```vhdl
library IEEE;
use IEEE.std_logic_1164.all;
USE ieee.numeric_std.ALL;

ENTITY Register_File is
  PORT(
       RegWrite : IN  std_logic := '0';
       RegS     : IN  STD_LOGIC_VECTOR(4  DOWNTO 0) := "00000";
       RegT     : IN  STD_LOGIC_VECTOR(4  DOWNTO 0) := "00000";
       RegD     : IN  STD_LOGIC_VECTOR(4  DOWNTO 0) := "00000";
       WriteData: IN  STD_LOGIC_VECTOR(31 DOWNTO 0) :=
       "00000000000000000000000000000000";
       ReadData1: OUT STD_LOGIC_VECTOR(31 DOWNTO 0) :=
       "00000000000000000000000000000000";
       ReadData2: OUT STD_LOGIC_VECTOR(31 DOWNTO 0) := "00000000000000000000000000000000"
       );
END Register_File;

ARCHITECTURE behavior OF Register_File IS

    signal R0 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R1 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000001";
    signal R2 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000010";
    signal R3 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000011";
    signal R4 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R5 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R6 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R7 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R8 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R9 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R10 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R11 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R12 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R13 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R14 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R15 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R16 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R17 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R18 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R19 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R20 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R21 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R22 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R23 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R24 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R25 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R26 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R27 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R28 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R29 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R30 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";
    signal R31 : STD_LOGIC_VECTOR(31 DOWNTO 0):= "00000000000000000000000000000000";

    signal CLK : STD_LOGIC := '0';

  BEGIN

    CLOCK: PROCESS
        BEGIN
            WHILE 1 = 1 LOOP
                CLK <= NOT CLK;
            WAIT FOR 0.5 NS;
            END LOOP;
        END PROCESS CLOCK;

    PROCESS(CLK)
      BEGIN
        if (CLK = '1') Then
          case to_integer(unsigned(RegS)) is
```

```
68                when 0 =>
69                  ReadData1 <= R0;
70                when 1 =>
71                  ReadData1 <= R1;
72                when 2 =>
73                  ReadData1 <= R2;
74                when 3 =>
75                  ReadData1 <= R3;
76                when 4 =>
77                  ReadData1 <= R4;
78                when 5 =>
79                  ReadData1 <= R5;
80                when 6 =>
81                  ReadData1 <= R6;
82                when 7 =>
83                  ReadData1 <= R7;
84                when 8 =>
85                  ReadData1 <= R8;
86                when 9 =>
87                  ReadData1 <= R9;
88                when 10 =>
89                  ReadData1 <= R10;
90                when 11 =>
91                  ReadData1 <= R11;
92                when 12 =>
93                  ReadData1 <= R12;
94                when 13 =>
95                  ReadData1 <= R13;
96                when 14 =>
97                  ReadData1 <= R14;
98                when 15 =>
99                  ReadData1 <= R15;
100               when 16 =>
101                 ReadData1 <= R16;
102               when 17 =>
103                 ReadData1 <= R17;
104               when 18 =>
105                 ReadData1 <= R18;
106               when 19 =>
107                 ReadData1 <= R19;
108               when 20 =>
109                 ReadData1 <= R20;
110               when 21 =>
111                 ReadData1 <= R21;
112               when 22 =>
113                 ReadData1 <= R22;
114               when 23 =>
115                 ReadData1 <= R23;
116               when 24 =>
117                 ReadData1 <= R24;
118               when 25 =>
119                 ReadData1 <= R25;
120               when 26 =>
121                 ReadData1 <= R26;
122               when 27 =>
123                 ReadData1 <= R27;
124               when 28 =>
125                 ReadData1 <= R28;
126               when 29 =>
127                 ReadData1 <= R29;
128               when 30 =>
129                 ReadData1 <= R30;
130               when 31 =>
131                 ReadData1 <= R31;
132               when others =>
133               end case;
134         else
135         end if;
136     END PROCESS;
```

```
137
138       PROCESS(CLK)
139          BEGIN
140            if (CLK = '1') Then
141              case to_integer(unsigned(RegT)) is
142                when 0 =>
143                  ReadData2 <= R0;
144                when 1 =>
145                  ReadData2 <= R1;
146                when 2 =>
147                  ReadData2 <= R2;
148                when 3 =>
149                  ReadData2 <= R3;
150                when 4 =>
151                  ReadData2 <= R4;
152                when 5 =>
153                  ReadData2 <= R5;
154                when 6 =>
155                  ReadData2 <= R6;
156                when 7 =>
157                  ReadData2 <= R7;
158                when 8 =>
159                  ReadData2 <= R8;
160                when 9 =>
161                  ReadData2 <= R9;
162                when 10 =>
163                  ReadData2 <= R10;
164                when 11 =>
165                  ReadData2 <= R11;
166                when 12 =>
167                  ReadData2 <= R12;
168                when 13 =>
169                  ReadData2 <= R13;
170                when 14 =>
171                  ReadData2 <= R14;
172                when 15 =>
173                  ReadData2 <= R15;
174                when 16 =>
175                  ReadData2 <= R16;
176                when 17 =>
177                  ReadData2 <= R17;
178                when 18 =>
179                  ReadData2 <= R18;
180                when 19 =>
181                  ReadData2 <= R19;
182                when 20 =>
183                  ReadData2 <= R20;
184                when 21 =>
185                  ReadData2 <= R21;
186                when 22 =>
187                  ReadData2 <= R22;
188                when 23 =>
189                  ReadData2 <= R23;
190                when 24 =>
191                  ReadData2 <= R24;
192                when 25 =>
193                  ReadData2 <= R25;
194                when 26 =>
195                  ReadData2 <= R26;
196                when 27 =>
197                  ReadData2 <= R27;
198                when 28 =>
199                  ReadData2 <= R28;
200                when 29 =>
201                  ReadData2 <= R29;
202                when 30 =>
203                  ReadData2 <= R30;
204                when 31 =>
205                  ReadData2 <= R31;
```

```
206               when others =>
207               end case;
208         else
209         end if;
210       END PROCESS;
211
212        PROCESS(CLK)
213         BEGIN
214           if (RegWrite = '1') AND (CLK = '0' and CLK'event) Then
215             case to_integer(unsigned(RegD)) is
216               when 0 =>
217                 R0 <= WriteData;
218               when 1 =>
219                 R1 <= WriteData;
220               when 2 =>
221                 R2 <= WriteData;
222               when 3 =>
223                 R3 <= WriteData;
224               when 4 =>
225                 R4 <= WriteData;
226               when 5 =>
227                 R5 <= WriteData;
228               when 6 =>
229                 R6 <= WriteData;
230               when 7 =>
231                 R7 <= WriteData;
232               when 8 =>
233                 R8 <= WriteData;
234               when 9 =>
235                 R9 <= WriteData;
236               when 10 =>
237                 R10 <= WriteData;
238               when 11 =>
239                 R11 <= WriteData;
240               when 12 =>
241                 R12 <= WriteData;
242               when 13 =>
243                 R13 <= WriteData;
244               when 14 =>
245                 R14 <= WriteData;
246               when 15 =>
247                 R15 <= WriteData;
248               when 16 =>
249                 R16 <= WriteData;
250               when 17 =>
251                 R17 <= WriteData;
252               when 18 =>
253                 R18 <= WriteData;
254               when 19 =>
255                 R19 <= WriteData;
256               when 20 =>
257                 R20 <= WriteData;
258               when 21 =>
259                 R21 <= WriteData;
260               when 22 =>
261                 R22 <= WriteData;
262               when 23 =>
263                 R23 <= WriteData;
264               when 24 =>
265                 R24 <= WriteData;
266               when 25 =>
267                 R25 <= WriteData;
268               when 26 =>
269                 R26 <= WriteData;
270               when 27 =>
271                 R27 <= WriteData;
272               when 28 =>
273                 R28 <= WriteData;
274               when 29 =>
```

```
275                 R29 <= WriteData;
276               when 30 =>
277                 R30 <= WriteData;
278               when 31 =>
279                 R31 <= WriteData;
280               when others =>
281               end case;
282         else
283         end if;
284       END PROCESS;
285 
286   END behavior;
287 
288 
289 
```