

RAZVOJ DIGITALNIH SISTEMOV

1. kolokvij

18. 12. 2023

1. Izrazite podano logično funkcijo f samo s Pierceovimi operatorji. Morebitne negacije realizirajte s Pierceovim operatorjem.

$$f(x_1, x_2, x_3) = (x_1 \downarrow x_2) \cdot \overline{x_3} + ((\overline{x_2} \equiv x_3) \downarrow \overline{x_1})$$

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.

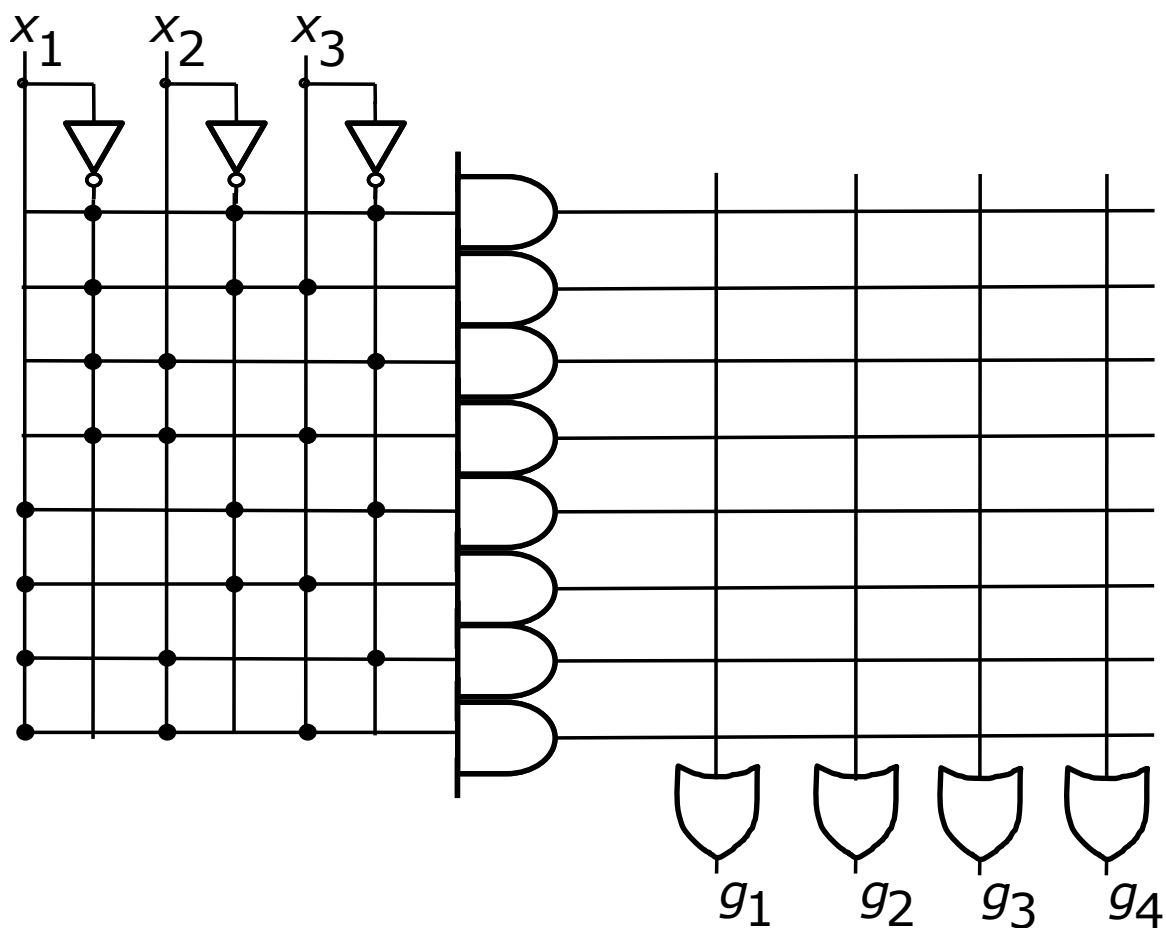
$$f(x_1, x_2, x_3, x_4) = \&(1, 2, 5, 6, 9, 10, 14) \text{ in } \&_x(0, 4, 7, 11, 13)$$

3. Programirajte ROM vezje za realizacijo naslednjih funkcij:

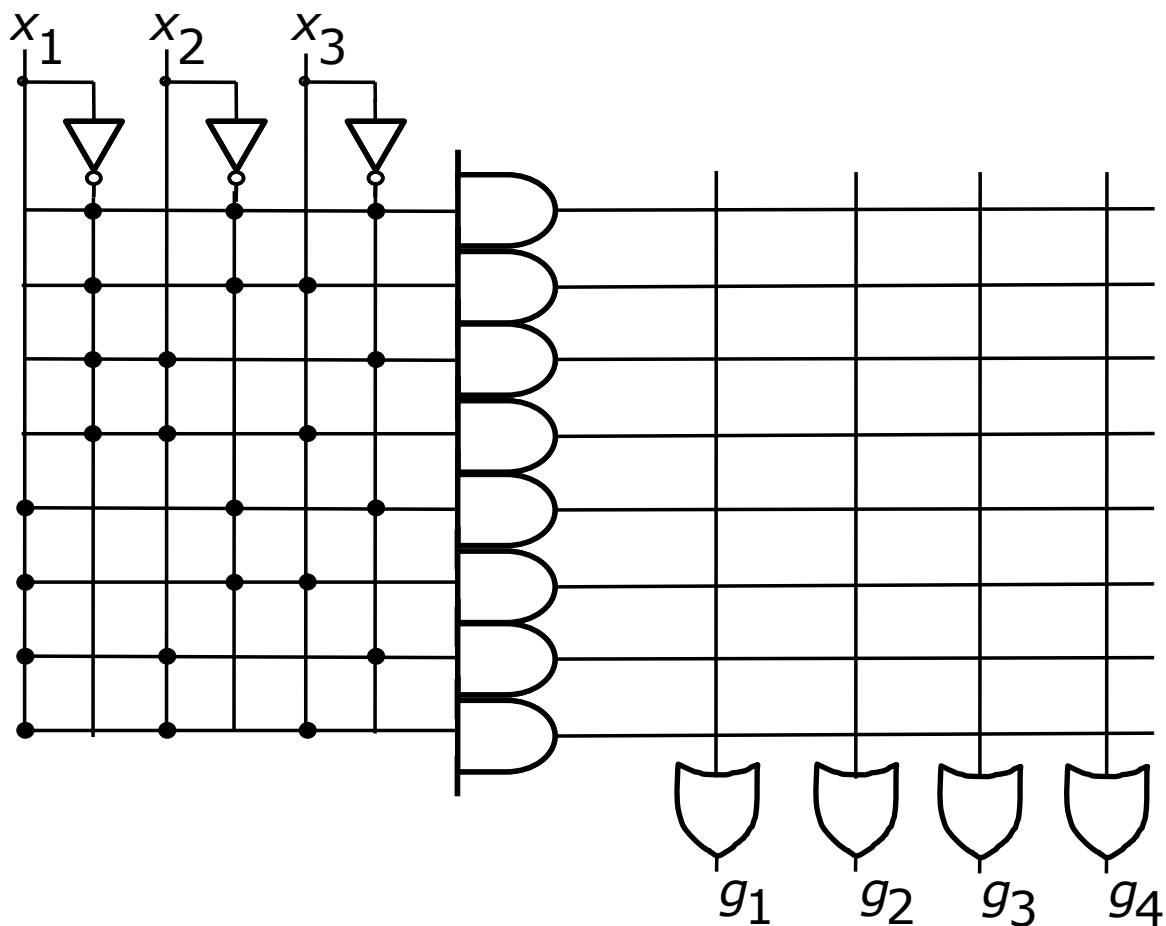
$$g_1 = x_1 + \overline{x_2} \cdot \overline{x_3} \quad g_2 = \overline{x_1} \cdot x_3 + x_1 \cdot x_2 \quad g_3 = \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \quad g_4 = \overline{x_2} \cdot x_3 + x_1$$

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

4. Pretvorite število $2024_{24} = 27700_{10} = 6C34_{16}$ v BCD zapis z uporabo "double dabble" algoritma.

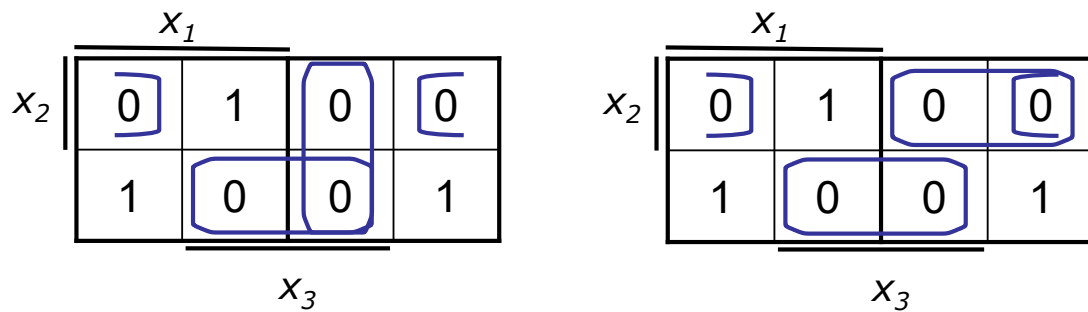


Če se zmotite, uporabite spodnjo shemo - ne obeh!



Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.
Na list z rešitvami se podpišite in napišite še vpisno številko.
Rezultati bodo objavljeni na domači strani predmeta.

Druga pot do izražave s Pierceovimi operatorji vodi preko MKNO. Dobljeno MDNO izrišemo v Veitchev diagram in določimo MKNO. Možnosti za minimalno združevanje sta dve:



MKNO najenostavneje določimo tako, da izražamo *negacijo* f s tem da združujemo ničle in združene člene izražamo v disjunktivni obliki, nato pa negacijo funkcije f prenesemo na drugo stran enačbe.

$$\overline{f(x_1, x_2, x_3)} = \overline{x_2 \cdot x_3 + x_1 \cdot x_3 + x_2 \cdot x_3}$$

$$f(x_1, x_2, x_3) = \overline{\overline{x_2 \cdot x_3 + x_1 \cdot x_3 + x_2 \cdot x_3}}$$

Nad dobljeno enačbo uporabimo De Morganov teorem:

$$f(x_1, x_2, x_3) = \overline{\overline{x_2 \cdot x_3 + x_1 \cdot x_3 + x_2 \cdot x_3}}$$

$$f(x_1, x_2, x_3) = \overline{(\overline{x_2 \cdot x_3}) \cdot (\overline{x_1 \cdot x_3}) \cdot (\overline{x_2 \cdot x_3})}$$

$$f(x_1, x_2, x_3) = \overline{(x_2 + x_3) \cdot (x_1 + x_3) \cdot (x_2 + x_3)}$$

Iz dobljene MKNO preidemo v Pierceovo obliko tako, da celotno funkcijo dvakrat negiramo:

$$f(x_1, x_2, x_3) = \overline{\overline{(x_2 + x_3) \cdot (x_1 + x_3) \cdot (x_2 + x_3)}}$$

$$f(x_1, x_2, x_3) = \overline{\overline{\overline{x_2 + x_3 + x_1 + x_3 + x_2 + x_3}}}$$

$$f(x_1, x_2, x_3) = (x_2 \downarrow x_3) \downarrow (x_1 \downarrow x_3) \downarrow (x_2 \downarrow x_3)$$

Preostale negacije spremenljivk izrazimo z uporabo lastnosti $\overline{\overline{x}} = x \downarrow x$:

$$f(x_1, x_2, x_3) = (x_2 \downarrow x_3) \downarrow (x_1 \downarrow x_3) \downarrow (x_2 \downarrow x_3)$$

$$f(x_1, x_2, x_3) = (x_2 \downarrow (x_3 \downarrow x_3)) \downarrow (x_1 \downarrow (x_3 \downarrow x_3)) \downarrow ((x_2 \downarrow x_2) \downarrow x_3)$$

Cenejšo rešitev dobimo preko izvedbe z MKNO.

Rahlo drugačno rešitev dobimo z združevanjem na desnem Veitchevem diagramu:

$$\overline{f(x_1, x_2, x_3)} = \overline{x_2 \cdot x_3 + x_1 \cdot x_2 + x_2 \cdot x_3}$$

$$f(x_1, x_2, x_3) = \overline{\overline{x_2 \cdot x_3 + x_1 \cdot x_2 + x_2 \cdot x_3}}$$

$$f(x_1, x_2, x_3) = (x_2 \downarrow x_3) \downarrow (x_1 \downarrow x_2) \downarrow (x_2 \downarrow x_3)$$

$$f(x_1, x_2, x_3) = (x_2 \downarrow (x_3 \downarrow x_3)) \downarrow (x_1 \downarrow (x_2 \downarrow x_2)) \downarrow ((x_2 \downarrow x_2) \downarrow x_3)$$

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
		0	X	1	0
		0	X	1	0
		X	0	X	1
		x_3			

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

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

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

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

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

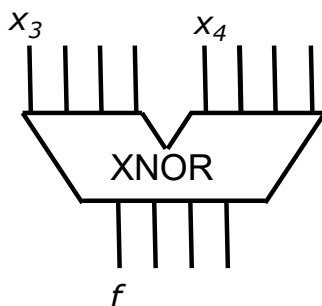
Iz tabele sledi: MNO=MDNO=MKNO.

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:

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

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'.

$$g_1(x_1, x_2, x_3) = x_1 + \overline{x_2} \cdot \overline{x_3} = x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3) + (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot \overline{x_3}$$

$$g_1(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$$

$$g_1(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$$

$$g_1(x_1, x_2, x_3) = V(4, 6, 5, 7, 0)$$

Podobno storimo še za preostale funkcije:

$$g_2 = \overline{x_1} \cdot x_3 + x_1 \cdot x_2 = \overline{x_1} \cdot (\overline{x_2} + x_2) \cdot x_3 + x_1 \cdot x_2 \cdot (\overline{x_3} + x_3)$$

$$g_2(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3$$

$$g_2(x_1, x_2, x_3) = V(1, 3, 6, 7)$$

$$g_3 = \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 =$$

$$g_3(x_1, x_2, x_3) = (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot (\overline{x_3} + x_3) =$$

$$g_3(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3$$

$$g_3(x_1, x_2, x_3) = V(0, 4, 6, 7)$$

$$g_4(x_1, x_2, x_3) = \overline{x_2} \cdot x_3 + x_1$$

$$g_4(x_1, x_2, x_3) = (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot x_3 + x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3)$$

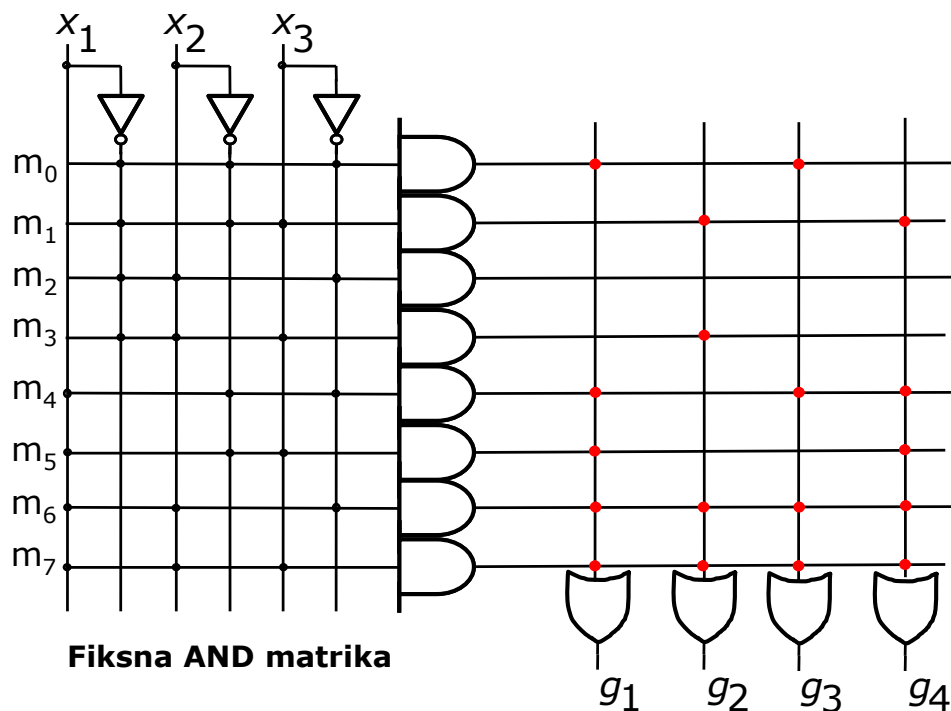
$$g_4(x_1, x_2, x_3) = (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot x_3 + x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3)$$

$$g_4(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

$$g_4(x_1, x_2, x_3) = V(1, 5, 4, 6, 7)$$

PDNO je najprimernejša oblika za realizacijo z ROM, ker je matrika AND fiksna. Programirane vrednosti AND matrike 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.



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 $2024_{24} = 27700_{10} = 6C34_{16}$ zapišemo v dvojiški obliki $0110\ 1100\ 0011\ 0100_2$. Vodilna ničla v zapisu nas ne zanima, zato jo lahko v zapisu začetne vrednosti v tabeli ne pišemo. Prve tri pomike lahko preskočimo, saj do prvega prištevanja pride šele, ko prva '1' doseže vrednost 0110_2 .

					1 1 0	1 1 0 0	0 0 1 1	0 1 0 0
				1 1 0	1 1 0	0 0 0 1	1 0 1 0	0
			1 0 0 1	1 0 0 1	1 1 0 0	0 0 0 1	1 0 1 0	0
		1 0	0 1 1 1	0 0 0 1	1 0 0 0	0 0 1 1	0 1 0 0	0
		1 0	1 0 1 0	1 0 1 0	0 0 0 0	1 1 0 1	0 1 0 0	0
		1 0 1	0 1 0 0	0 1 0 0	0 0 0 1	1 1 0 1	0 1 0 0	0
		1 0 0 0	1 0 0 0	0 1 0 0	0 0 0 1	1 1 0 1	0 1 0 0	0
		1 0 0 0	1 0 0 0	1 0 1 1	0 0 1 1	0 1 0 1	0 1 0 0	0
		1 0	0 0 0 1	0 1 1 0	0 1 1 0	0 1 0 1	0 1 0 0	0
		1 0	0 0 0 1	1 0 0 1	0 0 1 0	1 1 0 1	0 1 0 0	0
		1 0 0	0 0 1 1	0 0 1 1	0 0 0 1	0 1 0 0	0 1 0 0	0
		1 0 0 0	0 1 1 0	0 1 0 1	0 1 0 1	0 0 0 0	0 0 0 0	0
		1 0 1 1	1 0 0 1	0 0 1 1	0 0 0 1	0 1 0 0	0 1 0 0	0
		1 1 0 1	1 0 0 0	0 0 1 1	0 0 0 1	0 1 0 0	0 1 0 0	0
		1 1	0 1 0 0	0 1 1 0	0 0 1 0	1 0 0 0	1 0 0 0	0
		1 1	0 1 0 0	1 0 0 1	0 0 1 0	1 0 0 0	1 0 0 0	0
		1 1 0	1 0 0 1	0 0 1 0	0 1 0 1	0 0 0 0	0 0 0 0	0
		1 0 0 1	1 1 0 0	0 0 1 0	1 0 0 0	0 0 0 0	0 0 0 0	0
		1 0 0 1 1	1 0 0 0	0 1 0 1	0 0 0 0	0 0 0 0	0 0 0 0	0
		1 0 0 1 1	1 0 1 1	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
		1 0 0 1 1	0 1 1 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
2	7	7	0	0				

Zapis posameznih števk rezultata v BCD zapisu (27700_{10}) se glasi: 0010 0111 0111 0000 0000.

Srečno 2024_{24} ☺!