```
    reg [31:0] A;
    wire [31:0] b;

    always @(posedge clock) A <= b;

module registerfile (Read1,Read2,WriteReg,WriteData,RegWrite,
Data1,Data2,clock);

   input [5:0] Read1,Read2,WriteReg; // the register numbers
to read or write
   input [31:0] WriteData; // data to write
   input RegWrite, // the write control
     clock; // the clock to trigger write
   output [31:0] Data1, Data2; // the register values read
   reg [31:0] RF [31:0]; // 32 registers each 32 bits long

   assign Data1 = RF[Read1];
   assign Data2 = RF[Read2];

   always begin
      // write the register with new value if Regwrite is
high
      @(posedge clock) if (RegWrite) RF[WriteReg] <=
WriteData;
   end
endmodule
```

FIGURE B.8.11 A MIPS register fi le written in behavioral Verilog. This register file writes on the rising clock edge.

Copyright © 2021 Elsevier Inc. All rights reserved