

RAZVOJ DIGITALNIH SISTEMOV

1. kolokvij
12. 12. 2016

1. Izrazite podano logično funkcijo v konjunktivni normalni obliki samo s Shefferjevimi operatorji. Morebitne negacije realizirajte s Shefferjevim operatorjem.

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

2. Določite minimalno normalno obliko (MNO) funkcije f z uporabo diagramov za minimizacijo funkcij in COST funkcije. Dobljeno obliko MNO realizirajte z eno 4-bitno aritmetično-logično enoto (ALU). Negacije vhodnih spremenljivk izvedite z ALU.

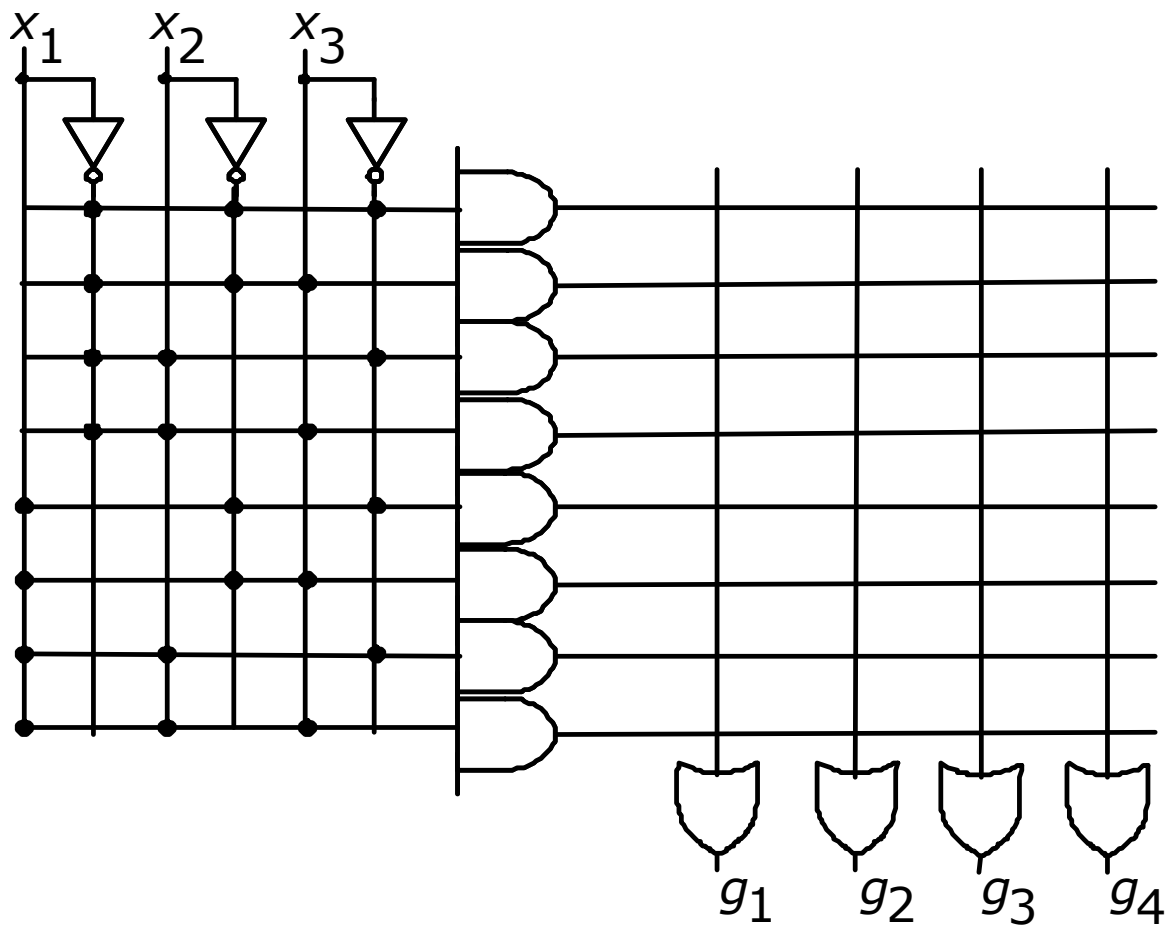
$$f(x_1, x_2, x_3, x_4) = \&(1, 2, 5, 6, 9, 10, 14) \text{ in } \&_x(0, 4, 7, 11, 13)$$

3. Programirajte ROM vezje za realizacijo naslednjih funkcij:

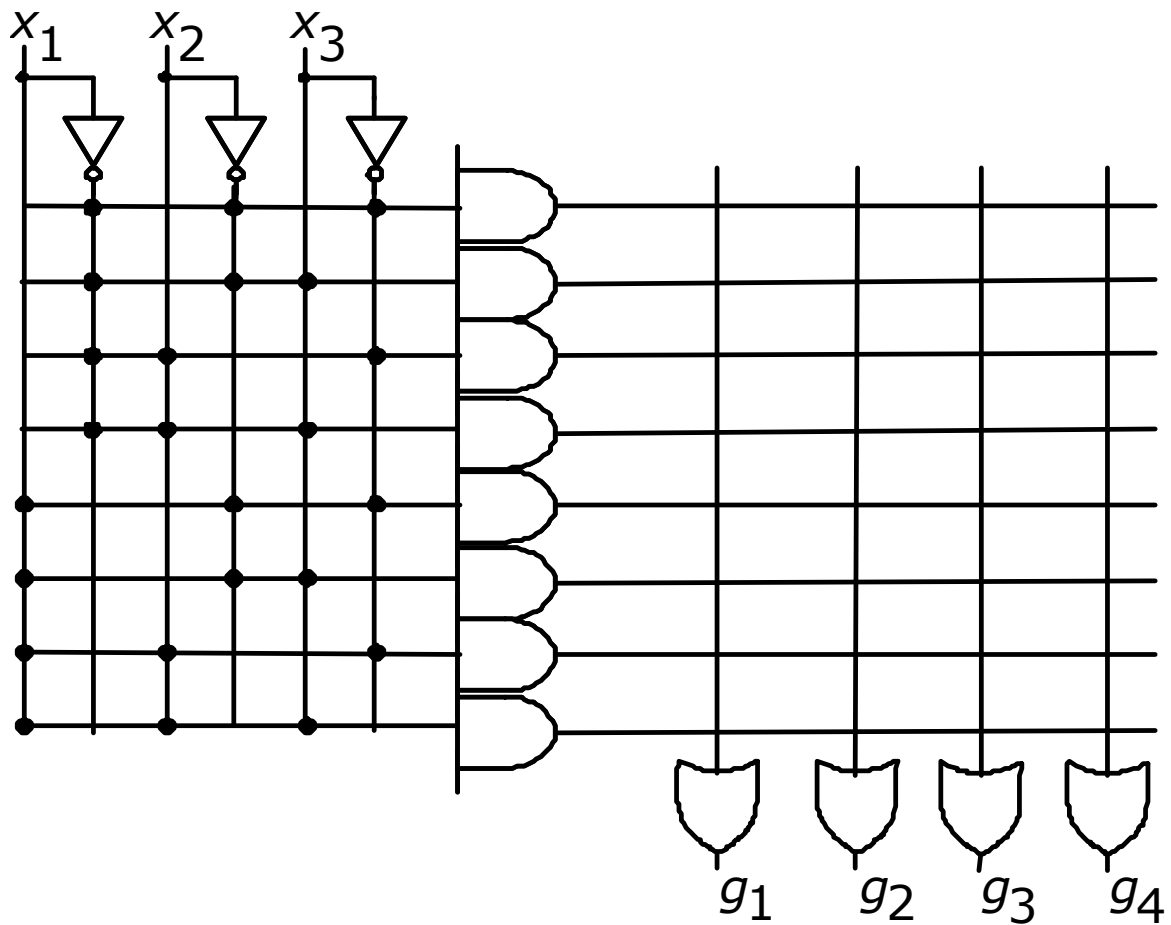
$$g_1 = x_1 + \bar{x}_2 \cdot \bar{x}_3 \quad g_2 = \bar{x}_1 \cdot x_3 + x_1 \cdot x_2 \quad g_3 = \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \quad g_4 = \bar{x}_2 \cdot x_3 + x_1$$

ROM vezje ima 3 vhodne spremenljivke in 4 bitno vsebino. Povezave oz. 'varovalke' označite s piko (●). Uporabite shemo na hrbtni strani.

4. Pretvorite število 2017_{10} ($7E1_{16}$) v BCD zapis z uporabo "double dabble" algoritma.



Če se zmotite, uporabite spodnjo shemo - ne obeh!



DEVELOPMENT OF DIGITAL SYSTEMS

Midterm Examination

12. 12. 2016

1. Implement a given function f , expressed in a conjunctive form below, using only Sheffer operators. All possible variable negations have to be implemented using Sheffer operator.

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

2. Write the minimal normal form (MNO) of a given function f , using Veitch minimization diagrams and COST function. Implement this minimal normal form using a single 4-bit arithmetic logic unit (ALU). All possible variable negations have to be implemented using this ALU.

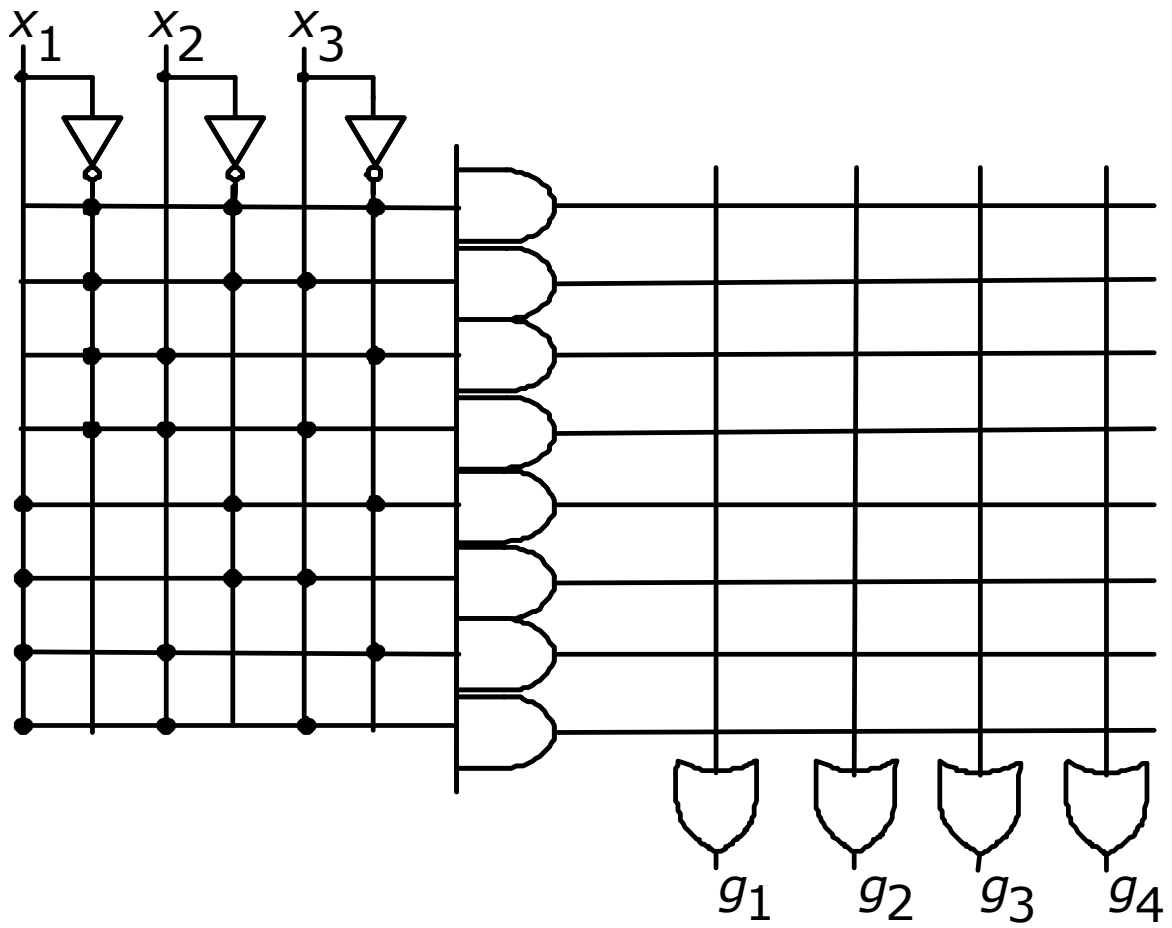
$$f(x_1, x_2, x_3, x_4) = \&(1, 2, 5, 6, 9, 10, 14) \text{ in } \&_x(0, 4, 7, 11, 13)$$

3. Program a ROM element, which implements following functions:

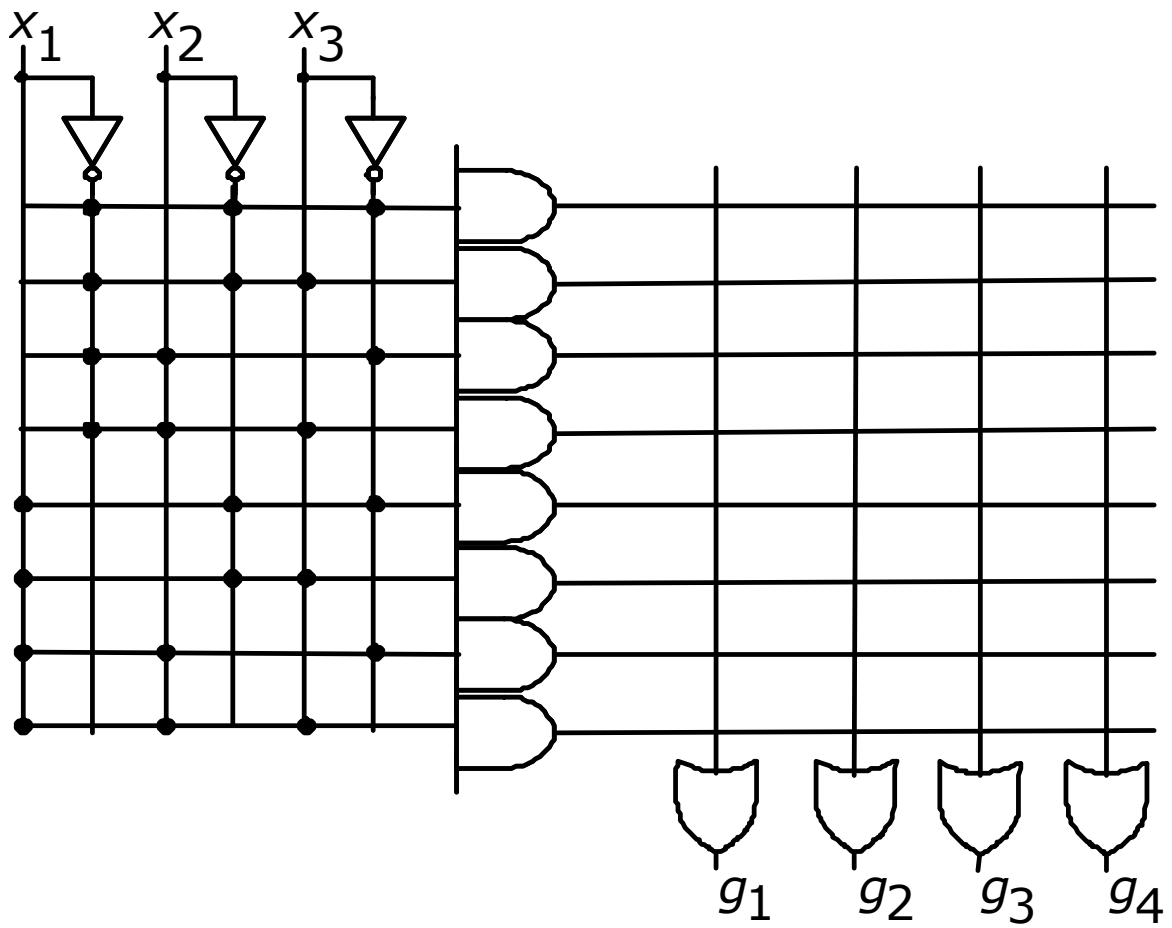
$$g_1 = x_1 + \overline{x_2} \cdot \overline{x_3} \quad g_2 = \overline{x_1} \cdot x_3 + x_1 \cdot x_2 \quad g_3 = \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \quad g_4 = \overline{x_2} \cdot x_3 + x_1$$

The ROM element has 3 inputs and 4-bit contents. Mark the programmed connections using a dot symbol (●). Use the provided schematic on the back side.

4. Convert the decimal number 2017_{10} ($7E1_{16}$) into its BCD representation using "double dabble" algorithm.



Upon error, use the scheme below - not both!



Rešitev 1. naloge

Shefferjeva normalna oblika zahteva realizacijo s samimi [NAND](#) (\uparrow) operatorji:

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

Celotno funkcijo negiramo dvakrat in obenem negiramo dvakrat tudi vse disjunktivne člene (OR) ter zapišemo na novo:

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

Nad disjunktivnima členoma uporabimo De Morganov teorem, tako da iz teh členov nastanejo konjukcije (AND). Preostala negacija nad členom skupaj s konjunkcijo tvori NAND operator.

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

Iz nastale funkcije samo še preberemo NAND operatorje:

$$f(a, b, c) = \overline{(\bar{a} \uparrow \bar{b}) \uparrow (b \uparrow c) \uparrow c}$$

Negacije v nastalem izrazu izpišemo, pri čemer uporabimo lastnost Shefferjevih operatorjev ($\bar{x} = x \uparrow x$ oziroma $\bar{x} = 1 \uparrow x$) in dobimo končni rezultat:

$$f(a, b, c) = (((a \uparrow a) \uparrow (b \uparrow b)) \uparrow (b \uparrow c) \uparrow c) \cdot (((a \uparrow a) \uparrow (b \uparrow b)) \uparrow (b \uparrow c) \uparrow c)$$

Enostavnejšo in cenejšo (COST) rešitev dobimo, če funkcijo predhodno poenostavimo z uporabo pravil Boole-ove algebre, pri čemer velja ($x \cdot \bar{x} = 0$ in $x \cdot x = x$):

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

Dobljena funkcija ima samo en člen, nad katerim izvedemo dvojno negacijo, eno uporabimo za izražavo s Sheffer-jevimi operatorji, drugo pa izrazimo kot ($\bar{x} = x \uparrow x$ oziroma $\bar{x} = 1 \uparrow x$):

$$\begin{aligned} f(a, b, c) &= \overline{a \cdot \bar{b} \cdot c} = \overline{a \uparrow \bar{b} \uparrow c} = \overline{a \uparrow (b \uparrow b) \uparrow c} \\ f(a, b, c) &= (a \uparrow (b \uparrow b) \uparrow c) \uparrow (a \uparrow (b \uparrow b) \uparrow c) \end{aligned}$$

Rešitev 2. naloge:

Funkcija f je podana v obliki PKNO z redundancami:

$$f(x_1, x_2, x_3, x_4) = \&(1, 2, 5, 6, 9, 10, 14) \text{ in } \&_x(0, 4, 7, 11, 13)$$

Najprej jo pretvorimo v obliko PDNO, 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.

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	X
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

Dobimo:

$$f = V(0, 3, 7, 12) \text{ in } V_x(2, 4, 8, 11, 15)$$

Funkcijo minimiziramo, zapišemo v MDNO in MKNO ter poiščemo MNO.

	x_1				
x_2	1	0	0	X	x_4
	0	X	1	0	
	0	X	1	0	
	X	0	X	1	
	x_3				

$$f_{MDNO} = x_3 \cdot x_4 + \overline{x_3} \cdot \overline{x_4}$$

Podobno storimo še za MKNO.

	x_1				
x_2	1	0	0	X	x_4
	0	X	1	0	
	0	X	1	0	
	X	0	X	1	
	x_3				

$$\overline{f_{MKNO}} = \overline{x_3 \cdot x_4 + \overline{x_3} \cdot \overline{x_4}}$$

$$f_{MKNO} = \overline{x_3 \cdot x_4 + \overline{x_3} \cdot \overline{x_4}}$$

$$f_{MKNO} = (\overline{x_3 \cdot x_4}) \cdot (\overline{\overline{x_3} \cdot \overline{x_4}})$$

$$f_{MKNO} = (\overline{x_3} + x_4) \cdot (x_3 + \overline{x_4})$$

V obeh realizacijah preštejemo operatorje ter število vhodov, ter rezultate povzamemo v spodnji tabeli

Tabela 1: COST funkcija.

COST	VRAT	VHODOV
MDNO	3	6
MKNO	3	6

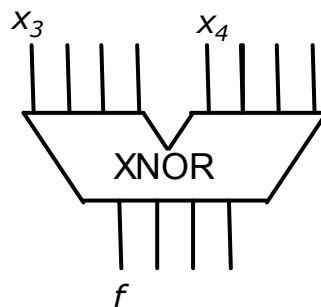
Iz tabele sledi: MNO=MDNO=MKNO.

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 (Sheffer, Pierce, linearne funkcije).

Podana funkcija v MNO zato za neposredno realizacijo s 4-bitno ALU ni primerna. Enostavno jo prevedemo na operator enega tipa, če izrazimo MDNO ali MKNO, saj gre v obeh primerih za XNOR operacijo:

$$f_{MDNO} = f_{MKNO} = x_3 \cdot x_4 + \overline{x_3} \cdot \overline{x_4} = \overline{x_3 \cdot x_4 + \overline{x_3} \cdot \overline{x_4}} = \overline{x_3 \oplus x_4} = x_3 \oplus x_4 = (x_3 \equiv x_4)$$

Aritmetično logično enoto nastavimo tako, da opravlja 4-bitno XNOR operacijo. Na vhoda npr. prvega mesta postavimo x_3 in x_4 ter na prvem izhodu dobimo f .



Rešitev 3. naloge:

Če se funkcije ne nahajajo v popolni disjunktivni normalni obliki (PDNO), jih prevedemo v to obliko z uporabo pravil Boole-ove algebre. Funkcijo lahko tudi izpišemo v Veitch-ev diagram in izpišemo številke mintermov, kjer je funkcija enaka '1'.

$$\begin{aligned}g_1(x_1, x_2, x_3) &= x_1 + \overline{x_2} \cdot \overline{x_3} = x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3) + (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot \overline{x_3} \\g_1(x_1, x_2, x_3) &= x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \\g_1(x_1, x_2, x_3) &= x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \\g_1(x_1, x_2, x_3) &= V(4, 6, 5, 7, 0)\end{aligned}$$

Podobno storimo še za preostale funkcije:

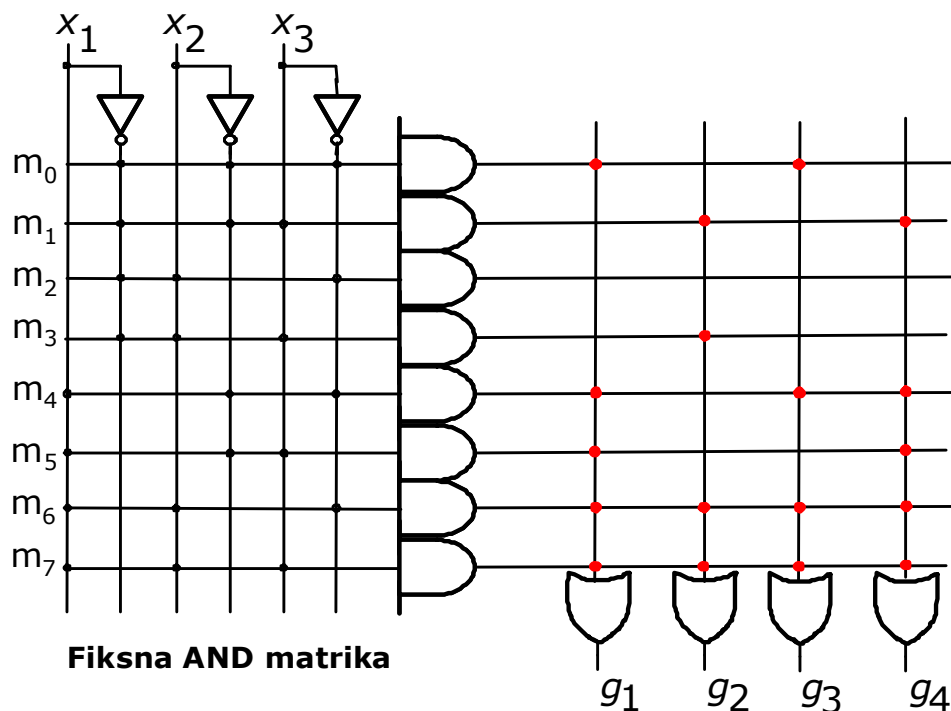
$$\begin{aligned}g_2 &= \overline{x_1} \cdot x_3 + x_1 \cdot x_2 = \overline{x_1} \cdot (\overline{x_2} + x_2) \cdot x_3 + x_1 \cdot x_2 \cdot (\overline{x_3} + x_3) \\g_2(x_1, x_2, x_3) &= \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3 \\g_2(x_1, x_2, x_3) &= V(1, 3, 6, 7)\end{aligned}$$

$$\begin{aligned}g_3 &= \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 = \\g_3(x_1, x_2, x_3) &= (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot (\overline{x_3} + x_3) = \\g_3(x_1, x_2, x_3) &= \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3 \\g_3(x_1, x_2, x_3) &= V(0, 4, 6, 7)\end{aligned}$$

$$\begin{aligned}g_4(x_1, x_2, x_3) &= \overline{x_2} \cdot x_3 + x_1 \\g_4(x_1, x_2, x_3) &= (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot x_3 + x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3) \\g_4(x_1, x_2, x_3) &= (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot x_3 + x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3) \\g_4(x_1, x_2, x_3) &= \overline{x_1} \cdot \overline{x_2} \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 \\g_4(x_1, x_2, x_3) &= V(1, 5, 4, 6, 7)\end{aligned}$$

PDNO je najprimernejša oblika za realizacijo z ROM, ker je matrika AND fiksna. Programirane vrednosti AND matrice predstavljajo vse minterme funkcije treh spremenljivk (x_1, x_2, x_3) od m_0 do m_7 . Številka minterma določa naslov lokacije ROM pomnilnika.

Narišemo celotno vezje ROM strukture in vstavimo pike (•) v OR matriki tam, kjer želimo programirati določeno spremenljivko v členu PDNO.



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 domači strani predmeta.

Vsebino ROM pomnilnika kratko opišemo s tabelo v kateri programirane povezave (●) v OR matriki pišemo kot '1' na danem mestu vsebine.

<i>Minterm</i> $g_i(x_1x_2x_3)$	<i>Naslov lokacije</i> $x_1x_2x_3$	<i>Vsebina</i> <i>lokacije</i> $g_1g_2g_3g_4$
m_0	000_2	1010_2
m_1	001_2	0101_2
m_2	010_2	0000_2
m_3	011_2	0100_2
m_4	100_2	1011_2
m_5	101_2	1001_2
m_6	110_2	1111_2
m_7	111_2	1111_2

Realni ROM elementi imajo 8 bitne podatke, zloge ali včasih oktete (ang. *byte, octet*). Vsebina ROM elementov se podaja v datoteki, ki jo nato programiramo z posebnim inštrumentom (ROM programatorjem).

Najenostavnejši način podajanja zapisa vsebine ROM elementa je v surovi dvojiški obliki (ang. *raw binary file*) v kateri si 8 bitni podatki sledijo zapisani v dvojiški obliki. Kompleksnejša zapisa podajanja vsebine ROM s tabelo sta INTEL šestnajstiška oblika (ang. [*Intel hex record*](#)) in Motorola SREC oblika (ang. [*Motorola S record*](#)).

Rešitev 4. naloge:

Število $2017_{10} = 111\ 1110\ 0001_2$.

Zapis posameznih števk v BCD zapisu se glasi: 0010 0000 0001 0111. Prva pomika smo v spodnji shemi preskočili, ker do prištevanja ne more priti.

TISOČICE		STOTICE		DESETICE		ENICE		Dvojiško število										
								1	1	1	1	1	1	0	0	0	0	1
							1	1	1	1	1	0	0	0	0	1		
							1	1	1	1	0	0	0	0	1			
							1	0	1	0								
					1		0	1	0	1	1	1	0	0	0	1		
							1	0	0	0								
					1	1	0	0	0	0	1	1	0	0	1			
					1	1	0	0	0	1	1	0	0	0	1			
					1	0	0	1	1									
		1			0	0	1	0	0	1	1	0	0	0	1			
							1	0	0	1								
		1	0		0	1	0	1	0	0	1	0	0	0	1			
					1	0	0	0										
					1	0	0	0										
		1	0	1	0	0	0	0	0	1	0	0	0	1				
		1	0	0	0													
		1	0	0	0	0												
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0	0	0	0	0	0	0	0	0	0	0	1				
		1	0</															

Happy 2017. ;-)

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 domači strani predmeta.