Frog Eye User Manual

The following components are required to access and run the project:

- An external monitor/desktop
- Project board
- Laser apparatus, including laser power source
- 9-12V board power source (either battery or function generator)

In addition, the following software tools are required.

- Vivado
- Jupyter notebook
- TeraTerm

1. Connect the ethernet cable and USB power source from your desktop to the FPGA.
2. From the GitLab, download and unzip the frog_eye_spi.zip file of the Vivado project to your preferred destination.
3. Navigate to the project from your chosen filepath "/frog_eye_spi" and open the project in Vivado.
4. Inside the project, go to "File -> Export Block Design", keep the .tcl file named "design_1", and click "Ok". Now, in the main "/frog_eye_spi" folder, you should see an updated "design_1.tcl" file.
5. Now, two more files must be located and copy-pasted in the main project folder.
    1. The first .hw file, labeled "design_1.hw", can be found in the "project_1.hw" folder.
    2. The second .bit file, labeled "design_1_wrapper.bit", can be found in the "project_1.srcs" folder, at the location "sources_1/bd/design_1". This file needs to be renamed "design_1.bit" before being pasted.
6. Once copied, power on the FPGA board, and open the program "Tera Term" to verify that serial communication with the FPGA has been established. If the board has not been connected via TeraTerm before, the user must configure the terminal settings and locate the board's emulated serial port. To do this, select "Serial Port Setup" from the Setup menu, and select the highest numbered port available in the drop list for Ports. Set the speed to "115200" and the data to "8 bit". In the Terminal Setup menu, ensure that the terminal ID is set to "VT1000" and "Local echo" is unselected. Once settings are configured, restart TeraTerm. It will take about 30 seconds-1 minute for the connection to be established. You will know the connection with the board is complete once the four LED's on the FPGA begin to blink, and TeraTerm outputs a wall of colorful text.
7. In an empty web browser, search "Pynq006:9090". This is the web address for the Python API Overlay, which allows users to control data transfer between the board and the computer using Jupyter Notebook. In the event the website asks for a password, the password is always "Xilinx".
8. Once the notebook is open, navigate to "Files -> frog_eye_spi". It may take a moment for the "Files" page to refresh and fill with projects, so be patient.
9. In the "frog_eye_spi" folder, upload the .bit, .tcl, and .hw files from the main project folder using the "upload" button in the top-right corner. This step is necessary to ensure the firmware configuration is up-to-date with the Vivado block diagram.
10. Once uploaded, upload the python code file "frog_eye.ipynb" from the GitLab, and run the first five cells of code sequentially.

    1. The first block of code imports necessary python libraries.
    2. The second block of code creates variables for the AXI peripherals.
    3. The third block of code initializes and debugs GPIO ports.
    4. The fourth block of code performs necessary memory mapping and declares variables for writing out the FPGA control signals.
    5. The fifth block of code defines functionality for reading and writing to/from the board.
11. Power on the circuit board and the laser by flipping the first and last switch on the project board's switch board. The switch board should already be connected to active power sources, whether a battery or function generator. If connected to a function generator, ensure that the generator's output is a 9-12V DC.
12. With the board fully powered, run the final cell of code in the notebook This code begins writing signals from the FPGA and reading in from the circuit's ADC. The user can now shine the powered infrared laser on the pixel array. Once concluded, the process can be stopped by clicking on the four LED buttons on the FPGA.
    1. Below the previously executed block of code, the user should now be able to see a diagram that reconstructs the laser's image and predicts its path.