```
`timescale 1ns / 1ps
/****************************** C E C S  4 4 0 ******************************
 *
 * File Name:  regfile32.v
 * Project:    Lab_Assignment_2
 * Designer:   Rosswell Tiongco
 * Email:      rosswelltiongco@gmail.com
 * Rev. No.:   Version 1.0
 * Rev. Date:  Current Rev. Date
 *
 * Purpose: The Register File is the component that contains the "user
 *          registers" for the processor. Writing is done synchronously to
 *          the clock, whereas reading is done asynchronously. The core
 *          memory is a 32 wide by 32 deep register array
 *
 * Notes:
 *
 ****************************************************************************/
module regfile32(clk, reset, S_Addr, D, D_En, D_Addr, T_Addr, S, T);
    input         clk, reset, D_En;
    input  [ 4:0] D_Addr, S_Addr, T_Addr;
    input  [31:0] D;
    output [31:0] S, T;

    // Instantiating a 32x32 register array
    reg [31:0] memory [31:0];

    // Synchronous Write Section
    // Behaviorally modeled - sensitive to posedge clk, posedge rst
    always @ (posedge clk, posedge reset) begin
        // Clear register 0 if reset
        if (reset)
            memory[0] <= 32'h0;
        // Prevent writing to memory location 0
        else if (D_En == 1'b1 && D_Addr != 5'b0)
            memory[D_Addr] <= D;
        else
            memory[D_Addr] <= memory[D_Addr];
    end

    // Asynchronous read Section - using continuous assign statements
   assign S = (reset) ? 32'b0 : memory[S_Addr];
   assign T = (reset) ? 32'b0 : memory[T_Addr];

endmodule

```