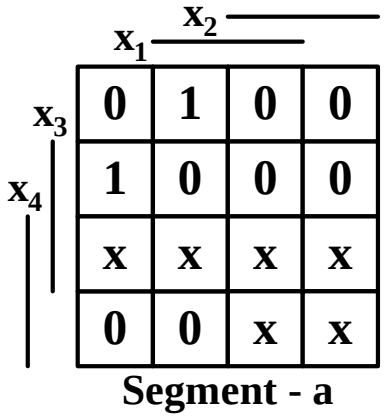
NÁVRH DEKODÉRU BCD->7SEGMENT

Ukázka návrhu kombinačního obvodu (dekodéru) všemi obvody, s kterými lze LKO realizovat. Na displeji chceme pro číslo (x4,x3,x2,x1) z intervalu <0,9> zobrazit následující symboly.

Před vytvořením Karnaughových map musíme určit zda rosvícení segmentu bude zajišťovat log.0 nebo log.1. Předpokládejme segmenty se společnou anodou (svítí=log.0).



Pro realizaci obvody NOR vycházíme ze součinové formy

$$a = (x_1 + x_3).(\overline{x}_2).(x_4).(\overline{x}_1 + \overline{x}_3)$$

kterou budeme realizovat dvoustupňovou formou

NÁVRH DEKODÉRU BCD->7SEGMENT

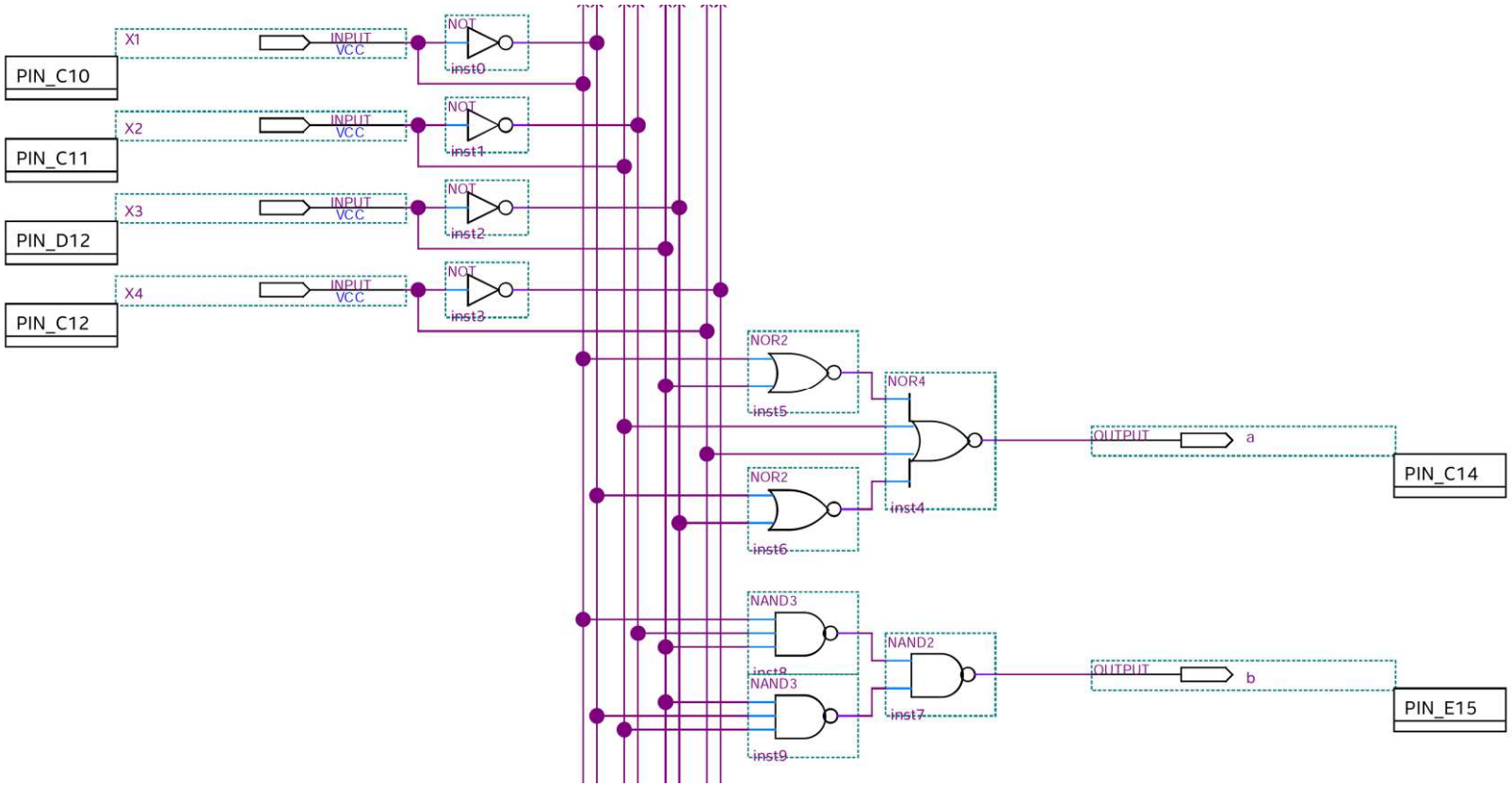
x_4 \ x_3	x_1 \ x_2	00	01	11	10
00		0	0	0	0
01		0	1	0	1
11		x	x	x	x
10		0	0	x	x

Segment - b

Pro realizaci obvody NAND vycházíme ze součtové formy

$$b = x_1.\overline{x}_2.x_3 + \overline{x}_1.x_2.x_3$$

kterou budeme realizovat také dvoustupňovou formou



NÁVRH DEKODÉRU BCD->7SEGMENT

x_3 \ x_1 x_2	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	x	x	x	x
10	0	0	x	x

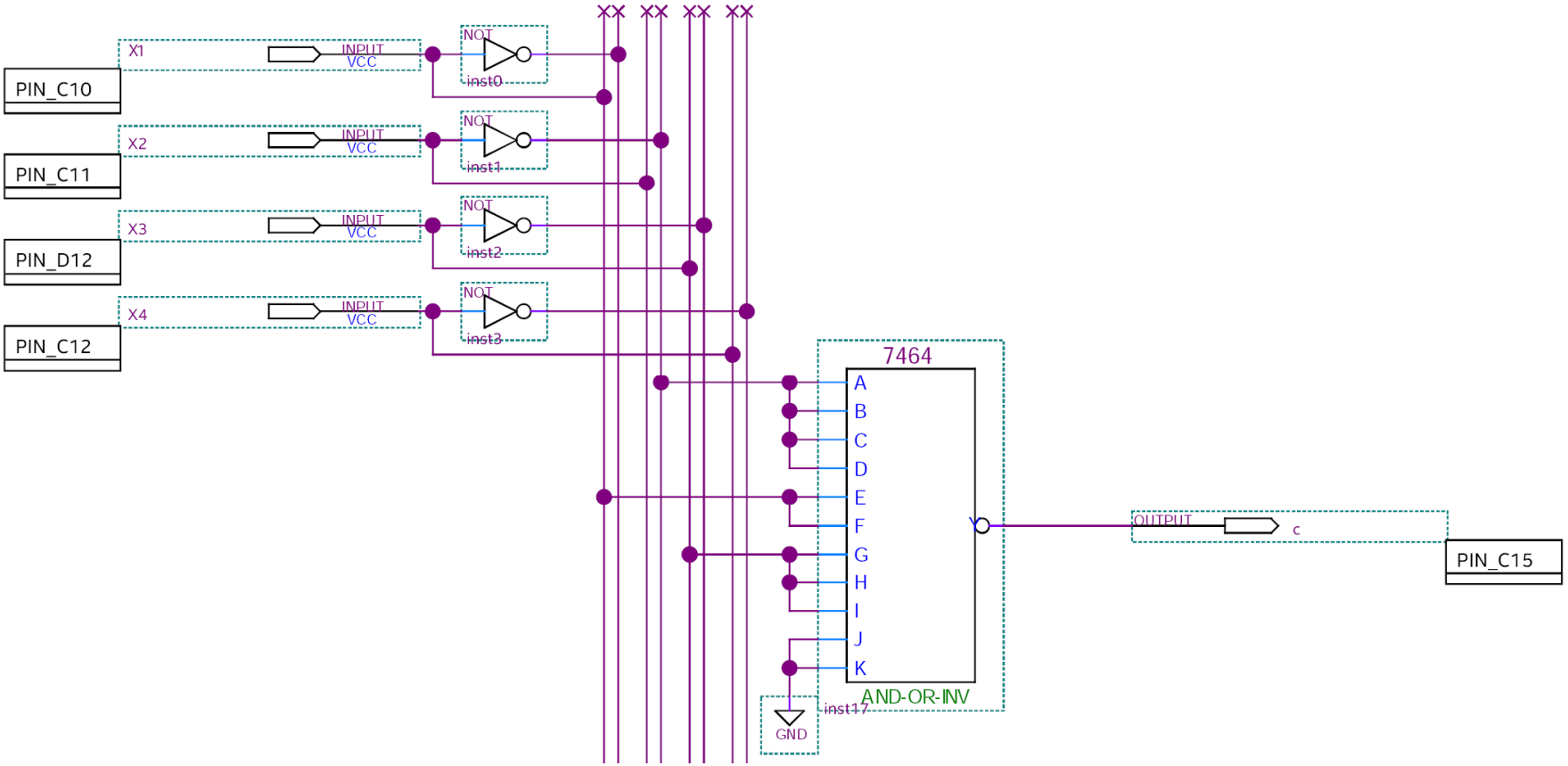
x_4

Segment - c

Pro realizaci obvody AND-OR-INVERT potřebujeme součtovou formu z negace funkce

$$c = \overline{\overline{x}_2 + x_1 + x_3}$$

Z obvodů, které jsou k dispozici, nám pro jednostupňovou formu vyhoví obvod 7464 .

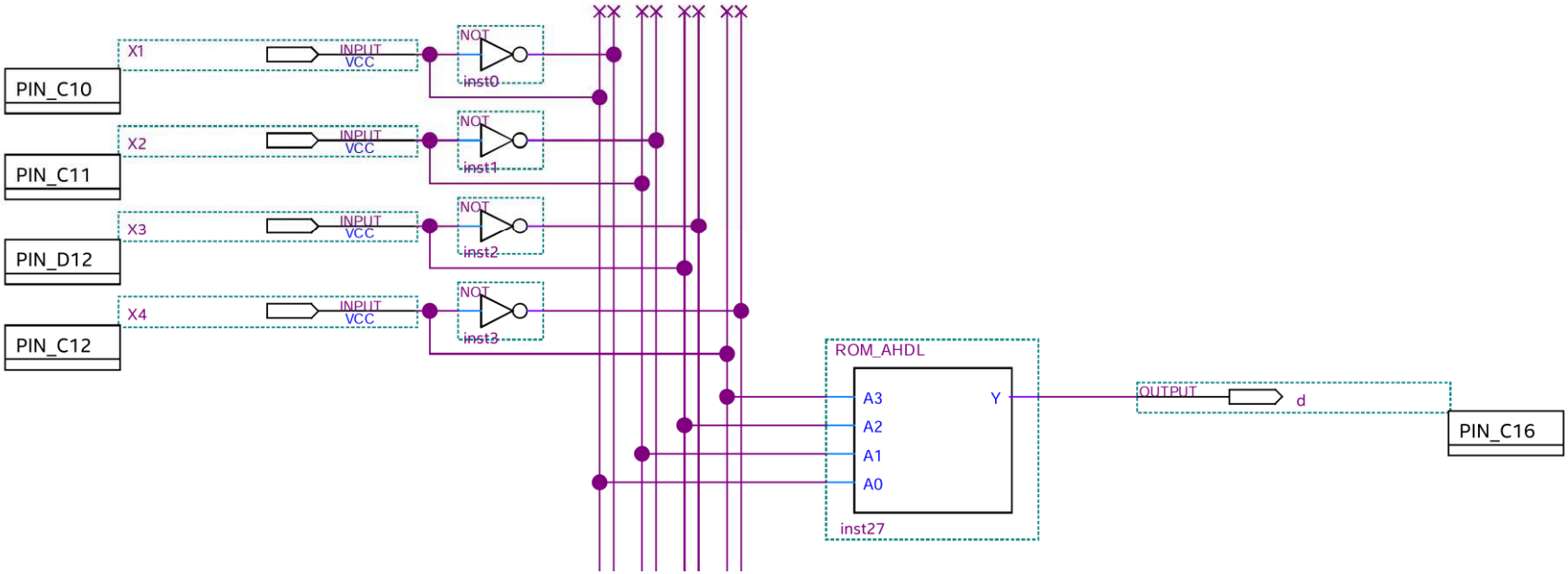


NÁVRH DEKODÉRU BCD->7SEGMENT

x_4 \ x_3 \ x_1 x_2				
	0	1	0	0
	1	0	1	0
	x	x	x	x
	0	0	x	x

Segment - d

Při realizaci pamětí ROM v podstatě realizujeme pravdivostní tabulku, z které vývojový systém vytvoří LKO. Tabulku můžeme zapsat v jazyce AHDL nebo VHDL. Připojení vstupních proměnných na adresové vodiče je libovolné. Je vhodné připojení udělat tak, aby se nám snadno určovalo příslušné paměťové místo.



NÁVRH DEKODÉRU BCD->7SEGMENT

```
SUBDESIGN ROM_AHDL
(   A3,A2,A1,A0        : INPUT;
    Y                  : OUTPUT; )
BEGIN
       TABLE
                A[3..0]       =>          Y;
                H"0"          =>          0;
                H"1"          =>          1;
                H"2"          =>          0;
                H"3"          =>          0;
                H"4"          =>          1;
                H"5"          =>          0;
                H"6"          =>          0;
                H"7"          =>          1;
                H"8"          =>          0;
                H"9"          =>          0;
                H"A"          =>          1;
                H"B"          =>          1;
                H"C"          =>          1;
                H"D"          =>          1;
                H"E"          =>          1;
                H"F"          =>          1;
        END TABLE;
 END;
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity ROM16x1 is
port (address : in INTEGER range 0 to 15;
      data : out std_logic);
end entity ROM16x1;

architecture lko of ROM16x1 is
type rom_array is array (0 to 15) of std_logic;
constant rom : rom_array :=
    ('0','1',
     '0','0',
     '1','0',
     '0','1',
     '0','0',
     '1','1',
     '1','1',
     '1','1');
     begin
        data <= rom(address);
end architecture lko;
```

NÁVRH DEKODÉRU BCD->7SEGMENT

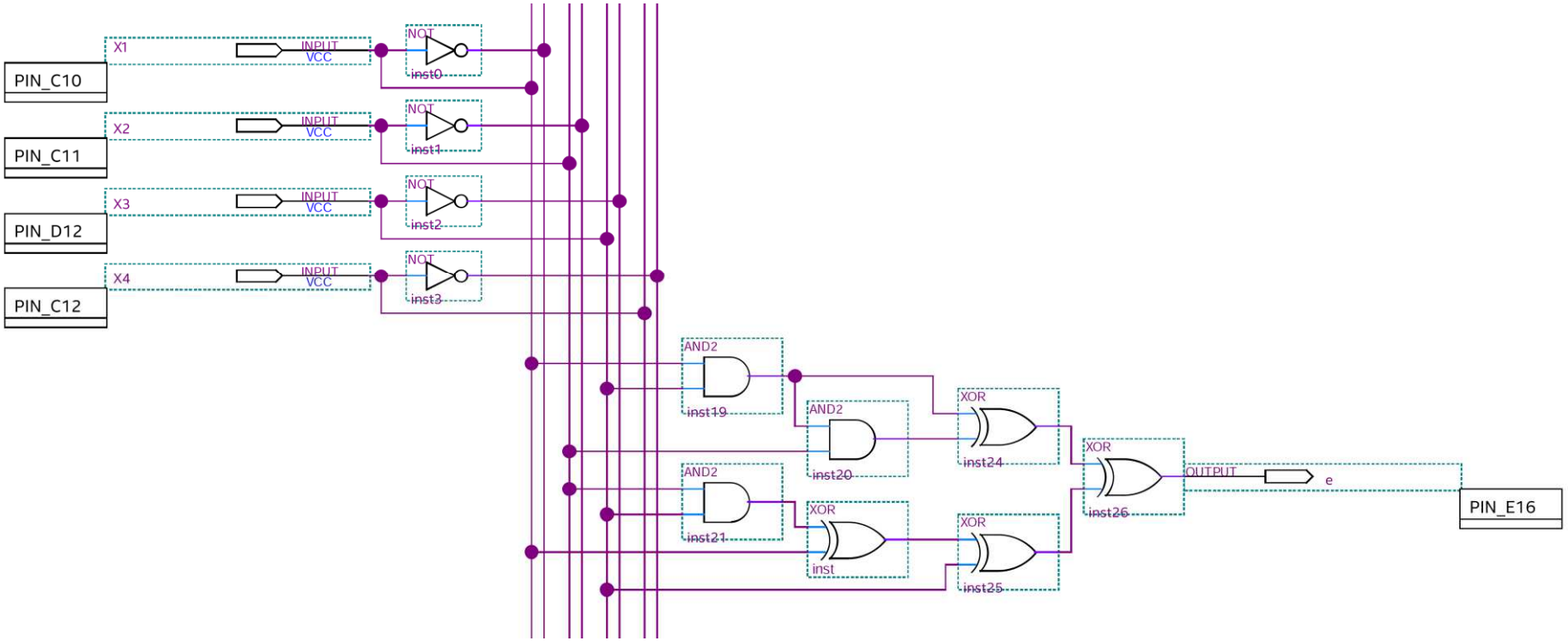
x_4 \ x_3 \ x_1 x_2				
	0	1	1	0
	1	1	1	0
	x	x	x	x
	0	1	x	x

Segment - e

Realizace obvody AND a EX-OR. V úplné součtové formě nahradíme log. součet operací EX-OR a invertované proměnné x_j nahradíme $(1 \oplus x_j)$.

$$e = x_1 \oplus \bar{x}_1.\bar{x}_2.x_3 = x_1 \oplus (1 \oplus x_1).(1 \oplus x_2).x_3 =$$

$$x_1 \oplus x_3 \oplus x_1.x_3 \oplus x_2.x_3 \oplus x_1.x_2.x_3$$



NÁVRH DEKODÉRU BCD->7SEGMENT

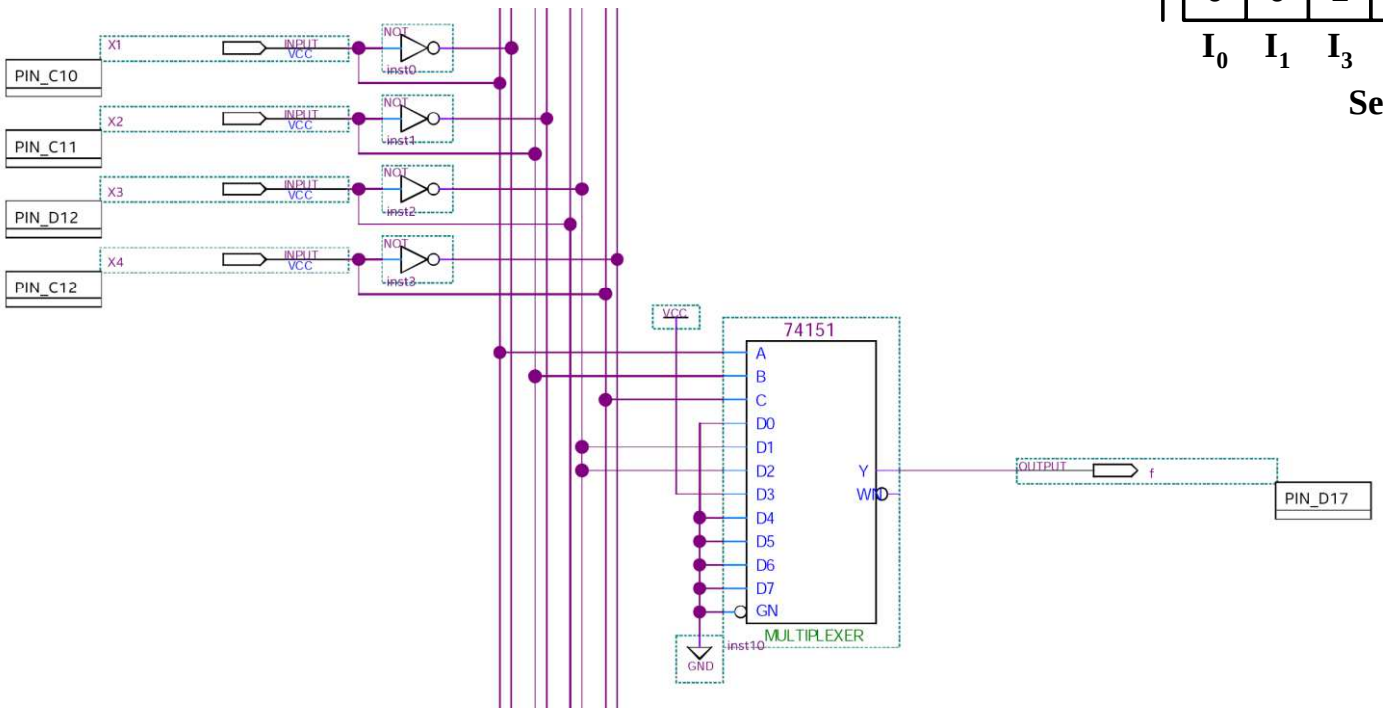
Realizace LKO multiplexorem vychází z úplné součtové formy, kde algebraicky vytkneme výrazy N-1 proměnných. V závorkách zbude 1 proměnná s funkčními hodnotami (odpovídá sloupci u druhé mapy). Přiřazení proměnných k adresovacím vstupům MUX ovlivňuje přiřazení hodnot ze sloupce druhé mapy k jednotlivým vstupům I_j.

x_3 \ x_1 x_2	00	01	11	10
0	0	1	1	1
0	0	0	1	0
1	x	x	x	x
1	0	0	x	x

Segment - f

x_3 \ x_1 x_2 x_4	000	001	011	010	110	111	101	100
0	0	1	1	1	x	x	0	0
1	0	0	1	0	x	x	x	x
	I_0	I_1	I_3	I_2	I_6	I_7	I_5	I_4

Segment - f

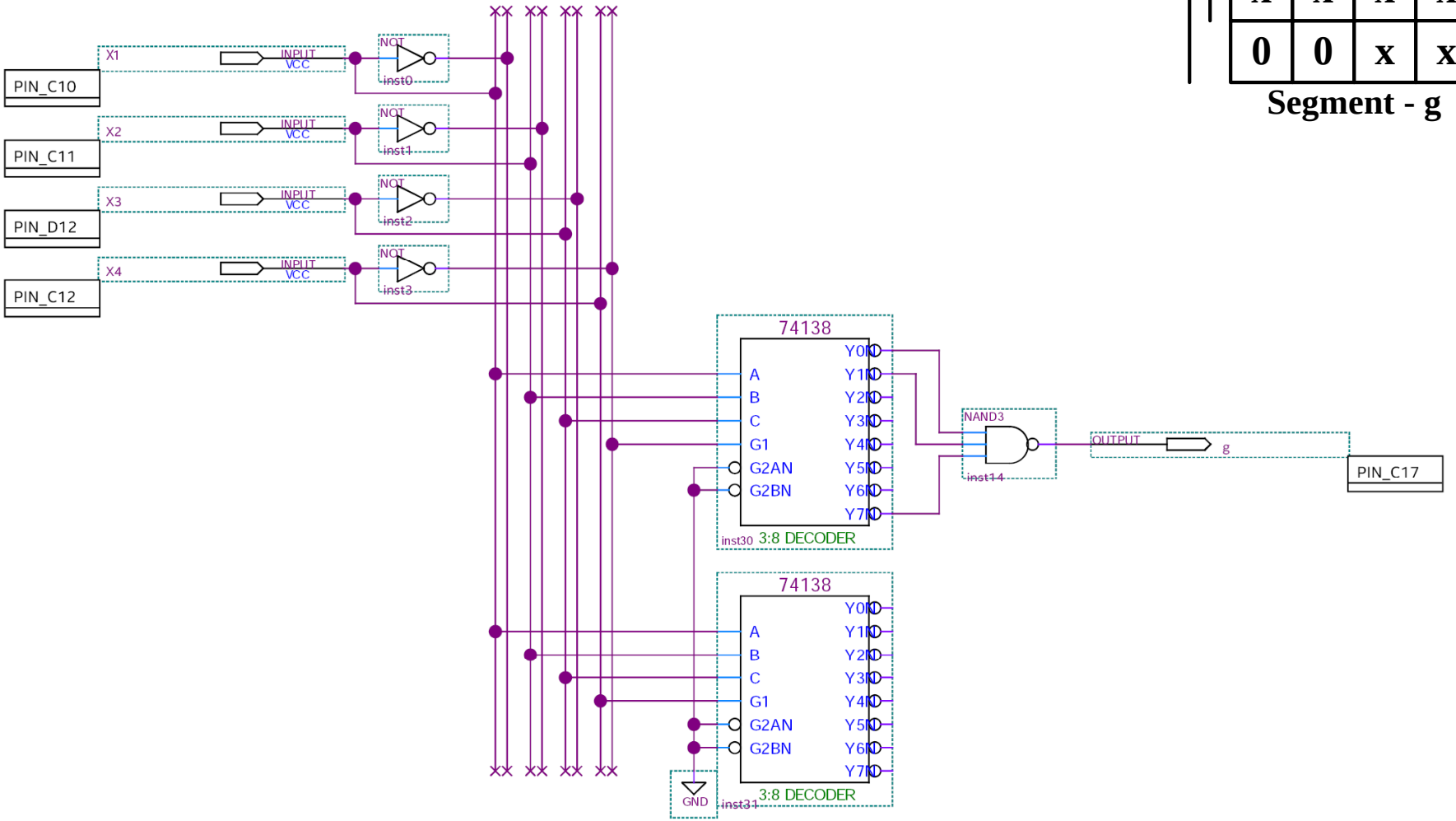


NÁVRH DEKODÉRU BCD->7SEGMENT

Realizace LKO dekodérem v podstatě představuje jiný způsob realizace pravdivostní tabulky. Vstupní proměnné mohou být přiřazeny na vstupy dekodéru libovolně. Druhý dekodér je pro tuto mapu nevyužitý.

x_4 \ x_3 \ x_2 \ x_1				
	1	1	0	0
	0	0	1	0
	x	x	x	x
	0	0	x	x

Segment - g



NÁVRH DEKODÉRU BCD->7SEGMENT

Dekodér BCD->7segment byl navržen všemi možnými způsoby realizace LKO. Funkční simulací bylo ověřeno chování navrženého dekodéru.