

RAZVOJ DIGITALNIH SISTEMOV

Izpit 09. 02. 2012

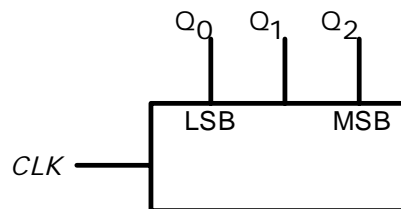
1. Določite redundance podane funkcije f tako, da bo nastala funkcija linearna in izračunajte koeficiente linearnosti.

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

2. Realizirajte funkcijo f z dvovhodnimi vpoglednimi tabelami (ang. look-up table).

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

3. Prikažite sintezo sinhronnega 3-bitnega števca navzdol z uporabo T flip-flopov: Zapišite tabelo prehajanja stanj in določite enačbe flip-flopov ter vezje narišite. Imena signalov so razvidna iz spodnje slike.

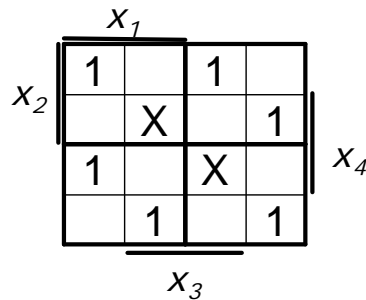


4. Narišite diagram stanj za avtomat končnih stanj, ki ima vhod w in izhod z . Avtomat končnih stanj postavi izhod $z=1$, ko se na vhodu pojavi zaporedje **110** ali **101**, sicer je $z=0$. Prekrivanje vzorcev je dovoljeno. Delovanje avtomata končnih stanj povzema spodnje časovno zaporedje vhoda in izhoda. Tip avtomata je razviden iz podanega časovnega zaporedja.

CLK	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
w	0	1	1	0	1	1	0	1	1	1	0	0	0
z	—	0	0	1	1	0	1	1	0	0	1	0	0

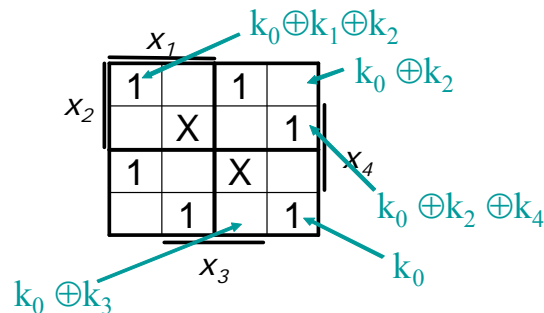
Rešitev 1. naloge:

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:

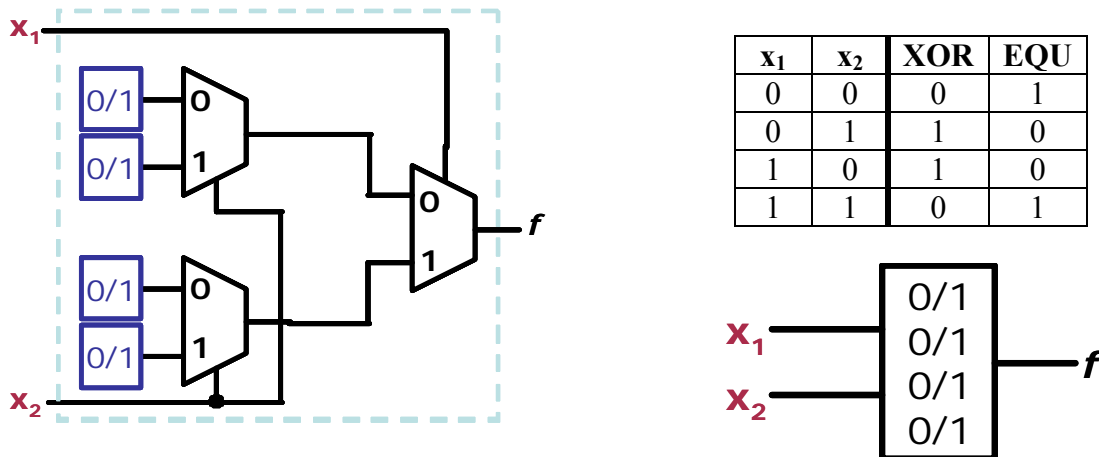
$$f(x_1, x_2, x_3, x_4) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4$$

Rešitev 2. naloge:

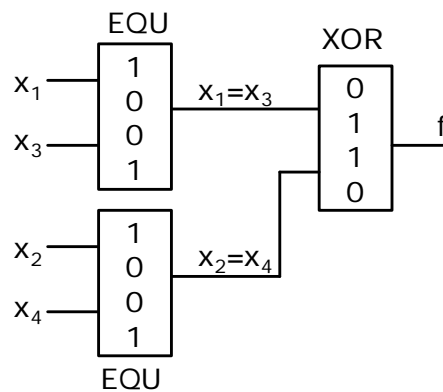
Funkcijo v je že primerna za realizacijo z dvovhodnimi vpoglednimi tabelami (ang. look-up table) v vezju FPGA in je ni treba posebej predelovati.

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

Dvovhodne vpogledne tabele (LUT2) so sestavljene iz pomnilnika (4 RAM celice) in treh 2/1 izbiralnikov. V vsako RAM celico so vpisane vrednosti, ki realizirajo eno od 16 osnovnih dvovhodnih funkcij.



Na zgornji sliki sta prikazani struktura dvovhodne vpogledne tabele (levo) in splošni simbol, ki ga uporabljamo pri risanju realizacij funkcij (desno) ter primer vsebine RAM celic za nekaj osnovnih funkcij (XOR, EQU). Tako lahko LUT uporabljamo za realizacijo funkcij v več nivojih.



Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števca:

Trenutno stanje			Naslednje stanje			Enačbe FF		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	T ₂	T ₁	T ₀
0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	1
1	0	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0	1
1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	0	0	1

Normalna analiza bi zahtevala, da narišemo Veitch–eve diagrame za tri spremenljivke za vsak vhod T–FF, vendar ker so T–FF po svoji naravi primerni za realizacijo števcov, so praviloma njihove vhodne enačbe zelo enostavne. Iz tabele prehajanja stanj števca določimo enačbe T–FF:

Iz stolpca T_0 se vidi:

$$T_0 = 1$$

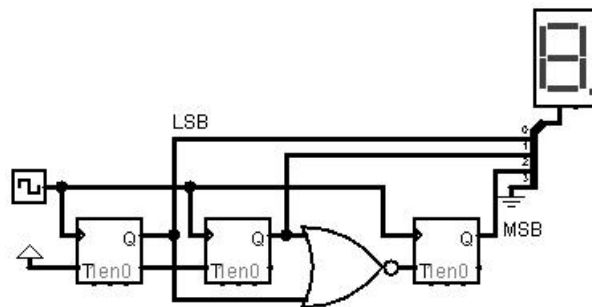
Z opazovanjem stolpcev trenutnega stanja določimo T_1 :

$$T_1 = \overline{Q_0}$$

Podobno lahko določimo T_2 :

$$T_2 = \overline{Q_0} \cdot \overline{Q_1} = \overline{Q_0 + Q_1}$$

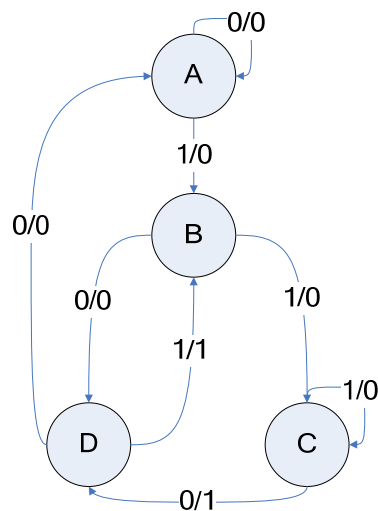
Opis delovanja in vezje števca je v predlogah vaj na domači strani predmeta v imeniku Logisim\counter\ counter_7_0_using_T_FF.circ:



Rešitev 4. naloge:

Naloga zahteva realizacijo z Mealy–evim tipom avtomata. Zapišemo začetno stanje A, v katerem ostajamo toliko časa, dokler se ne začne ena od sekvenc, ki ju zaznavamo. Obe sekvenci se začneta z '1', zato v stanje B preidemo, ko je na vhodu prva '1'. V stanju B ne moremo ostati, saj se na vhodu lahko pojavi '0' ali '1' – v obeh primerih gre za del zaznavanega zaporedja "10X" ali "11X". Iz stanja B preidemo v stanje C, če se vmes pojavi '1', tako da v tem stanju pomeni detekcijo sekvence "11X", v stanje D pa preidemo če se pojavi na vhodu '0', kar pomeni detekcijo sekvence "10X".

Prekrivanje zaporedij: Če se v stanju C pojavi '1' na vhodu, potem gre za sekvenco "111" na vhodu – kar še vedno pomeni, da ostajamo v stanju C, saj je prekrivanje vzorcev dovoljeno. Drugače se diagram obnaša, ko smo v stanju D in pride na vhod še ena '0' – takrat smo imeli na vhodu sekvenco "100", tako da se moramo vrniti v stanje A, saj se nobena od zaznavanih sekvenc ne začneja z '0'.



Delovanje avtomata preizkusimo na testnem zaporedju:

stanje	A	B	D	B	D	A	B	C	D	B	D	A	B	C	D	B	D	B	C	D	A	B	C	D
w	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	0	1	1	0
z	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	0	1