Practical Manual
Lab: DSD
(EE-330)

Electronics and Communication Engg. (ECE/CSE)

RAO PAHALD SINGH GROUP OF INSTITUTIONS
BALANA(MOHINDER GARH)123029
EXPERIMENT 1

Aim:-
    Write the vhdl code and simulate it for the following gates.
    i) Two i/p AND Gates.
    ii) Two i/p OR Gates.
    iii) Two i/p NAND Gates
    iv) Two i/p NOR Gates.
    v) Two i/p EX-OR Gates.
    vi) NOT Gates.

SPECIFICATION OF APPARATUS USED:-
    Mentor graphics FPGA advantage software modelsim simulation tool.

Program:-
    i) Behavior Model if two i/p AND Gate.

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity and2 is
Port (a, b: in bit; z: out bit);
End and2;

Architecture beh of and2 is
Begin process (a, b)
Begin
If (a='0' and b='0') then
Z<='0';
ElsIf (a='0' and b='1') then
Z<='0';
ElsIf (a='1' and b='0') then
Z<='0';
ElsIf (a='1' and b='1') then
Z<='1';
End if;
End process;
End beh;
EXPERIMENTAL SET UP:-

![AND_2 Circuit Diagram]

**TRUTH TABLE**

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Logic Equation:  \( Z = a \text{ AND } b \)

**ii) Behavior Model if two i/p OR Gates**

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity OR_2 is
Port (a, b: in bit; z: out bit);
End OR_2;

Architecture OR_2_beh of OR_2 is
Begin process (a,b)
Begin
If (a='0' and b='0') then
Z<='0';
Elsif (a='0' and b='1') then
Z<='1';
Elsif (a='1' and b='0') then
Z<='1';
Elsif (a='1' and b='1') then
Z<='1';
End if;
End process;
End OR_2_beh;
EXPERIMENTAL SET UP:-

A

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OR_2

TRUTH TABLE

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Logic Equation: \( Z = a \text{ OR } b \)

iii) Behavior Model if two i/p NAND Gate

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity NAND_2 is
Port (a, b: in bit; z: out bit);
End NAND_2;

Architecture NAND_2_beh of NAND_2 is
Begin process (a, b)
Begin
If (a='0' and b='0') then
Z<='1';
Elsif (a='0' and b='1') then
Z<='1';
Elsif (a='1' and b='0') then
Z<='1';
Elsif (a='1' and b='1') then
Z<='0';
End if;
End process;
End NAND_2_beh;
EXPERIMENTAL SET UP:

A  
B  
\[ Z \]

**NAND_2**

**TRUTH TABLE**

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Logic Equation: \( Z = a \ AND \ b \)

iv) Behavior Model if two i/p NOR Gates

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity NOR_2 is
Port (a, b: in bit; z: out bit);
End NOR_2;

Architecture NOR_2_beh of NOR_2 is
Begin process (a, b)
Begin
If (a='0' and b='0') then
\( Z='1' \);
Elsif (a='0' and b='1') then
\( Z='0' \);
Elsif (a='1' and b='0') then
\( Z='0' \);
Elsif (a='1' and b='1') then
\( Z='0' \);
End if;
End process;
End NOR_2_beh;
EXPERIMENTAL SET UP:-

NOR_2

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Logic Equation: \( Z=a \ OR \ b \)

(iv) Behavior Model if two i/p EX-OR Gates

Library IEEE;
Use IEEE.std_logic_1164_all;
Use IEEE.std_logic_arith_all;

Entity EXOR_2 is
Port (a, b: in bit; z: out bit);
End EX-OR_2;

Architecture EXOR_2_beh of EXOR_2 is
Begin
Process (a, b)
Begin
If (a='0' and b='0') then
Z<='0';
Elsif (a='0' and b='1') then
Z<='1';
Elsif (a='1' and b='0') then
Z<='1';
Elsif (a='1' and b='1') then
Z<='0';
End if;
End process;
End EXOR_2_beh;
EXPERIMENTAL SET UP:-

EX-OR_2

TRUTH TABLE

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Logic Equation: \( Z = a \text{ EX-OR } b \)

v) Behavior Model of NOT Gate

Library IEEE;
Use IEEE.std_logic_1164_all;
Use IEEE.std_logic_arith_all;

Entity NOT_2 is
Port (a: in bit; z: out bit);
End NOT_2;

Architecture EXOR_2_beh of EXOR_2 is
Begin
Process (a)
Begin
If (a='0') then
Z<='1';
Elsif (a='1') then
Z<='0';
End if;
End process;
End NOT_2_beh;
EXPERIMENTAL SET UP:-

A \rightarrow_o^1 \rightarrow \neg^2 \rightarrow Z

Precautions:-

Make sure that there is no syntax and semantic error.

Result:-

All the VHDL codes of AND, OR, NAND, NOR, EX-OR and NOT gates are simulated and find correct
EXPERIMENT 2

Aim:-

Write the VHDL code and simulate it for Comparator.

SPECIFICATION OF APPARATUS USED:-

Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program:-

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity CMP_2 is
Port (a, b: in bit; ALB, AGB, AEB: out bit);
End CMP_2;

Architecture CMP_2_beh of CMP_2 is
Begin
Process (a, b)
Begin
If (a='0' and b='0') then
ALB<='0';
AGB<='0';
AFB<='1';
Elsif (a='0' and b='1') then
ALB<='1';
AGB<='0';
AFB<='0';
Elsif (a='1' and b='0') then
ALB<='0';
AGB<='1';
AFB<='0';
Elsif (a='1' and b='1') then
ALB<='0';
AGB<='0';
AFB<='1';
End if;
End process;
End CMP_2_beh;
EXPERIMENTAL SET UP:-

![Diagram of 1-Bit Comparator]

Precautions:-

Make sure that there is no syntax and semantic error.

Result:-
All the VHDL codes of 1-bit Comparator is simulated and synthesized.
EXPERIMENT NO.-3

Aim:-
Write a program for behavior model of 4-bit Comparator

SPECIFICATION OF APPARATUS USED:-
Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program:-

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity COM_2 is
Port (a, b: in bit_Vector (3 down to 0); z: out bit_vector (2 down to 0));
End COM_2;

Architecture COM_2_beh of COM_2 is
Begin
Process (a, b)
Begin
If (a=b) then
Z<="100’;
Elsif (a<b) then
Z<="010’;
Elsif (a>b) then
Z<="001’;
End if;
End process;
End COM_2_beh;

Precautions:-
Make sure that there is no syntax and semantic error.

Result:-
All the VHDL codes of 4-bit Comparator is simulated and synthesized.
EXPERIMENT NO.-4

Aim:-
Write the VHDL code and simulate it for 4:1 mux and 4:1 demultiplexer.

SPECIFICATION OF APPARATUS USED:-
Mentor graphics FPGA advantage software modelsim simulation tool.

Program: - Multiplexer’s Behavior Model
Library IEEE;
   Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity MUX_2 is
Port (i0, i1, i2, i3, s0, s1: in bit; z: out bit);
End MUX_2;

Architecture MUX_2_beh of MUX_2 is
Begin
Process (so, s1)
Begin
If (s1='0' and s0='0') then
   Z<='i0';
Elif (s1='0' and s0='1') then
   Z<='i1';
Elif (s1='1' and s0='0') then
   Z<='i2';
Elif (s1='1' and s0='1') then
   Z<='i3';
End if;
End process;
End MUX_2_beh;
EXPERIMENTAL SET UP:-

Truth table of 4-to-1 Multiplexer

<table>
<thead>
<tr>
<th>Select Input’s</th>
<th>Output Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>S0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Program: - Behavior Model of Demultiplexer (1 to 4)

Library IEEE;
Use IEEE.std_logic_1164_all;
Use IEEE.std_logic_arith_all;

Entity DEMUX_2 is
Port (a, s0, s1: in bit; z: out bit_vector (3 down to 0));
End DEMUX_2;

Architecture DEMUX_2_beh of DEMUX_2 is
Begin
Process (so, s1)
Begin
If (s1='0' and s0='0') then
Z (0) <=a;
Z (1) <='0';
Z (2) <='0';
Z (3) <='0';
Elsif (s1='0' and s0='1') then
  Z (0) <='0';
  Z (1) <=a;
  Z (2) <='0';
  Z (3) <='0';
Elsif (s1='1' and s0='0') then
  Z (0) <='0';
  Z (1) <='0';
  Z (2) <=a;
  Z (3) <='0';
Elsif (s1='1' and s0='1') then
  Z (0) <='0';
  Z (1) <='0';
  Z (2) <='0';
  Z (3) <=a;
End if;
End process;
End DEMUX_2_beh;

Precautions:-

Make sure that there is no syntax and semantic error.

Result:-
A) All the VHDL codes of 4 to 1 Multiplexer is simulated and synthesized.
B) All the VHDL codes of 4 to 1 Demultiplexer is simulated and synthesized.
EXPERIMENT NO.-5

Aim:

Write a program for behavior model of BCD to Seven Segment Decoder.

SPECIFICATION OF APPARATUS USED:-
Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program:

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity BCD_2 is
Port (b: in bit_Vector (3 down to 0); z: out bit_vector (6 down to 0));
End BCD_2;

Architecture BCD_2_beh of BCD_2 is
Begin
Process (b)
Begin
Case B is
When “0000”=>
S<="1111110”;
When “0001”=>
S<="0110000”;
When “0010”=>
S<="1101101”;
When “0011”=>
S<="1111001”;
When “0110”=>
S<="1011111”;
When “0111”=>
S<="1111000”;
When “1000”=>
S<="1111111”;
When “1001”=>
S<="1110011”;
When other =>
S<="000000”;
End case;
End process;
End BCD_Beh;
EXPERIMENTAL SET UP:

<table>
<thead>
<tr>
<th>BCD Input's</th>
<th>Output's</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D</td>
<td>a b c d e f g</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>1 1 1 1 1 1 0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>0 1 1 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>1 1 0 1 1 0 1</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>1 1 1 1 0 0 1</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0 1 1 0 0 1 1</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>1 0 1 1 0 1 1</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>1 0 1 1 1 1 0</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 1 1 0 0 0 0</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1 1 1 1 0 1 1</td>
</tr>
</tbody>
</table>

Truth table of BCD-to-Seven Segment Decoder
Wave Form of BCD-to-Seven Segment Decoder:-

Precautions:-

Make sure that there is no syntax and semantic error.

Result:-
All the VHDL codes of BCD to Seven Segment is simulated and synthesized.
Aim:

Write a VHDL program for behavior model of PIPO.

SPECIFICATION OF APPARATUS USED:
Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program:

```vhdl
Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity PIPO_2 is
Port (Pr, Cr, Clk: in bit; D: is bit_vector (2 down to 0); Q: out bit_vector (2 down to 0));
End PIPO_2;

Architectures PIPO_2 is
Begin
Process (Pr, Cr, Clk, D)
Begin
If (Pr='0' and Cr='1') then
Q<='111';
Elself (Pr='1' and Cr='0') then
Q<='000';
Elself (Pr='0' and Cr='1') then
Q<='0';
Elself (Pr='1' and Cr='1') then
Q<='111';
Elself (Pr='1' and Cr='1'and Clk='0' and Clk's event) then
Q<='D';
End if;
End process;
End PIPO_2 beh;
```
Block diagram of 4-bit Parallel in Parallel out Register

Waveform of 4-bit Parallel in Parallel Out Register:-

<table>
<thead>
<tr>
<th>/pico/d</th>
<th>/pico/clk</th>
<th>/pico/pr</th>
<th>/pico/ct</th>
<th>/pico/q</th>
<th>[0]</th>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D(0)</th>
<th>D(1)</th>
<th>D(2)</th>
<th>D(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clock</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1111</th>
<th>0000</th>
<th>1100</th>
<th>0011</th>
<th>1100</th>
<th>0000</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>0000</td>
<td>1100</td>
<td>0011</td>
<td>1100</td>
<td>0000</td>
<td>1111</td>
</tr>
</tbody>
</table>
Precautions:–

Make sure that there is no syntax and semantic error.

Result:–
All the VHDL codes of PIPO is simulated and synthesized.
EXPERIMENT NO.-7

Aim:-
Write VHDL programs for the following Circuit, check the wave forms and hardware generated.

A. Half Adder.
B. Full Adder.

SPECIFICATION OF APPARATUS USED:-
Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program: -
A). Behavior Model of Half Adder:-
Library IEEE:
    Use ieee.std_logic_1164.all;
    Use ieee.std_logic_arith.all;

    Entity HA_2 is
    Port (a, b: in bit; s, c: out bit);
    End HA_2;
    Architecture HA_2_beh of HA_2 is
    Begin
    Process (a, b)
    Begin
    If (a="0" and b="0") then
    s<="0";
    c<="0";
    Elsif (a="0" and b="1") then
    s<="1";
    c<="0";
    Elsif (a="1" and b="0") then
    s<="1";
    c<="0";
    Elsif (a="1" and b="1") then
    s<="0";
    c<="1";
    End if;
    End process;
    End HA_2_beh;
B) Behavior Model of FULL Adder:-

Library IEEE;
    Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Entity FA_2 is
    Port (a, b, cin: in bit; s, c: out bit);
End FA_2;
Architecture FA_2_beh of FA_2 is
    Begin
        Process (a, b,cin)
            Begin
                S<=a XOR B XOR Cin;
                C<= (a and b) OR (a and cin) OR (b and cin);
            End process;
    End FA_2_beh;

EXPERIMENTAL SET UP:-

<table>
<thead>
<tr>
<th>Input’s</th>
<th>Output’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>A  B  Cin</td>
<td>S  C</td>
</tr>
<tr>
<td>0  0  0</td>
<td>0  0</td>
</tr>
<tr>
<td>0  0  1</td>
<td>1  0</td>
</tr>
<tr>
<td>0  1  0</td>
<td>1  0</td>
</tr>
<tr>
<td>0  1  1</td>
<td>0  1</td>
</tr>
<tr>
<td>1  0  0</td>
<td>1  0</td>
</tr>
<tr>
<td>1  0  1</td>
<td>0  1</td>
</tr>
<tr>
<td>1  1  0</td>
<td>0  1</td>
</tr>
<tr>
<td>1  1  1</td>
<td>1  1</td>
</tr>
</tbody>
</table>
Truth table of Full Adder

Wave Form:-

| /full_adder/a | 1 |
| /full_adder/b | 1 |
| /full_adder/cin | 1 |
| /full_adder/c | 1 |

Precautions:-

Make sure that there is no syntax and semantic error.

Result:-
A) All the VHDL codes of Half Adder is simulated and synthesized.
B) All the VHDL codes of Full Adder is simulated and synthesized.
EXPERIMENT NO.-8

Aim:-
Write VHDL programs for ALU.

SPECIFICATION OF APPARATUS USED:-
Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program: -

Behavior Model of ALU.

Library IEEE:
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;

Entity ALU_2 is
Port (p, q: in bit_vector (3 down to 0); s: in bit_vector (2 down to 0); f: in bit_vector (3 down to 0));
End ALU_2;
Architecture ALU_2_beh of ALU_2 is
Function of “+”
Function add (a, b: bit_vector (2 down to 0)
Return bit_vector is
Variable cout: bit;
Variable cin: bit;
Variable sum: bit_vector (2 down to 0);
Begin
For i: in 0 to 2 loop
Sum (i): = a (i) XOR b (i) XOR Cin;
Cout: = (a (i) and b (i)) OR (b (i) and Cin) OR (Cin And a (i));
Cin: = Cout
End loop;
Return sum;
End “+”
--function of subtraction of 2 bit array
Function “-“(a, b: bit_vector (3 down to 0
Return bit_vector is
Variable cout: bit;
Variable Cin: bit=’0’;
Variable diff (i): bit_vector (3 down to 0);
Begin
For I in 0 to 3 loop
Cout: = ((not a (i) and b (i)) or ((b (i) and cin) or ((not a (i) and cin))
Diff (i):= a (i) xor b (i) xor cin;
Cin: = cout;
End loop;
Return diff (i);
End "-";

Begin
Process (p, q, and s)
Begin
Case s is
When “000”=>
F<= “0000”;  
When “001” =>
F =q-p;
When “010”=>
F=p-q;
When “001”=>
F =p+q;
When “100”=>
F =p and q;
When “101”=>
F = p xor q;
When “110”=>
F = p or q;
When “111” =>
F<= “1111”;
End case;
End process;
End ALU_Beh;
EXPERIMENTAL SET UP:-

<table>
<thead>
<tr>
<th>Select Input S2 S1 S0</th>
<th>Operation</th>
<th>Output F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>CLEAR</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>Q – P</td>
<td>Q – P</td>
</tr>
<tr>
<td>0 1 0</td>
<td>P – Q</td>
<td>P – Q</td>
</tr>
<tr>
<td>0 1 1</td>
<td>ADD</td>
<td>P + Q</td>
</tr>
<tr>
<td>1 0 0</td>
<td>XOR</td>
<td>P XOR Q</td>
</tr>
<tr>
<td>1 0 1</td>
<td>OR</td>
<td>P OR Q</td>
</tr>
<tr>
<td>1 1 0</td>
<td>AND</td>
<td>P AND Q</td>
</tr>
<tr>
<td>1 1 1</td>
<td>PRESET</td>
<td>1 1 1 1</td>
</tr>
</tbody>
</table>

Function Table of ALU(74381)
Waveform of ALU:

Precautions:-
Make sure that there is no syntax and semantic error.

Result:-
All the VHDL codes of ALU is simulated & synthesized.
EXPERIMENT NO.-9

Aim:-

Write a VHDL program for behavior model of D Flip-Flop.

SPECIFICATION OF APPARATUS USED:-
Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program:-

Library IEEE;
    Use IEEE.std_logic_1164.all;
    Use IEEE.std_logic_arith.all;

Entity DIFF_2 is
    Port (Pr, Cr, Clk: in bit; D: is bit_vector (2 down to 0); Q: out bit_vector (2 down to 0));
End DIFF_2;

Architecture DIFF_2_beh of DIFF_2 is
    Begin
    Process (Pr, Cr, Clk, D)
    Begin
    If (Pr='0' and Cr='1') then
        Q<='1';
    Elsif (Pr='1' and Cr='0') then
        Q<='0';
    Elsif (Pr='1' and Cr='1' and Clk='0' and Clk's event) then
        Q<=D;
    End if;
    End process;
End DIFF_2_beh;
EXPERIMENTAL SET UP:-

<table>
<thead>
<tr>
<th>PR</th>
<th>CR</th>
<th>D(i/p)</th>
<th>Output Q(t + 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Truth table of D flip-flop
Waveforms of D flip flop:-

<table>
<thead>
<tr>
<th>/dfl/dlk</th>
<th>/dfl/pr</th>
<th>/dfl/α</th>
<th>/dfl/d</th>
<th>/dfl/q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Precautions:-
Make sure that there is no syntax and semantic error.

Result:-
All the VHDL codes of D-Flip-Flop is simulated & synthesized.
EXPERIMENT NO.-10

Aim:

Write a VHDL program for behavior model of 3- Bit UP Counter.

SPECIFICATION OF APPARATUS USED:-

Mentor Graphics FPGA Advantage Software Modelsim Simulation Tool.

Program:-

Library IEEE;
Use IEEE.std_logic_1164.all;
Use IEEE.std_logic_arith.all;

Entity COUNTER_2 is
Port (Pr, Cr, Clk, t: in bit; Q: out bit_vector (0 to 2));
End DIFF_2;

Architecture COUNTER_2_beh of COUNTER_2 is
Function of “+”
Function add (a, b: bit_vector (0 down to 2))
Return bit_vector is
Variable cout: bit;
Variable cin: bit: = ‘0’;
Variable sum: bit_vector (0 to 2):= “000”;
Begin
For i: in 0 to 2 loop
Cout: = (a (i) and b (i)) OR (b (i) and Cin) OR (Cin And a (i));
Sum (i): a (i) XOR b (i) XOR Cin;
Cin: = Cout
End loop;
Return sum;
End “+”;
Begin
Process (Clk, Pr, Cr)
Begin
If (Pr=’0’ and Cr=’1’) then
Q<= ‘111’;
Elsif (Pr=’1’ and Cr=’0’) then
Q<= ‘000’;
Elsif (Pr=’1’ and Cr=’1’and Clk=’0’ and Clk’s event) then
Q<=Q+ “000”;
End


End if;
End process;
End COUNTER _2_beh

EXPERIMENTAL SET UP:-

Waveform of 3-bit Up Counter:-
**Precautions:**
Make sure that there is no syntax and semantic error.

**Result:** All the VHDL codes of D-Flip-Flop is simulated & synthesized.