

RAZVOJ DIGITALNIH SISTEMOV

Izpit

22. 01. 2024

1. Realizirajte funkcijo f s čim manj izbiralniki 4/1.

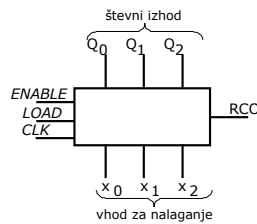
$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot ((a \cdot c \cdot d) \cdot (\bar{c} + d))$$

2. Realizirajte podano funkcijo f z redundancami s čim manj 4-bitnimi aritmetičnimi–logičnimi enotami (ALU). Negacije vhodnih spremenljivk izvedite z ALU.

$$f(x_1, x_2, x_3, x_4) = V(0, 5, 6, 9, 10, 12) \text{ in } V_x(3, 15)$$

3. Prikažite sintezo 3-bitnega sinhronnega števca navzgor z omogočanjem štetja (ENABLE) in vzporednim nalaganjem (LOAD) z D flip-flopi, izbiralniki 2/1 in logičnimi vrati. Podatek se vzporedno nalaga z vhodov (x_2, x_1, x_0).

Števec naj ima poleg števnega izhoda (Q_2, Q_1, Q_0) tudi izhodni prenos za krmiljenje naslednjih stopenj (RCO – ripple carry out). Logika vseh krmilnih signalov je pozitivna. Uporabite poimenovanje signalov, kot je narisano na spodnji sliki.



4. Načrtajte diagram stanj Moore–ovega avtomata končnih stanj, ki krmili delovanje garažnih vrat: Garažna vrata imajo vhod VRATA ter vhod ZAŠČITA, ki postane '1' vedno, ko preko motorja steče dovolj velik tok. Z meritvijo toka na motorju obenem izdelamo funkcijo detekcije obeh končnih položajev, kot tudi zaščito proti oviram na poti vrat. Vezje ima 2-bitni izhod za enosmerni motor:

Koda operacije		Funkcija izhoda
OP ₁	OP ₀	
0	0	motor stoji
0	1	motor pomika vrata navzgor
1	0	motor pomika vrata navzdol

Če pritisnemo gumb VRATA, se vrata začno pomikati navzgor. Če na poti naletijo na oviro ali pridejo do zgornje končne lege, se motor ustavi. Če pritisnemo gumb VRATA ponovno, se začnejo gibati v smeri navzdol. Podobno je v obratni smeri: Če na poti naletijo na oviro ali pridejo do spodnje končne lege, se motor ustavi. Če pritisnemo gumb VRATA ponovno, se začnejo pomikati v smeri navzgor.

Rešitev 1. naloge:

Funkcija f je podana v večnivojski (nenormalni) obliki:

$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot ((a \cdot c \cdot d) \cdot (\bar{c} + d))$$

zato jo najprej poenostavimo z uporabo pravil Boole-ove logike. Izpišemo desni člen funkcije in uporabimo lastnost Boole-ove logike $x \cdot \bar{x} = 0$, lastnost $x \cdot x = x$ in lastnost $0 + x = x$.

$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot (a \cdot c \cdot d \cdot \bar{c} + a \cdot c \cdot d \cdot d)$$

Nad rezultatom ponovno uporabimo lastnost Boole-ove logike $x \cdot \bar{x} = 0$ in lastnost $x \cdot x = x$.

$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot (a \cdot c \cdot d)$$

Rezultat vnesemo v levi del funkcije in znova uporabimo omenjene lastnosti Boole-ove logike:

$$f(a,b,c,d) = (a \cdot \bar{b} \cdot a \cdot c \cdot d + b \cdot c \cdot \bar{d} \cdot a \cdot c \cdot d + b \cdot c \cdot a \cdot c \cdot d)$$

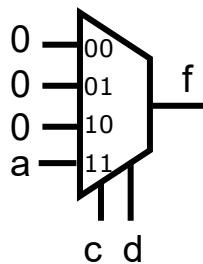
Dobimo dva člena in ju zapišemo v obliki PDNO, ki jo nato minimiziramo s pomočjo Veitch-evega diagrama ali z uporabo lastnosti združevanja Boole-ove algebre $x + \bar{x} = 1$:

$$f(a,b,c,d) = a \cdot \bar{b} \cdot c \cdot d + b \cdot a \cdot c \cdot d$$

$$f_{PDNO}(a,b,c,d) = V(11,15)$$

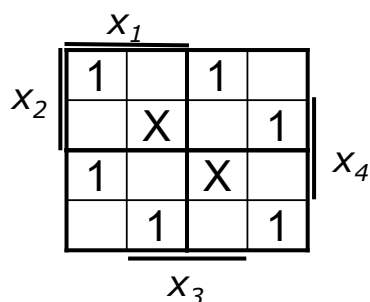
$$f_{MDNO}(a,b,c,d) = a \cdot c \cdot d \cdot (\bar{b} + b) = a \cdot c \cdot d$$

in jo realiziramo z enim izbiralnikom 4/1, tako da naredimo Shannon-ov razvoj funkcije. Glede na kombinacijo naslovnih vhodov izbiralnika dobimo 6 možnih rešitev (ac, ca, ad, da, cd, dc).



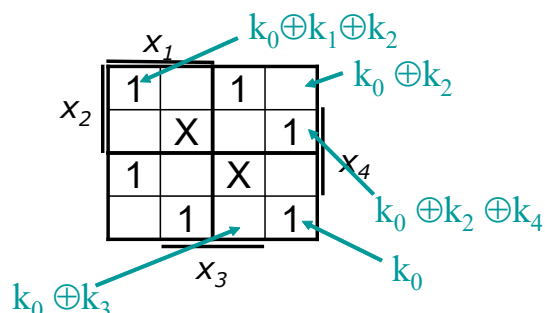
Rešitev 2. naloge:

Funkcijo najprej izrišemo v Veitch–ev diagram:



Funkcija vsebuje same diagonalne člene, zato realizacija v obliki KNO oz. DNO ne nudi minimalne oblike. Če se izkaže, da je funkcija linearna, jo lahko realiziramo s pomočjo XOR funkcij. Linearnost funkcije ugotavljamo tako, da prepogibamo kvadrate diagrama: Začnemo v desnem spodnjem kotu (kjer je minterm 0) in prepognemo kvadrat navzgor, da se spremeni samo ena spremenljivka naenkrat (x_4 postane 0 v prvi iteraciji).

Opazujemo, ali se prepogne na novi kvadrat čisto enako ali pa popolnoma negirano. Če postavimo obe redundanci na '1', lahko s prepogibanjem ugotovimo, da je funkcija linearna.



Podana funkcija je funkcija 4 spremenljivk, zato lahko njeno splošno izražavo kot linearno funkcijo pišemo kot:

$$f(x_1, x_2, x_3, x_4) = k_0 \oplus k_1 x_1 \oplus k_2 x_2 \oplus k_3 x_3 \oplus k_4 x_4$$

S pomočjo Veitch–evega diagrama izračunamo koeficiente.

Iz enačb sledi: $k_0=1$ in $k_0 \oplus k_3=0$, kar pomeni $1 \oplus k_3=0 \rightarrow k_3=1$.

In če napišemo še enačbo za $k_0 \oplus k_2=0$, kar pomeni $1 \oplus k_2=0$ sledi da je $k_2=1$.

Iz enačbe $k_0 \oplus k_2 \oplus k_4=1$, kar pomeni $1 \oplus 1 \oplus k_4=1 \rightarrow k_4=1$.

Analiziramo naprej in dobimo $k_0 \oplus k_1 \oplus k_2=1$, kar pomeni $1 \oplus k_1 \oplus 1=0 \rightarrow k_1=1$.

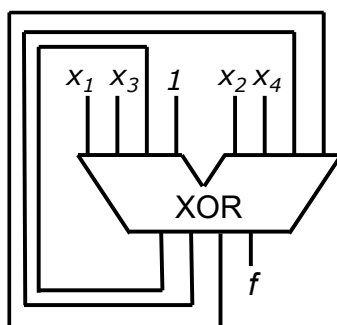
Vstavimo dobljene koeficiente v enačbo za splošno izražavo in dobimo:

$$f(x_1, x_2, x_3, x_4) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4$$

Aritmetično–logično enota lahko poleg aritmetičnih naenkrat realizira štiri dvovhodne logične operacije *istega tipa* (OR, AND, NOT, NOR, NAND, XOR, XNOR), zato nas zanima realizacija zgornje funkcije z dvovhodnimi operatorji enega tipa. Pri realizaciji uporabimo lastnost združevanja, ki velja za XOR funkcijo.

$$f(x_1, x_2, x_3, x_4) = 1 \oplus ((x_1 \oplus x_2) \oplus (x_3 \oplus x_4))$$

Rešitev:



Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števca:

trenutno stanje			naslednje stanje			D–FF		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	D ₂	D ₁	D ₀
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

Iz tabele prehajanja stanj števca določimo enačbe D–FF:

Za D₀ se iz tabele vidi $D_0 = Q_0'$.

Za D₁ narišemo Veitchev diagram

D_1 :

	Q_2			
Q_1	1	0	0	1
	0	1	1	0
	Q_0			

$$D_1 = Q_0 \oplus Q_1$$

Podobno za D₂ narišemo Veitchev diagram

D_2 :

	Q_2			
Q_1	1	0	1	0
	1	1	0	0
	Q_0			

Za D₂ sledi:

$$D_2 = Q_2 \cdot Q_1' + Q_2 \cdot Q_0' + Q_2' \cdot Q_1 \cdot Q_0$$

iz česar lahko izpostavimo:

$$D_2 = Q_2 \cdot (Q_1' + Q_0') + Q_2' \cdot Q_1 \cdot Q_0$$

Uporabimo De Morganovo enakost:

$$D_2 = Q_2 \cdot (Q_1 Q_0)' + Q_2' \cdot Q_1 \cdot Q_0$$

iz česar sledi:

$$D_2 = Q_2 \cdot (Q_1 \cdot Q_0)' + Q_2' \cdot Q_1 \cdot Q_0$$

Upoštevamo definicijo XOR operacije ($a \oplus b = a' \cdot b + a \cdot b'$)

$$D_2 = Q_2 \oplus Q_1 \cdot Q_0$$

Signal RCO postane '1' takrat, ko števec prešteje do svoje največje vrednosti – v našem primeru postane '1', ko gre stanje števca iz "111" na "000".

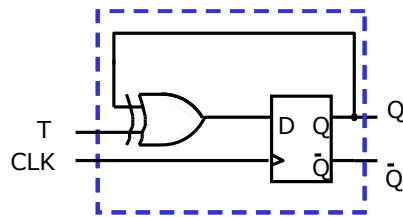
$$RCO = Q_2 \cdot Q_1 \cdot Q_0$$

Naloga pravi, da moramo izdelati števec, ki ima vhod za omogočanje štetja (ENABLE). Če vezje analiziramo, vidimo, da smo pravzaprav realizirali T–FF s pomočjo D–FF in XOR vrat, kot kaže spodnja slika:

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

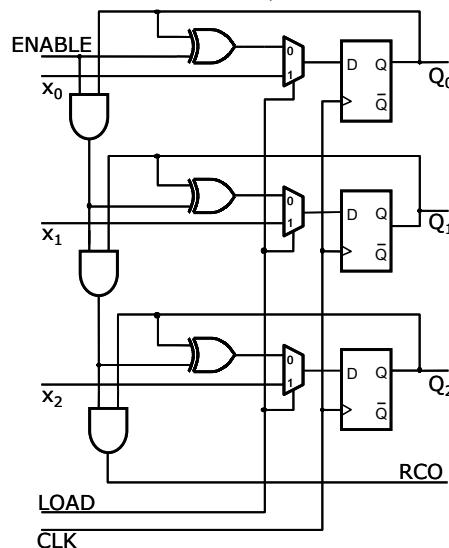
Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>



Slika 1: Realizacija T-FF s pomočjo D-FF.

Če prvemu "T-FF" (D-FF z XOR vrati) postavimo vhod $T_0 = '0'$ namesto $T_0 = '1'$, vsi T-FF ne bodo šteli, ampak bodo ohranjali stanje. Torej, če na vhod T_0 postavimo zunanji signal ENABLE, števec ne bo štel, ampak ohranjal stanje, če bo ENABLE='0'. V verigi sinhronega števca so namreč vsi T-FF vezani tako, da so odvisni od prvega T-FF: Če stanje ohranja prvi, ga bodo tudi vsi ostali.

Za realizacijo signala za vzporedno nalaganje pa izkoristimo osnovno lastnost D-FF (pomnjenje). To storimo tako, da na vhod vsakega D-FF postavimo 2/1 izbiralnik, s katerim določimo, ali se bo dana informacija vpisala s števnege vhoda ali preko zunanjih priključkov. Do iste realizacije bi prišli, če bi v osnovni analizi upoštevali ta dva krmilna signala – analiza je veliko bolj zapletena, saj vsebuje Veitcheve diagrame 5 spremenljivk (ENABLE, LOAD, Q_2 , Q_1 , Q_0).



Slika 2: Sinhroni števec z vzporednim nalaganjem (LOAD) in omogočanjem štetja (ENABLE) (3-bitna izvedba).

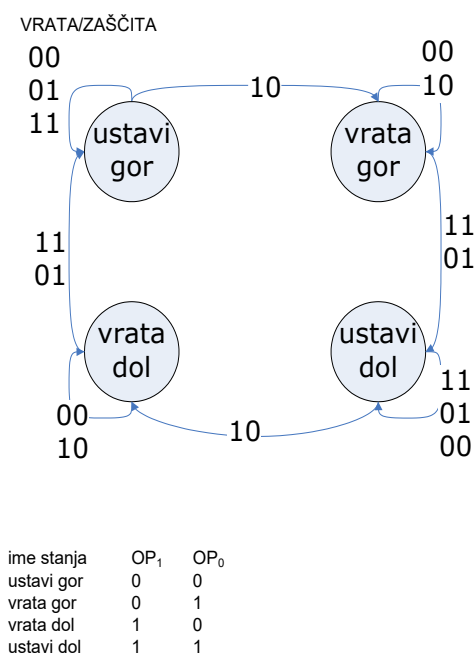
Če želimo z nastalim števcem šteti naraščajoče ... 2, 3, 4, 5, 2, 3, 4, 5 ..., moramo števec, ko le-ta pride do stanja 5 ($Q_2Q_1Q_0=101_2$) postaviti nazaj na stanje v stanje 2 ($Q_2Q_1Q_0=010_2$), torej na LOAD vhod pripeljemo s pomočjo dodatnih dvovhodnih AND vrat.

Pomembno pri tem je, da se pri dekodiranju zavedamo, da se ($Q_2=1$ in $Q_0=1$ v števeni sekvenci pojavlja samo enkrat – če bi se večkrat bi morali dekodirati tudi Q_1).

Pri tovrstnih števcih ponavadi uporabljamo še zunanji signal RESET, s katerim postavimo števec v začetno stanje, kar dosežemo tako, da na vhod izbiralnikov vodimo LOAD OR RESET.

Rešitev 4. naloge:

Narišemo Moore-ov diagram stanj:



Iz opisa naloge je razvidno, da stanje "ustavi" ni samo eno, ker si moramo zapomniti v katero smer so se gibala vrata, da bi lahko šli v nasprotni smeri. Glede na to imamo stanji "ustavi gor", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala gor in stanje "ustavi dol", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala dol. Če stanja ločimo tako, potem v stanju "ustavi gor" ostajamo toliko časa, dokler ne pritisnemo VRATA in jasno na motorju ni napake, se pravi kombinacija "10". Vrata se nato pomikajo gor (preidemo v stanje "vrata gor"). V tem stanju lahko tipko spustimo in vrata se pomikajo navzgor. To se dogaja toliko časa, dokler ne naletimo na pogoj ZAŠČITA='1' (se pravi kombinaciji

"11" in "01". Ko postane pogoj ZAŠČITA='1' se postavimo v stanje "ustavi dol" in v tem stanju ostajamo dokler vztraja pogoj ZAŠČITA='1' oz. dokler ne pritisnemo tipke VRATA='1' (kombinacija "10"). Takrat na podoben način preidemo v stanje "vrata dol", kjer ostanemo dokler ne naletimo na oviro (tla prostora recimo), ko preidemo v stanje "ustavi gor".

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>