Leveraging the ESnet SmartNIC Setup for Simulation of "p4_with_extern" Example using Vivado 2023.1

This guide was written on 07/10/2023 using the (https://github.com/esnet/esnet-smartnic-hw) commit 7f78792.
The original ESnet documentation can be found here:
https://github.com/esnet/esnet-smartnic-hw/tree/main/examples/p4_with_extern

To run tests for an example with externs, follow these steps:

1. Open the github esnet-smartnic-hw repository.
2. Review the two changes that have been merged.
3. New 'p4_with_extern' example design:
    a. Locate the c++ code for the user extern in the file p4/sim/user_externs/smartnic_extern.cpp.
    b. Launch p4bm simulations from the p4/sim/ directory to incorporate the user extern functionality in the behavioral simulation.
    c. Place the RTL for the custom extern function in the file extern/rtl/smartnic_extern.sv.
    d. If there are additional files in the design hierarchy of the extern function, place them in the extern/rtl/ directory.
    e. Execute 'make' at the root of the application design directory to build the smartnic design with the extern functionality.
4. New support for adding the vitisnetp4 example design:
    a. Navigate to the root level of the application design directory.
    b. Type 'make example' to create the vitisnetp4 example design in the subdirectory 'example/sdnet_0_ex'.
    c. Open the 'sdnet_0_ex.xpr' project file in Vivado to support a GUI-based simulation workflow.
    d. From the `Flow Navigator` menu, select 'Simulation->Run Simulation->Run Behavioural Simulation'.
    e.
5. This feature is supported for any SmartNIC application design that uses the P4 workflow, including externs.
    a. If an extern is part of the application design, the script infrastructure will include its behavioral model and RTL in the example design project.
    b. The standard simulation flow for a vitisnetp4 example involves running the p4bm (behavioral) simulation first to generate expected output.

    c. Next, run an RTL simulation that compares the output to the expected data, flagging any errors.
    d. Use the waveform viewer for debugging.
6. Review the README updates for more information on these new features.
7. Take note of any file name changes for some of the example p4bm simulation testcases.
8. If you have any questions or issues related to these changes, feel free to reach out.

Note: The P4 program used in the Vivado workflow can be found under example/.src/.
Note: For users trying to run the simulation over SSH, using VNC has proven to be the easiest most straightforward way. Nonetheless, I've included the PyAutoGUI script that I have written to automate the process of emulating the GUI via CLI to use over SSH.