

СИНТЕЗ НА ФУНКЦИОНАЛНИ ЛОГИЧЕСКИ ВЪЗЛИ

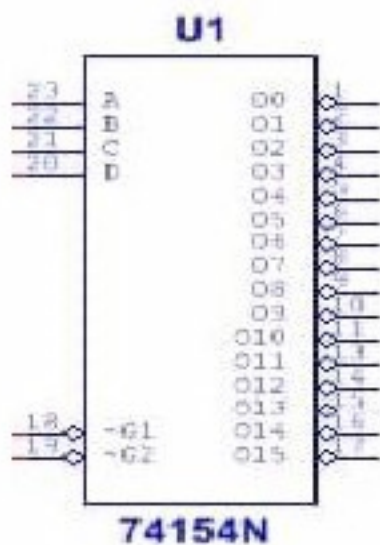
Дешифратори

Задача: Да се синтезира 6-битов (битов и двоичен е едно и също!) пълен дешифратор на базата на 4-битов дешифратор (интегрална схема 74154).

Дешифраторът преобразува n - разряден двоичен код в унитарен (позиционен) код.

Анализ на задачата:

Дадено:



4 битовият дешифратор има 4 входа ($n=4$) и 16 (2^n) изхода.

За всяка комбинация на входните сигнали е активен само един от изходите – този, чийто номер съответства на двоичната комбинация, подадена на входовете. Таблицата на истинност на 4-битовия дешифратор е следната:

$\overline{G1}$	$\overline{G2}$	A	B	C	D	Активен изход (O_i)
0	0	0	0	0	0	O_0
0	0	0	0	0	1	O_1
0	0	0	0	1	0	O_2
0	0	0	0	1	1	O_3
0	0	0	1	0	0	O_4
0	0	0	1	0	1	O_5
0	0	0	1	1	0	O_6
0	0	0	1	1	1	O_7
0	0	1	0	0	0	O_8
0	0	1	0	0	1	O_9
0	0	1	0	1	0	O_{10}
0	0	1	0	1	1	O_{11}
0	0	1	1	0	0	O_{12}
0	0	1	1	0	1	O_{13}
0	0	1	1	1	0	O_{14}
0	0	1	1	1	1	O_{15}
0	1	*	*	*	*	Няма активен изход
1	0	*	*	*	*	Няма активен изход
1	1	*	*	*	*	Няма активен изход

Логическите уравнения на изходите са (забележете, че изходите са с активно нулево ниво):

$$\begin{aligned} \overline{O_0} &= \overline{G1.G2.A \overline{B} \overline{C} \overline{D}} & \overline{O_1} &= \overline{G1.G2.A \overline{B} C \overline{D}} & \overline{O_2} &= \overline{G1.G2.A \overline{B} C D} \\ \overline{O_3} &= \overline{G1.G2.A \overline{B} C D} & \dots & & \overline{O_{15}} &= \overline{G1.G2.A B C D} \end{aligned}$$

Търси се:



Таблицата на истинност на 6-битовия дешифратор е следната:

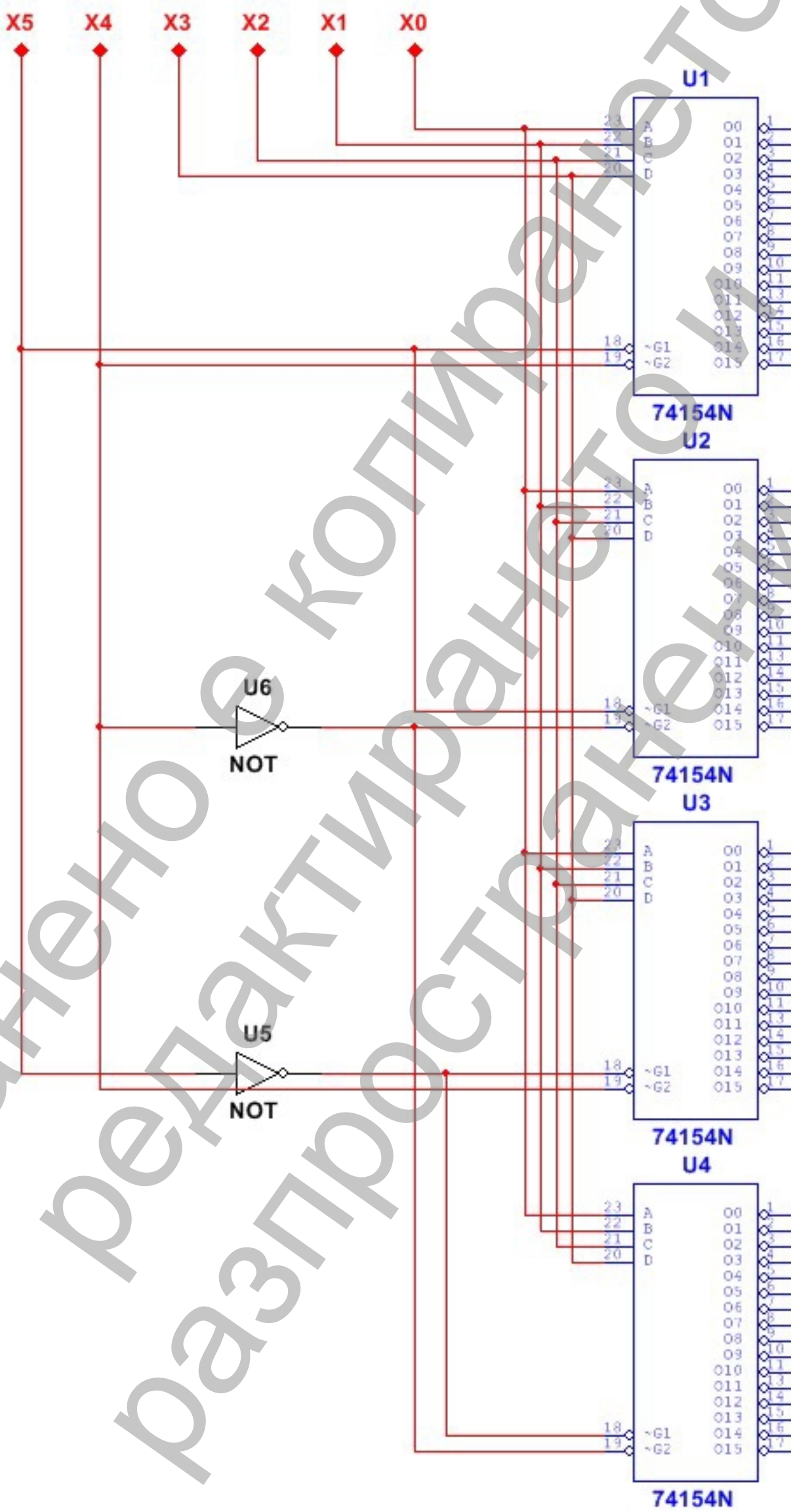
X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	Y _i
0	0	0	0	0	0	Y ₀
0	0	0	0	0	1	Y ₁
0	0	0	0	1	0	Y ₂
0	0	0	0	1	1	Y ₃
0	0	0	1	0	0	Y ₄
.....						
0	0	1	1	1	1	Y ₁₅
0	1	0	0	0	0	Y ₁₆
0	1	0	0	0	1	Y ₁₇
0	1	0	0	1	0	Y ₁₈
.....						
0	1	1	1	1	1	Y ₃₁
1	0	0	0	0	0	Y ₃₂
1	0	0	0	0	1	Y ₃₃
1	0	0	0	1	0	Y ₃₄
.....						
1	0	1	1	1	1	Y ₄₇
1	1	0	0	0	0	Y ₄₈
1	1	0	0	0	1	Y ₄₉
1	1	0	0	1	0	Y ₅₀

.....						
1	1	1	1	1	1	Y_{63}

Ако се абстрахираме от колоните за променливите x_5 и x_4 и съпоставим А, В, С и D съответно с x_3 , x_2 , x_1 и x_0 , виждаме, че таблицата на истинност на 4-битовия дешифратор се повтаря четири пъти. Следователно за решението на тази задача ще са необходими поне четири 4-битови дешифратора.

Тези четири 4-битови дешифратора ще дешифрират едновременно четири еднакви входни комбинации на променливите x_3 , x_2 , x_1 и x_0 , т. е. вместо един активен изход, както е по дефиниция, ще има четири едновременно активни изхода. Кой точно от дешифраторите трябва да работи, може да се избере с комбинацията x_5 , x_4 . Ако трябва да се дешифрират комбинациите от 0 до 15 включително, трябва да работи първият дешифратор. Това ще се случи, ако на разрешаващите входове G_1 и G_2 има подадена комбинация 00. Но тази комбинация е същата като комбинацията на входните променливи x_5 и x_4 на търсения дешифратор. Следователно работата на първия дешифратор ще бъде разрешена при $\overline{G_1} \cdot \overline{G_2} = \overline{x_5} \cdot \overline{x_4} = 00$. Аналогично определяме, че вторият дешифратор ще работи при $\overline{G_1} \cdot G_2 = \overline{x_5} \cdot x_4 = 01$; третият – при $G_1 \cdot \overline{G_2} = x_5 \cdot \overline{x_4} = 10$ и четвъртият – при $G_1 \cdot G_2 = x_5 \cdot x_4 = 11$.

Схемно решение:



Забранено редактиране! Прочетете!

Мултиплексорът като функционално пълен базис за синтез на комбинационни логически схеми

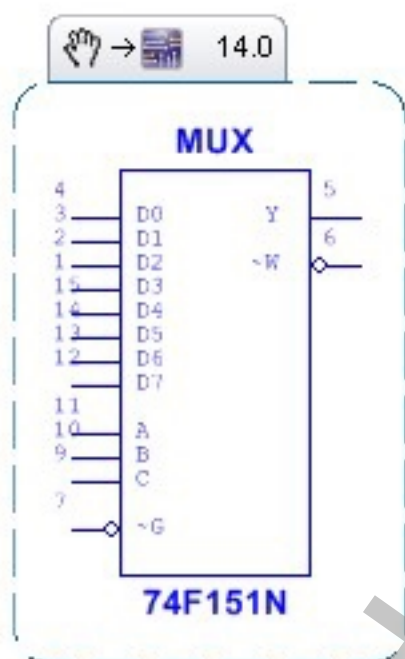
Задача 1: Да се синтезира комбинационна логическа схема с 4 входа и един изход, чиято работа се задава със следната функция:

$$f = \sum m(1, 2, 7, 9, 12, 14, 15)^1.$$

Схемата да се реализира с мултиплексор 8:1 (интегрална схема 74151).

Мултиплексор 8:1 има 8 информационни входа, 3 адресни входа и 1 изход.

Таблицата на истинност на мултиплексор 8:1 е:



G	A ₂ (C)	A ₁ (B)	A ₀ (A)	Y
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7
1	*	*	*	0

Дадената функция се състои от четири променливи, поради тази причина трябва да се избере една от променливите да бъде даннова (да се подава на данновите входове), а другите три да бъдат подавани на адресните входове на мултиплексора. Стъпките за изпълнение при синтеза на схема с мултиплексор 8:1, при функция с четири променливи са следните:

1. Съставя се Карта на Карно (КК) за дадената функция.
2. Избира се коя променлива да бъде даннова (кои да бъдат адресни).
3. Изразяват се данновите входове чрез тази променлива.
4. Синтезира се схемата.

I вариант на решение на задачата

1. Карта на Карно

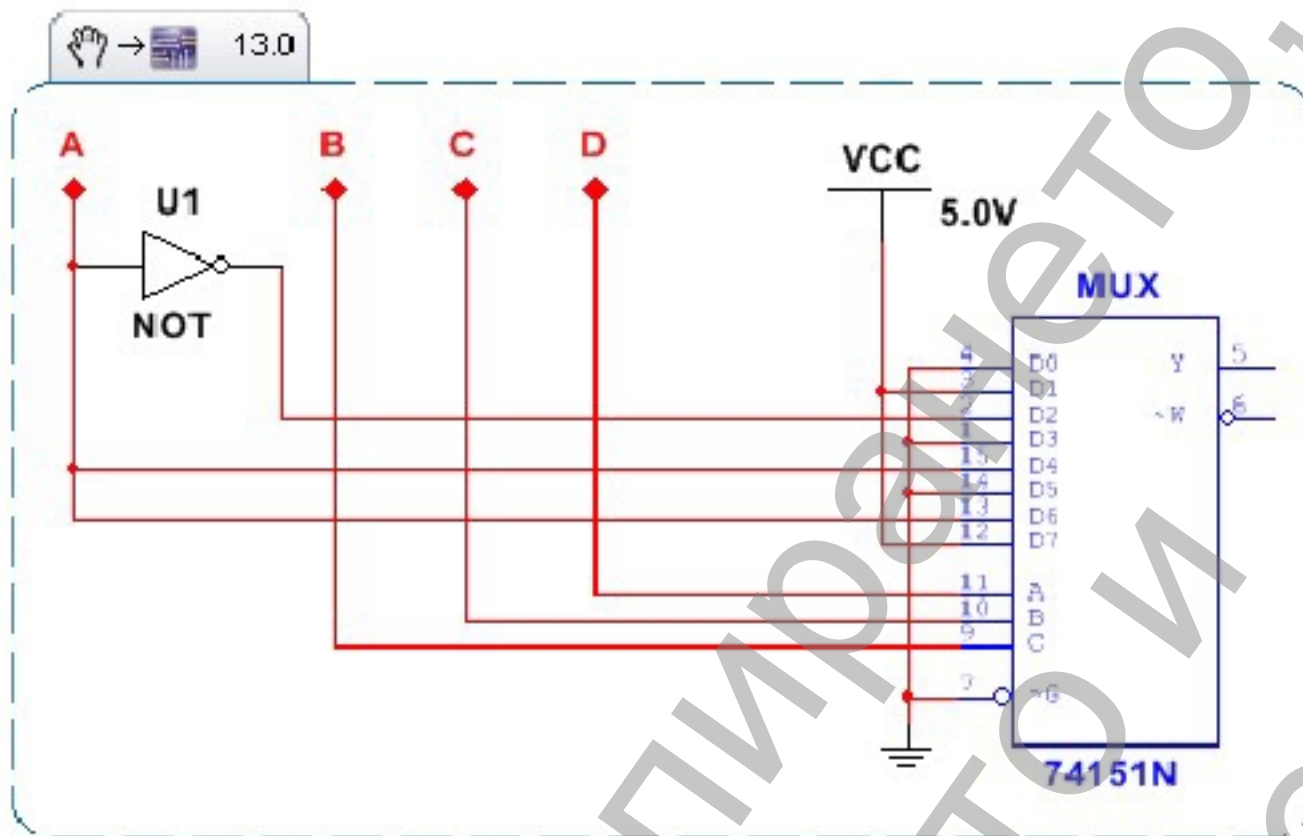
	D0	D1	D3	D2	
AB\CD	00	01	11	10	
00	0	1	0	1	D7
01	0	0	1	0	D5
11	1	0	1	1	D4
10	0	1	0	0	D6

2. За даннова се избира променлива A (B,C,D – адресни променливи). Това означава, че е необходимо да се осъществи групиране на клетките в картата на Карно според повтарящите се комбинации от стойности на адресните променливи.

3. Изразяване на данновите входове чрез избраната адресна променлива.

BCD	Даннов вход
000	D0 = 0
001	D1 = 1
010	D2 = \bar{A}
011	D3 = 0
100	D4 = A
101	D5 = 0
110	D6 = A
111	D7 = 1

4. Схемна реализация



II вариант на решение на задачата

1. Карта на Карно

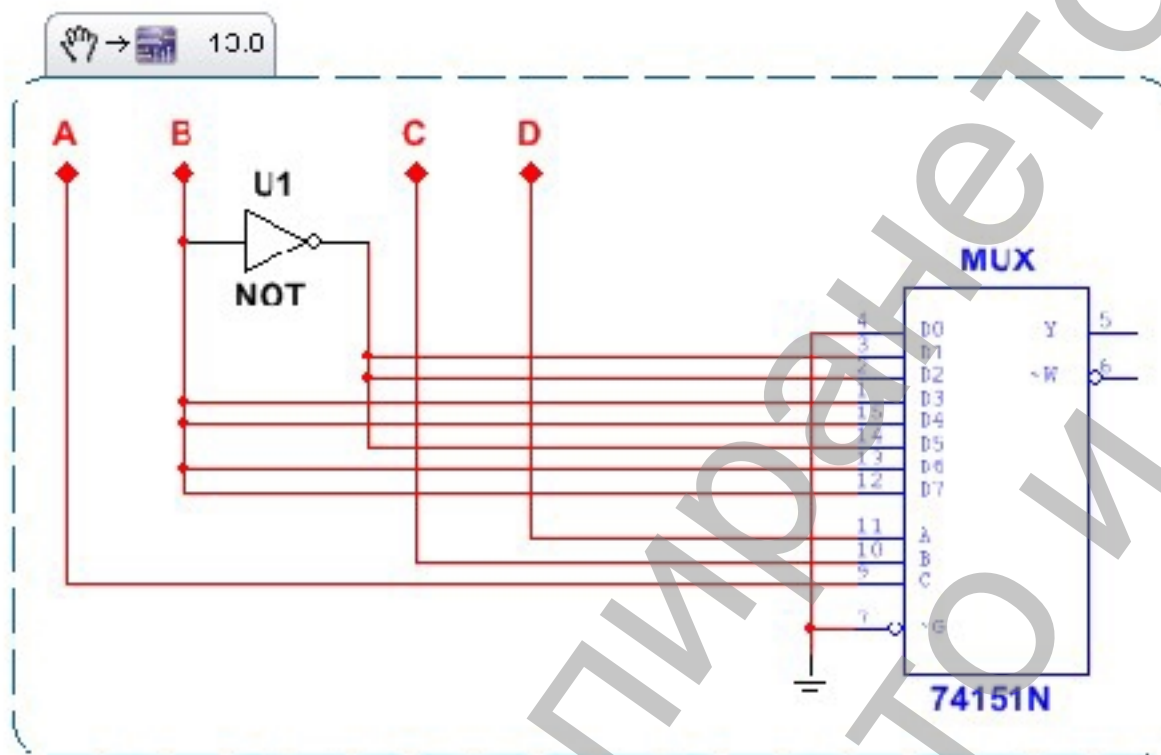
	D0	D1	D3	D2
AB\CD	00	01	11	10
00	0	1	0	1
01	0	0	1	0
11	1	0	1	1
10	0	1	0	0
	D4	D5	D7	D6

2. За даннова се избира променлива B (ACD – адресни променливи). Това означава, че е необходимо да се осъществи групиране на клетките в картата на Карно според повтарящите се комбинации от стойности на адресните променливи.

3. Изразяване на данновите входове спрямо избраната адресна променлива.

ACD	Даннов вход
000	D0 = 0
001	D1 = \bar{B}
010	D2 = \bar{B}
011	D3 = B
100	D4 = B
101	D5 = \bar{B}
110	D6 = B
111	D7 = B

4. Схемна реализация



III вариант на решение на задачата

1. Карта на Карно

	AB\CD	00	01	11	10
D0	00	0	1	0	1
D2	01	0	0	1	0
D6	11	1	0	1	1
D4	10	0	1	0	0

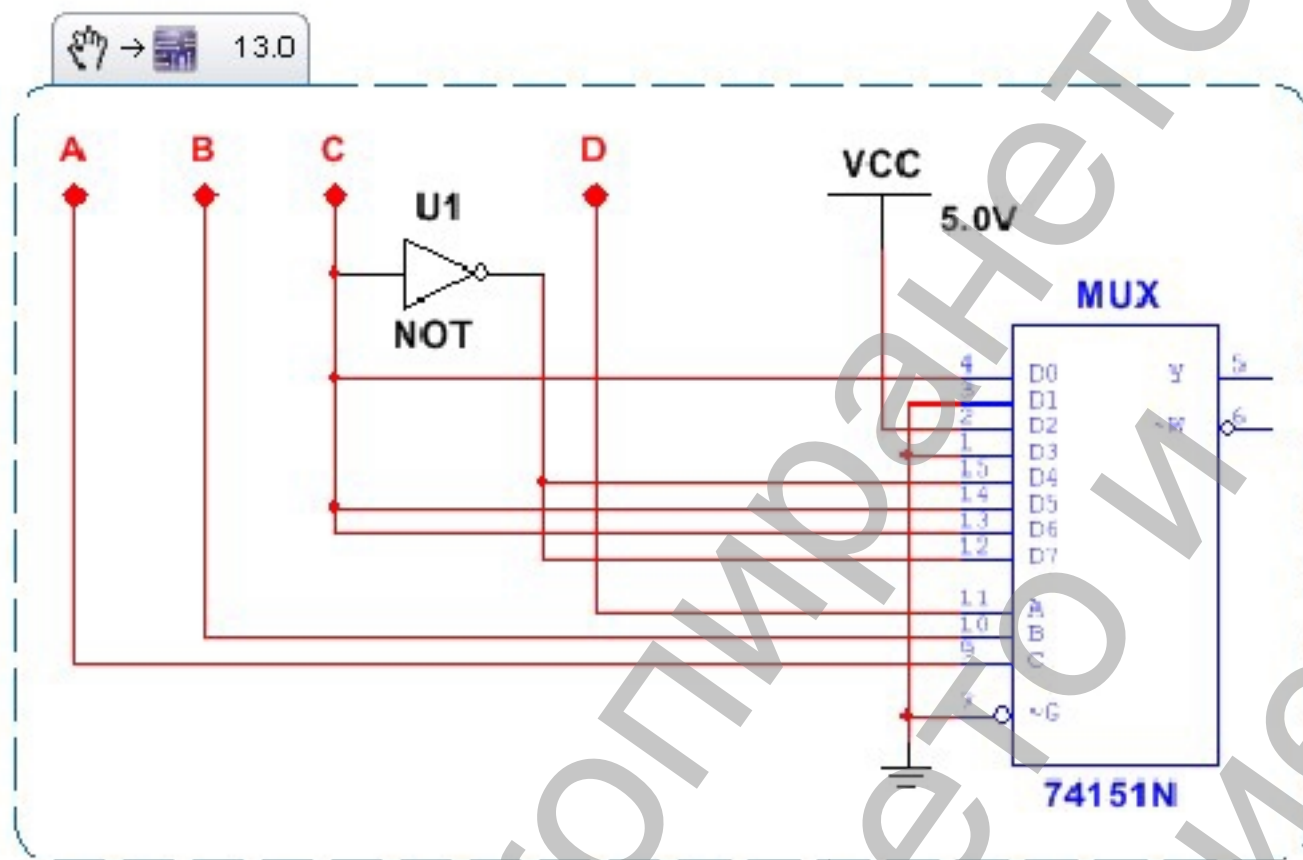
Arrows indicate the mapping of the 4-bit combinations to the 8 data inputs D0 through D7 of the MUX.

2. За даннова се избира променлива C (ABD – адресни променливи). Това означава, че е необходимо да се осъществи групиране на клетките в картата на Карно според повтарящите се комбинации от стойности на адресните променливи.

3. Изразяване на данновите входове спрямо избраната адресна променлива.

ABD	Даннов вход
000	D0 = 0
001	D1 = 1
010	D2 = \bar{A}
011	D3 = 0
100	D4 = A
101	D5 = 0
110	D6 = A
111	D7 = 1

4. Схемна реализация



IV вариант на решение на задачата

1. Карта на Карно

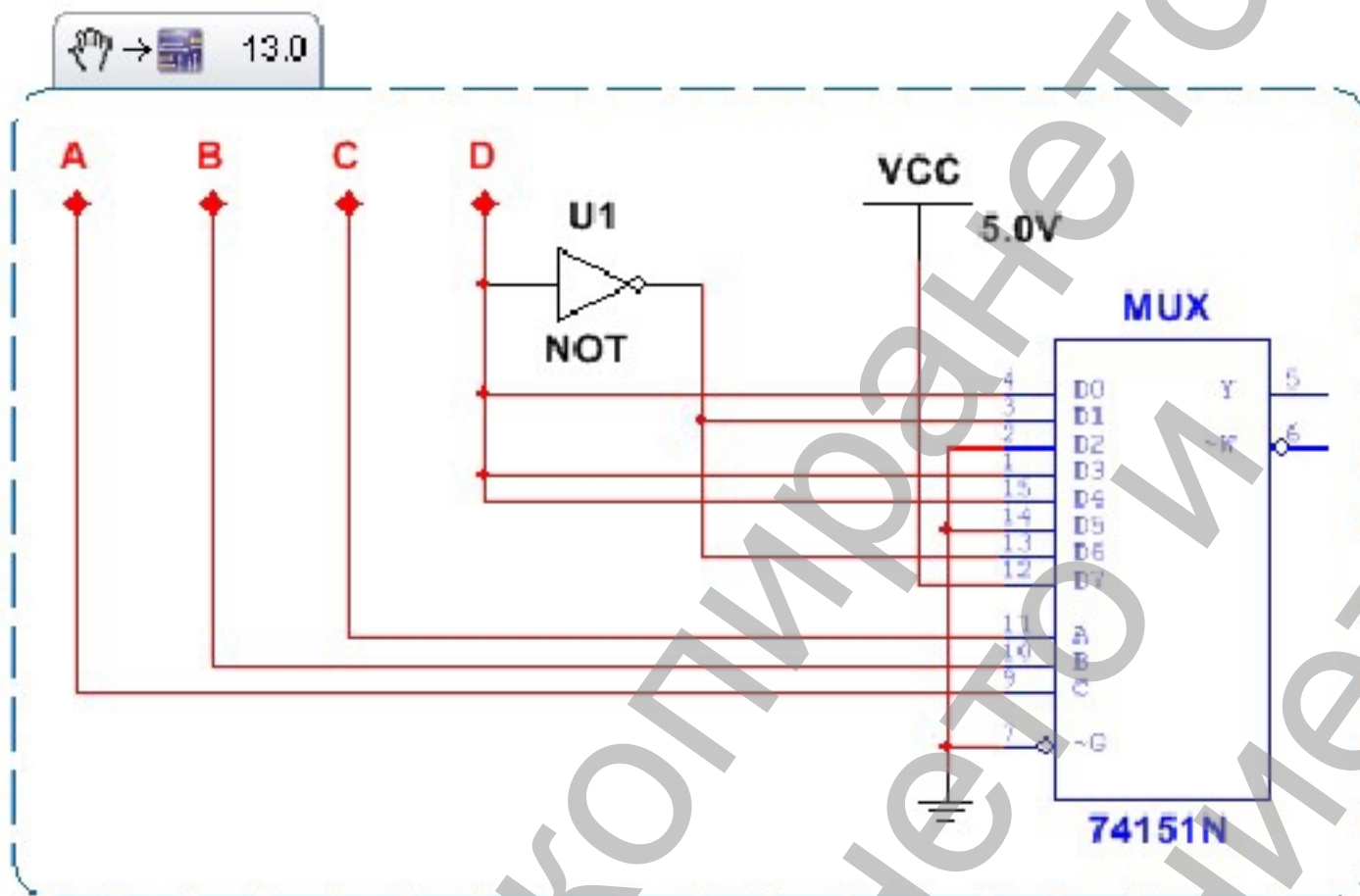
AB\CD	00	01	11	10
D0 ← 00	0	1	0	1
D2 ← 01	0	0	1	0
D6 ← 11	1	0	1	1
D4 ← 10	0	1	0	0
				D5 →

2. За даннова се избира променлива D (ABC – адресни променливи). Това означава, че е необходимо да се осъществи групиране на клетките в картата на Карно според повтарящите се комбинации от стойности на адресните променливи.

3. Изразяване на данновите входове спрямо избраната адресна променлива.

ABC	Даннов вход
000	D0 = D
001	D1 = \bar{D}
010	D2 = 0
011	D3 = D
100	D4 = D
101	D5 = 0
110	D6 = \bar{D}
111	D7 = 1

4. Схемна реализация



Задача 2: В стая има 3 сензора за движение и 1 ел. ключ за хранване. Осветлението е включено, само ако поне два от датчиците засекаат движение и има хранване. Да се синтезира схема за управление на осветлението с използване на МХ 8:1.

Анализ на задачата:

За да синтезираме логическата схема, е необходимо да представим функцията, която тя реализира, с таблица на истинност.

Очевидно променливите в тази задача ще са трите сензора и ключът за хранване (т.е. общо 4 променливи), но дали всички са с еднакъв приоритет?

Избираме трите сензора да бъдат с еднакъв приоритет, а хранващият ключ да бъде с най-висок приоритет, затова ще го бележим с променлива А, а трите датчика – с В, С и D.

След като определихме променливите, следва да представим функцията с таблица на истинност.

A	B	C	D	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

- **A=0** ще означава, че няма захранване, следователно **f=0** за първите 8 набора;
- **A=1** ще означава, че има захранване, но **f=1** само когато два от датчиците засекат движение т.е. са единици.

Таблицата на истинност може да бъде представена и по-кратко:

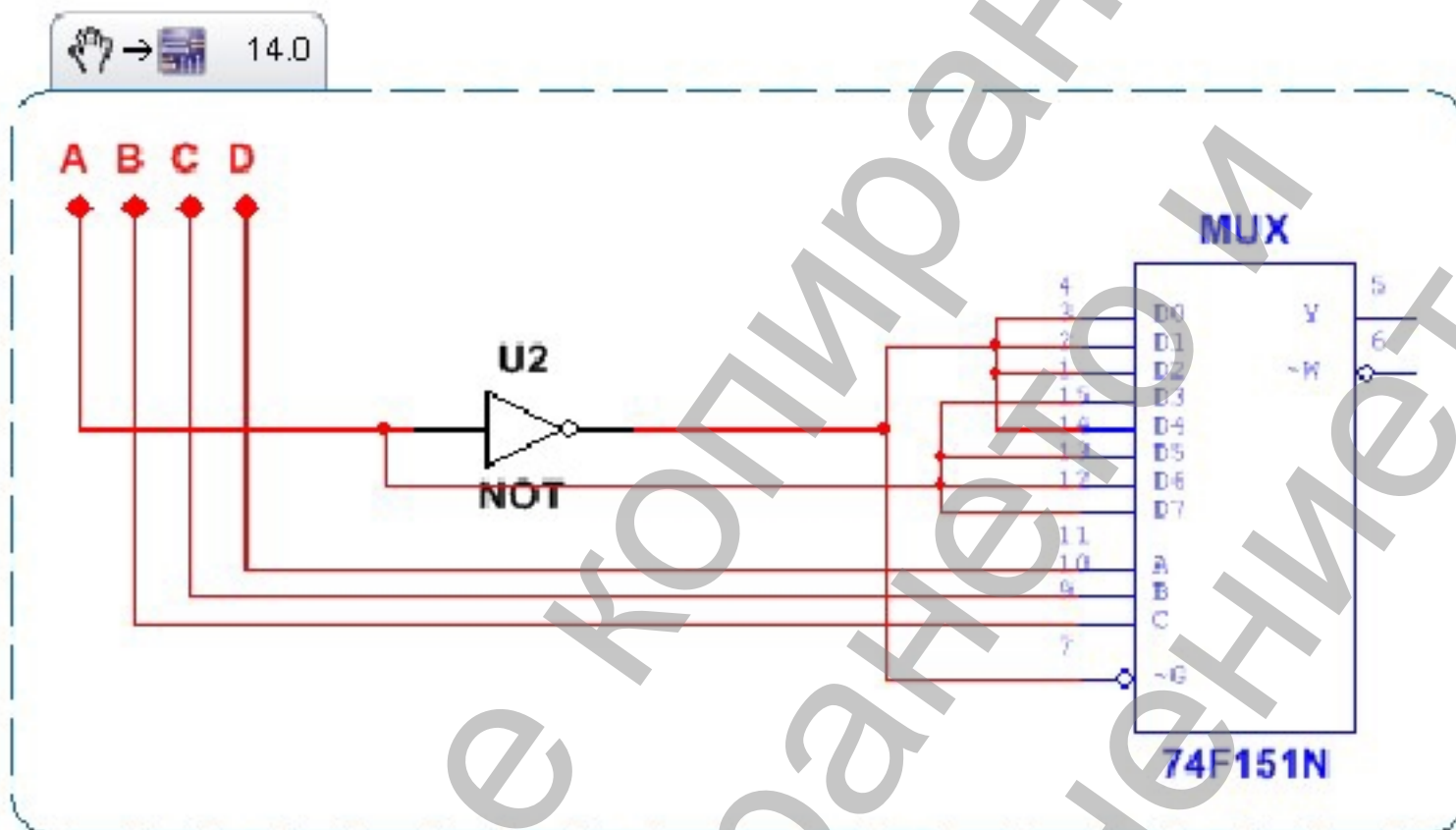
A	B	C	D	f
0	*	*	*	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

! Решение

Ако съпоставим ТИ на мултиплексора и ТИ на логическата функция (съкратения вариант), ще видим, че A може да бъде избрана за разрешаваща променлива, а B, C и D – за адресни. Налагайки една върху друга двете таблици на истинност, получаваме логическите константи,

които трябва да бъдат подадени към всеки от данните входове на мултиплексора, за да работи той съгласно зададената функция.

Схемна реализация



II Решение

Възможно е задачата да се реши и с мултиплексор 4:1 (интегрална схема 74153 – два мултиплексора 4:1), като променливата A отново е разрешаваща, а от останалите три (BCD) една се избира за даннова, а другите две - за адресни. Тогава трябва да се използва и карта на Карно.

Избира се даннова променлива на бъде променлива B.

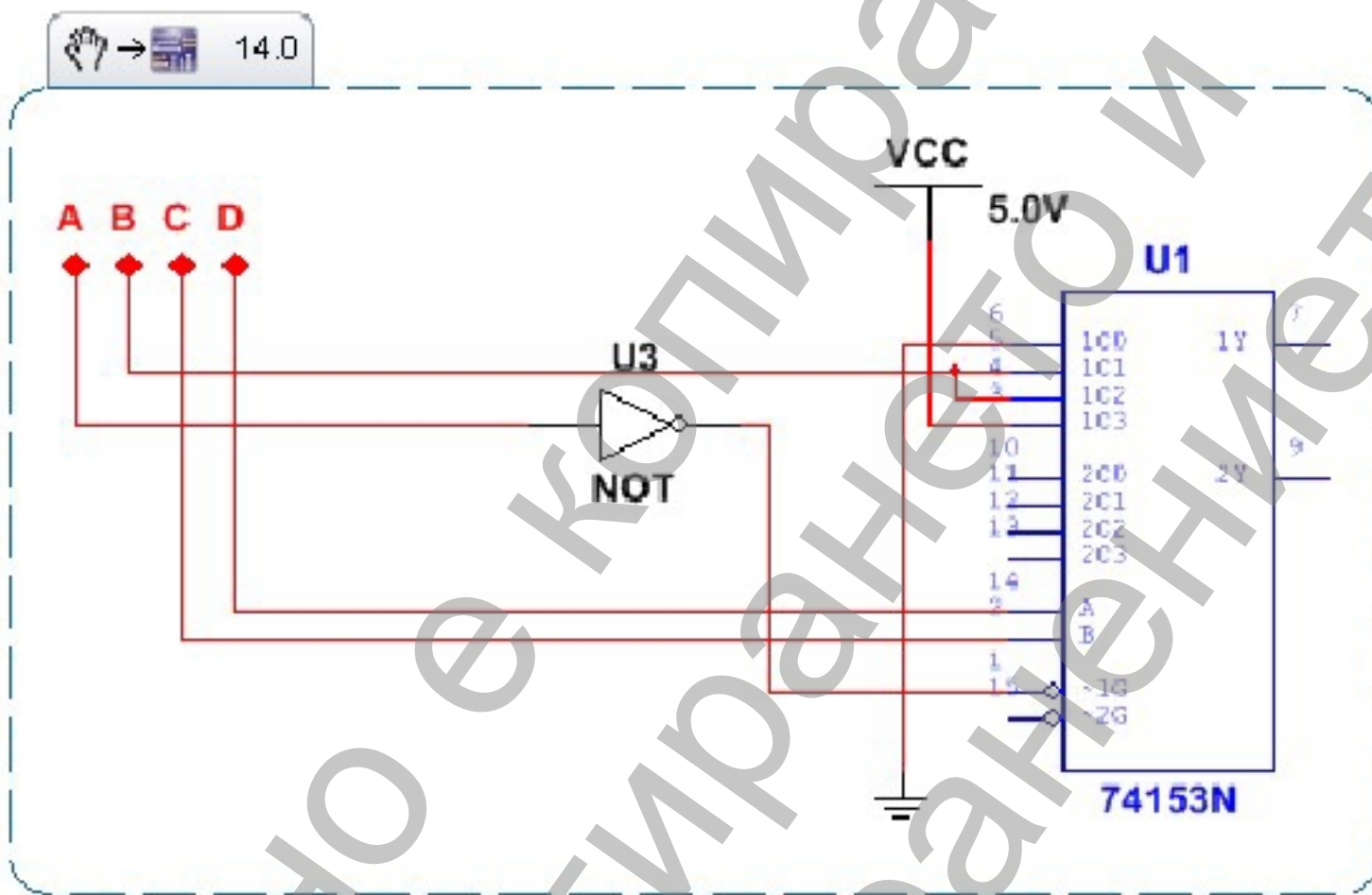
B \ CD	00	01	11	10
0	0	0	1	0
1	0	1	1	1

↓ D0
 ↓ D1
 ↓ D3
 ↓ D2

Изразяване на данните входове спрямо избраната адресна променлива за всяка комбинация от стойности на адресните променливи.

CD	Даннов вход
00	D0 = 0
01	D1 = B
10	D2 = B
11	D3 = 1

Схемна реализация



Могат да се реализират и другите 2 комбинации от адресни променливи и даннова променлива: BC – адресни, D – даннова; BD – адресни, C – даннова. Това можете да направите самостоятелно.