

B2B32DITA

**Laboratorní úloha č. 3**

**Podmínkové konstrukce v jazyce VHDL, konverze typů, realizace sčítačky**

**Postup práce**

1. Nejprve prostudujte úkol dne.

**a) Prvotní realizace úplné 1bitové sčítačky**

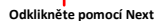
2. Spusťte program *Quartus Lite Edition*.<sup>1</sup>
3. Vytvořte nový projekt pomocí ikony na úvodní obrazovce *New Project Wizard*.
4. V prvním okně jen dole odklikněte *Next*.  
V dalším okně specifikujte cestu pro uložení projektu.<sup>2</sup>  
V další řádce vyplňte vhodný název souboru<sup>3</sup>, např. *adder*.  
Poslední řádek se jménem top-level entity bude vyplněn automaticky a okno v dolní části odklikněte *Next*.  
V dalším okně ponechte volbu na možnosti „*Empty project*“ a odklikněte *Next*,  
další okno opět jen odklikněte pomocí *Next*.

1 - Pokud se po spuštění Quartus otevře předchozí projekt, ukončete ho (File → Close Project).

2 – Zvolte pro umístění složky projektu Vaši pracovní složku na disku ve složce *Users* a v něm podsložku s Vaším přihlašovacím jménem.

3 – Nepoužívejte diakritiku, mezeru, speciální znaky a jméno nesmí začínat číslem.

5. V dalším okně „*New Project Wizard – Family, Device & Board Settings*“ zvolte správný přípravek DE10-Lite<sup>1</sup>.




Zde v levém roletovém menu „Family:“ zkontrolujte, že je označena položka „MAX 10“.

Kliknutím levým tlačítkem myši ji označte.

Další okno opět jen odklikněte pomocí Next a konečně v posledním okně klikněte na Finish.

**a) Prvotní realizace úplné 1bitové sčítačky**

6. V menu hlavního okna „File“ klikněte na položku „New...“, případně klikněte na ikonu  v hlavní liště ikon.  
Z nabídky zdrojů vyberte položku „VHDL File“ v sekci „Design Files“ a potvrďte pomocí OK.
7. Vytvoří se prázdný soubor pro zápis VHDL kódu.  
Nejprve soubor uložte<sup>1</sup>.  
Zvolte vhodný název souboru např. *adder*.
8. Nyní postupně zkopírujte a doplňte do vytvořeného souboru předpřipravenou část VHDL kódu z kapitoly 8 teoretického úvodu.<sup>2</sup>  
Kód je rovněž uveden na následující straně.

1 – V menu „File“ zvolte „Save As...“. V nově otevřeném okně zkontrolujte, že soubor bude uložen ve složce projektu. Název ukládaného souboru by měl odpovídat názvu projektu s příponou „.bdf“. Zkontrolujte, že volba „Add file to current project“ je zaškrtnuta a klikněte na tlačítko *Uložit*.

2 - Ta obsahuje deklaraci knihoven, deklaraci entity a jejích portů a dále úvodní deklaraci architektury s názvem *Behavioral\_concurrent*. Tento název lze samozřejmě libovolně změnit. V rámci architektury je zde rovněž deklarován signál s názvem *vstup* typu *std\_logic\_vector* o velikosti 3 bitů, do kterého je přiřazeno zřetězení 3 vstupů 1bitové sčítačky v daném pořadí.

**a) Prvotní realizace úplné 1bitové sčítačky**

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

entity adder is
port (a,b,cin : in std_logic;
      s,cout : out std_logic);
end adder;

architecture Behavioral_concurrent of adder is
signal vstup : std_logic_vector(2 downto 0);
begin
vstup<=cin&b&a;
...
end Behavioral_concurrent;
```

#### a) Prvotní realizace úplné 1bitové sčítačky

9. Do takto předpřipravené entity a její architektury `Behavioral_concurrent` nyní doplňte místo trojice teček behaviorální popis úplné 1bitové sčítačky na základě tab. č. 1 v teoretickém úvodu s použitím podmínkové konstrukce `when/else` v paralelním (`concurrent`) prostředí. Celý projekt uložte.<sup>1</sup>

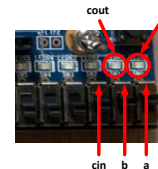
10. Spusťte kompilaci projektu.<sup>2</sup>

11. V případě úspěšně zkompilovaného projektu přiřaďte vstupům/výstupům jejich fyzické zapojení na přípravku DE10-Lite.<sup>3</sup>

Ve sloupečku *Location* uveďte u každého portu v seznamu jeho přiřazení na správný pin přípravku:

Vstup/výstup	a	b	cin	s	cout
Pin	C10	C11	D12	A8	A9

Tomuto přiřazení odpovídají 3 přepínače a 2 LED ve spodní části přípravku:



1 – V menu *File* → *Save All* s ikonou několika disket za sebou v hlavní nabídce programu Quartus.

2 – Postup:

Kompilaci zahájíte pomocí ikony modrého trojúhelníku v hlavní liště ikon v horní části programu Quartus.

Pokud se v průběhu kompilace nevyskytla žádná chyba či problém, bude u všech jejích fází v okně *Tasks* uvedeno zelené úspěšné potvrzení (fajfka), v opačném případě bude u příslušné fáze kompilace zobrazena červená ikona chyby (křížek).

Počet a bližší popis chyb lze nalézt v konzoli níže kliknutím na ikonu bílého křížku v červeném kolečku a listováním ve zprávách.

Proveďte postupně opravy případných chyb a problémů, začněte vždy s opravou chyb odshora.

Po každé opravě chyby opět uložte celý projekt v menu *File* → *Save All* v hlavní nabídce programu Quartus a spusťte pomocí ikony modrého trojúhelníku novou kompilaci projektu.

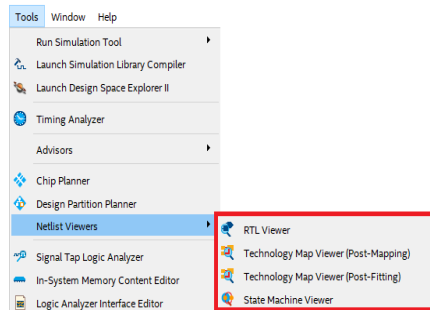
3 – V hlavní horní liště ikon klikněte tedy na ikonu *Pin Planner*, případně vyberte v menu *Assignments* → *Pin Planner*. Otevře se okno *Pin Planner* pro přiřazování pinů

vstupům/výstupům v projektu.

Ve spodní části okna *Pin Planneru* se nachází seznam všech vstupů a výstupů definovaných v projektu (schématu či top-level entitě projektu).

**a) Prvotní realizace úplné 1bitové sčítačky**

12. Po přiřazení všech pinů okno *Pin Planner* uzavřete, uložte celý projekt a spusťte pomocí ikony modrého trojúhelníku ▶ novou kompilaci projektu.
13. Nyní v hlavní menu otevřete položku *Tools* → *Netlist Viewers* → *RTL Viewer*, viz obrázek níže. Zobrazí se RTL model navrženého obvodu tak, jak jej syntezátor navrhne na základě popisu v architektuře entity.  
Poznamenejte si zjištění, která se vztahují k tomuto kroku.  
Zavřete okno *RTL Viewer*.





**a) Prvotní realizace úplné 1bitové sčítačky**

14. Dále opět ve stejném menu programu Quartus otevřete *Tools* → *Netlist Viewers* → *Technology Map Viewer (Post-Mapping)*.  
Spustí se okno *Technology Map Viewer*<sup>1</sup>.  
Najděte dvojici bloků s označením „*LOGIC\_CELL\_COMB*“, které představují dvojici bloků obsahujících kombinační logické obvody, v podstatě LUT tabulky.  
Na libovolný z bloků klikněte pravým tlačítkem myši a z menu vyberte položku „*Properties*“.  
V levé části okna *Technology Map Vieweru* se otevře dvojice oken s názvem *Properties*.  
V horním okénku je zobrazeno zapojení kombinačního obvodu<sup>2</sup>.  
Poznamenejte si zjištění, která se vztahují k tomuto kroku.  
Zavřete okno *Technology Map Viewer*.

1 – V předchozím kroku č. 13 představuje RTL model obecný návrh realizace popsaného logického obvodu tak, jak jej syntezátor z popisu v jazyce VHDL idealizovaně navrhne bez uvažování dostupných HW prostředků, a tedy na čistě SW bázi.

Model v bodě č. 14 představuje reálně navrženou implementaci téhož obvodu po aplikaci všech metod optimalizace a namapovanou na prostředky dostupné v daném FPGA poli (což odpovídá jeho výsledné skutečné realizaci).

2 – Kombinační obvod je vytvořen pomocí základních elementárních hradel (AND, OR, XOR, NOT apod.) a se vstupy a výstupem, které odpovídají portům v entitě (*a*, *b*, *c<sub>in</sub>*, *c<sub>out</sub>*, *s* – pokud jste zvolili označení ve shodě s tímto návodem) dle zapojení v hlavním schématu v okně *Technology Map Vieweru*.

**a) Prvotní realizace úplné 1bitové sčítačky**

15. Připojte přípravek Terasic DE10-Lite pomocí USB kabelu do PC.<sup>1</sup>
16. V okně *Tasks* (levé dolní okno) klikněte dvakrát levým tlačítkem na položku „*Program Device (Open Programmer)*“.  
Spustí se okno pro naprogramování přípravku DE10-Lite s názvem *Programmer*.
17. Zkontrolujte, že je přes USB rozhraní korektně načten přípravek DE10-Lite s FPGA MAX 10.<sup>2</sup>  
Klikněte na tlačítko „*Start*“.  
V okénku „*Progress:*“ proběhne naprogramování přípravku, pokud je vše v pořádku, zobrazí se „*100% (successful)*“ v zeleném poli.
18. Otestujte funkčnost a správnost navržené sčítačky.<sup>3</sup>  
Výsledek ukažte učiteli.

1 – Poznámka: dvojice zelených LED, jedna na levém okraji přípravku (tzv. POWER GOOD) a jedna v levém horním rohu přípravku (tzv. CONF\_D), by se měla rozsvítit. Červené indikační LED v pravé dolní části a 7segmentové displeje ve spodní části přípravku začnou blikat.

2 – V levém horním rohu okna *Programmer* klikněte na tlačítko „*Hardware Setup...*“. V tomto okně v položce „*Currently selected hardware:*“ rozklikněte roletové menu a vyberte položku „*USB-Blaster [USB-0]*“.

Zavřete okno pomocí *Close*.

Zkontrolujte, že ve spodní části okna se nachází FPGA pole s označením 10M50DAF484ES.

Ověřte, že v horní části okna je vybrán soubor pro naprogramování FPGA pole z aktuální složky projektu.

Zkontrolujte, že je zaškrtnuta volba ve sloupečku *Program/Configure*.

3 – Pomocí přepínačů nastavte postupně všechny vstupní kombinace proměnných *a*, *b*, *c* in a na dvojici LED zkontrolujte výstupy *s* a *c* out.

Porovnejte s pravdivostní tabulkou sčítačky (tab. č. 2 v teoretickém úvodu).

**b) Druhý způsob realizace úplné 1bitové sčítačky**

19. Zavřete okno *Programmer* křížkem v pravém horním rohu.

20. Nyní do entity 1bitové sčítačky doplňte další architekturu `Behavioral_sequential` pod předchozí `Behavioral_concurrent`.

Výsledná entita tak bude vypadat např.:<sup>1</sup>

```
...
architecture Behavioral_concurrent of adder is
signal vstup : std_logic_vector(2 downto 0);
begin
vstup<=cin&b&a;
...
end Behavioral_concurrent;

architecture Behavioral_sequential of adder is
signal vstup : std_logic_vector(2 downto 0);
begin
vstup<=cin&b&a;
. end..
Behavioral_sequential;
```

1 – bez opakování úvodní deklarace knihoven a portů

**b) Druhý způsob realizace úplné 1bitové sčítačky**

21. Do architektury `Behavioral_sequential` doplňte místo trojice teček behaviorální popis úplné 1bitové sčítačky<sup>1</sup> s použitím podmínkové konstrukce `case` v sekvenčním (`sequential`) prostředí. Uložte celý projekt.<sup>2</sup>  
Zkompilujte celý projekt a opravte případné chyby.
22. Zobrazte RTL model.<sup>3</sup>  
Zobrazte technologický model.  
Implementujte sčítačku do přípravku DE10-Lite.  
Ověřte funkčnost sčítačky a ukažte ji učiteli.  
Poznamenejte si zjištění, která se vztahují k tomuto kroku.

1 – na základě tab. č. 1 z teoretického úvodu

2 – Menu *File* → *Save All* s ikonou několika disket v hlavní nabídce programu Quartus.

3 – Postup odpovídá bodům č. 13, 14, 15, 16, 17 a 18 z předchozí části.

**c) Třetí způsob realizace úplné 1bitové sčítačky**

23. Zavřete okno *Programmer* křížkem v pravém horním rohu.

24. Doplňte do entity 1bitové sčítačky další architekturu RTL pod předchozí dvojici architektur `Behavioral_concurrent` a `Behavioral_sequential`.<sup>1</sup>  
Výsledný kód je uveden na následující straně.

1 – Výsledná entita tak bude obsahovat 3 architektury pod sebou (opět bez opakování úvodní deklarace knihoven a portů).

c) Třetí způsob realizace úplné 1bitové sčítačky  
24.

```
...  
architecture Behavioral_concurrent of adder is  
signal vstup : std_logic_vector(2 downto 0);  
begin  
vstup<=cin&b&a;  
...  
end Behavioral_concurrent;  
  
architecture Behavioral_sequential of adder is  
signal vstup : std_logic_vector(2 downto 0);  
begin  
vstup<=cin&b&a;  
...  
end Behavioral_sequential;  
  
architecture RTL of adder is  
begin  
...  
end RTL;
```

**c) Třetí způsob realizace úplné 1bitové sčítačky**

25. Do nové architektury RTL nyní doplňte místo trojice teček RTL (dataflow) popis úplné 1bitové sčítačky<sup>1</sup> s použitím výstupních funkcí<sup>2</sup>, které přepíšete do jazyka VHDL.  
Uložte celý projekt.<sup>3</sup>  
Zkompilujte celý projekt a opravte případné chyby.
26. Zobrazte RTL model.<sup>3</sup>  
Zobrazte technologický model.  
Implementujte sčítačku do přípravku DE10-Lite.  
Ověřte funkčnost sčítačky a výsledek ukažte učiteli.  
Poznamenejte si zjištění, která se vztahují k tomuto kroku.
27. Pokud nechcete vypracovávat bonusový úkol, zavřete okno *Programmer* křížkem v horním pravém rohu okna.  
Zavřete celý projekt *File* → *Close Project*.  
Ukončete program Quartus.  
Odhlase se z PC.

1 - na základě tab. č. 1 teoretického úvodu.

2 – viz vztah (1) z teoretického úvodu.

3 – Menu *File* → *Save All* s ikonou několika disket v hlavní nabídce programu Quartus.

4 – Postup odpovídá bodům č. 13, 14, 15, 16, 17 a 18 z předchozí části.