Sekvenčna vezja

- Trenutno stanje
- Naslednje stanje } sklenjeno vezje - povratna zanka (pomnilni člen)

Asinhronska pomnilna celica (zapah - latch)

RS zapah

R	S	Q	Q^+
0	0	Q	Q
0	1	0	1
1	0	1	0
1	1	X	X

v enačbi za naslednje stanje

karnaugh

Q^+	R=00	01	11	10
S=0	Q	0	X	1
1	1	0	X	0

$Q^+ = S \vee \overline{R} Q$

RSNOR vezje

R	S	Q^+
0	0	Q
0	1	1
1	0	0
1	1	0

Q^+	R=00	01	11	10
S=0	Q	0	0	1
1	1	0	0	1

$Q^+ = S \vee \overline{R} Q$

RSNAND vezje

R	S	Q^+
0	0	X
0	1	1
1	0	0
1	1	Q

Q^+	R=00	01	11	10
S=0	X	1	0	X
1	X	1	0	Q

$Q^+ = \overline{S} \vee R Q$

časovni diagram

→ upoštevanje vezja skozi čas

Sinhronska sekvenčna vezja

→ sinhronizacija + urinska zanka

→ prekine vpliv povratne zanke

→ pravilo: ure = vpr. ena v RS

→ sinhrona ure = $\overline{clk_1} \vee \overline{clk_2} = clk$

→ delay zapah (delay: clk) $\frac{t_{clk}}{2} = \Delta$ delta zakasnitev pripomni

Sinhronizacija na nivo ure (level triggered)

- to pomeni na nivo ure (1 ali 0)
- pravilo za vstop (za prehod na naslednje stanje (vpr. L)

ure	vpr.	Q^+
0	0	Q
0	1	Q
1	0	Q
1	1	X
1	0	0
1	1	1

$D'_Q = d$

→ podatek d_i se prepiše v pomnilni člen

→ podatek d_i se ne prepiše v pomnilni člen

→ podatek d_i se shrani v naslednjem stanju

→ vse povratne zanke

(ravno ure (clock edge)

- pozitivno fronto (spremenijo se iz 0 v 1
- negativno fronto (spremenijo se iz 1 v 0
- Sinhronizacija na fronto ure = vse povratne + vstop vklopa
- latch neobstoja - interval ure je enako + vstop vklopa

- D'_Q: vsaj prvotno vklopa + naslednjega vpisano kodo

Sinhronizacija na fronto (edge - triggered flip-flop)

RS zapah preko ure na nivo

spremeni stanje le pri 0 v 1

spremeni stanje le pri 1 v 0

ES celica preko ure fronte

Master-Slave RS flip-flop (celica z povratnogo)

RS preko ure pozitivne fronte ure

OSTALE POMNILNE

JK pomnilna celica ~ RS

clock	D'_Q
0	Q
0	Q
1	0
1	1

→ povratnogo na $\overline{clk}_2$ ure

q	D'_Q	k
0	0	0
0	1	1
1	0	↓
1	1	↑

$D'_Q = q\overline{k} \vee \overline{q} j$

D pomnilna celica

$D'_Q = d$ → delayni člen

d	D'_Q
0	0
1	1

q	D'_Q	d
0	0	0
0	1	1
1	0	0
1	1	1

T pomnilna celica

→ trigger

t	D'_Q
0	Q
1	$\overline{Q}$

q	D'_Q	t
0	0	0
0	1	1
1	0	1
1	1	0

$D'_Q = t \nabla Q$

Izvedba sekvenčnih vezij s pomnilnimi celicami

→ opis delovanja sekvenčnega vezja (enačbe, tabele, ...)

→ pomilne celice

Primer: S pomnilja JK pomilne celice načrtaj sekvenčno vezje ki deluje po enačbi:

$D'_Q = q(x \equiv y) \vee \overline{x} \overline{y}$

① Tabela odvisnosti naslednjega stanja (D'_Q) od trenutnega stanja (q) in vhodov (x,y):

x	y	q	D'_Q	
0	0	0	1	D'_Q=1 (postane 1)
0	0	1	1	
0	1	0	0	D'_Q=0 (ostane 0)
0	1	1	0	
1	0	0	1	D'_Q=$\overline{x}$ (invert)
1	0	1	0	
1	1	0	0	D'_Q=2 (hold)
1	1	1	1	

② Na podlagi podatkov $q \to D'_q$ in izdelamo tabelo določitve spodnjih vhodov j in k v pomnilni celici tabele:

x	y	q	D'_Q	j	k
0	0	0	1	1	↓
0	0	1	1	1	0
0	1	0	0	0	↑
0	1	1	0	↓	↑
1	0	0	1	1	↓
1	0	1	0	↓	1
1	1	0	0	0	↑
1	1	1	1	↓	0

Vzbujevalna tabela:

q	D'_Q	j	k
0	0	0	↑
0	1	1	↓
1	0	↓	1
1	1	↓	0

③ Iskanje funkcije za j in k