

# RAZVOJ DIGITALNIH SISTEMOV

1. kolokvij  
12. 12. 2013

1. Določite minimalno normalno obliko (MNO) za logično funkcijo  $f$  z redundantnimi vhodnimi kombinacijami. Izbiro funkcije utemeljite z izračunom  $COST$  funkcije.

$$f^4 = \sum (0,1,6,7,14,15) \text{ in } \sum_x (3,5,9,11,13)$$

2. Z uporabo "carry lookahead" načina seštevanja seštejte števili  $x$  in  $y$ . Na vseh mestih določite vrednosti funkcij tvorjenja ( $g_i$ ) in širjenja ( $p_i$ ). Končni rezultat izračunajte z uporabo prenosov ( $c_i$ ), ki izvirajo iz pomočjo funkcij ( $g_i$ ) in ( $p_i$ ), ter vhodnih operandov  $x$  in  $y$ .

$$x + y = 19_{10} + 23_{10} = 42_{10}$$

3. Realizirajte podano funkcijo  $f$  z redundantnimi makstermi z enim izbiralnikom 4/1.

$$f(x_1, x_2, x_3, x_4) = \&(1,2,5-7,9,10,14) \text{ in } \&_x(0,4,11,13)$$

4. Pretvorite število  $293_{10}$  ( $100100101_2$ ) v BCD zapis z uporabo "double dabble" algoritma.

# DEVELOPMENT OF DIGITAL SYSTEMS

Midterm Examination

12. 12. 2013

1. Determine the minimal normal form (MNO) of a given incompletely specified function  $f$ . Write the minimal POS (product-of-sums) and SOP (sum-of-products) form, calculate the COST function of each one and determine MNO based on calculated COST function.

$$f^4 = \sum (0,1,6,7,14,15) \text{ in } \sum_x (3,5,9,11,13)$$

2. Add two unsigned numbers  $x$  and  $y$  using carry look-ahead addition technique. Evaluate generate ( $g_i$ ) and propagate ( $p_i$ ) functions for all bit positions. Calculate the sum ( $S_i$ ) using carry bits ( $c_i$ ) and both inputs ( $x_i, y_i$ ). Carry bits ( $c_i$ ) have to be evaluated using generate ( $g_i$ ) and propagate ( $p_i$ ) functions.

$$x + y = 19_{10} + 23_{10} = 42_{10}$$

3. Implement the incompletely specified function  $f$  in POS (product-of-sums) form using a single 4/1 multiplexer using Shannon expansion theorem.

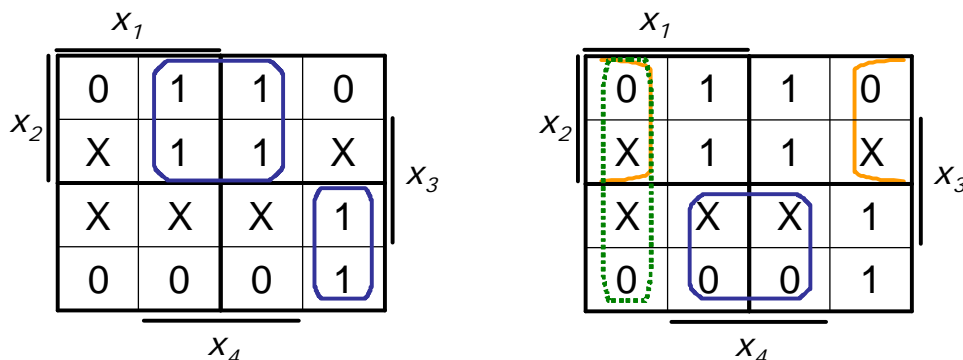
$$f(x_1, x_2, x_3, x_4) = \&(1,2,5-7,9,10,14) \text{ in } \&_x(0,4,11,13)$$

4. Convert the number  $293_{10}$  ( $100100101_2$ ) into BCD representation using "double dabble" algorithm.

Rešitev 1. naloge:

Funkcijo v PDNO z redundancami moramo izpisati v Veitch–ev diagram, od koder bomo lahko izvajali postopek minimizacije.

$$f^4 = \sum (0, 1, 6, 7, 14, 15) \text{ in } \sum_x (3, 5, 9, 11, 13)$$



Prikazana sta dva Veitch–eva diagrama. Levega uporabljamo za tvorbo MDNO, saj prikazuje funkcijo  $f$ , desnega za tvorbo MKNO, ker združujemo minterme negirane funkcije.

MDNO:

$$f_{MDNO} = x_2 \cdot x_4 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_4}$$

Za MKNO pa združujemo ničle (izražamo negirano funkcijo) in izrazimo minimalno obliko negirane funkcije. Nato uporabimo De Morgan–ov teorem, da pridemo do konjunktivne izražave.

$$\begin{aligned} \overline{f} &= x_1 \cdot \overline{x_4} + \overline{x_2} \cdot x_4 + x_2 \cdot \overline{x_4} \\ f &= \overline{x_1 \cdot \overline{x_4} + \overline{x_2} \cdot x_4 + x_2 \cdot \overline{x_4}} \\ f &= \overline{x_1 + x_4} \cdot \overline{x_2 + x_4} \cdot \overline{x_2 + x_4} \\ f_{MKNO} &= (\overline{x_1 + x_4}) \cdot (\overline{x_2 + x_4}) \cdot (\overline{x_2 + x_4}) \end{aligned}$$

Za minimalno normalno obliko moramo določiti, katera izvedba funkcije je cenejša (manjši COST vezja).

OBLIKA	VRAT	VHODOV	COST
MDNO	3	7	10
MKNO	4	9	13

Cenejša izvedba je MDNO, saj je strošek vezja manjši (COST), zato je MNO=MDNO.

## Rešitev 2. naloge:

Uporaba "carry look ahead" načina seštevanja zahteva določitev vrednosti funkcij tvorjenja ( $g_i$ ) in širjenja ( $p_i$ ). Končni rezultat izračunamo z uporabo prenosov, ki izvirajo iz pomožjo funkcij ( $g_i$ ) in ( $p_i$ ), ter vhodnih operandov  $x$  in  $y$ . Bistvena prednost tega načina seštevanja je, da prenose lahko izračunamo vnaprej (ang. look ahead) vzporedno na osnovi enačb, s katerimi dani prenos izrazimo kot funkcijo ( $g_i$ ) in ( $p_i$ ) ter vhodnega prenosa. Zaradi vzporednega načina izračuna prenosov je izračun vsote neodvisen od števila mest seštevalnika. Problem tega načina so kompleksne funkcije, ki nastanejo pri naraščajočem številu mest seštevanja, zato se CLA seštevalniki združujejo po 4 mesta, izhodi za funkcije tvorjenja ( $g$ ) in širjenja ( $p$ ) teh CLA pa se vežejo na generator izhodnih prenosov (ang. carry look ahead generator oz. CLAG).

Način seštevanja "carry look ahead" zahteva določitev vrednosti funkcij tvorjenja ( $g_i$ ) in širjenja ( $p_i$ ) ter prenosa na naslednje mesto ( $c_{i+1}$ ) za vsako mesto posebej.

Osnovne enačbe "carry look ahead" so:

$$\begin{aligned}g_i &= x_i \cdot y_i \\p_i &= x_i \oplus y_i \\c_{i+1} &= g_i + p_i \cdot c_i\end{aligned}$$

Z iterativnim vstavljanjem formule za prenos  $c_{i+1}$  od mesta 0 do mesta 5 (mesto izhodnega prenosa) dobimo:

$$\begin{aligned}c_1 &= g_0 + p_0 \cdot c_{in} \\c_2 &= g_1 + p_1 \cdot c_1 = g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in}) \\c_3 &= g_2 + p_2 \cdot c_2 = g_2 + p_2 \cdot (g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in})) \\c_4 &= g_3 + p_3 \cdot c_3 = g_3 + p_3 \cdot (g_2 + p_2 \cdot (g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in}))) \\c_5 &= g_4 + p_4 \cdot c_4 = g_4 + p_4 \cdot (g_3 + p_3 \cdot (g_2 + p_2 \cdot (g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in}))))\end{aligned}$$

Pri seštevanju upoštevamo, da vhodnega prenosa ni ( $C_{in}=0$ ). Preostali prenosi se izračunajo vzporedno iz izrazov iterativnega vstavljanja, kar je bistvena prednost "carry look ahead" načina seštevanja pred običajnim "ripple carry" načinom seštevanja.

$$\begin{aligned}c_0 &= c_{in} = 0 \\c_1 &= g_0 + p_0 \cdot c_{in} = 1 + 0 \cdot 0 = 1 \\c_2 &= g_1 + p_1 \cdot c_1 = g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in}) = 1 + 0 \cdot 1 = 1 \\c_3 &= g_2 + p_2 \cdot c_2 = g_2 + p_2 \cdot (g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in})) = 0 + 1 \cdot 1 = 1 \\c_4 &= g_3 + p_3 \cdot c_3 = g_3 + p_3 \cdot (g_2 + p_2 \cdot (g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in}))) = 0 + 0 \cdot 1 = 0 \\c_5 &= g_4 + p_4 \cdot c_4 = g_4 + p_4 \cdot (g_3 + p_3 \cdot (g_2 + p_2 \cdot (g_1 + p_1 \cdot (g_0 + p_0 \cdot c_{in})))) = 1 + 0 \cdot 0 = 1\end{aligned}$$

$i$	5	4	3	2	1	0
$x_i$	0	1	0	0	1	1
+	$y_i$	0	1	0	1	1
<hr/>						
	$g_i$	0	1	0	0	1
	$p_i$	0	0	0	1	0
	$c_i$	1	0	1	1	0
<hr/>						
	$S_i$	1	0	1	0	1

Mesta vsote  $S_i$  dobimo z XOR operacijo  $S_i = x_i \oplus y_i \oplus c_i$  pri čemer XOR predstavlja vsoto po modulu 2 oz. funkcijo v PDNO:  $S_i(x_i, y_i, c_i) = V(1, 2, 4, 7)$ .

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. Rezultati bodo objavljeni na <https://studij.fe.uni-lj.si>

### Rešitev 3. naloge:

Funkcija  $f$  je podana v obliki PKNO z redundancami.

$$f(x_1, x_2, x_3, x_4) = \&(1, 2, 5 - 7, 9, 10, 14) \text{ in } \&_x(0, 4, 11, 13)$$

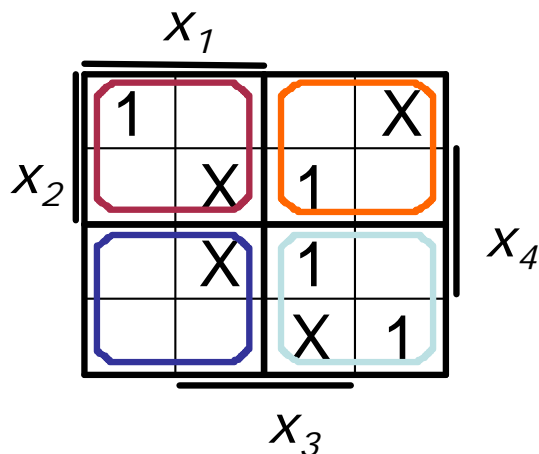
Za potrebe realizacije jo najprej pretvorimo v obliko PDNO. To storimo tako, da maksterme preslikamo v minterme. V pravilnostno tabelo funkcije najprej zapišemo številke mintermov ( $m$ ) in pripadajoče številke makstermov ( $M$ ). Vpišemo  $f=0$  za vse maksterme in  $f=X$  za vse redundantne maksterme. Na preostala mesta vpišemo  $f=1$  in preberemo pri katerih mintermih je  $f=1$  oz.  $f=X$  ter funkcijo izrazimo v obliki PDNO.

Dobimo:

$$f = V(0, 3, 7, 12) \text{ in } V_x(2, 4, 11, 15)$$

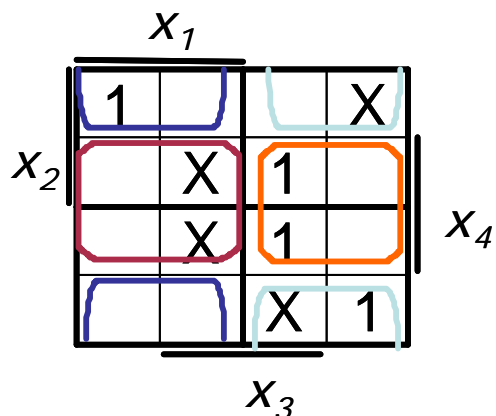
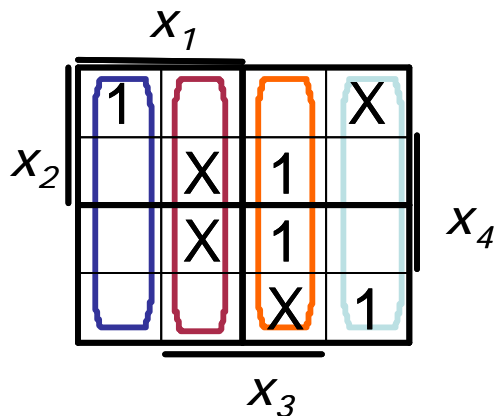
Dobljeno funkcijo vrišemo v Veitch–ev diagram. Ker iščemo najcenejšo realizacijo z izbiralnikom 4/1, bomo naredili razvoj po vseh kombinacijah naslovnih spremenljivk v Veitchev–em diagramu. Če izberemo kot naslovni spremenljivki  $x_1$   $x_2$ , potem dobimo:

$m$	$M$	$x_1$	$x_2$	$x_3$	$x_4$	$f$
0	15	0	0	0	0	1
1	14	0	0	0	1	0
2	13	0	0	1	0	X
3	12	0	0	1	1	1
4	11	0	1	0	0	X
5	10	0	1	0	1	0
6	9	0	1	1	0	0
7	8	0	1	1	1	1
8	7	1	0	0	0	0
9	6	1	0	0	1	0
10	5	1	0	1	0	0
11	4	1	0	1	1	X
12	3	1	1	0	0	1
13	2	1	1	0	1	0
14	1	1	1	1	0	0
15	0	1	1	1	1	X

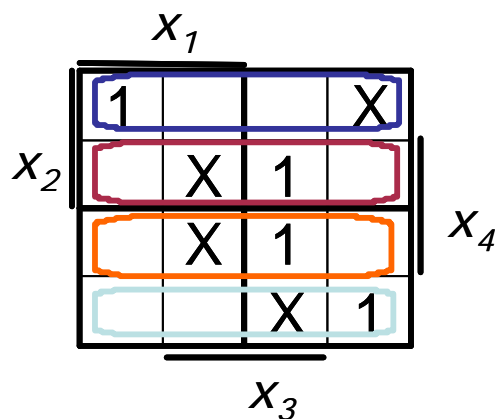
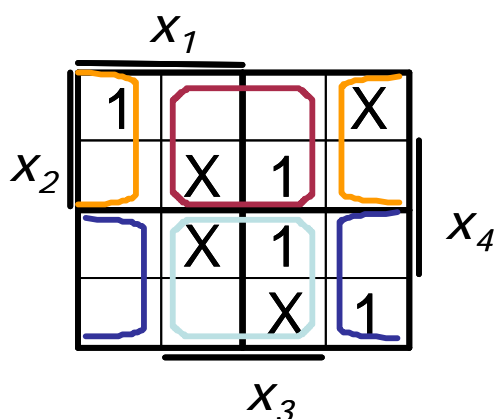


V zgornjem Veitchevem diagramu so označena vsa štiri polja funkcijskih ostankov, če izberemo vhodni spremenljivki  $x_1$   $x_2$ . Zgornji levi kvadrat (rdeč) pomeni, da bo to polje izbrano ko bosta  $x_1x_2="11"$ , oranžni kvadrat ko bo  $x_1x_2="01"$ , temno modri ko bo  $x_1x_2="10"$  in svetlo modri ko bo  $x_1x_2="00"$ . Vsakega od teh kvadratov poskušamo opisati s čimbolj enostavno funkcijo: Vrednost zgornjega levega kvadrata je komplicirana, saj moramo vsako '1' opisati posebej: Za zgornjo '1' v tem kvadratu velja  $x_3x_4$ . Če bi v tem kvadratu postavili redundanco na '1' jo bi opisali kot  $x_3x_4$ . Funkcija bo torej  $x_3x_4 + x_3x_4$ , kar je enačba funkcije ekvivalence (XNOR). Podobno sklepanje velja za zgornji in spodnji desni kvadrat. Najbolj enostavna realizacija je spodnji levi kvadrat je konstanta '0', če postavimo redundanco na '0'. Za ostale možnosti realizacije moramo narisati še preostalih pet kombinacij dveh naslovnih vhodov. Če izberemo kot naslovni spremenljivki  $x_1$  in  $x_3$ , dobimo levi Veitchev diagram, če  $x_1$  in  $x_4$ , pa desnega. Podobno kot v prejšnjem primeru poiščemo realizacije ustreznih kvadratov in iščemo najenostavnejšo realizacijo: Izogibamo se veliko različnim funkcijam in iščemo drugače kvadratov, ki vsebujejo konstante (same '1' ali same '0'). Pri razvoju po  $x_1$  in  $x_3$  imamo pri  $x_1x_3="10"$  najneugodnejšo funkcijo, saj vsebuje eno samo '1'. Pri razvoju po  $x_1$  in  $x_4$  nastopa ena sama '1' pri kombinaciji  $x_1x_4="10"$ .

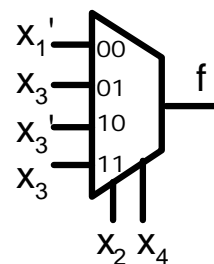
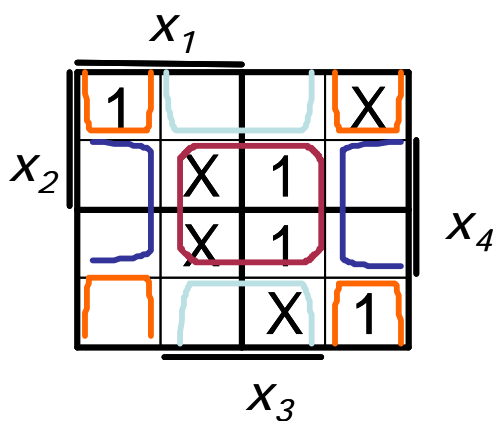
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk. Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete. Rezultati bodo objavljeni na <https://studij.fe.uni-lj.si>



Nato izberemo naslovni spremenljivki  $x_2$  in  $x_3$ , (levi diagram) in  $x_2$  in  $x_4$  (desni diagram). Pri razvoju po  $x_2$  in  $x_3$  imamo pri  $x_2x_3="00"$  najneugodnejšo funkcijo (moder), saj vsebuje eno samo '1'. Pri razvoju po  $x_2$  in  $x_4$  nikjer nimamo osamljene '1', zato se dajo funkcijski ostanki enostavno realizirati, če vse redundance postavimo na '1'.



Zadnja kombinacija naslovnih vhodov je  $x_3$  in  $x_4$ . Pri razvoju po  $x_3$  in  $x_4$  imamo pri  $x_3x_4="00"$  najneugodnejšo funkcijo (oranžen), saj dve '1' opišemo s funkcijo ekvivalence. Možno rešitev torej predstavlja kombinacija naslovnih vhodov  $x_2$  in  $x_4$ .



Vezje izbiralnika je v predlogah avditornih vaj na domači strani predmeta:  
Logisim\MUX\mux\_4\_1\_f\_V\_0\_3\_7\_12\_in\_Vx\_2\_4\_11\_15.circ

Rešitev 4. naloge:

Število  $293_{10}$  pretvorimo v dvojiško obliko:  $1\ 0010\ 0101_2$ . Zapis posameznih števk v BCD zapisu se glasi: 0010 1001 0011. Število  $293_{10}$  je 9 bitno število, zato je pomikov 9.

STOTICE	DESETICE	ENICE	1	0	0	1	0	0	1	0	1	START
		1	0	0	1	0	0	1	0	1		POMIK1
		1 0	0	1	0	0	1	0	1			POMIK2
		1 0 0	1	0	0	1	0	1				POMIK3
		1 0 0 1	0	0	1	0	1					POMIK4
		1 1 0 0	0	0	1	0	1					ADD3
	1	1 0 0 0	0	1	0	1						POMIK5
	1	1 0 1 1	0	1	0	1						ADD3
	1 1	0 1 1 0	1	0	1							POMIK6
	1 1	1 0 0 1	1	0	1							ADD3
	1 1 1	0 0 1 1	0	1								POMIK7
	1 0 1 0	0 0 1 1	0	1								ADD3
1	0 1 0 0	0 1 1 0	1									POMIK8
1	0 1 0 0	1 0 0 1	1									ADD3
1 0	1 0 0 1	0 0 1 1										POMIK9
$2_{10}$	$9_{10}$	$3_{10}$										