Instructions to synthesize and implement the hardware:

1. Unzip the file *VERC_seq_ddos.zip* to a directory. This is an archive of the Xilinx ISE project. You will find the project files and design sources inside the unzipped directory.
2. Install Xilinx ISE Design suite. This project is created using Xilinx Design suite version 14.5.
3. Open the Xilinx project file (*VERC_seq.xise*) in Xilinx Project Navigator. The target FPGA of the implemented design is Xilinx Virtex 5 (XC5VLX50T). You may change the target FPGA by double clicking on the device ID on the design pane and selecting the Xilinx FPGA of your choice.
4. Click on the top module (*circuit.vhd*) in the design pane.
5. On the process pane, double click on "Check Syntax". It will run the syntax checker. If “Check Syntax” process completes successfully (green tick), then go to step 6. If it fails, go through the design sources and fix the problem (s) (mostly due to compatibility issues).
6. Once the Syntax Check is completed successfully, synthesize the design (double click on "Synthesize"). Upon successful synthesis, the console window will print the message *Process "Synthesize - XST" completed successfully*. The console will also print the estimated performance parameters (such as maximum frequency). However, this is just an estimate and not the actual performance of the design.
7. Check the warnings generated by the synthesis process in "Design Summary". Warnings related to unconnected signals may be ignored. There may be warnings related to black-box instantiations as well, which are due to instantiation of the Xilinx Coregen IP cores (divider and Cordic cores) in the design. These warnings may be ignored.
8. Now that the design has been synthesized, implement the design on the target FPGA (double click "Implement Design"). This process consists of three sub-processes, namely Translate, Map, and Place & Route. These sub-processes will be completed sequentially. The implementation properties, such as optimization strategies for implementation, can

be selected using "Process Properties" (By right clicking "Implement Design"). On successful completion of the Implementation process, go to step 9.

9. After the design is implemented, you can check the resource requirements and performance of the design on the target FPGA. Open the Place & Route (PAR) report in the "Design Summary". In the PAR report, check the "Best Case achievable" for the "clock net clk_BUFG" constraint. This value is the minimum time period (maximum frequency) of the design. In the "Device Utilization Summary" section of the PAR report, the resource requirements (such as Slices, DSPs and BRAMs) of the design on the target FPGA may be found.
10. After successful implementation, generate the programming file (double click on "Generate Programming File"). Download the generated .bit file onto the target FPGA for testing the design for DDoS detection on hardware.

I/O and Simulation of the design:

1. On top of the Design pane, click on view: Simulation.
2. Click on the top module *circuit_test.vhd*. This file is a wrapper to the top module of the design (*circuit.vhd*) and contains the test-bench for simulation.
3. On the process pane, double click on "Simulate Behavioral Model". It will open the Xilinx ISim simulator. The Simulator will contain waveforms depicting the inputs specified by the test-bench file (*circuit_test.vhd*), as well as the outputs generated by the design.
4. Observe the input values given to the system during the simulation run time (set at 3,000 ns, can be changed using "Simulate Behavioral Model -> Process Properties").
    a) *clk* : Clock input signal
    b) *reset* : Resets the operation of the design to the initial state
    c) *start* : starts execution on input data

d) *threshold[7:0]* : 8-bit input that specifies the threshold value between attack and normal

e) *data_in_r_x1[7:0]*, *data_in_r_x2[7:0]*, *data_in_r_x3[7:0]* : Normal profile vector (from the security manager)

f) *data_in_r_y1[7:0]*, *data_in_r_y2[7:0]*, *data_in_r_y3[7:0]* : Input traffic instance vector (from the pre-processor)

5. Observer the outputs of the design for the given inputs.

a) *is_attack* : 1-bit attack detection output signal (normal/attack)

b) *r_valid* : 1-bit output signal representing whether the detection output (*is_attack* signal) is valid or not

c) *r_out[19:0]*: 20-bit signal in 12.8 fixed point representation. Represents the NaHiD_VERC value for the input (normal profile, traffic) pair.