Abdallah Ayoub 1190994 Sondos Shahin 1200166

Verification Plan for Dictionary-Based Compression/Decompression Chip

1. Description of Verification Levels:

The design will be verified under the following verification levels:

- **Unit Level Verification:** This level verifies the functionality of each unit in the system individually. This includes verifying the compression function, decompression function and the memory.
- **Chip Level Verification:** This level verifies the correctness of the chip's functionality for both compression and decompression operations. It ensures the packaged units behaves as specified in the documentation.
- **System Level Verification:** This level focuses on verifying the interconnections between the chip components, and their integration on the chip. It also ensures the whole system design is correct.

2. Functions to be Verified:

- Compression functionality:
  - Compression of data found in the dictionary
  - Compression of new data (adding to dictionary)
  - Handling of full dictionary scenario
- Decompression functionality:
  - Decompression of valid compressed data
  - Error handling for invalid compressed data (index out of range)
- Response signal generation
  - Valid output (compressed/decompressed data) with valid response signal
  - Error condition with appropriate response signal
- Reset functionality
  - The dictionary and the index register are cleared

3. Resource Requirements:

- Hardware: Sufficient disk space for simulation logs and results, in addition to memory allocation for dictionary storage during simulation (memory [256][80]: the dictionary has 256 locations, each location is 80 bit)
- Software: System Verilog simulator with testbench generation capabilities
- Verification libraries: libraries for random data generation, coverage analysis, etc. are needed.

4. Required Tools:

- System Verilog simulator (e.g., Icarus Verilog)
- Text editor or IDE for writing testbenches

5. Schedule:

- **Week 1:** Develop basic testbench framework and functional test cases for compression and decompression.
- **Week 2:** Implement additional test cases for corner cases and error handling.
- **Week 3:** Run simulations, analyze results, and debug any issues.
- **Week 4:** Refine test cases and address coverage gaps.
- **Week 5:** Regression testing and verification report generation.

6. Specific Tests and Methods:

- **Functional Tests:** These tests verify the chip's behavior matches the specifications. We will use a directed testbench approach with specific data patterns to exercise various functionalities and corner cases.
- **Randomization:** Test cases will incorporate random data generation for compression to increase test coverage.

7. Coverage Requirements (Coverage Plan):

- We will target achieving high code coverage for the RTL code using a combination of statement coverage and functional coverage metrics.
- Statement coverage ensures all HDL statements are executed at least once during simulations.
- Functional coverage ensures critical functionalities like handling dictionary full condition and invalid decompression index are adequately tested.

8. Completion Criteria:

- All functional test cases pass successfully with no errors.
- Code coverage metrics meet the set targets.
- No critical bugs or functionality issues are identified.

9. Test Scenarios (Test Plan):

Test Case ID	Description	Cmd	Data (data_in"1", compressed_in"2")	Expected Output (compressed_out, decompressed_out, response)
#01	Compress data found in dictionary	01	"1" Known data in dictionary	Index of the data, N/A, 01 (valid compressed_out)
#02	Compress new data	01	"1" New data not in dictionary	Index of the new data (added to dictionary), N/A, 01 (valid compressed_out)
#03	Compress with full dictionary	01	"1" Data exceeding dictionary size	N/A, N/A, 11 (Error)
#04	Decompress valid data	10	Index within dictionary range	N/A, Original data, 10 (valid decompressed_out)
#05	Decompress invalid data	10	Index exceeding dictionary range	N/A, N/A, 11 (Error)
#06	Invalid command	11	-	N/A, N/A, 11 (Error)

#07	No operation	00	-	N/A, N/A, 00
#08	Reset	-	-	- “dictionary and index register are cleared”

10. Risks and Dependencies:

- **Schedule Risks:** Delays in testbench development or simulation execution could impact the project timeline.
- **Tool Limitations:** The chosen simulator or verification libraries might have limitations that require adapting the verification strategy and test coverage.
- **Dependency on RTL Code:** Verification progress depends on the availability and stability of the RTL code.