```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY datapath_tb IS
END datapath_tb;
ARCHITECTURE behavior OF datapath_tb IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT datapath
         control : in STD_LOGIC_VECTOR(16 downto 0);
         clk : IN std_logic;
         const_in : IN std_logic_vector(15 downto 0);
         data_in : IN std_logic_vector(15 downto 0);
        V : OUT std_logic;
        C : OUT std_logic;
        N : OUT std_logic;
         Z : OUT std_logic;
         address_out : OUT std_logic_vector(15 downto 0);
         data_out : OUT std_logic_vector(15 downto 0)
   END COMPONENT;
   --Inputs
   signal control : std_logic_vector(16 downto 0) := (others => '0');
   signal clk : std_logic := '0';
   signal const in : std logic vector(15 downto 0) := (others => '0');
   signal data_in : std_logic_vector(15 downto 0) := (others => '0');
        --Outputs
   signal V : std_logic;
   signal C : std_logic;
   signal N : std_logic;
   signal Z : std logic;
   signal address_out : std_logic_vector(15 downto 0);
   signal data_out : std_logic_vector(15 downto 0);
   -- Clock period definitions
  constant clk_period : time := 10 ns;
BEGIN
        -- Instantiate the Unit Under Test (UUT)
  uut: datapath PORT MAP (
                         control => control,
          clk => clk,
          const_in => const_in,
          data_in => data_in,
          V => V,
          C => C,
          N \Rightarrow N,
          Z => Z,
          address_out => address_out,
          data_out => data_out
       );
   -- 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
      -- hold reset state for 100 ns.
                control <= "00000000100000001";</pre>
                const in <= "0000000000000000";</pre>
                wait for 100 ns;
```

```
control <= "00000000100000011";</pre>
                 data_in <= "0000111100000000";
                 wait for 20 ns;
                 control <= "00100000100000011";</pre>
                 data_in <= "11110000111111111";
                 wait for 20 ns;
                 control <= "00100000100000010";</pre>
                 wait for 20 ns;
                 control <= "01000000100000000";</pre>
                 wait for 20 ns;
                 control <= "01000000100000001";</pre>
                 wait for 20 ns;
                 control <= "01000000100000101";
                 wait for 20 ns;
                 control <= "01000000100001001";</pre>
                 wait for 20 ns;
                 control <= "01000000100001101";</pre>
                 wait for 20 ns;
                 control <= "01000000100010001";</pre>
                 wait for 20 ns;
                 control <= "01000000100010101";</pre>
                 wait for 20 ns;
                 control <= "01000000100011001";</pre>
                 wait for 20 ns;
                 control <= "01000000100011101";</pre>
                 wait for 20 ns;
                 control <= "01000000100100001";
                 wait for 20 ns;
                 control <= "01000000100101001";</pre>
                 wait for 20 ns;
                 control <= "01000000100110001";</pre>
                 wait for 20 ns;
                 control <= "01000000100111001";</pre>
                 wait for 20 ns;
                 control <= "01000000101000001";</pre>
                 wait for 20 ns;
                 control <= "01000000101010001";</pre>
                 wait for 20 ns;
                 control <= "01000000101100001";</pre>
      wait;
   end process;
END;
```