```
`timescale 1ns / 1ps

module registerfile(read_data1,read_data2,write_data,
                    read_rs1,read_rs2,w_reg,readsig1,readsig2,read_rd,clock,reset);
//////////////////////////// Register I/o
input [4:0] read_rs1;
input [4:0] read_rs2;
input [4:0] read_rd;
output reg [31:0]  read_data1;
output reg [31:0]  read_data2;
input [31:0] write_data;
input w_reg,readsig1,readsig2;
input clock,reset;
integer i;
reg [31:0] registerbank [0:31];
//////////////////////// Register Read data1 operation
always @(posedge clock or negedge reset)
begin
if(reset == 1'b0)
read_data1 <= 32'h0000;
else if(readsig1 == 1)
read_data1 <= registerbank[read_rs1];
end
//////////////////////// Register Read data2 operation

always @(posedge clock or negedge reset)
begin
if(reset == 1'b0)
read_data2 <= 32'h0000;
else if(readsig2 == 1)
read_data2 <= registerbank[read_rs2];
end
//////////////////////// Register Read write operation
always @(posedge clock)
begin
if(w_reg == 1)
registerbank[read_rd] <= write_data;
end



initial begin
  for(i=0;i<32;i=i+1)
  registerbank[i] = 32'h0001;
  end
endmodule
```