XILINE VIVADO Simulation:

Boolean Expression Design:

$(A, B, C, D)=\sum m(2, 3, 4, 5, 6, 8, 10, 12, 13, 15)$

		1	1
1	1		1
1	1	1	
1			1

X=($C'B+AC'D'+ABD+A'CD'+A'B'C+B'CD'$)

Explanation:

In this task we implement the K_map actually k_map is used to convert the truth table to Boolean expression as shown above, this work on 1,2 4, 8 pairs , Here we deal with SOP mean sum of product to make it simple we take the 1 in K_map if it was POS (product of sum) we take the 0 in K_Map.

Truth Table:

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Explanation:

The above table called truth table , it play a leading role in design of logic circuit, without this we are not able to create logic circuit in this task we have given some parameters, where it was locate we take here 1 and other will be zero.

VHDL Code:

Design.vhd:

```
-- Simple design
library IEEE;
use IEEE.std_logic_1164.all;

entity combine is
port(
  a: in std_logic;
  b: in std_logic;
  c: in std_logic;
  d: in std_logic;
  q: out std_logic);
end or_gate;

architecture Behavioral of combine is
begin
process(a, b,c,d)
begin
if(a='0' and b='0' and c='0' and d='0') then
q<='0';
else if(a='0' and b='0' and c='0' and d='1') then
q<='0';
else if(a='0' and b='0' and c='1' and d='0') then
q<='1';
else if(a='0' and b='0' and c='1' and d='1') then
q<='1';
else if(a='0' and b='1' and c='0' and d='0') then
q<='1';
else if(a='0' and b='1' and c='0' and d='1') then
q<='1';
else if(a='0' and b='1' and c='1' and d='0') then
q<='1';
else if(a='0' and b='1' and c='1' and d='1') then
q<='0';
else if(a='1' and b='0' and c='0' and d='0') then
q<='1';
else if(a='1' and b='0' and c='0' and d='1') then
q<='0';
else if(a='1' and b='0' and c='1' and d='0') then
q<='1';
else if(a='1' and b='0' and c='1' and d='1') then
q<='0';
else if(a='1' and b='1' and c='0' and d='0') then
q<='1';
else if(a='1' and b='1' and c='0' and d='1') then
q<='1';
else if(a='1' and b='1' and c='1' and d='0') then
q<='0';
else if(a='1' and b='1' and c='1' and d='1') then
q<='1';
end if;
```

```
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;
end if;

end process;
end Behavioral;
```

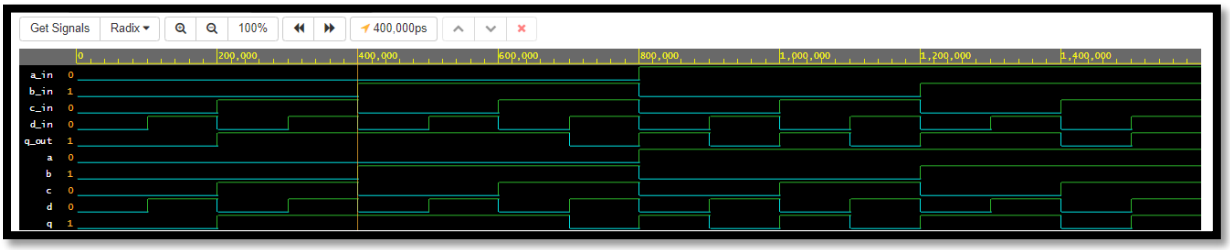
TestBench.vhd

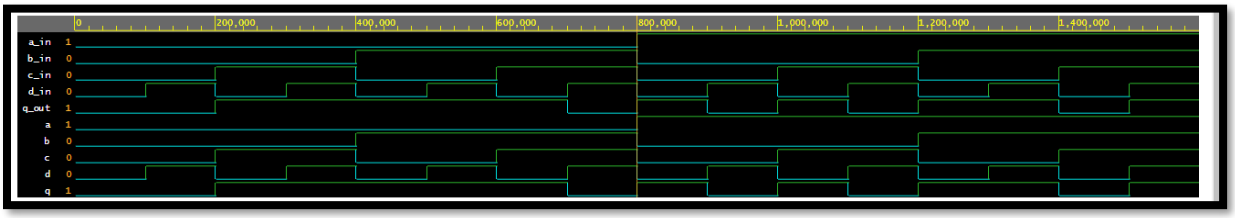
```
library IEEE;
use IEEE.std_logic_1164.all;

entity testbench is
-- empty
end testbench;

architecture tb of testbench is

-- DUT component
component combine is
port(
  a: in std_logic;
  b: in std_logic;
  c: in std_logic;
  d: in std_logic;
  q: out std_logic);
end component;

signal a_in, b_in,c_in,d_in, q_out: std_logic;

begin

  -- Connect DUT
  DUT: combine port map(a_in, b_in,c_in,d_in, q_out);

  process
  begin
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    d_in <= '0';
    wait for 100 ns;
```

```
a_in <= '0';
b_in <= '0';
c_in <= '0';
d_in <= '1';
wait for 100 ns;

a_in <= '0';
b_in <= '0';
c_in <= '1';
d_in <= '0';
wait for 100 ns;

a_in <= '0';
b_in <= '0';
c_in <= '1';
d_in <= '1';
wait for 100 ns;

a_in <= '0';
b_in <= '1';
c_in <= '0';
d_in <= '0';
wait for 100 ns;

a_in <= '0';
b_in <= '1';
c_in <= '0';
d_in <= '1';
wait for 100 ns;

a_in <= '0';
b_in <= '1';
c_in <= '1';
d_in <= '0';
wait for 100 ns;

a_in <= '0';
b_in <= '1';
c_in <= '1';
d_in <= '1';
wait for 100 ns;

a_in <= '1';
b_in <= '0';
c_in <= '0';
d_in <= '0';
wait for 100 ns;

a_in <= '1';
b_in <= '0';
c_in <= '0';
d_in <= '1';
wait for 100 ns;

a_in <= '1';
```

```
    b_in <= '0';
    c_in <= '1';
    d_in <= '0';
    wait for 100 ns;

    a_in <= '1';
    b_in <= '0';
    c_in <= '1';
    d_in <= '1';
    wait for 100 ns;

    a_in <= '1';
    b_in <= '1';
    c_in <= '0';
    d_in <= '0';
    wait for 100 ns;

    a_in <= '1';
    b_in <= '1';
    c_in <= '0';
    d_in <= '1';
    wait for 100 ns;


    a_in <= '1';
    b_in <= '1';
    c_in <= '1';
    d_in <= '0';
    wait for 100 ns;


    a_in <= '1';
    b_in <= '1';
    c_in <= '1';
    d_in <= '1';
    wait for 100 ns;

    -- Clear inputs
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    d_in <= '0';


    wait;
  end process;
end tb;
```

Waveform:


Get Signals
Radix
100%
400,000ps
0
200,000
400,000
600,000
800,000
1,000,000
1,200,000
1,400,000
a_in 0
b_in 1
c_in 0
d_in 0
q_out 1
a 0
b 1
c 0
d 0
q 1


0
200,000
400,000
600,000
800,000
1,000,000
1,200,000
1,400,000
a_in 1
b_in 0
c_in 0
d_in 0
q_out 1
a 1
b 0
c 0
d 0
q 1