

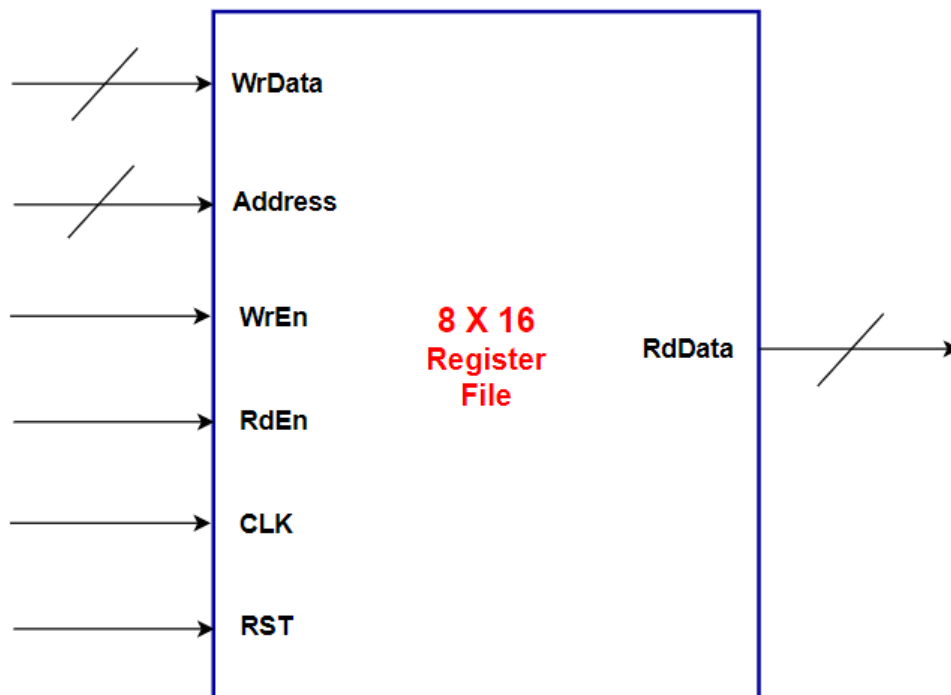
Assignment 4.2

Verilog Design of 8 x 16 Register File

:Register File Specification

- A register file consists of 8 registers, each register of 16-bit width.
- The register file has read data bus(**RdData**), write data bus(**WrData**) and one address bus (**Address**) used for both read and write operations.
- Each register can be read and written by applying a register address to be accessed.
- Only one operation (read or write) can be evaluated at a time.
- Write Operation is done only when **WrEn** is high
- Read operation is done only when **RdEn** is high.
- Read and Write operations are done on positive edge of **Clock**
- All the registers are cleared using Asynchronous **active low Reset** signal

- :Block Interface



Hint: To clear the register using asynchronous reset signal, you need to put the reset condition the first one as following example

```
(always @(posedge clock or posedge rst
begin
  (if(rst
  begin
    ; Reg_File[0] <= 16'b0
    ; Reg_File[1] <= 16'b0
    .....
    ; Reg_File[7] <= 16'b0
  end
  else
    .....
  )
  )
```

Hint: You do not need to make this code parameterized for .simplicity

Hint: Let Session_4/Lab_2 is your starting point for this assignment

- :Requirements

Design a register file with the above specifications using -1 Verilog language and determine Read and Write data buses .width and address bus width

Write a testbench to validate your design including at least 4 -2 scenarios two for writing in the registers and two for reading .from the registers