Experiment 6: String Recognizer

Nishant Thakre Roll Number 200070051

EE-214, WEL, IIT Bombay

October 5, 2021

## Overview of the experiment:

|  |
| --- |
| * The purpose of the experiment was to use Behavioral and Structural modelling to design a string recognizer which detects the word “covid’ in a given input of string * We used behavioral modelling to describe the circuit * In this report, I will describe the working of the circuit and show the subsequent simulations obtained from the said experiment. |

## Approach to the experiment:

|  |
| --- |
| I used Mealy type Finite State Machine to design the string detector .I used Behavioral and Dataflow modelling. If-else and case-when statements were used to describe the above circuit. I used the skeleton code given for use |

## Design document and VHDL code if relevant:

|  |
| --- |
| The truth table made using the figure shown in the previous section is:    Output 1 means the word covid has been detected.  VHDL code for behavioral modelling  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.numeric\_std.all;  entity cov\_detect is  port(inp:in std\_logic\_vector(4 downto 0);  reset,clock:in std\_logic;  outp: out std\_logic);  end cov\_detect;  architecture rch of cov\_detect is  ---------------Define state type here-----------------------------  type state is (init,s1,s2,s3,s4); -- Fill the code  ---------------Define signals of state type-----------------------  signal y\_present,y\_next: state:=init;  begin  clock\_proc:process(clock,reset)  begin  if(clock='1' and clock' event) then  if(reset='1') then  y\_present<=init; -- Fill the code  else  y\_present<=y\_next;  end if;  end if;    end process;  state\_transition\_proc:process(inp,y\_present)  begin  case y\_present is  when init=>  if(unsigned(inp)=3) then --c  y\_next<=s1;-- Fill the code  else  y\_next<=init;  end if;  outp<='0';  when s1=>  if(unsigned(inp)=15) then --c  y\_next<=s2;-- Fill the code  else  y\_next<=s1;  end if;  outp<='0';  when s2=>  if(unsigned(inp)=22) then --c  y\_next<=s3;-- Fill the code  else  y\_next<=s2;  end if;  outp<='0';  when s3=>  if(unsigned(inp)=9) then --c  y\_next<=s4;-- Fill the code  else  y\_next<=s3;  end if;  outp<='0';  when s4=>  if(unsigned(inp)=4) then --c  y\_next<=init;-- Fill the code  outp<='1';  else  y\_next<=s4;  outp<='0';  end if;  END CASE;  end process;  end rch; |

## RTL View:

|  |
| --- |
|  |

## DUT Input/Output Format:

|  |
| --- |
| Input format= < I4 I3 I2 I1 I0 reset clock>  Output format=<Out>  Input MSB=I4  Input LSB=clock  Output MSB=Out  Output LSB=Out  Some testcases are:  0001001 0 1  0001000 0 1  0001001 0 1  0001000 0 1 |

## RTL Simulation:

|  |
| --- |
|  |

## Gate-level Simulation:

|  |
| --- |
|  |

## Observations\*:

|  |
| --- |
| We obseved that all testcases have been passed and all test cases in scan chain also passed and the output file obtained from scan-chain also had all tests as SUCCESS |

## 