```
1   ----------------------------------------------------------------------------
2   --
3   -- Title       : register_file
4   -- Design      : processor
5   -- Author      :
6   -- Company     :
7   --
8   ----------------------------------------------------------------------------
9   --
10  -- File        :
    C:\Users\gavin\Desktop\Study\ESE345\project\processor\processor\src\register file.vhd
11  -- Generated   : Sat Nov 30 15:35:50 2019
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ----------------------------------------------------------------------------
16  --
17  -- Description :
18  --
19  ----------------------------------------------------------------------------
20
21  --{{ Section below this comment is automatically maintained
22  --   and may be overwritten
23  --{entity {register file} architecture {behavioral}}
24
25  library IEEE;
26  use IEEE.std_logic_1164.all;
27  use ieee.numeric_std.all;
28
29  entity register_file is
30       port(
31            WrData : in STD_LOGIC_VECTOR(127 downto 0);
32            ins : in STD_LOGIC_VECTOR(24 downto 0);
33            old ins : in std_logic_vector(24 downto 0);
34            r1 : out STD_LOGIC_VECTOR(127 downto 0);
35            r2 : out STD_LOGIC_VECTOR(127 downto 0);
36            r3 : out STD_LOGIC_VECTOR(127 downto 0);
37            ins_out : out std_logic_vector(24 downto 0);
38            clk : in std_logic
39            );
40  end register_file;
41
42  --}} End of automatically maintained section
43
44  architecture behavioral of register file is
45  type RegisterFile is array(31 downto 0) of std_logic_vector(127 downto 0);
46  signal regs : RegisterFile := ((others=>(others=>'0')));

47  begin
48      process(clk, WrData, ins)
49      begin
50          if rising_edge(clk) then
51              if old_ins(24 downto 23) = "11" then
52                  if old_ins(19 downto 15) = "00000" then
53                  null;
54                  else
55                      regs(to integer(unsigned(old ins(4 downto 0)))) <= WrData;
56                  end if;
57              else
58                  regs(to integer(unsigned(old ins(4 downto 0)))) <= WrData;
59              end if;
60              if ins(24) = '0' then
61                  r1 <= regs(to_integer(unsigned(ins(4 downto 0))));
62                  r2 <= regs(to_integer(unsigned(ins(14 downto 10))));

63                  r3 <= regs(to_integer(unsigned(ins(19 downto 15))));
```

```
64              end if;
65              if ins(9 downto 5) = old_ins(4 downto 0) then
66                  r1 <= WrData;
67                  r2 <= regs(to_integer(unsigned(ins(14 downto 10))));

68                  r3 <= regs(to integer(unsigned(ins(19 downto 15))));
69              elsif ins(9 downto 5) = old ins(4 downto 0) then
70                  r2 <= WrData;
71                  r1 <= regs(to integer(unsigned(ins(9 downto 5))));

72                  r3 <= regs(to integer(unsigned(ins(19 downto 15))));
73              elsif ins(9 downto 5) = old ins(4 downto 0) then
74                  r3 <= WrData  ;
75                  r2 <= regs(to_integer(unsigned(ins(14 downto 10))));

76                  r1 <= regs(to_integer(unsigned(ins(9 downto 5))));
77              else
78                  r1 <= regs(to_integer(unsigned(ins(9 downto 5))));
79                  r2 <= regs(to integer(unsigned(ins(14 downto 10))));

80                  r3 <= regs(to integer(unsigned(ins(19 downto 15))));
81              end if;
82              ins out <= ins;
83          end if;
84      end process;
85
86       -- enter your statements here --
87
88  end behavioral;
89
```