



Laboratorní cvičení z předmětu CITE

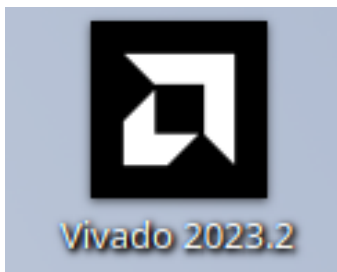
Entita, architektura, nepodmíněné přiřazení, základní logické funkce



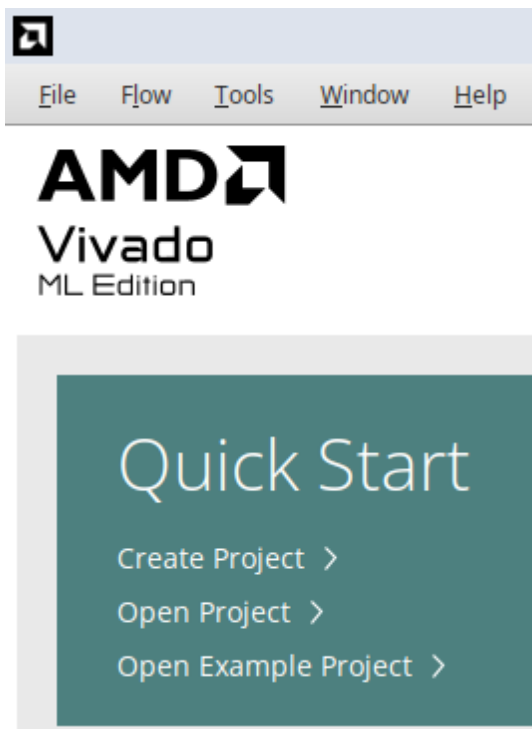
1. Z kurzu předmětu na elearning.tul.cz stáhněte a rozbalte projekt LAB03.zip.

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:

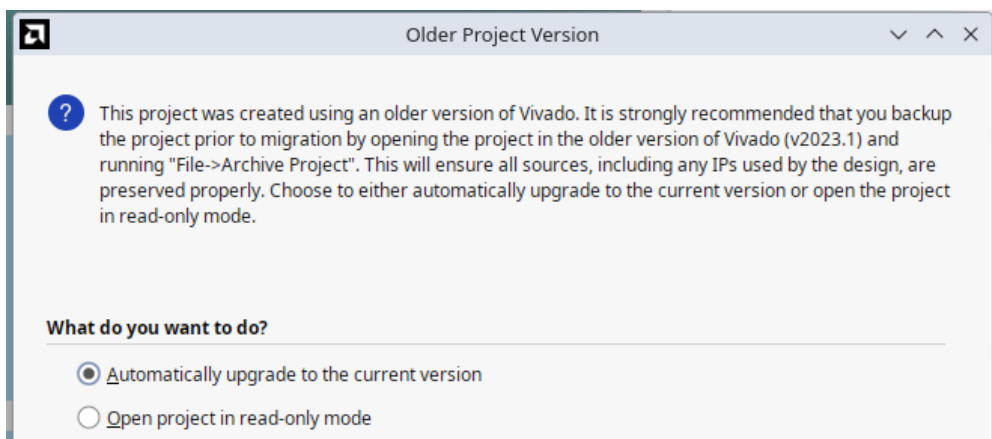


3. V úvodním okně aplikace klikněte na Quick Start – Open Project:

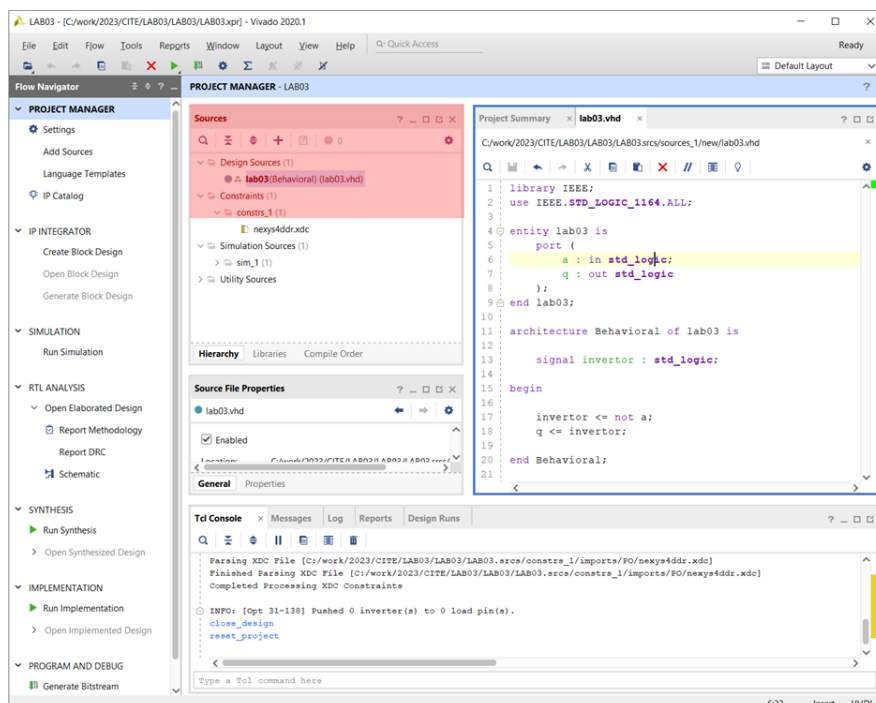




- a. Případné upozornění o novější verzi prostředí, než je projekt vyřešte automatickým upgrade projektu:



- b. Otevře se okno s otevřeným projektem:



4. V okně **Sources** rozbalte složku **Design Sources** a dvakrát klikněte na soubor **lab03 (lab03.vhd)**. Otevře se okno editoru s předpřipraveným kódem v jazyce VHDL.



5. Povězte si:

a. Každý VHDL soubor obsahuje deklarace použitých knihoven:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

b. Vnější popis obvodu ve formě entity. Náš obvod se jmenuje **lab03** a obsahuje deklaraci portů – výstupního portu **q_inv** a vstupního portu **a**:

```
entity lab03 is  
  port(  
    a      : in  std_logic;  
    q_inv  : out std_logic  
  );  
end lab03;
```

c. Popis chování nebo struktury vnitřku obvodu, který se nazývá **architektura** a je přiřazen k naší entitě. V našem případě je architektura **structural** přiřazena k entitě **lab03**:

```
architecture structural of lab03 is  
  
  signal inverter : std_logic;  
  
begin  
  
  inverter <= not a;  
  q_inv    <= inverter;  
  
end structural;
```

d. Architektura se skládá z **deklarační** části a z **těla**. V **deklarační** části je deklarován signál **inverter**. V **těle architektury** se nachází paralelní příkazy. V našem případě dva příkazy nepodmíněného přiřazení.



6. Upravte kód:

- a. Do deklarace portů přidejte vstupy **b, c** typu `std_logic`.
- b. Do deklarace portů přidejte výstupy **q_log0, q_log1, q_and, q_or, q_xor, q_xnor, q_nand, q_nor** typu `std_logic`.

- Záznamy v sekci port jsou oddělovány středníkem
- Za poslední záznam středník nepíšeme (další záznam nenásleduje)
- V případě stejných datových typů a módu (*in, out...*) lze na levé straně záznamu oddělovat jednotlivé porty čárkou
- Názvy portů jsou v rámci Vivado case-sensitive

7. V těle architektury vytvořte třívstupá hradla AND, OR, XOR, XNOR, NAND, NOR se vstupy **a, b, c**. Např. pro hradlo XOR.

```
q_xor <= a xor b xor c;
```

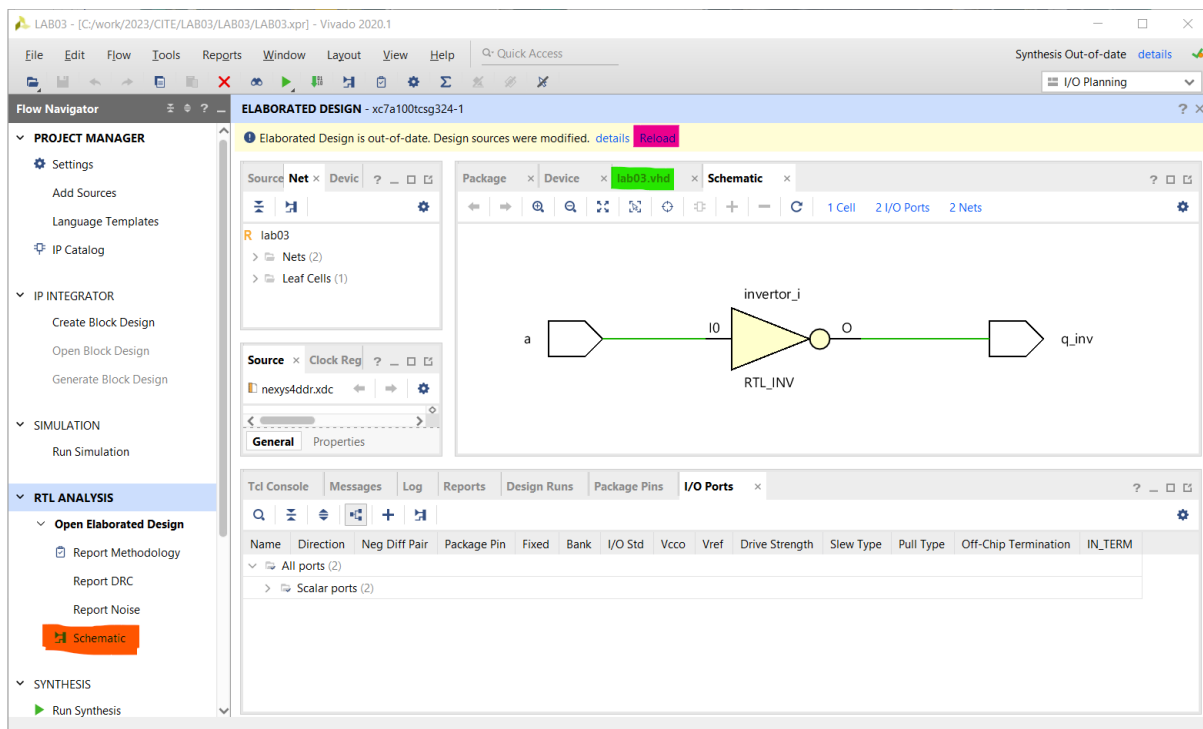
nebo

```
q_xor <= xor(a, b, c);
```

- a. Hradla AND, OR, XOR a XNOR vytvořte pomocí příkazů operátorů **and, or, xor, xnor**.
- b. Hradla AND a OR využijte pro vytvoření hradel NAND a NOR (s pomocí operátoru logické negace **not**)
- c. Na výstupy **q_log0**, a **q_log1** přiřadte odpovídající konstanty. Konstanta datového typu `std_logic` je typu znak a píšeme ji do jednoduchých uvozovek, např. log 0 je **'0'**.



8. Klikněte tlačítko **Schematic** v oddílu **Flow Navigator** (vlevo) v podskupině **RTL analysis**. Na obrázku zvýrazněno oranžově:

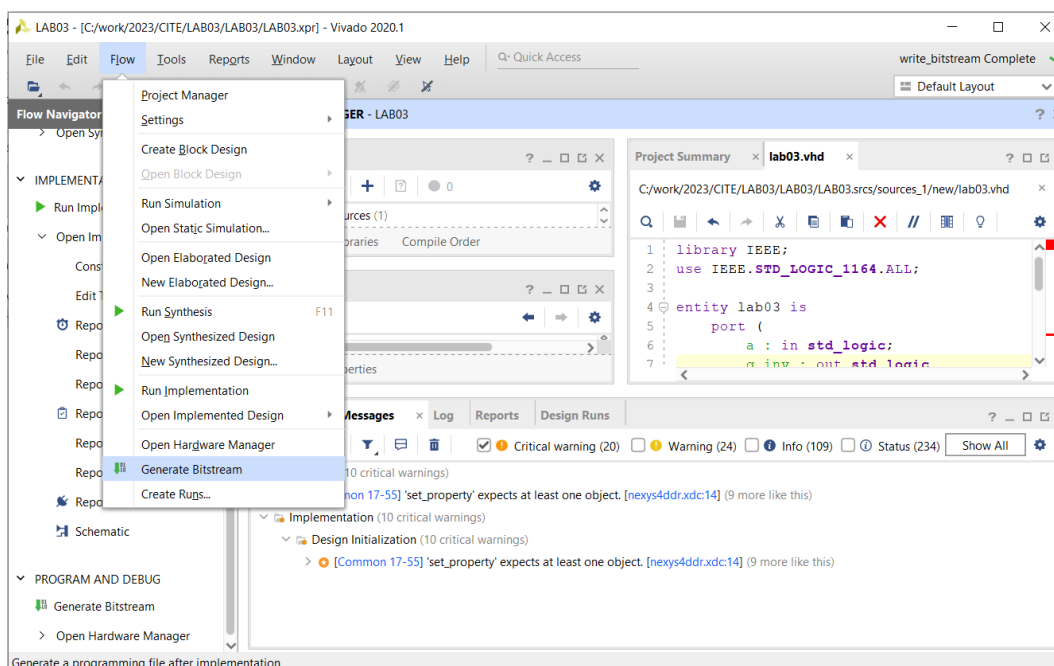


9. Ze schématu ověřte, že váš popis vytvořil příslušná hradla (bude jich víc, než invertor z obrázku výše).
10. Přepněte na záložku VHDL kódu (zeleně) a změňte pořadí příkazů. Znovu prozkoumejte schéma.
- Po změně VHDL kódu a jeho uložení (ctrl + s, nebo ikona diskety vlevo nahoře v okně editoru) se objeví žlutá lišta s tlačítkem **Reload** (na obrázku fuchsiově). Po každé změně kódu je nutné znovu provést **syntézu** obvodu.
 - Všimněte si, že se vytvořila stejná hradla. „Ověřili“ jsme, že jsou příkazy v těle architektury paralelní, nezáleží na jejich pořadí.

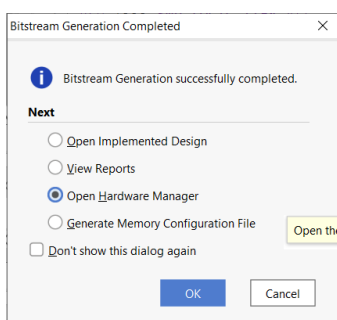


11. Nahrajte váš návrh do vývojové desky:

a. Klikněte na položku **Generate bitstream** v podokně **Flow** nebo z podmenu **Flow**:

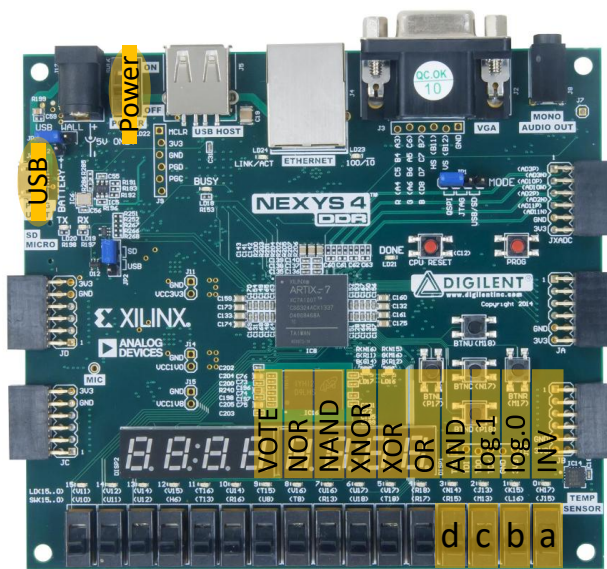


b. Vyčkejte na pop-up okno a zvolte **Open Hardware Manager**:

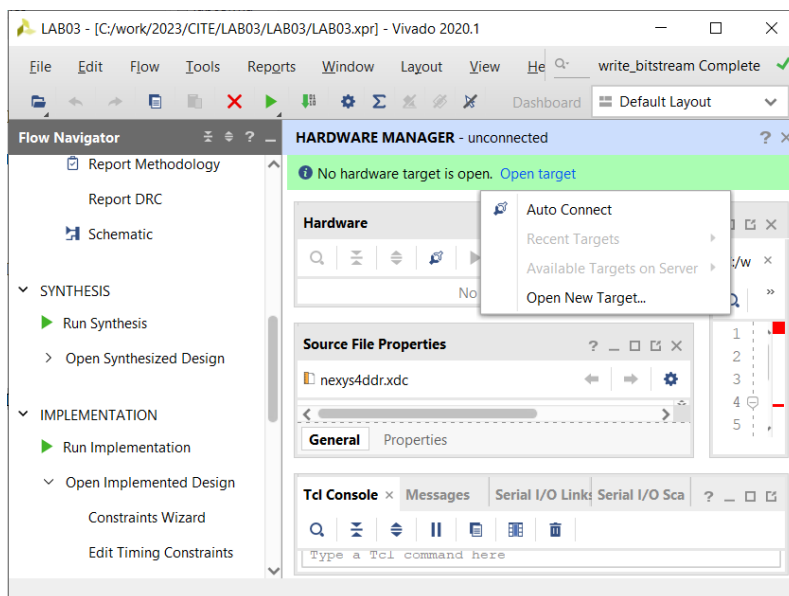




- c. Připojte vývojovou desku k USB portu PC. Přepněte přepínač POWER do polohy zapnuto.

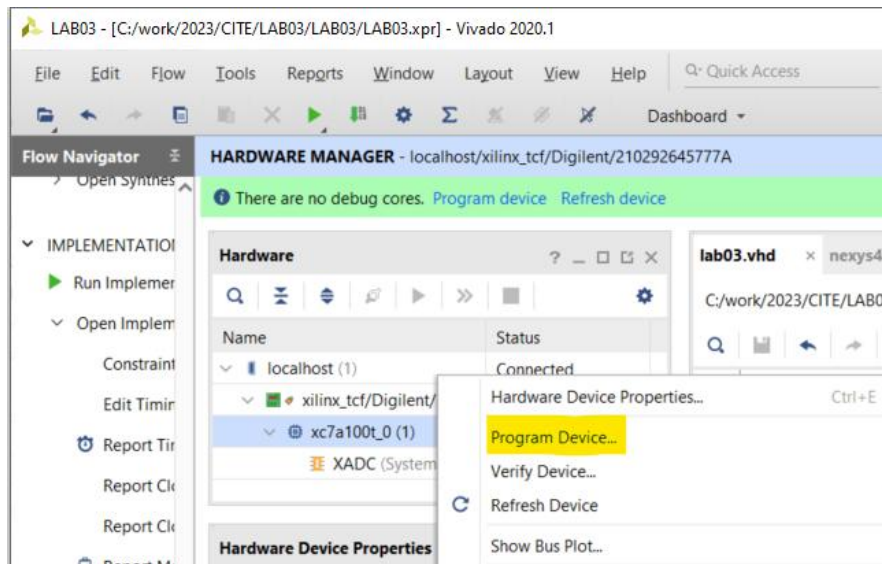


- d. Na zelené liště ve Vivado klikněte na Open target. Vyberte položku Auto Connect





- e. Pravým tlačítkem klikněte na položku xc7a100t_0 a zvolte **Program Device** (viz obrázek dole). V dalším dialogu klikněte na tlačítko **Program**.



12. Ověřte funkci hradel dle pravdivostních tabulek. Vstupy **c**, **b**, **a** jsou namapovány na přepínače, výstupy hradel na LED diody nad přepínači, viz obrázek výše.
13. Otevřete okno s editorem kódu a upravte VHDL soubor:
- Doplňte výstupní port **q_vote** typu **std_logic**.
 - Doplňte vstupní port **d** typu **std_logic**
14. Do těla architektury přidejte obvod, který realizuje funkci tajného hlasování správní rady.
- Vstupy: **a**=předseda, vstupy **b**, **c**, **d** = členové. Log. 1=Pro, 0=Proti. Všichni členové jsou si rovni, ale v případě shody počtu hlasů má hlas předsedy váhu dvou hlasů. Výstup funkce, který indikuje výsledek hlasování zapojte na výstup **q_vote**.
 - Pro dílčí p-termíny minimalizované funkce vytvořte v deklarační části architektury signály (např. pterm0, pterm1, atp.). Příklad deklarace signálu jsou již v souboru.
15. Prohlédněte si navržený obvod schéma navrženého obvodu v **RTL-Analysis\Schematic**
16. Nahrajte obvod do hradlového pole a ověřte jeho funkci.