



Laboratorní cvičení z předmětu CITE

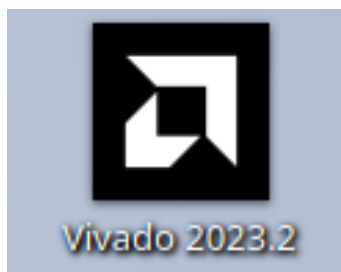
Registry



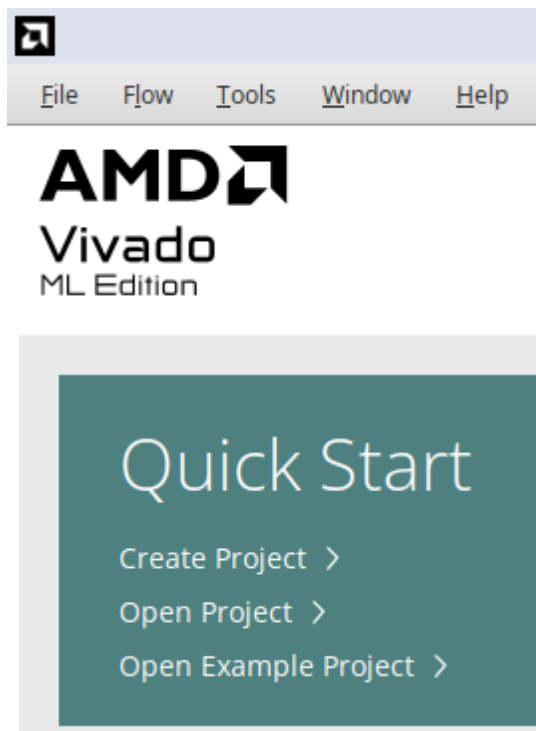
Z kurzu předmětu na elearning.tul.cz stáhněte a rozbalte projekt LAB07.zip.

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.

Otevřete Vivado – dvakrát klikněte na spouštěč na ploše:



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



Otevřete soubor LAB07.xpr uvnitř dekomprimované složky.



Základní klopné obvody

1. Otevřete soubor `simple_reg.vhd`. Seznamte se s porty entity:

clock	Synchronizační hodinový signál
reset	V log.1 aktivní reset signál
data	Datový vstup
enable	Povolení funkce obvodu
q_???	Výstupy klopných obvodů (registrů)

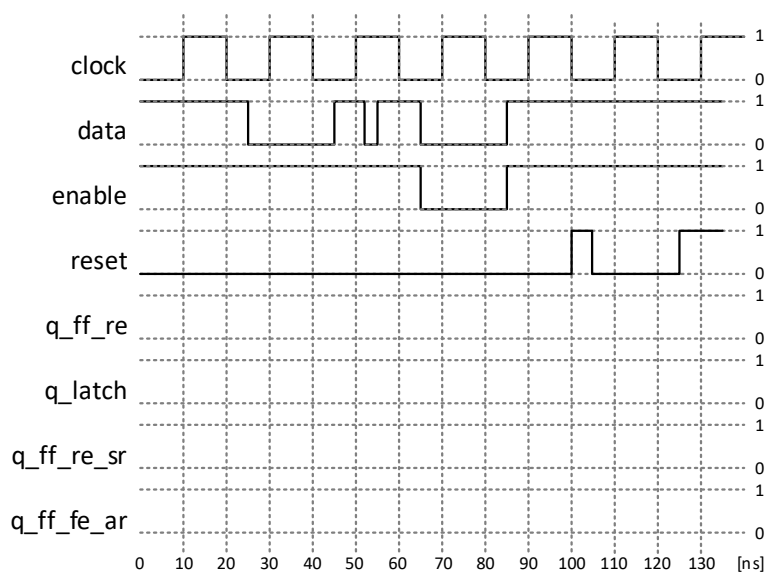
2. Základní klopný obvod typu D (= Delay) lze popsat pomocí následujícího procesu.
Povšimněte si seznamu citlivostních proměnných:

```
signal ff_re : std_logic := '1';  
  
begin  
  
    process(clock)  
    begin  
        if rising_edge(clock) then  
            ff_re <= data;  
        end if;  
    end process;  
    q_ff_re <= ff_re;
```

3. Při řešení úlohy 4 se držte následujících pravidel.
 - Pro každý klopný obvod deklarujte vlastní signál a proces. Výchozí hodnotu signálu nastavte u různých obvodů různou.
 - Rozmyslete, které signály musí být vyjmenovány v citlivostních seznamech procesů. Pro popis sekvenčních obvodů **se nesmí používat** klíčové slovo **all**.



4. Vytvořte:
 - a. **Hladinový** klopný obvod
 - i. Obvod reaguje na **hodinový** signál aktivní v **log. 1**
 - ii. Obvod je opatřen asynchronním resetem
 - iii. Výstup obvodu připojte na výstup **q_latch**
 - b. **Hranový** klopný obvod se synchronním resetem a vstupem enable
 - i. Obvod reaguje na **náběžnou** hranu **hodinového** signálu
 - ii. Funkce resetu má nižší prioritu než hodiny
 - iii. Funkce enable má nižší prioritu než reset
 - iv. Výstup obvodu připojte na výstup **q_ff_re_sr (Flip Flop Rising Edge Synchronous Reset)**
 - c. **Hranový** klopný obvod s **asynchronním resetem**:
 - i. Obvod reaguje na **sestupnou** hranu **hodinového** signálu
 - ii. **Hodiny** mají **nižší** prioritu než **reset**
 - iii. **Enable** má nižší prioritu než hodiny
 - iv. Výstup obvodu připojte na výstupu **q_ff_fe_ar**
5. Otevřete si simulaci **simple_regs_tb.vhd**, upravte ji a ověřte chování vašich obvodů. Pro porovnání obvodů využijte následujícím buzení a výsledek si poznamenejte. Cvičícímu vysvětlíte rozdíly mezi obvody.





V jazyce VHDL lze popsat celou škálu typů registrů (ano, i JK, RS, T, aj.). Pokud vyrábíme obvod ASIC, nebo jen simulujeme, pak lze popsat téměř cokoliv. U obvodů FPGA jsou ale registry a hodinové signály již vyrobeny a náš popis by měl být takový, abychom je co nejlépe využili. Proto se domluvíme na následujících pravidlech:

- Podmínka `rising_edge(argument)` v procesu provádí inferenci klopného obvodu řízeného synchronizačním signálem. Jako argument nikdy nepoužijeme datový signál, ale pouze hodiny.
- Registry v FPGA umí reagovat na náběžnou i sestupnou hranu. Neumí však obě současně. Z dobrých důvodů se v našich návrzích budeme držet pouze náběžné hrany, a to u všech registrů v obvodu.
- Nebudeme používat asynchronní reset, ani set.
- Nebudeme používat set a reset současně.

Poslední pravidla nejsou nutná, ale zlepšují výsledky časování obvodu:

- Řídící signály `enable`, `set` a `reset`, jsou navrženy pro pozitivní polaritu. Má-li náš obvod polaritu negativní, raději vytvoříme invertovaný signál, který k našim registrům rozvedeme.
- Pokud je to možné, tak `set` nebo `reset` nepoužijeme.