

RAZVOJ DIGITALNIH SISTEMOV

Izpit 23. 9. 2011

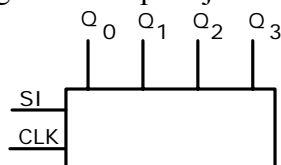
1. Z uporabo pravil Boole-ove logike zapišite logično funkcijo v KNO z NAND (Sheffer-jevimi) operatorji.

$$f(x_1, x_2, x_3) = (x_1 + x_2) \cdot (\overline{x_2} + \overline{x_3}) \cdot x_3$$

2. Narišite strukturo vezja PAL3L3 (izmišljeno vezje) in ga uporabite za realizacijo funkcij:

- $f_1 = x_1 \oplus x_2$
 - f_2 = konjunkcijo treh spremenljivk
 - f_3 = funkcijo treh spremenljivk, ki vrne 1 pri vsaj dveh enicah na vseh vhodih.
- Vsaka disjunkcija (OR) v PAL3L3 ima 4 konjunkcije (AND). Oznaka L pomeni, da je vezje tipa AND–NOR. Povezave označite s piko (●).

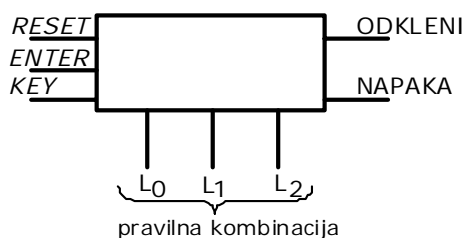
3. Sestavite 4-bitni pomikalni register s T-celicami in izbiralniki 2/1. Register ima zaporedni vhod SI (ang. serial input), in vzporedni izhod (Q_0 , Q_1 , Q_2 , Q_3). Uporabite poimenovanje signalov na spodnji sliki.



4. Narišite diagram stanj Moore-ov avtomat končnih stanj, ki deluje kot 3-bitna sekvenčna ključavnica. Ključavnica ima tipko za ponastavitev (*RESET*), ki postavlja ključavnico v začetno stanje, tipko (*ENTER*) za vnos nastavljenе kombinacije in preklopnik (*KEY*) za vnos enega bita kombinacije. Izhod ključavnice je (*ODKLENI*), ki postane '1' ko uporabnik vnesе pravilno kombinacijo in izhod (*NAPAKA*), ki postane '1' če je vnesena kombinacija napačna. Uporabo ključavnice povzema spodnje zaporedje:

- 1.) pritisnemo *RESET*
- 2.) s preklopnikom *KEY* nastavimo bit kombinacije odklepanja
- 3.) pritisnemo *ENTER*
- 4.) izvede se primerjava i-tega bita ($KEY \equiv L_i$)
- 5.) dvakrat ponovimo korake 2 – 4
- 6.) če je vnesena 3-bitna kombinacija pravilna, se postavi *ODKLENI*='1', sicer se postavi *NAPAKA*='1'.
- 7.) ponoven pritisk na *RESET* nas vrne na korak 1.

Pravilno kombinacijo, ki odklene ključavnico, nastavljamo z biti L_0 , L_1 in L_2 .



1. Z uporabo pravil Boole-ove logike moramo zapisati logično funkcijo v KNO z NAND (Sheffer-jevimi) operatorji:

Direktne poti za pretvorbo KNO v zapis s NAND (Sheffer-jevimi) operatorji ni, zato KNO najprej pretvorimo v DNO obliko, tako da izvedemo AND operacije nad dvočleniki:

$$f(x_1, x_2, x_3) = (x_1 + x_2) \cdot (\overline{x_2} + \overline{x_3}) \cdot x_3$$

V drugem členu bomo dobili člen $x \cdot x' = 0$ po lastnostih Boole-ove algebre:

$$f(x_1, x_2, x_3) = (x_1 \cdot \overline{x_2} + x_1 \cdot \overline{x_3} + x_2 \cdot \overline{x_3}) \cdot x_3$$

S preostalimi členi izpišemo AND operacije:

$$f(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot \overline{x_3} \cdot x_3 + x_2 \cdot \overline{x_3} \cdot x_3$$

Podobno pri drugem in tretjem členu funkcije dobimo $x \cdot x' = 0$:

$$f(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot x_3$$

Nastala funkcija ima samo en člen, ki ga moramo izraziti z NAND operatorji. Funkcijo negiramo in dobimo:

$$\overline{f(x_1, x_2, x_3)} = \overline{x_1 \cdot \overline{x_2} \cdot x_3}$$

Končno izrazimo še negacijo funkcije f :

$$\overline{x} = 1 \uparrow x$$

Nazadnje izrazimo original funkcije f , kot zahteva naloga:

$$f(x_1, x_2, x_3) = 1 \uparrow (x_1 \uparrow \overline{x_2} \uparrow x_3)$$

2. Za funkcije zapišemo najprej pravilnostno tabelo, nato narišemo Veitch–eve diagrame. Uporabite PAL3L3 (namišljen čip) za realizacijo naslednjih funkcij:

- $f_1 = x_1 \oplus x_2$
- $f_2 =$ konjunkcija treh spremenljivk
- $f_3 =$ funkcijo treh spremenljivk, ki vrne 1 pri vsaj dveh enicah na vseh vhodih.

Vezje ima 3 vhode in 3 izhode. Vsaka disjunkcija (OR) ima 4 produkte (AND). L pa pomeni, da je izhod negiran. Povezave oz. ‘varovalke’ označite s piko (•).

Za funkcije zapišemo najprej pravilnostno tabelo, nato narišemo Veitch–eve diagrame.

x_1	x_2	x_3	f_1	f_2	f_3
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	0	1	1

Vezje PAL ima negirane izhode, zato bomo pri realizaciji funkcij z Veitch–evimi diagrami realizirali \overline{f} in ne f .

$\overline{f_1}$:

	x_1		
x_2	0	0	1
	1	1	0
		x_3	

Prvo funkcijo zapišemo enostavno, saj je negacija XOR funkcije dveh spremenljivk kar funkcija ekvivalence:

$$\overline{f_1} = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$$

Podobno lahko naredimo za drugo funkcijo, kjer za negacijo konjunkcije treh spremenljivk uporabimo De Morgan–ovo enakost.

$\overline{f_2}$:

	x_1		
x_2	0	1	0
	0	0	0
		x_3	

$$f_2 = \overline{\overline{x_1 \cdot x_2 \cdot x_3}}$$

$$\overline{f_2} = \overline{x_1} + \overline{x_2} + \overline{x_3}$$

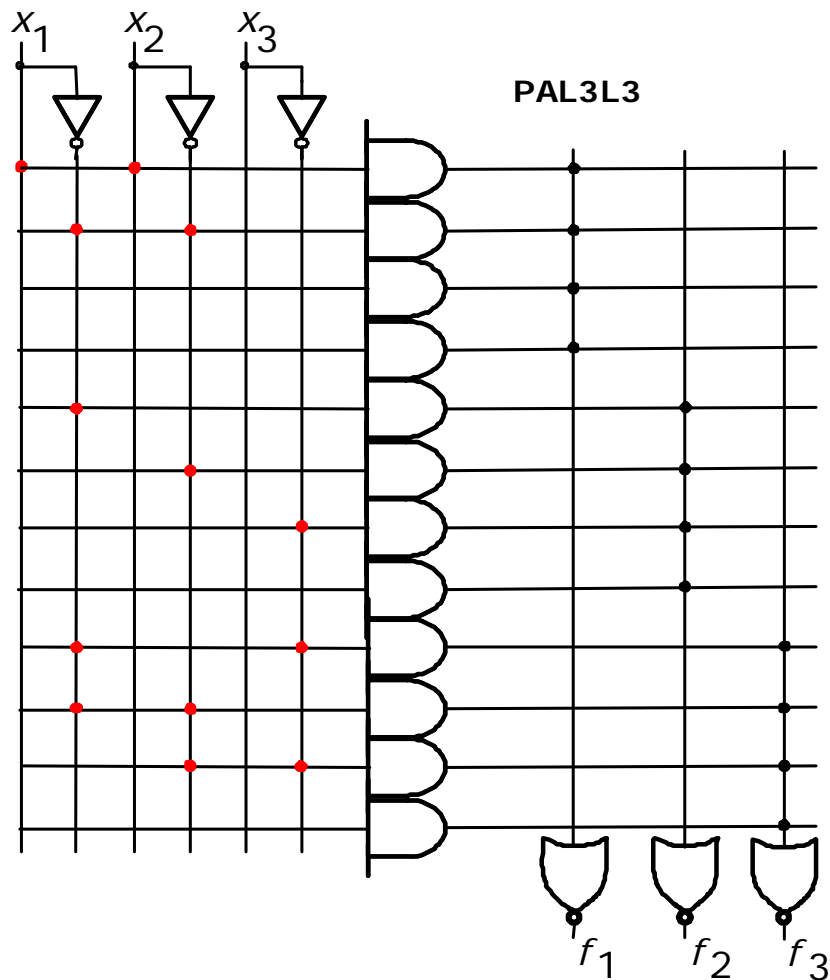
Zadnjo funkcijo minimiziramo z uporabo Veitch–evega diagrama, tako da zbiramo ničle.

$\overline{f_3}$:

	x_1		
x_2	1	1	1
	0	1	0
		x_3	

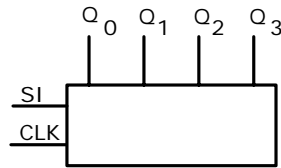
$$\overline{f_3} = \overline{x_1} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} + \overline{x_2} \cdot \overline{x_3}$$

Pri realizaciji PAL vezja upoštevamo poenostavljeno strukturo, pri kateri ne vežemo vsake povezave na konjunkcije, saj so AND vrata na narisani strukturi 6-vhodna. Vezje PAL3L3 je AND-NOR arhitekture in vsebuje 4 konjunkcije na en NOR člen. Pri PAL vezju je programabilen samo AND del vezja.

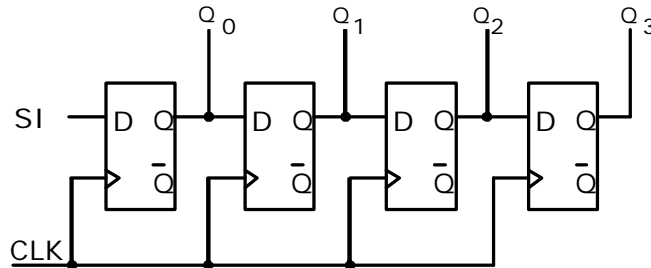


Predstavljeno PAL3L3 vezje je sicer izmišljeno, vendar demonstrira strukturo in uporabo večjih (realnih) PAL vezij kot so npr. PAL14L4, GAL16V8 in GAL22V10. GAL vezja so nadgradnja osnovne PAL strukture. Slednji so izključno kombinacijski, GAL vezja pa imajo v OLMC strukturi še D-FF, s katerim lahko realiziramo tudi sekvenčna vezja.

3. Sestavite 4-bitni pomikalni register s T-celicami in izbiralniki 2/1. Register ima zaporedni vhod SI (ang. serial input), in vzporedni izhod (Q_0, Q_1, Q_2, Q_3)



Zaporedno-vzporedni (SIPO) pomikalni register, realiziran s pomočjo D-FF, je veriga kaskadno vezanih D-FF, v kateri je izhod prejšnjega flip-flopa Q_{i-1} vezan na vhod naslednjega flip-flopa D_i .



Če želimo pomikalni register sestaviti iz T-FF in 2/1 izbiralnikov, moramo pravzaprav realizirati celico D-FF s pomočjo T-FF in 2/1 izbiralnikov. V ta namen zapišemo tabelo D-FF, pri kateri dodamo izhodni stolpec T vhoda.

D	$Q(t)$	$Q(t+1)$	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

XOR vrata moramo realizirati s pomočjo 2/1 izbiralnikov, zato zapišemo enačbo XOR funkcije:

$$f = x \oplus y = \bar{x} \cdot y + x \cdot \bar{y}$$

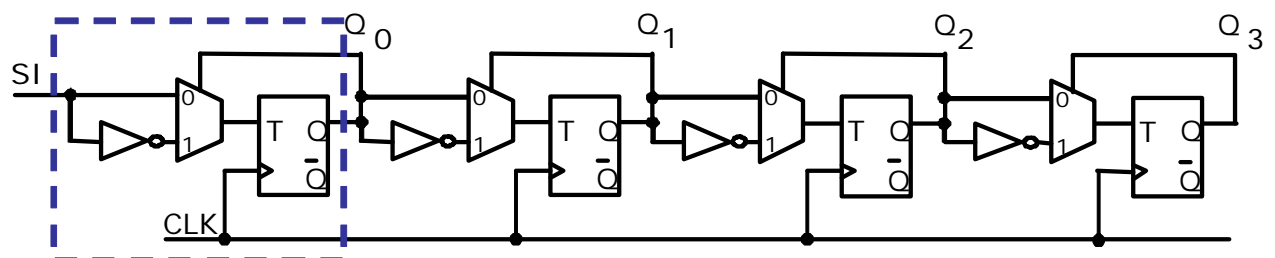
Iz tabele sledi, da je T vhod XOR operacija $Q(t)$ in vhoda D-FF, ki ga realiziramo.

Funkcijo f realiziramo z izbiralnikom tako, da naredimo razvoj po spremenljivki x in dobimo:

$$Q(t+1) = Q(t) \oplus D$$

x	f
0	y
1	y'

Če nastali D-FF iz T-FF in 2/1 izbiralnika sestavimo skupaj v 4-bitni pomikalni register dobimo spodnjo realizacijo, v kateri je izvedba D-FF označena črtkano.

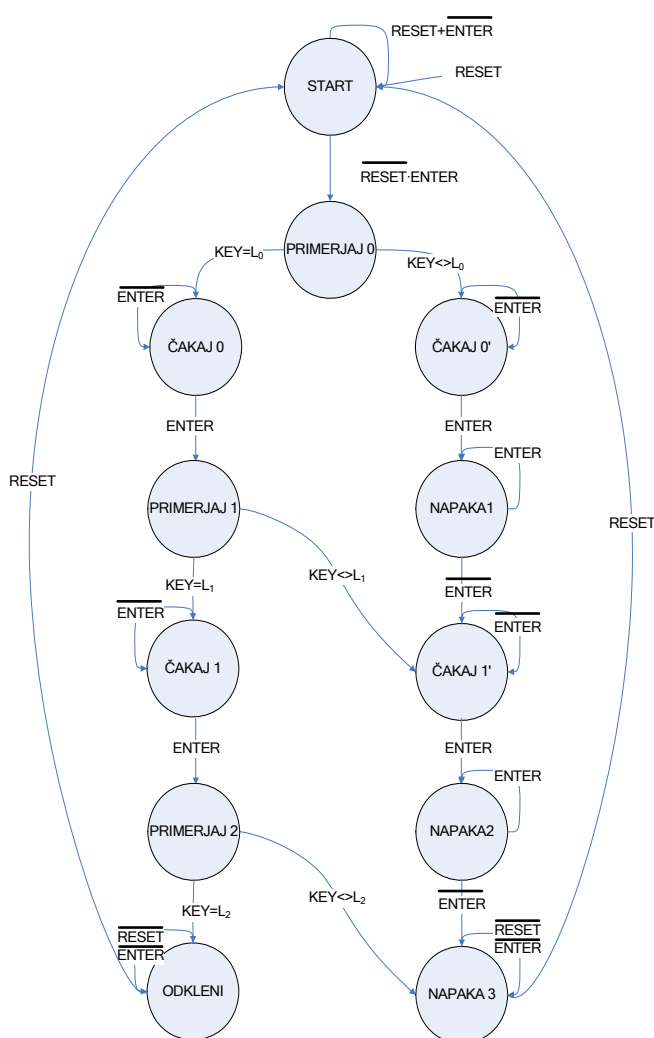
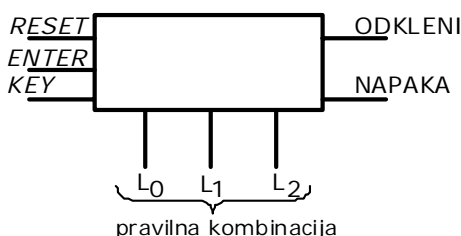


4. Narišite diagram stanj Moore-ov avtomat končnih stanj, ki deluje kot 3-bitna sekvenčna ključavnica. Ključavnica ima tipko za ponastavitev (*RESET*), ki postavlja ključavnico v začetno stanje, tipko (*ENTER*) za vnos nastavljene kombinacije in preklopnik (*KEY*) za vnos enega bita kombinacije. Izhod ključavnice je (*ODKLENI*), ki postane '1' ko uporabnik vnese pravilno kombinacijo in izhod (*NAPAKA*), ki postane '1' če je vnesena kombinacija napačna.

Uporabo ključavnice povzema spodnje zaporedje:

- 1.) pritisnemo *RESET*
- 2.) s preklopnikom *KEY* nastavimo bit kombinacije odklepanja
- 3.) pritisnemo *ENTER*
- 4.) izvede se primerjava i-tega bita ($KEY = L_i$)
- 5.) dvakrat ponovimo korake 2 – 4
- 6.) če je vnesena 3-bitna kombinacija pravilna, se postavi *ODKLENI*='1', sicer se postavi *NAPAKA*='1'.
- 7.) ponoven pritisk na *RESET* nas vrne na korak 1.

Pravilno kombinacijo, ki odklene ključavnico, nastavljamo z biti L_0 , L_1 in L_2 .



Ključavnica se ob vklopu ali ob ponastavitvi (*RESET*) nahaja v stanju *START*. V tem stanju uporabnik nastavi preklopnik *KEY* v stanje prvega bita kombinacije ('0' ali '1'). Iz tega stanja lahko pride v stanje *PRIMERJAJ 0* samo ob pogoju, da *RESET* ni pritisnjen in da je *ENTER* pritisnjen. V tem stanju je lahko vnesena kombinacija pravilna ($KEY = L_0$) ali nepravilna ($KEY \neq L_0$). Če je pravilna, potem preide v stanje *ČAKAJ 0*, v katerem čaka da uporabnik spusti tipko *ENTER*. Uporabnik v tem stanju nastavi drugi bit kombinacije (*KEY*) in ponovno pritisne *ENTER*. Avtomat preide v stanje

PRIMERJAJ 1, od koder sta zopet dve možnosti. Omenjeni postopek se lahko ponavlja za več bitov kombinacije. Bistveno je, da pred vsako primerjavo postavimo stanje čakanja, v katerem čakamo, da uporabnik spusti tipko *ENTER*, nato nastavi bit kombinacije in šele nato preide v stanje nove primerjave ob ponovnem pritisku na *ENTER*. V zadnjem stanju primerjave avtomat preide v stanje *ODKLENI*.

Če se uporabnik pri vnašanju zmoti, preide avtomat v sekvenco stanj napake, po kateri mora vnesti še dve mesti kode (lahko samo dvakrat pritisne *ENTER*) in šele nato preide v stanje *NAPAKA 3*, v katerem se postavi izhod *NAPAKA*.