



Laboratorní cvičení z předmětu CITE

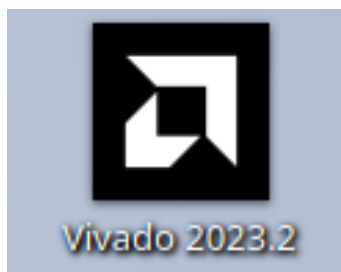
Výběrové přiřazení, dekodéry



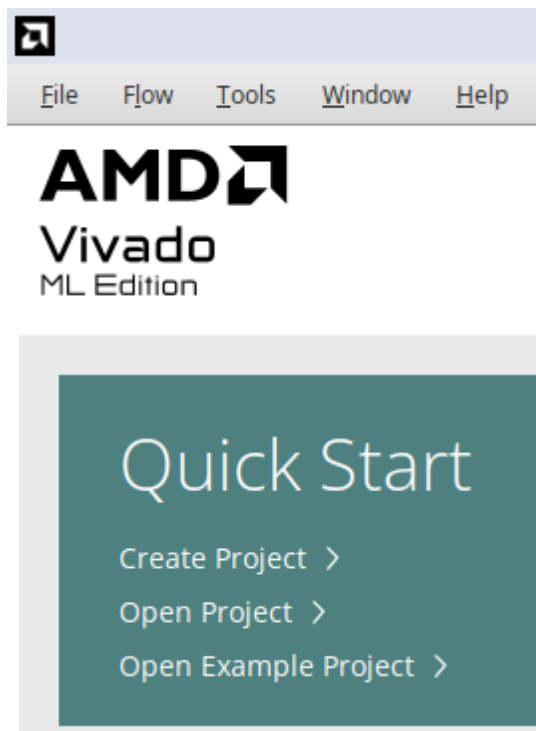
1. Z kurzu předmětu na elearning.tul.cz stáhněte a rozbalte projekt LAB05.zip.

Cesta k projektu nesmí obsahovat diakritiku, mezery nebo speciální znaky. Vhodné umístění je v laboratoři A107 vaše domovská složka, ve kterém si můžete vytvořit podadresář s vaším jménem bez diakritiky.

2. Otevřete Vivado – dvakrát klikněte na spouštěč na ploše:



3. V úvodním okně aplikace klikněte na Quick Start – Open Project:



4. Otevřete soubor LAB05.xpr uvnitř dekomprimované složky.



5. Otevřete kód souboru **decoder.vhd**:

Výběrové přiřazení:

1. Seznamte se syntaxí příkazu výběrového přiřazení, který je obdobou příkazu switch/case, jež znáte z ostatních programovacích jazyků:

```
with výraz_výběru select cíl <=  
výraz_přiřazení1 when volba01,  
[výraz_přiřazení2 when volba11,  
...]  
výraz_přiřazenín when others;
```

- Cíl** je typicky jeden prvek (signál nebo výstup), do kterého chceme přiřadit.
- Výraz výběru** může být signál, vstup nebo obecný výraz, podle kterého volíme, co do cíle přiřadíme. Výraz výběru se porovnává s **volbou**.
- Výraz přiřazení** bude na cíl přiřazen při splnění rovnosti **výrazu výběru** a **volby**.
- Volba **others** bude aktivována, pokud rovnost nesplní žádná z voleb předešlých.
- Výběrové přiřazení nemá, na rozdíl od podmíněného přiřazení prioritní charakter.

Multiplexory z minulé hodiny popsaný příkazem výběrového přiřazení by vypadaly:

mux 2:1

```
with sel select q <=  
a when '0',  
b when others;
```

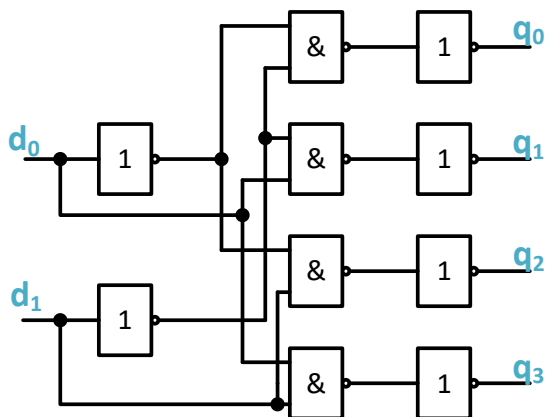
mux 8:1

```
with s select q <=  
d(0) when "000",  
d(1) when "001",  
d(2) when "010",  
... atd.  
d(7) when others;
```

Příkaz se nejčastěji používá na tvorbu dekodérů



Dekodér 1:N

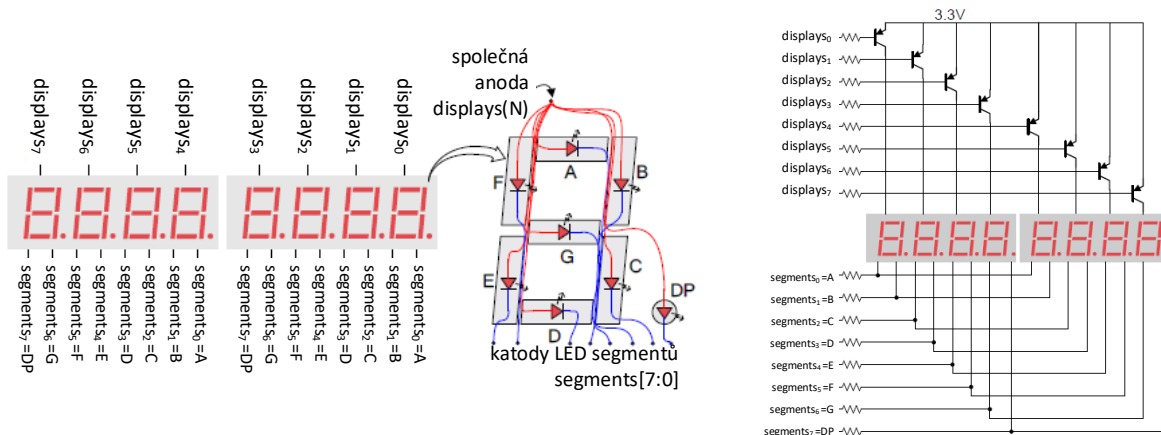


1. Pomocí příkazu **výběrového přiřazení** v souboru **decoder.vhd** popište dekodér z 2bitového binárního kódu do kódu 1:4.
 - a. Do sešitu (nebo rovnou do kódu) si napište pravdivostní tabulku dekodéru do kódu 1:4.
 - b. Cíl přiřazení je výstupní vektor **q**
 - c. Výraz výběru je vstupní vektor **d**
 - d. Výrazy přiřazení budou konstanty (výstupní hodnoty **q**), stejně jako volby (vstupní hodnoty **d**).
2. Zobrazte si schéma obvodu (RTL Analysis, Schematic)
3. Obvod nahrajte do vývojové desky
 - a. Klikněte na Generate Bitstream v menu Flow.
 - b. Po ukončení procesu zvolte Open Hardware Manager.
 - c. Připojte desku a zapněte ji.
 - d. Klikněte v zelené liště na Open target – Auto Connect.
 - e. Klikněte v zelené liště na tlačítko Program a potvrďte.
4. Vstupy **d** jsou namapovány na přepínače [1:0], výstupy na LED [3:0]
5. Porovnejte funkci obvodu s pravdivostní tabulkou.



Dekodér sedmissegmentových displejů

1. Zůstaňte v Hardware manageru.
 - a. Klikněte pravým tlačítkem na obvod **xc7a100t** a zvolte možnost **Program**.
 - b. V dialogovém okně změňte cestu k připravenému bitstream. Zvolte soubor z adresáře projektu **LAB05\manual_displays.bit**
 - c. Stiskněte tlačítko program
2. Seznamte se se zapojením osmice sedmissegmentových displejů na desce Nexys:



- Displej se skládá z 8 LED segmentů. Na obrázku jsou označeny jako A, B, C, ... G, DP. V našem obvodu je máme připojené na přepínačích [7:0], které jsou rovnou zapojeny na výstup **segments**.
 - Všechny segmenty mají v rámci číslice společné napájení (anoda). Každou anodu můžeme zvlášť ovládat pomocí přepínačů [15:8], které jsou připojené na výstup **displays**.
3. Seznamte se s funkcí obvodu. Zapište si:
 - a. Kódy rozsvícení dílčích displejů.
 - b. Kombinace segmentů, které zobrazí znaky 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, b, C, d, E, F.
 - Poloha přepínače „zapnuto“ je opravdu nahoru, je to dáno zapojením displejů se společnou anodou a využitím spínacích tranzistorů typu PNP.
 4. Otevřete soubor **displays.vhd** a nastavte jej na top modul. Pomocí dvou příkazů výběrového přiřazení navrhnete dva dekodéry.
 - a. První dekodér bude převádět 3bit binární váhový kód na vstupu **disp_sel** na kód ovládání rozsvícení jednoho displeje na výstupu **displays**.
 - i. 0=“000“ rozsvítí displej nejvíc vpravo.



- ii. Jedná se o modifikaci dekodéru do kódu 1:N, tentokrát s negativním výstupem.
- b. Druhý dekodér bude převádět 4bit binární váhový kód na vstupu **char_sel** na kód rozsvícení segmentů na výstupu **segments**.
 - i. $0_{10}=0000_2$ = znak 0 na displejích, ... $15_{10}=1111_2$ = znak F na displejích
- 5. Zobrazte si schéma obvodu, vytvořte bitstream a nahrajte jej do FPGA.
 - a. V Hardware Manager po zvolení **program** musíte načíst správný bitstream (jinak vám zůstane manuální ovládání). Bitstream vámi navrženého obvodu se bude nacházet v adresáři projektu LAB05/LAB05.runs/impl_1/displays.bit

