EEDG/CE 6370
Design and Analysis of Reconfigurable Systems
Homework 3
FPGA Soft IPs - CORDIC

1. Laboratory Objectives

- Understand how to use the IPs included in Quartus Prime's IP generator.
- Learn how to specify synthesis options and understand their impact on the QoR.
- Understand the impact of placement on the design using the Quartus's Chip Planner

2. Summary

In this lab you will learn to use Quartus's IP generator and configure a given IP based on different specifications.

3. Pre-lab

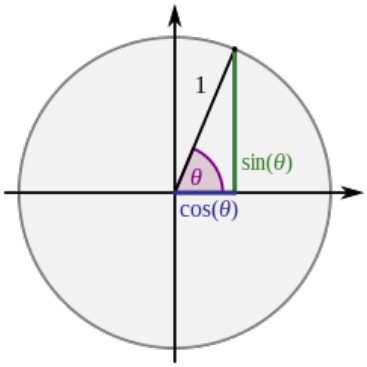
- Review the lecture slides that covers soft IPs
- Review Intel's CORDIC IP core user guide

4. Tool Requirements

- Quartus Prime
- Questa simulator
- DE1-SoC board

5. Sine and Cosine Calculation- CORDIC Algorithm

Trigonometric functions are often calculated in hardware using CORDIC: Coordinate Rotation Digital Computer



CORDIC uses simple shift-add operations for several computing tasks such as the calculation of trigonometric, hyperbolic and logarithmic functions, real and complex multiplications, division, square-root calculation Because of this it is often used in hardware implementations due to not having to use expensive functional units (less hardware resources)

6. Sin/Cos Design using IP Generator

Table 1 Sin/cose Parameters (from Intel's CODIC IP manual)

Parameter	Values	Description
Input data widths		
Fraction F	1 to 64	Number of fraction bits.
Width w	Derived	Width of fixed-point data.
Sign	signed or unsigned	The sign of the fixed-point data.
Output data widths		
Fraction	1 to 64, where $F_{OUT} \leq F_{IN}$	Number of fraction bits.
Width	Derived	Width of fixed-point data.
Sign	Derived	The sign of the fixed-point data.
Generate enable port	On or off	Turn on for `enable` signal.

7. System Overview

Figure 1 shows how the CORDIC function will be implemented on the DE1-SoC board. The 10 switches will be used for input and the results of the cosine (c) and the sine (s) will be displayed on the 10 LEDs. Key0 will be used as rest.

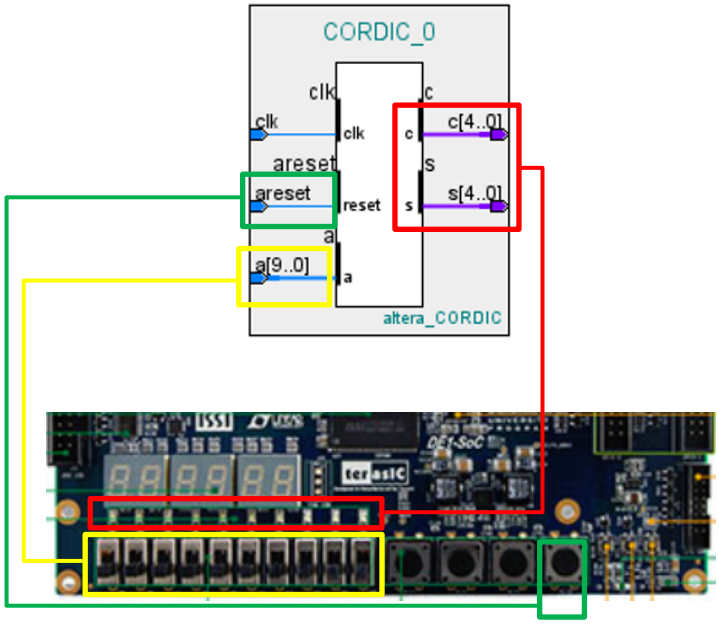


Figure 1 DE1-SoC board peripherals overview used to enter and display result of CORDIC IP

8. Design using IP Generator

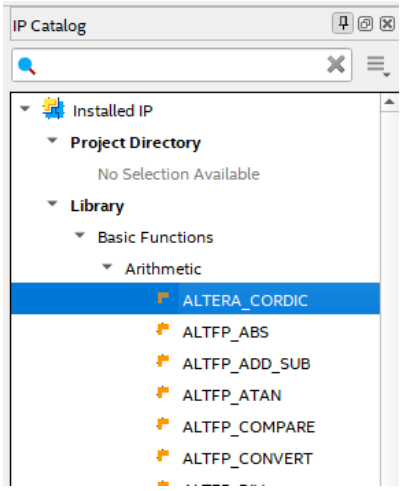
- Open Quartus Prime click in icon)

In case of Windows run the tools always as administrator (right

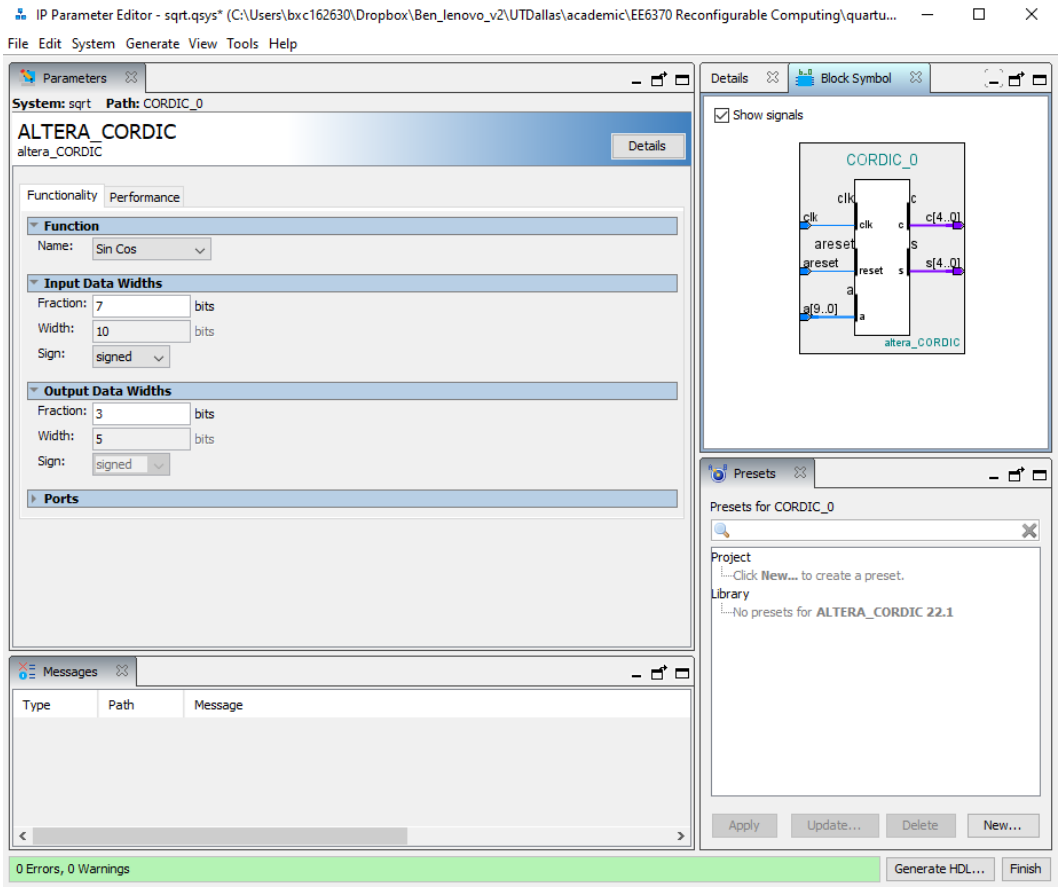
- Create a new project.
  File ➔ New Project Wizard → sincos

- Click next → empty project →
- Click next (do not add any files)
- Select Cyclone V 5CSEMA5F31C6 device.
- Click next and Finish.

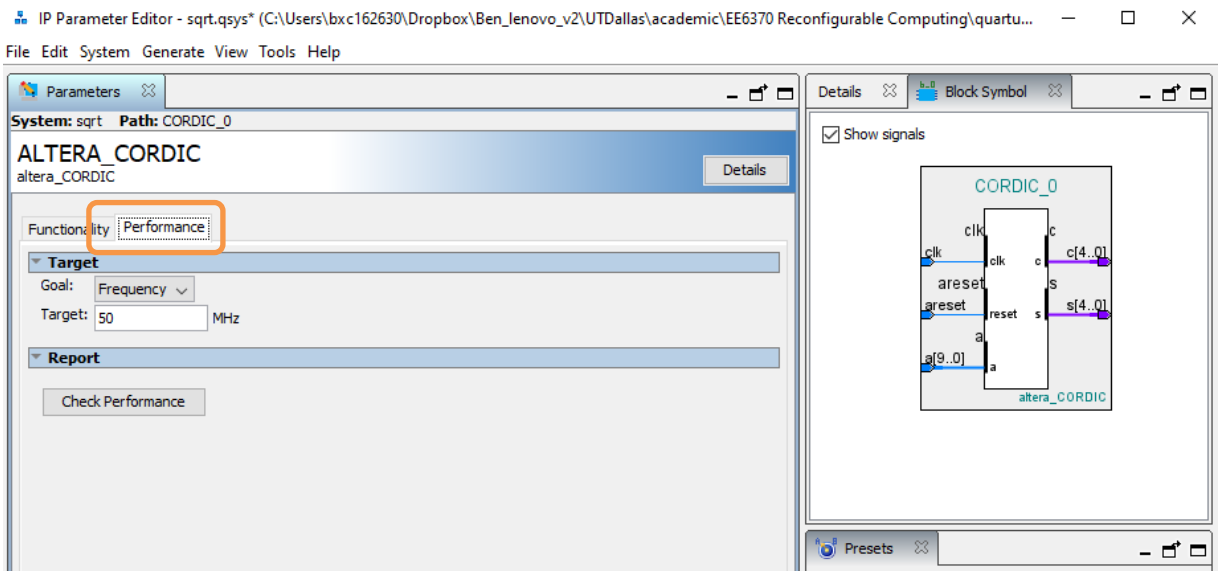
- Open the IP core generator (right pane of Quartus main window)→ Select ALTERA_CORDIC from the Basic Functions→Arithmetic this will open a window to name the IP.



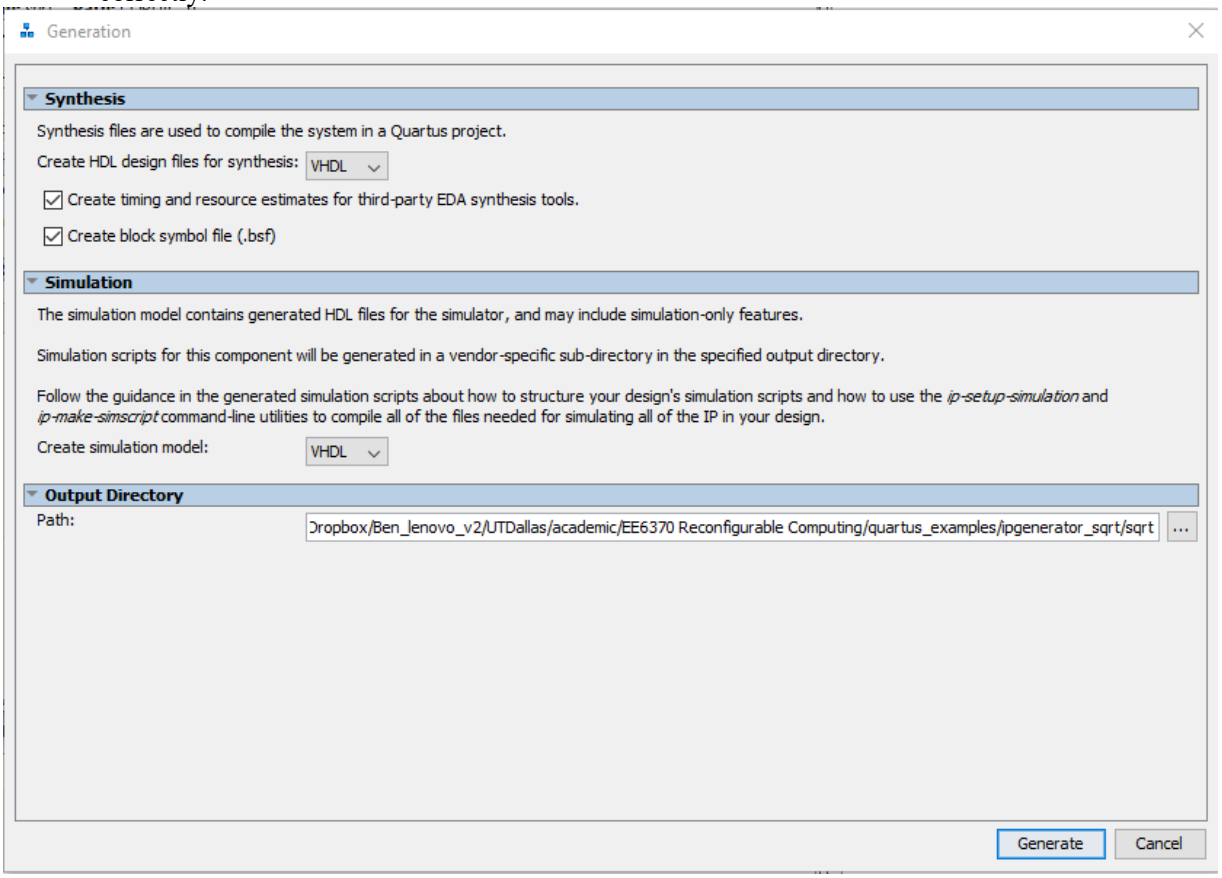
- Name the IP → sincos
- Configure the IP as shown below : 10 bits for the input, 5 bits for the outputs (restricted by the number of switches and LEDs that our board has.



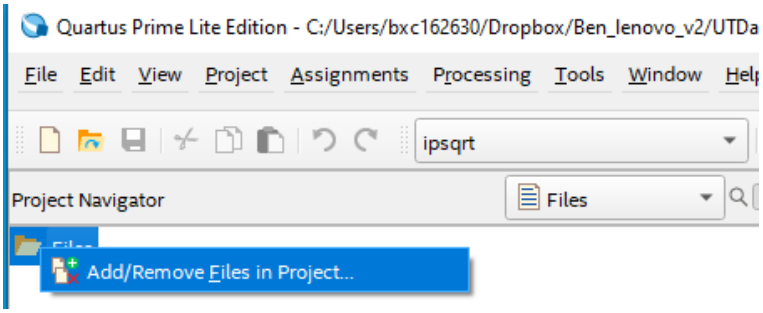
- In the performance pane set the clock frequency to 50Mhz (frequency of DE1-SoC board)



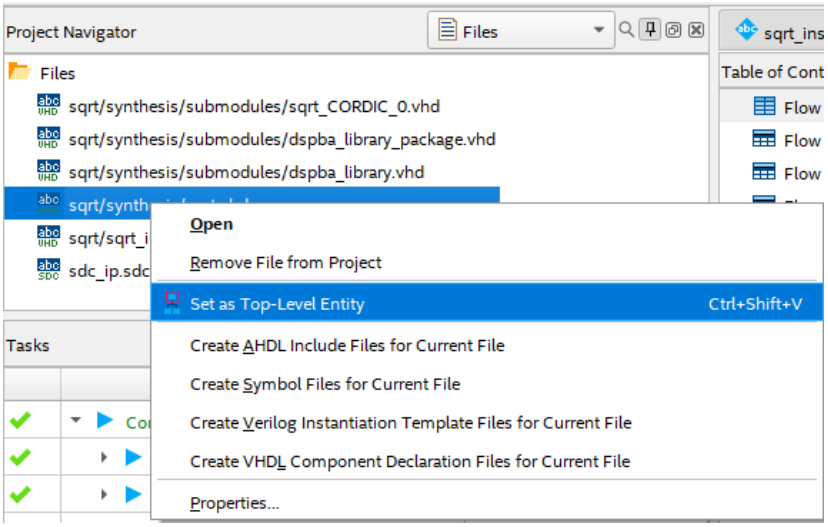
- Click on “Generate HDL”
- Specify either Verilog or VHDL and also generate a simulation model to verify that the IP works correctly.



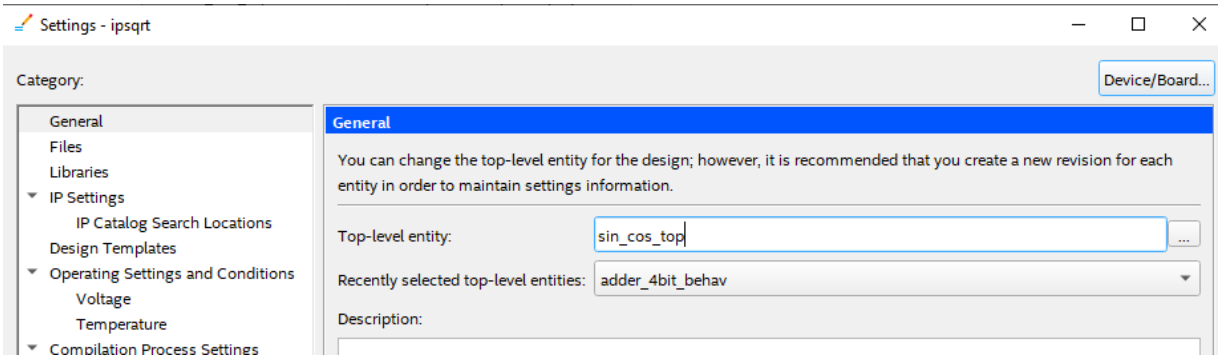
- The IP generator will store the new Verilog/VHDL files in sincos folder/synthesis and sincos/synthesis/submodules
- Add all of the files to the project



- Set the main file sincos.v/.vhd that instantiates the IP module as top-level entity



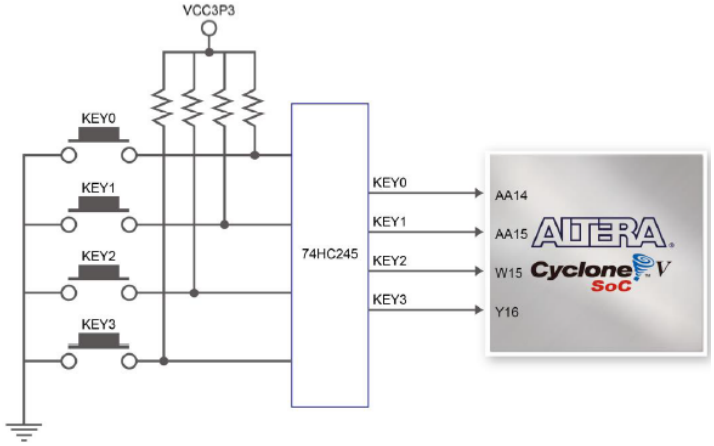
- If the file name does not match the entity name, set the top-level entity in settings→General->top-level entity.



- Create a new SDC file where you specify the FPGA clock running at 50MHz (File→New→SDC file)

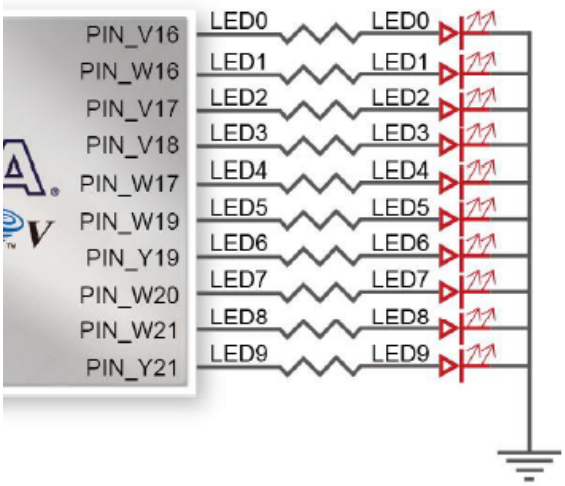
```
create_clock -name "clk" -period 2.000ns [get_ports {clk}]
derive_pll_clocks
derive_clock_uncertainty
```

- Connect the top entity IOs to the DE1-SoC board switches, keys and LEDs through the Pin Planner (assignment→Pin Planner) based on the datasheet given by Terasic



1. Rest connected to KEY0 (Pin AA14)

2. Switches to module's input



3. LEDs to module's output

Note: Be careful with the order to the Pins and MSB to LSB bits.

- Synthesize and generate the bitstream
- Connected the DE1-SoC board→ Auto-detect→ Select new .sof file and program FPGA.
- Check that the design works.

UT Dallas Honor Code

UT Dallas values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures.

Group assignments must be completed solely by the members of the group. Cross-group work is not allowed. Moreover, similar assignments have been offered before at UT Dallas and other universities. Any use of information from previous assignments is prohibited. The tutorials are to be taken individually. Failure to respect this rule constitutes dishonesty and is a direct violation of the University Honor Code.

[END]