Practical 5 Key Pad

Code

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

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity keypad12 is
   Port ( rows : in  STD_LOGIC_VECTOR (3 downto 0);
          columns : in  STD_LOGIC_VECTOR (3 downto 0);
          dout : out  STD_LOGIC_VECTOR (7 downto 0));
end keypad12;

architecture Behavioral of keypad12 is
begin
process(rows, columns)
begin
if(rows="0001")then
if (columns="0001")then
dout<="00111111";
end if;
if (columns="0010")then
dout<="00000110";
end if;
if (columns="0100")then
dout<="01011011";
end if;
```

```
if (columns="1000")then
dout<="01001111";
end if;
end if;
if(rows="0010")then
if (columns="0001")then
dout<="01100110";
end if;
if (columns="0010")then
dout<="01101101";
end if;
if (columns="0100")then
dout<="01111101";
end if;
if (columns="1000")then
dout<="00000111";
end if;
end if;
if(rows="0100")then
if (columns="0001")then
dout<="01111111";
end if;
if (columns="0010")then
dout<="01100111";
end if;
if (columns="0100")then
dout<="01110111";
end if;
if (columns="1000")then
dout<="01111100";
end if;
end if;
```

```
if(rows="1000")then
if (columns="0001")then
dout<="00111001";
end if;
if (columns="0010")then
dout<="01011110";
end if;
if (columns="0100")then
 dout<="01111001";
 end if;
if (columns="1000")then
   dout<="01110001";
end if;
end if;
end process;
end Behavioral;
```

Test Bench Key Pad

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;
ENTITY keypad1234_vhd IS
END keypad1234_vhd;
ARCHITECTURE behavior OF keypad1234_vhd IS
        -- Component Declaration for the Unit Under Test (UUT)
        COMPONENT keypad12
        PORT(
                rows : IN std_logic_vector(3 downto 0);
                columns : IN std_logic_vector(3 downto 0);
                dout : OUT std_logic_vector(7 downto 0)
                );
        END COMPONENT;
```

```
        --Inputs
        SIGNAL rows :  std_logic_vector(3 downto 0) := (others=>'0');
        SIGNAL columns :  std_logic_vector(3 downto 0) := (others=>'0');
        --Outputs
        SIGNAL dout :  std_logic_vector(7 downto 0);
BEGIN
        -- Instantiate the Unit Under Test (UUT)
        uut: keypad12 PORT MAP(
                rows => rows,
                columns => columns,
                dout => dout
        );
        tb : PROCESS
        BEGIN
                -- Wait 100 ns for global reset to finish
                wait for 100 ns;
                rows <= "0001";
 columns <= "0001";
     wait for 100 ns;
 rows <= "0001";
 columns <= "0010";
     wait for 100 ns;
 rows <= "0001";
 columns <= "0100";
     wait for 100 ns;
 rows <= "0001";
 columns <= "1000";
     wait for 100 ns;
 -- for second row =>
 rows <= "0010";
 columns <= "0001";
     wait for 100 ns;
  rows <= "0010";
 columns <= "0010";
```

```
    wait for 100 ns;
rows <= "0010";
columns <= "0100";
    wait for 100 ns;
rows <= "0010";
columns <= "1000";
    wait for 100 ns;
-- for third row =>
rows <= "0100";
columns <= "0001";
    wait for 100 ns;
                rows <= "0100";
columns <= "0010";
    wait for 100 ns;
rows <= "0100";
columns <= "0100";
   wait for 100 ns;
rows <= "0100";
columns <= "1000";
    wait for 100 ns;
-- for fourth row =>
rows <= "1000";
columns <= "0001";
    wait for 100 ns;

rows <= "1000";
columns <= "0010";
    wait for 100 ns;
rows <= "1000";
columns <= "0100";
    wait for 100 ns;
rows <= "1000";
columns <= "1000";
    wait for 100 ns;
```

```
    --wait for <clock>_period*10;
wait; -- will wait forever
END PROCESS;
END;
```