Ammar Qureshi-14318618

```
----------------------------------------------------------------------------
--- register file implementation
----------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


entity regFiles is

Port ( src_s0 : in std_logic;
src_s1 : in std_logic;
src_s2 : in std_logic;

des_A0 : in std_logic;
des_A1 : in std_logic;
des_A2 : in std_logic;

Clk : in std_logic;
data_src : in std_logic;

data : in std_logic_vector(15 downto 0);

reg0 : out std_logic_vector(15 downto 0);
reg1 : out std_logic_vector(15 downto 0);
reg2 : out std_logic_vector(15 downto 0);
reg3 : out std_logic_vector(15 downto 0);
reg4 : out std_logic_vector(15 downto 0);
reg5 : out std_logic_vector(15 downto 0);
reg6 : out std_logic_vector(15 downto 0);
reg7 : out std_logic_vector(15 downto 0));
end regFiles;

architecture Behavioral of regFiles is


COMPONENT register4
PORT(
D : IN std_logic_vector(15 downto 0);
load : IN std_logic;
Clk : IN std_logic;
Q : OUT std_logic_vector(15 downto 0)
);

END COMPONENT;

--3 to 8 decoder.
COMPONENT three_to_eight_decoder
PORT(
a0 : IN std_logic;
a1 : IN std_logic;
a2 : IN std_logic;

q0 : OUT std_logic;
q1 : OUT std_logic;
q2 : OUT std_logic;
q3 : OUT std_logic;
q4 : OUT std_logic;
q5 : OUT std_logic;
q6 : OUT std_logic;
q7 : OUT std_logic
);
END COMPONENT;



--2 to 1 line MUX

COMPONENT mux_16bit
PORT(
in0 : IN std_logic_vector(15 downto 0);
in1 : IN std_logic_vector(15 downto 0);
s : IN std_logic;
Z : OUT std_logic_vector(15 downto 0)
);
END COMPONENT;



--8 to 1 line multiplexer
COMPONENT mux8
PORT(
in0 : IN std_logic_vector(15 downto 0);
in1 : IN std_logic_vector(15 downto 0);
in2 : IN std_logic_vector(15 downto 0);
in3 : IN std_logic_vector(15 downto 0);
in4 : IN std_logic_vector(15 downto 0);
in5 : IN std_logic_vector(15 downto 0);
in6 : IN std_logic_vector(15 downto 0);
in7 : IN std_logic_vector(15 downto 0);
s0 : IN std_logic;
s1 : IN std_logic;
s2 : IN std_logic;
Z : OUT std_logic_vector(15 downto 0)
);
END COMPONENT;



--signals

signal load_reg0, load_reg1, load_reg2, load_reg3 ,load_reg4
,load_reg5 , load_reg6 , load_reg7  : std_logic;

signal reg0_q, reg1_q, reg2_q, reg3_q, reg4_q, reg5_q,
reg6_q, reg7_q,
data_src_mux_out, src_reg : std_logic_vector(15 downto 0);
```

```
begin


--port maps for registers 0 to 7

-- register 0
reg00: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg0,
Clk => Clk,
Q => reg0_q
);
-- register 1
reg01: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg1,
Clk => Clk,
Q => reg1_q
);

-- register 2
reg02: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg2,
Clk => Clk,
Q => reg2_q
);

-- register 3
reg03: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg3,
Clk => Clk,
Q => reg3_q
);

-- register 4
reg04: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg4,
Clk => Clk,
Q => reg4_q
);

-- register 5
reg05: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg5,
Clk => Clk,
Q => reg5_q
);

-- register 6
reg06: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg6,
Clk => Clk,
Q => reg6_q
);
```

Ammar Qureshi-14318618

```
-- register 7
reg07: register4 PORT MAP(
D => data_src_mux_out,
load => load_reg7,
Clk => Clk,
Q => reg7_q
);


--destination register decoder


des_decoder_3to8: three_to_eight_decoder PORT MAP(

a0 => des_A0,
a1 => des_A1,
a2 => des_A2,

q0 => load_reg0,
q1 => load_reg1,
q2 => load_reg2,
q3 => load_reg3,
q4 => load_reg4,
q5 => load_reg5,
q6 => load_reg6,
q7 => load_reg7

);


-- 2 to 1 data source multiplexer
data_src_mux2_4bit: mux_16bit PORT MAP(
in0 => data,
in1 => src_reg,
s => data_src,
Z => data_src_mux_out
);


-- 4 to 1 source register multiplexer
Inst_mux4_4bit: mux8 PORT MAP(
in0 => reg0_q,
in1 => reg1_q,
in2 => reg2_q,
in3 => reg3_q,
in4 => reg4_q,
in5 => reg5_q,
in6 => reg6_q,
in7 => reg7_q,


s0 => src_s0,
s1 => src_s1,
s2 => src_s2,
Z => src_reg
);


reg0 <= reg0_q;
reg1 <= reg1_q;
reg2 <= reg2_q;
```

```vhdl
reg3 <= reg3_q;
reg4 <= reg4_q;
reg5 <= reg5_q;
reg6 <= reg6_q;
reg7 <= reg7_q;

end Behavioral;


----------------------------------------------------------------------------
-- register-16 bits
----------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


entity register4 is
port ( D : in std_logic_vector(15 downto 0);
load, Clk : in std_logic;
Q : out std_logic_vector(15 downto 0));
end register4;

architecture Behavioral of register4 is
begin
process(Clk)
begin
if (rising_edge(Clk)) then
if load='1' then
Q<=D after 5 ns;
end if;
end if;
end process;
end Behavioral;


----------------------------------------------------------------------------
-- 3 to 8 decoder 16 bits implementation
----------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity three_to_eight_decoder is
port(
a0: in std_logic;
a1: in std_logic;
a2: in std_logic;
q0: out std_logic;
q1: out std_logic;
q2: out std_logic;
q3: out std_logic;
q4: out std_logic;
q5: out std_logic;
q6: out std_logic;
q7: out std_logic
);

end three_to_eight_decoder;
```

Ammar Qureshi-14318618

```
architecture behvOfDecoder of three_to_eight_decoder is
begin
q0<= not a0 and not  a1 and not  a2;
q1<= not a0 and not  a1 and      a2;
q2<= not a0 and      a1 and not  a2;
q3<= not a0 and      a1 and      a2;
q4<=     a0 and not  a1 and not  a2;
q5<=     a0 and not  a1 and      a2;
q6<=     a0 and      a1 and not  a2;
q7<=     a0 and      a1 and      a2;

end behvOfDecoder;




----------------------------------------------------------------------------
--mux 2 to 1 (16 bits) implementation
----------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;



entity mux_16bit is
port ( in0 : in std_logic_vector(15 downto 0);
in1 : in std_logic_vector(15 downto 0);
s : in std_logic;
Z : out std_logic_vector(15 downto 0));
end mux_16bit;
architecture Behavioral of mux_16bit is
begin
Z <= in0 after 5 ns when s='0' else
in1 after 5 ns when s='1'else
"0000000000000000" after 5 ns;
end Behavioral;


----------------------------------------------------------------------------
-- mux 8 to 1 (16 bits) implementation
----------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mux8 is
Port (
in0, in1, in2, in3, in4, in5, in6, in7: in std_logic_vector (15 downto 0);
s0, s1 ,s2: in std_logic;

Z: out std_logic_vector (15 downto 0));
end mux8;
```

Ammar Qureshi-14318618

```
architecture behavioural of mux8 is
begin
Z <=
in0 after 5ns when s0 = '0' and s1 = '0' and s2 = '0' else
in1 after 5ns when s0 = '0' and s1 = '0' and s2 = '1' else
in2 after 5ns when s0 = '0' and s1 = '1' and s2 = '0' else
in3 after 5ns when s0 = '0' and s1 = '1' and s2 = '1' else
in4 after 5ns when s0 = '1' and s1 = '0' and s2 = '0' else
in5 after 5ns when s0 = '1' and s1 = '0' and s2 = '1' else
in6 after 5ns when s0 = '1' and s1 = '1' and s2 = '0' else
in7 after 5ns when s0 = '1' and s1 = '1' and s2 = '1' else
X"0000" after 5ns;
end behavioural;



----------------------------------------------------------------------------
--3 to 8 decoder test-bench
----------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY decoder3to8_tb IS
END decoder3to8_tb;

ARCHITECTURE behavior OF decoder3to8_tb IS

   -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT three_to_eight_decoder
    PORT(
         a0 : IN  std_logic;
         a1 : IN  std_logic;
         a2 : IN  std_logic;
         q0 : OUT  std_logic;
         q1 : OUT  std_logic;
         q2 : OUT  std_logic;
         q3 : OUT  std_logic;
         q4 : OUT  std_logic;
         q5 : OUT  std_logic;
         q6 : OUT  std_logic;
         q7 : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal a0 : std_logic := '0';
   signal a1 : std_logic := '0';
   signal a2 : std_logic := '0';

    --Outputs
   signal q0 : std_logic;
   signal q1 : std_logic;
   signal q2 : std_logic;
   signal q3 : std_logic;
   signal q4 : std_logic;
   signal q5 : std_logic;
   signal q6 : std_logic;
   signal q7 : std_logic;

   constant clk_period : time := 10 ns;
```

Ammar Qureshi-14318618

```
BEGIN
    -- Instantiate the Unit Under Test (UUT)
   uut: three_to_eight_decoder PORT MAP (
          a0 => a0,
          a1 => a1,
          a2 => a2,
          q0 => q0,
          q1 => q1,
          q2 => q2,
          q3 => q3,
          q4 => q4,
          q5 => q5,
          q6 => q6,
          q7 => q7
        );


   -- Stimulus process
   stim_proc: process

    begin


        --test enable for reg 0
        wait for 10 ns;
        a0<='0';
        a1<='0';
        a2<='0';

        --test enable for reg 1
        wait for 10 ns;
        a0<='0';
        a1<='0';
        a2<='1';

        --test enable for reg 2
        wait for 10 ns;
        a0<='0';
        a1<='1';
        a2<='0';


        --test enable for reg 3
        wait for 10 ns;
        a0<='0';
        a1<='1';
        a2<='1';



      --test enable for reg 4
        wait for 10 ns;
        a0<='1';
        a1<='0';
        a2<='0';
```

```
       --test enable for reg 5
        wait for 10 ns;
        a0<='1';
        a1<='0';
        a2<='1';


    --test enable for reg 6
        wait for 10 ns;
        a0<='1';
        a1<='1';
        a2<='0';



    --test enable for reg 7
        wait for 10 ns;
        a0<='1';
        a1<='1';
        a2<='1';


     -- wait;
   end process;

END;





--------------------------------------------------------------------------
-- mux 2 to 1 (16 bits) test-bench
--------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY mux2to1_Tb IS
END mux2to1_Tb;

ARCHITECTURE behavior OF mux2to1_Tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT mux_16bit
    PORT(
         in0 : IN  std_logic_vector(15 downto 0);
         in1 : IN  std_logic_vector(15 downto 0);
         s : IN  std_logic;
         Z : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;


   --Inputs
   signal in0 : std_logic_vector(15 downto 0) := (others => '0');
   signal in1 : std_logic_vector(15 downto 0) := (others => '0');
   signal s : std_logic := '0';

    --Outputs
   signal Z : std_logic_vector(15 downto 0);
```

```
BEGIN
    -- Instantiate the Unit Under Test (UUT)
   uut: mux_16bit PORT MAP (
          in0 => in0,
          in1 => in1,
          s => s,
          Z => Z
        );




   -- Stimulus process
   stim_proc: process
   begin
        in0<=X"1c2d";
        in1<=X"ffff";


      wait for 100 ns;
-----select input 0 ---------------------------
        s<='0';

   wait for 100 ns;
-----select input 1 ---------------------------
        s<='1';
   end process;

END;



----------------------------------------------------------------------------
-- mux 8 to 1 (16  bits) test-bench
----------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY mux_tb IS
END mux_tb;



ARCHITECTURE behavior OF mux_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT mux8
    PORT(
         in0 : IN  std_logic_vector(15 downto 0);
         in1 : IN  std_logic_vector(15 downto 0);
         in2 : IN  std_logic_vector(15 downto 0);
         in3 : IN  std_logic_vector(15 downto 0);
         in4 : IN  std_logic_vector(15 downto 0);
         in5 : IN  std_logic_vector(15 downto 0);
         in6 : IN  std_logic_vector(15 downto 0);
         in7 : IN  std_logic_vector(15 downto 0);
```

```
          s0 : IN  std_logic;
          s1 : IN  std_logic;
          s2 : IN  std_logic;
          Z : OUT  std_logic_vector(15 downto 0)
         );
     END COMPONENT;


    --Inputs
    signal in0 : std_logic_vector(15 downto 0) := (others => '0');
    signal in1 : std_logic_vector(15 downto 0) := (others => '0');
    signal in2 : std_logic_vector(15 downto 0) := (others => '0');
    signal in3 : std_logic_vector(15 downto 0) := (others => '0');
    signal in4 : std_logic_vector(15 downto 0) := (others => '0');
    signal in5 : std_logic_vector(15 downto 0) := (others => '0');
    signal in6 : std_logic_vector(15 downto 0) := (others => '0');
    signal in7 : std_logic_vector(15 downto 0) := (others => '0');
    signal s0 : std_logic := '0';
    signal s1 : std_logic := '0';
    signal s2 : std_logic := '0';

     --Outputs
    signal Z : std_logic_vector(15 downto 0);


BEGIN

     -- Instantiate the Unit Under Test (UUT)
    uut: mux8 PORT MAP (
           in0 => in0,
           in1 => in1,
           in2 => in2,
           in3 => in3,
           in4 => in4,
           in5 => in5,
           in6 => in6,
           in7 => in7,
           s0 => s0,
           s1 => s1,
           s2 => s2,
           Z => Z
         );

    -- Stimulus process
    stim_proc: process
    begin

      in0<="1111111111111111";
      in1<="1000110111111001";
      in2<="0110110111111010";
      in3<="0110010111111010";
      in4<="0110110111111010";
      in5<="0110110011111010";
      in6<="0110110110111010";
      in7<="0110110111110010";
```

Ammar Qureshi-14318618

```
---------in0 selected--------------------------
      wait for 50 ns;
    s0<='0';
    s1<='0';
    s2<='0';



---------in1 selected--------------------------

  wait for 50 ns;
    s0<='0';
    s1<='0';
    s2<='1';

---------in2 selected--------------------------

  wait for 50 ns;
    s0<='0';
    s1<='1';
    s2<='0';

---------in3 selected--------------------------
  wait for 50 ns;
    s0<='0';
    s1<='1';
    s2<='1';

---------in4 selected--------------------------
  wait for 50 ns;
    s0<='1';
    s1<='0';
    s2<='0';

---------in5 selected--------------------------
  wait for 50 ns;
    s0<='1';
    s1<='0';
    s2<='1';

---------in6 selected--------------------------

  wait for 50 ns;
    s0<='1';
    s1<='1';
    s2<='0';

---------in7 selected--------------------------

  wait for 50 ns;
    s0<='1';
    s1<='1';
    s2<='1';


   end process;

END;
```

Ammar Qureshi-14318618

```
--------------------------------------------------------------------------
-- register files (8 registers-16 bits) test-bench
--------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY regFiles_tb IS
END regFiles_tb;

ARCHITECTURE behavior OF regFiles_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT regFiles
    PORT(
         src_s0 : IN  std_logic;
         src_s1 : IN  std_logic;
         src_s2 : IN  std_logic;
         des_A0 : IN  std_logic;
         des_A1 : IN  std_logic;
         des_A2 : IN  std_logic;
         Clk : IN  std_logic;
         data_src : IN  std_logic;
         data : IN  std_logic_vector(15 downto 0);
         reg0 : OUT  std_logic_vector(15 downto 0);
         reg1 : OUT  std_logic_vector(15 downto 0);
         reg2 : OUT  std_logic_vector(15 downto 0);
         reg3 : OUT  std_logic_vector(15 downto 0);
         reg4 : OUT  std_logic_vector(15 downto 0);
         reg5 : OUT  std_logic_vector(15 downto 0);
         reg6 : OUT  std_logic_vector(15 downto 0);
         reg7 : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;

   --Inputs
   signal src_s0 : std_logic := '0';
   signal src_s1 : std_logic := '0';
   signal src_s2 : std_logic := '0';
   signal des_A0 : std_logic := '0';
   signal des_A1 : std_logic := '0';
   signal des_A2 : std_logic := '0';
   signal Clk : std_logic := '0';
   signal data_src : std_logic := '0';
   signal data : std_logic_vector(15 downto 0) := (others => '0');

 --Outputs
   signal reg0 : std_logic_vector(15 downto 0);
   signal reg1 : std_logic_vector(15 downto 0);
   signal reg2 : std_logic_vector(15 downto 0);
   signal reg3 : std_logic_vector(15 downto 0);
   signal reg4 : std_logic_vector(15 downto 0);
   signal reg5 : std_logic_vector(15 downto 0);
   signal reg6 : std_logic_vector(15 downto 0);
   signal reg7 : std_logic_vector(15 downto 0);
```

Ammar Qureshi-14318618

```
   -- Clock period definitions
   constant Clk_period : time := 10 ns;
 

BEGIN

    -- Instantiate the Unit Under Test (UUT)
   uut: regFiles PORT MAP (
          src_s0 => src_s0,
          src_s1 => src_s1,
          src_s2 => src_s2,
          des_A0 => des_A0,
          des_A1 => des_A1,
          des_A2 => des_A2,
          Clk => Clk,
          data_src => data_src,
          data => data,
          reg0 => reg0,
          reg1 => reg1,
          reg2 => reg2,
          reg3 => reg3,
          reg4 => reg4,
          reg5 => reg5,
          reg6 => reg6,
          reg7 => reg7
        );

   -- Clock process definitions
   Clk_process :process
   begin
        Clk <= '0';
        wait for Clk_period/2;
        Clk <= '1';
        wait for Clk_period/2;
   end process;
 

   -- Stimulus process
   stim_proc: process
   begin

   --load arbitrary hex values into registers.
    --load hex value 3f3f to register 0
     data<=X"3f3f";
     des_A0<='0';
     des_A1<='0';
     des_A2<='0';
     data_src<='0';

     wait for 30 ns;
      --load hex value 1e5c to register 1
     data<=X"1e5c";
     des_A0<='0';
     des_A1<='0';
     des_A2<='1';
     data_src<='0';
```

```
wait for 30 ns;
--load hex value 2d2d to register 2
data<=X"2d2d";
des_A0<='0';
des_A1<='1';
des_A2<='0';
data_src<='0';



wait for 30 ns;
--load hex value 5a5a to register 3
data<=X"5a5a";
des_A0<='0';
des_A1<='1';
des_A2<='1';
data_src<='0';



wait for 30 ns;
--load hex value 1c2c to register 4
data<=X"1c2c";
des_A0<='1';
des_A1<='0';
des_A2<='0';
data_src<='0';

wait for 30 ns;
--load hex value 5B5B to register 5
data<=X"5b5b";
des_A0<='1';
des_A1<='0';
des_A2<='1';
data_src<='0';


wait for 30 ns;
--load hex value 7d7d to register 6
data<=X"7d7d";
des_A0<='1';
des_A1<='1';
des_A2<='0';
data_src<='0';

wait for 30 ns;
--load hex value ffff to register 7
data<=X"ffff";
des_A0<='1';
des_A1<='1';
des_A2<='1';
data_src<='0';
```

```
     -Transferring data from register 0 to register 1-------------------


     wait for 30 ns;
     -------------- selecting register 0 as source register----------------
     src_s0<='0';
     src_s1<='0';
     src_s2<='0';

    -------------- selecting register 1 as destination register------------

     des_A0<='0';
     des_A1<='0';
     des_A2<='1';


    -------------- select input to 1 to enable transfer data(with select
line =0 you can load your own data)------------------------
     data_src<='1';



    -------Transferring data from register 3 to register 7-----------------

    -------------- selecting register 3 as source register----------------

     wait for 100 ns;
     src_s0<='0';
     src_s1<='1';
     src_s2<='1';



 -------------- selecting register 7 as destination register-----------

     des_A0<='1';
     des_A1<='1';
     des_A2<='1';

     data_src<='1';


      wait;
   end process;

END;
```

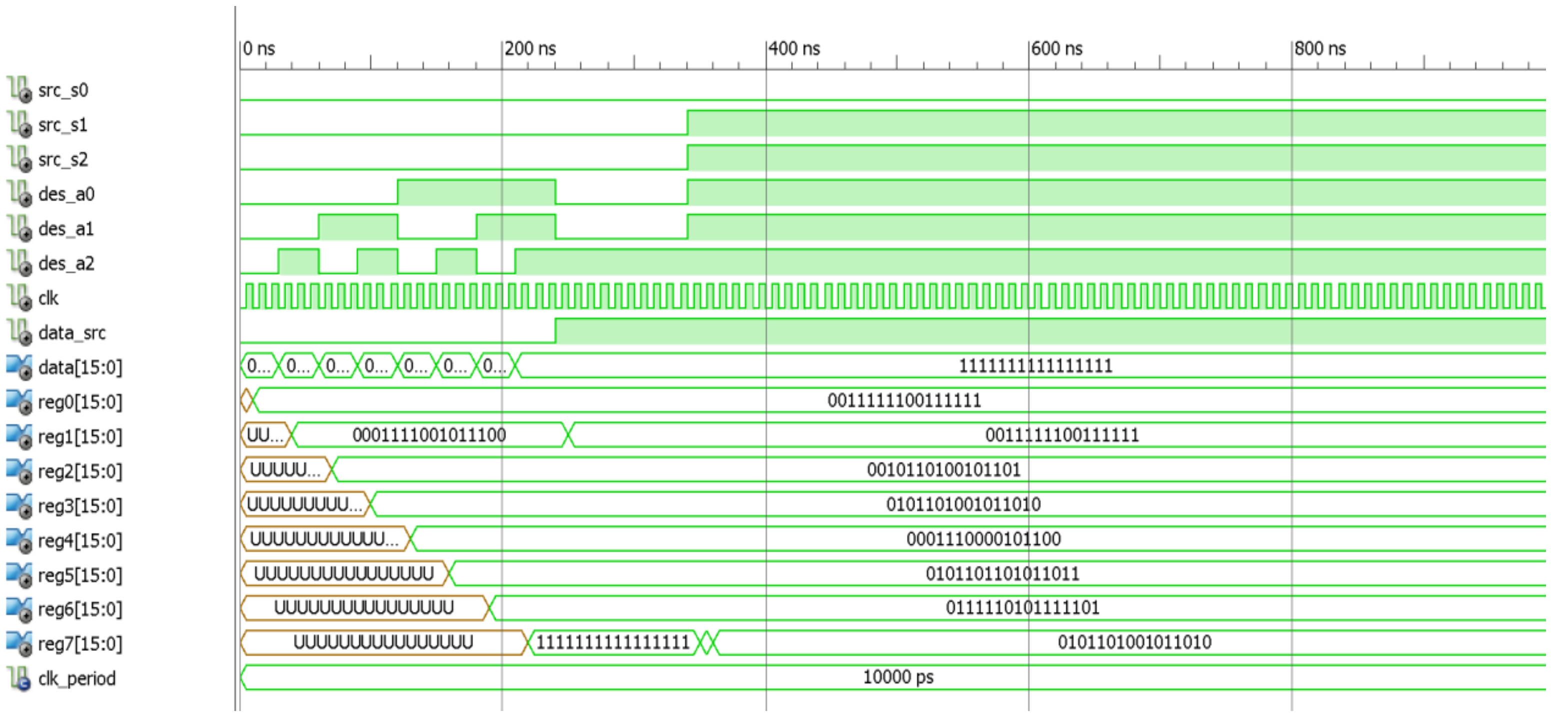
Ammar Qureshi-14318618

```
----------------------------------------------------------------------------
--single register 16 bits test-bench
----------------------------------------------------------------------------

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY register_tb IS
END register_tb;

ARCHITECTURE behavior OF register_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT register4
    PORT(
         D : IN  std_logic_vector(15 downto 0);
         load : IN  std_logic;
         Clk : IN  std_logic;
         Q : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;


   --Inputs
   signal D : std_logic_vector(15 downto 0) := (others => '0');
   signal load : std_logic := '0';
   signal Clk : std_logic := '0';

    --Outputs
   signal Q : std_logic_vector(15 downto 0);

   -- Clock period definitions
   constant Clk_period : time := 10 ns;


BEGIN

    -- Instantiate the Unit Under Test (UUT)
   uut: register4 PORT MAP (
          D => D,
          load => load,
          Clk => Clk,
          Q => Q
        );

   -- Clock process definitions
   Clk_process :process
   begin
        Clk <= '0';
        wait for Clk_period/2;
        Clk <= '1';
        wait for Clk_period/2;
   end process;
   -- Stimulus process
   stim_proc: process
   begin


----data wont be transfered to register as load enable bit is set to zero
        D<="1111111111111111";
        wait for 100 ns;
        load<='0';


----- loading 0111111111111111 into register when load enable = 1----------
------
        wait for 100 ns;
        load<='1';


------ load another value into register------------------------------------
----------
        wait for 100 ns;
        D<="0000000000000000";
    wait;

   end process;

END;
```

Ammar Qureshi-14318618

Register file test-bench simulation

Initially all registers are undefined. Arbitrary hex values are loaded into each register 0 to 7. To test the correct transfer of any register to any other register, data of register 0 is transferred to register 1 and data of register 3 is transferred to register 7. It is evident from the image below after around 250 ns contents of register 1 has changed and is identical to contents in register 0. At around 350 ns contents of register 7 are identical to contents of register 3.

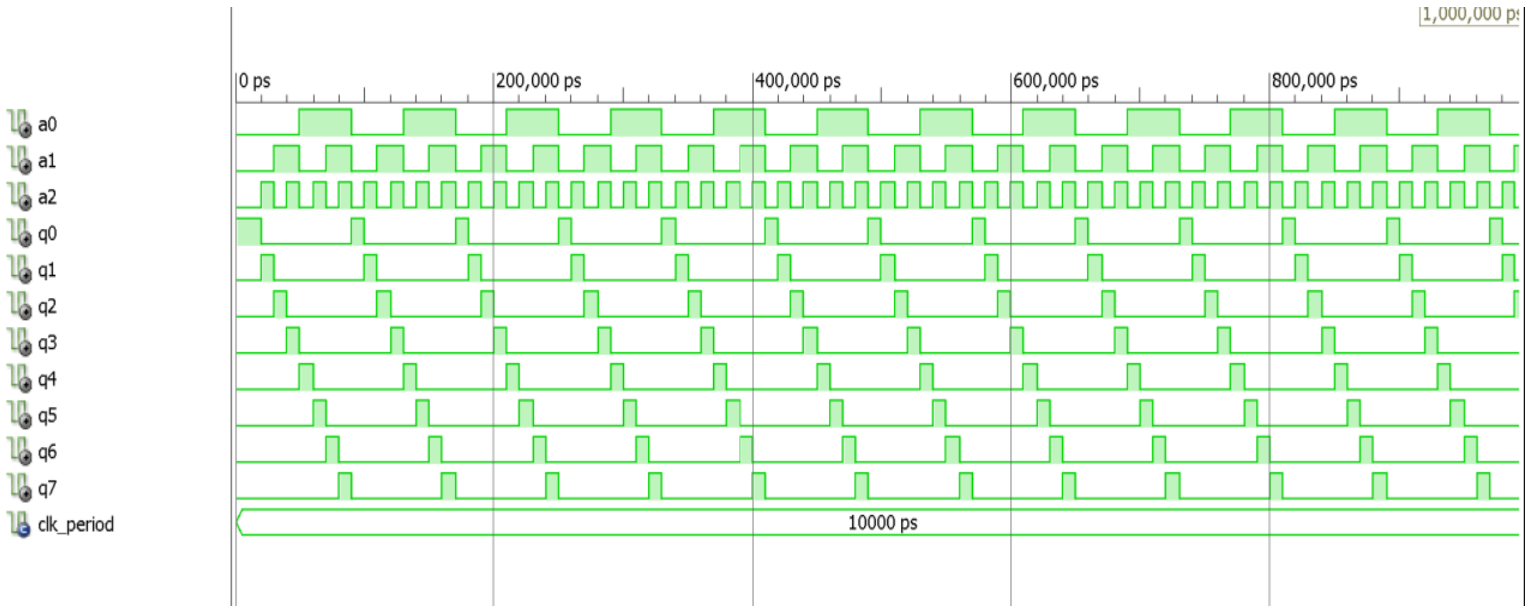


Ammar Qureshi-14318618

Simulation Results

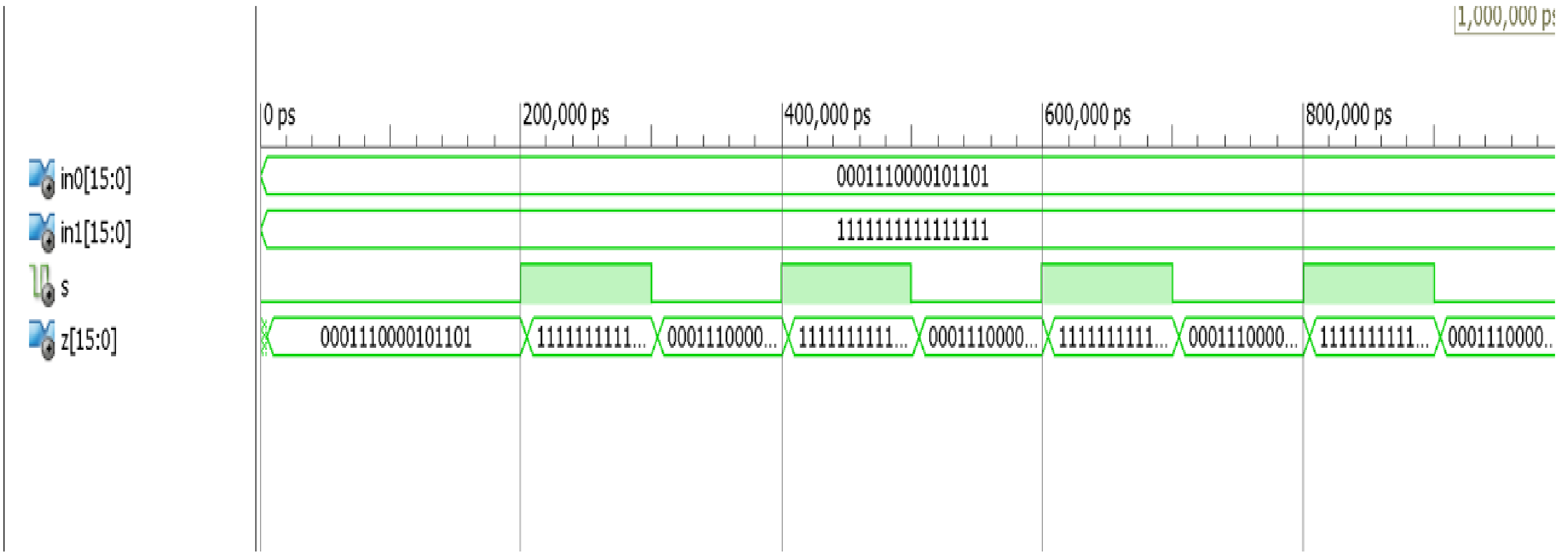
Decoder 3 to 8

When a0,a1,a2 are all low, q0 is selected. When a0,a1,a2 are all high then q7 is selected.



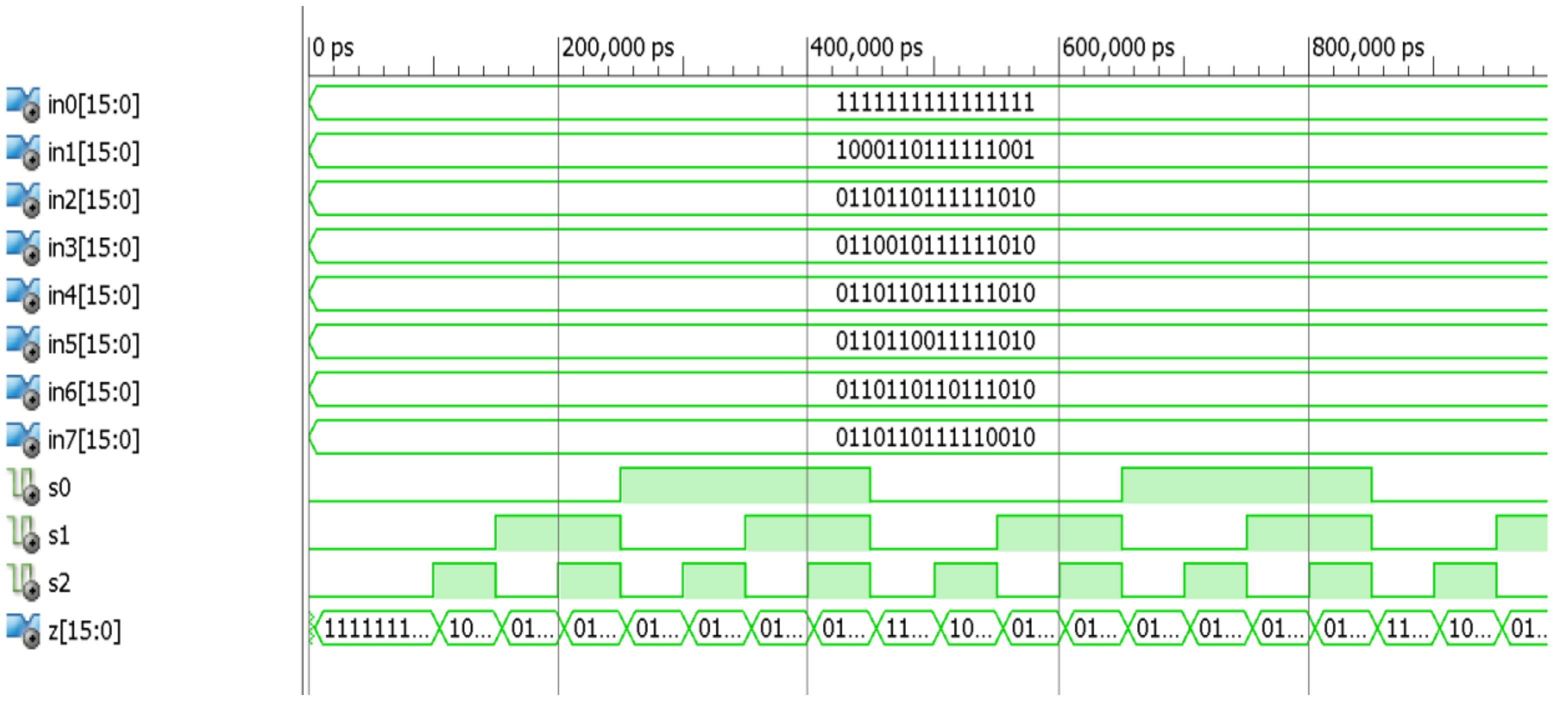
Mux 2 to 1 – 16 bits

The output, z alternates between the data in in0 and in1 as the select bit, s alternates after a fixed period of time.

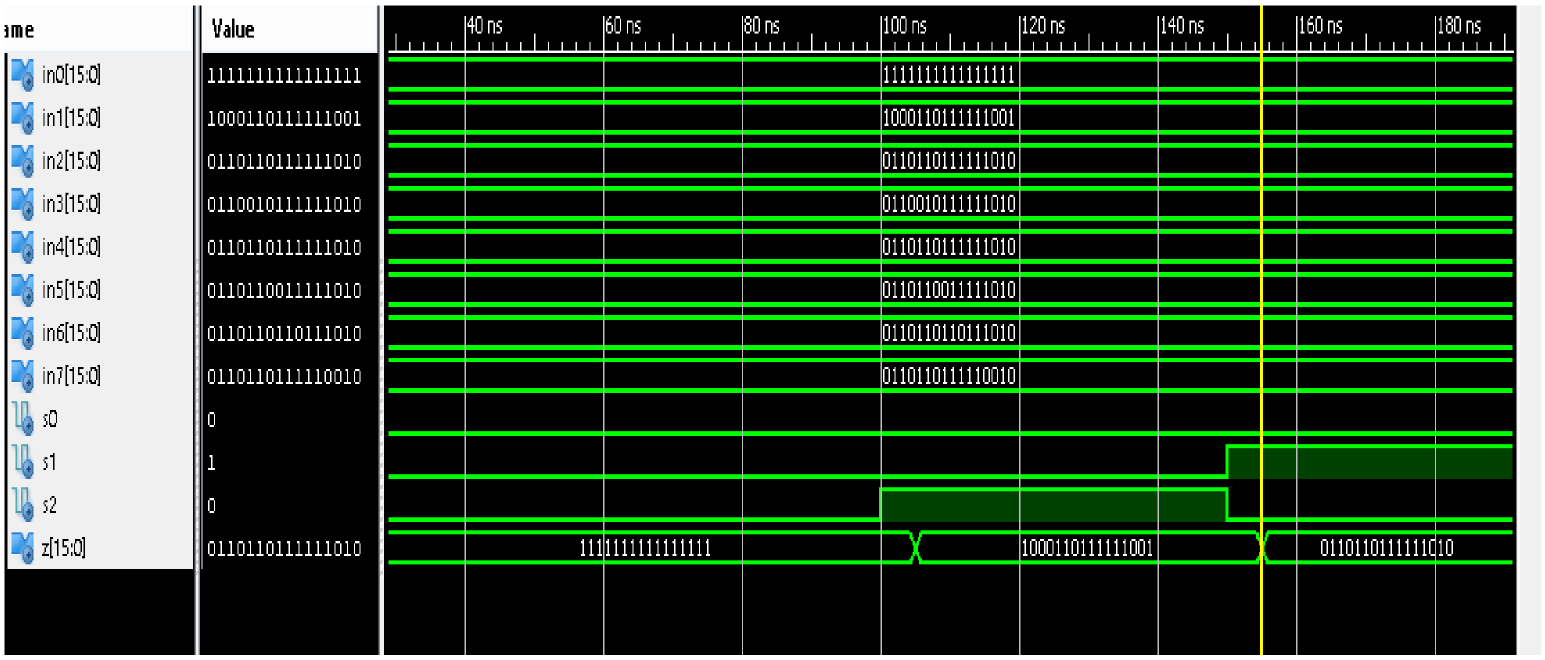


Ammar Qureshi-14318618

Mux 8 to 1 -16 bits



When the select line is 010, in2 contents become the output,z.

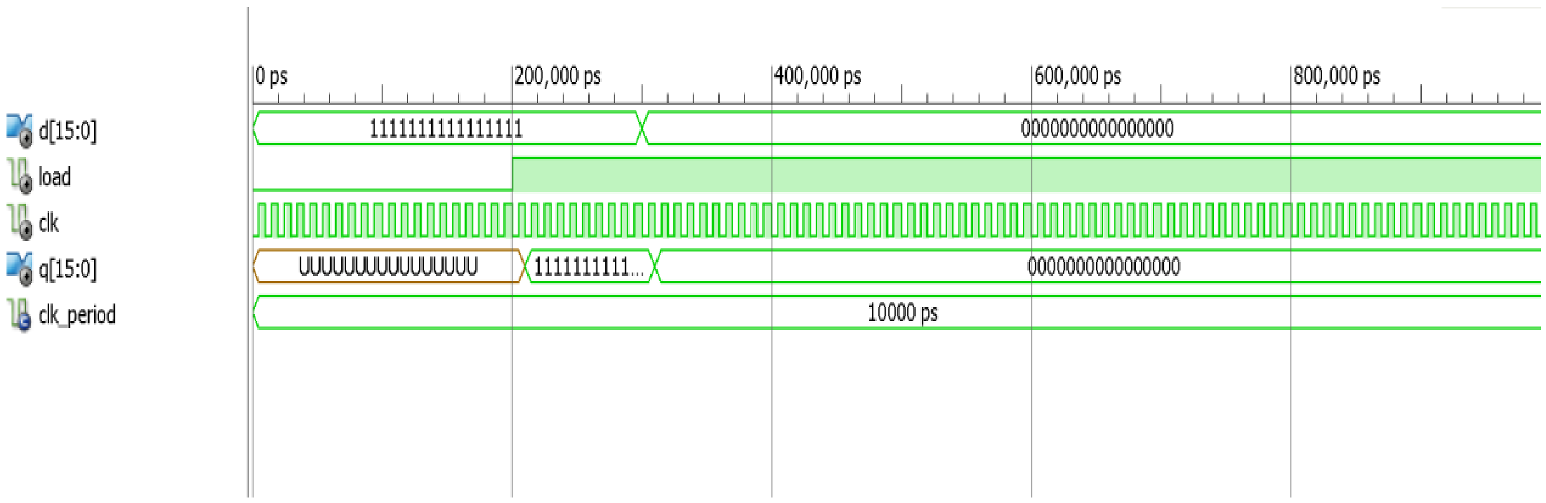


ame	Value
in0[15:0]	1111111111111111
in1[15:0]	1000110111111001
in2[15:0]	0110110111111010
in3[15:0]	0110010111111010
in4[15:0]	0110110111111010
in5[15:0]	0110110011111010
in6[15:0]	0110110110111010
in7[15:0]	0110110111110010
s0	0
s1	1
s2	0
z[15:0]	0110110111111010

Ammar Qureshi-14318618

Register- 16 bit

To test that each register works properly, at around 200 ns content of the register changes from undefined to 11111111111111 after 100 ns contents change to 00000000000000.


0 ps
200,000 ps
400,000 ps
600,000 ps
800,000 ps
d[15:0]
load
clk
q[15:0]
clk_period
UUUUUUUUUUUUUUUU
10000 ps