Cvičenie 3 Funkčná a časová simulácia kombinačnej logiky v ModelSime.

Otvorit' Quartus II

Horné menu: File: New Project Wizard



Next...

Zadajte cestu k projektu, názov projektu a meno "top level design entity" (odporúčam používať rovnaké názvy)

New Project Wizard: Directory, Name , Top-Level Entity [page 1 of 5]		×
What is the <u>w</u> orking directory for this project?		
D:\= PHD =\= VYUKA =\PL0_2008\cv3_basic_logic		
What is the name of this project?		
cv3_basic_logic		
What is the name of the top-level design entity for this project? This name is case sensitive a exactly match the entity name in the design file.	and must	
cv3_basic_logic		
Lies Evisting Region Cottings		
Use Existing Project Settings		
< Rack Next > Finish	Storne	
	Storno	

Next... Next...

Vyberte súčiastku EPM3064ATC44-10

Family: MAX3000A			-	Show in 'Availa	ble device' list	
Tarast darden			_	Package:	Any	-
C Auto device selected by the Fitter			Pin <u>c</u> ount:	Any	-	
Specific device selected in 'Available devices' list		Speed grade:	Any	•		
			Show advanced devices			
				HardCopy of		
Available devices:						
Name	Core v	Macro				~
EPM3032ATI44-10	3.3∨	32	1			_
EPM3064ALC44-4	3.3V	64				
EPM3064ALC44-7	3.3V	64				
EPM3064ALC44-10	3.3V	64				
EPM3064ATC44-4	3.3V	64				
EPM3064ATC44-7	3.3V	64				
EPM3064ATC44-10	3.3V	64				
EPM3064ATC100-4	3.3V	64	_			
EPM3064ATC100-7	3.3V	64				
EPM3064ATC100-10	3.3∨	64				~
- Companion device						
			Ψ.			
HardCopy II:						
HardCopy II:	JardCanu II.a					

Next...

Výber EDA (Electronic Design Automation) nástrojov. Pre simuláciu budeme používať ModelSim-Altera (domáca inštalácia) alebo ModelSim (školská inštalácia) :

New Project V	/izard: EDA Tool Settings [page 4 of 5]	X
Specify the ot	ner EDA tools in addition to the Quartus II software used with the project.	
Design Entry	/Synthesis	
Tool name:	<none></none>	
Format:		
🗖 Bun this	tool automatically to synthesize the current design	
- Simulation -		-
Tool name:	ModelSim-Altera	
Format:	VHDL	
🔲 Run gat	e-level simulation automatically after compilation	
– Timing Anal	sis	_
Tool name:	<none></none>	
Format:		
🗖 Bun this	tool automatically after compilation	
	<back next=""> Finish Storno</back>	

(Pozor!!! Toto dialogove okno je z verzie Quartusu 7.2, v školskej inštalácii je Quartus verzie 6.1 a vyzerá inak, avšak, filozofia výberu ModelSimu sa nemení.) Next..

New Project Wizard: Summa	iry [page 5 of 5]
When you click Finish, the projec	t will be created with the following settings:
Project directory:	
D:/= PHD =/= VYUKA =-	-/PL0_2008/cv3_basic_logic/
Project name:	cv3_basic_logic
Top-level design entity:	cv3_basic_logic
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	MAX3000A
Device:	EPM3064ATC44-10
EDA tools:	
Design entry/synthesis:	<none></none>
Simulation:	ModelSim-Altera (VHDL)
Timing analysis:	<none></none>
Operating conditions:	
Core voltage:	3.3V
Junction temperature range:	0-85 °C
	< Back Next > Finish Storno

Finish.

Teraz vytvoríme popis jednoduchej logiky (Hradlo AND) v jazyku VHDL. Horné menu: File: New: VHDL File

New	×
Device Design Files Other Files AHDL File Block Diagram/Schematic File EDIF File SOPC Builder System State Machine File Verilog HDL File VHDL File	
OK Cancel	
OK.	

Na začiatku kódu je nutné uviesť knižnice, v ktorých sú definované napr. typy signálov, ktoré nie sú štandardne zahrnuté vo VHDL:

```
library ieee;
use ieee.std_logic_1164.all;
```

Budeme používať knižnicu IEEE (Institute of Electrical and Electronics Engineers). Z nej použijeme definíciu štandardnej logiky "ieee.std_logic_1164" (viac na prednáške). Túto knižnicu budeme používať vždy.

Popis Entity. Entitu si môžeme predstaviť ako elektronickú logickú súčiastku, ktorá má svoje meno, vstupy a výstupy. Tento popis musí obsahovať každý VHDL kód, ktorým popisujeme hardvér, ktorý chceme syntetizovať v CPLD alebo FPGA.

```
entity cv3_basic_logic is
   port(
      my_input : in std_logic_vector (1 downto 0);
      my_output : out std_logic
   );
end cv3_basic_logic;
```

entity, is, port, in, out, downto, end sú klúčové slová a v editore sú zvýraznené modrou farbou. "cv3_basic_logic" je názov entity. "my_input" je vstup, v tomto prípade dvojbitový vektor. "my_output" je výstup, v tomto prípade jeden bit.

Ďalej pokračujeme popisom architektúry. Vo vnútri architektúry bude popis logiky, ktorá bude syntetizovaná. Aj keď VHDL umožňuje použitie viacerých architektúr, v rámci jedného VHDL súboru, vždy budeme používať len jednu.

```
architecture arch of cv3_basic_logic is
begin
end arch;
```

architecture, of, is, begin, end sú kľúčové slová. "arch" je názov architektúry. Riadok "end arch" je posledným riadkom v súbore a zvyšný kód budeme písať medzi "begin" a "end arch".

Pokračujeme popisom samotnej logiky. Keďže chceme realizovať hradlo AND priradíme výstupu AND kombináciu vstupného vektora:

```
my_output <= my_input(1) and my_input(0);</pre>
```

symbol <= sa používa na priradenie jedného signálu k druhému signálu. Okrem AND môžeme použiť OR pre logický súčet, alebo NOT pre logickú negáciu.

Súbor uložíme ako cv3_basic_logic.vhdl.

```
library ieee;
use ieee.std_logic_1164.all;
entity cv3_basic_logic is
  port(
    my_input : in std_logic_vector (1 downto 0);
    my_output : out std_logic
  );
end cv3_basic_logic;
architecture arch of cv3_basic_logic is
begin
  my_output <= my_input(1) and my_input(0);</pre>
```

end arch;

Skompilovať a priradiť piny podľa postupu z predošlého cvičenia. Súbor je možné odsimulovať v Quartuse podľa postupu uvedenom v predošlom cvičení.

Simulácia v ModelSime.

Filozofia: ModelSim je externý EDA program, ktorý buď treba Quartusom zavolať (spolu s konfiguráciou simulácie, ktorú Quartus za nás urobí), alebo spustiť samostatne (tu bude treba konfiguráciu popísať skriptom). Na tomto cvičení bude uvedený postup, ako spustiť simuláciu v ModelSime prostredníctvom Quartusu. Pre simuláciu bude treba simulovanej logike vnútiť vstupy, ktoré nezadáme graficky, ale VHDL kódom. Takýto vstup je pomerne jednoduché vytvoriť a editovať, zvlášť pre simulácie rozsiahlych dizajnov.

Vytvoríme simulačný VHDL skript (tzv. test bench) Horné menu: File: New: VHDL File:

```
library ieee;
use ieee.std_logic_1164.all;
entity stim_cv3_basic_logic is
end stim_cv3_basic_logic;
```

```
architecture stim of stim_cv3_basic_logic is
  component cv3_basic_logic is
  port(
    my_input : in std_logic_vector (1 downto 0);
    my_output : out std_logic
  );
  end component;
  signal input_data : std_logic_vector (1 downto 0);
  signal output_data : std_logic;
begin
  the_cv3_basic_logic: cv3_basic_logic
    port map (
      my_input => input_data,
      my_output => output_data
    );
  stimulus: process
    begin
      input_data <= "00";</pre>
      wait for 100ns;
      input data <= "01";</pre>
      wait for 100ns;
      input_data <= "10";</pre>
      wait for 100ns;
      input_data <= "11";</pre>
      wait for 100ns;
      input_data <= "00";</pre>
      wait;
    end process stimulus;
```

end stim;

Súbor uložíme ako stim_cv3_basic_logic.vhdl

Nakonfigurujeme Quartus II pre priame spúšťanie simulácie v ModelSime:

Zadáme cestu k modelsim.exe:

Horné menu: Tool: Options: General: EDA Tool Options Dvojklikom na ModelSim-Altera (pre domácu inštaláciu) alebo na ModelSim (pre školskú inštaláciu) a nasledným kliknutím na "…" zadáme cestu k modelsim.exe

ategory:		
🖃 General 🔼 🔼	EDA Tool Options	
EDA Tool Options Internet Connectivity License Setup	NativeLink options Specify the location of the	NativeLink executable for each third-party EDA tool:
Global User Libraries (All Proje	EDA Tool	Location of Executable
Assignment Editor Colors Colors Fonts Dock/Symbol Editor Colors Fonts Colors Colors Fonts Colors Gotto Colors Colors Colors Cons Colors Cons Colors Co	LeonardoSpectrum Precision Synthesis Synplify Synplify Pro Active-HDL ModelSim ModelSimAltera NCSim	< double-click to change path > < double-click to change path > D:\aterAV2sp3\modelsim_ae\win32 < double-click to change path >
Netlist Viewers TL Viewer Colors Fonts	Use NativeLink with a	Synplify/Synplify Pro node-locked license
Pin Planner Fonts Programmer		
Report Window Colors		
- Fonts		

Nakonfigurujeme simuláciu:

Horné menu: Assignments: EDA Tool Settings: Simulation

V Tool name bude uvedený ModelSim-Altera (domáca inštalácia) alebo ModelSim (školská inštalácia)

V EDA Netlist Writer Options: Output Directory zadajte cestu k projektovému adresáru. V NativeLink settings zaškrtnite Compile test bench a kliknite na tlačidlo Test Benches...



New...

New Test Bench Settings			
Create new test bench settings.			
Test bench name: stim_cv3_basic_logic			
Top level module in test bench: stim_cv3_t	pasic_logic		
Design instance name in test bench: the_c	v3_basic_logic		
Simulation period © Run simulation until all <u>v</u> ector stimuli ar	e used		
C End simulation at:	•		
Test bench files			
Eile name:			Add
File name	Library	HDL Version	<u>R</u> emove
			Цр
			Down
			Properties
		OK	Cancel

Test bench name: stim cv3 basic logic

Top level module in test bench: stim_cv3_basic_logic Design instance name in test bench: the_cv3_basic_logic

Vložime test bench súbor kliknutím na ...



Kliknete na Add

New Test Bench Settings			
Create new test bench settings.			
Lest bench name: [stim_cv3_basic_logic Top level module in test bench: [stim_cv3_t] Besign instance name in test bench: [the_cv3_b] Simulation period © Put simulation unbial subjector stimulation [misimulation at:	pasic_logic v3_basic_logic e used		
Test bench files Eile name: File name stim_cv3_basic_logic.vhd	Library	HDL Version Default	Add <u>R</u> emove <u>Up</u> <u>Down</u> <u>Properties</u>
		OK	Cancel
OK.			
OK.			

OK.

Prekompilujeme celý dizajn.

Spustíme funkčnú simuláciu: Horné menu: Tools: EDA Simulation Tool: Run EDA RTL Simulation

M wave - default		
<u> Eile Edit V</u> iew Insert F <u>o</u> rmat <u>T</u> ools <u>W</u> indow		
] 🗅 🚅 🖬 🚭 🐰 🐚 🛍 🗅 😂 🗛	E S ♦ ₩ 2 X ↑ E 10 ∞ ± L E 7 0 % M S × E ±	۲.
💽 🔜 🍳 🍳 🕵 🗱 头 🌋		
ra⊷ /stim_cv3_basic_logi 00 ↓ /stim_cv3_basic_logi 0		
Now 400000 ps Cursor 1 0 ps	0 ps	
0 ps to 566719 ps	Now: 400 ns Delta: 1	1.

Prekompilujeme celý dizajn.

Spustíme časovú simuláciu: Horné menu: Tools: EDA Simulation Tool: Run EDA Gate Level Simulation



Dizajn nahrajte do dosky do dosky CPLD_KIT.

Námet na samostatnú prácu:

Realizujte funkčnú a časovú simuláciu v ModelSime prevodníka BCD na 7 segmentový displej. Projekt nahrajte do dosky.

```
Prevodník:
library ieee;
use ieee.std_logic_1164.all;
entity comb_unit is
port(
    data : in std_logic_vector (3 downto 0);
    seg : out std_logic_vector (6 downto 0)
    );
end comb unit;
architecture a of comb unit is
begin
  with data select
    seg <=
      "0000001" when "1111", -- 0
      "1001111" when "1110", -- 1
      "0010010" when "1101", -- 2
      "0000110" when "1100", -- 3
      "1001100" when "1011", -- 4
      "0100100" when "1010", -- 5
      "0100000" when "1001", -- 6
      "0001111" when "1000", -- 7
      "0000000" when "0111", -- 8
      "0000100" when "0110", -- 9
      "0001000" when "0101", -- A
      "1100000" when "0100", -- b
      "1110010" when "0011", -- c
      "1000010" when "0010", -- d
      "0110000" when "0001", -- e
      "0111000" when "0000", -- f
      "11111111" when others; -- else
end a;
Test Bench:
library ieee;
use ieee.std logic 1164.all;
entity stim_comb_unit is
end stim_comb_unit;
architecture stim of stim_comb_unit is
  component comb_unit is
  port(
      data : in std_logic_vector (3 downto 0);
      seg : out std_logic_vector (6 downto 0)
      );
```

```
end component;
  signal stim_data : std_logic_vector (3 downto 0);
  signal out_data : std_logic_vector (6 downto 0);
begin
  the comb unit: comb unit
    port map (
      data => stim_data,
      seg => out_data
     );
  stimulus: process
    begin
       stim_data <= "1111"; wait for 100ns;</pre>
       stim_data <= "1110"; wait for 100ns;</pre>
       stim_data <= "1101"; wait for 100ns;</pre>
       stim_data <= "1100"; wait for 100ns;</pre>
       stim_data <= "1011"; wait for 100ns;</pre>
       stim_data <= "1010"; wait for 100ns;</pre>
       stim data <= "1001"; wait for 100ns;</pre>
       stim_data <= "1000"; wait for 100ns;</pre>
       stim_data <= "0111"; wait for 100ns;</pre>
       stim data <= "0110"; wait for 100ns;</pre>
       stim_data <= "0101"; wait for 100ns;</pre>
       stim_data <= "0100"; wait for 100ns;</pre>
       stim_data <= "0011"; wait for 100ns;</pre>
       stim_data <= "0010"; wait for 100ns;</pre>
       stim_data <= "0001"; wait for 100ns;</pre>
       stim_data <= "0000";</pre>
       wait;
    end process stimulus;
```

end stim;