```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
library work;

ENTITY testbench IS
END testbench;

ARCHITECTURE testbench_arch OF testbench IS
   --initialization and declaration of inputs
signal ren0_tb:   std_logic;
signal ren1_tb:   std_logic;
signal ren2_tb:   std_logic;
signal ren3_tb:   std_logic;
signal ren4_tb:   std_logic;
signal ren5_tb:   std_logic;
signal ren6_tb:   std_logic;
signal ren7_tb:   std_logic;
signal ren8_tb:   std_logic;
signal ren9_tb:   std_logic;
signal renA_tb:   std_logic;
signal renB_tb:   std_logic;
signal renC_tb:   std_logic;
signal renD_tb:   std_logic;
signal renE_tb:   std_logic;
signal renF_tb:   std_logic;
signal clr_tb  :  std_logic;
signal clk_tb  :  std_logic;
signal R0out_tb   :  std_logic;
signal R1out_tb   :  std_logic;
signal R2out_tb   :  std_logic;
signal R3out_tb   :  std_logic;
signal R4out_tb   :  std_logic;
signal R5out_tb   :  std_logic;
signal R6out_tb   :  std_logic;
signal R7out_tb   :  std_logic;
signal R8out_tb   :  std_logic;
signal R9out_tb   :  std_logic;
signal R10out_tb  :std_logic;
signal R11out_tb  :std_logic;
signal R12out_tb  :std_logic;
signal R13out_tb  :std_logic;
signal R14out_tb  :std_logic;
signal R15out_tb  :std_logic;
signal Zhighout_tb:  std_logic;
signal Zlowout_tb :std_logic;
signal InPortout_tb: std_logic;
signal HIin_tb    :std_logic;
signal PCin_tb    :std_logic;
signal LOin_tb    :std_logic;
signal Empty_tb   :  STD_LOGIC_VECTOR(31 DOWNTO 0);
signal readin_tb     :std_logic;
signal MDRin_tb   :  std_logic;
signal Mdatain_tb :STD_LOGIC_VECTOR(31 DOWNTO 0);
signal HIoutEn_tb :std_logic;
signal LOoutEn_tb :std_logic;
signal PCoutEn_tb :std_logic;
signal MDRoutEn_tb:  std_logic;
signal CoutEn_tb  :std_logic;
signal reny_tb    :std_logic;
signal operation_tb: STD_LOGIC_VECTOR(3 DOWNTO 0);
signal renz_tb    :std_logic;
signal rout2_tb       :std_logic_vector(31 downto 0);
signal rout3_tb       :std_logic_vector(31 downto 0);
signal rout4_tb       :std_logic_vector(31 downto 0);
signal rout5_tb       :std_logic_vector(31 downto 0);
signal rout6_tb       :std_logic_vector(31 downto 0);
signal rout7_tb       :std_logic_vector(31 downto 0);
signal rout8_tb       :std_logic_vector(31 downto 0);
signal rout9_tb       :std_logic_vector(31 downto 0);
signal routa_tb       :std_logic_vector(31 downto 0);
signal routb_tb       :std_logic_vector(31 downto 0);
signal routc_tb       :std_logic_vector(31 downto 0);
signal routd_tb       :std_logic_vector(31 downto 0);
signal route_tb       :std_logic_vector(31 downto 0);
signal routf_tb       :std_logic_vector(31 downto 0);
signal rout0_tb       :std_logic_vector(31 downto 0);
signal rout1_tb       :std_logic_vector(31 downto 0);
signal busmuxout_tb   :std_logic_vector(31 downto 0);
signal PCout_tb       :std_logic_vector(31 downto 0);
signal Zhigh_tb       :std_logic_vector(31 downto 0);
signal LOout_tb       :std_logic_vector(31 downto 0);
signal HIout_tb       :std_logic_vector(31 downto 0);
signal MDRout_tb   :std_logic_vector(31 downto 0);
signal routy_tb :     STD_LOGIC_VECTOR(31 DOWNTO 0);
signal Zlow_tb     :std_logic_vector(31 downto 0);


TYPE  State IS(default, Reg_load1, Reg_load2, Reg_load3, T0, T1, T2, T3, T4, T5);
SIGNAL Present_state: State := default;

component ELEC374
      PORT
   (
      ren0      :   IN STD_LOGIC;
      ren1      :   IN STD_LOGIC;
      ren2      :   IN STD_LOGIC;
      ren3      :   IN STD_LOGIC;
      ren4      :   IN STD_LOGIC;
      ren5      :   IN STD_LOGIC;
      ren6      :   IN STD_LOGIC;
      ren7      :   IN STD_LOGIC;
      ren8      :   IN STD_LOGIC;
      ren9      :   IN STD_LOGIC;
      renA      :   IN STD_LOGIC;
      renB      :   IN STD_LOGIC;
      renC      :   IN STD_LOGIC;
      renD      :   IN STD_LOGIC;
      renE      :   IN STD_LOGIC;
      renF      :   IN STD_LOGIC;
      clr       :   IN STD_LOGIC;
      clk       :   IN STD_LOGIC;
      R0out     :   IN STD_LOGIC;
      R1out     :   IN STD_LOGIC;
      R2out     :   IN STD_LOGIC;
      R3out     :   IN STD_LOGIC;
      R4out     :   IN STD_LOGIC;
      R5out     :   IN STD_LOGIC;
      R6out     :   IN STD_LOGIC;
      R7out     :   IN STD_LOGIC;
      R8out     :   IN STD_LOGIC;
      R9out     :   IN STD_LOGIC;
      R10out       :   IN STD_LOGIC;
      R11out       :   IN STD_LOGIC;
      R12out      :   IN STD_LOGIC;
      R13out      :   IN STD_LOGIC;
      R14out      :   IN STD_LOGIC;
      R15out      :   IN STD_LOGIC;
      Zhighout    :   IN STD_LOGIC;
      Zlowout     :   IN STD_LOGIC;
      InPortout      :   IN STD_LOGIC;
      HIin     :   IN STD_LOGIC;
      PCin     :   IN STD_LOGIC;

      LOin     :   IN STD_LOGIC;
      Empty    :   IN STD_LOGIC_VECTOR(31 DOWNTO 0);
      ReadIn :  IN  STD_LOGIC;
      MDRin    :   IN STD_LOGIC;
      Mdatain     :   IN STD_LOGIC_VECTOR(31 DOWNTO 0);
      HIoutEn     :   IN STD_LOGIC;
      LOoutEn     :   IN STD_LOGIC;
      PCoutEn     :   IN STD_LOGIC;
      MDRoutEn    :   IN STD_LOGIC;
      CoutEn      :   IN STD_LOGIC;
      reny     :   IN STD_LOGIC;
      operation:   IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      renz     :   IN STD_LOGIC;
      rout2    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout3    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout4    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout5    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout6    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout7    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout8    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout9    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      routa    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      routb    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      routc    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      routd    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      route    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      routf    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout0    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      rout1    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      busmuxout      :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      PCout    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      Zhigh    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      Zlow     :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      LOout    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      HIout    :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      MDRout      :   INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      routy    : INOUT STD_LOGIC_VECTOR(31 DOWNTO 0)

   );
end component ELEC374;

BEGIN

DUT: ELEC374 PORT MAP(
ren0        =>ren0_tb,
ren1        =>ren1_tb,
ren2        =>ren2_tb,
ren3        =>ren3_tb,
ren4        =>ren4_tb,
ren5        =>ren5_tb,
ren6        =>ren6_tb,
ren7        =>ren7_tb,
ren8        =>ren8_tb,
ren9        =>ren9_tb,
renA        =>renA_tb,
renB        =>renB_tb,
renC        =>renC_tb,
renD        =>renD_tb,
renE        =>renE_tb,
renF        =>renF_tb,
clr         =>clr_tb,
clk         =>clk_tb,
R0out       =>R0out_tb,
R1out       =>R1out_tb,
R2out       =>R2out_tb,
R3out       =>R3out_tb,
R4out       =>R4out_tb,
R5out       =>R5out_tb,
R6out       =>R6out_tb,
R7out       =>R7out_tb,
R8out       =>R8out_tb,
R9out       =>R9out_tb,
R10out      =>R10out_tb,
R11out      =>R11out_tb,
R12out      =>R12out_tb,
R13out      =>R13out_tb,
R14out      =>R14out_tb,
R15out      =>R15out_tb,
Zhighout =>Zhighout_tb,
Zlowout     =>Zlowout_tb,
InPortout   =>InPortout_tb,
HIin     =>HIin_tb,
PCin     =>PCin_tb,
LOin     =>LOin_tb,
Empty    =>Empty_tb,
ReadIn      =>readin_tb,
MDRin    =>MDRin_tb,
Mdatain     =>Mdatain_tb,
HIoutEn     =>HIoutEn_tb,
LOoutEn     =>LOoutEn_tb,
PCoutEn     =>PCoutEn_tb,
MDRoutEn =>MDRoutEn_tb,
CoutEn      =>CoutEn_tb,
reny     =>reny_tb,
operation   =>operation_tb,
renz     =>renz_tb,
rout2    =>rout2_tb,
rout3    =>rout3_tb,
rout4    =>rout4_tb,
rout5    =>rout5_tb,
rout6    =>rout6_tb,
rout7    =>rout7_tb,
rout8    =>rout8_tb,
rout9    =>rout9_tb,
routa    =>routa_tb,
routb    =>routb_tb,
routc    =>routc_tb,
routd    =>routd_tb,
route    =>route_tb,
routf    =>routf_tb,
rout0    =>rout0_tb,
rout1    =>rout1_tb,
busmuxout   =>busmuxout_tb,
PCout    =>PCout_tb,
Zhigh    =>Zhigh_tb,
Zlow     =>Zlow_tb,
LOout    =>LOout_tb,
HIout    =>HIout_tb,
MDRout      =>MDRout_tb,
routy => routy_tb
   );

clk_process : process
   begin
         clk_tb<='0','1' after 10 ns;
         wait for 20 ns;
end process clk_process;

process (clk_tb)

begin
   if (clk_tb'event and clk_tb ='1') then
      case Present_state is
         when Default =>
               Present_state<=Reg_load1;
         when reg_load1 =>
               present_state<=Reg_load2;
         when reg_load2 =>
               present_state<=Reg_load3;
         when reg_load3 =>
               present_state<=T0;
         when T0 =>
               present_state<=T1;
         when T1 =>
               present_state<=T2;
         when T2 =>
               present_state<=T3;
         when T3 =>
               present_state<=T4;
         when T4 =>
               present_state<=T5;
         when others =>
      end case;
   end if;
end process;

Process(present_state)
begin
      CASE present_state IS
            WHEN default =>
               PCin_tb<='0'; Zlowout_tb<='0'; mdroutEn_tb<='0';
               R1Out_tb<='0';R2Out_tb<='0';R3Out_tb<='0';
               PCin_tb <= '0'; MDRin_tb <= '0';
               ren1_tb<='0'; readin_tb<='0';
               clr_tb <= '1', '0' after 4 ns;
               renz_tb <= '0';
               ren0_tb <= '0';
               ren1_tb <= '0';
               ren2_tb <= '0';
               ren3_tb <= '0';
               ren4_tb <= '0';
               ren5_tb <= '0';
               ren6_tb <= '0';
               ren7_tb <= '0';
               ren8_tb <= '0';
               ren9_tb <= '0';
```

```
                 renA_tb <= '0';
                 renB_tb <= '0';
                 renC_tb <= '0';
                 renD_tb <= '0';
                 renE_tb <= '0';
                 renF_tb <= '0';
                 R0out_tb <= '0';
                 R1out_tb <= '0';
                 R2out_tb <= '0';
                 R3out_tb <= '0';
                 R4out_tb <= '0';
                 R5out_tb <= '0';
                 R6out_tb <= '0';
                 R7out_tb <= '0';
                 R8out_tb <= '0';
                 R9out_tb <= '0';
                 R10out_tb <= '0';
                 R11out_tb <= '0';
                 R12out_tb <= '0';
                 R13out_tb <= '0';
                 R14out_tb <= '0';
                 R15out_tb <= '0';
                 Zhighout_tb <= '0';
                 InPortout_tb <= '0';
                 HIin_tb <= '0';

                 Loin_tb <='0';
                 HIoutEn_tb <= '0';
                 LOoutEn_tb<='0';
                 PcoutEn_tb <= '0';
                 CoutEn_tb<='0';
                 reny_tb <= '0';
                 operation_tb <= "0000";
                 Mdatain_tb<=x"00000012";

                 --routy_tb <= x"00000000";
                 empty_tb <= x"00000000";
                 operation_tb<="0000";



              when reg_load1 =>
                 mdatain_tb<= x"00006666";
                 readin_tb<='1','0' after 10 ns;
                 mdrin_tb<='1','0'after 10 ns;
                 mdroutEn_tb<='1', '0' after 10 ns ;

                 ren1_tb<='1', '0' after 10 ns ;

              when reg_load2 =>
                 mdatain_tb <= x"00000015";
                 readin_tb<='1','0' after 10 ns;
                 mdrin_tb<='1','0'after 10 ns;
                 mdroutEn_tb<='1', '0' after 10 ns ;


                 ren2_tb<='1', '0' after 10 ns ;

              when reg_load3 =>
                 mdatain_tb <= x"00000002";
                 readin_tb<='1','0' after 10 ns;
                 mdrin_tb<='1','0'after 10 ns;
```

```vhdl
               mdroutEn_tb<='1', '0' after 10 ns ;
               ren3_tb<='1', '0' after 10 ns ;

            when T0 =>
               PCoutEn_tb<='1', '0' after 10 ns ;
               renz_tb<='1', '0' after 10 ns ;


            when T1 =>
               Zlowout_tb<='1', '0' after 10 ns ; PCin_tb<='1', '0' after 10 ns ;
               readin_tb<='1', '0' after 10 ns ; MDRin_tb<='1', '0' after 10 ns ;
               --Mdatain_tb<=x"294c0000";

            when T2 =>

               MDRoutEn_tb<='1', '0' after 10 ns ;

            when T3 =>
               --
               r2out_tb<='1', '0' after 10 ns;
               reny_tb<='1', '0' after 10 ns ; --readin_tb<='1', '0' after 10 ns ;MDRin_tb<='1', '0' after 10 ns ;
            when T4 =>
               r3out_tb<='1', '0' after 10 ns ; renz_tb<='1', '0' after 10 ns ;
               operation_tb<="0100";
            when T5 =>
               zlowout_tb<='1', '0' after 10 ns ; ren1_tb<='1', '0' after 10 ns ; --operation_tb<="1111"; renz_tb<='0';
            when others =>

         end case;
      end process;



end architecture testbench_arch;

```