Design and implement 1:8 DEMUX circuit in VHDL using Behavioural Architecture:

Truth Table:

DIN	SEL (S2 S1 S0)	D0	D1	D2	D3	D4	D5	D6	D7
0	000	0	0	0	0	0	0	0	0
1	000	1	0	0	0	0	0	0	0
1	001	0	1	0	0	0	0	0	0
1	010	0	0	1	0	0	0	0	0
1	011	0	0	0	1	0	0	0	0
1	100	0	0	0	0	1	0	0	0
1	101	0	0	0	0	0	1	0	0
1	110	0	0	0	0	0	0	1	0
1	111	0	0	0	0	0	0	0	1

VHDL Source Code:

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

entity demux_1_to_8_beh is
    Port (
        din  : in  STD_LOGIC;
        sel  : in  STD_LOGIC_VECTOR(2 downto 0);
        dout : out STD_LOGIC_VECTOR(7 downto 0)
    );
end demux_1_to_8_beh;

architecture Behavioral of demux_1_to_8_beh is
begin
    process(din, sel)
    begin
        dout <= (others => '0');
        if din = '1' then
            case sel is
                when "000" => dout(0) <= '1';
                when "001" => dout(1) <= '1';
                when "010" => dout(2) <= '1';
```

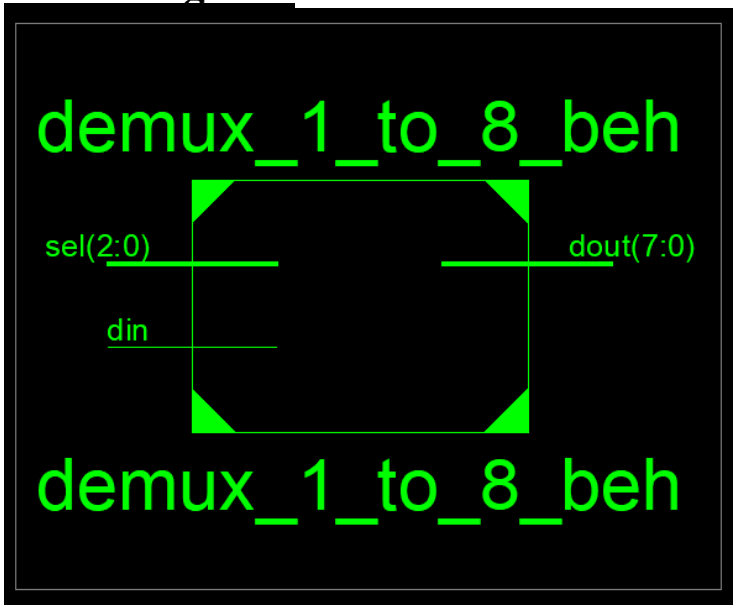
```
            when "011" => dout(3) <= '1';
            when "100" => dout(4) <= '1';
            when "101" => dout(5) <= '1';
            when "110" => dout(6) <= '1';
            when "111" => dout(7) <= '1';
            when others => dout <= (others => '0');
        end case;
    end if;
  end process;
end Behavioral;
```

VHDL Test Bench Code:

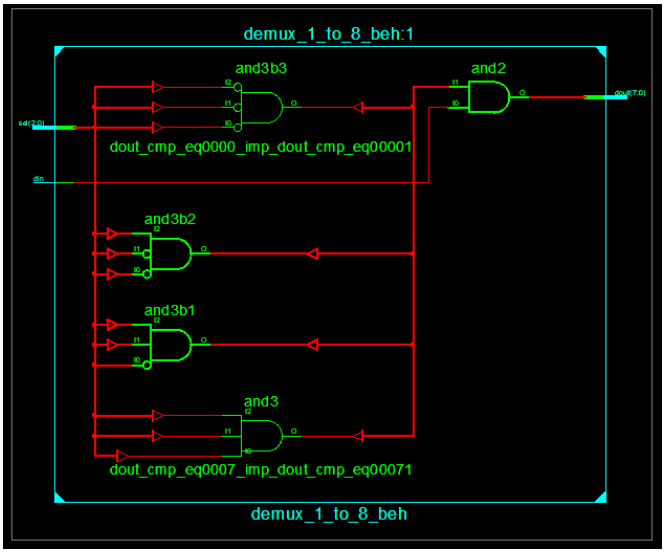
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity tb_demux_truth_table is
end tb_demux_truth_table;

architecture Test of tb_demux_truth_table is
  signal din  : STD_LOGIC := '0';
  signal sel  : STD_LOGIC_VECTOR(2 downto 0) := (others => '0');
  signal dout : STD_LOGIC_VECTOR(7 downto 0);

  -- Use this to switch between behavior/dataflow
  component demux_1_to_8_beh
    Port (
      din  : in  STD_LOGIC;
      sel  : in  STD_LOGIC_VECTOR(2 downto 0);
      dout : out STD_LOGIC_VECTOR(7 downto 0)
    );
  end component;

begin
  uut: demux_1_to_8_beh
    port map (
      din  => din,
      sel  => sel,
      dout => dout
    );

  stimulus: process
  begin
    -- Test: DIN = 1 with SEL from 000 to 111
    din <= '1';
    for i in 0 to 7 loop
      sel <= std_logic_vector(to_unsigned(i, 3));
```

```
        wait for 10 ns;
    end loop;

    -- Test: DIN = 0 with SEL from 000 to 111
    din <= '0';
    for i in 0 to 7 loop
        sel <= std_logic_vector(to_unsigned(i, 3));
        wait for 10 ns;
    end loop;

    wait;
  end process;
end Test;
```

Block Diagram:



Circuit Diagram:



Test Bench Waveform: