UVM Project

The UVM project will include creating 2 full UVM environments for the Dual-port RAM and the SPI slave wrapper. Check the SV project to get the full details SPI and RAM specifications and verification requirements. **Once the SV project is marked, you must check my comments and implement them in this project.**

Part #1:

Full RAM environment:

- Create a full environment for the RAM design.
- Copy the constraints done in the sequence item class.
- Copy the covergroups, coverpoints done in the coverage collector class.
- Add the assertions file and bind it in the top module.
- Split the RAM main sequence into multiple sequences based on your test plan, for example:
    - write_only_sequence
    - read_only_sequence
    - write_read_sequence

Part #2:

Full SPI Wrapper environment:

- Create a full environment for the SPI wrapper design.
- Copy the constraints done in the sequence item class.
- Copy the covergroups, coverpoints done in the coverage collector class.
- Add the assertions file and bind it in the top module.

Part #3:

Add the RAM environment to the SPI Wrapper UVM test:

- The steps on how to do this part are explained in details in assignment 6.

<u>Requirements needed to test each design:</u>

- Test plan, adding a UVM testbench diagram showing the UVM structure.
- Code Coverage report
- Functional Coverage report
- Sequential Domain Coverage report
- Bug report
- QuestaSim snippets

Submission file:

.rar file containing the following:

- PDF file having the requirements
- Testbench and design files
- 3 Do files to run each part of the above

Also, provide a table showing the assertions used as follows

Feature	Assertion
Whenever the FIFO is full, wr_ack is always = 0	@(posedge clk) (full |-> !wr_ack)