Register File

Top Level Schematic (decoder_2to4.sch)

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

entity decoder_2to4 is
    Port ( A0 : in std_logic;
           A1 : in std_logic;
           Q0 : out std_logic;
           Q1 : out std_logic;
           Q2 : out std_logic;
           Q3 : out std_logic);
end decoder_2to4;

architecture Behavioral of decoder_2to4 is

begin
Q0<= ((not A0) and (not A1)) after 5 ns;
Q1<= (A0 and (not A1)) after 5 ns;
Q2<= ((not A0) and A1) after 5 ns;
Q3<= (A0 and A1) after 5 ns;
end Behavioral;
```

Register File

Top-level Schematic (S编程辅导)

```
library
use IE                IC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mux2_4bit is
    port ( In0 : in std_logic_vector(3 downto 0);
           In1 : in std_logic_vector(3 downto 0);
           S : in std_logic;
           Z : out std_logic_vector(3 downto 0));
end mux2_4bit;

architecture Behavioral of mux2_4bit is
begin
Z <= In0 after 5 ns when S='0' else
          In1 after 5 ns when S='1' else
               "0000" after 5 ns;
end Behavioral;
```

Register File

Top-level Schematic (S编程辅导)

```
library
use IE                IC_1164.ALL;
use IE                IC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mux4_4bit is
    Port ( In0, In1, In2 In3 : in std_logic_vector(3 downto 0);
           S0, S1 : in std_logic;
           Z : out std_logic_vector(3 downto 0));
end mux4_4bit;

architecture Behavioral of mux4_4bit is
begin
Z <= In0 after 5 ns when S0='0' and S1='0' else
     In1 after 5 ns when S0='1' and S1='0' else
     In2 after 5 ns when S0='0' and S1='1' else
     In3 after 5 ns when S0='1' and S1='1' else
     "0000" after 5 ns;
end Behavioral;
```

Register File

Top-level Schematic (4 bit MUX)

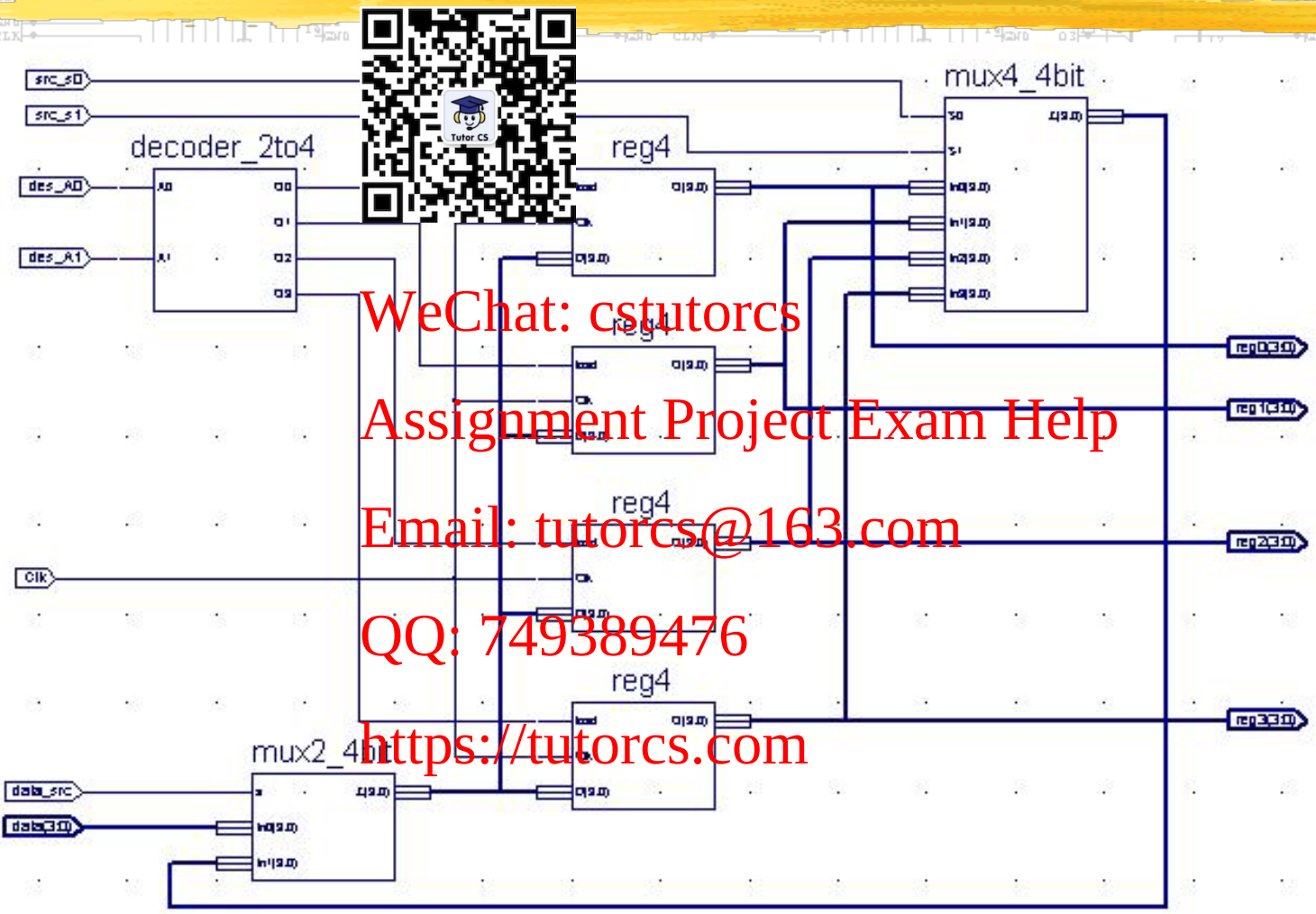
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity reg4 is
    port ( D : in std_logic_vector(3 downto 0);
           load, Clk : in std_logic;
           Q : out std_logic_vector(3 downto 0));
end reg4;

architecture Behavioral of reg4 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;
```

Register File

Top-level Schematic



Register File

Top-level VHDL (entity register_file)

```
librar
use IE                    OGIC_1164.ALL;
use IE                    OGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity register_file is
    Port ( src_s0 : in std_logic;
           src_s1 : in std_logic;
           des_A0 : in std_logic;
           des_A1 : in std_logic;
           Clk : in std_logic;
           data_src : in std_logic;
           data : in std_logic_vector(3 downto 0);
           reg0 : out std_logic_vector(3 downto 0);
           reg1 : out std_logic_vector(3 downto 0);
           reg2 : out std_logic_vector(3 downto 0);
           reg3 : out std_logic_vector(3 downto 0));
end register_file;
```

Register File

Top-level VHDL (reg4)

```
architecture behavioral of register_file is
-- components
    -- inst: Register for register file
    COMPONENT reg4
    PORT(
            D : IN std_logic_vector(3 downto 0);
            load : IN std_logic;
            Clk : IN std_logic;
            Q : OUT std_logic_vector(3 downto 0)
        );
    END COMPONENT;
```

Register File

Top-level VHDL (COMPONENT decoder_2to4/ mux2_4bit)

```
Decoder
NENT decoder_2to4

        A0 : IN std_logic;
        A1 : IN std_logic;
        Q0 : OUT std_logic;
        Q1 : OUT std_logic;
        Q2 : OUT std_logic;
        Q3 : OUT std_logic
        );
END COMPONENT;
-- 2 to 1 line multiplexer
COMPONENT mux2_4bit
PORT(
        In0 : IN std_logic_vector(3 downto 0);
        In1 : IN std_logic_vector(3 downto 0);
        s : IN std_logic;
        Z : OUT std_logic_vector(3 downto 0)
        );
END COMPONENT;
```

Register File

Top-level VHDL (COMPONENT decoder_2to4/ mux2_4bit)

```
Decoder
NENT decoder_2to4

        A0 : IN std_logic;
        A1 : IN std_logic;
        Q0 : OUT std_logic;
        Q1 : OUT std_logic;
        Q2 : OUT std_logic;
        Q3 : OUT std_logic
        );
END COMPONENT;
-- 2 to 1 line multiplexer
COMPONENT mux2_4bit
PORT(
        In0 : IN std_logic_vector(3 downto 0);
        In1 : IN std_logic_vector(3 downto 0);
        s : IN std_logic;
        Z : OUT std_logic_vector(3 downto 0)
        );
END COMPONENT;
```

Register File

Top-level VHDL (COMPONENT 程序代写代做 CS编程辅导 mux4_4bit)

```
ine multiplexer
NENT mux4_4bit
    PORT(
            In0 : IN std_logic_vector(3 downto 0);
            In1 : IN std_logic_vector(3 downto 0);
            In2 : IN std_logic_vector(3 downto 0);
            In3 : IN std_logic_vector(3 downto 0);
            S0 : IN std_logic;
            S1 : IN std_logic;
            Z : OUT std_logic_vector(3 downto 0)
            );
    END COMPONENT;
-- signals
signal load_reg0, load_reg1, load_reg2, load_reg3 : std_logic;
signal reg0_q, reg1_q, reg2_q, reg3_q,
       data_src_mux_out, src_reg : std_logic_vector(3 downto 0);
```

Register File

Top-level VHDL (part missing)

...aps ;-)

```
-- register 0
reg00: reg4 PORT MAP(
        D => data_src_mux_out,
        load => load_reg0,
        clk => clk,
        Q => reg0_q
);
-- register 1
reg01: reg4 PORT MAP(
        D => data_src_mux_out,
        load => load_reg1,
        clk => clk,
        Q => reg1_q
);
```

CS2022 Register File

Top-level VHDL (port maps)

```
        -- register 2
reg4 PORT MAP(
        D => data_src_mux_out,
        load => load_reg2,
        Clk => Clk,
        Q => reg2_q
    );
    -- register 3
    reg03: reg4 PORT MAP(
        D => data_src_mux_out,
        load => load_reg3,
        Clk => Clk,
        Q => reg3_q
    );
    -- Destination register decoder
    des_decoder_2to4: decoder_2to4 PORT MAP(
        A0 => des_A0,
        A1 => des_A1,
        Q0 => load_reg0,
        Q1 => load_reg1,
        Q2 => load_reg2,
        Q3 => load_reg3
    );
```

Register File

Top-level VHDL (part 3)

```
    Data source multiplexer
    _mux2_4bit: mux2_4bit PORT MAP(
        In0 => data,
        In1 => src_reg,
        s => data_src,
        Z => data_src_mux_out
    );
    -- 4 to 1 source register multiplexer
    Inst_mux4_4bit: mux4_4bit PORT MAP(
        In0 => reg0_q,
        In1 => reg1_q,
        In2 => reg2_q,
        In3 => reg3_q,
        S0 => src_s0,
        S1 => src_s1,
        Z => src_reg
    );
    reg0 <= reg0_q;
    reg1 <= reg1_q;
    reg2 <= reg2_q;
    reg3 <= reg3_q;
end Behavioral;
```

Register File

Top Level VHDL (Behavioral)