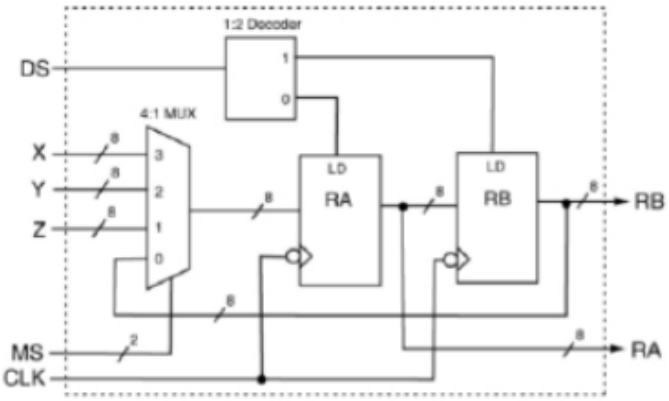
Assignment 1

Name: Nour-aldin Ibrahim Ahmed Elbadawy
ID: 2018-13394

Q7)

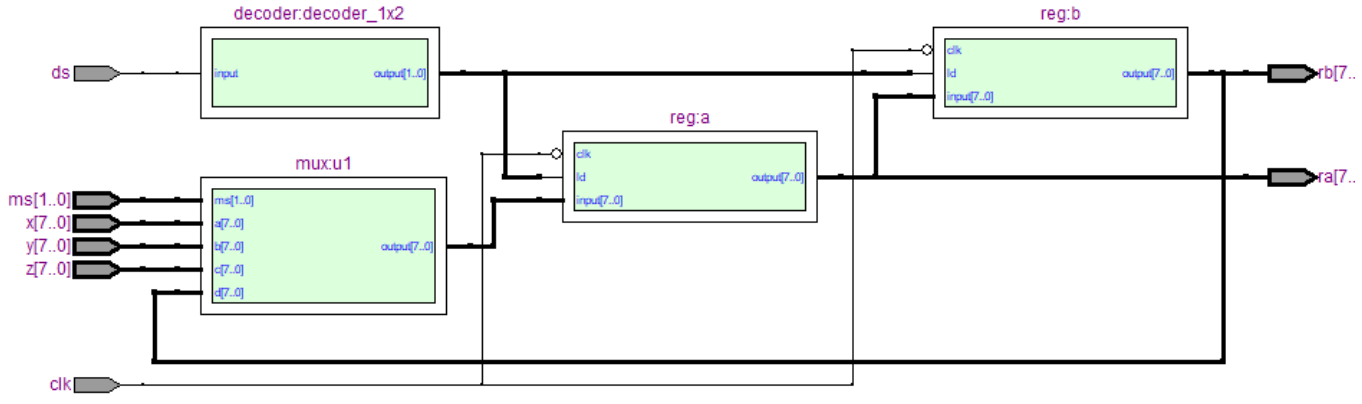
Provide a VHDL model that can be used to implement the following circuit.



Main Code

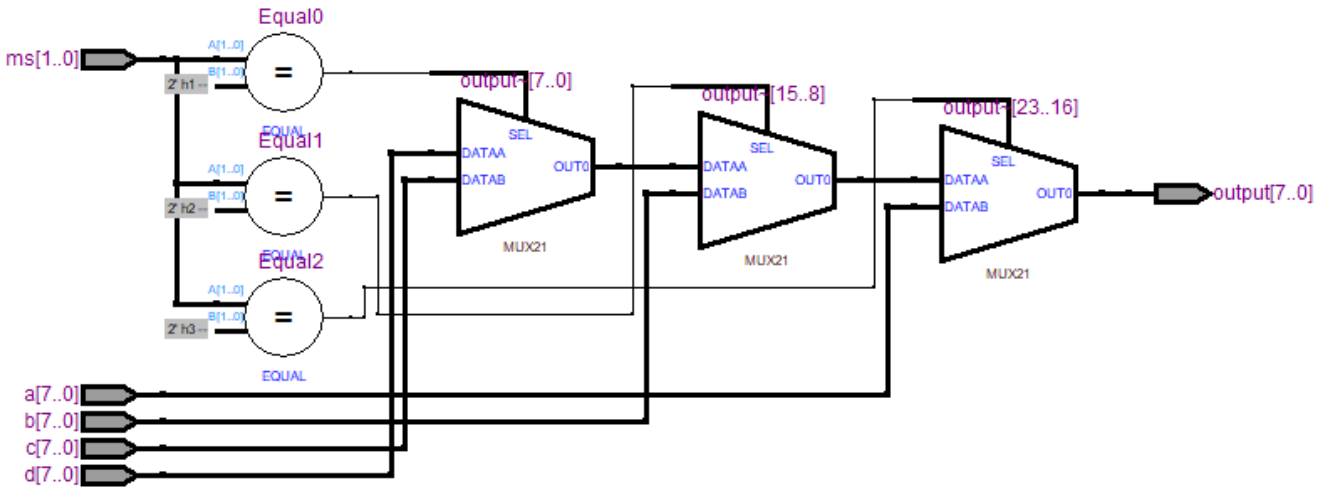
```
library ieee;
use ieee.std_logic_1164.all;

entity Q7 is
        port(
                        ds, clk : in std_logic;
                        x,y,z : in std_logic_vector( 7 downto 0);
                        ms : in std_logic_vector( 1 downto 0);
                        rb, ra : out std_logic_vector( 7 downto 0)
        );
end entity;

architecture behave of Q7 is
signal sig1, sig2, sig3: std_logic_vector( 7 downto 0 );
signal lda, ldb: std_logic;

component reg is
        port(
                        ld, clk : in std_logic;
```

```
                        input : in std_logic_vector( 7 downto 0);
                        output : out std_logic_vector( 7 downto 0)
        );
end component;

component mux is
        port(
                        ms : in std_logic_vector( 1 downto 0);
                        a,b,c,d : in std_logic_vector( 7 downto 0);
                        output : out std_logic_vector( 7 downto 0)
        );
end component;

component decoder is
        port(
                        input : in std_logic;
                        output : out std_logic_vector( 1 downto 0 )
                );
end component;

begin
u1: mux port map( a => x, b => y, c => z, d => sig3, ms => ms, output => sig1);
a: reg port map( ld => lda, clk => not( clk ), input => sig1, output => sig2);
b: reg port map( ld => ldb, clk => not( clk ), input => sig2, output => sig3);
decoder_1x2 : decoder port map ( input => ds, output(0) => lda, output(1) => ldb);
rb <= sig3;
ra <= sig2;
end architecture;
```



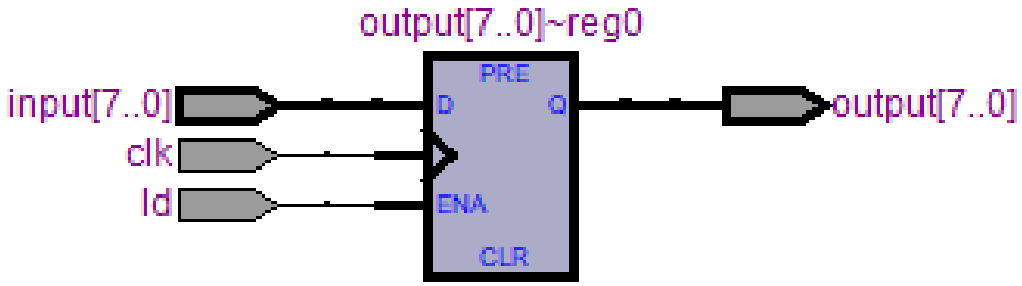
Mux Code

```
library ieee;
use ieee.std_logic_1164.all;

entity mux is
        port(
                        ms : in std_logic_vector( 1 downto 0);
                        a,b,c,d : in std_logic_vector( 7 downto 0);
                        output : out std_logic_vector( 7 downto 0)
        );
end entity;

architecture behave of mux is
begin

        output <= a when( ms = "11") else
                        b when( ms = "10") else
                        c when( ms = "01") else
                        d ;

end architecture;
```



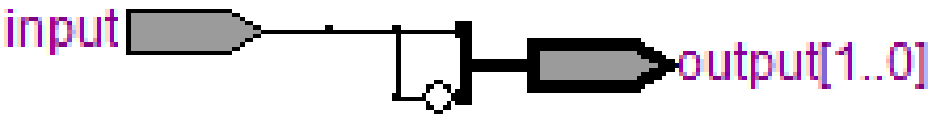
Register Code

```
library ieee;
use ieee.std_logic_1164.all;

entity reg is
        port(
                        ld, clk : in std_logic;
                        input : in std_logic_vector( 7 downto 0);
                        output : out std_logic_vector( 7 downto 0)
        );
end entity;

architecture behave of reg is
begin

        process( ld, clk )
        begin

                if( rising_edge( clk ) ) then
                        if ( ld = '1' ) then
                                output <= input ;
                        end if;
                end if;

        end process;

end architecture;
```



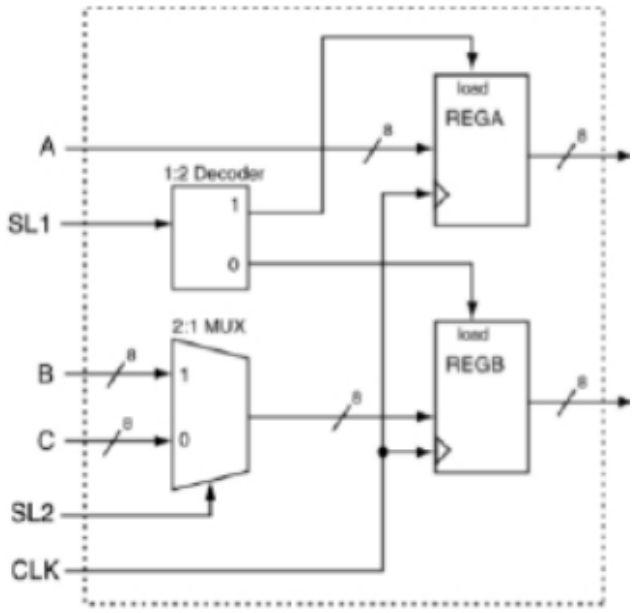
Decoder Code

```
library ieee;
use ieee.std_logic_1164.all;

entity decoder is

        port(
                        input : in std_logic;
                        output : out std_logic_vector( 1 downto 0 )
                );
end entity;

architecture behave of decoder is

begin

With input select
        output <= "01" when '0',
                          "10" when '1';

end behave;
```



Q9)

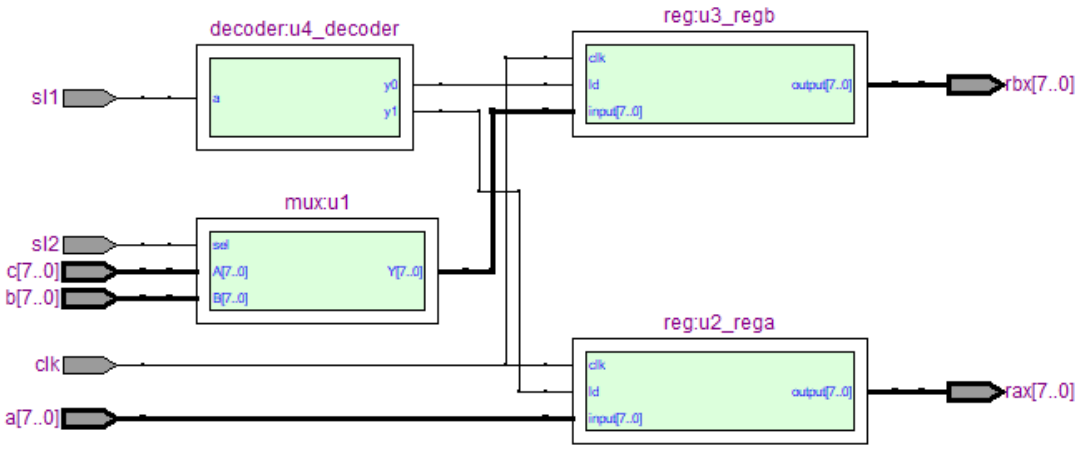
Provide a VHDL model that can be used to implement the following circuit.



Main Code

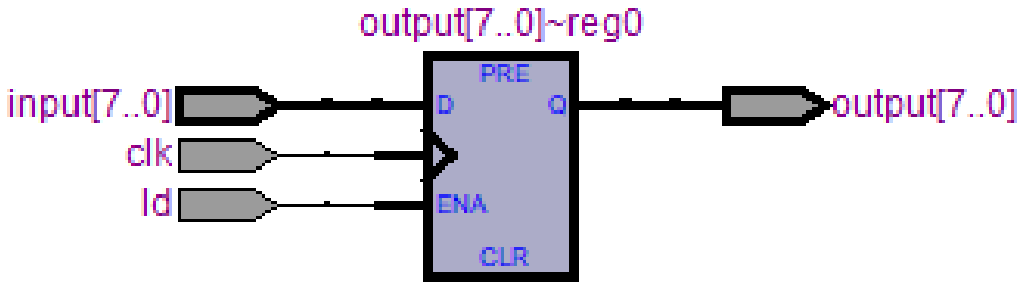
```
library ieee;
use ieee.std_logic_1164.all;

entity Q9 is
        port(
                        a, b, c : in std_logic_vector( 7 downto 0 );
                        sl1, sl2, clk : in std_logic;
                        rax, rbx : out std_logic_vector( 7 downto 0)
                );

end entity;

architecture behave of Q9 is
signal sig: std_logic_vector( 7 downto 0 );
signal lda,ldb : std_logic;

component reg is
port(
                input : in std_logic_vector( 7 downto 0 );
                clk, ld : in std_logic;
                output : out std_logic_vector( 7 downto 0)
);
```

```
end component;

component mux is
port(
                A,B: in std_logic_vector(7 downto 0);
                sel: in std_logic;
                Y: out std_logic_vector(7 downto 0)
);
end component;

component decoder is
port (
                a: in std_logic;
                y0, y1: out std_logic
          );
end component;

begin
u1: mux port map( A => C, B =>B, sel => sl2, y => sig);
u2_rega: reg port map( ld => lda, clk => clk, input => a , output => rax);
u3_regb: reg port map( ld => ldb, clk => clk, input => sig , output => rbx);
u4_decoder: decoder port map ( a => sl1, y0 => ldb, y1 => lda);
end architecture;
```



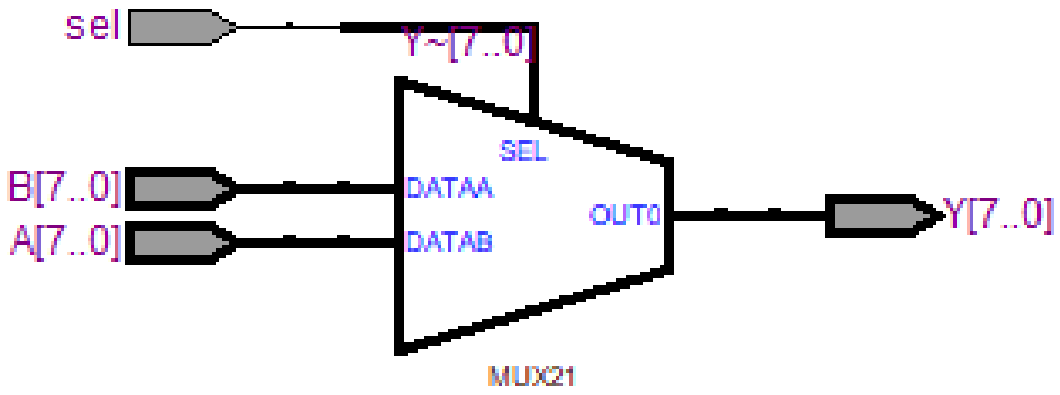
Register Component

```
library ieee;
use ieee.std_logic_1164.all;

entity reg is
        port(
                        input : in std_logic_vector( 7 downto 0 );
                        clk, ld : in std_logic;
                        output : out std_logic_vector( 7 downto 0)
                );

end entity;

architecture behave of reg is

begin

        process( clk, ld )
        begin
                if( rising_edge( clk ) ) then
                        if( ld = '1' ) then
                                output <= input;
                        end if;
                end if;
        end process;
end behave;
```



Mux Component

```
library ieee;
use ieee.std_logic_1164.all;

entity mux is
port(
        A,B: in std_logic_vector(7 downto 0);
        sel: in std_logic;
        Y: out std_logic_vector(7 downto 0) );
        end mux;

architecture behave of mux is
begin
        Y <= A when(sel = '1') else B;
end behave;
```



Decoder Component

```
library ieee;
use ieee.std_logic_1164.all;

entity decoder is
port (
                a: in std_logic;
                y0, y1: out std_logic
          );
end decoder;

architecture behave of decoder is
begin

process( a )
begin
    y0 <= not a;
    y1 <= a;
end process;

end behave;
```