# DE10 GCD Calculator

# Description

In this project, we will communicate with your GCD design through a virtual board interface instead of through the DE10’s switches allowing you to test different inputs from your computer.

# What you will need:

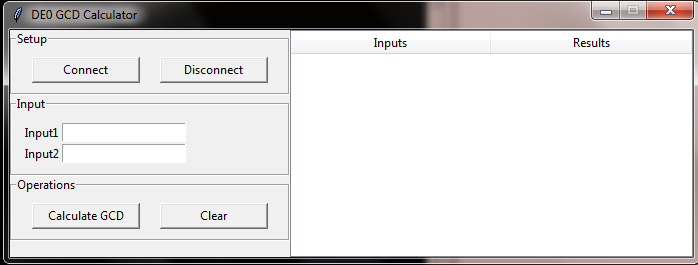
* Python (tested on 2.7.10 and 3.6.2) with the Tkinter package
* Quartus Prime 17.1 Lite Edition
* Terasic DE10 Development Board

# Setting up your design

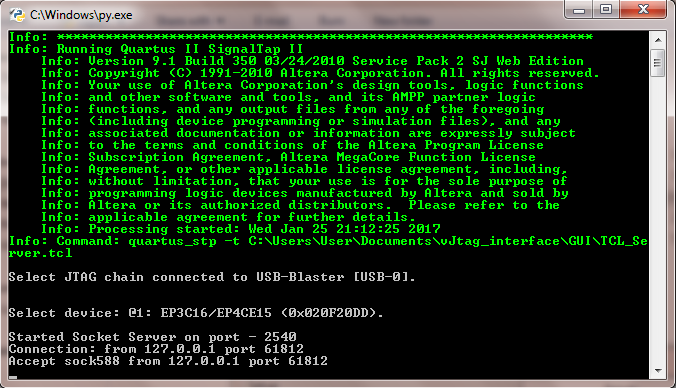
1. Add your lab5 GCD code to the provided vhdl files, but use the provided top\_level.vhd as your top level entity.
2. Setup the pins for the top\_level signals
3. Compile your design
4. Program your board

# Using the GUI

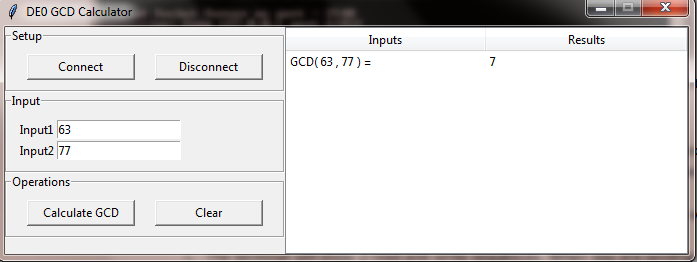
1. Either **(A)** put quartus’s bin folder on your PATH environment variable or **(B)** find the path to your quartus bin folder (e.g. C:\intelFPGA\_lite\17.1\quartus\bin64).
2. Run the main.py file through python. If you did 1B, pass the path as an argument on the command line (e.g. python main.py C:\intelFPGA\_lite\17.1\quartus\bin64)
3. After running main.py, a window should pop up as shown below



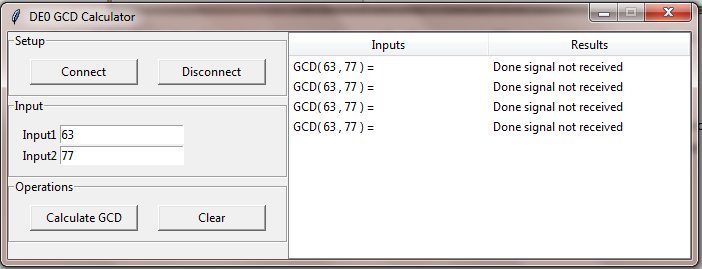
1. Program your DE10 board with your design in quartus and then click on the “Connect” button, which will run the TCL\_Server through the quartus\_stp.exe.   
     
   **Note**: If you need to reprogram the DE10, press “Disconnect” before reprogramming or else the reprogramming will fail.



1. To calculate a new GCD, enter “Input1” and “Input2”, and then click on the “Calculate GCD” button. For now, the inputs are restricted from between 1 and 32767.



**Note:** If the done signal is not implemented correctly, the message “Done signal not received” will display instead of the GCD result.

****

# How it works

The GUI is written in python and runs a TCL, TCP/IP server to communicate with the board. The port that is opened is 2540. Once the connection is made to the board from the computer, the python scripts will send binary data to the port number listed. At this point, the TCL script takes over and sends the data in serial. With the TCL script, there are a few main commands that are used that are supported by Quartus.

|  |  |
| --- | --- |
| Open\_device –hardware\_name $usbblaster\_name –device\_name $test\_device | This command opens the JTAG Device so that it will accept further commands |
| device\_virtual\_ir\_shift –instance\_index 0 –ir\_value 1 –no\_captured\_ir\_value | Setups up to send in data |
| device\_virtual\_dr\_shift –dr\_value $send\_data –instance\_index 0 -length 8 –no\_captured\_dr\_value | Data to be sent |
| device\_virtual\_ir\_shift –instance\_index 0 –ir\_value 0 –no\_captured\_ir\_value | Set IR register back to 0 which is bypass mode |
| device\_virtual\_dr\_shift –dr\_value $send\_data –instance\_index 0 -length 8] | This command sends data into the device and also reads back in a value from the tdo pin. |