rEE/CprE/SE 4910 WEEKLY REPORT 3

02/20/26-02/27/26

Group number: SD26-10

Project title: Open-sourced Radio Microcontroller for Fabrication

Client &/Advisor: Henry Duwe duwe@iastate.edu

Client/Company/Organization: ISU; ECPE

Team Members:

- ***Zane Salti***
- ***Jackson Doyle***
- ***Daniel Pytel***
- ***Tyler Bibus***
- ***Aiden Han-Lindemyer***
- ***Parker Pederson***
- ***Akash Gojuru***
- ***Seth Klaassen***

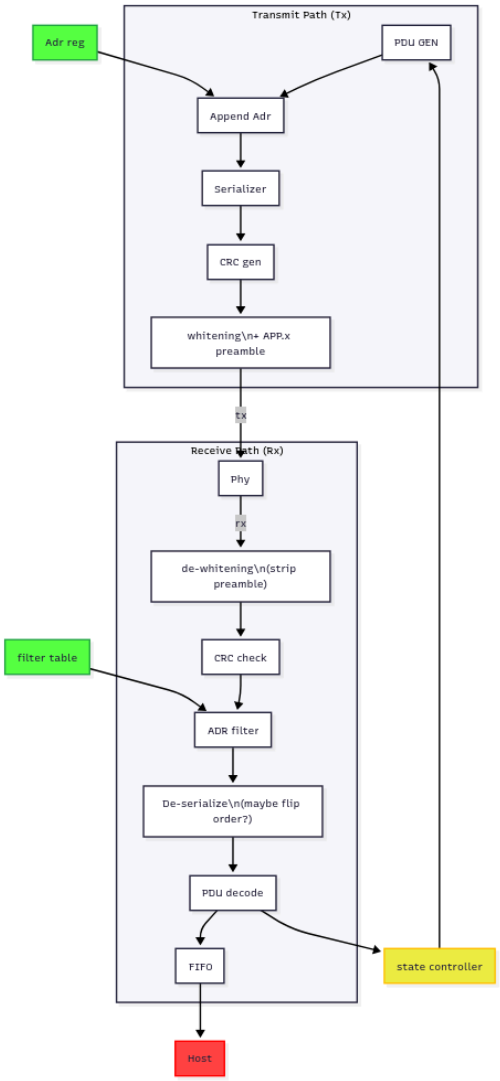
Weekly Summary:

Link Layer & HCI:

Architecture:

The overall Architecture of the Link Layer is still under heavy development; the Link Layer team has opted to work on simulating the overall structure and working backwards given the complexity of the structure. There is a living document in development that will provide overall project status, with a visual representation of the modules, and submodules.

The diagram below is effectively the transmission path from into the Physical Layer (Phy tx) and out of the Physical Layer (Phy rx). This diagram is ongoing but provides the general process of how data is received, processed, and transmitted back. Data flows in a loop with the host receiving and sending data through a first in first out register or shared memory. The host and state controller determines exactly what is sent, but all packets of data are sent through the same process.



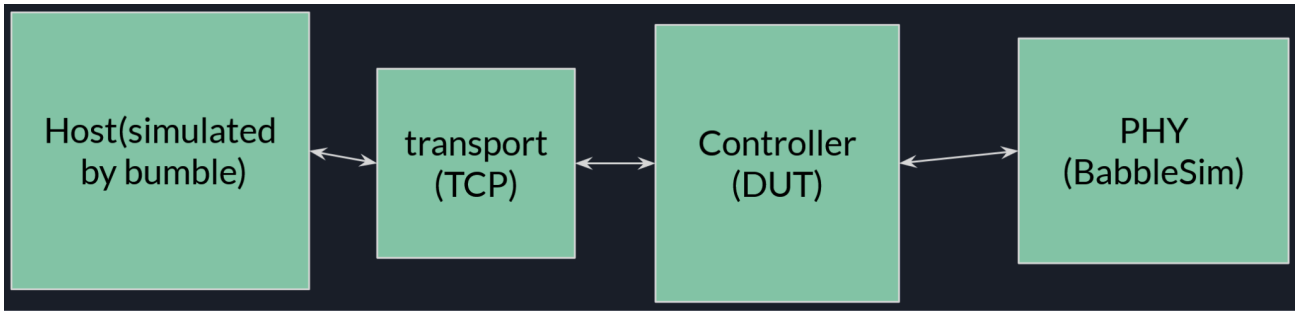
Bumble:

Bumble is a Bluetooth simulator maintained by Google under an open source license that can be used to emulate both normal and LE devices. The simulator can emulate multiple copies of hosts and controllers and pass connections between both simulated and non-simulated devices. Of particular interest is the ability to emulate a bluetooth host that can interact with generic controllers over a variety of transport methods. This week the team was able to set up a fake bluetooth host that can send HCI commands and receive HCI events. This will allow us to create a golden sample with which to test our HDL once it is created.

BabbleSim: https://babblesim.github.io/

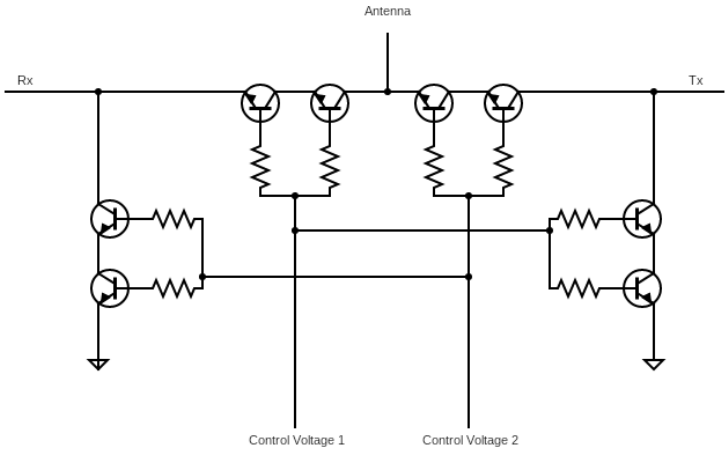
BabbleSim is a physical layer simulator for Bluetooth. This allows us to build a software based Host on top of it to verify correctness. BabbleSim creates sub-processes on linux, simulating and replicating radio communication down to the timing. This will integrate into our other simulation with Bumble, creating the ideal environment to focus on the Lower aspects of the Link Layer and HCI while also providing a ground truth to compare to, especially as there are working full examples we can communicate with to ensure compliance with Bluetooth Specification. Babble will be an important part of our simulation and procurement of a working model. More exploration into Babble from running examples is ongoing and should produce results for next week.

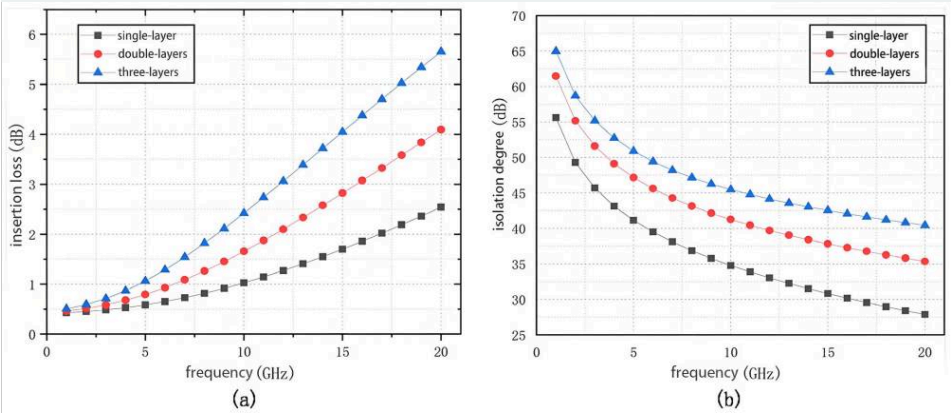
Simulation Plan:



Physical Layer

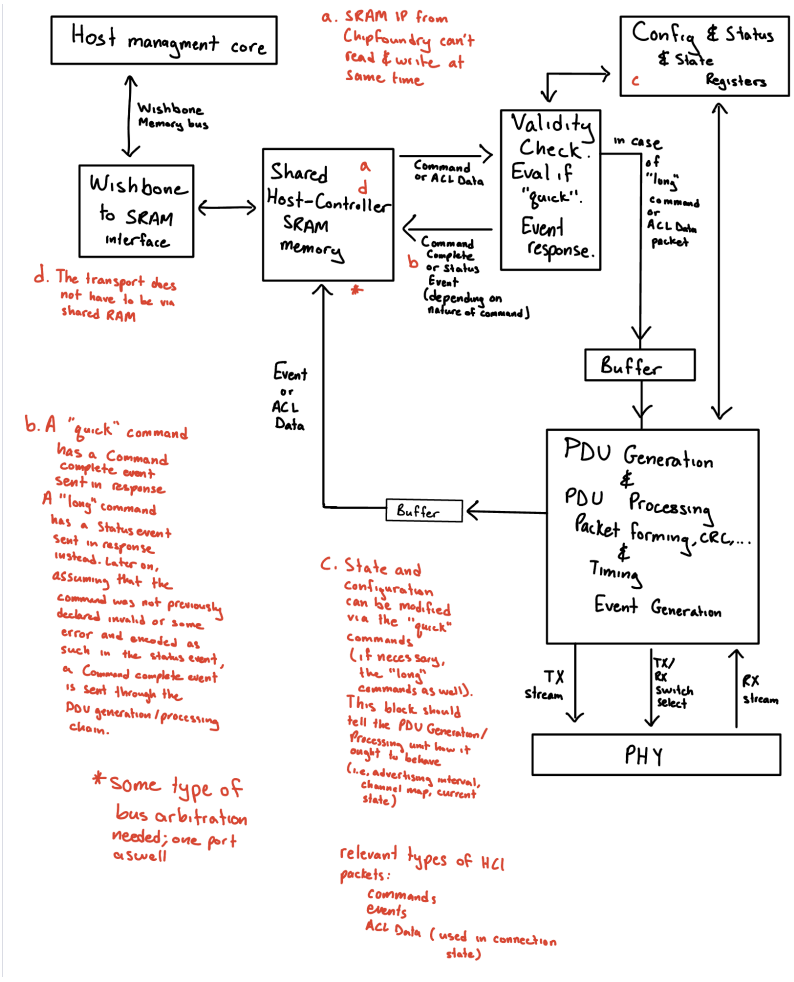
RF switch design has had some progress, some narrowing down of possible architectures and design parameters was completed during this week. The results below apply for nominal transistor sizing (100 x 2 micro meters for shunt and 75 x 2 micro meters for series transistors). Adding layers improves isolation with minimal additional insertion loss.





Past week accomplishments

- Zane Salti: Worked on the higher level architecture of the Link Layer with the HCI. We discussed whether the physical HCI transport for the host and controller should be memory mapped registers or a shared RAM (with SRAM). We decided to go with the shared RAM for now.



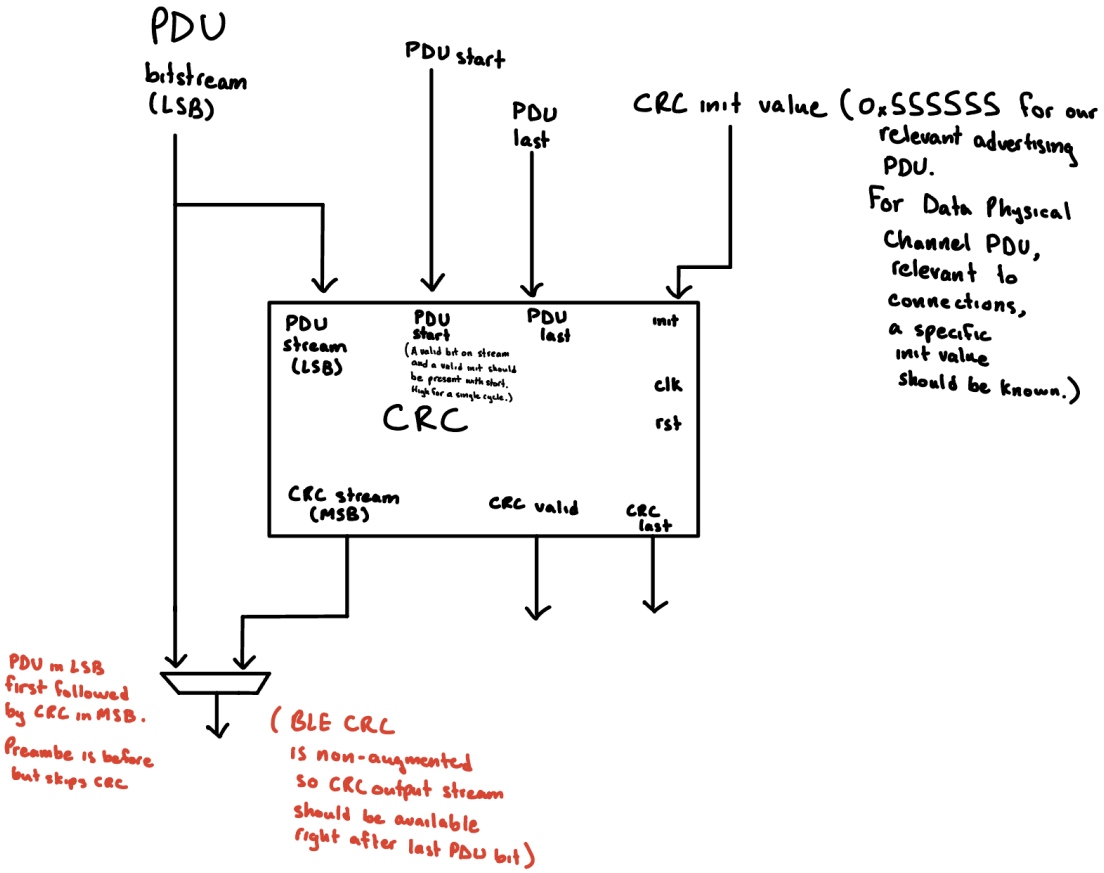
This architecture gives a way for “short” commands to be processed and reduce blocking while allowing the “longer” commands or ACL Data a path through. Considering that the SRAM IP has a single read/write port, some bus arbitration would be needed.

Looked at possible hardware transceivers for testing with the Link Layer. Identified three:

1. SX1280
2. nRF24 Series
3. CC1200

Some of these may have their own CRC and packet forming engines that would need to be disabled for our own comprehensive testing of the Link Layer. We would also have to see if the interface to these transceivers are high speed and reactive enough to mock a PHY and keep up with our timing constraints (it could be that you load these transceivers up and then they go on their own which wouldn’t be the desired behavior for us for mocking the PHY).

Got started with the HDL (using SystemVerilog) of the CRC module for the Link Layer. Had it in testing (some online CRC generators conflicted on what the “correct” CRC value should be). The CRC is implemented as a linear feedback shift register with a polynomial of x^24 + x^10 + x^9 + x^6 + x^4 + x^3 + x + 1. The CRC is calculated on the PDU from least significant bit first (which is the same order it should be transmitted over air). The module outputs a stream of the calculated CRC from the most significant bit first (which is the order the CRC should be transmitted over air). The CRC module should be used such that the PDU is streamed and simultaneously is fed into the CRC module. Once the PDU finishes, the CRC module should start streaming out.



- Jackson Doyle: Began simulations of previous groups work (PLL),

- Daniel Pytel: Went through previous group's notes and documents to figure out what was salvageable. Gained more familiarity with the toolflow.
- Tyler Bibus: Drafted sketches into professional diagrams for the Link Layer. Started exploring BabbleSim to simulate the Phy layer for simulation to achieve a ground truth.
- Parker Pederson: Setup local testing environment for bumble in order to simulate the host side of the system in python. Designed testing plan to allow for the creation of a golden sample that can be compared to results of the generated hardware in order to facilitate verification. Began designing linked layer architecture from the phy layer up.
- Aiden Han-Lindemyer: Set up bumble locally, will be setting up the environment in the VM requested from ETG.
- Akash Gojuru: I explored the main PDUs types and their functionalities and created a sheet their types values and purposes.
- Seth Klaassen: Improved our design for our RF switch, giving a higher degree of isolation, and is able to stably run at the required 2.8Ghz with minimal insertion loss

Pending issues

- Zane Salti: CRC module needs to be verified. With shared ram as transport between the host and controller, a way of synchronization needs to be more worked out as well.
- Jackson Doyle: Limit testing the PLL. To do this various tests need be employed on the design to find performance metrics that can assist in the process of completing the PLL to be usable in our design.
- Daniel Pytel: Still wrestling with various toolflow issues.
- Tyler Bibus: Working/waiting for a VM to set up a proper environment for BabbleSim.
- Aiden Han-Lindemyer: Working on testing plan for python sym
- Parker Pederson: Need to set up a VM in order facilitate a common testing environment for both babble and bumble sim. Working in the VDI is suboptimal, bumble sim does not run due to the VDI using an outdated version of python. Only transport setup that appeared functional is the TCP layer which is not optimal for simulation of our design. A filepointer approach would more closely mimic the approach of a shared memory region.
- Akash Gojuru: Working on implementing PDU decoding in the tool flow.
- Seth Klaassen: Needs to continue progress in modeling switch in toolchain for testing the optimizing transistor size for our frequency of 2.4Ghz

Time Tracking

NAME	Hours this week	HOURS cumulative
Zane Salti	9	21
Jackson Doyle	3	10
Daniel Pytel	4	9
Tyler Bibus	5	12

Aiden Han-Lindemyer	3	11
Parker Pederson	5	13
Akash Gojuru	4	11
Seth Klaassen	5	11

Plans for the upcoming week

- Zane Salti: Verify CRC module correctness. Incorporate a whitening and CRC module into a serializer module that generates a bitstream containing the proper preamble, address, PDU, and CRC code.
- Jackson Doyle: Implement components and testing using the toolchain software. Gain comfort in xschem. Seek assistance of 4920 ASIC group to speed up my process in gaining comfort using xschem/magic. Concept verify a component of the PLL for the required specs for bluetooth and adjust the design accordingly and see performance.
- Daniel Pytel: Run simulations of previous design work, arrange meeting with other faculty to aid scoping physical layer.
- Tyler Bibus: Get BabbleSim working, ensure that the shared VM environment has the dependencies for BabbleSim. Draft additional diagrams into professional graphs. Get a living project status document running.
- Aiden Han-Lindemyer: Set up VM for babble w/ needed dependencies.
- Parker Pederson: Set up combined simulation environment inside of a custom virtual machine. Switch the transport style over to a file pointer and finish basic HCI responses in python. Finally, create a living diagram to monitor the state of each module in the system.
- Akash Gojuru: Continue learning Channel Selection Algorithm 1 and start using it to hop between channels so connections are more stable and have less interference.
- Seth Klaassen: create a model of our RF switch in magic for the sky130 process

Summary of weekly advisor meeting

The team sought feedback from Dr. Duwe regarding our problem statement. He was able to provide advice to narrow it in certain areas to make it more concise. Next, we discussed the physical layer and the design of the RF switch. Dr. Duwe advised us to seek help from an outside professor, particularly Dr. Neihart for advice about the design of the analogue components and the ability of the skywater process to meet our requirements. Dr. Duwe also suggested that the team submit a previously design PLL for the May tapeout.

Next, we discussed the top-level architecture of the digital side of the device. He was concerned that the shared memory buffer strategy of communication between the host and controller may not match our goals. The team shared the simulation plan with Dr. Duwe, who was largely supportive of the idea. He was concerned about using a TCP transport for the simulation and advised changing it to better align with a shared memory buffer.

Finally, we were advised to create a living graphical document of the design architecture for both project tracking and presentation.