Python
Project

Note: ***Follow the instructions to reach the best clean code.***

Project Description:

The aim of this project is to create a program that generates template designs and testbench Verilog modules. The program will handle the following tasks:

-Module Design: Generate a file with a name matching the module and write the template module design code into it.

-Testbench Design: Generate a file with a name matching the module and append "Testbench" to the filename. Write the testbench module code into this file.

-Testbench Instantiation and Signal Connection: Inside the testbench module, instantiate the module designed in step 1 and establish connections for all the necessary signals for testing.

-Min Cases For testbensh :2 Cases or signal user Enter by same width

Additional Notes:

- Files: Design module and testbensh module (separated Files)
- Design: The module should be named such as "Adder".
- Testbench: The testbench module should be named by concatenating "_TB" to the design module name. This name generation will be automated after the user enters the design module name.
- Signal: For the design module, signals should be named "Datain", while for the testbench module, they should be named "Datain_TB".
- User Input: The program should prompt the user to specify whether the design is synchronous or asynchronous.
- Template Design: The template design should include both combinational and sequential blocks. An example of the template design structure is as follows:

```
always @(posedge clk or negedge rst_n)
begin
if (!rst_n)
begin
    // Reset condition
end
else
begin
    // Non-reset condition
end
end
```

By implementing this program, students will be able to automatically generate template designs and testbench modules, facilitating the development and testing of Verilog modules.