5 Testing

Testing is an ***extremely*** *important component of most projects, whether it involves a circuit, a process, power system, or software.*

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

5.1 Unit Testing

What units are being tested? How? Tools?

- Software Compression Algorithms
  - We will verify that the output after compressing and decompressing data matches the input. This will be done with a Python program to compute the output, and a diff tool if needed to see how the output and input differ. We should also verify that the algorithm is reducing the amount of storage needed to store data.
- Compression IP
  - The initial software testing will give us a way to validate that our intermediate values in hardware are correct. The compression IP will be tested using Vivado tools, HDL testbenches, and eventually tested physically during system level testing.
- Decompression IP
  - The decompression IP will operate very similarly to the Compression IP. It will also have to pass tests from Vivado and HDL testbenches to verify that it is converting compressed input back into the original output.

5.2 Interface Testing

What are the interfaces in your design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?

- Hardware Interface(Zybo board): Using Vivado Synthesis, and Implementation
- Software Interface: Binary/Opcode files (pre-compiled files) (C/C++)

5.3 Integration Testing

What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?

The first step in our project is to create an HDMI pass-through demonstration. This will take an HDMI source, send it through the FPGA and output it to an HDMI display. To do this, we will start by seeing if we can get data from HDMI RX on the board in the FPGA. We will verify this using Vivado. Via unit testing we should be able to confirm that Compression and Decompression IPs work correctly so the last step is to make sure we can display the output of the Decompression IP on a display via HDMI. We will test this by verifying an image is displayed to the HDMI display.

5.4 System Testing

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

To test the entire system we plan on running a video stream through an HDMI cable that is plugged into the Zybo board. The video would go through an FPGA pipeline where it is compressed and stored in memory, afterwards there will be a decompressor in the pipeline that will take the compressed data and decompress it. The video will then go out through an HDMI cable into a monitor. Tests for this would include the Compression/Decompression IP unit tests, HDMI integration test, and interface tests for hardware and software.

5.5 Regression Testing

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure they do not break? Is it driven by requirements? Tools?

A tool that we will be using to ensure that new additions don't negatively impact old ones is the use of git version control, and code reviews. Using these features will ensure that we never break our main source of code (main branch), and that new code will have to be reviewed by at least two people in order to get merged. Additionally, before any new changes are merged to the main branch, it will have to hit certain standards through the pipeline such as code coverage in order to be merged. Assuming that all of the new, and old tests are passing, should be enough to validate that the code is safe to merge into the main branch.

5.6 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

In order to check whether the design requirements are met, and have been completed to an acceptable level by the client's standards, is to eventually get the software to run in a live demo, where an overlay will be brought over the video stream, to show different values such as data loss, and compression rate. If we run out of time during the development period, and don't have ample time to develop an overlay, we could still create software that outputs these values to a log file, where we can verify that the software is working as intended.

5.7 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.