

RAZVOJ DIGITALNIH SISTEMOV

Izpit

06. 02. 2017

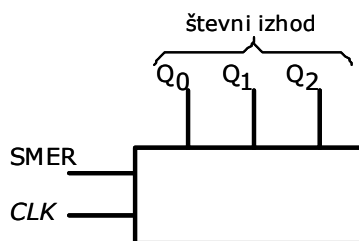
1. Realizirajte funkcijo f z dvovhodnimi vpoglednimi tabelami LUT2 (ang. look-up table).

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

2. Realizirajte podano funkcijo f z eno 4-bitno aritmetično-logično enoto (ALU). Morebitne negacije vhodnih spremenljivk izvedite z ALU.

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

3. Prikažite sintezo sinhronega dvosmernega 3-bitnega dvojiškega števca z uporabo T flip-flopov: Zapišite tabelo prehajanja stanj in določite enačbe flip-flopov. Števec ima vhod SMER, ki določa smer štetja in izhod štetja Q_2 , Q_1 , Q_0 . Če je SMER='0', števec šteje naraščajoče, sicer padajoče. Imena signalov so razvidna iz spodnje slike.



4. Minimizirajte podani avtomat končnih stanj z uporabo metode z razdelki ter zapišite tabelo prehajanja stanj nastalega minimalnega avtomata.

<i>Trenutno stanje</i>	<i>Naslednje stanje</i>		<i>Izhod</i>
	$w=0$	$w=1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

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>

DEVELOPMENT OF DIGITAL SYSTEMS

Written examination

06. 02. 2017

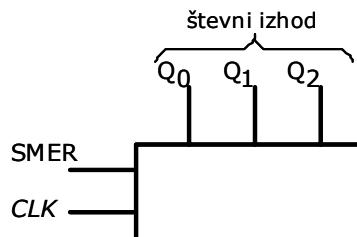
1. Implement a given function f using two input look-up-tables (LUT2).

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

2. Implement a given function f using a single 4-bit arithmetic logic unit (ALU). Any resulting negations of variables must be implemented within this ALU.

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

3. Implement a synchronous 3 bit *up/down* binary counter using T type flip-flops and logic gates: Design the state transition table, determine the flip-flop input equations and draw the resulting circuit using signal names as specified below. The counter has a count direction input (SMER) and a 3-bit binary output Q_2, Q_1, Q_0 . If (SMER='0'), the counter will count upwards, otherwise downwards. Name the signals according to figure below.



4. Minimize a given Moore type finite state machine.

<i>Current state</i>	<i>Next state</i>		<i>Output</i>
	$w=0$	$w=1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

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).

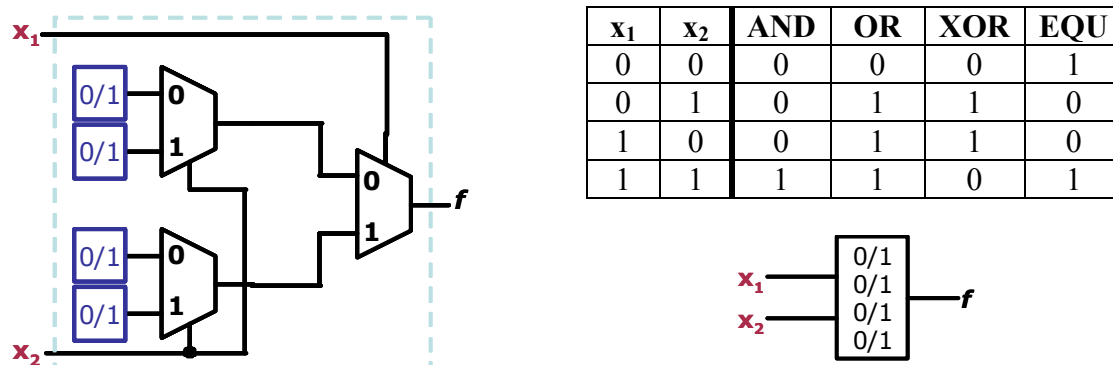
Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

Rešitev 1. naloge:

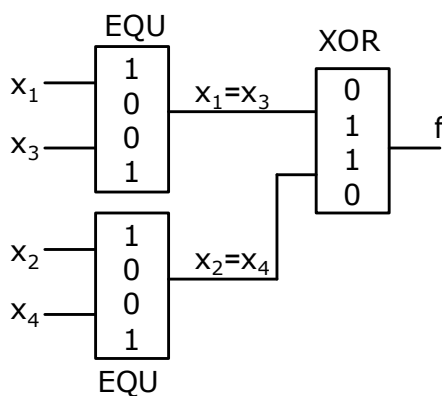
Funkcija je že primerna za realizacijo z dvovhodnimi vpoglednimi tabelami (ang. look-up table) v vezju FPGA in je ni treba posebej predelovati.

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

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 posplošeni 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.



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>

Rešitev 2. naloge:

Funkcija f je podana v obliki MDNO.

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

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 so zato primerne čimbolj nenormalne oblike (večnivojske oblike), samo da vsebujejo operatorje ene vrste. Podana funkcija je v MDNO, zato za neposredno realizacijo s 4-bitno ALU ni primerna, saj vsebuje operaciji AND in OR – torej bi za realizacijo rabili najmanj dve aritmetični–logični enoti in tretjo za izvedbo inverterjev. Funkcijo MDNO prevedemo na operator enega tipa – operator NAND, kar pomeni obliko SNO (Sheffer–jeva normalna oblika funkcije):

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

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

Najprej pri prvih dveh členih izpostavimo člen d , saj petih operacij z eno 4 bitno ALU ne moremo izvesti.

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

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

Za pretvorbo v SNO nad vsemi konjunkcijami izvedemo dvojno negacijo. Nad členom v oklepaju uporabimo De Morganov teorem, da dobimo izražavo z NAND operatorjem.

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

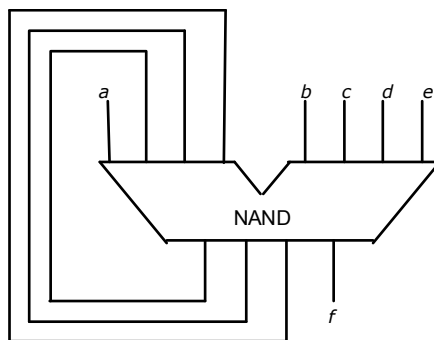
Podobno storimo še enkrat:

$$f(a,b,c,d,e) = \overline{\overline{\overline{\overline{(a \cdot b)} \cdot c} \cdot d} \cdot e}$$

Dobljene NAND operatorje predstavimo z oklepaji.

$$f(a,b,c,d,e) = (((a \uparrow b) \uparrow c) \uparrow d) \uparrow e$$

Narišemo realizacijo:



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>

Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števca:

SMER	Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	T ₂	T ₁	T ₀
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1

Normalna analiza bi zahtevala, da narišemo Veitch–eve diagrame za štiri 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 T₀ se vidi, da je T₀='1'. Iz stolpca T₁ se vidi, da se ponavlja

vzorec 01, če je SMER='0' in 10, če je SMER='1'.

SMER	T ₁
0	Q ₀
1	Q ₀ '

kar lahko kratko zapišemo kot:

$$T_1 = \text{SMER} \cdot \overline{Q_0} + \overline{\text{SMER}} \cdot Q_0 = \text{SMER} \oplus Q_0$$

Za T₂ se da enostavno ugotoviti realizacijo iz Veitch–evega diagrama:

SMER				Q ₀
Q ₂	1	0	0	
	0	0	1	
	0	0	1	
	1	0	0	
Q ₁				

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

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

V enačbi za T₂ poiščemo podobnosti z enačbo za T₁: Enačba za T₁ vsebuje konjunkciji SMER·Q₀' in SMER'·Q₀, ki sta vsebovani tudi v enačbi za T₂, kar nam dodatno poenostavi realizacijo števca. Obenem nam taka realizacija nakazuje osnovno strukturo, ki jo lahko s ponavljanjem razširimo v večbitni dvosmerni sinhroni števec.

Primer podobnega vezja 4-bitnega dvojiškega dvosmernega števca, ki ima še vzporedno nalaganje je 74191¹. Če boste primerjali našo realizacijo in realizacijo v podatkovnem listu, boste opazili, da je v dejanski realizaciji 74191 precej več večvhodnih AND vrat: Delno je razlog za to v dodani logiki za vzporedno nalaganje, delno pa tudi zato, da zagotovimo enakomerno zakasnitev med posameznimi stopnjami števca.

¹ <http://www.alldatasheet.com/view.jsp?Searchword=74191>

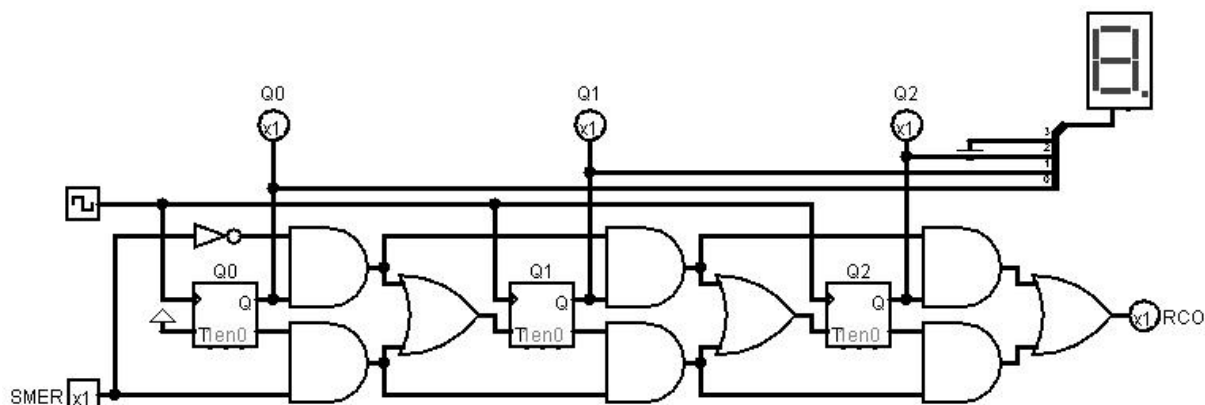
Ko enkrat narišemo vezje dvosmernega števca, zelo spominja na združitev sinhronnega števca za štetje navzgor in sinhronnega števca za štetje navzdol: Če bi števec vseboval samo zgornja AND vrata (vezanih neposredno na T vhod – brez OR) bi bil to števec navzgor, če pa samo spodnja AND bi bil števec navzdol. Signal SMER določa katera AND vrata so omogočena:

- zgornja AND vrata, ko je SMER='0' – štejemo naraščajoče,
- spodnja AND vrata, ko je SMER='1' – štejemo padajoče.

Pri tovrstnih števcih želimo realizirati tudi signal za proženje naslednjih stopenj števca (RCO – oz. ripple carry out, včasih tudi TC – terminal count). RCO je signal, ki postane '1' ob prehodu iz najvišjega stanja števca (v našem primeru je to "111") v stanje "000" pri štetju navzgor in ob prehodu "000" v najvišje stanje števca pri štetju navzdol:

SMER	RCO
0	$Q_0 \cdot Q_1 \cdot Q_2$
1	$Q_0' \cdot Q_1' \cdot Q_2'$

Tak signal uporabljamo pri realizaciji večbitnih števcov tako, da izdelane 3 bitne števec vežemo kaskadno – torej da signal RCO vežemo na EN signal naslednjega vezja. Za realizacijo takega signala bi narisali enako kombinacijo AND in OR vrat še na izhodu Q_2 , kot kaže spodnja slika:



Opis delovanja in vezje števca je v predlogah vaj na domači strani predmeta v imeniku Logisim\counter\ counter_up_down_3_bit_using_T_FF.circ

Večina števcov je realizirana v 4-bitni zasnovi, tako da glede na vrednost RCO signala ločimo dve skupini števcov:

- desetiški (BCD) števci, katerih RCO se postavi na '1' takrat, ko števec preide iz stanja "1001" v "0000" in
- dvojiški (binarni), katerih RCO se postavi na '1' takrat, ko števec preide iz stanja "1111" v "0000". Več o delovanju RCO najdete v opisu delovanja števcov 74161².

² <http://www.alldatasheet.com/view.jsp?Searchword=74161>

Rešitev 4. naloge:

V prvi iteraciji zberemo skupaj vsa stanja v enem razdelku: $P_1 = (ABCDEFGG)$

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

Naslednja iteracija loči stanja, ki imajo različne izhode: $P_2 = (ABD)(CEFG)$

- Pregledamo vsa naslednja stanja pri vhodu 0 in 1 v vsakem bloku:
 - Blok (ABD):
 - Naslednja stanja pri w=0 (BDB)
 - Naslednja stanja pri w=1 (CFG)
 - Blok (CEFG):
 - Naslednja stanja pri w=0 (FFEF)
 - Naslednja stanja pri w=1 (ECDG)

Vsa stanja niso v enem bloku. Problem je pri stanju F, ki ima naslednje stanje D. Zato bo stanje F NEEKVIVALENTNO ostalim CEG.

- Novo stanje F zato postavimo v svojo skupino.

Naslednja iteracija loči stanje F od ostalih $P_3 = (ABD)(CEG)(F)$

- Blok (ABD):
 - Naslednja stanja pri w=0 (BDB)
So vsa v istem bloku
 - Naslednja stanja pri w=1 (CFG) Niso v istem bloku, ker je F v drugem bloku kot C in G. Zato bo stanje B v novem bloku.
- Blok (CEG):
 - Naslednja stanja pri w=0 (FFF)
 - Naslednja stanja pri w=1 (ECG) C, E in G imamo lahko še vedno za ekvivalentna

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0

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>

<i>F</i>	<i>E</i>	<i>D</i>	<i>0</i>
<i>G</i>	<i>F</i>	<i>G</i>	<i>0</i>

Naslednja iteracija loči stanje B od ostalih $P_4 = (AD)(B)(CEG)(F)$

- Blok (*AD*)
 - Naslednja stanja pri $w=0$ (*BB*)
 - Naslednja stanja pri $w=1$ (*CG*)
 - So vsa v istem bloku.
- Blok (*CEG*)
 - Naslednja stanja pri $w=0$ (*FFF*)
 - Naslednja stanja pri $w=1$ (*ECG*) So vsa v istem bloku.

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
<i>A</i>	<i>B</i>	<i>C</i>	<i>1</i>
<i>B</i>	<i>D</i>	<i>F</i>	<i>1</i>
<i>C</i>	<i>F</i>	<i>E</i>	<i>0</i>
<i>D</i>	<i>B</i>	<i>G</i>	<i>1</i>
<i>E</i>	<i>F</i>	<i>C</i>	<i>0</i>
<i>F</i>	<i>E</i>	<i>D</i>	<i>0</i>
<i>G</i>	<i>F</i>	<i>G</i>	<i>0</i>

$P_5 = (AD)(B)(CEG)(F)$

Iteraciji P_5 in P_4 sta enaki, zato se postopek minimizacije zaključi. Stanji *A* in *D* sta ekvivalentni. Stanja *C*, *E* in *G* so ekvivalentna.

- Tabelo stanj zapišemo na novo
- Izbrišemo vrstice za *D*, *E* in *G*
- Zamenjamo stanja: $D \rightarrow A$ in vse $E \rightarrow C$ ter $G \rightarrow C$

Rezultat je nova tabela stanj minimiziranega avtomata:

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
<i>A</i>	<i>B</i>	<i>C</i>	<i>1</i>
<i>B</i>	<i>A</i>	<i>F</i>	<i>1</i>
<i>C</i>	<i>F</i>	<i>C</i>	<i>0</i>
<i>F</i>	<i>C</i>	<i>A</i>	<i>0</i>

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>