FPGA PRIMER

-Cordic Implementation

PRE-REQUISITES

- Verilog or VHDL
- Basic knowledge of FPGA architecture
- Xilinx ISE and PicoBlaze
- Assembly language fundamentals

OBJECTIVE

To implement CORDIC based trigonometric computational unit. In simple terms, we will build a hardware to compute **sine** and **cosine** of given angles. But using normal methods like Taylor's expansion would require a lot of multiplications, which are not suitable for hardware implementation. In contrast the Cordic follows an iterative approach which takes much lesser time and hardware.

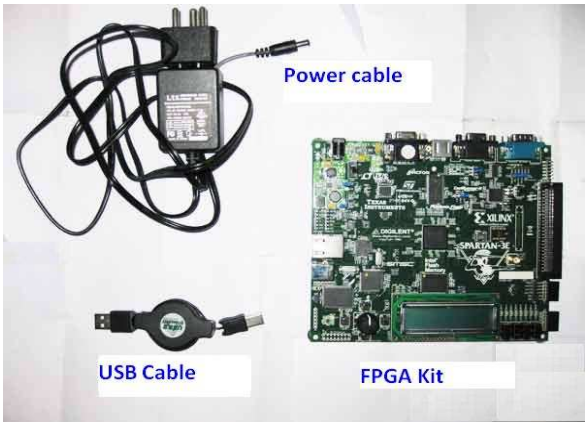
Please note that this tutorial is written in view of giving you all the typical steps involved in implementing **any program on FPGA**. But as a specific example I have used CORDIC. You can follow your own example.

The primary emphasis is given on making you familiar with the steps involved in working with FPGA kits. But I hope this would also be useful who are specifically working on Cordic based algorithms.

REQUIRED TOOLS

Hardware :Spartan - 3E starter kit
USB cable
Adapter

Software :Xilinx ISE - (preferably version 7 or above)



CORDIC BASICS

CORDIC-Coordinate Rotation Digital Computer

Let us not delve too much in to the inside complexities of Cordic, but only understand how it makes the computations easy compared to other methods.For understanding Cordic, please refer to my project report link or you may try Wikipedia. It is easy and derivable by hand. All one needs is basic high school math.

Understanding Cordic

Let me explain you the core idea of Cordic. I already mentioned that we need to eliminate multipliers(dividers) as much as possible while working on hardware. Well, how are we going to do that? Okay, here is a small exercise for you.

Try to multiply (say) 489 and 7. You may need a pen or a calculator.
The how about dividing 489 by 67. Even tougher..!!

Now try these 489*10 or 489 / 100.You get the answers on the spur!

So you realize that it is easy to perform computations if one of them is a power of 10(as we are in radix 10). **Similarly multiplying or dividing by powers of 2 in base 2 system is virtually adding or eliminating zeroes in the end**. So CORDIC takes advantage of this fact and limits all the multiplications that arise, only to the powers of 2. Any other constants that come up can be incorporated while initializing the values. Now being said that, you can easily understand the rest of it.

Our kit is still lying out there idle. Lets start making it busy!

FILES REQUIRED

NOTE: If you are trying to implement any other program, then you need to have a clear idea of your inputs and outputs. You should also assign the required pins by checking the FPGA kit manual.

If you want to write your own programs, then you may have to spend some more time before moving forward. But if you immediately want to see the FPGA kit spring up into life, then these are the files you need. You can download them and use. Follow the steps below.

audi.v
audi_rom.v
binary2bcd.v
audi.ucf

Wait ! You will need few more files.
(Please note that the code is not supplied with comments. You may face problems if you try to understand them.)

CONFIGURING LCD -PICOBLAZE

The major problem one faces while dealing with the FPGA kit is how to show the outputs. Of course it is clearly mentioned in the manual that we can use the LCD to display any useful information. Here comes the complexity. You need to get introduced to a new term called PICOBLAZE. It is a virtual microprocessor that comes with Xilinx kits.

In order to get the display working, we need to send the data in precise intervals or time delay. Getting the required delays using logic gates of FPGA is highly cumbersome. Where as PICOBLAZE provides us with fixed time delays for executing any of its commands. So we can make use of this property for controlling the date sent to display.

You need this folder KCPSM3 which can be downloaded from Xilinx web site in order to create the virtual microprocessor. If you don't want to mess up with it, then no problem, here are the final files you need. Just put these in the current working folder and add them to your project.

embedded_kcpsm3.v
kcpsm3.v
audi_lcd.v

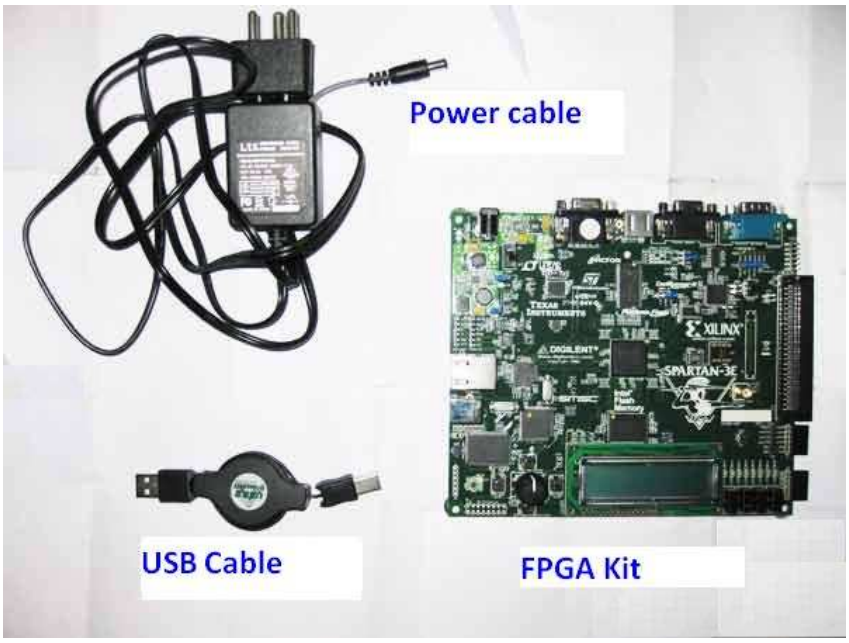
Note that these files are generated using assembly language code run with a program

that comes with KCPSM3 folder. These are necessary for the functioning of LCD screen.

Now lets get to the exact steps. If you have downloaded the above seven files, then you can simply follow these steps.

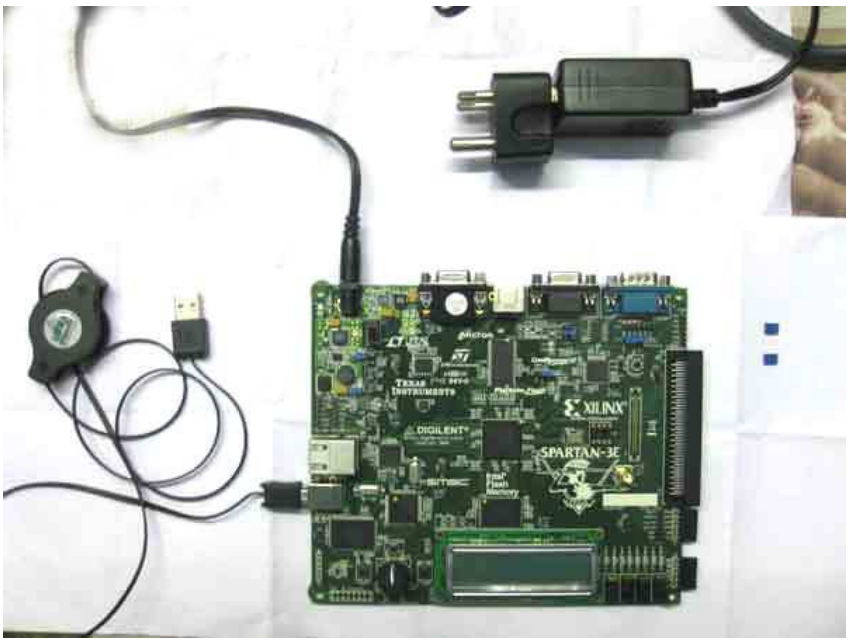
GET STARTED!

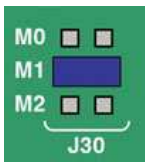
Check the Hardware



Connections:

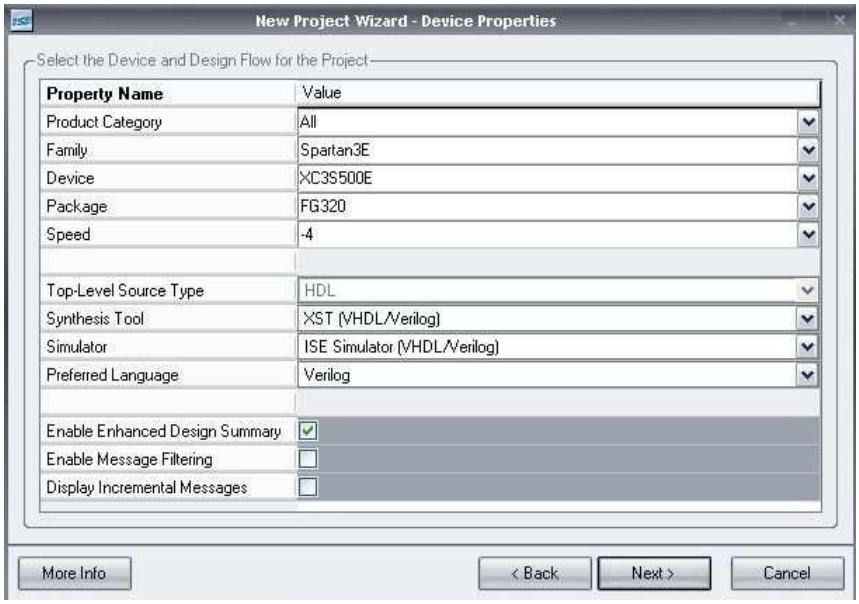
Remember to put the kit in USB mode by removing the jumper wires. The mode details are given in the manual. For Spratan-3E kit. Use the following settings. You can keep the power off.



Jumper settings: 

STEP 1

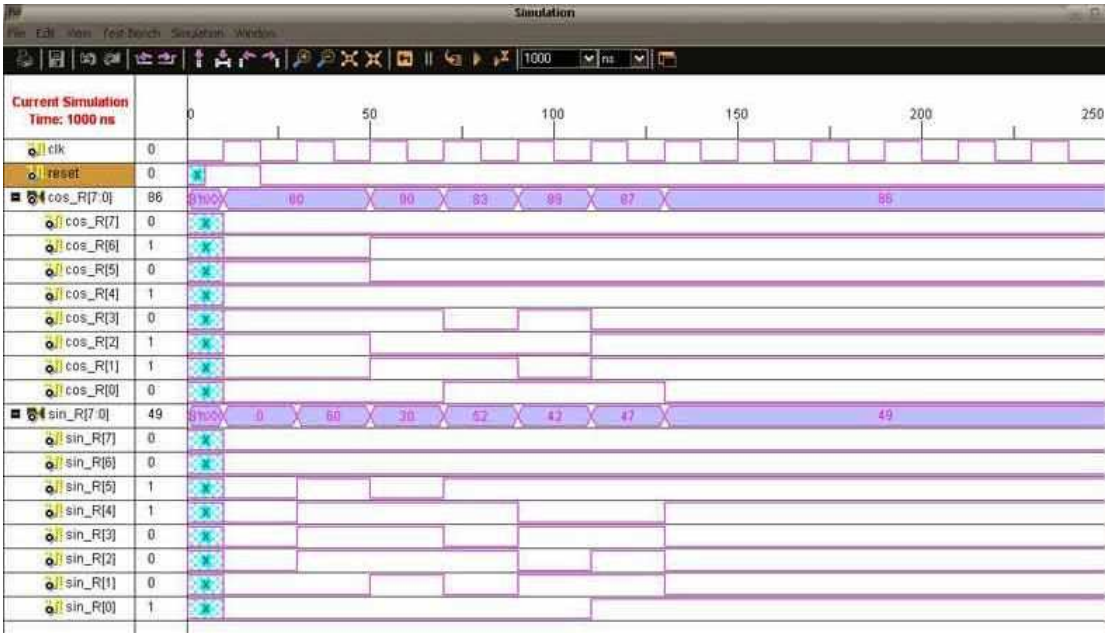
Start the Xilinx ISE -> Project navigator. Open new project and follow as shown.Enter the hardware specifications. For this example, I have used the specs as shown in the figure. You can find yours right on the board or in the user manual.



STEP 2
SIMULATION

Once the project is created. Just add all the above files to it. From the tab project -> add source.
After adding the files check for syntax errors. Then you can try simulating it.

But remember simulation is not going to be easy as the program includes delays of the order of several milliseconds, which would take a lot of time for simulators. So its always better to verify you program by directly implementing it on the hardware. Here is how a simulation looks for an input of 30 degrees.

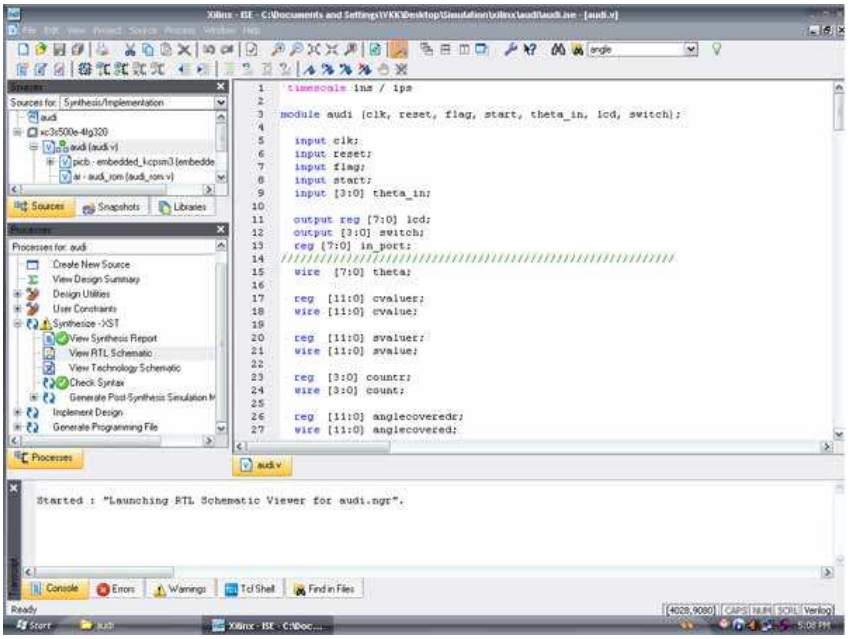


Cordic Iterations for angle 30 degrees. You can observe them on the waveform.

No. of Iterations		cosine decimal o/p		sine decimal o/p
1		.60		.00
2		.60		.60
3		.90		.30
4		.83		.52
5		.89		.42
6		.87		.47
7		.86		.49
8		.86		.49

STEP 3
SYNTHESIS

Double click on the synthesize tab. Ignore any warnings. You may check the RTL Schematic to get and idea of the exact inputs and outputs.

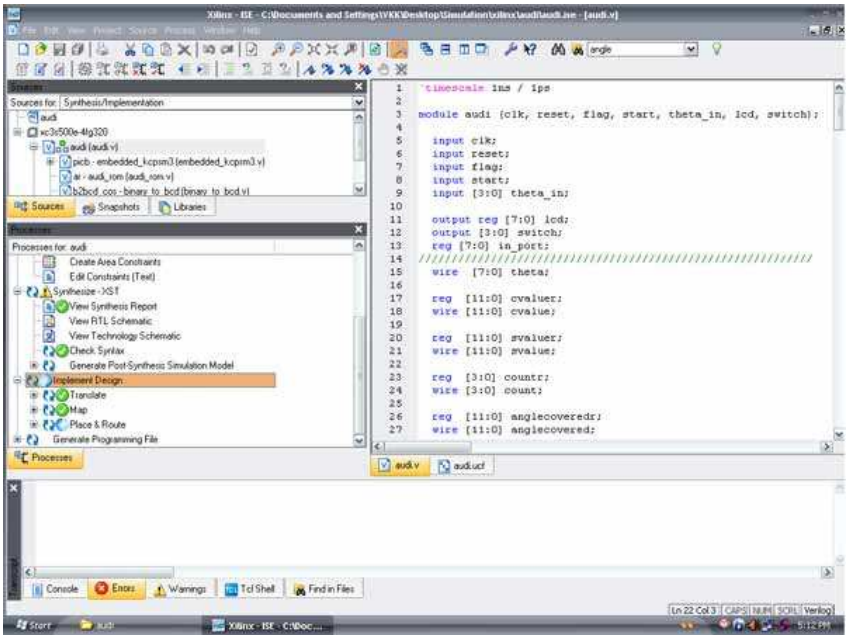


STEP 4
IMPLEMENT

Add the constraint file which contains both the timing constrains and pin assignments. If you have already added it, then you need not do it again.

- audi.ucf (Designed only for Spartan-3E starter kit)

Now, implement the design by clicking the appropriate tab.



STEP 5
CONSTRAINTS&
PIN-ASSIGNMENT

TIME CONSTRAINTS AND CLOCK

The timing and pin assignments are given in the file .ucf(user constraint file). For timing analysis you need to implement it several times with different **offset and hold** timings and check for the optimum one. Currently we are using 50 MHz clock available directly on the kit.

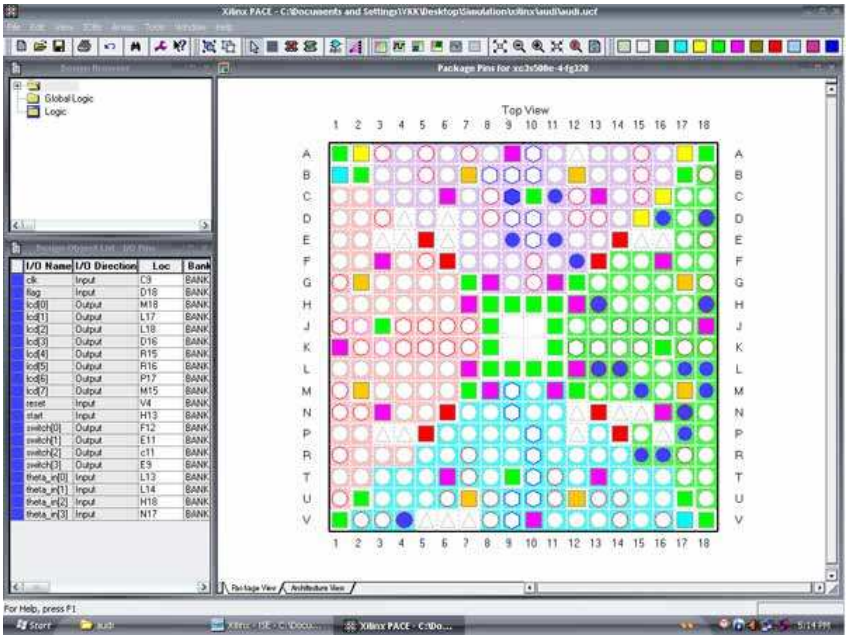
PIN ASSIGNMENT

Input data pins : Four switches for entering the BCD values of each digit of the input angle
Input interruption pins: Three direction buttons
North : RESET
West : FLAG (to indicate first digit is entered)
East : START(start the program)

Instead of the above seemingly cumbersome input method, one can use an alpha-numeric pad for inputs. But you have to spend those extra bucks or at least go to your college Lab to get one. For now, you don't need them.

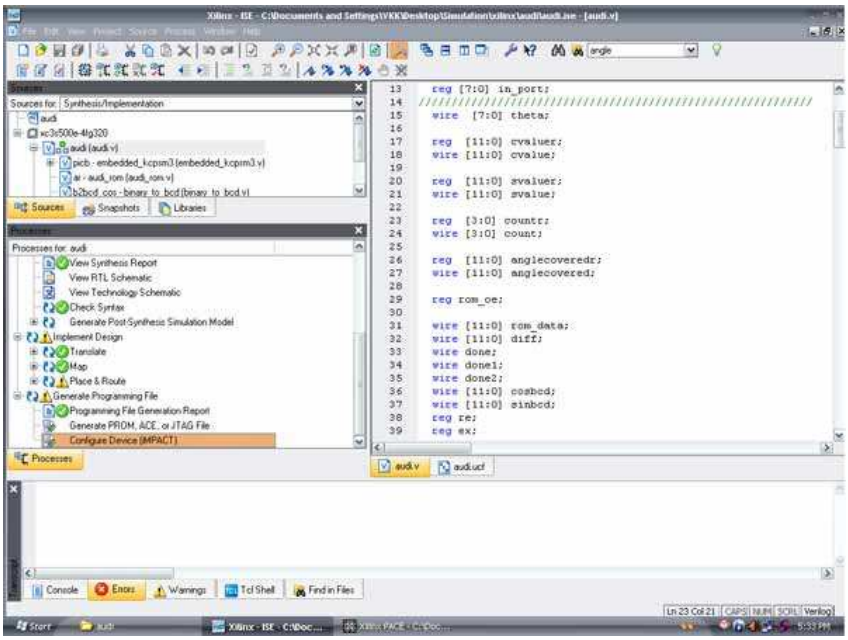
Output: LCD

Check if the pin assignments are correct by clicking the PIN constraints tab.PACE EDITOR opens up. The pin number should be verified from the spartan user manual and they differ with each kit. Please be careful if your kit is a different one. It is better if you can cross check the current file pins from the manual.



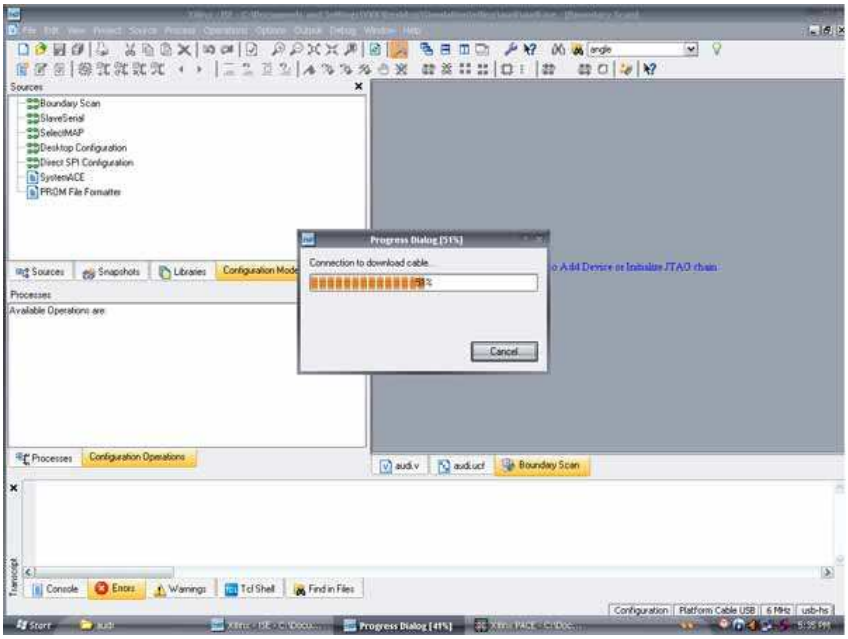
STEP 6
BIT FILE GENERATION

Click on Generate programming file tab. A file <your_program>.bit is generated in the work folder.
This single file is all we need..!!!



STEP 7
DOWNLOAD TO FPGA

Still the kit is powered off. Now concoct it through USB cable . Also connect the power cord and switch it on. You will be prompted to install few driver programs. Follow them and once you are done it shows that "Your hardware is ready to use". Note that it prompts you continuously three times. Do not skip any of them. Aft et it is ready, click on the IMPACT tool tab.As you start the impact tool with the kit powered on, You wi il find the following screen opening up.



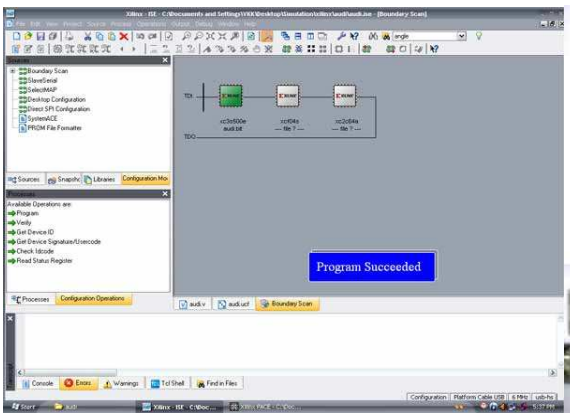
STEP 8
BURN THE FPGA..!!!!!!

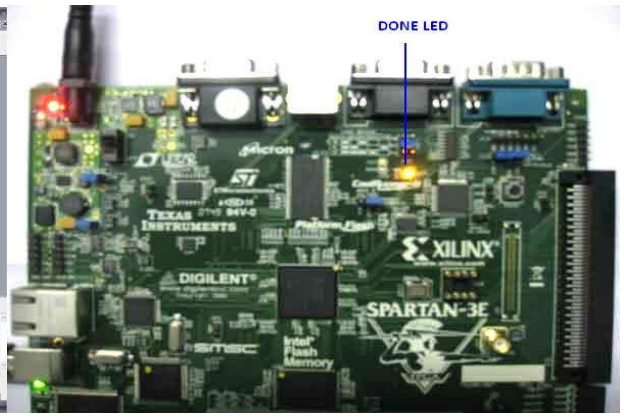
Now xilinx prompts you for file assignment. Select the .bit file. Cancel all the further prompts. They are necessary if you are using any of the memory storage like PROM or FLASH. At present we are directly programming the FPGA and do not need them.

That implies, we have to program the FPGA each time it is powered on. In order to store the bit file in any of the sources available on the kit, you need few more steps, which I have not included.

After assigning bit file, right click on the XILINX chip symbol and click on "program".

If it is successful, the following screen appears and the DONE led(on the board) is on.





STEP 9
SEE IT WORK

As soon as the done led shows up, the fpga starts working. And you will find the following lines on the LCD.

"CORDIC ALGO"
"ENTER ANGLE"

Now press RESET (Press each time you input a new angle)

EXMAPLE-1

Input angle 35 degrees. We are giving it as a BCD code. As we have only four switches we need to enter each digit saparately.Put these switches in On mode as shown. We enter the digit 3 now;its BCD code is 0011.

Now press left direction key to indicate that first digit has been entered

Now give the input as 5 i.e. the ones digit of input angle 35 degrees.

Immediately you will see the the following things popping up on the LCD screen.

EXAMPLE 2

You can try another example. Say 60 deg. Now follow the previous steps. For yo yr convenience I have given them again with respect to 60.

Do not forget to press the RESET button(we have assigned it to the NORTH direction pin)

OUTPUT:

Note that the accuracy is good for angles in the range of (5-85) degrees. For the rest it needs more steps for converging. Currently we are using only 8 iterations.

I hope you have enjoyed it. This is my first attempt to make a tutorial and I welcome your valuable suggestions to improve it. I admit that I am also a beginner to this field and I realize that there are many more concepts involved in working with FPGAs. Please do contact me if you have any thing to discuss.

KAUSHIK VEMULA
Department of Electrical Engineering
IIT Kharagpur
Contact Email: kvemula [at] iitkgp.ac.in
Phone : 91 9932750760

LINK TO HOME PAGE