

Alexandria University
Faculty of Engineering
CSED 2025



Computer Architecture Lab 1 Report

Submitted by

Marwan Yasser Sabri 20011870

Makario Michel Azer 20011982

Faculty of engineering Alex.
University

March 2024

Design Code:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use ieee.numeric_std.all;
4
5 entity register_file is
6   port (
7     clk : in  std_logic;
8     regwrite : in  std_logic;
9     dataIn : in  std_logic_vector(31 downto 0);
10    readRegA : in  unsigned(4 downto 0);
11    readRegB : in  unsigned(4 downto 0);
12    writeReg : in  unsigned(4 downto 0);
13    dataA : out std_logic_vector(31 downto 0);
14    dataB : out std_logic_vector(31 downto 0)
15  );
16 end register_file;
17
18 architecture Behavioral of register_file is
19   type registerFile is array (0 to 31) of std_logic_vector(31 downto 0);
20   signal registers : registerFile;
21 begin
22
23   process(clk)
24   begin
25     if rising_edge(clk) then
26       -- Write data in the first half cycle
27       if regwrite = '1' then
28         registers(to_integer(writeReg)) <= dataIn;
29       end if;
30     elsif falling_edge(clk) then
31       -- Read data in the second half cycle
32       dataA <= registers(to_integer(readRegA));
33       dataB <= registers(to_integer(readRegB));
34     end if;
35   end process;
36 end Behavioral;
```

Testbench Code:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use ieee.numeric_std.all;
4
5 entity register_file_tb is
6 end register_file_tb;
7
8 architecture tb_arch of register_file_tb is
9     constant CLK_PERIOD : time := 10 ns; -- Clock period (10 ns) -- clock per cycle
10
11     signal clk_tb      : std_logic := '0'; -- Testbench clock signal
12     signal regwrite_tb : std_logic := '0'; -- Testbench regwrite signal
13     signal dataIn_tb   : std_logic_vector(31 downto 0) := (others => '0'); -- Testbench data input signal
14     signal readRegA_tb : unsigned(4 downto 0) := to_unsigned(0, 5); -- Testbench readRegA signal
15     signal readRegB_tb : unsigned(4 downto 0) := to_unsigned(0, 5); -- Testbench readRegB signal
16     signal writeReg_tb : unsigned(4 downto 0) := to_unsigned(0, 5); -- Testbench writeReg signal
17     signal dataA_tb    : std_logic_vector(31 downto 0); -- Testbench dataA output signal
18     signal dataB_tb    : std_logic_vector(31 downto 0); -- Testbench dataB output signal
19
20     -- Component declaration
21     component register_file
22     port (
23         clk : in  std_logic;
24         regwrite : in  std_logic;
25         dataIn : in  std_logic_vector(31 downto 0);
26         readRegA : in  unsigned(4 downto 0);
27         readRegB : in  unsigned(4 downto 0);
28         writeReg : in  unsigned(4 downto 0);
29         dataA : out std_logic_vector(31 downto 0);
30         dataB : out std_logic_vector(31 downto 0)
31     );
32     end component;
33
34 begin
35     -- connect signals with ports --
36     -- Instantiate the register_file entity
37     dut : register_file
38     port map (
39         clk => clk_tb,
40         regwrite => regwrite_tb,
41         dataIn => dataIn_tb,
42         readRegA => readRegA_tb,
43         readRegB => readRegB_tb,
44         writeReg => writeReg_tb,
45         dataA => dataA_tb,
46         dataB => dataB_tb
47     );
48
49     -- Clock process
50     clk_process: process
51     begin
52         while now < 500 ns loop
53             clk_tb <= '0';
54             wait for CLK_PERIOD / 2;
55             clk_tb <= '1';
56             wait for CLK_PERIOD / 2;
57         end loop;
58         wait;
59     end process;
```

```

61  -- Stimulus process
62  stimulus_process: process
63  begin
64      wait for CLK_PERIOD/2;
65      -- Test case 1: Write to register 0, then read from it
66      regwrite_tb <= '1';
67      dataIn_tb <= x"12345678";
68      writeReg_tb <= to_unsigned(0, 5);
69      wait for CLK_PERIOD;
70      regwrite_tb <= '0';
71      readRegA_tb <= to_unsigned(0, 5);
72      readRegB_tb <= to_unsigned(1, 5);
73      wait for CLK_PERIOD;
74      assert dataA_tb = dataIn_tb report "Test case 1 failed: Incorrect data read from register 0." severity failure;
75      assert dataB_tb /= dataIn_tb report "Test case 1 failed: Incorrect data read from register 1." severity failure;
76
77      -- Test case 2: Write to register 10, then read from it
78      regwrite_tb <= '1';
79      dataIn_tb <= x"ABCDEF01";
80      writeReg_tb <= to_unsigned(10, 5);
81      wait for CLK_PERIOD;
82      regwrite_tb <= '0';
83      readRegA_tb <= to_unsigned(10, 5);
84      readRegB_tb <= to_unsigned(11, 5);
85      wait for CLK_PERIOD;
86      assert dataA_tb = dataIn_tb report "Test case 2 failed: Incorrect data read from register 10." severity failure;
87      assert dataB_tb = dataIn_tb report "Test case 2 failed: Incorrect data read from register 11." severity failure;
88
89
90      -- Test case 3: Not write to register 5, then read from it
91      regwrite_tb <= '0';
92      dataIn_tb <= x"ABCDEF02";
93      writeReg_tb <= to_unsigned(5, 5);      -- won't write
94      wait for CLK_PERIOD;
95      regwrite_tb <= '0';
96      readRegA_tb <= to_unsigned(5, 5);
97      readRegB_tb <= to_unsigned(5, 5);
98      wait for CLK_PERIOD;
99      assert dataA_tb /= dataIn_tb report "Test case 2 failed: Incorrect data read from register 10." severity failure;
100     assert dataB_tb /= dataIn_tb report "Test case 2 failed: Incorrect data read from register 11." severity failure;

103     -- Test case 4: Writing and reading from the same register in the same clock cycle
104     regwrite_tb <= '1';
105     dataIn_tb <= x"87654321";
106     writeReg_tb <= to_unsigned(2, 5);      -- write first then read
107     wait for CLK_PERIOD / 2;
108     readRegA_tb <= to_unsigned(2, 5);
109     readRegB_tb <= to_unsigned(2, 5);
110     wait for CLK_PERIOD / 2;
111     regwrite_tb <= '0';
112     wait for CLK_PERIOD;
113     assert dataA_tb = x"87654321" and dataB_tb = x"87654321" report "Test case 4 failed: Incorrect data read from register 20."
severity failure;
114
115     -- Test case 5: (Not writing) and reading from the same register in the same clock cycle
116     regwrite_tb <= '0';
117     dataIn_tb <= x"27654321";
118     writeReg_tb <= to_unsigned(20, 5);      -- write first then read
119     wait for CLK_PERIOD / 2;
120     readRegA_tb <= to_unsigned(20, 5);
121     readRegB_tb <= to_unsigned(20, 5);
122     wait for CLK_PERIOD / 2;
123     regwrite_tb <= '0';
124     wait for CLK_PERIOD;
125     assert dataA_tb /= x"27654321" and dataB_tb /= x"27654321" report "Test case 5 failed: Incorrect data read from register 20."
severity failure;
126
127     -- Test 6: write data and read other data in the same clock cycle
128     regwrite_tb <= '1';
129     dataIn_tb <= x"47654321";
130     writeReg_tb <= to_unsigned(21, 5);      -- write first then read
131     wait for CLK_PERIOD / 2;
132     readRegA_tb <= to_unsigned(20, 5);
133     readRegB_tb <= to_unsigned(20, 5);
134     wait for CLK_PERIOD / 2;
135     regwrite_tb <= '0';
136     wait for CLK_PERIOD;
137     assert dataA_tb /= x"47654321" and dataB_tb /= x"87654321" report "Test case 6 failed: Incorrect data read from register 20."
severity failure;
138
139     assert dataA_tb = dataIn_tb and dataB_tb = dataIn_tb report "Test case 6 failed: Incorrect data read from register 20." severity
failure;

```

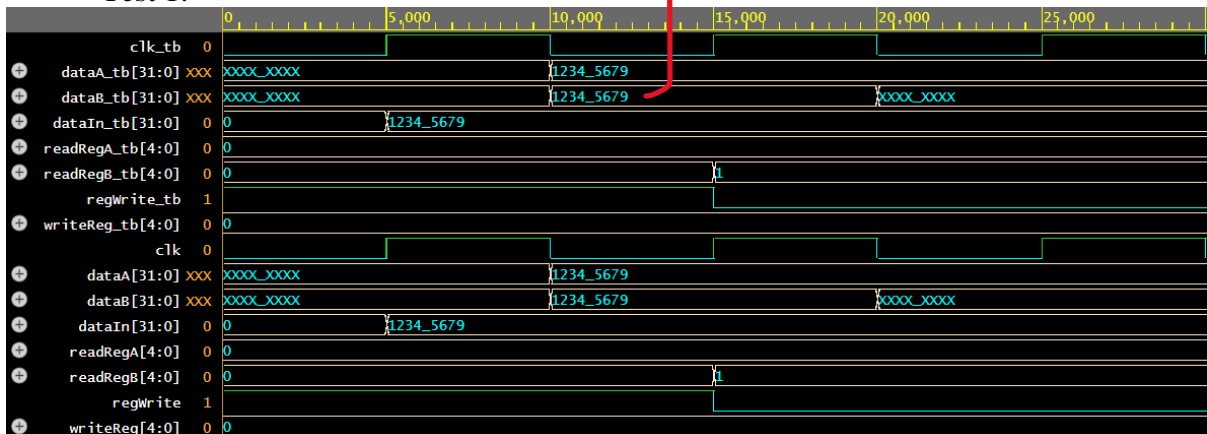
```

141 -- Test 7: many operations
142 readRegA_tb <= to_unsigned(2, 5);
143 readRegB_tb <= to_unsigned(10, 5);
144 wait for CLK_PERIOD;
145 assert dataA_tb = x"87654321" and dataB_tb = x"ABCDEF01" report "Test case 7 failed: Incorrect data read from register 20."
severity failure;
146 regwrite_tb <= '1';
147 dataIn_tb <= x"11111111";
148 writeReg_tb <= to_unsigned(22, 5);
149 wait for CLK_PERIOD / 2;
150 readRegA_tb <= to_unsigned(22, 5);
151 wait for CLK_PERIOD / 2;
152 regwrite_tb <= '0';
153 wait for CLK_PERIOD;
154 assert dataA_tb = x"11111111" and dataA_tb /= x"87654321" report "Test case 7 failed: Incorrect data read from register 20."
severity failure;
155
156 End simulation
157 wait;
158 end process;
159
160 end tb_arch;

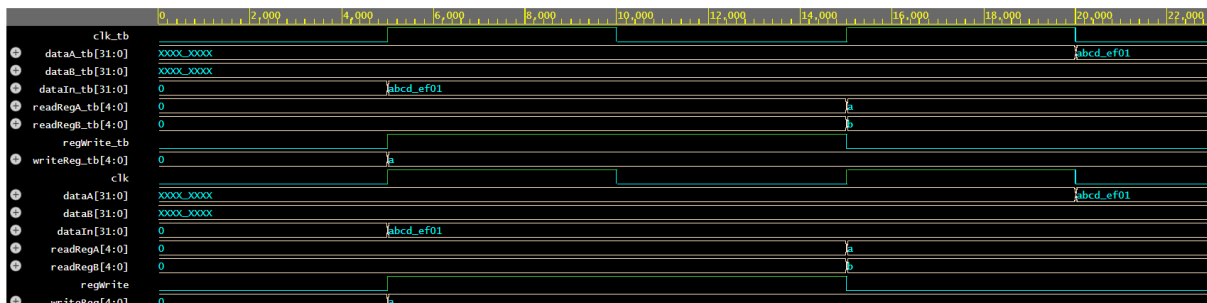
```

Screenshots of the output:

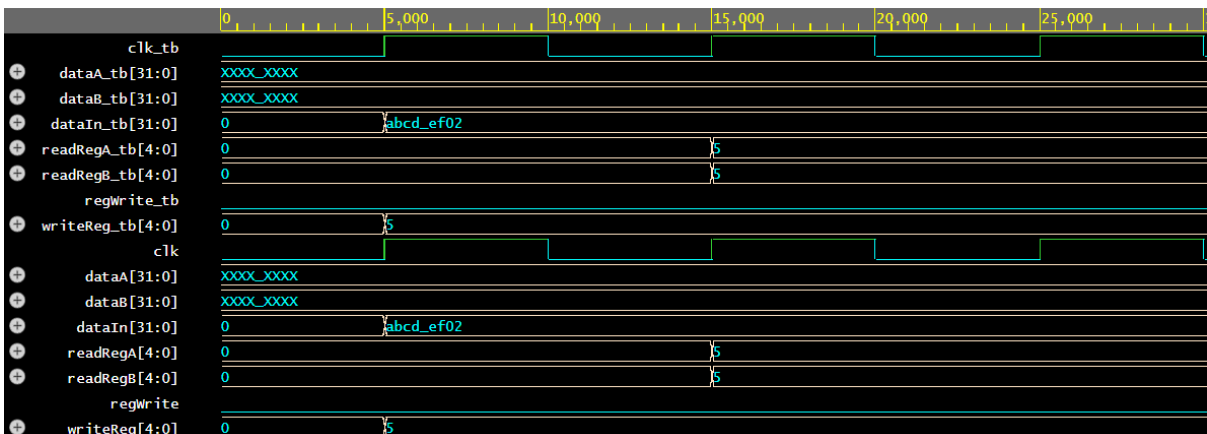
- Test 1:



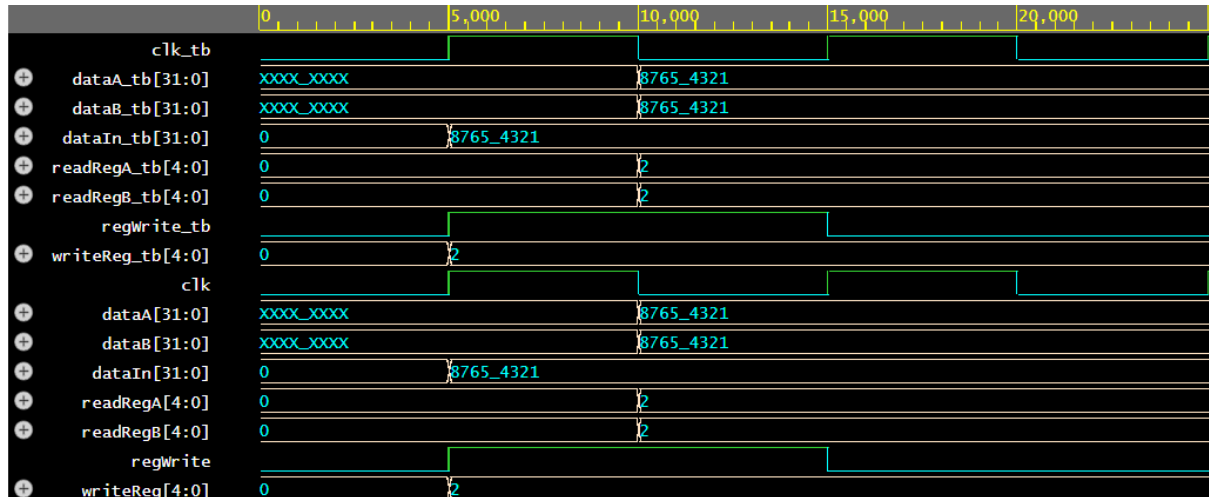
- Test 2:



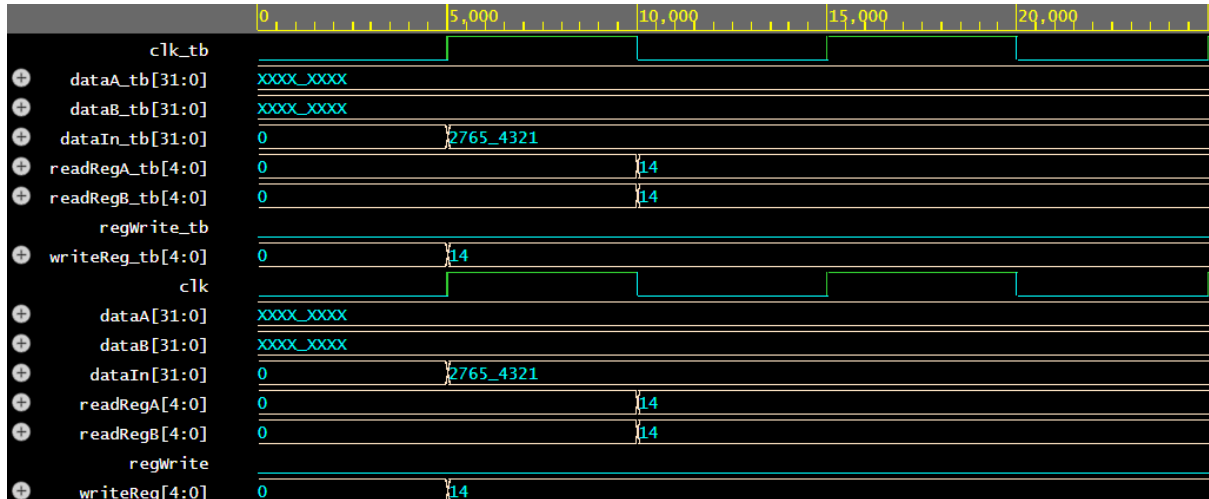
- Test 3:



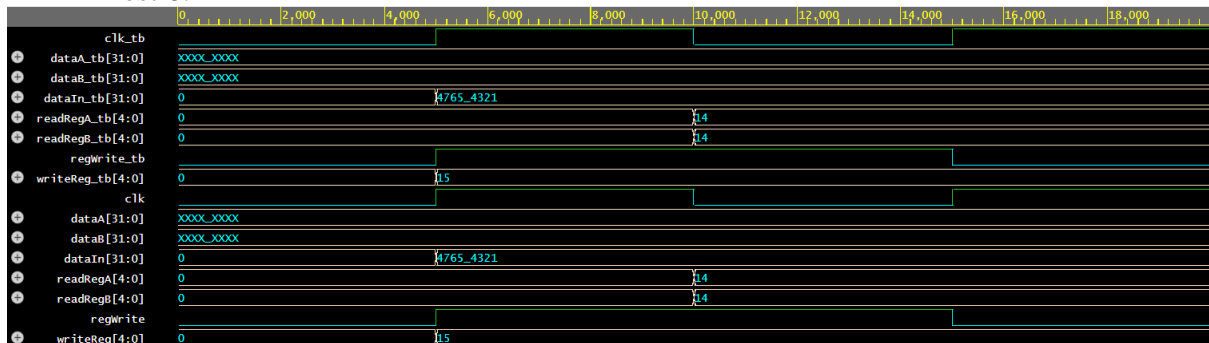
- Test 4:



- Test 5:



- Test 6:



- Test 7:



- All tests:

