Simulation of a VHDL code for a pipelined implementation of CORDIC

- Copy the VHDL file Cordic.Vhd for CORDIC algorithm verification from S:\TN\E\027_Digital_Kommunikationselektronik\CORDIC VHDL code\ to your own directory.
- Create a new FPGA project by clicking on File » New » Project » FPGA Project and rename the new project file by clicking on File » Save Project As.
- Add the VHDL file Cordic.Vhd to the project by right clicking the project file name in the Projects panel and selecting Add Existing to Project
- 4. Open Cordic.Vhd by double clicking the file name Cordic.Vhd.
- 5. Click on **Design » Create VHDL Testbench**. A testbench Test_cordic.VHDTST is created and opened as follows.

```
-- VHDL Testbench for cordic
-- 2006 10 23 11 57 33
-- Created by "EditVHDL"
-- "Copyright (c) 2002 Altium Limited"
_____
Library IEEE;
Use IEEE.std_logic_1164.all;
      IEEE.std logic textio.all;
Use
Use
      STD. textio.all;
_____
entity Testcordic is
end Testcordic;
                 _____
architecture stimulus of Testcordic is
   file RESULTS: TEXT open WRITE MODE is "results.txt";
   procedure WRITE RESULTS(
       angle: std logic vector(7 downto 0);
       clk: std logic;
       datax: std_logic_vector(11 downto 0);
       datay: std logic vector(11 downto 0);
       res: std logic;
       x n: std logic vector(11 downto 0);
       y_n: std_logic_vector(11 downto 0)
   ) is
       variable 1 out : line;
   begin
       write(l out, now, right, 15);
       write(l out, angle, right, 9);
       write(l out, clk, right, 2);
       write(l_out, datax, right, 13);
       write(l_out, datay, right, 13);
       write(l_out, res, right, 2);
       write(l_out, x_n, right, 13);
       write(l_out, y_n, right, 13);
       writeline(RESULTS, l out);
   end procedure;
```

```
component cordic
```

```
port (
           angle: in std_logic_vector(7 downto 0);
           clk: in std logic;
           datax: in std_logic_vector(11 downto 0);
           datay: in std logic vector(11 downto 0);
           res: in std logic;
           x_n: out std_logic_vector(11 downto 0);
           y_n: out std_logic_vector(11 downto 0)
        );
    end component;
    signal angle: std_logic_vector(7 downto 0);
    signal clk: std logic;
    signal datax: std_logic_vector(11 downto 0);
    signal datay: std_logic_vector(11 downto 0);
    signal res: std logic;
    signal x n: std logic vector(11 downto 0);
    signal y n: std logic vector(11 downto 0);
begin
    DUT:cordic port map (
       angle => angle,
       clk => clk,
       datax => datax,
       datay => datay,
       res => res,
        x n \Rightarrow x n,
        y n => y n
    );
    STIMULUS0:process
    begin
        -- insert stimulus here
       wait;
    end process;
    WRITE RESULTS (
       angle,
       clk,
       datax,
       datay,
       res,
       x_n,
        y_n
    );
end architecture;
  _____
```

6. In the testbench, insert the some statements before the STIMULUS0:**process** and some statements in the process as follows:

```
datax <= B"01000000000";
datay <= B"000000000000";
wait for 2 ns;
angle <= B"00110000";
datax <= B"011000000000";
datay <= B"00000000000";
wait for 2 ns;
end process;
```

The statement "clk <= not clk after 1 ns;" defines the behavior of a simulated clock signal. The time 1 ns is one half of the time period for the clock. The time 1 ns is chosen for convenience and is not a realistic value for the FPGA. The frequency of the clock may be changed by changing the time to other values. Notice that this statement can only be used for simulation. The statement "res <= '0', '1' after 2 ns;" sets the res signal to '0' at the beginning and sets it to '1' after 2 ns. In the process STIMULUSO, the values of input signals are changed after 2 ns. You can add other input signal values in the process.

7. Add an initial value to the signal clk as follows:

```
signal clk: std_logic := '0';
```

After the changes, the testbench looks like the following:

```
_____
-- VHDL Testbench for cordic
-- 2006 10 23 11 57 33
-- Created by "EditVHDL"
-- "Copyright (c) 2002 Altium Limited"
        _____
Library IEEE;
Use IEEE.std logic 1164.all;
     IEEE.std logic textio.all;
Use
     STD. textio.all;
Use
                    _____
   _____
entity Testcordic is
end Testcordic;
    _____
architecture stimulus of Testcordic is
   file RESULTS: TEXT open WRITE MODE is "results.txt";
   procedure WRITE RESULTS (
      angle: std logic vector(7 downto 0);
      clk: std logic;
      datax: std_logic_vector(11 downto 0);
      datay: std_logic_vector(11 downto 0);
      res: std logic;
      x_n: std_logic_vector(11 downto 0);
      y_n: std_logic_vector(11 downto 0)
   ) is
      variable 1 out : line;
   begin
      write(l out, now, right, 15);
      write(l out, angle, right, 9);
```

```
write(l out, clk, right, 2);
        write(l out, datax, right, 13);
        write(l out, datay, right, 13);
        write(l out, res, right, 2);
        write(l_out, x_n, right, 13);
        write(l_out, y_n, right, 13);
        writeline(RESULTS, l_out);
    end procedure;
    component cordic
        port (
            angle: in std logic vector(7 downto 0);
            clk: in std logic;
            datax: in std_logic_vector(11 downto 0);
            datay: in std_logic_vector(11 downto 0);
            res: in std logic;
            x n: out std_logic_vector(11 downto 0);
            y n: out std logic vector(11 downto 0)
        );
    end component;
    signal angle: std_logic_vector(7 downto 0);
    signal clk: std logic := '0';
    signal datax: std_logic_vector(11 downto 0);
    signal datay: std logic vector(11 downto 0);
    signal res: std logic;
    signal x n: std_logic_vector(11 downto 0);
    signal y_n: std_logic_vector(11 downto 0);
begin
    DUT:cordic port map (
        angle => angle,
        clk => clk,
        datax => datax,
        datay => datay,
        res => res,
        x n \Rightarrow x n,
        y_n => y_n
    );
     clk <= not clk after 1 ns;
     res <= '0', '1' after 2 ns;
    STIMULUS0:process
    begin
        -- insert stimulus here
        angle <= B"00100000";
        datax <= B"01000000000";</pre>
        datay <= B"00000000000";</pre>
        wait for 2 ns;
        angle <= B"00110000";</pre>
        datax <= B"01100000000";</pre>
        datay <= B"00000000000";
        wait for 2 ns;
    end process;
    WRITE RESULTS (
        angle,
        clk,
        datax,
        datay,
        res,
        x n,
        y_n
    );
```

8. Select the simulation tool and the testbench document by right clicking the project file name in the **Projects** panel and selecting **Simulation** tab from within the **Project Options** dialog.

S Altium Designer 6.4 - C:\Qin\VHDL Simulation\	Cordic_VHDL_Simulation\Test_cordic.VHDTST - Cordic_simu.PrjFpg, Licensed to LiTH-ITN	
DXP Elle Edit View Project Design Iools	imulator <u>W</u> indow <u>H</u> elp	C:\Qin\VHDL Simulation\Cordic_VHDL_: • 😋 • 🔿 - 🐢
D 🐸 🖬 🥵 🔕 🔌 🕹 🛍 🦛 🖤 🖤	年年 当 曽 [2] 日	
Projects 🗸 🖉 🗙 🛐	Cordic./Vhd Cordic.VHDTST	Favo
Workspace1.DsnWrk Workspace	writeline(RESULTS, 1_out);	vites 🔊
Cordic_simu.PriFpg Project	end procedure;	- F
⊙ File View ○ Structure Editor	component cordic	board
🗏 🐟 Cordic_simu.PrjFpg	angle: in std_logic_vector(7 downto 0);	5
Cordic.Vhd	Options for FPGA Project Cordic_simu.PrjFpg	
	Error Reporting Connection Matrix Comparator Options Multi-Channel Default Prints Search Paths Synthesis Simulation	Parameters
	Tool	
	DXP Simulator	
	Design	
	Testbench Document	
	Select a testbench document	<u> </u>
	Top-Level Entity/Module/Configuration Top-Level Architecture	
	SDF Options	
	SDF Instance SDF Uprimization	•
	SDF Instance Path	
	Options	
	VHDL Standard VHDL93	
	Verilon Standard Verilon2001	
	Set To Installation Defaults	OK Cancel
	datax <= 8"011000000000":	
	datay <= B"00000000000";	
	wait for 2 ns; end process;	
	angle,	
	clk, datay	
Files Projects (Navigator		Mask Level Clear
105:15	Inset	System Design Compiler Help Instruments VHDL
🛃 start 🔪 🗁 VHDL Simulation 🖓 Sin	ulation of VHDL co S Altium Designer 6.4	第3章 会 公 (公) 11:00

- 9. Select the simulation tool **DXP Simulator** from the drop-down list for **Tool**.
- 10. Select the testbench document "Test_cordic.VHDTST" from the drop-down list for **Testbench Document** and click the **OK** button.
- 11. Initiate a simulation session by selecting **Simulator** » **Simulate** from the menu.
- 12. When you first run a simulation from a testbench, the following window will be shown. Whilst performing this process, you may see an error appearing in the **Messages** panel with the message: "Unbounded instance DUT of component Cordic". Do not be concerned as this is normal when you first run a simulation.

Project Compile Order
Smart compile has determined the correct compile order of the source documents in the project. Would you like to reorder your project to reflect these changes?
Filename
Test_cordic.VHDTST
Lordic. Vhd
Yes No

13. If the correct compile order is shown as in the above window, click the **No** button. The following window will be shown.

Choose Top Level		? 🛛
Unit	Test_cordic	▼
Entity/Configuration	Testcordic	~
Architecture	stimulus	v
	(OK Cancel

14. Click the **OK** button. The following window will be shown.

E	lit Simulation Signals		? 🛛
E	Block Name 🔼		
W	atch Name	Show Wave	Enabled
Ξ	Block Name :		
	y_n	 Image: A start of the start of	 Image: A start of the start of
	x_n	 Image: A start of the start of	 Image: A start of the start of
	res		✓
	datay	 Image: A start of the start of	
	datax	 Image: A start of the start of	
	clk		
	angle		
Ξ	Block Name : DUT		
	Z		
	у		
	x		
5	Show on startup	(Done

- 15. You can add signals to the display of the simulation by setting a tick for the signals in **Show Wave**. The signals should also be **Enabled**. Click on the **Done** button. If you need to change the signals for the display later, this window can be accessed by clicking **Simulator » Signals**.
- 16. After the above step, the following window is shown. The following text explains details of some functions.
 - The "plus" icon next to the bus name indicates a bus signal. Clicking on this icon will expand the bus into its individual signals for closer inspection.
 - The time cursor (indicated by the purple vertical bar) can be dragged along the time axis via the mouse. The current position of the cursor is provided in the time bar across the top of the display. The values of signals under the time cursor will be shown in the column **Value**.
 - Zooming in or out is achieved by pressing the Page Up or Page Down keys respectively.
 - The display format of the individual signals can be altered via the menu item **Tools » Format and Radix**.

	S Altium Designer 6.4 - C:\Qin\VHDL S	imulation\Co	ordic_VHDL_Simu	ılation\C	ordic_simu.SO * - Cordic_simu	I.PrjFpg, Licensed to LiTH-ITN		- 2 🛛
	Dyp Elle Edit View Project Iools Simulator Window Help							_VHDL] • 🔾 • 🔿 - 🌴
	Projects VX 🗑 Cordic. Vha 🔯 Test_cordic. VAXST 🔛 Cordic_sinu.SO *							Favo
	Workspace1.DsnWrk Works			0			• 90•••• ns	
	Cordic_simu.PrjFpg Proj	ect	Name	Value	0.000ms			Cipb
	File View O Structure Editor	<u>></u>		(0, 0, 0	(0,0,0,0,0,0)			oard
	Cordic_simu.PrjFpg *	1	DUT/y	{0, 0, 0	(0, 0, 0, 0, 0, 0)			
	Cordic.Vhd	2	⊡ DUT/Z	{0, 0, 0	(10, 0, 0, 0, 0, 0)			ries
	Generated	4	≞-y_n ⊡-x_n	222	277			
	/	5	res	U	/222			
		7	erdatay ∎-datax	222	077			
		8	— clk	U	(32			
Dum	Simulation	9 8	e angle	11	<u>N//</u>			
Kull	Simulation							
To A	A Time		Charletter				7	
			Time is 0.00	fs(+[])				
			Hissarchiel C					
			- Testo	ordic				
			⊡ ⊕ DL	IT				
			Local Signal	2	Туре	Value		
			€• <mark>√⊡</mark> ⊻_n		array of std_logic array of std_logic	{U, U, U		
			VII res		std_logic	U'		
					array of std_logic array of std_logic	{\U; U; U		
			√⊡ clk		std_logic array of std_logic	'0' {11' 11' 11' 11' 11' 11' 11' 11' 11'		
			E the sign		andy of the_logic	(0,0,0,0,0,0,0,0,0)		
			Show Sign	nals 🔲 :	Show Variables	/ /		
			Simulation	X Stora	ge manager / messages / Output	<u></u>		
								~
	Files Projects Navigator							2
	32.708ns	Low		_			System Design Compiler Help	Instruments VHDL >>
	Start 👘 🗁 VHDL Simulation	💾 Simul	ation of VHDL co	🛛 🔕 Alti	um Designer 6.4			🔷 🔿 🛃 😓 11:46

17. Run the simulation to a time by clicking the button **Run Simulation To A Time**. The following window will be shown. You may change the time to 20 ns and click the **OK** button.

Enter time to run to	? 🔀
Time Step	ns 🗸
	OK Cancel

18. After the simulation, you can change the zoom and the time cursor to check the signal values.

S Altium Designer 6.4 - C:\Qin\VHDL Simulat	tion\Cordic_VHDL_Simu	lation\Cordic_simu.SO * - Cordi	c_simu.PrjFpg, Licensed to LiTH-ITN			
🚺 Dg9 Ejle Edit Vjew Project Iools Simulator Vjindov Help C:\(a)n\/HDLSimulation\/Cordic_VHDL3* 💿 • 🕤 • 💠						
I 🗋 💕 🖬 🎒 🛕 🗶 🔍 🔍 🛔 🐚 🛛 4	4 🕨 🕪 📝 🚈	N N 12 10 🖬 🖓 🖓 🕅	1 🕞 II 💷 🞦			
Projects 👻 🗸 🖉 🗙	🗑 Cordic.Vhd 😡 Test_	cordic.VHDTST 🔡 Cordic_simu.SO *		Fav		
Workspace1.DsnWrk Workspace			0 10	· · · · · · · ns		
Cordic_simu.PrjFpg Project	Name	Value		15.821ns		
⊙ File View O Structure Editor 💽 🔯 ▼				- Ba		
🖃 📣 Cordic_simu.PrjFpg * 🛛 🖺	0 ⊞-DUT/x	{1536, 1024, 768, 1408, 1608, 1475}				
Gourde Documents	1 ⊡-DUI/y 2 ⊡-DUI//2	(U, 1U24, 23U4, 896, 1944, 815) (48, 13, 23, 1, 2, 3)				
Va Test cordic.VHDTST	3 ⊞ y n	32F	(000	X734 X32F X734 X32F		
	4 ⊞ x_n	5C3	(000	X6CT X5C3 X6CT X5C		
	5 res	1				
	6 🕀 datay	000	(000			
	7 H datax	1	<u>400 X600 X400 X600 X400 X600</u>	<u></u>		
	9 ⊞ angle	30	20 X30 X20 X30 X20 X30	X20 X30 X20 X30		
	Simulation		* *			
	Time is 20.00					
	Hierarchial So	copes				
	E Testco	rdic T				
		•				
	Local Signals	Туре	Value			
		arvec	{48, -13, -23, -1, -2, 3} {0, 1024, 2204, 996, 1944, 916}			
		arvec	(1536, 1024, 768, 1408, 1608, 1475)			
	Show Sign	als Show Variables	Output / To-Do /			
	Sindicion					
Files Projects (Navigator /						
13.30bhs				System Design Compiler Help Instruments VHDL >>		
Start 🔁 VHDL Simulation	Simulation of VHDL co	👻 Altium Designer 6.4 🥂 🦄 4	Adobe Acrobat Stand	🔇 💕 🔁 12:26		

19. Click the Save button in the menu, the following dialog window will be shown. You can save the simulation result in a wave file.

Save [Cordic_s	imu.SO] As					? 🗙
Save in:	Cordic_VHDL_	Simulation	~	G 🦻	• 🖭 💐	
My Recent Documents Desktop My Documents	History ProjectOutputs					
My Computer	File name: Save as type:	<mark>Cordic_simu</mark> Wave files (*.so)			✓✓	Save Cancel
My Network					(Help

You can read the other simulation commands from Chapter 7 of the document "FPGA Design Training Module" in the file Training Module 5 FPGA Design.pdf in the directory S:\TN\E\027_Digital_Kommunikationselektronik\Altium Manuals and Tutorials\.

- 20. You can reset the simulator by clicking Simulator » Reset from the menu. Notice that you may only reset the simulator once. If it does not work after resetting the simulator. You must click Simulator » End to terminate the simulation. Then you should click Simulator » Simulate for further simulations.
- 21. You can try to change the statements in the stimulus process of the testbench and obtain other simulation results.

Notice that one can also simulate the VHDL code in your project previously created in Lab2. After opening your project, you should open a schematic document and select **Tools » Convert » Create VHDL Testbench** from the menu.

Reference:

"FPGA Design Training Module" in the file Training Module 5 FPGA Design.pdf in the directory S:\TN\E\027_Digital_Kommunikationselektronik\Altium Manuals and Tutorials\.