32 Cross 32 Register File with Reset Control

In this project we will design a 32 cross 32 Register File with Reset Control.

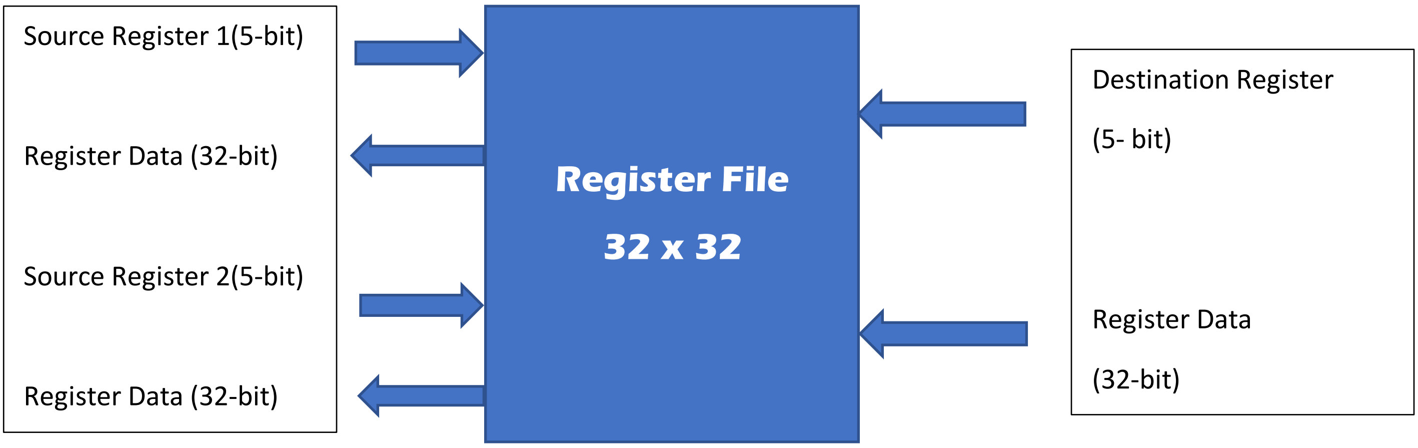
Software Used: -

Xilinx ISE Design Suite 14.7

INTRODUCTION

A register file or register bank is a group of registers, any of which can be randomly accessed.

- Can be implemented in Verilog as independent registers or as an array of registers, similar to memory.
- Register file often allow concurrent access.
- This design will allow two register reads and one register write on every clock cycle



Verilog Code: -

```
module regbank_v4(rdData1, rdData2, wrData, sr1, sr2, dr, write, reset, clk);     // Port List //
  input clk, write, reset;                                                        // Input Declaration//
  input [4:0] sr1, sr2, dr;                                   // Source and Destination Registers//
  input [31:0] wrData;
  output[31:0] rdData1, rdData2;                                                  //Output Declaration//
  integer k;


  reg[31:0]regfile[0:31];


  assign rdData1 = regfile[sr1];
  assign rdData2 = regfile[sr2];


 always@(posedge clk)
  begin
   if(reset)
     begin
    for(k = 0;k<32; k = k+1)
      regfile[k] <= 0;
    end
    else
                 begin
         if(write)
            regfile[dr] <= wrData;
      end
  end
endmodule
```