

<b>Předmět</b>	MLOS - LLOS - Logické systémy	
<b>Ústav</b>	ÚAMT	
<b>Úloha č. 2</b>	Návrh základních kombinačních obvodů: dekodér, enkodér, multiplexor, demultiplexor	
<b>Student</b>		

## Cíle

- Porozumění logickým obvodům typu dekodér, enkodér, multiplexor, demultiplexor.
- Schematický návrh v návrhovém prostředí Xilinx ISE WebPack.
- Návrh obvodu s využitím jazyka VHDL v návrhovém prostředí Xilinx ISE WebPack.
- Implementace návrhu do cílového obvodu FPGA Spartan6 na vývojové desce NEXYS3.

## Teoretický úvod

### Dekodér

Dekodér je kombinační obvod, který převádí vstupní kód na odlišný výstupní kód. Velkou skupinu dekodérů tvoří dekodéry převádějící binární kód na kód 1 z N.

Kód 1 z N je definován následovně. Vždy je aktivní maximálně jeden výstup (předpokládejme aktivní stav v log. 1). Pozice aktivního výstupu odpovídá hodnotě binárního čísla na vstupu. Při n-bitovém vstupu platí, že počet výstupů N se rovná maximálně  $2^n$ .

Tabulka 1 je pravdivostní tabulka pro převod dvoubitového binárního kódu na kód 1 z 4. Dekodér popsáný v tabulce 1 obsahuje také vstup blokování en, kterým lze všechny výstupy uvést do neaktivního stavu.

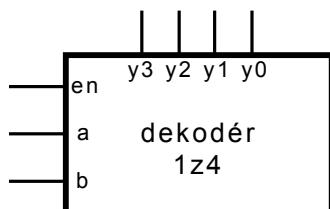
Tab. 1: Pravdivostní tabulka dekodéru binárního kódu na kód 1z4

a	b	en	y3	y2	y1	y0
0	0	0	0	0	0	1
0	1	0	0	0	1	0
1	0	0	0	1	0	0
1	1	0	1	0	0	0
x	x	1	0	0	0	0

x - libovolná hodnota

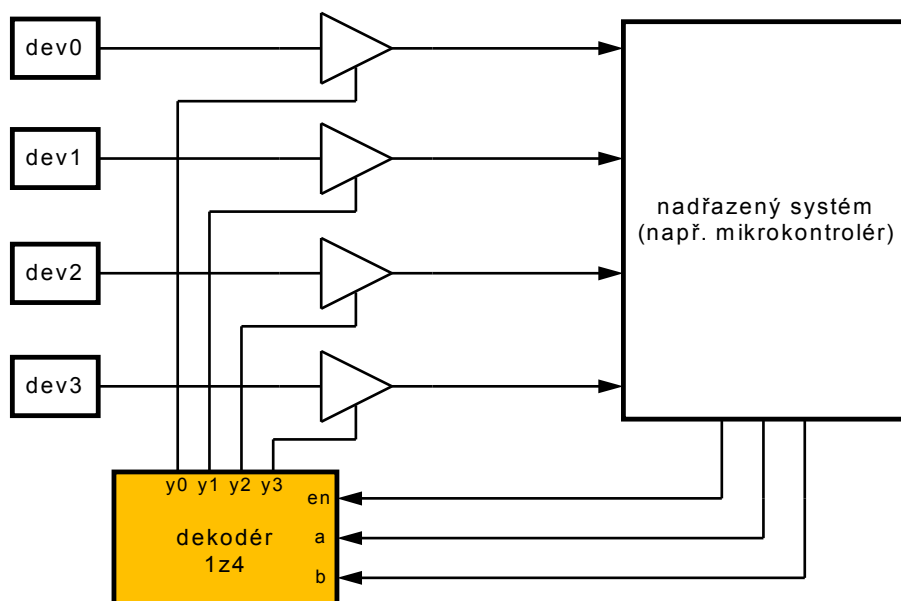
Na obrázcích 1, 2 a 3 je uveden příklad schematické značky dekodéru 1 z 4 (tato značka není definována normou a v praxi je možné se setkat s různými variantami), jeho vnitřní zapojení a příklad použití jako dekodéru pro řízení komunikace čtyř zařízení s nadřazeným systémem.

## Navrhněte zapojení obvodu dle tabulky 1



Obr. 1: Schematická značka dekodéru

Obr. 2: Vnitřní zapojení dekodéru 1z4



Obr. 3: Ovládání komunikace zařízení s nadřazeným systémem s využitím dekodéru 1z4

Další skupinu dekodérů tvoří ty, které dekódují BCD kód na kód pro LCD nebo LED displeje.

### Multiplexor

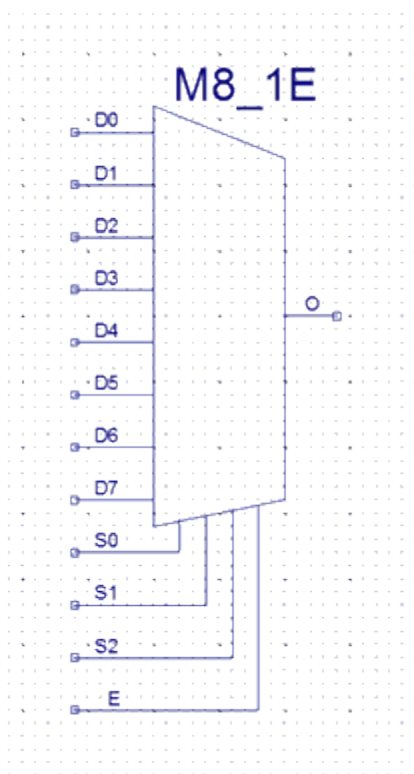
Multiplexor je číslicový přepínač. Má vstupy adresy (v binárním kódu) a vstupy přepínaných signálů. Ty jsou očíslované a čísla vyznačují, při které hodnotě adresy je daný vstupní signál převeden na výstup. Nejjednodušší je dvoukanálový multiplexor, má pouze jeden adresový vstup, který je používán velmi často. Počet přepínaných signálů  $N$  závisí na počtu adresových vstupů  $n$  následovně  $N = 2^n$ .

**POZOR** - směr přenosu signálu nelze obrátit na rozdíl od analogového multiplexoru založeného na spínačích CMOS.

## Realizace logické funkce pomocí multiplexoru

Realizovat logickou funkci můžeme pomocí multiplexoru pouze tehdy, když má multiplexor počet řídicích vstupů minimálně  $N-1$ , kde  $N$  je počet vstupních proměnných logické funkce. Protože v příkladu realizujeme funkci o čtyřech proměnných, vystačíme s multiplexorem se třemi řídicími vstupy.

Při diskrétní realizaci lze jako multiplexor použít například integrovaný obvod 74HC151. V návrhovém prostředí Xilinx ISE WebPack lze pro návrh s multiplexem využít součástky z knihovny MUX. Pro realizaci funkce o čtyřech vstupních proměnných můžeme využít například obvod M8\_1E. Jeho schematická značka je na obrázku 4. Funkce multiplexoru je taková, že multiplexor přenáší signál ze zvoleného vstupu D0 až D7 na výstup O. Vstup, ze kterého je informace převáděna na výstup, je vybrán adresou, binárním číslem na vstupech S2, S1, S0, přičemž vstup S2 má největší váhu. Pro přenos informace ze vstupu na výstup musí být na vstupu E úroveň log. 1. Při log. 0 na vstupu E je na výstupu O trvale log. 0.

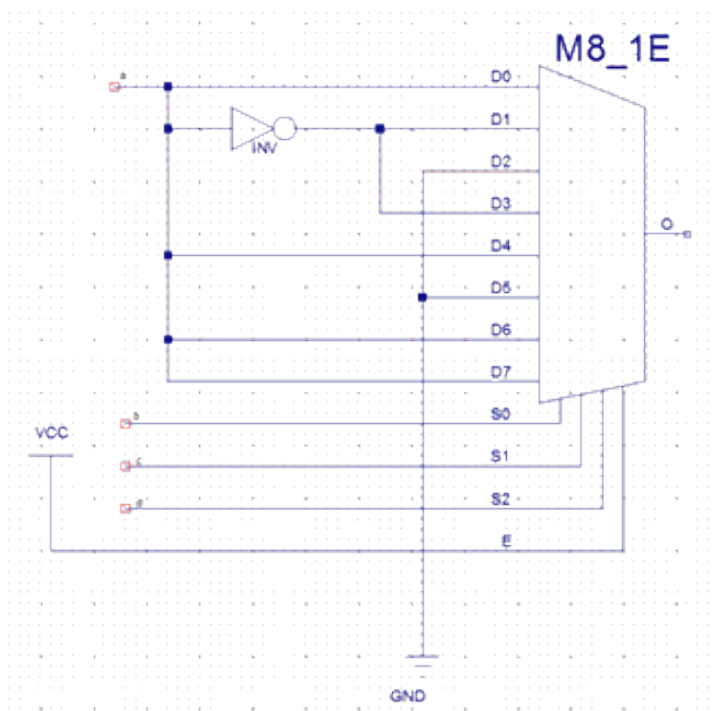


Obr. 4: Schematická značka multiplexoru M8\_1E

Realizaci logické funkce definované v tabulce 2 provádíme tak, že tři vstupní proměnné  $d$ ,  $c$ ,  $b$  přivedeme na vstupy multiplexoru S2, S1, S0. Pro kombinaci vstupních proměnných  $b = 0$ ,  $c = 0$  a  $d = 0$  je signál ze vstupu D0 přenášen na výstup multiplexoru O, což odpovídá stavovým řádkům s indexy 0 a 1 v pravdivostní tabulce. Na tento vstup D0 připojíme vstupní proměnnou  $a$ , protože hodnota výstupní proměnné  $y$  odpovídá hodnotě proměnné  $a$  ( $y = a$ ), viz. tabulka 3.

Tab. 2: Pravdivostní tabulka logické funkce y

e	d	c	b	a	y
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	1
0	x	x	x	x	0



Obr. 5: Realizace logické funkce z tab. 2 s použitím multiplexoru

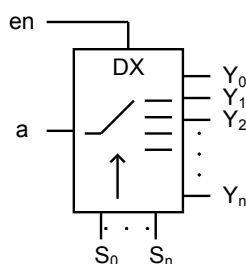
Tab. 3: Vyznačení řádků funkce pro vstupní kombinaci  $d = 0, c = 0, b = 0$

řádek	d	c	b	a	y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1

Mohou nastat další případy. Případ, kdy výstupní hodnota je negací hodnoty vstupní proměnné  $a$  ( $y = \bar{a}$ ) pro určité dva stavové řádky pravdivostní tabulky, řešíme tak, že na příslušný vstup multiplexoru připojíme vstupní proměnnou  $a$ . Dále můžou nastat dva případy, kdy se výstupní hodnota  $y$  pro určité dva stavové řádky nemění. Pak připojíme příslušný vstup multiplexoru na úroveň log. 0 nebo log. 1, podle toho, jaké logické úrovně nabývá výstupní proměnná  $y$  pro dané stavové řádky. Příklad realizace logické funkce zadané v tabulce 2 pomocí multiplexoru je na obrázku 5.

## Demultiplexor

Demultiplexor je opakem multiplexoru. Má jeden vstup ( $a$ ) a několik výstupů ( $y_0$  až  $y_N$ ). Adresou v binárním kódu přivedenou na vstupy ( $S_0$  až  $S_n$ ) je vybrán výstup na který je přenášén signál ze vstupu. Na všech ostatních výstupech je log. 0.



Obr. 6: Blokové schéma n-výstupového demultiplexoru

# Vypracování laboratorní úlohy

## Úkol č. 1 (1,0 bodu)

Doplňte následující tabulku dekodéru, který převede kód **2 z 5** na kód **BCD**.

	2 z 5					BCD			
	e	d	c	b	a	D	C	B	A
0	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1				
2	0	0	1	0	1				
3	0	0	1	1	0				
4	0	1	0	0	1				
5	0	1	0	1	0				
6	0	1	1	0	0				
7	1	0	0	0	1				
8	1	0	0	1	0				
9	1	0	1	0	0	1	0	0	1

## Úkol č. 2 (1,0 bodu)

Vyjádřete neminimalizovanou logickou funkci pro výstupní proměnnou D z předchozího příkladu.

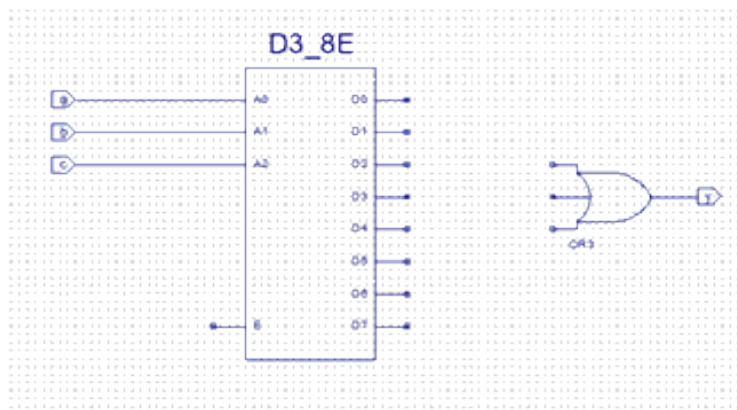
## Úkol č. 3 (1,0 bodu)

Pomocí dekodéru a součtového hradla realizujte funkci tří proměnných. Úlohu neřešte v návrhovém prostředí. Pro realizaci použijte komponentu D3\_8E (3-vstupový dekodér 1 z 8 se řídicím vstupem enable) a součtové hradlo OR z knihovny Logic. Doplňte pravdivostní tabulku funkce y a dokreslete schéma zapojení. K získání pravdivostní tabulky dekodéru D3\_8E využijte nápovědu v návrhovém prostředí. Klikněte na součástku dekodéru a zvolte možnost **SYMBOL INFO**.

$$y = \bar{a} \bar{b} \bar{c} + a \bar{b} c + a b c$$

S	E	c	b	a	y
0	1	0	0	0	
1	1	0	0	1	
2	1	0	1	0	
3	1	0	1	1	
4	1	1	0	0	
5	1	1	0	1	
6	1	1	1	0	
7	1	1	1	1	
8	0	x	x	x	

Pravdivostní tabulka



Návrh rozložení součástek

#### Úkol č. 4 (1,0 bodu)

V návrhovém prostředí Xilinx ISE WebPack navrhnete schematický obvod, který řeší funkci zadanou následující pravdivostní tabulkou. Pro realizaci použijte multiplexor se třemi řídicími vstupy (M8\_1E) a jeho funkci ověřte pomocí vývojové desky.

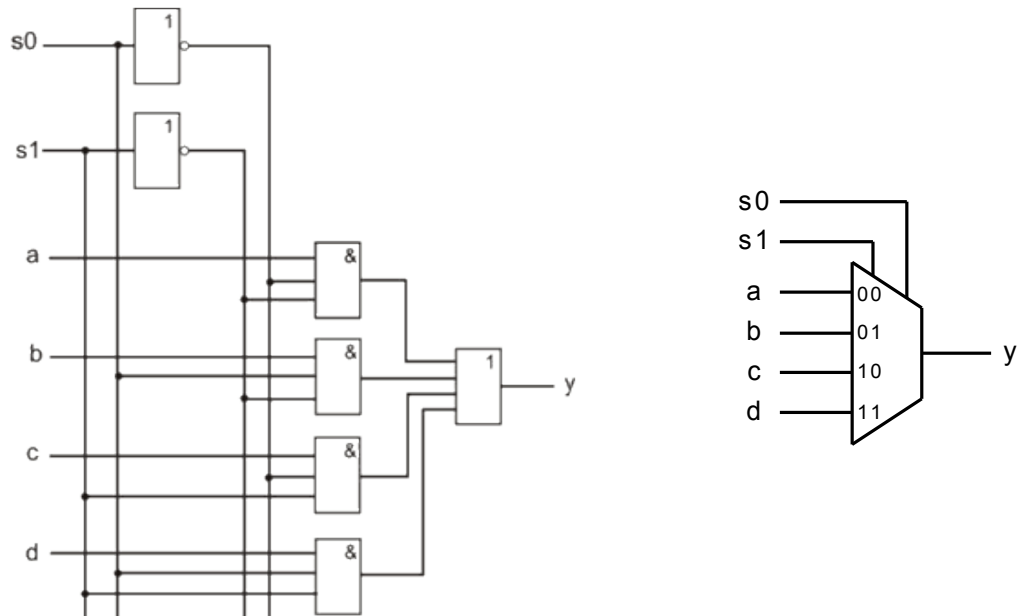
S	E	d	c	b	a	y
0	1	0	0	0	0	0
1	1	0	0	0	1	0
2	1	0	0	1	0	1
3	1	0	0	1	1	1
4	1	0	1	0	0	0
5	1	0	1	0	1	1
6	1	0	1	1	0	0
7	1	0	1	1	1	1
8	1	1	0	0	0	1
9	1	1	0	0	1	0
10	1	1	0	1	0	1
11	1	1	0	1	1	1
12	1	1	1	0	0	0
13	1	1	1	0	1	0
14	1	1	1	1	0	0
15	1	1	1	1	1	1
16	0	x	x	x	x	0

Zapojení vstupů a výstupů

Signál	E	d	c	b	a	y
Přepínač	SW7	SW3	SW2	SW1	SW0	LD0

### Úkol č. 5 (1,0 bodu)

Doplňte následující VHDL popis, který bude reprezentovat schematicky znázorněný 4-vstupový multiplexor. Tento VHDL popis následně zvolte jako vrcholovou jednotku (Top Module) místo schematického návrhu. **A NAHRAJTE DO DESKY !!!**

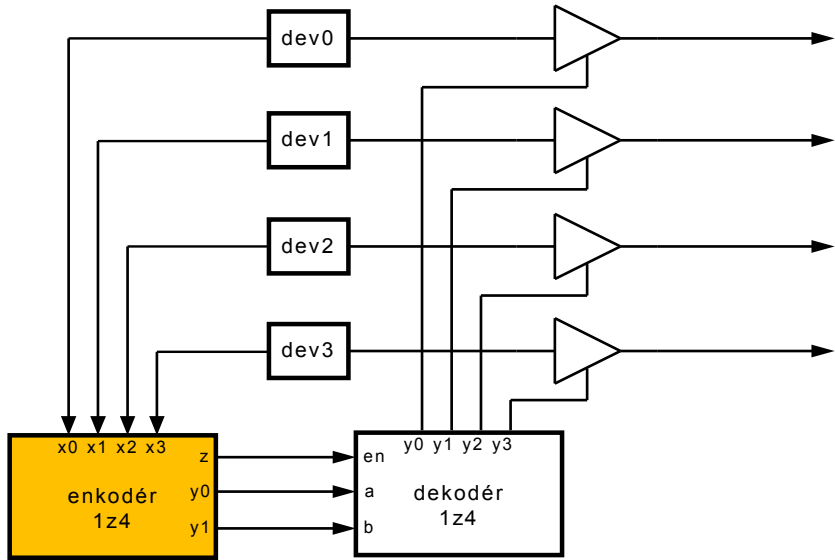


Vnitřní zapojení 4-vstupového multiplexoru    Blokové schéma 4-vstupového multiplexoru

```
-----  
entity mux is  
port (a : in  STD_LOGIC;           -- vstup a  
      b : in  STD_LOGIC;           -- vstup b  
      c : in  STD_LOGIC;           -- vstup c  
      d : in  STD_LOGIC;           -- vstup d  
      s : in  STD_LOGIC_VECTOR (1 downto 0); -- vyberovy vstup s  
      y : out STD_LOGIC);          -- vystup y  
end mux;  
  
architecture Behavioral of mux is  
begin  
    y <= b when s="01" else          -- kombinace S1=0 a S0=1, pak y=b  
          a when s="00" else          -- kombinace S1=0 a S0=0, pak y=a  
          -- doplnte chybejici kod !!!  
          -- doplnte chybejici kod !!!  
  
end Behavioral;  
-----
```

**Bonusový úkol č. 6 (1,0 bodu)**

V návrhovém prostředí Xilinx ISE WebPack navrhnete obvod, který řeší funkci prioritního enkodéru pro čtyři zařízení dle definované priority. Vytvořte VHDL popis a jeho funkci ověřte pomocí vývojové desky.



zařízení	priorita
dev0	1
dev1	3 (max)
dev2	2
dev3	0 (min)

zařízení	vstupy				výstupy		
	x3	x2	x1	x0	y1	y0	z
dev0	-	0	0	1	0	0	1
dev1	-	-	1	-	0	1	1
dev2	-	1	0	-	1	0	1
dev3	1	0	0	0	1	1	1
žádné	0	0	0	0	0	0	0