

<b>Předmět</b>	MLOS - LLOS - Logické systémy	
<b>Ústav</b>	ÚAMT	
<b>Úloha č. 3</b>	Návrh dekodéru BCD kódu na 7-segmentový displej, kombinační logika	
<b>Student</b>		

## Cíle

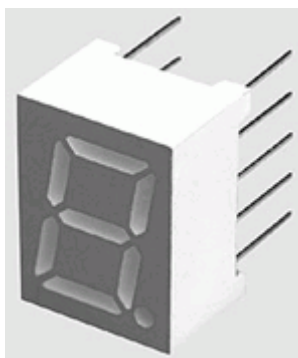
- Práce se 7-segmentovým displejem.
- Návrh kombinační logiky dekodéru pro 7-segmentový displej.
- Minimalizace logické funkce.
- Popis dekodéru pro 7-segmentový displej v jazyce VHDL v návrhovém prostředí Xilinx ISE WebPack.
- Implementace návrhu do cílového obvodu FPGA Spartan6 na vývojové desce NEXYS3.

## Teoretický úvod

Pro zobrazení výstupních dat se využívají zobrazovací jednotky (displeje). Nejčastěji jsou to LED, 7-segmentové displeje, LCD a další. Každý z těchto zobrazovačů vyžaduje individuální přístup při návrhu digitální řídicí logiky. Této digitální řídicí logice se často říká dekodér nebo zobrazovací řadič. Dekodér je digitální obvod složený pouze ze základních logických hradel NAND, NOR atd. a je to tedy kombinační obvod.

Pro návrh dekodéru je potřeba přesně znát specifikaci dané zobrazovací jednotky a podle ní navrhnout kombinační řídicí logiku. Na trhu je možné najít dekodéry v integrované podobě od několika výrobců.

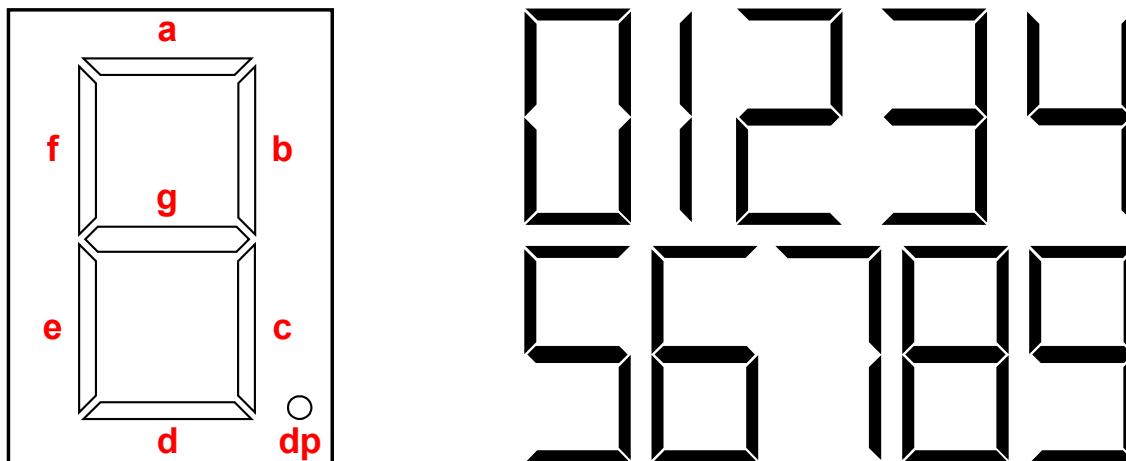
Na obrázku 1 je součástka 7-segmentového LED displeje.



Obr. 1: 7-segmentový LED displej

Na obrázku 2 je zobrazeno symbolické značení 7-segmentového displeje a ukázka zobrazení číslic 0 až 9. Písmena **a,b,c,d,e,f,g** představují jednotlivé segmenty a **dp** představuje tečku (dot point).

**Tečkou **dp** se v rámci této laboratorní úlohy nebudeme zabývat! Bude se navrhovat pouze kombinační logika pro segmenty **a,b,c,d,e,f,g**.**

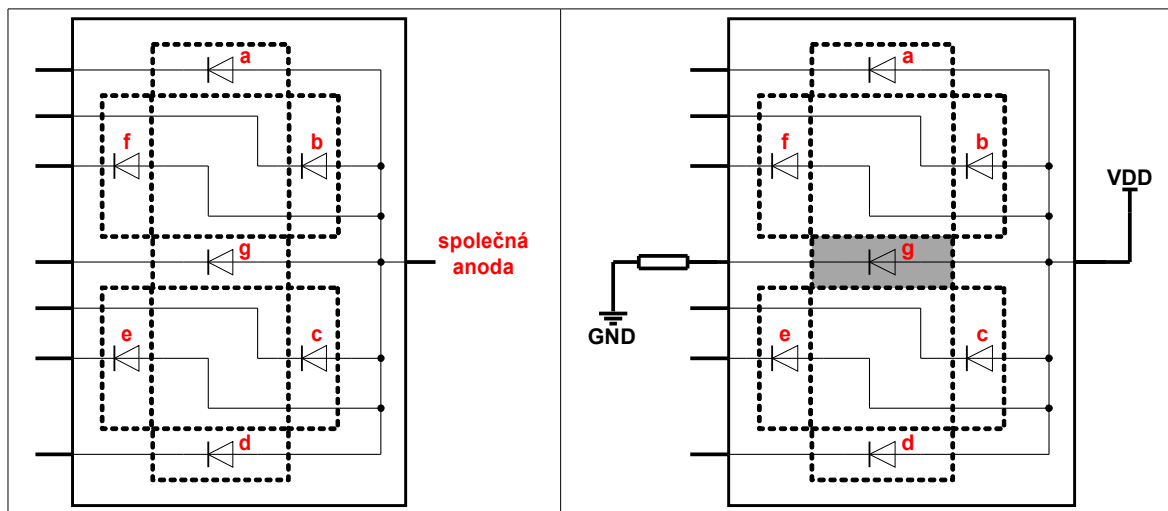


Obr. 2: Symbolické značení 7-segmentového displeje a ukázka zobrazení číslic 0 až 9

Existují dva typy zapojení 7-segmentového displeje:

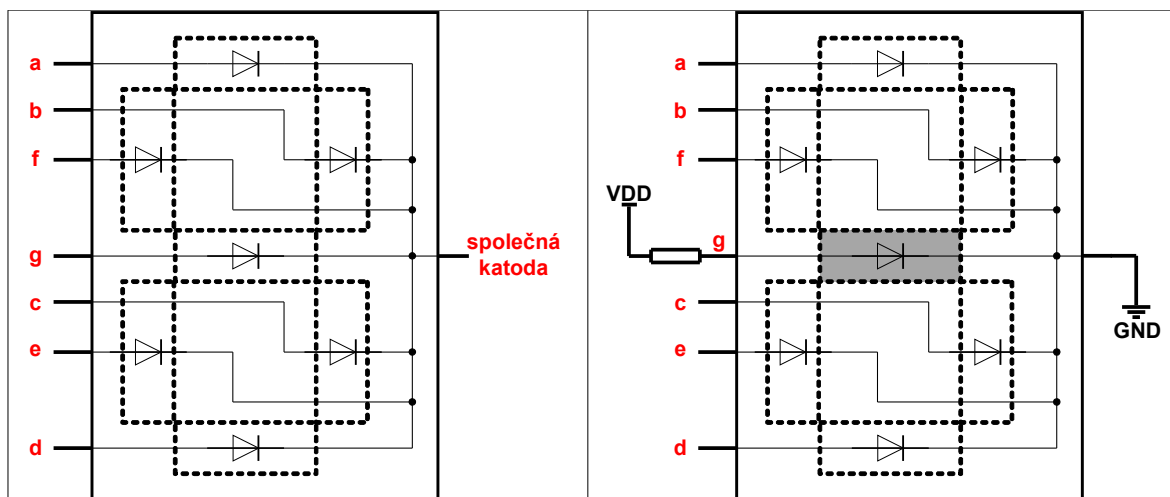
- Se společnou anodou – anoda všech segmentů je připojena na společný uzel. Na tento uzel se většinou připojuje napájecí napětí VDD (log. 1).
- Se společnou katodou – katoda všech segmentů je připojena na společný uzel. Na tento uzel se většinou připojuje zem GND (log. 0).

Na obrázku 3 je zapojení se společnou anodou a ukázka rozsvícení segmentu **g**.



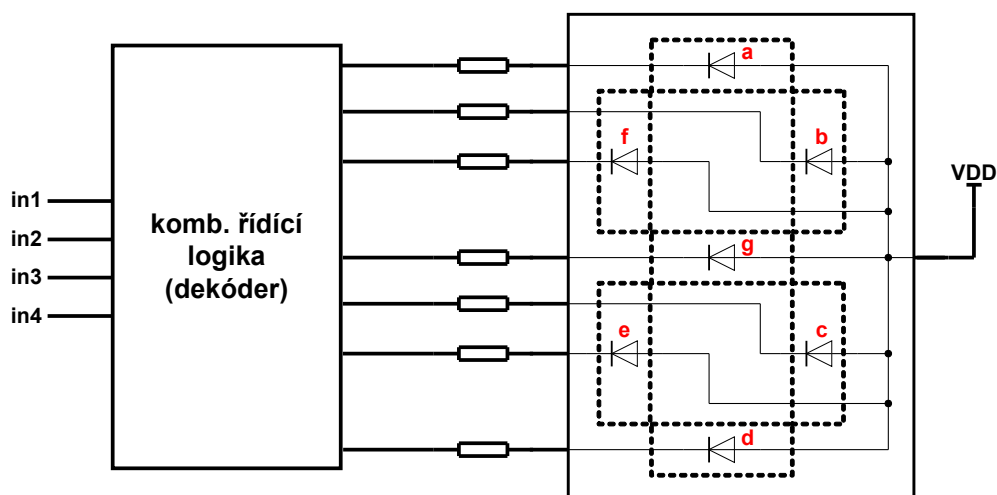
Obr. 3: Zapojení se společnou anodou a ukázka rozsvícení segmentu **g**.

Na obrázku 4 je zapojení se společnou katodou a ukázka rozsvícení segmentu **g**.

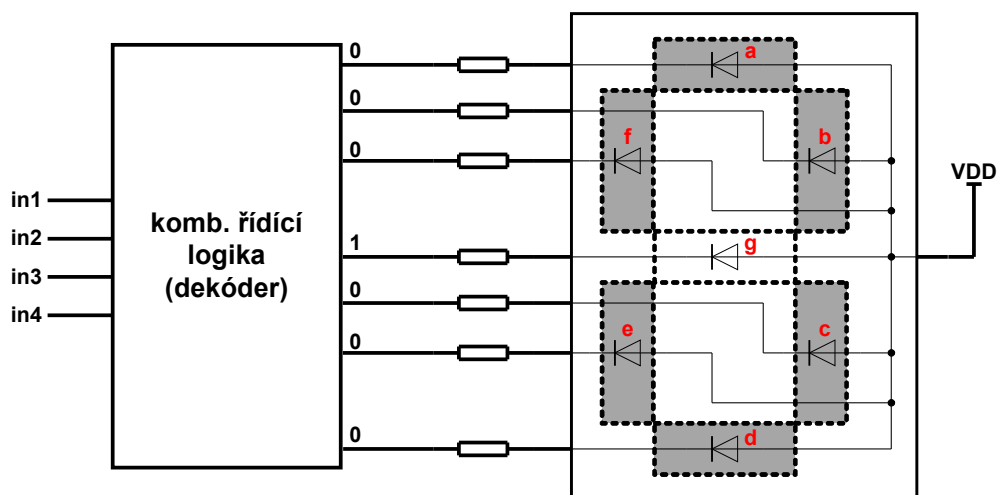


Obr. 4: Zapojení se společnou katodou a ukázka rozsvícení segmentu g.

Blokové schéma propojení 7-segmentového displeje se společnou anodou a dekodéru je na obrázku 5.



Obr. 5: Blokové schéma 7-segmentového displeje se společnou anodou a dekodéru



Obr. 6: Zobrazení číslice 0 na 7-segmentovém displeji se společnou anodou

Pokud budeme chtít zobrazit číslici 0 na displeji se společnou anodou, pak musí být na výstupu dekodéru kombinace 0001000 (obr. 6). Otázkou zůstává, jaká kombinace

je na vstupu dekodéru. V podstatě může být libovolná, to záleží na typu zadání pro návrh dekodéru. Číslice 0 může být zobrazena jak pro vstupní kombinaci 0000, tak i 0100 nebo 1000.

Je skutečně důležité si uvědomit, co od daného dekodéru očekáváme a následně pak navrhujeme příslušnou kombinační logiku.

## Vypracování laboratorní úlohy

Cílem této úlohy je navrhnout vlastní dekodér BCD kódu na 7-segmentový displej.

### Úkol č. 1 (0,2 bodu)

Jak je vyjádřen BCD kód?

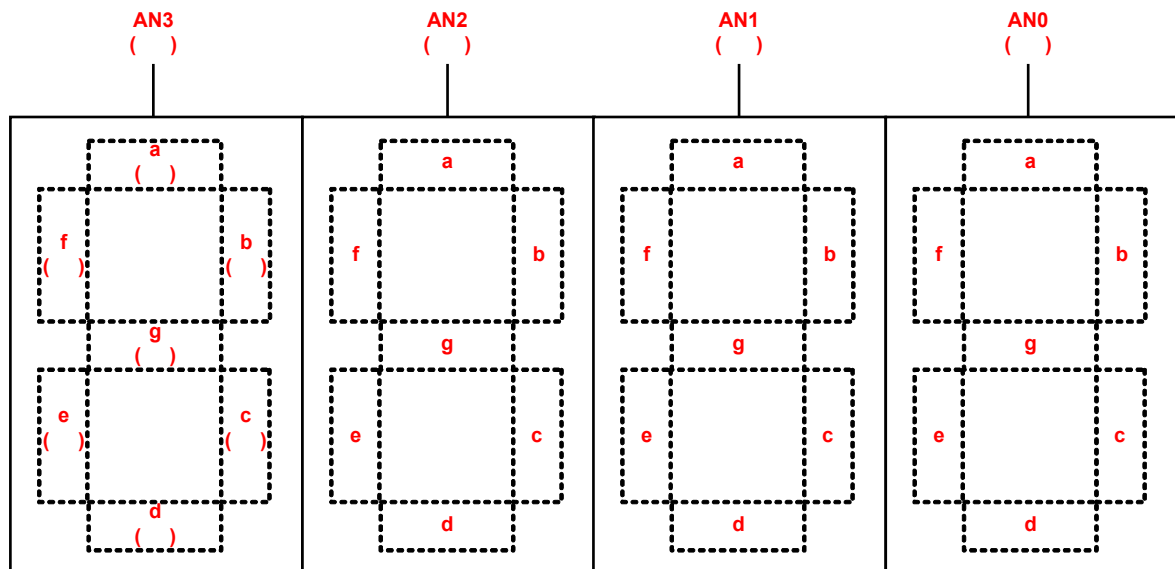
- a) hodnota 0 až 10 v dekadickém tvaru
- b) hodnota 0 až 15 v dekadickém tvaru
- c) hodnota 0000 až 1001 v binárním tvaru
- d) hodnota 0 až F v hexadecimálním tvaru
- e) hodnota 1 až 9 v dekadickém tvaru

### Úkol č. 2 (0,8 bodu)

Zjistěte podle dokumentace k vývojové desce o jaký typ 7-segmentového displeje se jedná:

- a) se společnou katodou
- b) se společnou anodou

Vepište do závorek příslušné piny, které jsou připojené z obvodu FPGA k 7-segmentovému displeji.



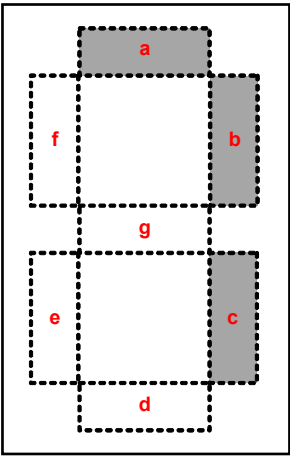
Přepište příslušné piny obvodu FPGA připojené k 7-segmentovému displeji do přehledné tabulky.

Segment	a	b	c	d	e	f	g
FPGA pin							

Anoda	AN3	AN2	AN1	AN0
FPGA pin				

Úkol č. 3 (1,0 bodu)

Doplňte to tabulky logické úrovně, tak aby na 7-segmentovém displeji svítila číslice 7. Zapojení displeje je se společnou anodou.



Segment	a	b	c	d	e	f	g
Log. úroveň							

Navrhněte celou pravdivostní tabulku pro dekodér BCD kódu na 7-segmentový displej.

vstupy				číslice	výstupy						
x3	x2	x1	x0		a	b	c	d	e	f	g
0	0	0	0	0							
0	0	0	1	1							
0	0	1	0	2							
0	0	1	1	3							
0	1	0	0	4							
0	1	0	1	5							
0	1	1	0	6							
0	1	1	1	7							
1	0	0	0	8							
1	0	0	1	9							
1	0	1	0	nic nebude svítit							
až											
1	1	1	1								

Navrhněte minimalizovanou logickou funkci pro segment f.

				x3			
				x2			
x0							
x1							

#### Úkol č. 4 (1,5 bodu)

Vytvořte nový projekt s názvem **uloha3**. Výslednou minimalizovanou funkci nakreslete ve schematicém prostředí (název **decoder**) v návrhovém prostředí Xilinx ISE WebPack, vytvořte konfigurační UCF soubor a implementujte výsledný návrh. Výslednou funkci otestujte na vývojové desce.

#### Úkol č. 5 (1,5 bodu)

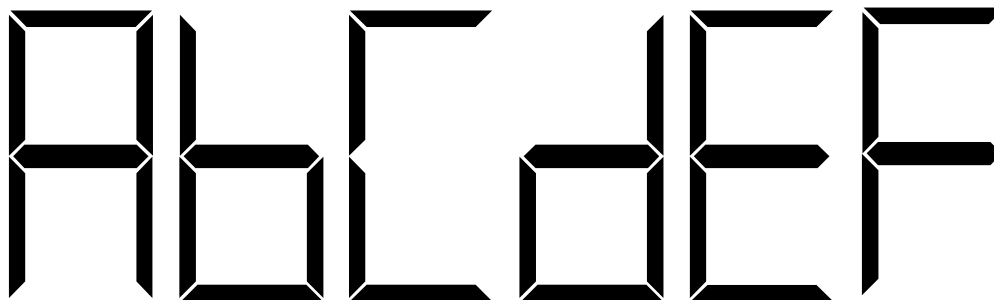
Doplňte prázdná místa ve VHDL popisu. Doplňte VHDL popis na PC tak, aby svítil 7-segmentový displej úplně vpravo. Vložte VHDL popis do projektu **uloha3**. Implementujte výsledný návrh do paměti PROM. Výslednou funkci ověřte pomocí vývojové desky.

```
-----  
entity decoder is                                     -- nazev entity  
port ( x      : in  std_logic_vector (3 downto _);    -- vstupy  
      an      : out std_logic_vector (3 downto 0);    -- vystupy - anody  
      leds    : ____ std_logic_vector (_ downto 0));  -- vystupy - katody  
end decoder;  
  
architecture Behavioral of decoder is                -- nazev architektury  
begin  
    with x select -- "abcdefg"  
        leds <= "      " when "      ", -- cislice 0  
               "1001111" when "0001", -- cislice 1  
               "0010010" when "      ", -- cislice 2  
               "0000110" when "0011", -- cislice 3  
               "      " when "0100", -- cislice 4  
               "0100100" when "      ", -- cislice 5  
               "      " when "      ", -- cislice 6  
               "      " when "      ", -- cislice 7  
               "      " when "      ", -- cislice 8  
               "      " when "      ", -- cislice 9  
               "      " when _____ -- nic nebude svitit  
  
    an <= "      "; -- bude svitit jen 7-segmentovy displej uplne vpravo  
end Behavioral;  
-----
```

#### Bonusový úkol č. 6 (1,0 bodu)

Výsledný VHDL popis upravte tak, aby navržený dekodér nezobrazoval pouze číslíce 0 až 9, ale rovněž hexadecimální znaky.

Požadované hexadecimální znaky:



Jak byste pojmenovali tento typ dekodéru?

Odpověď: