Course Name:	**Hardware Description Language Lab (2UXL401)**	**Semester:**	**IV**
Date of Performance:	**02 / 03 / 2021**	**Batch No:**	**B2**
Faculty Name:	**Prof.. Bhargavi Kaslikar**	**Roll No:**	**1912060**
Faculty Sign & Date:		**Grade/Marks:**	

Experiment No: 3

Title: Use of concurrent statements: Priority Encoders

Aim and Objective of the Experiment:

Write a VHDL code for implementing a 8:3 priority encoder with enable input
Write a testbench to verify your results.
Also, generate a programming file and download the code on CPLD kit and verify the results.

To study various types of concurrent statements in VHDL code and to understand use of test bench for simulation.

COs to be achieved:

CO 1: Use basic Concurrent and Sequential statements in VHDL and write codes for simple applications
CO 2: Test a VHDL code and verify the circuit model.
CO 3: Synthesize and Implement the designed circuits on CPLD/ FPGA.

Work to be uploaded

VHDL code for priority encoder.
Test bench for priority encoder and simulation waveform of the same.
Scanned copy of post lab questions

Code for Priority Encoder:

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

entity priority_encoder_nachiket is
port(
        a: in std_logic_vector(7 downto 0);
        enable: in std_logic;
        b: out std_logic_vector(2 downto 0)
    );
end entity;

architecture priority_encoder_nachiket_arch of priority_encoder_nachiket is

    signal b_s: std_logic_vector(2 downto 0);
    begin
        b_s<= "111" when a(7)='1' else
            "110" when a(6)='1' else
            "101" when a(5)='1' else
            "100" when a(4)='1' else
            "011" when a(3)='1' else
            "010" when a(2)='1' else
            "001" when a(1)='1' else
            "000";

        b <= b_s when enable='1' else
        "UUU";
end priority_encoder_nachiket_arch;
```

Testbench for Priority Encoder:

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

entity priority_encoder_nachiket_tb is
end entity;
architecture priority_encoder_nachiket_tb_arch of priority_encoder_nachiket_tb is
    component priority_encoder_nachiket is
        port(
            a: in std_logic_vector(7 downto 0);
            enable: in std_logic;
            b: out std_logic_vector(2 downto 0)
        );
    end component;
signal a_in: std_logic_vector(7 downto 0);
signal en: std_logic;
signal b_out: std_logic_vector(2 downto 0);
begin
    PE : priority_encoder_nachiket port map(a_in, en, b_out);

    process
        begin
            en <= '1';
            a_in<="00000000";
            wait for 10ns;

            a_in<="01010100";
            wait for 10ns;

            a_in<="10100100";
            wait for 10ns;

            a_in<="00000001";
            wait for 10ns;

            a_in<="00000010";
            wait for 10ns;

            a_in<="00011011";
            wait for 10ns;

            a_in<="00111011";
            wait for 10ns;
```

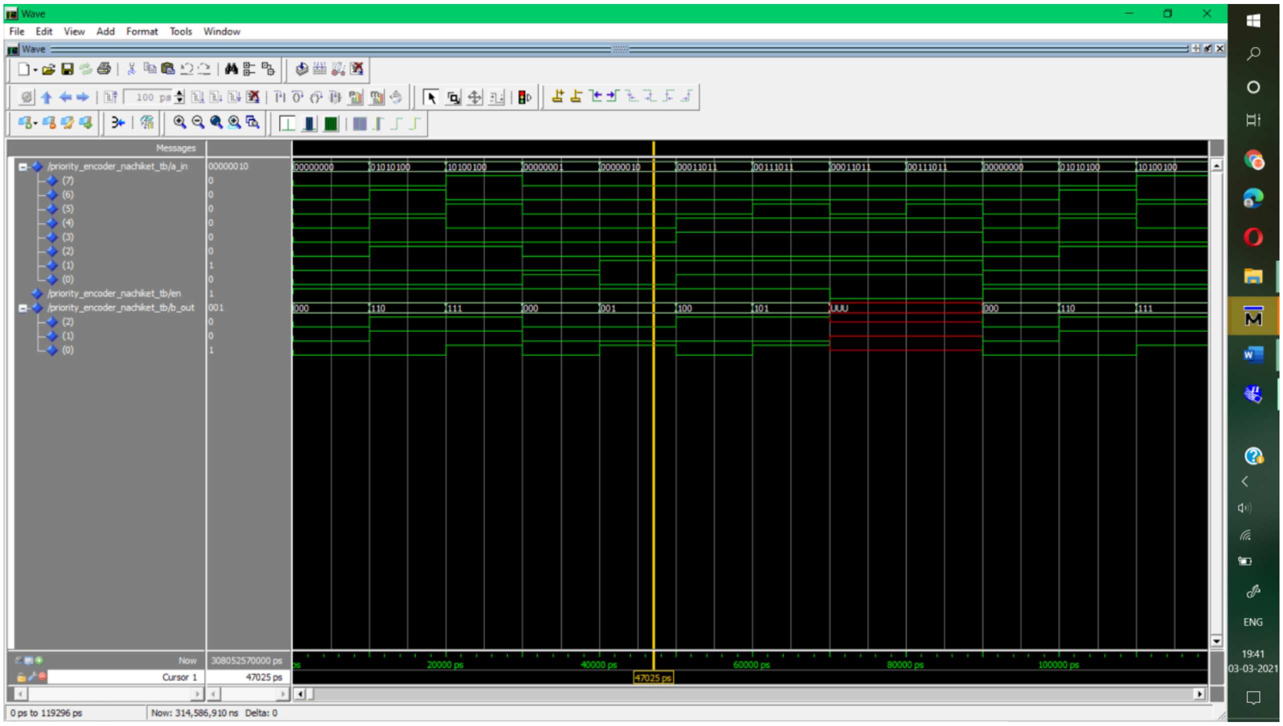
```
                        en <= '0';

                        a_in<="00011011";
                        wait for 10ns;

                        a_in<="00111011";
                        wait for 10ns;

            end process;
end priority_encoder_nachiket_tb_arch;
```

Simulation Waveform:

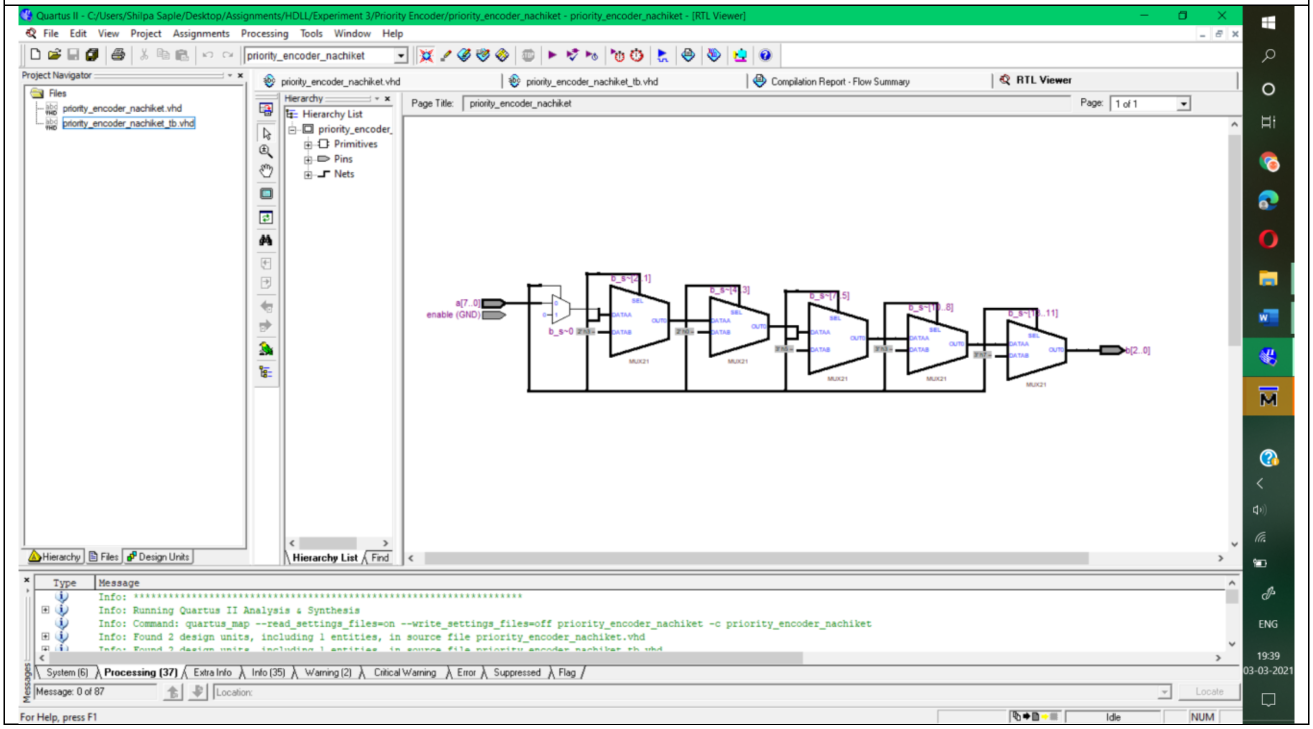


```
Messages
/priority_encoder_nachiket_tb/a_in   00000010
    (7)   0
    (6)   0
    (5)   0
    (4)   0
    (3)   0
    (2)   0
    (1)   1
    (0)   0
/priority_encoder_nachiket_tb/en     1
/priority_encoder_nachiket_tb/b_out  001
    (2)   0
    (1)   0
    (0)   1

Now        308052570000 ps
Cursor 1   47025 ps

0 ps to 119296 ps    Now: 314,586,910 ns  Delta: 0
```

RTL View:


Quartus II - C:/Users/Shilpa Saple/Desktop/Assignments/HDLL/Experiment 3/Priority Encoder/priority_encoder_nachiket - priority_encoder_nachiket - [RTL Viewer]
File Edit View Project Assignments Processing Tools Window Help
priority_encoder_nachiket
Project Navigator
Files
priority_encoder_nachiket.vhd
priority_encoder_nachiket_tb.vhd
Hierarchy
Hierarchy List
priority_encoder_
Primitives
Pins
Nets
priority_encoder_nachiket.vhd
priority_encoder_nachiket_tb.vhd
Compilation Report - Flow Summary
RTL Viewer
Page Title: priority_encoder_nachiket
Page: 1 of 1
a[7..0]
enable (GND)
b_s~0
b_s~[2..1]
b_s~[4..3]
b_s~[7..5]
b_s~[10..8]
b_s~[13..11]
SEL
DATAA
DATAB
OUT0
MUX21
b[2..0]
Hierarchy
Files
Design Units
Hierarchy List
Find
System (6)
Processing (37)
Extra Info
Info (35)
Warning (2)
Critical Warning
Error
Suppressed
Flag
Message: 0 of 87
Location:
Locate
For Help, press F1
Idle
NUM
ENG
19:39
03-03-2021

```
Type  Message
      Info: ********************************************************************
      Info: Running Quartus II Analysis & Synthesis
      Info: Command: quartus_map --read_settings_files=on --write_settings_files=off priority_encoder_nachiket -c priority_encoder_nachiket
      Info: Found 2 design units, including 1 entities, in source file priority_encoder_nachiket.vhd
      Info: Found 2 design units, including 1 entities, in source file priority_encoder_nachiket_tb.vhd
```

Post Lab Subjective/Objective type Questions:

Upload Answer of following question before coming to next laboratory.

Q1. What are the Shift Operators? Explain them with the help of examples

NACHIKET NAIK - 1912060 - B2, HDLL Exp 3,

POSTLAB Q1)

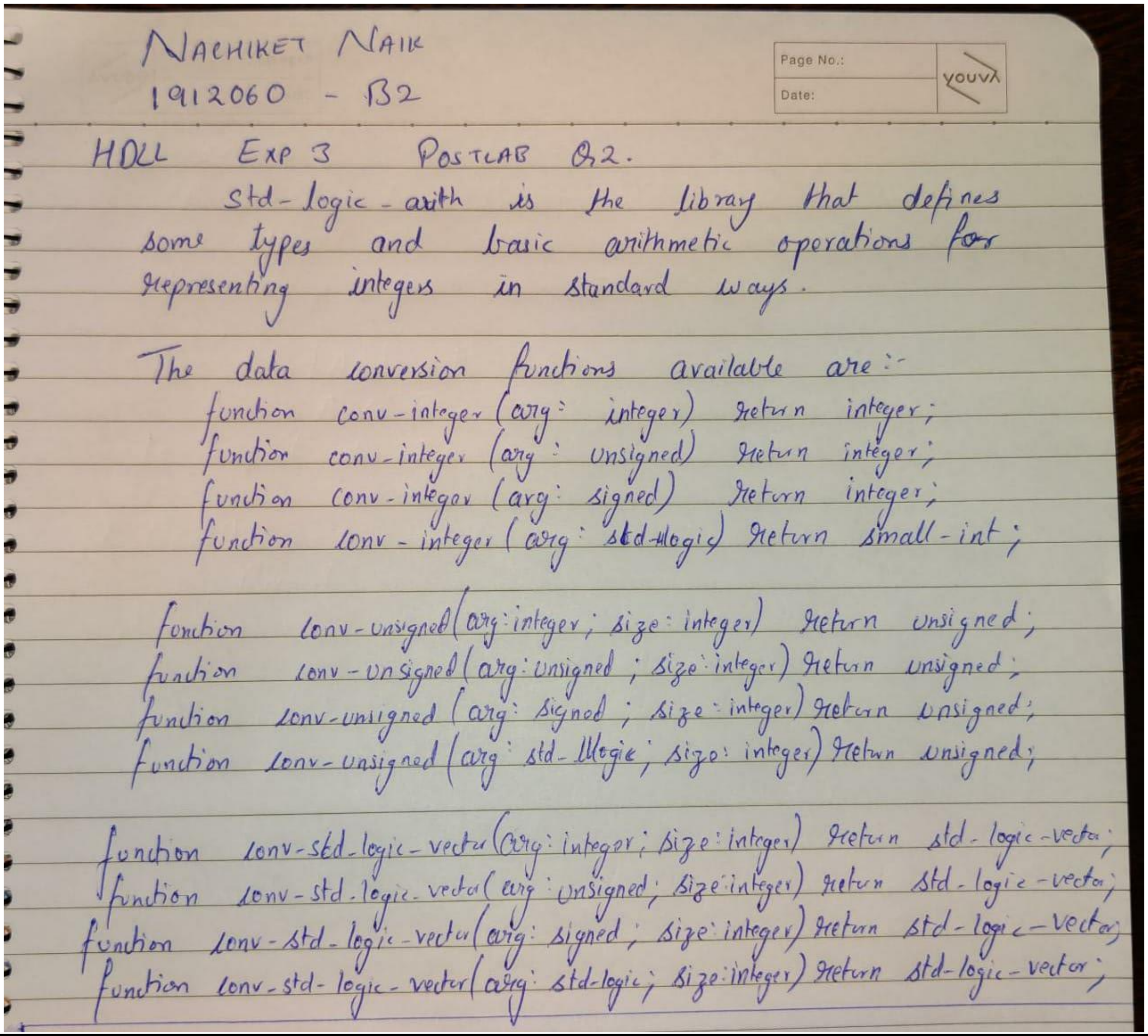
There are 6 shift operators available in VHDL :- sll, srl, sla, sra, rol, ror.

sll : shift left logical - fill value is '0'
srl : shift right logical - fill value is '0'
sla : shift left arithmetic - fill value is right hand bit
sra : shift right arithmetic - fill value is left-hand bit
rol : rotate left
ror : rotate right.

eg :- "1100" sll 1 → "1000"
"1100" sla 1 → "1000"
"1100" rol 1 → "1001"

~~"SRL 2~~ "1100" srl 2 → "0011"
"1100" sra 2 → "1111"
"1100" ror 2 → "0011"

Q2. What is std_logic_arith package in library ieee? Which data conversion functions are available in the same?



NACHIKET NAIK
1912060 - B2

Page No.:
Date:

HDLL EXP 3 POSTLAB Q2.

std-logic-arith is the library that defines some types and basic arithmetic operations for representing integers in standard ways.

The data conversion functions available are :-

```
function conv-integer (arg: integer) return integer;
function conv-integer (arg: unsigned) return integer;
function conv-integer (arg: signed) return integer;
function conv-integer (arg: std-logic) return small-int;

function conv-unsigned (arg: integer; size: integer) return unsigned;
function conv-unsigned (arg: unsigned; size: integer) return unsigned;
function conv-unsigned (arg: signed; size: integer) return unsigned;
function conv-unsigned (arg: std-ulogic; size: integer) return unsigned;

function conv-std-logic-vector (arg: integer; size: integer) return std-logic-vector;
function conv-std-logic-vector (arg: unsigned; size: integer) return std-logic-vector;
function conv-std-logic-vector (arg: signed; size: integer) return std-logic-vector;
function conv-std-logic-vector (arg: std-logic; size: integer) return std-logic-vector;
```

Conclusion:

In this experiment, we have implemented an 8:3 priority encoder in VHDL with enable input.

The priority encoders output corresponds to the currently active input which has the highest priority. So when an input with a higher priority is present, all other inputs with a lower priority will be ignored.
The enable has been given such that the priority encoder acts as active high; i.e. if enable is low, then it will give an undefined output ("UUU") else it will act as a priority encoder.

Signature of faculty in-charge with Date: