CYENG 351 – SP 23: Embedded Secure Networking
Instructor: Dr. Shayan (Sean) Taheri
Gannon University (GU)

Assignment 6: Chapter 6 - Wireless

Total Points: 100**; and Deadline:** April/30/2023, 11:59 PM.

Note – Cheating and Plagiarism: Cheating and plagiarism are not permitted in any form and they cause certain penalties. The instructor reserves the right to fail culprits.

Deliverable: All of your responses to the questions of assignment should be included in a single compressed file to be uploaded to the Gannon University (GU) – Blackboard Learn environment.

Question 1. Provide short answers (i.e., no more than five lines on average with the font size of 12) for the following items. The grade for each item is **20 points**.

1. Discuss how the **Bluetooth** protocol is secured. Explain the operations of three attacks of **Bluejacking**, **Bluebugging**, and **Blue-snarfing**.
2. Mention the design goal of the full **WPA-Enterprise** authentication suite. Explain usages of the **Temporal Key Integrity** protocol.

Question 2. Complete the laboratory part, titled "**Wireshark Lab: UDP v8.1**". The grade for this question is **30 points**. Provide the screenshots for all of the major steps/processes in your experiments as well as the answers to the laboratory questions.

Question 3. Implement networking modules, conduct experiments to evaluate and test their operations, and analyze their functionalities and results: **Universal asynchronous receiver-transmitter (UART)**, is one of the most used device-to-device communication protocols. When properly configured, UART can work with many different types of serial protocols that involve transmitting and receiving serial data. In serial communication, data is transferred bit by bit using a single line or wire. In two-way communication, two wires are used for successful serial data transfer. Depending on the application and system requirements, serial communications need less circuitry and wires, which reduces the cost of implementation. **Serial peripheral interface (SPI)** is one of the most widely used interfaces between microcontroller and peripheral integrated circuits (ICs) such as sensors, analog-to-digital converters (ADCs), digital-to-analog converters (DACs), shift registers, static random-access memory (SRAM), and so forth. SPI is a synchronous, full duplex main sub-node-based interface. The data from the main or the sub-node is synchronized on the rising or falling clock edge. Both main and sub-node can transmit data at the same time. The SPI interface can be either 3-wire or 4-wire. Complete the following Steps. The grade for this question is **50 points**.

A. Study the articles under the following section of "**Resources for Step A**" to fully understand the functionalities of the UART and the SPI networking modules.
B. Implement the UART and the SPI modules using **Verilog hardware description language (HDL)**. Sample implementations of them in VHDL are provided in the following section of "**Resources for Step B**" for your kind reference, guidance, and understanding of the implementation processes.
C. Perform the procedure from the "**EXPERIMENT #1: Introduction to Xilinx's FPGA Vivado HLx Software**" laboratory assignment for your implementations.

D. Include the following items in your submitting package:
   - Provision of the achieved results from your implementations in the **Steps B and C**.
   - All files of your implementations in the **Steps B and C**.
   - Provide a report that includes: (**1**) your overall understanding, evaluations, analyses, and conclusions from completing the experiments; (**2**) the interesting points and the challenges that you faced in this laboratory; and (**3**) the screenshots for all of the major steps in your experiments.

Resources for Step A – UART:

1. Universal asynchronous receiver-transmitter - Wikipedia
2. Serial Communication - SparkFun Learn
3. Universal Asynchronous Receiver/Transmitter (UART) for KeyStone Devices UG (ti.com)
4. Universal Asynchronous Receiver Transmitter (UART) (infineon.com)

Resources for Step A – SPI:

1. Serial Peripheral Interface - Wikipedia
2. Serial Peripheral Interface (SPI) - SparkFun Learn
3. Serial Peripheral Interface (SPI) for KeyStone Devices User's Guide (Rev. A) (ti.com)
4. Serial Peripheral Interface (SPI) Master (infineon.com)

Resources for Step B – UART:

1. jakubcabal/uart-for-fpga: Simple UART controller for FPGA written in VHDL (github.com)
2. pabennett/uart: A VHDL UART for communicating over a serial link with an FPGA (github.com)
3. akaeba/tinyUART: Lightweight UART core in VHDL (github.com)
4. Domipheus/UART: Simple UART implementation in VHDL (github.com)
5. tvanas/uart-vhdl: An RS232 communication controller implemented in VHDL (github.com)
6. alex-gudilko/UART-CORE-VHDL: UART core for FPGA written in VHDL (github.com)

Resources for Step B – SPI:

1. jakubcabal/spi-fpga: SPI master and SPI slave for FPGA written in VHDL (github.com)
2. nematoli/SPI-FPGA-VHDL: SPI Master and Slave components to be used in all of FPGAs, written in VHDL. (github.com)
3. akaeba/generic_spi_master: Customizable multi chip select supporting Serial Peripheral Interface master. (github.com)
4. lordofhyphens/Xilinx-SPI: Xilinx FPGA-based Serial Periphial Interface Bus implementation in VHDL. (github.com)
5. vinicvaz/VHDL: VHDL-SPI Protocol (github.com)
6. daleonpz/spi_vhdl: SPI-based Communication protocol on VHDL (github.com)
7. KevinAsher/spi-vhdl: SPI-VHDL (github.com)
8. berrekaraman/SPI: SPI Master and Slave for FPGA - VHDL (github.com)
9. MiguelNarv/SPI-Module-VHDL: SPI implementation for FPGA. (github.com)