Weekly Report

Project: **AHB TO WISHBONE BRIDGE**
Contributor: Bhavana Shreya Padala
Date: Jan 15th, 2026.

TASK 2 - Enhance RTL for Robustness & Scalability

Environment: Ubuntu Linux, Icarus Verilog, GTKWave

1. Overview of Work Completed

During this week, work was focused on enhancing RTL for Robustness and Scalability of the AHB-to-Wishbone bridge. The objective was to strengthen the RTL design by improving read/write path handling, adding defensive logic, and preparing the design for future expansion and additional test scenarios.

2. Technical Tasks Performed

AHB Transfer Handling

- Implemented protocol-correct AHB transfer detection using:
  - `hsel`
  - `hready_in`
  - `htrans[1]` (NONSEQ/SEQ filtering)
- Ensured IDLE and BUSY transfers are safely ignored.

Read/Write Path Separation

- Clearly separated **read and write intent signals** (ahb_read, ahb_write).
- Guaranteed mutual exclusivity to avoid ambiguous behavior.
- Captured address and write data into internal registers for timing decoupling.

Defensive and Robust RTL Design

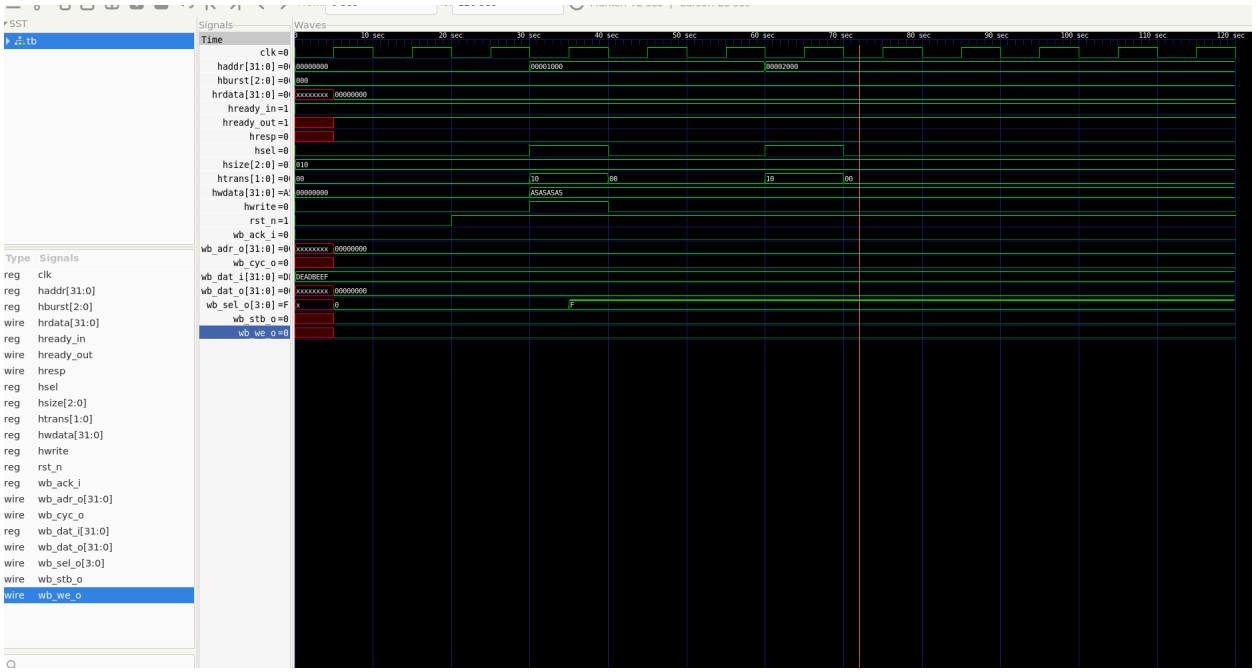
- Added **safe default assignments** every clock cycle to prevent:
  - Latch inference
  - Stale signal propagation
- Implemented **reset-safe initialization** for all outputs and internal registers.
- Added defensive filtering for unsupported `hsize` values.

Scalability Preparation

- Structured RTL to allow:
    - Easy integration of a Wishbone FSM in future tasks
    - Support for additional transfer sizes and error handling
- Maintained synthesizable, lint-clean coding style.

3. Verification Activities

- Developed a **self-contained testbench** to stimulate:
    - AHB write transfer
    - AHB read transfer
- Simulated using `iverilog` and `vvp` on Ubuntu Linux.
- Verified functionality using **GTKWave waveform inspection**.
- Confirmed:
    - Correct assertion of internal valid/read/write signals
    - No unintended Wishbone activity (as expected for this task)
    - Clean simulation completion with `$finish`.



4. Key Observations

- Task 2 does not generate console output by design; correctness is verified through waveform analysis.
- Internal signal behavior matched AHB-Lite protocol expectations.
- RTL is stable, readable, and ready for extension in the next phase.

5. Challenges and Resolutions

- **Challenge:** No visible output in terminal during simulation

  Resolution: Confirmed that Task 2 validation relies on waveform analysis rather than `$display` outputs.

- **Challenge:** Ensuring robustness against invalid transfer sizes

  Resolution: Added defensive logic to disable read/write intent for unsupported `hsize` values.

6. Plan for Next Week

The following activities are planned for the next phase of the project:

- Implement the **Wishbone transaction FSM**, including:
  - Assertion of `wb_cyc_o` and `wb_stb_o`
  - Handling of `wb_ack_i`
- Add **AHB wait-state control** using `hready_out`.
- Complete the **read data return path** by driving `hrdata` from Wishbone read data.
- Extend the testbench to cover:
  - Wait-state scenarios
  - Back-to-back transfers
  - Wishbone acknowledgment timing
- Perform additional simulations and waveform-based verification.