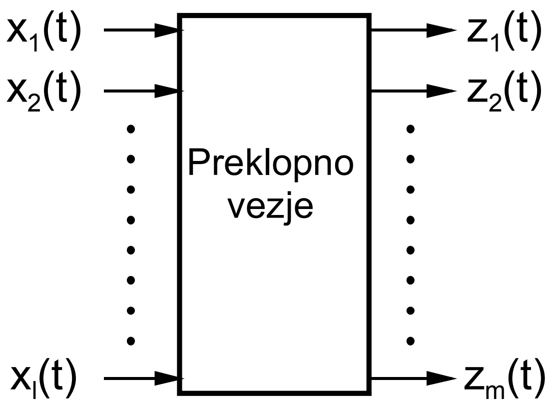
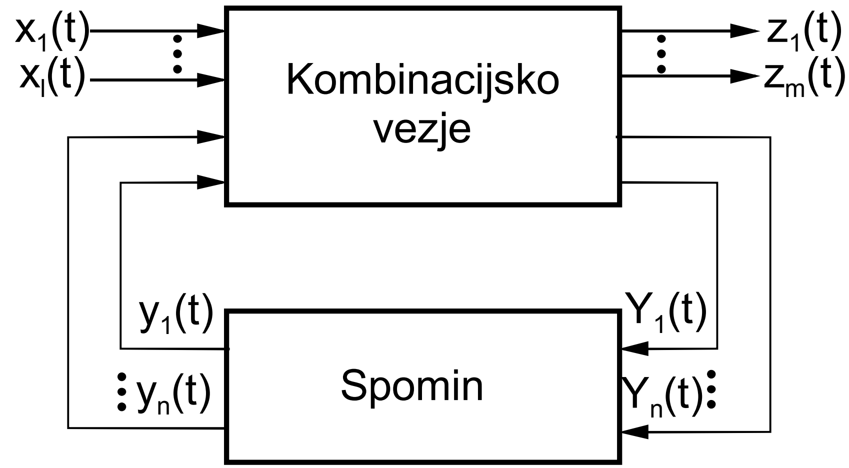
KONČNI AVTOMATI

Osnovne karakteristike



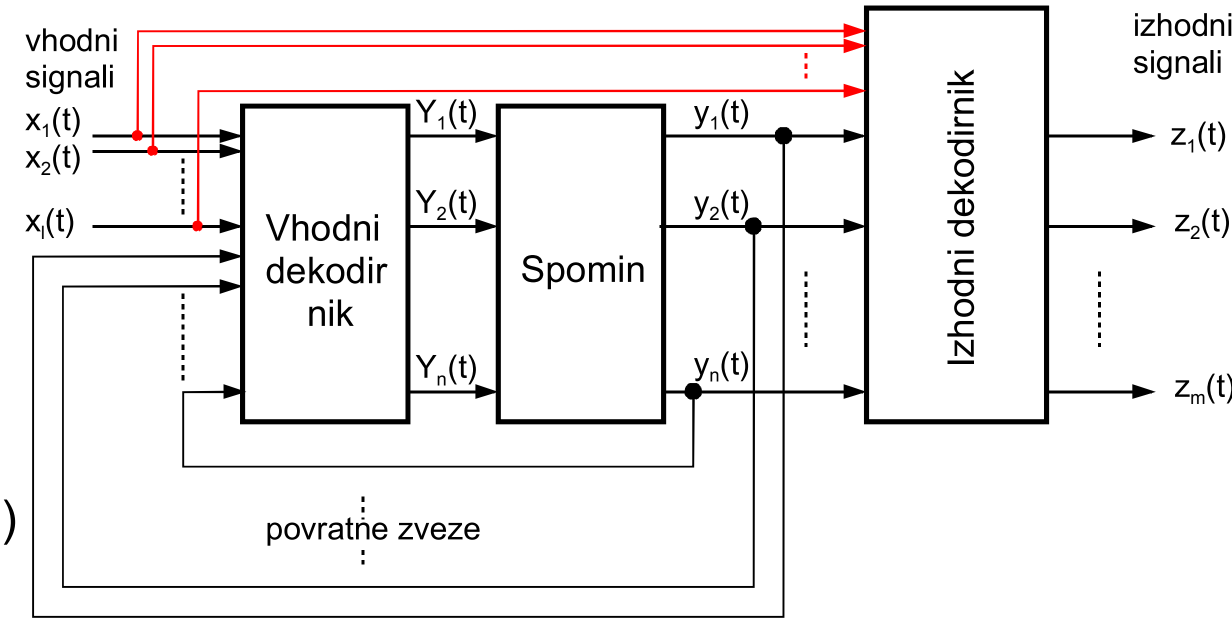


$$z_1(t) = f_1(x_1(t), x_2(t), \ldots, x_l(t))$$

$$z_2(t) = f_2(x_1(t), x_2(t), \ldots, x_l(t))$$

....

$$z_m(t) = f_m(x_1(t), x_2(t), \ldots, x_l(t))$$



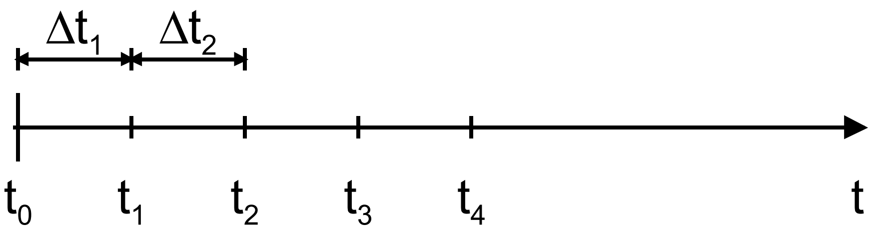
$$z_j(t) = f_1(x_1(t), x_2(t), \ldots, x_l(t), y_1(t), y_2(t), \ldots, y_n(t)),\ j = 1, 2, \ldots, m$$

$$Y_k(t) = f_k(x_1(t), x_2(t), \ldots, x_l(t), y_1(t), y_2(t), \ldots, y_n(t)),\ k = 1, 2, \ldots, n$$

Neodvisna časovna preklopna spremenljivka

Vpeljava časa v preklopne spremenljivke

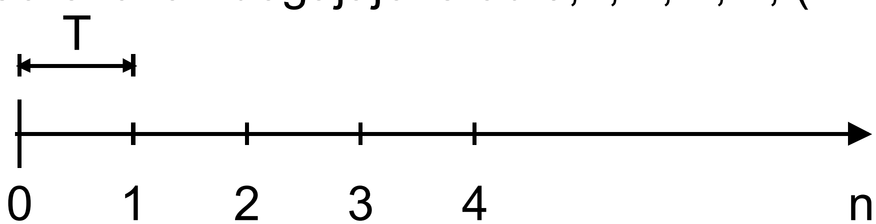
Ekvidistančni časovni interval:



$\Delta t_1 = \Delta t_2 = \cdots = \Delta t = T$

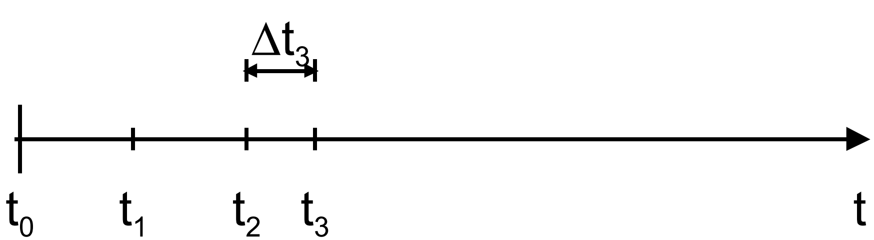
Čas razsekamo na intervale 0, Δt, $2\Delta t$ ... $n\Delta t$, $(n + 1)\Delta t$ ... $\Delta t = T$

Zaradi poenostavitve opustimo Δt in tako smemo reči, da se spremembe - vsaj teoretično - dogajajo le ob 0,1, 2,..., n, (n + 1).



$t = n\Delta t = nT$ nam bo tako pomenil sedanji čas

Neekvidistantna delitev časa:



Zapis neodvisne časovne spremenljivke

a) s funkcijsko odvisnostjo:

$x(t) = x(n\Delta t) = x(nT) \equiv x(n) \equiv x[n]$

b) z operatorjem:

$$\Delta^n x = \begin{cases} x, & \text{če je} \quad t \in (n\Delta t) \\ 0, & \text{če} \quad\quad t \notin (n\Delta t) \end{cases}$$

$\Delta^n x$ - neodvisna časovna spremenljivka

Časovni operator dodamo k FPSPF, pri tem pa potrebujemo še:

1. Izhodišče opazovanja; to je sedanji čas:

$\Delta^0 x = x$ - izhodišče opazovanja

2. Časovni premik in sočasnost:

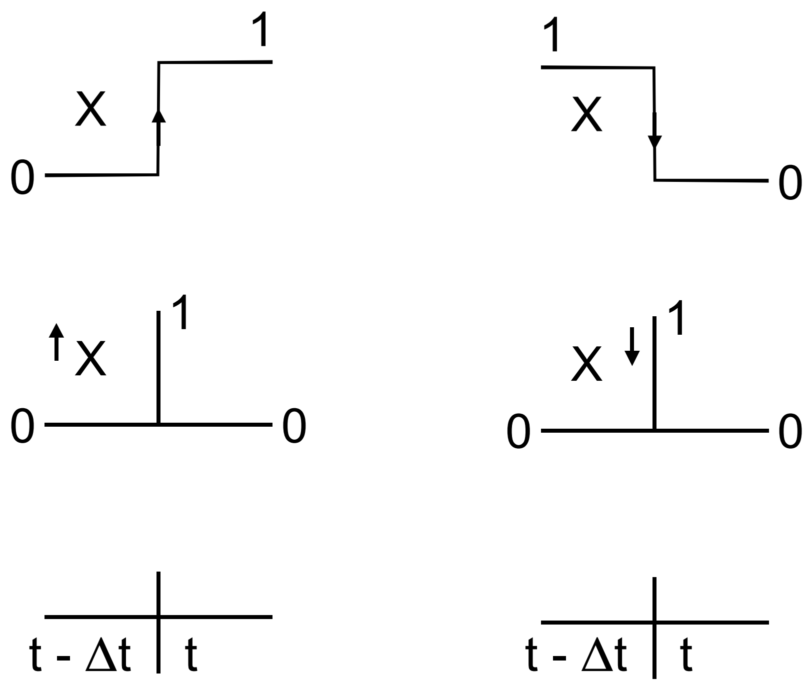
$\Delta^n(\Delta^m x) = \Delta^{n+m} x$ pomik v desno po časovni osi (prihodnji čas)

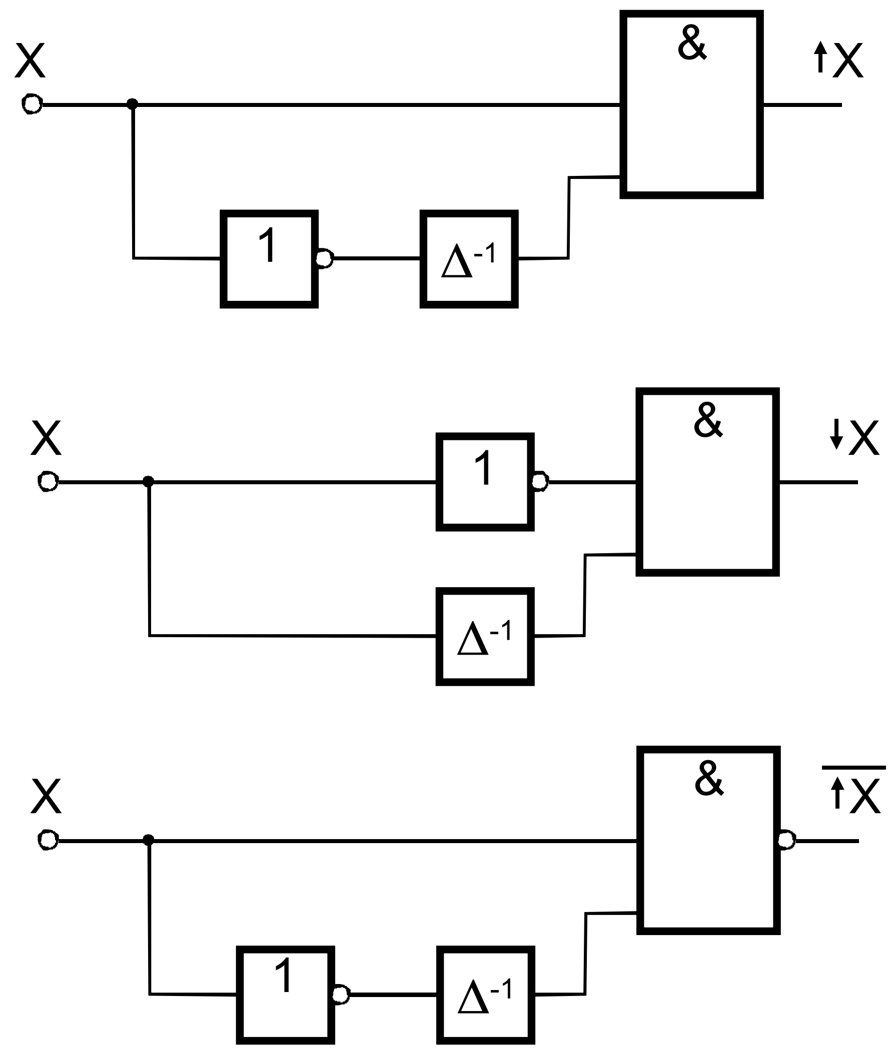
$\Delta^n(x_1 x_2) = \Delta^n x_1 \Delta^n x_2$ sočasnost nastopa spremenljivk

3. Komplement časovne spremenljivke:

$\Delta^{\bar{n}} x = \overline{\Delta}^n x = \Delta^n \overline{x}$ komplement časovne spremenljivke

Prehod med statičnimi signali in funkcija fronte





$\uparrow x = (\Delta^{-1}\overline{x})x$

$x\downarrow = (\Delta^{-1}x)\overline{x}$

$\uparrow x = (\overline{x})\downarrow$

$x\downarrow = \uparrow(\overline{x})$

Časovna preklopna funkcija

Odvodi osnovnih operacij

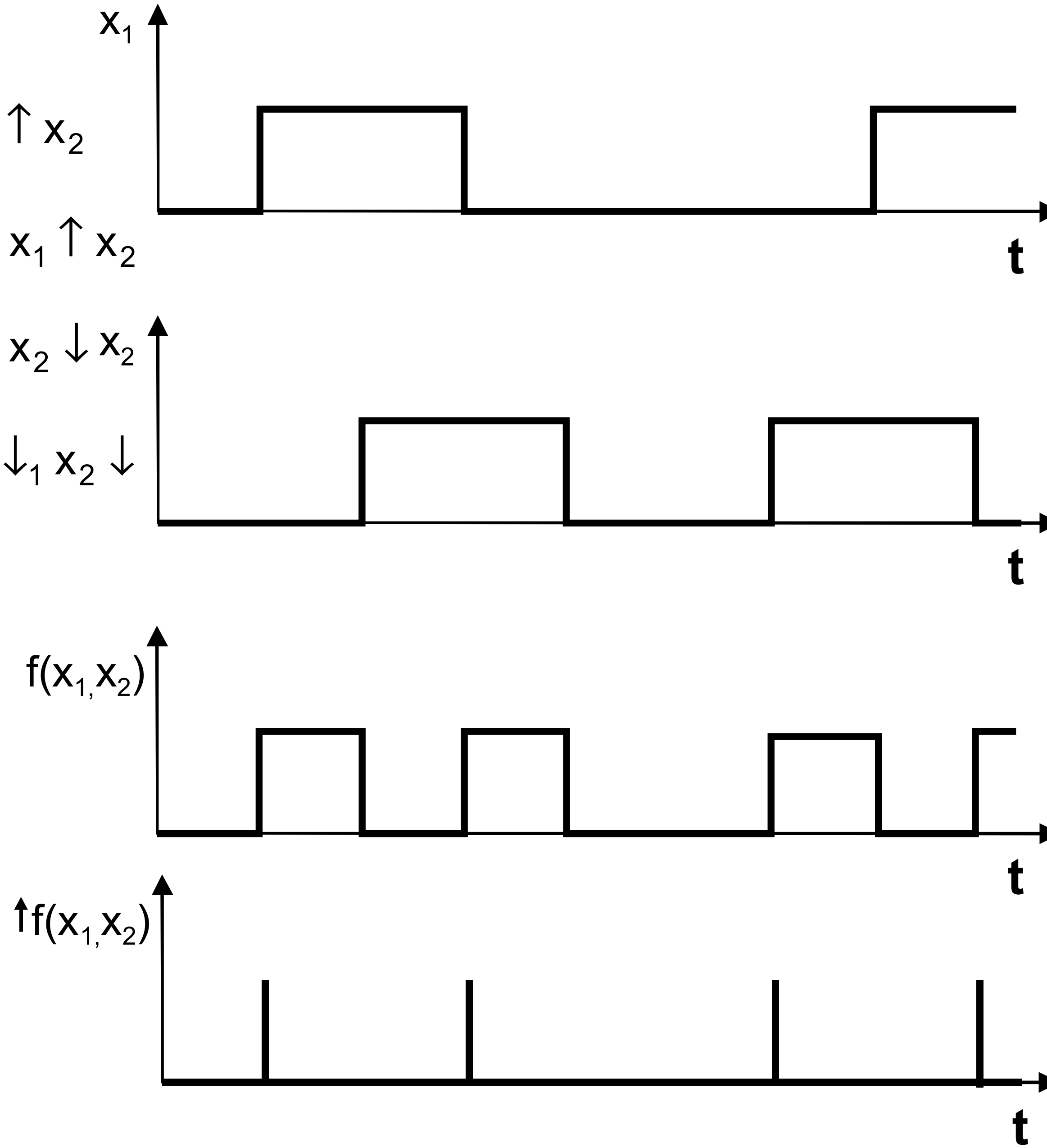
$$\uparrow(x_1x_2) = \uparrow x_1x_2 + x_1\uparrow x_2 + \uparrow x_1\uparrow x_2$$

$$\uparrow(x_1 + x_2) = \uparrow x_1\overline{x}_2 + \overline{x}_1\uparrow x_2 + \uparrow x_1\uparrow x_2$$

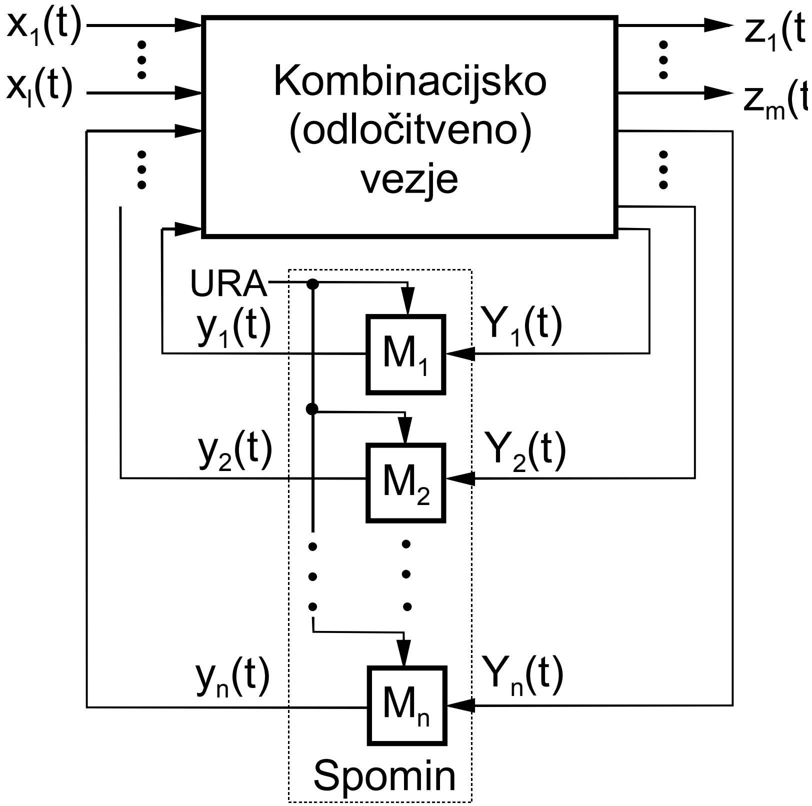
$$(x_1x_2)\downarrow = x_1\downarrow x_2 + x_1x_2\downarrow + x\downarrow_1 x_2\downarrow x_2$$

$$(x_1 + x_2)\downarrow = x_1\downarrow\overline{x}_2 + \overline{x}_1x_2\downarrow + x\downarrow_1 x_2\downarrow$$

$$\uparrow(\overline{x}) = x\downarrow \quad (\overline{x})\downarrow = \uparrow x$$



SINHRONSKI KONČNI AVTOMATI

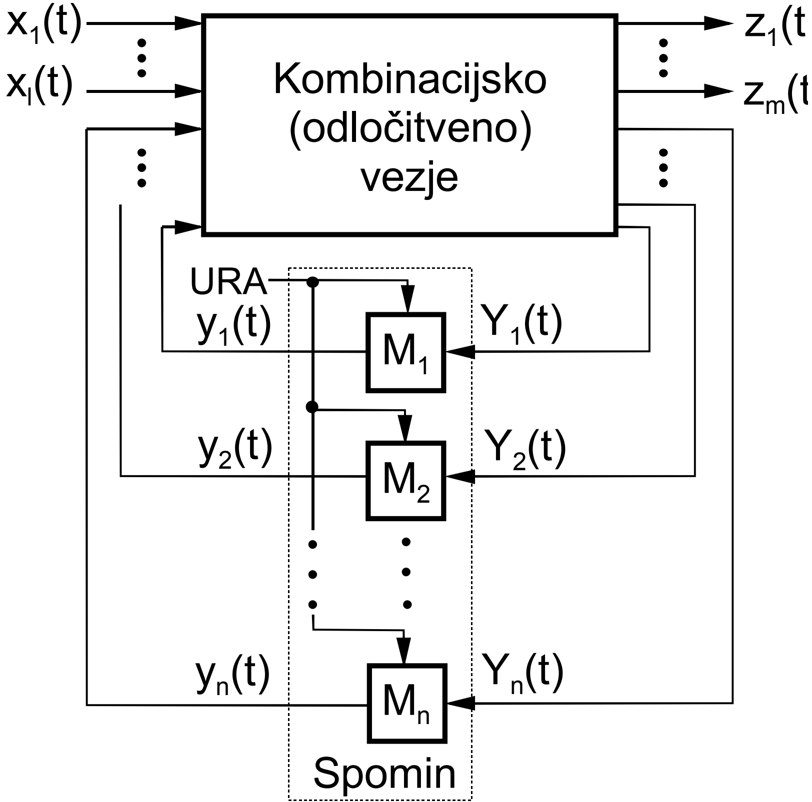


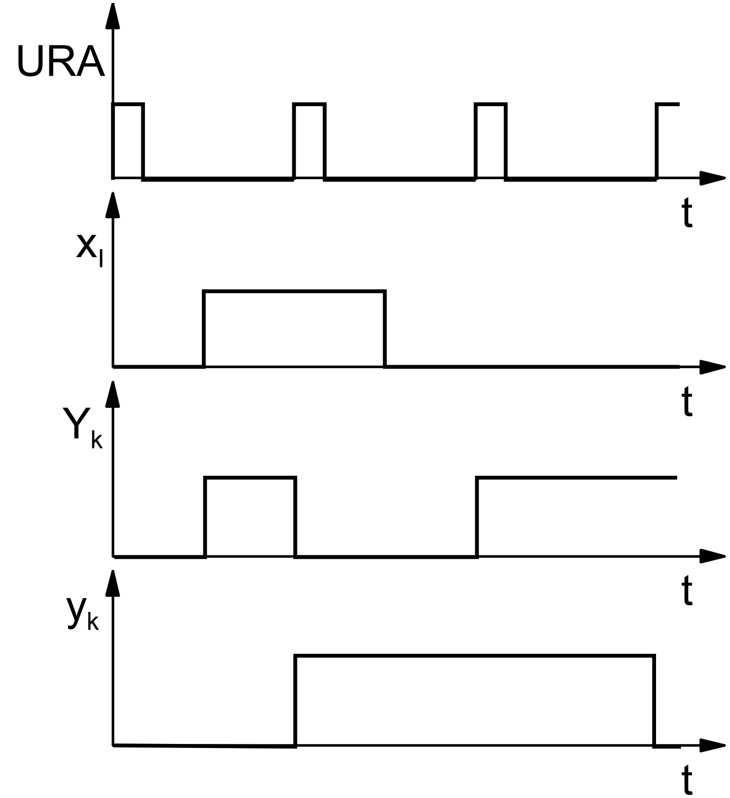
$x_1, x_2, \ldots x_l$ - vhodne spremenljivke

$y_1, y_2, \ldots, y_n$ - spremenljivke stanja (sekundarne vhodne spremenljivke)

$Y_1, Y_2, \ldots, Y_n$ - spremenljivke naslednjega stanja (vzbujalne spremenljivke)

$z_1, z_2, \ldots, z_m$ - izhodne spremenljivke

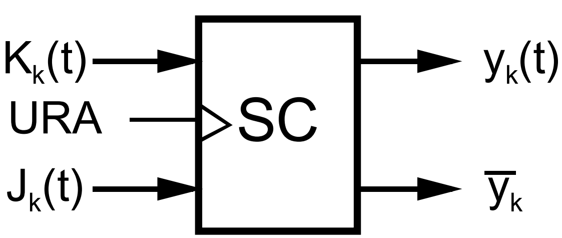




Vrste spomina.

$$J_k(t) = f_{J_k}(x_1(t), x_2(t), \ldots, x_l(t), y_1(t), y_2(t), \ldots, y_n(t));$$

$$K_k(t) = f_{K_k}(x_1(t), x_2(t), \ldots, x_l(t), y_1(t), y_2(t), \ldots, y_n(t)); \quad j=1,2,\ldots,n$$



$$\Delta^1 y_k = \bar{K}_k y_k + J_k \bar{y}_k$$

$$y_k(t + \Delta t) = \bar{K}_k(t) y_k(t) + J_k(t) \bar{y}_k(t)$$

$$\Delta^1 y_k = Y_k$$

$$y_k(t + \Delta t) = Y_k(t)$$

Definicija vhodov, stanj in izhodov

$\mathbf{x} = (x_1, x_2, \ldots, x_i, \ldots, x_L)$;

$x_i = 0,1$

$p = 2^l$ – število možnih vhodnih črk ali stanj

$\boldsymbol{X} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p})$; - vhodna abeceda avtomata

$\mathbf{z} = (z_1, z_2, \ldots, z_j, \ldots, z_m)$

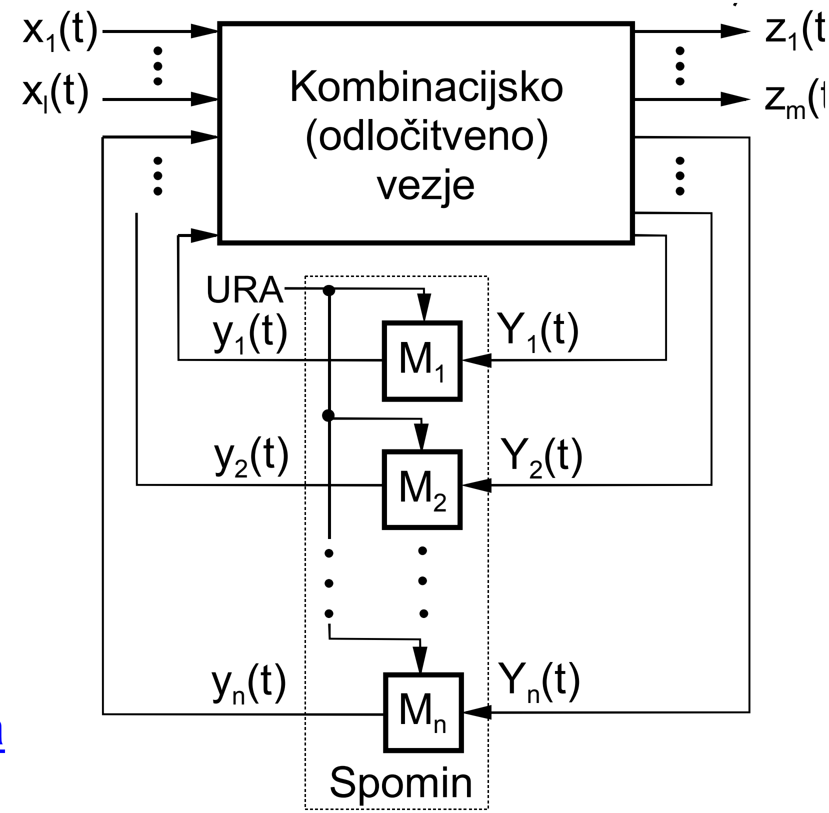
$q = 2^m$ – število možnih izhodnih črk ali stanj

$\boldsymbol{Z} = (\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_q})$; - izhodna abeceda avtomata

$\mathbf{s} \equiv \mathbf{y} = (y_1, y_2, \ldots, y_k, \ldots, y_n)$

$r = 2^n$ število možnih notranjih črk ali stanj

$\boldsymbol{S} = (\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_r})$; - notranja abeceda avtomata



Deterministični končni avtomat

$$\Delta^1 \mathbf{s} = \delta[\mathbf{s}, \mathbf{x}]; \quad \mathbf{z} = \lambda[\mathbf{s}, \mathbf{x}]$$

Definicija:

$\boldsymbol{A} = \langle \boldsymbol{X}, \boldsymbol{S}, \boldsymbol{Z}, \delta, \lambda \rangle$

δ, λ

δ: $\boldsymbol{X} \times \boldsymbol{S} \rightarrow \boldsymbol{S}$

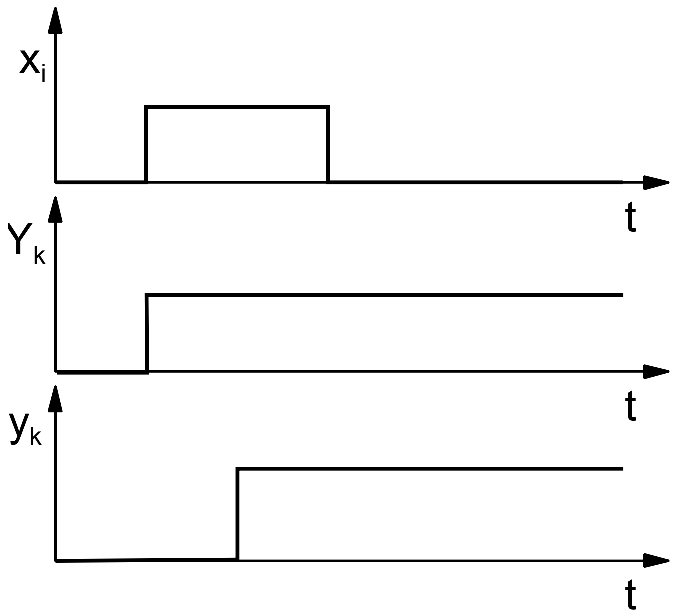
λ: $\boldsymbol{X} \times \boldsymbol{S} \rightarrow \boldsymbol{Z}$ pri Mealyjevem avtomatu

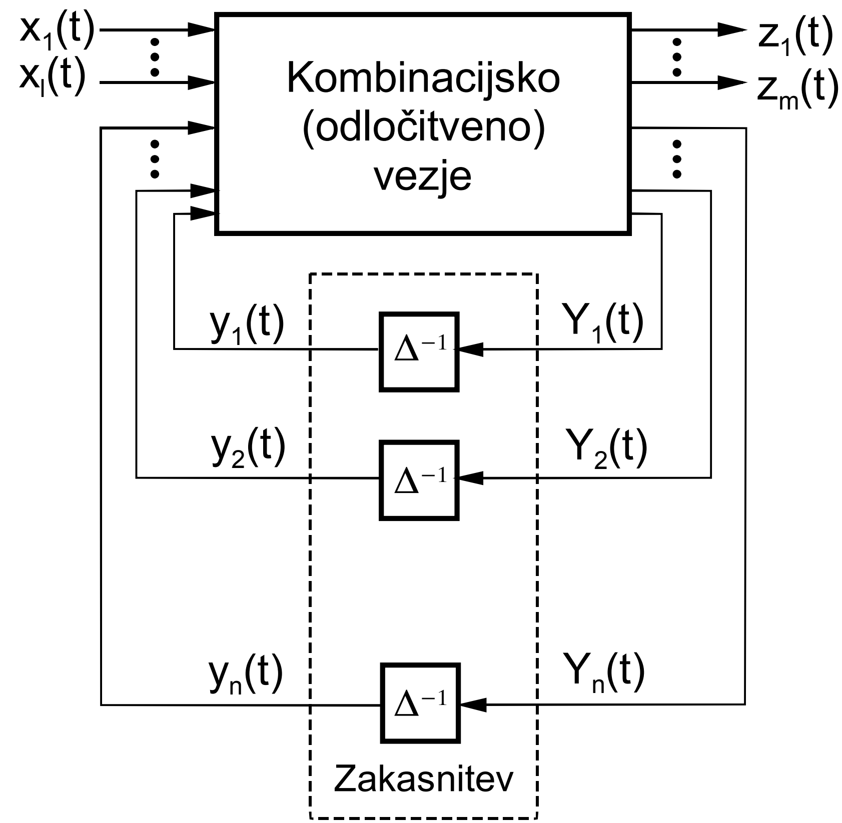
λ: $\boldsymbol{S} \rightarrow \boldsymbol{Z}$ pri Mooreovem avtomatu

$\boldsymbol{X} \times \boldsymbol{S}$ - kartezični produkt $(\mathbf{x_i} \mathbf{s_j})$

δ: $(\mathbf{x_i}, \mathbf{s_j}) \rightarrow \mathbf{s_k} \in \boldsymbol{S}$

ASINHRONSKI AVTOMATI





$\mathbf{x}= (x_1(t), x_2(t),\ldots,x_i(t),\ldots,x_l(t))$ – vhodno stanje avtomata

$\mathbf{y} = (y_1(t), y_2(t),\ldots y_k(t),\ldots, y_n(t))$ – notranje stanje avtomata

$\mathbf{z}= (z_1(t), z_2(t),\ldots, z_j(t),\ldots,z_m(t))$ – izhodno stanje avtomata

$y_k(t+\Delta t) = Y_k(t); j = 1, 2, \ldots, n$

Jezikovni opis avtomata

Standardna jezika:

- tabela prehajanja stanj

- diagram prehajanja stanj

Tabela prehajanja stanj: "p" stolpcev - po enega za vsako vhodno stanje in "r" vrstic za "r" mogočih sedanjih stanj avtomata

Pari (x_i, s_k) - specificirajo izhod oziroma naslednje stanje, v katerega bo prešel avtomat.

$l = \log_2 p$

$n = \log_2 r$

Diagram prehajanja stanj

$G = (V, E)$

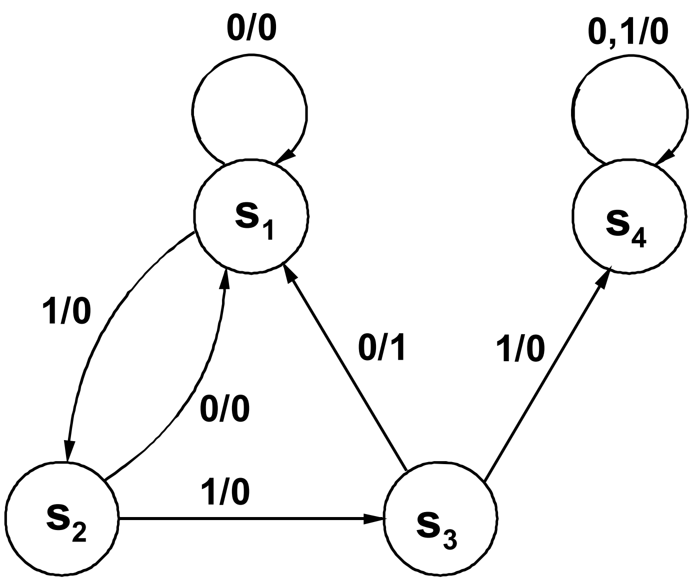
$v_i \in V$; $i = 1, 2, \ldots, r$ - število vozlišč

Iz vsakega vozlišča izhaja "p" vej

$e_i \in E$; $i = 1, 2, \ldots, p$ - število možnih vej

Tabela prehajanja stanj:

S	$\Delta^1 S, Z$	
	$x_1 = 0$	$x_2 = 1$
s_1	s_1, 0	s_2, 0
s_2	s_1, 0	s_3, 0
s_3	s_1, 1	s_4, 0
s_4	s_4, 0	s_4, 0



Abstraktna sinteza končnega avtomata

Mealy-jev avtomat

$A_{ME} = \langle \boldsymbol{X}, \boldsymbol{S}, \boldsymbol{Z}, \delta, \lambda \rangle$

$\delta: \boldsymbol{S} \times \boldsymbol{X} \rightarrow \boldsymbol{S}$

$\lambda: \boldsymbol{S} \times \boldsymbol{X} \rightarrow \boldsymbol{Z}$

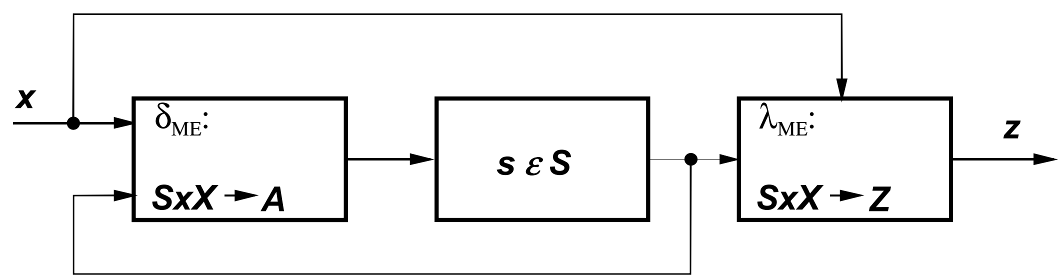
Enačba stanj in izhodna enačba:

$\Delta^1 \mathbf{s} = \delta(\mathbf{s}, \mathbf{x})$

$\mathbf{z} = \lambda(\mathbf{s}, \mathbf{x})$

Specificirana in nespecificirana stanja

Blokovni diagram Mealyjevega avtomata



Končno ali ponorno stanje končnega avtomata

Začetno ali izvorno stanje končnega avtomata

Povezanost avtomata: če za vsak par ($\mathbf{s}_i$, $\mathbf{s}_j$) obstoja vhodna sekvenca, ki prevede $\mathbf{s}_i \rightarrow \mathbf{s}_j$ je avtomat strogo povezan.

Mooreov končni avtomat

$\boldsymbol{A_{MO}} = \langle \boldsymbol{X}, \boldsymbol{S}, \boldsymbol{Z}, \delta, \lambda \rangle$

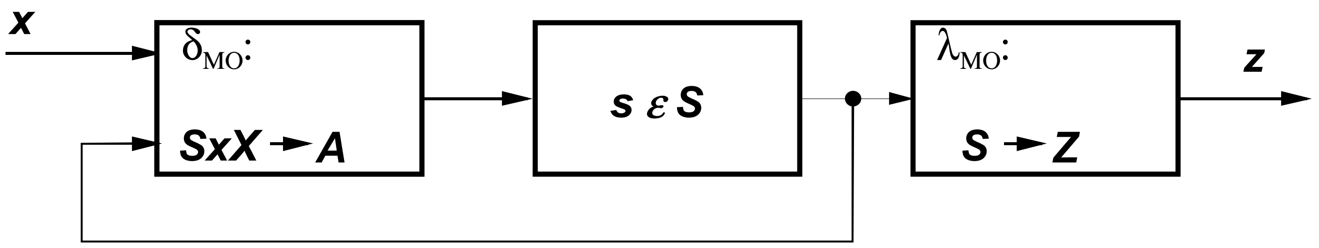
$\delta: \boldsymbol{S} \times \boldsymbol{X} \rightarrow \boldsymbol{S}$

$\lambda: \boldsymbol{S} \rightarrow \boldsymbol{Z}$

Enačba stanj in izhodna enačba:

$\Delta^1 \boldsymbol{s} = \delta(\boldsymbol{s}, \boldsymbol{x})$

$\boldsymbol{z} = \lambda(\boldsymbol{s})$



Uvod v strukturno sintezo

Na nivoju abstraktne sinteze s standardnimi jeziki enolično preslikavamo množico vhodnih stanj v množico izhodnih stanj.

Algoritem sinteze temelji na sledečem principu:

Vsa mogoča zaporedja ali sekvence vhodnih stanj razdelimo na podmnožice, katerih elementi s svojim nastopom povzročijo identično izhodno stanje avtomata.

Vhodne besede preslikavamo v izhodne črke oziroma sekvence vhodnih simbolov v izhodni simbol, ki se časovno ujema z zadnjim simbolom vhodne sekvence.

S	Δ^1**s**	**z**
	x=0	**x**=1
s_1	s_1,0	s_2,0
s_2	s_1,0	s_3,0
s_3	s_1,1	s_4,0
s_4	s_4,0	s_4,0

Pri začetnem stanju $\mathbf{s_1}$ sekvence 00110, 10110 itd. pripeljejo izhod v stanje 1.

Podmnožice vhodnih besed, ki so v avtomatu označene z isto izhodno črko imenujemo dogodek.

VHODNI DOGODEK	IZHODNA ČRKA
$X_1(x_1 x_2,\ldots,x_p)$	z_1
$X_2(x_1 x_2,\ldots,x_p)$	z_2
.	.
.	.
.	.
.	.
.	
$X_q(x_1 x_2,\ldots,x_p)$	z_q
$X_{q+1}(x_1 x_2,\ldots,x_p)$	-

Pri nepopolno specificiranih avtomatih posebno prikazujemo dogodek: $X_{q+1}(x_1 x_2,\ldots,x_p)$, ki je sestavljen iz vseh prepovedanih vhodnih besed.

Ker so besede tega dogodka prepovedane in se torej nikoli ne bodo pojavile na vhodu avtomata ostaja izhod neopredeljen.

Poudariti pa je potrebno, da je časovnemu sosledju vhodnih dogodkov pridruženo časovno sosledje izhodnih dogodkov.

S sintezo ustvarjamo množico pravil s pomočjo katerih avtomat opravi zahtevano enolično preslikavo.

Sinteza avtomata poteka v več zaporednih korakih.

1. Na osnovi besednega opisa algoritma delovanja razvijemo tabelo stanj oziroma diagram prehajanja stanj avtomata.

2. Opravimo minimizacijo stanj avtomata

3. Kodiramo stanja in izberemo tip spominskih elementov

4. Določimo kodirano tabelo prehajanja stanj in izhodno tabelo

5. Določimo vzbujalne in izhodne funkcije

6. Izdelamo simbolični načrt avtomata

7. Izberemo izvedbo kombinacijskega dela vezja in zrišemo vezalni načrt

Pri točki 3 je potrebno omeniti, da obstajajo določena priporočila glede uporabnosti spominskih celic, kot na primer:

- JK, splošna uporaba

- RS in D, kjer moramo prenašati oziroma premikati podatke (pomični registri, binarni števci).

Postopek sinteze bomo prikazali na primeru števca po modulu “m”.

S	$\Delta^1 S$
s_0	s_1
s_1	s_2
s_2	s_3
s_3	s_4
s_4	s_5
s_5	s_6
s_6	s_0

n	Q_1	Q_2	Q_3	$\Delta^1 Q_1$	$\Delta^1 Q_2$	$\Delta^1 Q_3$
0	0	0	0	1	0	0
1	1	0	0	0	1	0
2	0	1	0	1	1	0
3	1	1	0	0	0	1
4	0	0	1	1	0	1
5	1	0	1	1	1	1
6	1	1	1	0	0	0
	0	1	1	x	x	x

Karakteristična enačba izbrane spominske celice:

$J = g_2 \qquad K=\bar{g}_1$

Enačbe stanj:

$$\Delta^1 Q_1 = Q_1\bar{Q}_2 Q_3 + \bar{Q}_1(\bar{Q}_2\bar{Q}_3 + Q_2\bar{Q}_3 + \bar{Q}_2 Q_3)$$

$$\Delta^1 Q_2 = \bar{Q}_1 Q_2\bar{Q}_3 + \bar{Q}_2(Q_1\bar{Q}_3 + Q_1 Q_3)$$

$$\Delta^1 Q_3 = Q_1 Q_2\bar{Q}_3 + \bar{Q}_1\bar{Q}_2 Q_3 + Q_1\bar{Q}_2 Q_3$$