# LIST OF EXPERIMENTS

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>NAME OF THE EXPERIMENT</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>PERFORM ANY FIVE EXPERIMENT USING VHDL</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Design all gates using VHDL.</td>
<td>3</td>
</tr>
</tbody>
</table>
| 2     | Write VHDL programs for the following circuits, check the wave forms and the hardware generated  
|       | a. Half adder  
|       | b. Full adder                                                                         | 7        |
| 3     | Write VHDL programs for the following circuits, check the wave forms and the hardware generated  
|       | a. Multiplexer  
|       | b. Demultiplexer                                                                     | 13       |
| 4     | Write VHDL program for encoder and check the wave forms and the hardware generated.     | 19       |
| 5     | Write a VHDL program for a decoder and check the wave forms and the hardware generated. | 22       |
| 6     | Write a VHDL program for a Down counter and check the wave forms and the hardware generated. | 25       |
| 7     | Write a VHDL program for a BCD to GRAY code converter and check the wave forms and the hardware generated. | 28       |
| 8     | Write a VHDL program for a T FLIP-FLOP and check the wave forms and the hardware generated. | 31       |
|       | **PERFORM ANY FIVE EXPERIMENT USING FPGA AND CPLD**                                     |          |
| 9     | Implement Half Adder using FPGA & CPLD.                                                  | 34       |
| 10    | Implement Full Adder using FPGA & CPLD.                                                  | 35       |
| 11    | Implement Delay Flip flop using FPGA & CPLD.                                             | 38       |
| 12    | Implement BCD to 7 segments Decoder using FPGA & CPLD.                                   | 41       |
| 13    | Implement an Up Counter using FPGA & CPLD.                                               | 44       |
| 14    | Implement 1-bit Comparator using FPGA & CPLD.                                            | 47       |
| 15    | Implement ALU using FPGA & CPLD.                                                         | 49       |
EXPERIMENT No. 1

Aim:- To Design Logic Gates using VHDL

LOGIC GATES:

A logic gate performs a logical operation on one or more logic inputs and produces a single logic output. The logic normally performed is Boolean logic and is most commonly found in digital circuits. Logic gates are primarily implemented electronically using diodes or transistors, but can also be constructed using electromagnetic relays (relay logic), fluidic logic, pneumatic logic, optics, molecules, or even mechanical elements.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
<th>A B</th>
<th>A AND B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
<th>A B</th>
<th>A OR B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
<th>A NOT A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In electronics a NOT gate is more commonly called an inverter. The circle on the symbol is called a bubble, and is generally used in circuit diagrams to indicate an inverted (active-low) input or output.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
<th>A B</th>
<th>A NAND B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### NOR

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A NOR B</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>

**Symbol:** $\overline{A + B}$

### XOR

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A XOR B</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>

**Symbol:** $A \oplus B$

### XNOR

**Truth Table:**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A XNOR B</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>1</td>
</tr>
</tbody>
</table>

**Symbol:** $\overline{A \oplus B}$ or $A \odot B$
Program:

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

entity all_ga is
  Port ( a : in  STD_LOGIC;
         b : in  STD_LOGIC;
         c : out  STD_LOGIC;
         c1 : out  STD_LOGIC;
         c2 : out  STD_LOGIC;
         c3 : out  STD_LOGIC;
         c4 : out  STD_LOGIC;
         c5 : out  STD_LOGIC;
         c6 : out  STD_LOGIC);
end all_ga;

architecture Behavioral of all_ga is
begin
  c  <=  a and b;
  c1 <=  a or b;
  c2 <=  a nand b;
  c3 <=  a nor b;
  c4 <=  a xor b;
  c5 <=  a xnor b;
  c6 <=  not b;
end Behavioral;

OUTPUT:

RTL View
Simulation Waveform

### Quiz Questions with answer.

Q.1 What is VHDL?
Ans. VHDL is the VHSIC Hardware Description Language. VHSIC is an abbreviation for Very High Speed Integrated Circuit.

Q.2 How many truth table entries are necessary for a four-input circuit?
Ans. 16

Q.3 What input values will cause an AND logic gate to produce a HIGH output?
Ans. All inputs of AND gate must be HIGH.

Q.4 Name all the basic gates.
Ans. i) AND ii) OR iii) NOT

Q.5 Name all the universal gates.
Ans. i) NAND ii) NOR

Q.6 What is the full form of IEEE?
Ans. Institute of Electrical and Electronic Engineering.

Q.7 What is the full form of ASCII?

Q.8 Define Entity.
Ans. It is an external view of a design unit.

Q.9 Why NAND and NOR are called universal gates?
Ans. Because all the basic gates can be derived from them.

Q.10 How many architectures are present in VHDL?
Ans. 4, behavior, dataflow, structural and mixed.
EXPERIMENT No. 2 (A)

Aim:- To Design a Half Adder using VHDL

Half adder

A half adder is a logical circuit that performs an addition operation on two one-bit binary numbers often written as \( A \) and \( B \).
The half adder output is a sum of the two inputs usually represented with the signals \( C_{out} \) and \( S \) where
\[
sum = 2 \times C_{out} + S.
\]
Following is the logic table and circuit diagram for half adder:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>( B )</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>

A

B

S

C
Program:

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 ha is
  Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          s : out  STD_LOGIC;
          c : out  STD_LOGIC);
end ha;

architecture Behavioral of ha is
begin
  s <= a xor b;
  c <= a and b;
end Behavioral;

OUTPUT:
Simulation Waveform

<table>
<thead>
<tr>
<th>Now: 1000 ns</th>
<th>90</th>
<th>120</th>
<th>150</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EXPERIMENT No. 2 (B)

Aim:- To Design a Full Adder using VHDL

Full adder

A full adder is a logical circuit that performs an addition operation on three one-bit binary numbers often written as $A$, $B$, and $C_{in}$. The full adder produces a two-bit output sum typically represented with the signals $C_{out}$ and $S$ where

$$sum = 2 \times C_{out} + S.$$

The full adder’s truth table is:

Truth Table:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$B$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

![Full Adder Circuit Diagram]
Program:

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 fa is
  Port ( a : in  STD_LOGIC;
         b : in  STD_LOGIC;
         cin : in  STD_LOGIC;
         s : out  STD_LOGIC;
         cout : out  STD_LOGIC);
end fa;

architecture Behavioral of fa is
begin
  s <= (a xor b) xor cin;
  cout <= (a and b) or (b and cin) or (a and cin);
end Behavioral;

OUTPUT:

RTL View
Quiz Questions with answer.

Q.1 Who is the father of VHDL?

Q.2 What is a test bench in VHDL?
Ans. A Test Bench in VHDL is code written in VHDL that provides stimulus for individual modules (also written in VHDL). Individual modules are instantiated by a single line of code showing the port.

Q.3 How many inputs and output are used in Full adder?
Ans. Three inputs and two output.

Q.4 What are the advantages of designing?
Ans. Advantages of Designing:
1. Designing is useful in quick implementation, testing and useful in complex circuits.
2. Designing reduces the design cycle.

Q.5 Why HDL is used?
Ans. HDL is used because it is easy to design, implement, test and document increasingly complex digital systems.

Q.6 How many types of architecture in VHDL?
Ans. 4
Q7. What is the difference between sequential and combinational ckt.s.?
Ans: Seq ckt.s have memory cell inside it and combinational has no memory in it.

Q8. Is it possible to construct full adder using half adder?
Ans: Yes, by using two half adders.

Q9. How many i/ps required for half subtractor?
Ans: Two, difference and a borrow.

Q10. Is it possible to construct full subtractor using half subtractor?
Ans: Yes, by using two half subtractor.
EXPERIMENT No. 3(A)

Aim:- To Design a Multiplexer using VHDL

Multiplexer

In digital circuit design, the selector wires are of digital value. In the case of a 2-to-1 multiplexer, a logic value of 0 would connect \( I_0 \) to the output while a logic value of 1 would connect \( I_1 \) to the output. In larger multiplexers, the number of selector pins is equal to \( \lceil \log_2(n) \rceil \) where \( n \) is the number of inputs.

A 4-to-1 multiplexer has a boolean equation where A, B, C and D are the two inputs, \( S_1 \) and \( S_0 \) are the select lines, and \( Y \) is the output:

<table>
<thead>
<tr>
<th>S1</th>
<th>S0</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>D</td>
</tr>
</tbody>
</table>

A boolean equation for the multiplexer is:

\[
Y = A \cdot \overline{S_1} + B \cdot S_1
\]

where \( A \) and \( B \) are the inputs, \( S_1 \) is the select line, and \( Y \) is the output.
Program:

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 abcd is
  Port ( a : in  STD_LOGIC;
         b : in  STD_LOGIC;
         c : in  STD_LOGIC;
         d : in  STD_LOGIC;
         s0 : in  STD_LOGIC;
         s1 : in  STD_LOGIC;
         y : out  STD_LOGIC);
end abcd;

architecture Behavioral of abcd is

begin
  y <= a when s0 = '0' and s1 = '0' else
       b when s0 = '0' and s1 = '1' else
       c when s0 = '1' and s1 = '0' else
       d;

end Behavioral;

OUTPUT:

RTL View

![RTL View Diagram]
### Simulation Waveform

<table>
<thead>
<tr>
<th>Now: 1000 ns</th>
<th>80</th>
<th>120</th>
<th>150</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The diagram shows the waveforms for different variables over a time period of 200 units. The waveforms are represented with bars indicating the state of each variable at different time points.
EXPERIMENT No. 3 (B)

Aim:- To Design a Demultiplexer using VHDL

Demultiplexer

Demultiplexers take one data input and a number of selection inputs, and they have several outputs. They forward the data input to one of the outputs depending on the values of the selection inputs. Demultiplexers are sometimes convenient for designing general purpose logic, because if the demultiplexer's input is always true, the demultiplexer acts as a decoder.

<table>
<thead>
<tr>
<th>I</th>
<th>S1</th>
<th>S0</th>
<th>ABCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0100</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0010</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0001</td>
</tr>
</tbody>
</table>

![Diagram of a demultiplexer with inputs and outputs]
Program:

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

entity dm is
  Port ( i : in  STD_LOGIC;
          s0 : in  STD_LOGIC;
          s1 : in  STD_LOGIC;
          a : out  STD_LOGIC;
          b : out  STD_LOGIC;
          c : out  STD_LOGIC;
          d : out  STD_LOGIC);
end dm;

architecture Behavioral of dm is
signal p,q : STD_LOGIC;
begin
  p <= not s0;
  q <= not s1;
  a <= i and p and q;
  b <= i and q and s0;
  c <= i and p and s1;
  d <= i and s1 and s0;
end Behavioral;

OUTPUT:
Quiz Questions with answer.

Q.1 Name combinational logic circuit which sends data coming from a single source to two or more separate destinations.
Ans: Demultiplexer

Q.2 What is the another name of Multiplexer.
Ans. Data Selector.

Q.3 How many control lines will be used for a 8 – to – 1 multiplexer?
Ans. The number of control lines for an 8 to 1 Multiplexer is 3.

Q.4 Which device changes serial data to parallel data.
Ans. The device which changes from serial data to parallel data is demultiplexer.

Q.5 How many select lines will a 16 to 1 multiplexer will have?
Ans. 4

Q.6 Is it possible to construct 4:1 mux using two 2:1 mux?
Ans. Yes

Q.7 How many outputs are there in 1:8 mux?
Ans. 8.

Q8: What is the difference between sequential and combinational ckt.s.?
Ans: Seq ckt.s have memory cell inside it and combinational has no memory in it.

Q9: What is the function of select line in mux?
Ans: It selects the particular input to go on output.

A10: What is the function of enable pin?
Ans: It enables the IC.
EXPERIMENT No. 4

Aim:- To Design an Encoder using VHDL

Encoder:

An encoder is a device, circuit, transducer, software program, algorithm or person that converts information from one format or code to another, for the purposes of standardization, speed, secrecy, security, or saving space by shrinking size.

<table>
<thead>
<tr>
<th>A3</th>
<th>A2</th>
<th>A1</th>
<th>A0</th>
<th>F1</th>
<th>F0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Program:

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

entity encod1 is
  Port ( a : in  STD_LOGIC_VECTOR (3 downto 0);
        q : out  STD_LOGIC_VECTOR (1 downto 0));
end encod1;

architecture Behavioral of encod1 is

begin
  q<="00" when a="0001" else
       "01" when a="0010" else
       "10" when a="0100" else
       "11";

end Behavioral;

OUTPUT:
Quiz Questions with answer:

Q.1 Name the examples of combinational logic circuits.
Ans. Examples of common combinational logic circuits include: half adders, full adders, multiplexers, demultiplexers, encoders and decoders.

Q.2 What do you mean by Digital Encoder?
Ans. Digital Encoder is a combinational circuit that generates a specific code at its outputs such as binary or BCD in response to one or more active inputs.

Q.3 How many encoder are there in Digital Electronics?
Ans. There are two main types of digital encoder. The Binary Encoder and the Priority Encoder.

Q.4 What is Combinational Logic?
Ans. Combinational Logic: A logic circuit in which the outputs are a function of the inputs. At any time, if you know the inputs, you can determine the outputs.

Q.5 What is stable state?
Ans. Stable State: An internal or external signal maintains a constant magnitude (or specified range or function) for a period of time determined by external input signals.

Q.6 Write the applications of Encoder and decoder.
Ans. They are used in communication systems.

Q.7 Name some encoders.
Ans Priority encoder, 4:2 encoder and etc.

Q.8 How many i/ps are in 4:2 encoder?
Ans 4 i/ps and 2 o/ps.

Q.9 How many select lines are present in 2:4 decoder?
Ans none.
EXPERIMENT No. 5

Aim:- To Design a 2 to 4 Line Decoder using VHDL

Decoder:

A decoder is a device which does the reverse of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode.

In digital electronics, a decoder can take the form of a multiple-input, multiple-output logic circuit that converts coded inputs into coded outputs, where the input and output codes are different. Decoding is necessary in applications such as data multiplexing, 7 segment display and memory address decoding.

<table>
<thead>
<tr>
<th>EN</th>
<th>S1</th>
<th>S0</th>
<th>Z0</th>
<th>Z1</th>
<th>Z2</th>
<th>Z3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Program:

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

--library UNISIM;
--use UNISIM.VComponents.all;

decoder entity is
  Port ( en : in  STD_LOGIC;
           s : in  STD_LOGIC_VECTOR (1 downto 0);
           z : out  STD_LOGIC_VECTOR (3 downto 0));
end decoder;

architecture arch of decoder is
signal p,q : STD_logic;
begin
  p <= not s(1);
  q <= not s(0);
  z(0) <= p and q;
  z(1) <= p and s(0);
  z(2) <= q and s(1);
  z(3) <= s(0) and s(1);
end arch;

OUTPUT:

RTL View

[Diagram of decoder circuit]
Simulation Waveform

<table>
<thead>
<tr>
<th>Now: 1000 ns</th>
<th>120</th>
<th>160</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s[1:0]</td>
<td>2h0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1]</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0]</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z[3:0]</td>
<td>4h1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[3]</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[2]</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1]</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[0]</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Quiz Questions with answer.**

Q.1 Name the examples of combinational logic circuits.
Ans. Examples of common combinational logic circuits include: half adders, full adders, multiplexers, demultiplexers, encoders and decoders.

Q.2 How many two-input AND and OR gates are required to realize Y=CD+EF+G ?
Ans. Y=CD+EF+G
Number of two input AND gates = 2
Number of two input OR gates = 5
One OR gate to OR CD and EF and next to OR of G & output of first OR gate.

Q.3 Which device converts BCD to Seven Segment?
Ans. A device which converts BCD to Seven Segment is called DECODER.

Q.4 What is a testbench in VHDL?
Ans. A Test Bench in VHDL is code written in VHDL that provides stimulus for individual modules (also written in VHDL). Individual modules are instantiated by a single line of code showing the port.

Q.5 What are the advantages of designing?
Ans. Advantages of Designing:
1. Designing is useful in quick implementation, testing and useful in complex circuits.
2. Designing reduces the design cycle.

Q.6: Write the applications of Encoder and decoder.
Ans: They are used in communication systems.

Q7: Name some encoders.
Ans: Priority encoder, 4:2 encoder and etc.

Q8: How many i/ps are in 4:2 encoder?
Ans: 4 i/ps and 2 o/ps.

Q9: How many select lines are present in 2:4 decoder?
Ans: none.
EXPERIMENT No. 6

Aim:- To Design a Down Counter using VHDL.

Down Counter:

In a binary up counter, a particular bit, except for the first bit, toggles if all the lower-order bits are 1's. The opposite is true for binary down counters. That is, a particular bit toggles if all the lower-order bits are 0's and the first bit toggles on every pulse.

Taking an example, $A_4 A_3 A_2 A_1 = 0100$. On the next count, $A_4 A_3 A_2 A_1 = 0011$. $A_1$, the lowest-order bit, is always complemented. $A_2$ is complemented because all the lower-order positions ($A_1$ only in this case) are 0's. $A_3$ is also complemented because all the lower-order positions, $A_2$ and $A_1$ are 0's. But $A_4$ is not complemented the lower-order positions, $A_3 A_2 A_1 = 011$, do not give an all 0 condition.
Program:

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 up_counter is
    Port ( clk : in  STD_LOGIC;
             sload : in  STD_LOGIC;
             q : out  STD_LOGIC_VECTOR (3 downto 0));
end up_counter;

architecture Behavioral of up_counter is
signal tmp : std_logic_vector(3 downto 0);
begin
process(clk)
begin
if(clk' event and clk = '1') then
    if sload = '0' then
        tmp <= "1111";
    else tmp <= tmp -1;
    end if;
end if;
end process;
q <= tmp;
end Behavioral;

OUTPUT:

RTL View
Quiz Questions with answer.

Q.1 What is sequential logic?
Ans. Sequential Logic: A logic circuit in which the outputs are a function of the present, and past inputs. The memory of past inputs involves the "state" of the system. At any time, if you know the present inputs, and state of the circuit, you can determine the outputs.

Q.2 How many Flip-Flops are required for mod–16 counter?
Ans. The number of flip-flops is required for Mod-16 Counter is 4.

Q.3 A 4-bit synchronous counter uses flip-flops with propagation delay times of 15 ns each. How much maximum possible time required for change of state?
Ans. 15 ns because in synchronous counter all the flip-flops change state at the same time.

Q.4 How many flip flops are required to construct a decade counter?
Ans. Decade counter counts 10 states from 0 to 9 (i.e. from 0000 to 1001). Thus four Flip Flop's are required.

Q.5 How many flip-flops are required to construct mod 30 counter?
Ans. 5

Q.6 What is a flip flop?
Ans. It is memory element which stores previous data.

Q.7 What is the function of clock in counter ckt?
Ans. It synchronize the operation of flip flops in counter ckt.

Q.8 What is the maximum count for decade counter?
An. From 0 to 9.
Q9: What is down counter?
An. When the qbar signal of previous ff is connected to clock of next ff.
Q10: What is the count for decade down counter?
An. From 9 to 0.
EXPERIMENT No. 7

Aim:- To Design a BCD to GRAY converter using VHDL

**BCD to GRAY Converter:**
It is a digital circuit that converts BCD numbers into Gray codes.

<table>
<thead>
<tr>
<th>2-bit Gray code</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
</tr>
<tr>
<td>01</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3-bit Gray code</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
</tr>
<tr>
<td>001</td>
</tr>
<tr>
<td>011</td>
</tr>
<tr>
<td>010</td>
</tr>
<tr>
<td>110</td>
</tr>
<tr>
<td>111</td>
</tr>
<tr>
<td>101</td>
</tr>
<tr>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4-bit Gray code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
</tr>
<tr>
<td>0001</td>
</tr>
<tr>
<td>0011</td>
</tr>
<tr>
<td>0010</td>
</tr>
<tr>
<td>0110</td>
</tr>
<tr>
<td>0111</td>
</tr>
<tr>
<td>0101</td>
</tr>
<tr>
<td>0100</td>
</tr>
<tr>
<td>1100</td>
</tr>
<tr>
<td>1101</td>
</tr>
<tr>
<td>1111</td>
</tr>
<tr>
<td>1110</td>
</tr>
<tr>
<td>1010</td>
</tr>
<tr>
<td>1011</td>
</tr>
<tr>
<td>1001</td>
</tr>
<tr>
<td>1000</td>
</tr>
</tbody>
</table>
Program:

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 bg is
    Port ( i : in  STD_LOGIC_VECTOR (2 downto 0);
       g : out  STD_LOGIC_VECTOR (2 downto 0));
end bg;

architecture Behavioral of bg is
begin
    process(i)
    begin
        case i is
            when "000" => g <= "000";
            when  "001" => g <= "001";
            when  "010" => g <= "011";
            when  "011" => g <= "010";
            when  "100" => g <= "110";
            when  "101" => g <= "111";
            when  "110" => g <= "101";
            when others => g <= "100";
        end case;
    end process;
end Behavioral;
OUTPUT: RTL View

Simulation Waveform

<table>
<thead>
<tr>
<th>Now: 1000 ns</th>
<th>80</th>
<th>90</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2:0)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0)</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(2:0)</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0)</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(1)</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(0)</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(2:0)</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0)</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(1)</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(0)</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Quiz Questions with answer.
Q.1 What is VHDL?
Ans. VHDL is the VHSIC Hardware Description Language. VHSIC is an abbreviation for Very High Speed Integrated Circuit.
Q.2 How many truth table entries are necessary for a four-input circuit?
Ans. 16
Q.3 How many bits are there in BCD code?
Ans. 4
Q.4 What is Combinational Logic?
Ans. Combinational Logic: A logic circuit in which the outputs are a function of the inputs. At any time, if you know the inputs, you can determine the outputs.
Q.5 What is stable state?
Ans. Stable State: An internal or external signal maintains a constant magnitude (or specified range or function) for a period of time determined by external input signals.
Q.6 What is BCD to Gray converter?
Ans. The converter which converts bcd code into gray code.
Q.7 What is the application of above code converter?
Ans. We use in communication systems.
Q.8 BCD to Gray converter is a combinational or sequential ckt?
Ans. Combinational ckt.
Q9: Write down the method of Binary to Gray conversion.
Ans: Using the Ex-Or gates
Q10: Convert 0101 to Decimal.
Ans: 5
EXPERIMENT No. 8

Aim:- To Design a Toggle Flip Flop using VHDL.

T(Toggle) Flip Flop:

If the T input is high, the T flip-flop changes state ("toggles") whenever the clock input is strobed. If the T input is low, the flip-flop holds the previous value. This behavior is described by the characteristic equation:

\[ Q_{next} = T \oplus Q = T\overline{Q} + \overline{T}Q \]

<table>
<thead>
<tr>
<th>Characteristic table</th>
<th>Excitation table</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( Q )</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:

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 t_ff is
    Port ( pr : in  STD_LOGIC;
           clr : in  STD_LOGIC;
           clk : in  STD_LOGIC;
           t : in  STD_LOGIC;
           q : inout  STD_LOGIC;
           qbar : out  STD_LOGIC);
end t_ff;

architecture Behavioral of t_ff is
begin
    process(clk,clr,pr)
    begin
        if( pr='0' and clr ='1') then
            q <= '1';
        elsif ( pr ='1' and clr ='0') then
            q <='0';
        elsif(pr = '1' and clr ='1' and clk = '0' and clk' event)then
            q <= (not t and q) or (not q and t);
        end if;
    end process;
    qbar <= not q;
end Behavioral;

OUTPUT: RTL View
Quiz Questions with answer:

Q.1 Define flip-flop.
Ans. A flip-flop is a device that can maintain binary information until it is directed by an input signal to change its state. There are several different types of flip-flops, the more commonly used are the D-FF and the JK-FF. Flip-flops are used in sequential circuit design.

Q. 2 The MSI chip 7474 is
Ans. MSI chip 7474 dual edge triggered D Flip-Flop.

Q. 3 How many flip-flops are required to construct mod 30 counter?
Ans 5

Q.4 The output of SR flip flop when S=1, R=0 is
Ans As for the SR flip-flop S=set input R=reset input, when S=1, R=0, Flip-flop will be set.

Q.5 The number of flip flops contained in IC 7490 is
Ans 2.

Q6 What are the I/Ps of JK flip–flop where this race round condition occurs?
Ansh Both the inputs are 1

Q7 Flip flop is astable or bistable?
Ans Bistable.

Q8 When RS flip-flop is said to be in a SET state?
Ans When the output is 1

Q9 What is the function of clock signal in flip-flop?
Ans. To get the output at known time.

Q10: What is the advantage of JK flip-flop over RS flip-flop?
Ans. In RS flip-flop when both the inputs are 1 output is undetermined.
EXPERIMENT No. 9

Aim:- Implement Half Adder using FPGA & CPLD.

Program:

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 ha is
  Port ( a : in  STD_LOGIC;
         b : in  STD_LOGIC;
         s : out  STD_LOGIC;
         c : out  STD_LOGIC);
end ha;

architecture Behavioral of ha is
begin
  s <= a xor b;
c <= a and b;
end Behavioral;

Output

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</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>
Quiz Questions with answer.
Q.1  Who is the father of VHDL?
Q.2  What is a testbench in vhdl?
Ans. A Test Bench in VHDL is code written in VHDL that provides stimulus for individual modules (also written in VHDL). Individual modules are instantiated by a single line of code showing the port.
Q.3  How many inputs and output are used in Full adder?
Ans.  Three inputs and two output.
Q.4  What are the advantages of designing?
Ans.  Advantages of Designing:
1. Designing is useful in quick implementation, testing and useful in complex circuits.
2. Designing reduces the design cycle.
Q.5  Why HDL is used?
Ans. HDL is used because it is easy to design, implement, test and document increasingly complex digital system.
Q.6  Give the basic rules for binary addition?
Ans.  0+0 = 0; 0+1 = 1 ; 1+1 = 1 0 ; 1+0 = 1.
Q.7  What is the drawback of half adder?
Ans: We can’t add carry bit from previous stage.
Q.8  What is the difference b/w half adder & half sub tractor?
Ans: Half adder can add two bits & half sub tractor can subtract two bits.
Q.9  Define Nibble?
Ans. Combination of four bits
EXPERIMENT No. 10

Aim:- Implement Full Adder using FPGA & CPLD.

Full adder

A full adder is a logical circuit that performs an addition operation on three one-bit binary numbers often written as $A$, $B$, and $C_{in}$. The full adder produces a two-bit output sum typically represented with the signals $C_{out}$ and $S$ where

$$\text{sum} = 2 \times C_{out} + S.$$ 

The full adder's truth table is:

Truth Table:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$ $B$</td>
<td>$C_i$ $C_o$ $S$</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>
Program:

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;

dentity fa is
  Port ( a : in  STD_LOGIC;
        b : in  STD_LOGIC;
        cin : in  STD_LOGIC;
        s : out  STD_LOGIC;
        cout : out  STD_LOGIC);
end fa;

architecture Behavioral of fa is
begin
  s <= (a xor b) xor cin;
  cout <= (a and b) or (b and cin) or (a and cin);
end Behavioral;

OUTPUT:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
**Quiz Questions with answer.**

Q.1 Who is the father of **VHDL**?

Q.2 What is a testbench in **vhdl**?
Ans. A Test Bench in VHDL is code written in VHDL that provides stimulus for individual modules (also written in VHDL). Individual modules are instantiated by a single line of code showing the port.

Q.3 How many inputs and output are used in Full adder?
Ans. Three inputs and two output.

Q.4 What are the advantages of designing?
Ans. Advantages of Designing:
1. Designing is useful in quick implementation, testing and useful in complex circuits.
2. Designing reduces the design cycle.

Q.5 Why **HDL** is used?
Ans. HDL is used because it is easy to design, implement, test and document increasingly complex digital system.

Q.6 Give the basic rules for binary addition?
Ans. 0+0 = 0; 0+1 = 1; 1+1 = 10; 1+0 = 1.

Q.7 What is the drawback of half adder?
Ans. We can’t add carry bit from previous stage.

Q.8 What is the difference b/w half adder & half sub tractor?
Ans. Half adder can add two bits & half sub tractor can subtract two bits.

Q.9 Define Nibble?
Ans. Combination of four bits.
EXPERIMENT No. 11

Aim:- Implement Delay Flip-Flop using FPGA & CPLD.

Flip-Flops:

In digital circuits, a flip-flop is a term referring to an electronic circuit (a bistable multivibrator) that has two stable states and thereby is capable of serving as one bit of memory. Today, the term flip-flop has come to mostly denote non-transparent (clocked or edge-triggered) devices.

A flip-flop is usually controlled by one or two control signals and/or a gate or clock signal. The output often includes the complement as well as the normal output.

D(Delay) Flip-Flop:

The D flip-flop is the most common flip-flop in use today. It is better known as delay flip-flop. The Q output always takes on the state of the D input at the moment of a positive edge (or negative edge if the clock input is active low). It is called the D flip-flop for this reason, since the output takes the value of the D input or Data input, and delays it by maximum one clock count. The D flip-flop can be interpreted as a primitive memory cell, zero-order hold, or delay line. Whenever the clock pulses, the value of $Q_{\text{next}}$ is D and $Q_{\text{prev}}$ otherwise.

Truth table:

<table>
<thead>
<tr>
<th>Clock</th>
<th>D</th>
<th>Q</th>
<th>$Q_{\text{prev}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rising edge</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>Rising edge</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>Non-Rising</td>
<td>X</td>
<td>$Q_{\text{prev}}$</td>
<td></td>
</tr>
</tbody>
</table>
Program:

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 d_ff is
  Port ( clk : in  STD_LOGIC;
         clr : in  STD_LOGIC;
         d : in  STD_LOGIC;
         pr : in  STD_LOGIC;
         q : out  STD_LOGIC);
end d_ff;

architecture Behavioral of d_ff is
begin
process(clk,clr,pr)
begin
  if( pr='0' and clr ='1') then
    q<='1';
  elsif ( pr ='1' and clr ='0') then
    q<='0';
  elsif(pr = '1' and clr ='1'and falling_edge(clk))then
    q<= d;
  end if;
end process;
end Behavioral;
OUTPUT:

<table>
<thead>
<tr>
<th>Clock</th>
<th>D</th>
<th>Q</th>
<th>Q\text{prev}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rising edge</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>Rising edge</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>Non-Rising</td>
<td>X</td>
<td>Q\text{prev}</td>
<td></td>
</tr>
</tbody>
</table>

Quiz Questions with answer.

Q.1 Define flip-flop.
Ans. A flip-flop is a device that can maintain binary information until it is directed by an input signal to change its state. There are several different types of flip-flops, the more commonly used are the D-FF and the JK-FF. Flip-flops are used in sequential circuit design.

Q. 2 The MSI chip 7474 is
Ans. MSI chip 7474 dual edge triggered D-Flip-Flop.

Q. 3 How many flip-flops are required to construct mod 30 counter?
Ans 5

Q. 4 The output of SR flip flop when S=1, R=0 is
Ans As for the SR flip-flop S=set input R=reset input ,when S=1, R=0, Flip-flop will be set.

Q. 5 The number of flip flops contained in IC 7490 is
Ans 2.

Q. 6 What are the I/Ps of JK flip–flop where this race round condition occurs?
Ans; Both the inputs are 1

Q. 7 Flip flop is astable or bistable?
Ans Bistable.

Q. 8 When RS flip-flop is said to be in a SET state?
Ans. When the output is 1

Q. 9 What is the function of clock signal in flip-flop?
Ans. To get the output at known time.

Q. 10 What is the advantage of JK flip-flop over RS flip-flop?
Ans. In RS flip-flop when both the inputs are 1 output is undetermined.
EXPERIMENT No. 12

Aim:- Implement BCD to 7 segment Decoder using FPGA & CPLD.

BCD to 7 segment Decoder:
It is a digital circuit that decodes BCD numbers into 7 segment numbers that can be used for 7 segment displays and other applications.

<table>
<thead>
<tr>
<th>INPUTS (ABCD)</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>e</th>
<th>f</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0010</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0011</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0100</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0101</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0110</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0111</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Program:

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 bs is
  Port ( i : in  STD_LOGIC_VECTOR (3 downto 0);
         o : out  STD_LOGIC_VECTOR (6 downto 0));
end bs;

architecture Behavioral of bs is

begin

  o <= "1111110" when i = "0000" else
      "0110000" when i = "0001" else
      "1101101" when i = "0010" else
      "1111001" when i = "0011" else
      "0110011" when i = "0100" else
      "1011011" when i = "0101" else
      "1011111" when i = "0110" else
      "1110000" when i = "0111" else
      "1111111" when i = "1000" else
      "1111011" when i = "1001";

end Behavioral;

OUTPUT:
QUIZ QUESTIONS WITH ANSWERS.

Q.1 Name the examples of combinational logic circuits.
Ans. Examples of common combinational logic circuits include: half adders, full adders, multiplexers, demultiplexers, encoders and decoders.

Q.2 Which device converts BCD to Seven Segment?
Ans. A device which converts BCD to Seven Segment is called DECODER.

Q.3 What is BCD to Seven segment decoder?
Ans. A binary coded decimal (BCD) to 7-segment display decoder such as the TTL 74LS47 or 74LS48, have 4 BCD inputs and 7 output lines, one for each LED segment. This allows a smaller 4-bit binary number (half a byte) to be used to display all the ternary numbers from 0 to 9 and by adding two displays together; a full range of numbers from 00 to 99 can be displayed with just a single byte of 8 data bits.

Q.4 What is decoder?
Ans. A Decoder IC, is a device which converts one digital format into another and the most commonly used device for doing this is the Binary Coded Decimal (BCD) to 7-Segment Display Decoder.

Q5: Q.5 What are the advantages of designing?
Ans. Advantages of Designing:
1. Designing is useful in quick implementation, testing and useful in complex circuits.
2. Designing reduces the design cycle.

Q6: Write the applications of Encoder and decoder.
Ans: They are used in communication systems.

Q7: Name some encoders.
Ans: Priority encoder , 4:2 encoder and etc.

Q8: How many i/ps are in 4:2 encoder?
Ans: 4 i/ps and 2 o/ps.

Q9: How many select lines are present in 2:4 decoder?
Ans: none.

Q10: How many outputs are present in 3:8 decoder?
Ans: 8.
EXPERIMENT No. 13

Aim:- Implement an Up Counter using FPGA & CPLD.

Up Counter:

A synchronous binary counter counts from 0 to $2^N - 1$, where $N$ is the number of bits/flip-flops in the counter. Each flip-flop is used to represent one bit. The flip-flop in the lowest-order position is complemented/toggled with every clock pulse and a flip-flop in any other position is complemented on the next clock pulse provided all the bits in the lower-order positions are equal to 1.

Take for example $A_4 A_3 A_2 A_1 = 0011$. On the next count, $A_4 A_3 A_2 A_1 = 0100$. $A_1$, the lowest-order bit, is always complemented. $A_2$ is complemented because all the lower-order positions ($A_1$ only in this case) are 1's. $A_3$ is also complemented because all the lower-order positions, $A_2$ and $A_1$ are 1's. But $A_4$ is not complemented the lower-order positions, $A_3 A_2 A_1 = 011$, do not give an all 1 condition.

To implement a synchronous counter, we need a flip-flop for every bit and an AND gate for every bit except the first and the last bit. The diagram below shows the implementation of a 4-bit synchronous up-counter.
Program:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

--library UNISIM;
--use UNISIM.VComponents.all;

entity up_counter is
  Port ( clk : in  STD_LOGIC;
         sload : in  STD_LOGIC;
         clr : in  STD_LOGIC;
         q : out  STD_LOGIC_VECTOR (3 downto 0));
end up_counter;

architecture Behavioral of up_counter is
signal tmp : std_logic_vector(3 downto 0);
begin
process(clk)
begin
  if(clk' event and clk = '1') then
    if clr = '0' then
      tmp <= "0000";
    elsif sload = '1' then
      if tmp = "1010" then
        tmp <= "0000";
      else tmp <= tmp +1;
      end if;
    end if;
    end if;
    end if;
end process;
q <= tmp;
end Behavioral;
Q.1 What is sequential logic?
Ans. Sequential Logic: A logic circuit in which the outputs are a function of the present, and past inputs. The memory of past inputs involves the "state" of the system. At any time, if you know the present inputs, and state of the circuit, you can determine the outputs.

Q.2 How many Flip-Flops are required for mod-16 counter?
Ans. The number of flip-flops is required for Mod-16 Counter is 4.

Q.3 A 4-bit synchronous counter uses flip-flops with propagation delay times of 15 ns each. How much maximum possible time required for change of state?
Ans. 15 ns because in synchronous counter all the flip-flops change state at the same time.

Q.4 How many flip-flops are required to construct a decade counter?
Ans. Decade counter counts 10 states from 0 to 9 (i.e. from 0000 to 1001). Thus four Flip Flop’s are required.

Q.5 How many flip-flops are required to construct mod-30 counter?
Ans. 5

Q6: What is a flip flop?
Ans. It is memory element which stores previous data.

Q7: What is the function of clock in counter ckt?
Ans. It synchronize the operation of flip-flops in counter ckt.

Q8: What is the maximum count for decade counter?
Ans. From 0 to 9.

Q9: What is down counter?
Ans. When the qbar signal of previous ff is connected to clock of next ff.

Q10: What is the count for decade down counter?
Ans. From 9 to 0.
EXPERIMENT No. 14

Aim:- Implement a 1-bit Comparator using FPGA & CPLD.

Comparator:

In electronics, a comparator is a device which compares two voltages or currents and switches its output to indicate which is larger.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A&lt; B</th>
<th>A= B</th>
<th>A&gt; B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Program:

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 compare is
  Port ( a : in  STD_LOGIC;
         b : in  STD_LOGIC;
         agb : out  STD_LOGIC;
         aeb : out  STD_LOGIC;
         alb : out  STD_LOGIC);
end compare;

architecture Behavioral of compare is
signal p,q : std_logic;
begin
  p <= not a;
  q <= not b;
  agb <= a and q;
  aeb <= (a and b) or (p and q);
  alb <= p and b;
end Behavioral;

QUIZ QUESTIONS WITH ANSWERS.
Q.1 What is Identity Comparator?
Ans. Identity Comparator - an Identity Comparator is a digital comparator that has only one output terminal for when A = B either "HIGH" A = B = 1 or "LOW" A = B = 0
Q.2 What is Magnitude Comparator?
Ans. Magnitude Comparator - a Magnitude Comparator is a type of digital comparator that has three output terminals, one each for equality, A = B greater than, A > B and less than A < B.
Q3: Q.3 How many inputs and output are used in Full adder?
Ans. Three inputs and two output.
Q.4 What are the advantages of designing?
Ans. Advantages of Designing:
1. Designing is useful in quick implementation, testing and useful in complex circuits.
2. Designing reduces the design cycle.
Q.5 Why HDL is used?
Ans. HDL is used because it is easy to design, implement, test and document increasingly complex digital system.

Q6. How many types of architecture in VHDL?
Ans: 4

Q7. What is the difference between sequential and combinational ckt.s?
Ans: Seq ckt.s have memory cell inside it and combinational has no memory in it.

Q8. Is it possible to construct full adder using half adder?
Ans: Yes, by using two half adders.

Q9. How many i/p s required for half subtractor?
Ans: Two, difference and a borrow.

Q10. Is it possible to construct full subtractor using half subtractor?
Ans: Yes, by using two half subtractor.
EXPERIMENT No. 15

Aim:- Implement an ALU using FPGA & CPLD.

ALU:

In computing, an arithmetic logic unit (ALU) is a digital circuit that performs arithmetic and logical operations. The ALU is a fundamental building block of the central processing unit (CPU) of a computer, and even the simplest microprocessors contain one for purposes such as maintaining timers. The processors found inside modern CPUs and graphics processing units (GPUs) accommodate very powerful and very complex ALUs; a single component may contain a number of ALUs.

The inputs to the ALU are the data to be operated on (called operands) and a code from the control unit indicating which operation to perform. Its output is the result of the computation.

Program:

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

entity alu_1 is
  Port ( a : in  STD_LOGIC_VECTOR (3 downto 0);
         b : in  STD_LOGIC_VECTOR (3 downto 0);
         op : in  STD_LOGIC_VECTOR (2 downto 0);
         y : out  STD_LOGIC_VECTOR (7 downto 0));
end alu_1;

architecture Behavioral of alu_1 is
begin
  y<=
   ("0000"&a) + ("0000"&b) when op="000" else
   ("0000"&a) - ("0000"&b) when op="001" else
   a*b when op="010" else
   ("0000"&a) and ("0000"&b) when op="011" else
   ("0000"&a) or ("0000"&b) when op="100" else
   ("0000"&a) xor ("0000"&b) when op="101" else
   ("0000"&a) nor ("0000"&b) when op="110" else
   ("0000"&a) nand ("0000"&b) when op="111";
end Behavioral;
Q.1 What is VHDL?
Ans. VHDL is the VHSIC Hardware Description Language. VHSIC is an abbreviation for Very High Speed Integrated Circuit.

Q.2 How many truth table entries are necessary for a four-input circuit?
Ans. 16

Q.3 What input values will cause an AND logic gate to produce a HIGH output?
Ans. All inputs of AND gate must be HIGH.

Q.4 Name all the basic gates.
Ans. i) AND ii) OR iii) NOT

Q.5 Name all the universal gates.
Ans. i) NAND ii) NOR

Q.6 What is the full form of IEEE?
Ans. Institute of Electrical and Electronic Engineering.

Q.7 What is the full form of ASCII?

Q.8 Define Entity.
Ans. It is an external view of a design unit.

Q.9 Why NAND and NOR are called universal gates?
Ans. Because all the basic gates can be derived from them.

Q.10 How many architectures are present in VHDL?
Ans. 4, behavior, dataflow, structural and mixed.