## Reflection Report

LIUKEE:

We chose this project because I²C is a very common communication protocol in embedded systems and chip design, and the overall size of the project was something we could realistically finish within the deadline. The PCA9632 LED driver also made sense for us because it mixes digital logic (I²C interface) with PWM-based LED control, so it gave us a good chance to apply the SystemVerilog concepts we learned in class.

Riley was the one who originally suggested this design, and he and Mohammad worked together to decide which parts of the PCA9632 we should actually implement. They helped narrow the scope so we focused only on the important digital functions like register control, brightness settings, and LED modes. As a team, we split the modules, worked on them in parallel, and met several times to share progress and fix issues.

For my part, the main challenges were understanding the I²C timing and building a clean state machine for bit-level shifting, START/STOP detection, and communication with the I²C controller. Integrating my interface with Mohammad's controller also brought up timing and handshake problems, which required debugging in QuestaSim. To deal with those issues, I created a clear interface document so everyone understood how the signals should interact, and I spent a lot of time checking waveforms and adjusting the logic until the modules worked together correctly.

Working with the group helped me learn from others, too. Mohammad helped me understand how the controller handled address decoding and ACK behavior, and Riley and Sal explained the PWM timing and how the LED controller processed the registers. I also helped the team by preparing the communication interface, sharing debugging tips, and giving feedback when someone needed help verifying their part.

This project strengthened many concepts we learned in class, especially always_ff/always_comb usage, interface design, FSM structure, and writing effective testbenches. I also learned a lot of new things, like the details of I²C behavior (START, STOP, bit sampling) and how PWM brightness control is built with counters.

Overall, we met the expectations we set at the beginning. Each team member finished their module, we were able to integrate the system, and after debugging we completed all the required tests on schedule.

RILEY:
I advocated for this project because I thought it would be a good challenge that was still within an achievable scope. I also wanted to pick something that incorporated a decent width of design aspects (e.g. I2C, PWM, & clocking). The reason for picking an existing design to base our project on was to have a strong starting point and lots of options for features we could implement. It also gave us insight into what real product design might look like and what considerations come with it.

Early on, the challenge we struggled with was effective communication. We took longer than we should have to get moving on the project due to being slow to communicate. Even once we got moving there were a few instances of overlapping work and misunderstandings with our individual responsibilities. We overcame this challenge by meeting more frequently during the last few weeks, sometimes even twice a day, and being diligent about sending and responding to emails to stay on the same page. On the technical side, I faced a challenging bug within my Group PWM module. There was an issue happening when I ran the testbench where the dimming mode worked just fine, but the blinking mode did not. I was able to track down the source of the bug by drilling into the QuestaSim module tree and finding where the signal was failing to be driven correctly. It ended up being just a typo in one of the ports that was causing an 'x' value to propagate through the system.

Throughout the course of the project I learned to lean on and trust my teammates more and more. We are all capable engineers and it was a good feeling knowing that we were all committed to the success of our project. I helped out my team by reliably designing my sections of the project, utilizing rigorous verification at every hierarchical level.

We used many concepts from class such as packages, interfaces, structs, arrays, enumerations, and timing control. It was beneficial to use these concepts in a larger design because it helped me better understand their value. Specifically the timing control using the 'timeunit' and 'timeprecision' keywords was very useful since our design is highly timing focused. For example, the 400 KHz oscillator requires a nanosecond time scale but the final testbench operates on a 'human' time scale (seconds) due to the long group blinking periods. Overall, the most important new skill I learned was how to organize and manage a large HDL project. This was the biggest one I've been a part of so it was a great opportunity for learning. On a more technical level, I learned this one-shifting trick to calculate a power of two:
[(1 << ADDR_BITS)-1:0].

This project met our expectations for schedule since we were able to complete it within the term. I would say we exceeded our expectations for scope. We originally thought of removing the group PWM feature to make the scope more manageable, but we ended up having enough bandwidth to complete that feature. My personal expectation for this project was to gain a greater understanding of SystemVerilog and HDL projects in general, and that was definitely met.

SAL:
We selected this project because it provided a practical way to apply the SystemVerilog concepts we learned in class while still staying within a manageable scope. Designing an I²C-based LED driver allowed us to work with common digital design elements such as timing logic, PWM generation, register control, and interface communication. The project also aligned well with real-world embedded systems, which made it both interesting and educational.

During the development process, I faced several technical challenges, mainly related to implementing the timing and LED control logic. Creating the individual PWM modules required careful use of counters and clock-division techniques to produce stable brightness levels. Designing the signal-mux logic that selects between different LED modes also required attention to detail to make sure transitions happened smoothly without causing unexpected behavior in simulation. To address these challenges, I relied on waveform analysis, step-by-step testing, and reviewing hardware documentation to understand the required functionality more clearly.

Working in a team environment helped me learn new approaches and ideas from others. I gained a better understanding of how the I²C interface and controller work together, and I also learned about the implementation of group PWM and the overall design's timing structure. At the same time, I contributed by completing my assigned modules, sharing my simulation results, and helping ensure the LED behavior matched the register settings produced by the other components.

This project strengthened many concepts from class, including module design, synchronous logic, reset handling, and simulation-based debugging. I also learned new techniques such as PWM implementation, signal multiplexing, and the use of realistic timing controls in SystemVerilog. These skills helped me improve my confidence in writing hardware code and understanding how different modules interact in a larger system.

Overall, the project met our expectations for both schedule and scope. We were able to complete all main features, integrate our modules successfully, and verify the system through simulation. The experience provided a solid foundation in digital design workflow and showed how classroom concepts apply to a complete hardware project.

MOHAMMAD:

I supported choosing the PCA9632 LED driver project because it offered a perfect balance between control logic and hardware interface design. Specifically, I was interested in the challenge of implementing the I2C Controller, as it required bridging the gap between serial communication protocols and parallel register management. It felt like a realistic representation of the work required in industry-standard ASIC design, rather than just a theoretical exercise.

The most significant challenge I faced was during the verification phase involving false stop conditions during readback operations. We encountered a scenario where read transactions for even-numbered addresses would fail intermittently. Initially, this looked like a logic error in the FSM, but deep waveform analysis revealed it was a race condition in the testbench where the SDA line was changing state while SCL was still high. Another major hurdle was simulation overhead, particularly when simulating large parts of the system and logging data for debugging purposes; our initial full-system tests generated massive waveform files (>6GB) due to the high-speed clock toggling required for blink tests, which made debugging painfully slow.

I overcame the readback challenge by implementing a "bus parking" strategy in the testbench, explicitly forcing SCL low before allowing data transitions. When I dug in deeper, I realized the root cause was a simulation time precision mismatch, where the simulator was rounding our fractional delays to zero. I optimized our verification strategy to selectively dump waveforms only during critical read/write phases.

It was a great experience working with everybody. They were dedicated, competent, though at times slowly communicative, and are effective engineers. I appreciated the view this project provided into our differing approaches and how we all saw the problem slightly differently, and were able to coalesce that into a project that exceeded our scope.

This project significantly strengthened my grasp of FSMs. Designing the i2c controller required me to move beyond simple state transitions and consider how FSMs interact with asynchronous external signals. It further required me to correctly structure my testbench and be very thorough in my approach.

The most impactful new thing I learned was the nuance of complex simulation with these tools. I feel like people have this general idea that you just have to write some code, paste it in the window, compile and move on. However, like we learned, such an approach will cause many headaches, because you have to talk to the tool as well and effectively communicate what you want. The time precision issue was a hard lesson for this, but I feel that it was effective.

Overall, the project met and exceeded my expectations. While the scope of the I2C protocol was complex, we successfully verified the full register map and PWM functionality. We managed the schedule effectively and each learned more from our efforts.