```
1   `timescale 1ns / 1ps
2   /**********************************************************************************
3    *
4    * Author:   Jesus Luciano
5    * Filename: regfile32.v
6    * Date:     1/28/2019
7    * Version:  1.0
8    *
9    * Notes:    Register file contains 32 registers, each 32 bits wide
10   *           reset sets $r0 to zero, which cannot be overwritten
11   *           Module contains 2 32-bit outputs, S and T, which output the
12   *           contents of the register specified by S_Addr and T_Addr
13   *           D_Addr specifies the register to overwrite with input data
14   *           specified by D
15   *
16   **********************************************************************************/
17  module regfile32(clk, reset, S, T, D, S_Addr, T_Addr, D_Addr, D_En);
18
19      input         clk, reset, D_En;
20      input   [4:0] S_Addr, T_Addr, D_Addr;
21      input  [31:0] D;
22
23      output [31:0] S, T;
24
25      //create 2 dimensional array of 32 registers each 32 bits wide
26      reg    [31:0] reg32 [0:31];
27
28      //read uses 2 cts assign statements, asynchronous
29      assign S = reg32[S_Addr];
30      assign T = reg32[T_Addr];
31
32      //write behavioral, sensitive to clk and reset, synchronous
33      always @ (posedge clk, posedge reset) begin
34          if(reset)//assign register $r0 to zero, all others are uninitialized
35              reg32[0] <= 32'h0;
36          else begin
37              //only write if write enable is active and address is not zero
38              if(D_En == 1'b1 && D_Addr > 5'b0)
39                  reg32[D_Addr] <= D;
40              else
41                  reg32[D_Addr] <= reg32[D_Addr];
42
43          end
44      end
45
46  endmodule
47
```