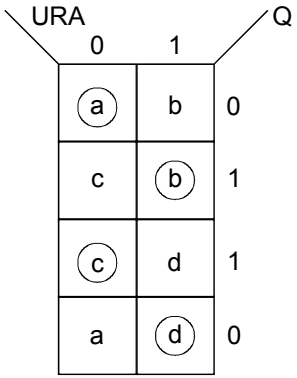
To pomeni, da bodo določene zahteve za delovanje sekvenčnega vezja že potencialno vsebovale možnost za pojav vgrajenega hazarda.

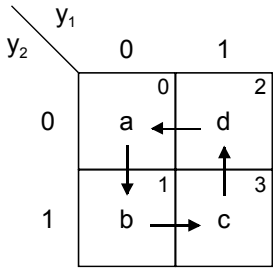
Tipična vezja, ki vsebujejo možnost vgrajenega hazarda so vezja z lastnostmi klecnega stikala (Toggle switch).

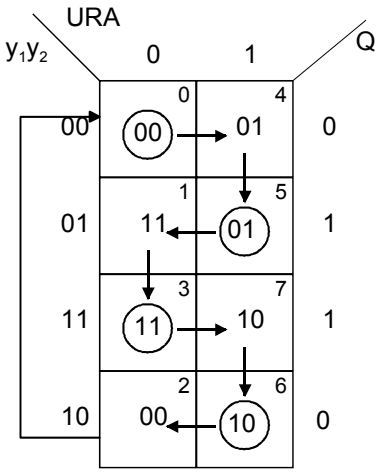


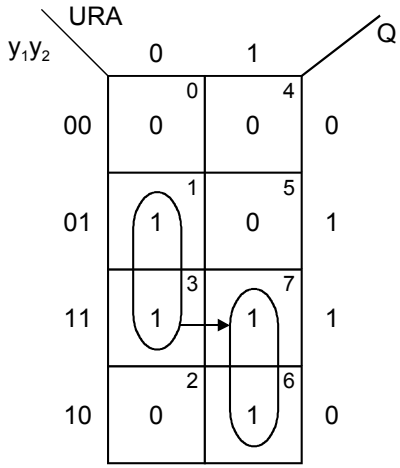




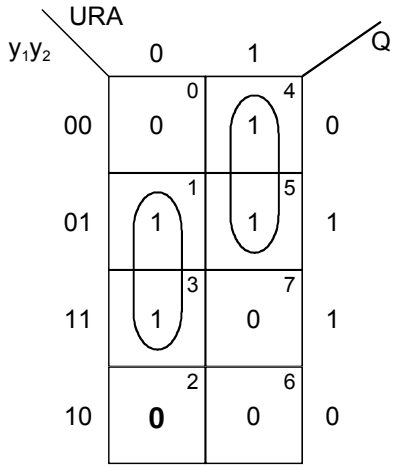








$$Y_1 = \overline{URA}y_2 + URAy_1$$

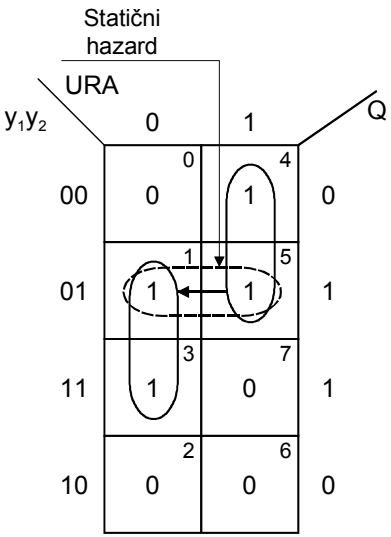


$$Y_2 = \overline{URA}y_2 + URA\bar{y}_1$$

Statični hazardni prehod nastopi pri prehodu iz b v c:

$$Y_2 = \overline{URA}y_2 + URA\bar{y}_1$$

URA = 0 – y_2 pade za trenutek na nič, potem pa se vrne na ena.



$$Y_2 = \overline{URA}y_2 + URA\bar{y}_1 + \bar{y}_1 y_2$$

Analiza vgrajenega hazarda pokaže, da pri enakih zakasnitvah elementov:

$\Delta t_1 = \Delta t_2 = \Delta t_3 = ... = \Delta t_7 = \Delta t$ vezje deluje brez napake.

Bistven ali vgrajeni hazard se pokaže šele, če Δt_1 povečamo na $7\Delta t$. Takrat pride do tega, da se vezje pri prehodu: a→b→b→d→d sploh ne ustavi v stanju b**,** ki je stabilno stanje vezja, temveč zaradi velike zakasnitve y_1 nadaljuje svoj cikel.