DRONACHARYA
College of Engineering

LABORATORY MANUAL

B.Tech. Semester- VI

VLSI DESIGN LAB
Subject code: LC-EE-312G

Prepared by:	**Checked by:**	**Approved by:**
Mrs. Sonia Arora	Mrs. Dimple Saproo	Name : Prof. (Dr.) Isha Malhotra
Sign.: ……………………..	**Sign.: …………………..**	**Sign.: …………………..**

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
DRONACHARYA COLLEGE OF ENGINEERING
KHENTAWAS, FARRUKH NAGAR, GURUGRAM (HARYANA)

Table of Contents

Vision and Mission of the Institute

Vision:

To impart Quality Education, to give an enviable growth to seekers of learning, to groom them as World Class Engineers and managers competent to match the expending expectations of the Corporate World has been ever enlarging vision extending to new horizons of Dronacharya College of Engineering

Mission:

1. To prepare students for full and ethical participation in a diverse society and encourage lifelong learning by following the principle of 'Shiksha evam Sahayata' i.e. Education & Help.
2. To impart high-quality education, knowledge and technology through rigorous academic programs, cutting-edge research, & Industry collaborations, with a focus on producing engineers& managers who are socially responsible, globally aware, & equipped to address complex challenges.
3. Educate students in the best practices of the field as well as integrate the latest research into the academics.
4. Provide quality learning experiences through effective classroom practices, innovative teaching practices and opportunities for meaningful interactions between students and faculty.
5. To devise and implement programmes of education in technology that are relevant to the changing needs of society, in terms of breadth of diversity and depth of specialization.

Vision and Mission of the Department

Vision:

To become a Centre of Excellence in teaching and research in Information Technology for producing skilled professionals having a zeal to serve society.

Mission:

M1: To create an environment where students can be equipped with strong fundamental concepts, programming and problem solving skills.

M2: To provide an exposure to emerging technologies by providing hands on experience for generating competent professionals.

M3: To promote Research and Development in the frontier areas of Information Technology and encourage students for pursuing higher education.

M4: To inculcate in students ethics, professional values, team work and leadership skills.

Programme Educational Objectives (PEOs)

PEO1: Engineers will practice the profession of engineering using a systems perspective and analyze, design, develop, optimize & implement engineering solutions and work productively as engineers, including supportive and leadership roles on multidisciplinary teams.

PEO2: Continue their education in leading graduate programs in engineering & interdisciplinary areas to emerge as researchers, experts, educators & entrepreneurs and recognize the need for, and an ability to engage in continuing professional development and life-long learning.

PEO3: Engineers, guided by the principles of sustainable development and global interconnectedness, will understand how engineering projects affect society and the environment.

PEO4: Promote Design, Research, and implementation of products and services in the field of Engineering through Strong Communication and Entrepreneurial Skills.

PEO5: Re-learn and innovate in ever-changing global economic and technological environments of the 21st century.

Programme Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 :Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6 :The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 :Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 :Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 : Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO 1 Equip themselves to potentially rich & employable field of Engineering.

PSO 2 Pursue higher studies in the area of Microelectronics and Chip Designing.

PSO 3 Take up self-employment in Indian & global chip designing market.

PSO 4 Meet the requirements of the Industrial standard.

University Syllabus

1) Design of Half-Adder, Full Adder, Half Subtractor, Full Subtractor
2) Design a parity generator
3) Design a 4 Bit comparator
4) Design a RS & JK Flip flop
5) Design a 4: 1 Multiplexer
6) Design a 4 Bit Up / Down Counter with Loadable Count
7) Design a 3: 8 decoder
8) Design a 8 bit shift register
9) Design a arithmetic unit
10) Implement ADC & DAC interface with FPGA
11) Implement a serial communication interface with FPGA

Course Outcomes (COs)

After completion of this course, students will be able to :

CO1: Demonstrate a clear Understanding in hardware design language.

CO2: Model Combinational circuits using hardware description language and validate its functionality.

CO3: Model Sequential circuits using hardware description language and validate its functionality.

CO4: Design and implement sub systems on a FPGA board.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	1	1	2						2
CO2	2	3	2	2	3	2			1			3
CO3	2	3	2	2	3	2			1			3
CO4	1	1	2	3	3	2	1		3	1	3	3

CO-PSO Mapping

	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	2
CO2	3	2	3	3
CO3	3	2	3	3
CO4	3	3	3	3

Course Overview

Understand the physical design process of Digital Integrated circuits. Implement various combinational and sequential circuits using VHDL on FPGA. Implement schematic and layout of various digital logic circuits using Xilinx EDA tool.

The following guidelines should be followed regarding award of marks:

Class Work : 25 Marks
Theory : 25Marks
Total : 50 Marks
Duration of Exam. : 3 Hrs.

General instructions: Practical examination to be conducted immediately after the Pre-university examinations covering entire lab experiments given above. Evaluation is a serious process that is to be conducted under the equal responsibility of both the internal and external examiners. The number of candidates evaluated per group should not exceed 5 and in a batch size of not more than 30. Students shall be allowed for the University examination only on submitting the duly certified internal examination record. The external examiner shall endorse the record and conduct the external examination as per guidelines laid by affiliated university

List of Experiments mapped with Cos

S. No	List of Experiments	Course Outcomes
1.	Design of Half-Adder, Full Adder, Half Subtractor, Full Subtractor	CO1,CO2
2.	Design a parity generator	CO1,CO2
3.	Design a 4 Bit comparator	CO1,CO2
4.	Design a RS & JK Flip flop	CO1,CO3
5.	Design a 4: 1 Multiplexer	CO1,CO2
6.	Design a 4 Bit Up / Down Counter with Loadable Count	CO1,CO2
7.	Design a 3: 8 decoder	CO1,CO2
8.	Design a 8 bit shift register	CO1,CO3
9.	Design a arithmetic unit	CO1,CO2, CO3
10.	Implement ADC & DAC interface with FPGA	CO1,CO2, CO3,CO4
11.	Implement a serial communication interface with FPGA	CO1,CO2, CO3,CO4

DOs and DON'Ts

DOs

1. Login-on with your username and password.
2. Log off the Computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

DON'Ts

1. Do not share your username and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites is strictly prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

General Safety Precautions

Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.
2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.
3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.
4. Immediately call medical emergency and security. Remember! Time is critical; be best.

Precautions (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.
2. If fire continues, try to curb the fire, if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible isolate the burning equipment from the other surrounding equipment.
3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.
4. Call security and emergency department immediately:

Emergency: Reception
Security: Gate No.1)

Guidelines to students for report preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows:-

1) All files must contain a title page followed by an index page. ***The files will not be signed by the faculty without an entry in the index page.***

2) Student's Name, Roll number and date of conduction of experiment must be written on all pages.

3) For each experiment, the record must contain the following :

(i) Aim/Objective of the experiment .

(ii) Pre-experiment work (as given by the faculty) .

(iii) Lab assignment questions and their solutions.

(iv) Test Cases (if applicable to the course) .

(v) Results/ output.

Note:

1. Students must bring their lab record along with them whenever they come for the lab.
2. Students must ensure that their lab record is regularly evaluated.

Lab Assessment Criteria

An estimated 10 lab classes are conducted in a semester for each lab course. These lab classes are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment as well as internal marks in the lab course.

Grading Criteria	Exemplary (4)	Competent (3)	Needs Improvement (2)	Poor (1)
AC1: Pre-Lab written work (this may be assessed through viva)	Complete procedure with underlined concept is properly written	Underlined concept is written but procedure is incomplete	Not able to write concept and procedure	Underlined concept is not clearly understood
AC2: Program Writing/ Modeling	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied, Program/solution written is readable	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied	Assigned problem is properly analyzed & correct solution designed	Assigned problem is properly analyzed
AC3: Identification & Removal of errors/ bugs	Able to identify errors/ bugs and remove them	Able to identify errors/ bugs and remove them with little bit of guidance	Is dependent totally on someone for identification of errors/ bugs and their removal	Unable to understand the reason for errors/ bugs even after they are explicitly pointed out
AC4:Execution & Demonstration	All variants of input /output are tested, Solution is well demonstrated and implemented concept is clearly explained	All variants of input /output are not tested, However, solution is well demonstrated and implemented concept is clearly explained	Only few variants of input /output are tested, Solution is well demonstrated but implemented concept is not clearly explained	Solution is not well demonstrated and implemented concept is not clearly explained
AC5:Lab Record Assessment	All assigned problems are well recorded with objective, design constructs and solution along with Performance analysis using all variants of input and output	More than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	Less than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	

LAB EXPERIMENTS

PRELAB SESSION

VLSI Questions with answer.

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

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

Q. 3What 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.

Q7. What is the full form of ASCII?
Ans. American Standard Code for information Interchange.

Q8. Define Entity.
Ans. It is an external view of a design unit.

Q9. Why NAND and NOR are called universal gates?
Ans. Because all the basic gates can be derive from them.

Q10. How many architectures are present in VHDL?
Ans. .behavior, dataflow, structural and mixed.

EXPERIMENT No. 1

Aim:- Design of Half adder, Full adder, Half Subtractor, Full Subtractor.

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

Following is the logic table and circuit diagram for half adder:

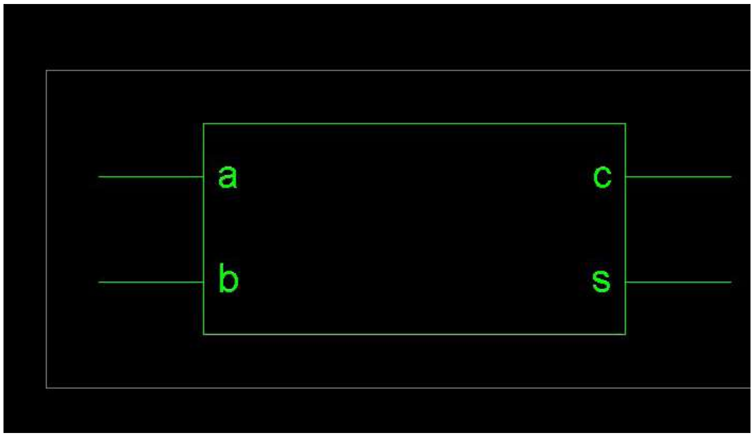
Inputs		Outputs	
A	***B***	***C***	***S***
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Program:

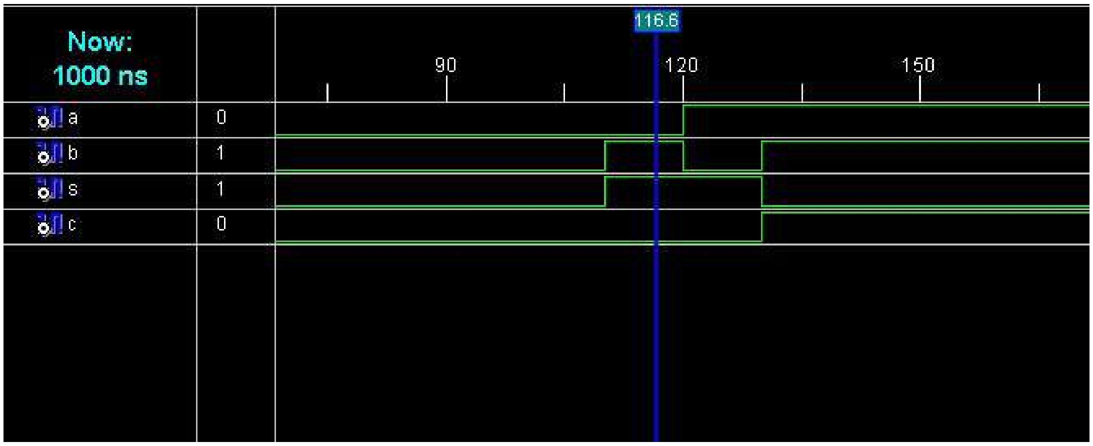
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL; use
IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.V Components.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:

RTL View



Simulation Waveform



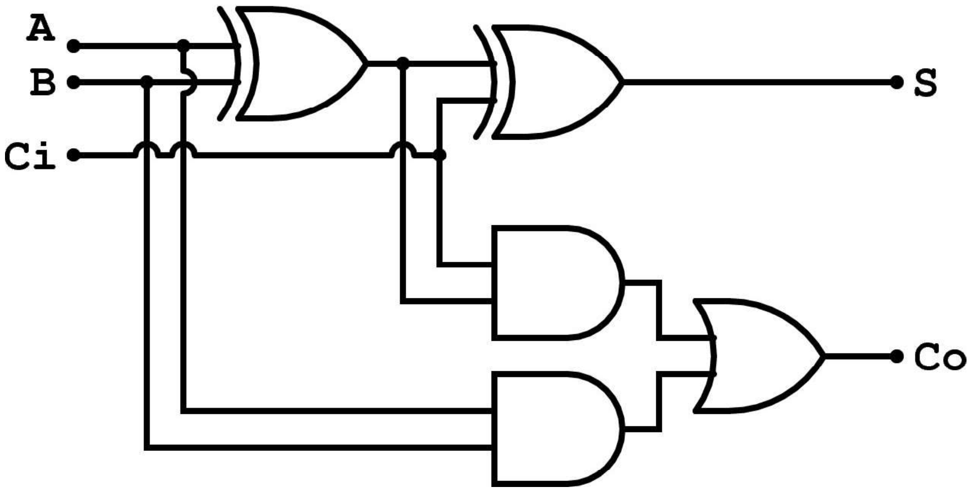
Full adder

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:

Input			Outputs	
A	***B***	C_i	C_o	***S***
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

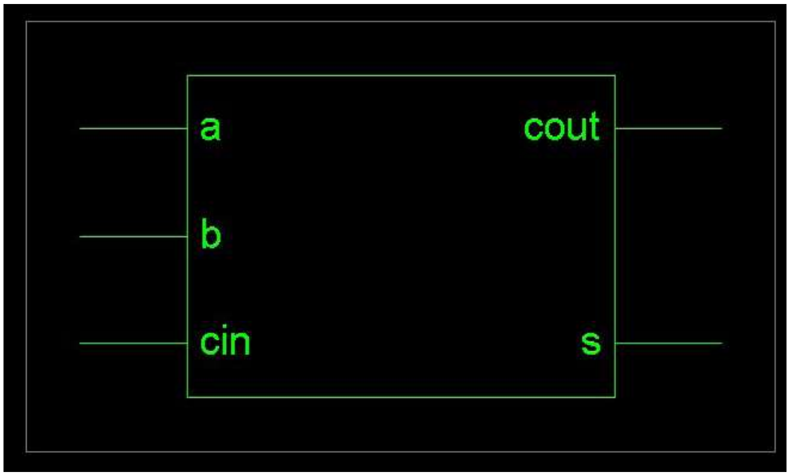


Program:

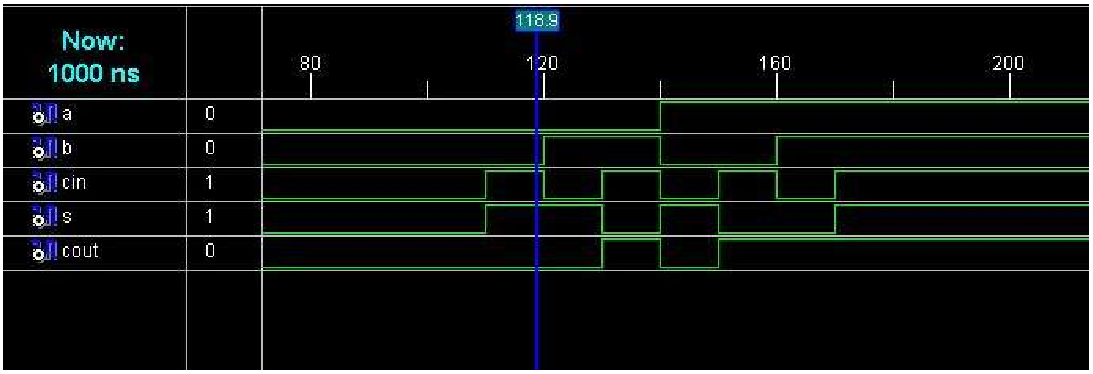
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL; use
IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- 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



Simulation Waveform



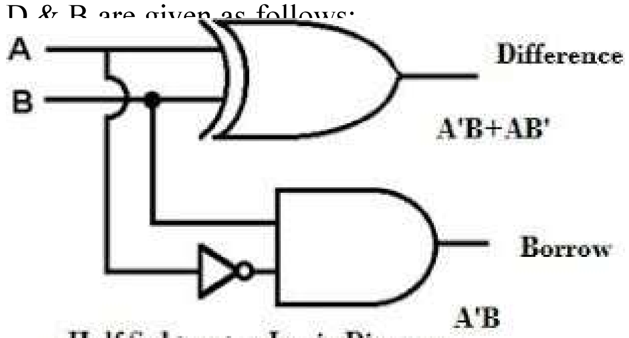
Half Subtractor

A **Half Subtractor** is a logical circuit that performs subtraction operation on two one- bit binary numbers often written as *A* and *B*. Difference and borrow is denoted by D & B. The truth table and circuit diagram for half subtractor are as follows:

The Boolean expressions for D & B are given as follows:

D = A’B + AB’ B = A’B

The designing is as follows:

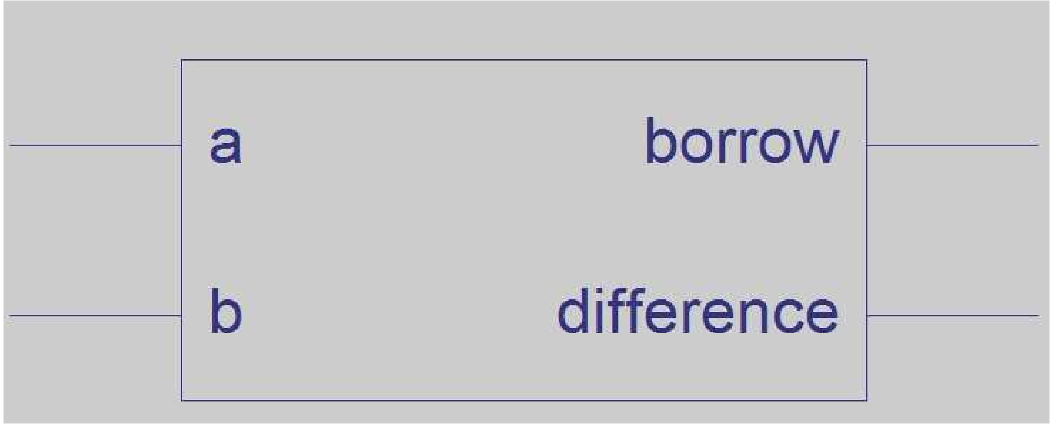


Half Subtractor Logic Diagram

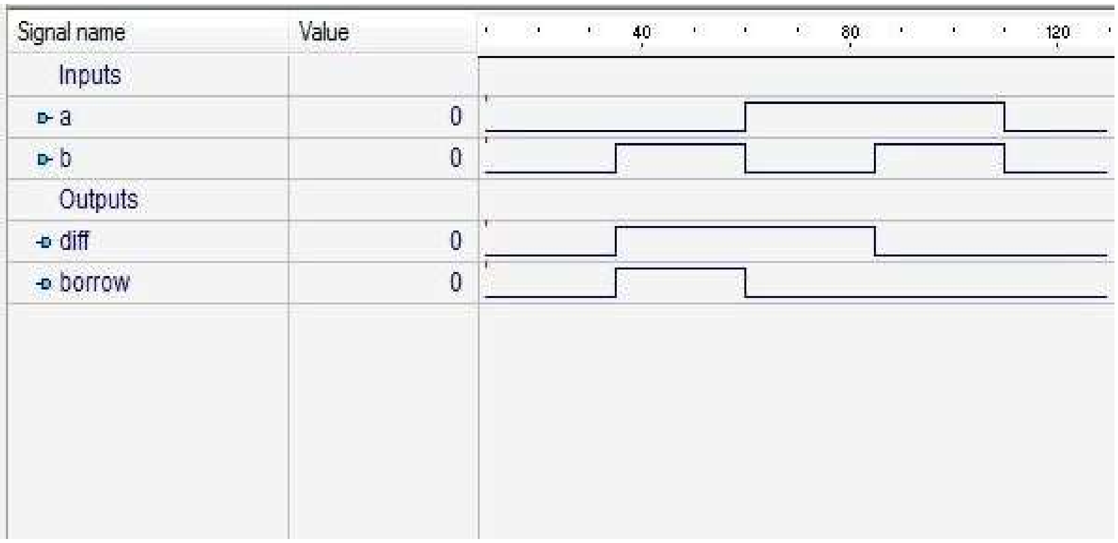
Program:

```
-------------------------------------------------------------------------------------
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 hs is
   Port ( a : in STD_LOGIC; b : in STD_LOGIC;
          difference : out STD_LOGIC; borrow : out
          STD_LOGIC);
end hs;

architecture Behavioral of hs is begin
difference<= a xor b;
borrow<= (not a) and b; end Behavioral;
```

RTL view



Simulation waveforms



Signal name	Value
Inputs	
a	0
b	0
Outputs	
diff	0
borrow	0

Full Subtractor

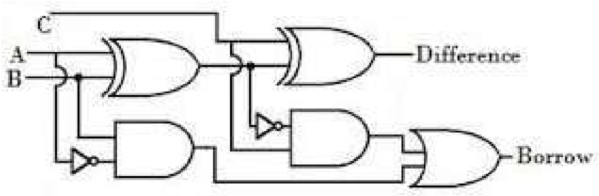
A logic circuit which is used for subtracting three single bit binary numbers is known as full subtractor. The truth table of full subtractor is shown below:

S = A’B’C + AB’C’ + A’BC’ + ABC andB = A’C + A’B + BC

The truth table for full subtractor is shown below:

Inputs			**Outputs**	
A	***B***	***C***	***D***	***B***
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	0	0	1
0	0	1	1	0
1	0	1	0	0
0	1	1	0	0
1	1	1	1	1

Truth table & circuit diagram for full subtractor are given below:



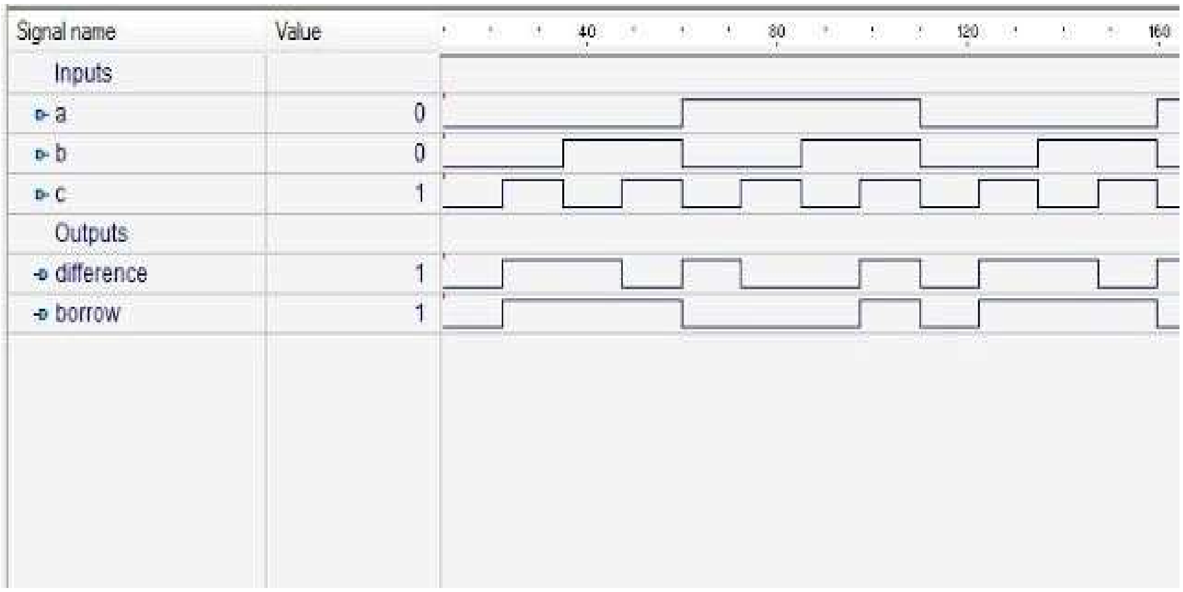
Full Subtractor-Logic Diagram

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 hs is
    Port ( a : in STD_LOGIC; b : in STD_LOGIC;
           difference : out STD_LOGIC; borrow : out
           STD_LOGIC);
end hs;

architecture Behavioral of hs is
```

```
begin
difference<= a xor b; borrow<= (not a) and b;

end Behavioral;
```

Simulation waveforms



Signal name	Value
Inputs	
a	0
b	0
c	1
Outputs	
difference	1
borrow	1

Quiz Questions with answer.

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

Ans. John Hines, Wright Patterson AFB, Daton Ohio.

1)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.3How many inputs and output are used in Full adder? Ans. Three inputs and

two output.

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.

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 ckts.?

Ans: Seq ckts 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.

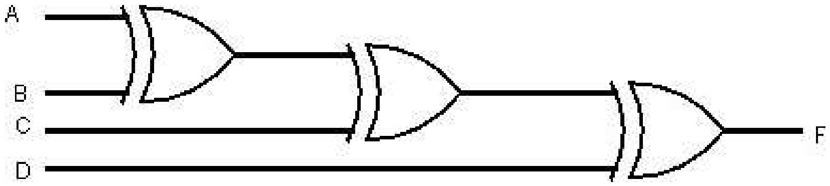
EXPERIMENT No. 2

Aim:- Design a parity generator.

Parity Generator

A **parity generator** is a combinational circuit which analyse two or more than two bits and tells about the parity of the circuit whether it is odd parity or even parity. It is very important for finding any error ,if occurs, while sending the data bits.

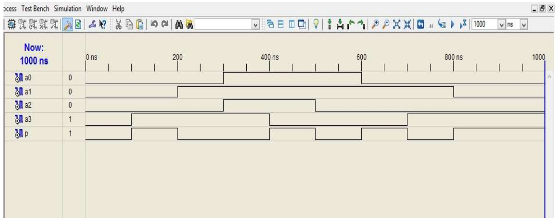
Given below is the circuit diagram for 4- bit parity checker:



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 parity is
   Port ( a0 : in STD_LOGIC; a1 : in STD_LOGIC;
          a2 : in STD_LOGIC; a3 : in STD_LOGIC;
          p : out STD_LOGIC); end parity;

architecture Behavioral of parity is signal r,s : std_logic;
begin -- parity_checker_ar

  r <= a0 xor a1; s <= a2 xor r; p <= s
  xor a3;
  end Behavioral; RTL View
```

Simulation Waveforms



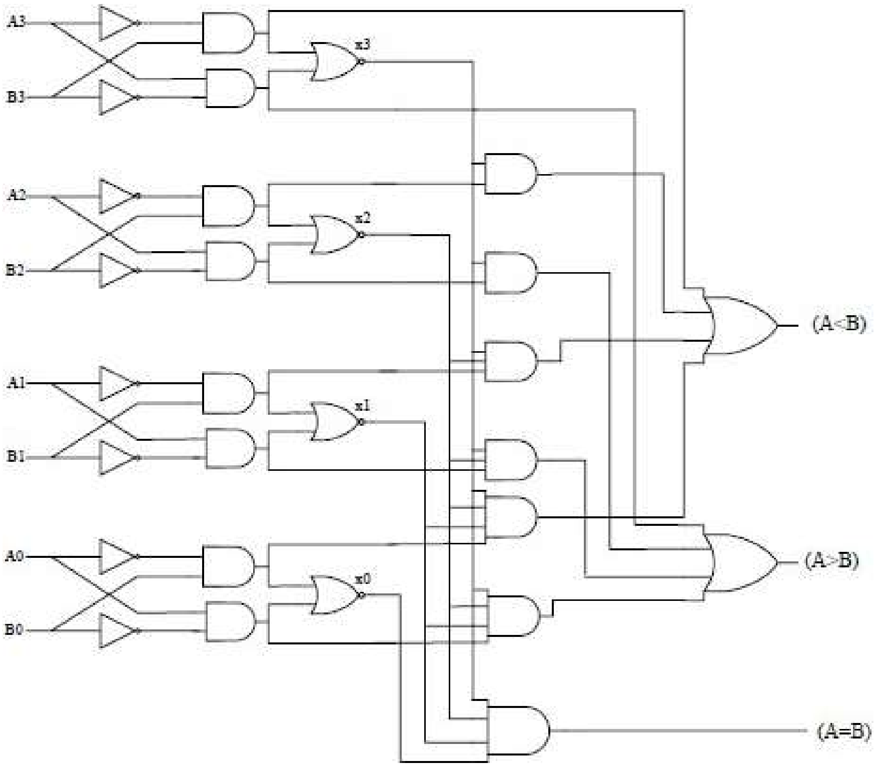
EXPERIMENT No. 3

Aim : Design a 4-bit comparator 4 bit Comparator

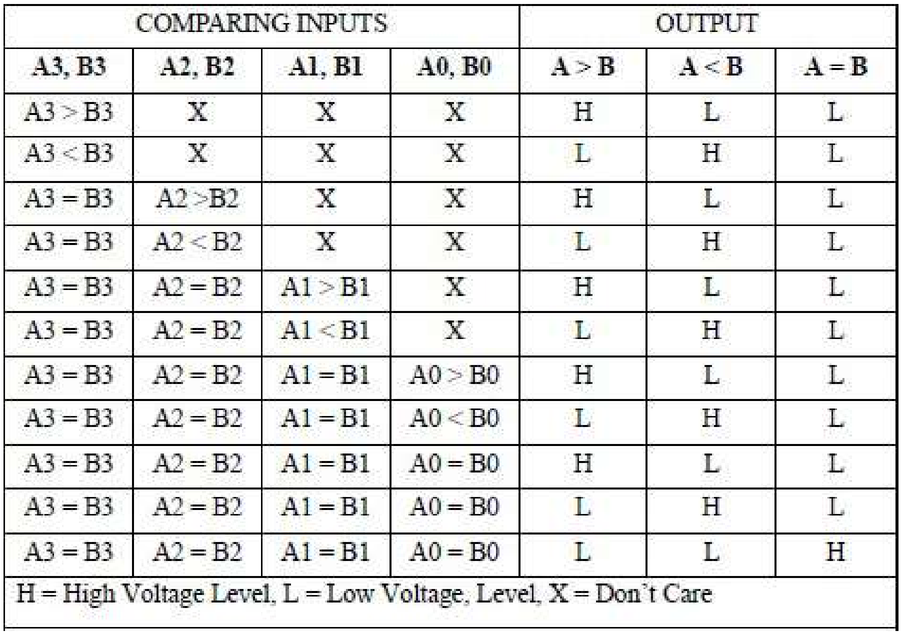
Magnitude comparator is a combinational circuit that compares two numbers anddetermines their relative magnitude.

The procedure for binary numbers with more than 2 bits can also be found in the similarway.

The figure shows the 4-bit magnitude comparator.



The truth table for 4-bit comparator is :

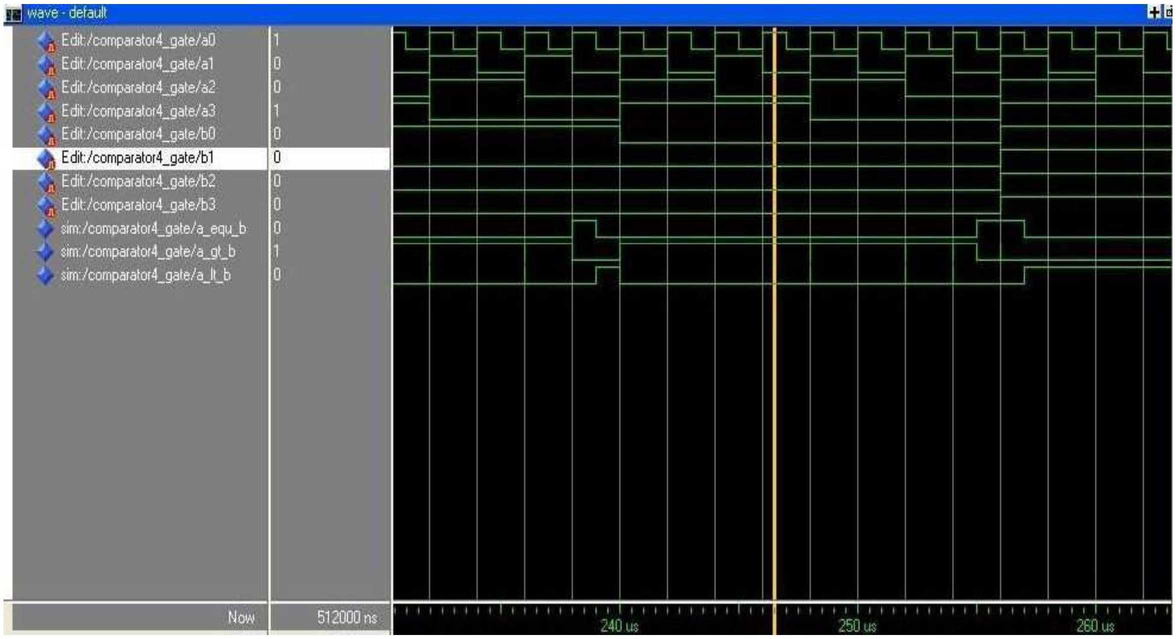


COMPARING INPUTS				OUTPUT		
A3, B3	**A2, B2**	**A1, B1**	**A0, B0**	**A > B**	**A < B**	**A = B**
A3 > B3	X	X	X	H	L	L
A3 < B3	X	X	X	L	H	L
A3 = B3	A2 >B2	X	X	H	L	L
A3 = B3	A2 < B2	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H
H = High Voltage Level, L = Low Voltage, Level, X = Don't Care						

```
Program library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity comparator_4bit is port(
      a   :   in   STD_LOGIC_VECTOR(3   downto   0);   b   :   in
      STD_LOGIC_VECTOR(3 downto 0); equal : out STD_LOGIC;
      greater  :  out  STD_LOGIC;  lower  :  out
      STD_LOGIC
      );
end comparator_4bit;

architecture comparator_4bit_arc of comparator_4bit is begin

   comparator : process (a,b) is begin
      if (a=b) then equal <= '1';
                                  greater <= '0';
           lower <= '0'; elsif (a<b) then equal
                                        <= '0';
                                  greater <= '0';
```

```
            lower <= '1'; else
            equal <= '0';
            greater <= '1';
            lower <= '0'; end if;
    end process comparator; end
comparator_4bit_arc;
```

Simulation Waveforms



```
Edit:/comparator4_gate/a0        1
Edit:/comparator4_gate/a1        0
Edit:/comparator4_gate/a2        0
Edit:/comparator4_gate/a3        1
Edit:/comparator4_gate/b0        0
Edit:/comparator4_gate/b1        0
Edit:/comparator4_gate/b2        0
Edit:/comparator4_gate/b3        0
sim:/comparator4_gate/a_equ_b    0
sim:/comparator4_gate/a_gt_b     1
sim:/comparator4_gate/a_lt_b     0
                       Now       512000 ns
```

QUIZ questions

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.
One OR gate to OR CD and EF and next to OR of G & output of first OR gate.

2. Which device converts BCD to Seven Segment ?

Ans. A device which converts BCD to Seven Segment is called DECODER.

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.

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.

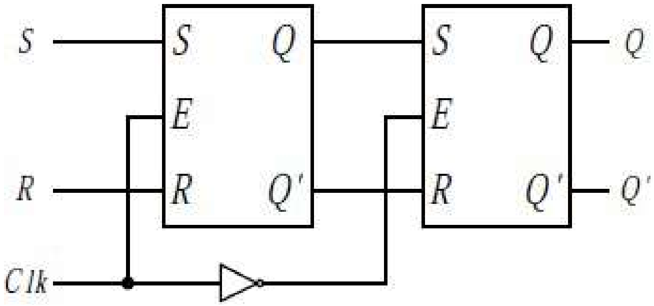
EXPERIMENT No. 4

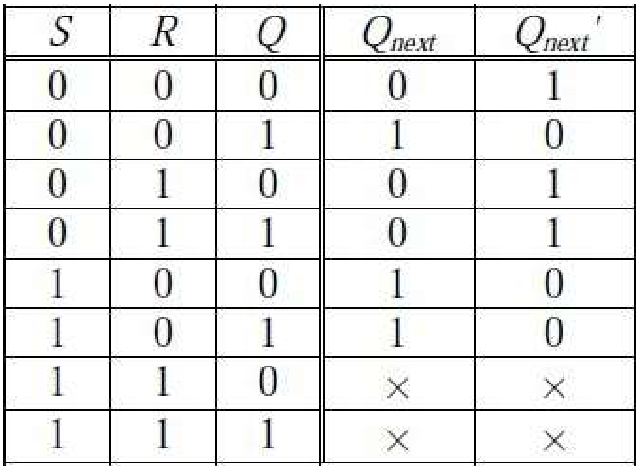
Aim : Design a RS & JK flip-flop

RS & JK FLIP-FLOP

RS flip-flops are useful in control applications where we want to be able to set or reset the data bit. However, unlike SR latches, SR flip-flops change their content only at the active edge of the clock signal. Similar to SR latches, SR flip-flops can enter an undefined state when both inputs are asserted simultaneously.

The truth table and circuit diagram are as follows:





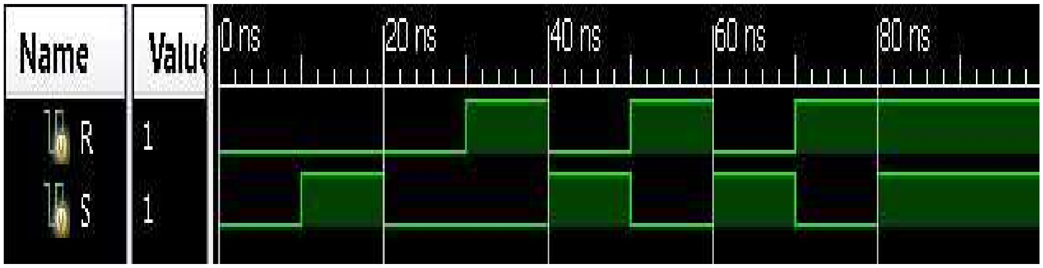
S	R	Q	Q_{next}	Q_{next}'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	×	×
1	1	1	×	×

Program:

```
                  -------------------------------------------------------------------
library ieee;
use ieee. std_logic_1164.all; use ieee.
std_logic_arith.all;
use ieee. std_logic_unsigned.all;
```

```
entity SR-FF is
PORT( S,R,CLOCK,CLR,PRESET: in std_logic;
                                          Q, QBAR: out std_logic);
end SR-FF;

Architecture behavioral of SR-FF is begin
P1: PROCESS(CLOCK,CLR,PRESET)
variable x: std_logic; begin
if(CLR='0') then x:='0';

elsif(PRESET='0')then x:='1';

elsif(CLOCK='1' and CLOCK'EVENT) then if(S='0' and R='0')then
 x:=x;
 elsif(S='1' and R='1')then x:='Z';

elsif(S='0' and R='1')then x:='0';

else x:='1';

end if; end if;

   Q<=x;
   QBAR<=not x; end PROCESS;
end behavioral;
```

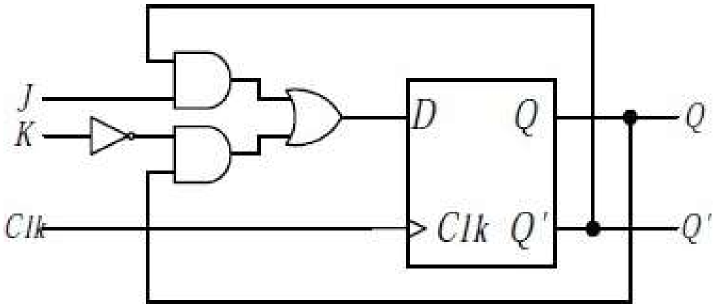
Simulation Waveforms

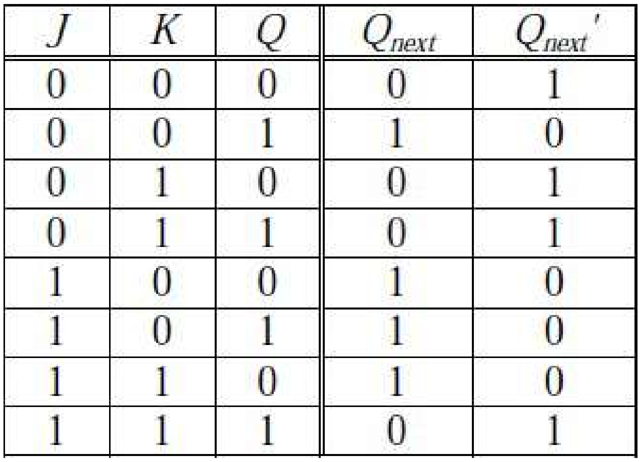


JK FLIP-FLOP

JK flip-flops are very similar to SR flip-flops. The J input is just like the *S* input in that when asserted,it sets the flip-flop. Similarly, the K input is like the R input where it clears the flip-flop when asserted. The only difference is when both inputs are asserted. For the SR flip-flop, the next state is undefined, whereas, for the JK flip-flop, the next state is the inverse of the current state. In other words, the JK flip-flop toggles its state when both inputs are asserted.

The truth table and circuit diagram are drawn below:





J	K	Q	Q_{next}	Q_{next}'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Program:

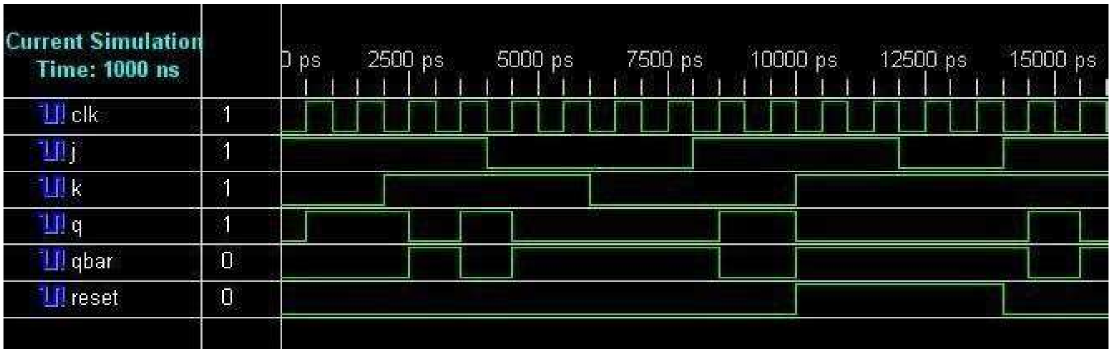
```
library ieee;--------------------------------------------------------------------------------
use ieee. std_logic_1164.all; use ieee.
std_logic_arith.all;
use ieee. std_logic_unsigned.all;

entity JK-FF is
PORT( J,K,CLK,PRST,CLR: in std_logic;
                        Q, QB: out std_logic);
end JK-FF;
```

```
Architecture behavioral of JK-FF is begin
P1: PROCESS(CLK,CLR,PRST)
variable x: std_logic; begin
if(CLR='0') then x:='0';

elsif(PRST='0')then x:='1';

elsif(CLK='1' and CLK'EVENT) then if(J='0' and K='0')then
       x:=x;
   elsif(J='1' and K='1')then x:= not x;

   elsif(J='0' and K='1')then x:='0';
   else
       x:='1';

   end if; end if;
   Q<=x;
   QB<=not x; end PROCESS;
end behavioral;
```

Simulation Waveforms

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. 2The 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.4The 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? Ans; .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. 5

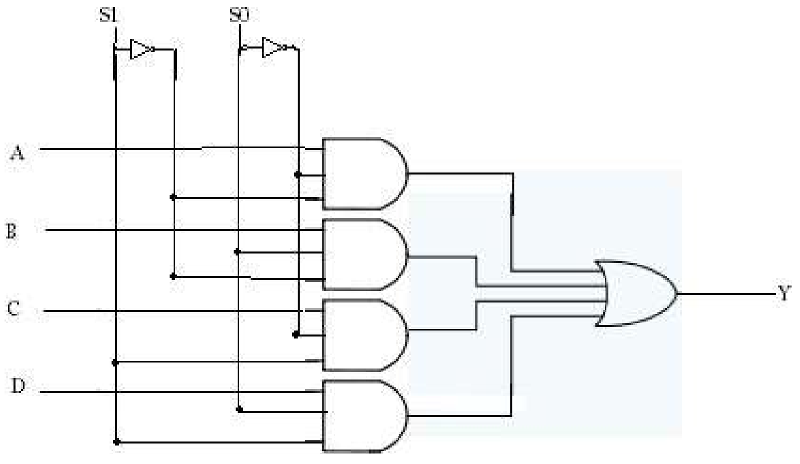
Aim : Design a 4:1 Multiplexer

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 $[\log_2(n)]$ 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, 1and S0 are the select lines, and Y is the output:

S1	S0	Y
0	0	A
0	1	B
1	0	C
1	1	D

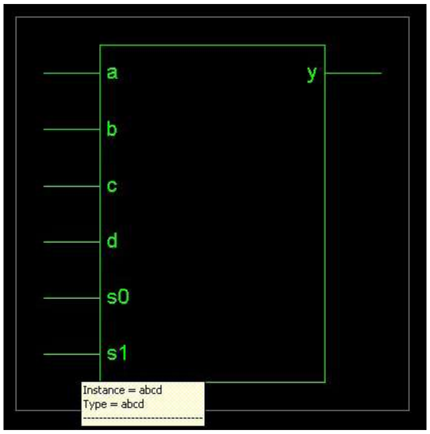


Program:

```
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;
```

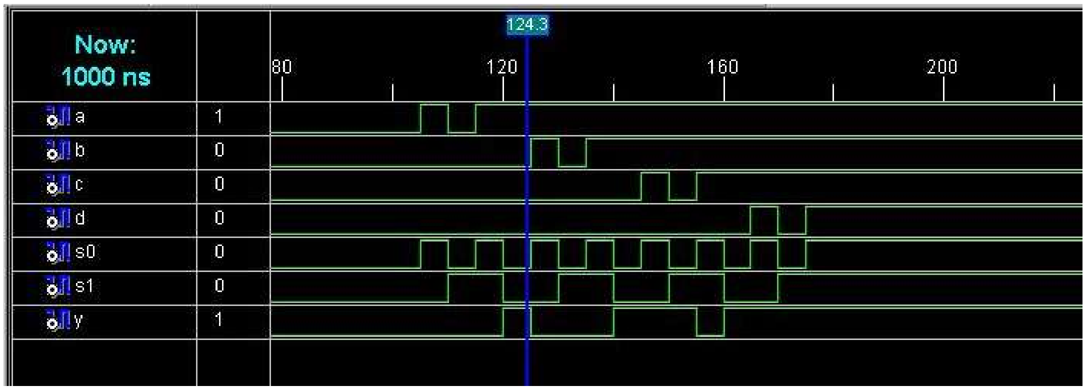
OUTPUT:

RTL View



Instance = abcd
Type = abcd

Simulation Waveform



Quiz Questions with answer.

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

Ans: Demultiplexer

2. What is the another name of Multiplexer

3.. Ans. Data Selector.

4. How many control lines will be used for a 8 – to – 1 multiplexer?

5. Ans. The number of control lines for an 8 to 1 Multiplexer is 3.

6. Which device changes serial data to parallel data .

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

7. How many select lines will a 16 to 1 multiplexer will have?

8. Ans. 4

Q6. Is it possible to construct 4:1 mux using two 2:1 mux?

Ans. Yes

Q7. How many outputs are there in 1:8 mux?

Ans 8.

EXPERIMENT No. 6

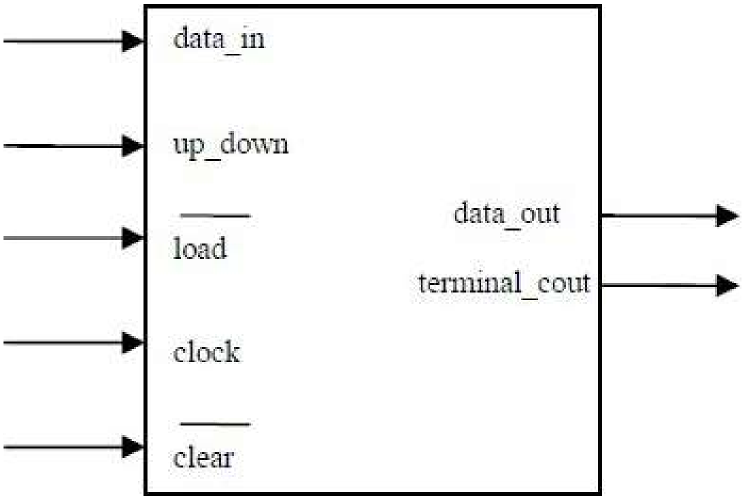
Aim:- Design a 4-bit Up/Down Counter with Loadable Count

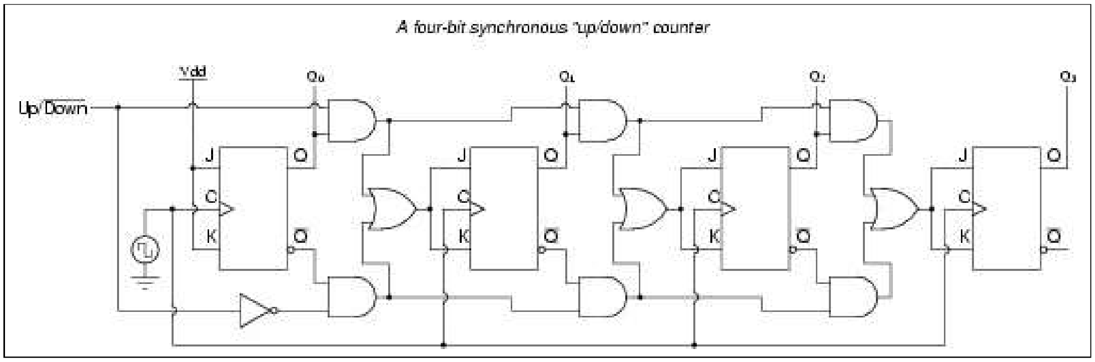
4-bit up/down Counter with loadable count

A counter is a sequential circuit which is used to count clock pulses. In other words, a counter is a sequential machine constructed with the help of flip-flops & changes its state according to state diagram.

A 4-bit Up/Down counter counts the clock pulses in ascending as well as descending order and recycle again from its initial value.

A graphical schematic for a 4-bit up/down counter is depicted in the given figure.

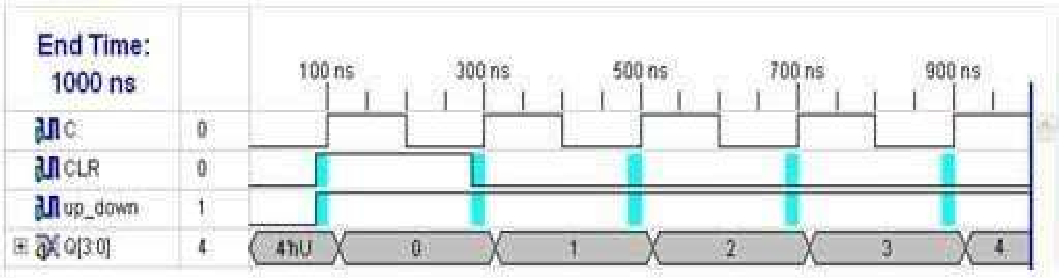



A four-bit synchronous "up/down" counter

Program:

```
ENTITY counter_4bit IS PORT(
data_in: IN std_logic_vector(3 downto 0);
clock: IN std_logic; load: IN std_logic; clear:
IN std_logic; up_down: IN std_logic;
terminal_count: OUT std_logic; data_out: OUT
std_logic_vector (3 downto 0));
END counter_4bit;

ARCHITECTURE counter_4bit_arh OF counter_4bit IS
SIGNAL
count:std_logic_vector(3 downto 0)
:="0000"; BEGIN
PROCESS (clock) BEGIN IF (clear = '0') THEN
count <= "0000"; ELSIF(load = '0') THEN
count <= data_in; ELSE
IF (clock'EVENT AND clock = '0') AND(clock'LAST_VALUE
= '1') THEN
IF(up_down = '1') THEN count <= count + 1;
END IF;
IF(up_down = '0') THEN count <= count - 1;
END IF; END IF; END IF;
IF (count = "1111") THEN
terminal_count <= '1'; ELSE
terminal_count <= '0'; END IF;
data_out <= count; END PROCESS;
END counter_4bit_arh;
```

Simulation Waveforms



Q.1What 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.
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.
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.
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.
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. 7

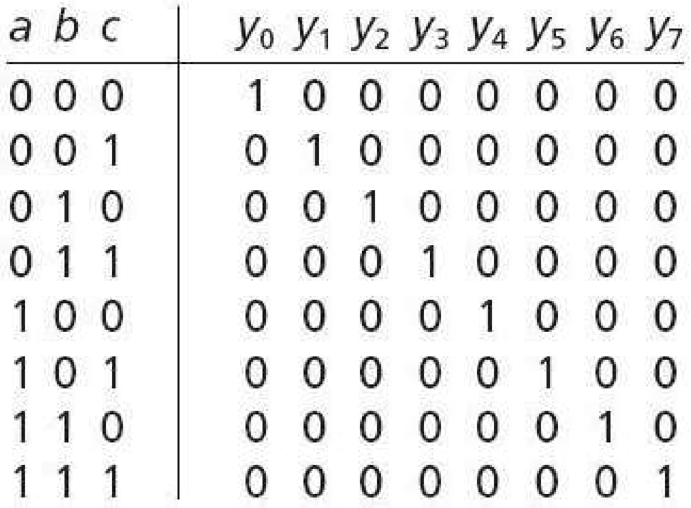
Aim:- To Design a 3:8 Decoder using VHDL

Decoder:

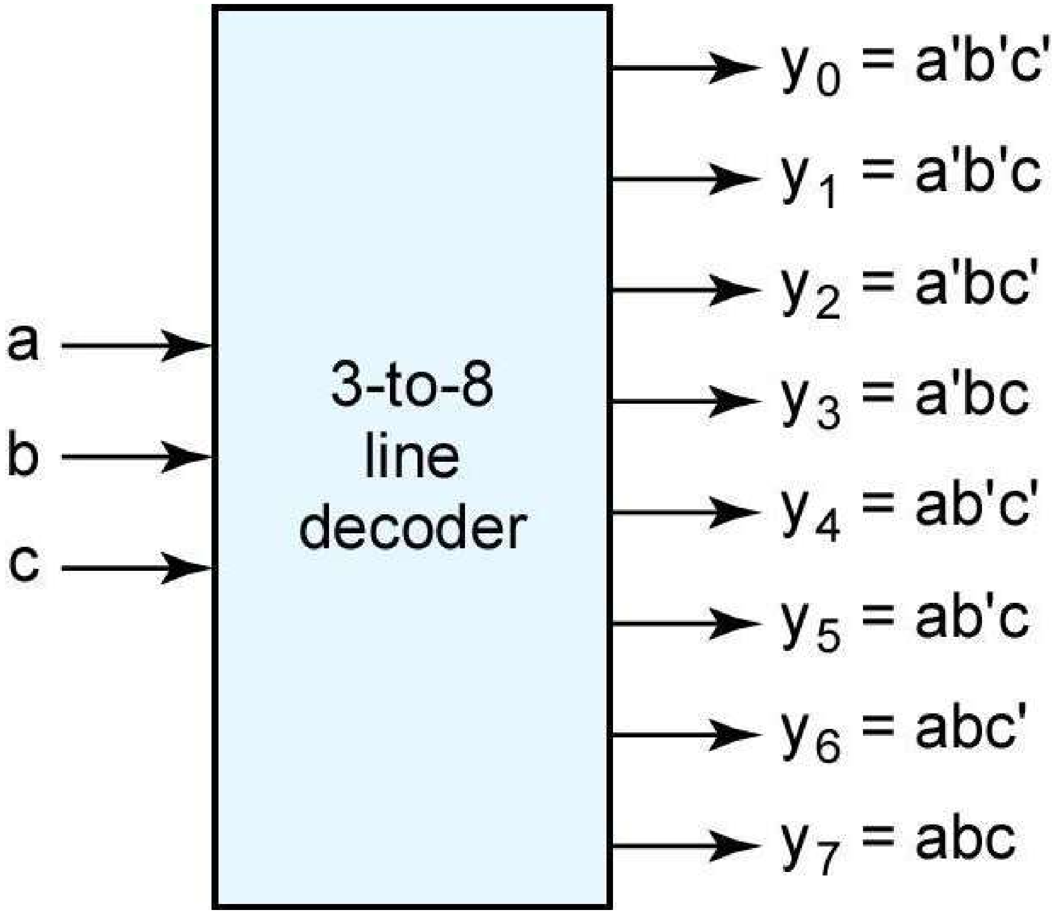
A **decoder** is a device which does the reverse of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode.

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.

The truth table for 3:8 decoder and respective circuit diagram is as follows:



a	b	c	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

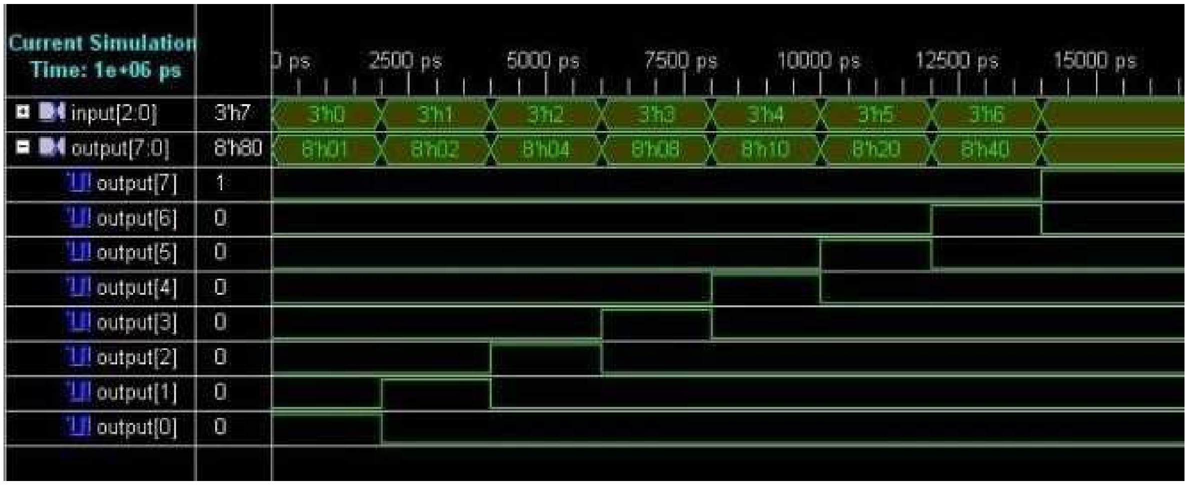

a
b
c
3-to-8
line
decoder
$y_0 = a'b'c'$
$y_1 = a'b'c$
$y_2 = a'bc'$
$y_3 = a'bc$
$y_4 = ab'c'$
$y_5 = ab'c$
$y_6 = abc'$
$y_7 = abc$

Program:

```
library ieee;


    use ieee.std_logic_1164.all;

entity decoder_using_case is port (
        enable      :in std_logic;                    -- Enable for the decoder binary_in :in
        std_logic_vector (2 downto 0); -- 3-bit Input decoder_out :out std_logic_vector (7
        downto 0) -- 8-bit Output);
end entity;

architecture behavior of decoder_using_case is begin
    process (enable, binary_in) begin
        decoder_out <= X"000"; if (enable = '1') then
            case (binary_in) is
                when X"0" => decoder_out <= X"000"; when X"1" =>
                decoder_out <= X"001"; when X"2" => decoder_out <=
                X"010"; when X"3" => decoder_out <= X"011"; when X"4" =>
                decoder_out <= X"100"; when X"5" => decoder_out <=
                X"101"; when X"6" => decoder_out <= X"110"; when X"7" =>
                decoder_out <= X"111";
            end case; end if;
    end process; end architecture;
```

OUTPUT:

SimulationWaveform



Quiz Questions with answer.

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.

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 = 2

One OR gate to OR CD and EF and next to OR of G & output of first OR gate.

3. Which device converts BCD to Seven Segment ?

Ans. A device which converts BCD to Seven Segment is called DECODER.

4. 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.

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

EXPERIMENT No. 8

Aim:- To Design a 8 bit shift register

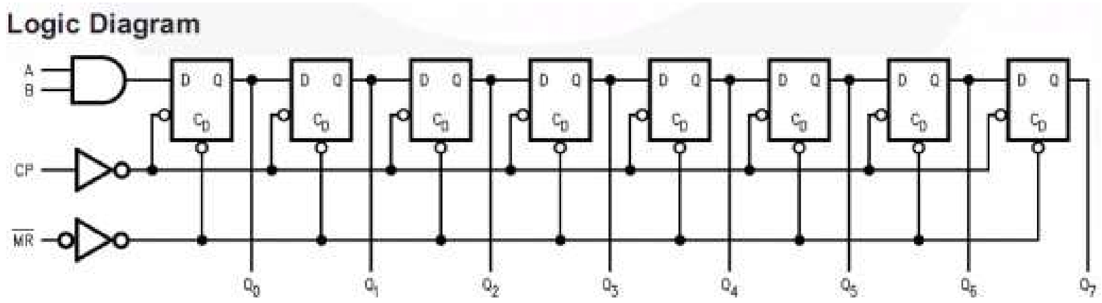
8. bit Shift Register

Shift Register is a type of sequential circuit formed by combination of flip-flops and is capable of shifting data from left to right or vice-versa. Shift register basically performs two functions:

i. Shifting of data(Transfer of data)

ii. Storage function

The circuit diagram for 8-bit Shift Register is given as:



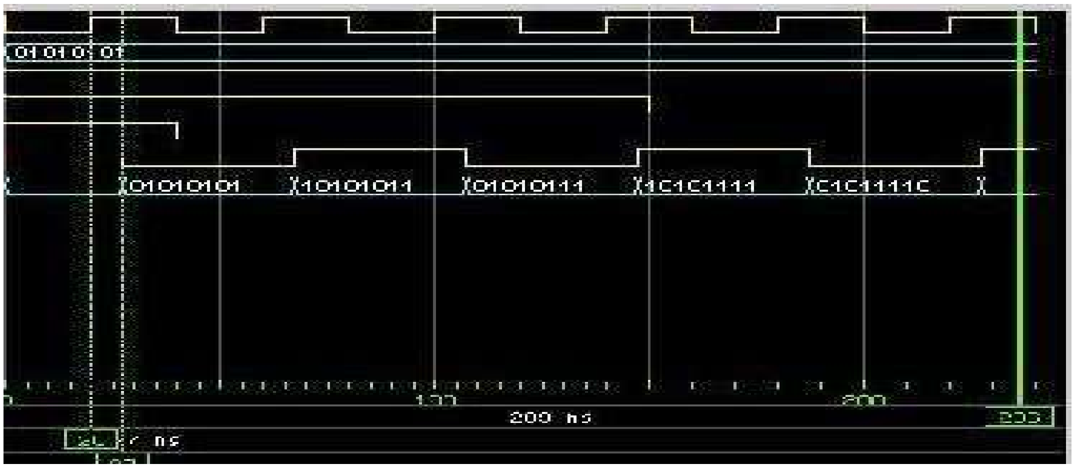
The vhdl program for 8-bit shift-left register with a positive-edge clock, serial in, and serial out.

Program:

```
library ieee;
use ieee.std_logic_1164.all;

entity shift is
port(C, SI : in std_logic; SO : out std_logic); end shift;

architecture archi of shift is
signal tmp: std_logic_vector(7 downto 0);

begin process (C) begin
if (C'event and C='1') then for i in 0 to 6 loop
```

```
tmp(i+1) <= tmp(i); end loop;
tmp(0) <= SI;
end if;
end process;

SO <= tmp(7);
end archi;
```

Simulation Waveforms



Quiz Questions with answer.

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.

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.

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.

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.

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. 9

Aim:- To Design an arithmetic unit

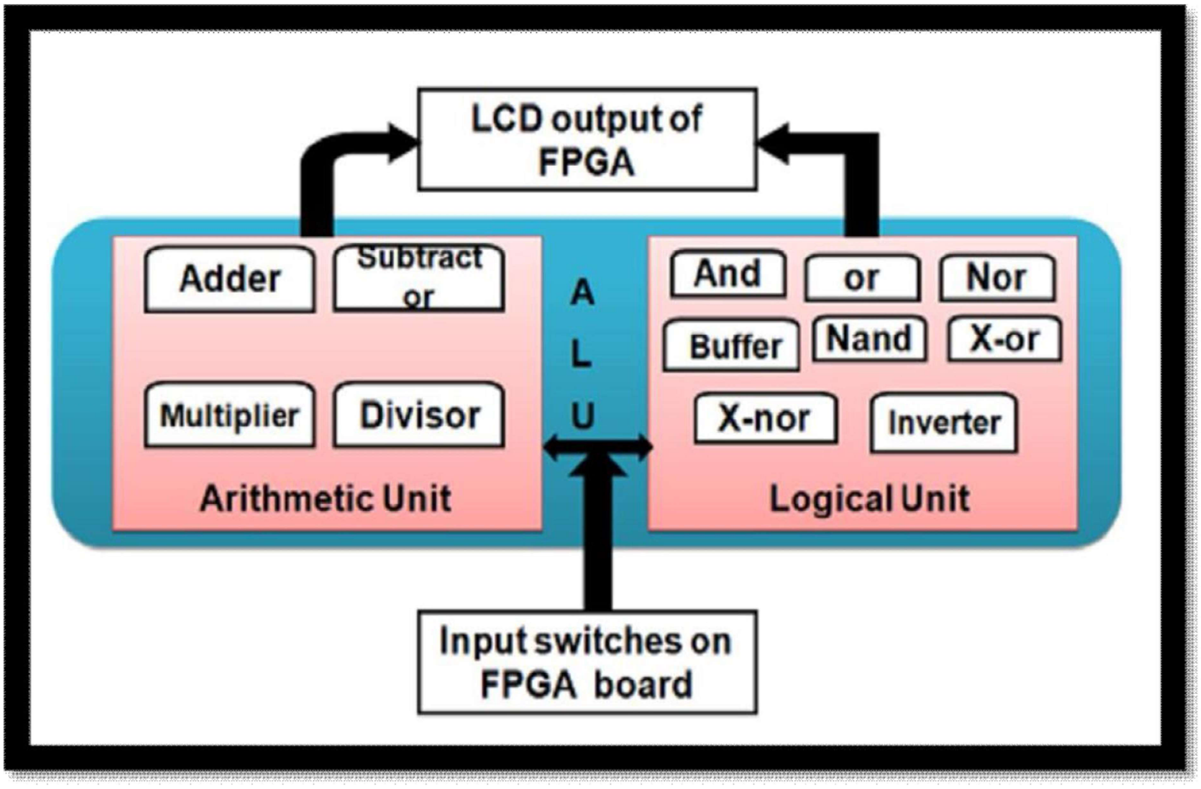
4-bit Arithmetic Logic Unit

The design and implementation of FPGA based Arithmetic Logic Unit is of core significance in digital technologies as it is an integral part of central processing unit. ALU is capable of calculating the results of a wide variety of basic arithmetical and logical computations. The ALU takes, as input, the data to be operated on (called operands) and a code, from the control unit, indicating which operation to perform. The output is the result of the computation. Designed ALU will perform the following operations:

- Arithmetic operations
- Bitwise logic operations

All the modules described in the design are coded using VHDL which is a very useful tool with its degree of concurrency to cope with the parallelism of digital hardware.

The block diagram for ALU is shown below:

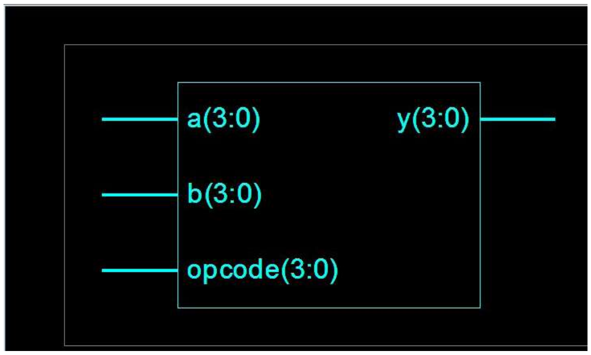


There are two kinds of operation which an ALU can perform first part deals with arithmetic computations and is referred to as Arithmetic Unit. It is capable of addition, subtraction, multiplication, division, increment and decrement. The second part deals with the Gated results in the shape of AND, OR, XOR, inverter, rotate, left shift and right shift, which is referred to as Logic Unit. The functions are controlled and executed by selecting operation or control bits.

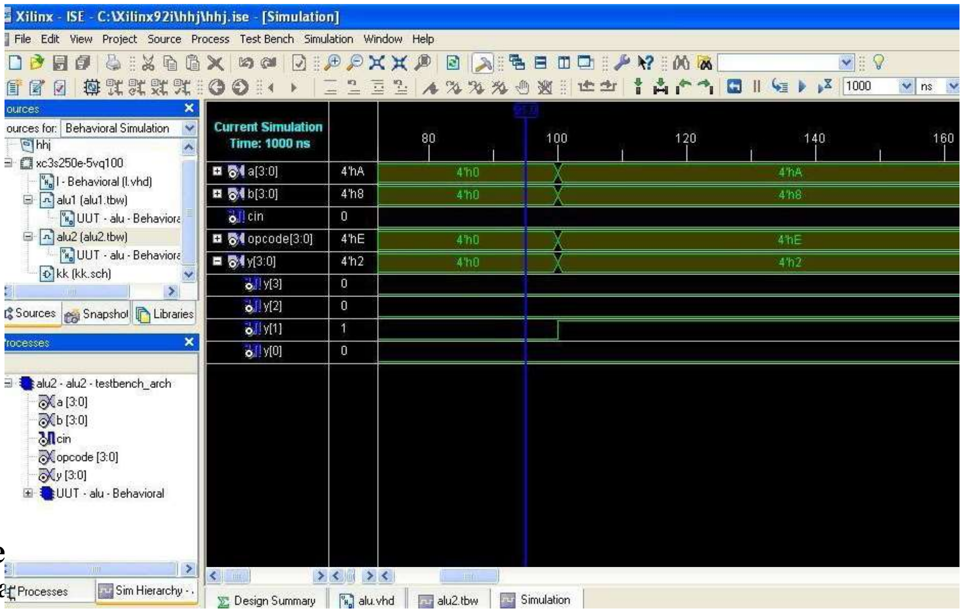
Program:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL; use
IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity alu is
Port ( a : in STD_LOGIC_VECTOR (03
         downto 0);
         b : in STD_LOGIC_VECTOR (03
         downto 0);
         opcode : in STD_LOGIC_VECTOR (03 downto 0);
         y : out STD_LOGIC_VECTOR (03
         downto 0)); end alu;

Architecture Behavioral of alu is begin
with opcode (3 downto 0) select y<= a when "0000",
(not a) when "0001", b when "0010",
(not b) when "0011", a and b when "0100",
a or b when "0101",
a nand b when "0110", a nor b when "0111",
a xor b when "1000", a+1 when"1001",
b+1 when "1010",
a+b when "1011",
a-1 when "1100",
b-1 when "1101",
a-b when "1110",
a xnor b when "1111", "0000" when others;
end Behavioral;
```

RTL View



Simulation Waveforms



Quiz Que

1. Wha

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

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

3. How many bits are there in BCD code? Ans. 4

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.

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.

Q6. What is BCD to Gray converter?

Ans: The converter which converts bcd code into gray code

. Q7: What is the application of above code converter?

Ans We use in communication systems.

Q8. 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.

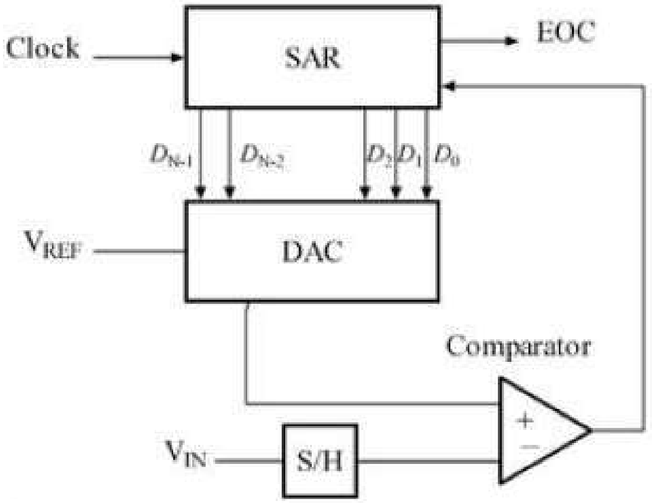
Ans; 5

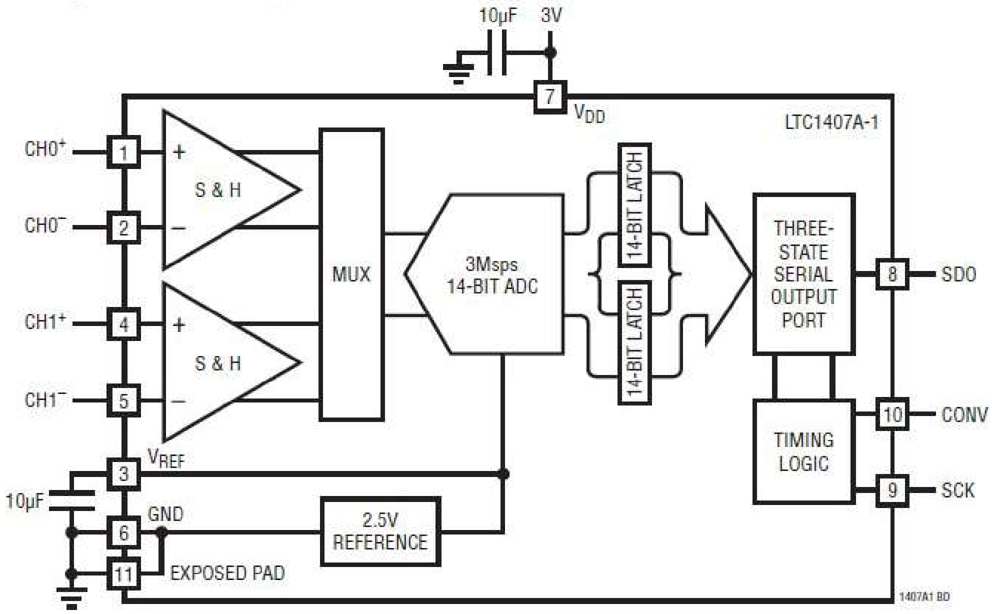
EXPERIMENT No. 10

Aim: implement ADC & DAC interface with FPGA

The Spartan 3E Starter Kit provides us the basic features as provided by the Spartan 3E FPGA. It also provides easy way to test various programs in the FPGA itself, by dumping the 'bit' file into the FPGA and then observing the output. The Spartan 3E FPGA board comes built in with many peripherals that help in the proper working of the board and also in interfacing the various signals to the board itself.

ADCs (Analog to Digital Convertor) are of various types. The one used for our purpose is the Successive Approximation Type ADC (SAR-ADC), where the main components include a DAC (digital to analog convertor), a clock, a comparator and a SAR register for storing the values of the digital data which comes after the comparator compares the values of the DAC with the analog input and outputs a '1' or a '0' depending on the condition.





Circuit Diagram of the LTC1407A ADC chip

Working of ADC

ADC presents a 14-bit, 2’s complement digital output of the analog input. The input voltage given to the ADC depends on the programmable gain settings of the pre- amplifier. The maximum input range is for the gain = -1 for which the input voltage range from 0.4V to 2.9V. The Analog to Digital Conversion formula is given below:

$$D[13:0] = GAIN \times \frac{V_{IN}-1.65V}{1.25V} \times 8192$$

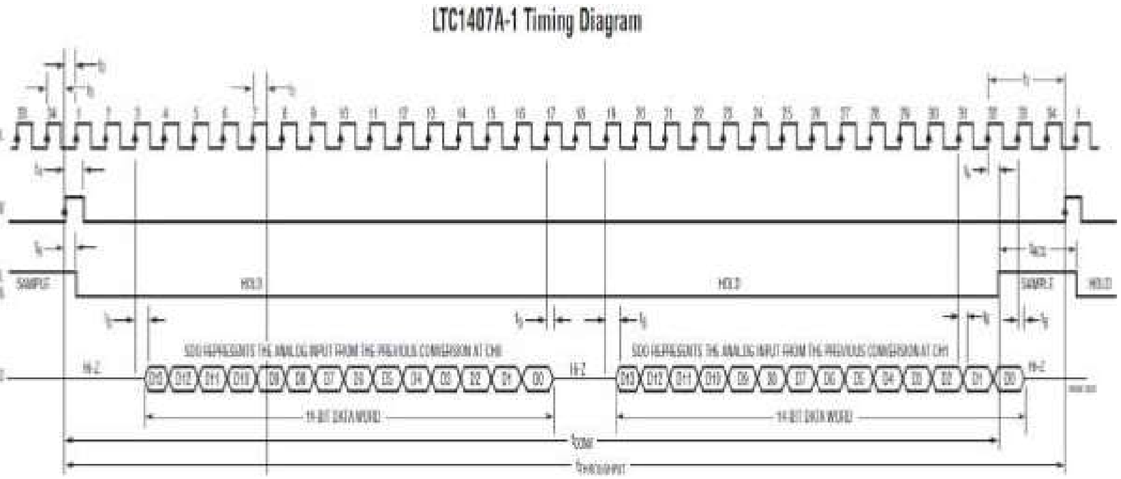
Here D [13:0] represents the 14 bit two’s complement value of the analog input. It is output to the FPGA from the ADC via the SPI_MISO signal, as will be discussed later in the interfacing signals of the FPGA and the ADC. GAIN is the gain setting given via the programming of the gain register bit by bit. VIN is the input voltage to the ADC. 1.65V is the reference voltage of the ADC. This is achieved by the voltage divider circuit provided in the ADC circuit (dividing the Vcc which is 3.3V). The range of the ADC used is ±1.25V. Hence the output is scaled by 1.25V. Also the output obtained is in 14 bit 2’s complement form and hence the output is scaled by 8192. Both the input channels [VIN(A) and VIN(B)] are sampled simultaneously.

Communication Between FPGA and ADC

o **AD_CONV:** This signal is active high shutdown and reset signal. This signal marks the beginning of the conversion of the analog signal. It is an internal signal of the FPGA board, which can’t be viewed with the help of an external oscilloscope. Pin P11 is responsible for this signal. This signal is directed from FPGA to ADC.

o **SPI_MISO:** This signal is the serial data output from the ADC chip to the FPGA board. It is the one that gives the digital representation of the sampled analog value as 14-bit 2’s complement binary value. It is again an internal signal and pin N10 is responsible for this signal. This signal is directed from FPGA to ADC.

o **SPI_SCK:** As described earlier, this is the clock signal which plays an important role in the analog to digital conversion process and also sending the data from the ADC unit to the FPGA.


LTC1407A-1 Timing Diagram
SCK
CONV
INTERNAL S/H STATUS
SAMPLE
HOLD
HOLD
SAMPLE
HOLD
SDO
Hi-Z
SDO REPRESENTS THE ANALOG INPUT FROM THE PREVIOUS CONVERSION AT CH0
D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
Hi-Z
SDO REPRESENTS THE ANALOG INPUT FROM THE PREVIOUS CONVERSION AT CH1
D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
Hi-Z
14-BIT DATA WORD
14-BIT DATA WORD
tCONV
tTHROUGHPUT
tACQ

Serial interface timing diagram for ADC conversion and sampling

The Spartan-3E Starter Kit board includes an SPI-compatible, four-channel, serial Digital to-Analog Converter (DAC). The DAC device is a Linear Technology LTC2624 quad DAC with 12-bit unsigned resolution.

SPI Communication

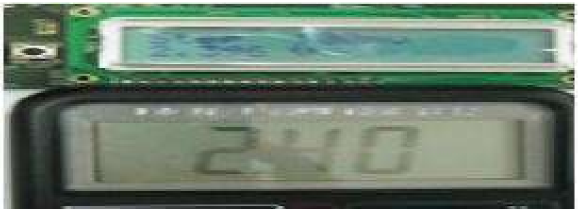
The FPGA uses a Serial Peripheral Interface (SPI) to send digital values to each of the four DAC channels. The SPI bus is a full-duplex, synchronous, character-oriented channel employing a simple four-wire interface. The interface signals between the FPGA and the DAC are the SPI_MOSI, SPI_MISO, and SPI_SCK which are shared with the other devices on the SPI bus. The DAC_CS signal is the active-Low slave select input to the DAC. The DAC_CLR signal is the active-Low, asynchronous reset input to the DAC. As a bus master the FPGA drives the bus clock signal (SPI_SCK) and transmits serial data (SPI_MOSI) to the selected DAC bus slave (Xilinx,2006).

WORKING PRINCIPLE

This system produces analog voltage at the output of DAC according to the digital value provided by the FPGA . The digital input of the DAC is controlled with the help of rotary switch. The rotation of switch in clockwise direction increase the analog output value and rotation in anticlockwise direction result in the decrement of the analog output value. The step voltage of increment is controlled with a push- button switch. Default step voltage is 100mv. With pressing this switch we can change the step from 100mv to 10mv. For the next push it switches to previous state.

RESULT ANALYSIS

The developed VHDL code has been implemented on the Spartan-3E starter kit. The design is synthesized and implemented on the FPGA by using the Xilinx ISE 13.2 software.



Conclusion

- The ADC was analyzed for a constant voltage supply.
- The onboard ADC of the Spartan 3E Starter Kit FPGA Board was properly interfaced with real world signals.

Quiz Questions with answer.

1. Who is the father of VHDL?

Ans. John Hines, Wright Patterson AFB, Daton Ohio.

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.

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

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.

5. Why HDL is used?

Ans. HDL is used because it is easy to design, implement, test and document increasingly complex digital system.

Q6. Give the basic rules for binary addition?

Ans. 0+0 = 0; 0+1 = 1 ; 1+1 = 1 0 ; 1+0 = 1.

Q7: What is the drawback of half adder?

Ans: We can’t add carry bit from previous stage.

Q8: 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.

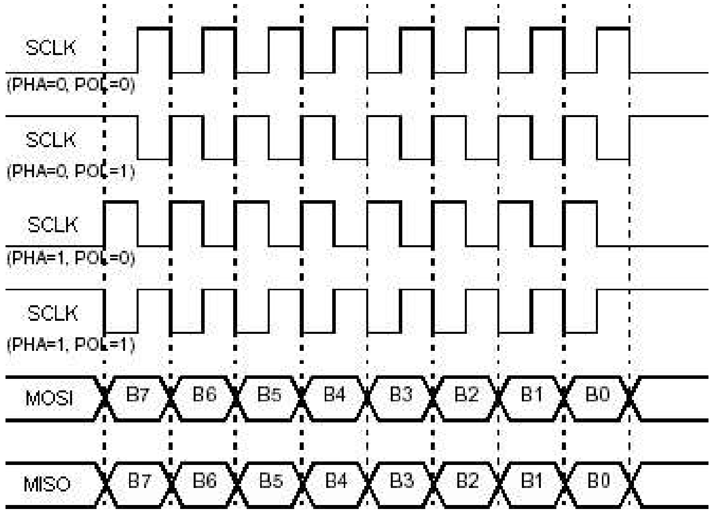
Q9: Define Nibble?

Ans. Combination of four bits

EXPERIMENT No. 11

Aim:-Implement a serial communication interface with FPGA

The Serial Peripheral Interface (SPI) is a high speed (up to 400 Mhz) synchronous serial interface/protocol designed by Motorola. It is a popular interface used for connecting peripherals to each other and to microprocessors. Most literature indicates that the interface can only be used for eight or sixteen bit block data transfers, but many Motorola microcontrollers allow transfers of any range of blocks between two and sixteen bits at a time. Because of the serial nature of the interface, data transfers of more than sixteen bits at a time can be implemented easily through control signals. There are four possibilities for clocking the data based on the clock polarity and the clock phase:



Conclusion: Complete coding of SPI Protocol using VHDL can be implemented on FPGA XC3S400.

Quiz Questions with answer.

1. Who is the father of VHDL?

Ans. John Hines, Wright Patterson AFB, Daton Ohio.

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.

3. How many inputs and output are used in Full adder?

Ans. Three inputs and two output.

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.

5. Why HDL is used?

Ans. HDL is used because it is easy to design, implement, test and document increasingly complex digital system.

Q6. Give the basic rules for binary addition? Ans. 0+0 = 0;
0+1 = 1 ; 1+1 = 1 0 ; 1+0 = 1.

Q7: What is the drawback of half adder?

Ans: We can’t add carry bit from previous stage.

Q8: 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. Q9: Define Nibble?

Ans. Combination of four bits