

# EE214 - Digital Circuits Lab

## Multiple String Detector

Thursday Batch

06/10/2022

### Instructions:

1. Use **Behavioural** modeling for writing VHDL description
2. Perform RTL simulation using the provided testbench and tracefile.
3. Demonstrate the simulations to your TA
4. Perform **Scanchain** on the Xenon board and verify with your TA.
5. Submit the entire project files in .zip format in moodle.

## 1 Problem statement

In this experiment, you will design a string detector using a Mealy type FSM which outputs '1' when input sequence of letters given thus far contains the following sub-sequences:

*run*

*cry*

*broom*

in a string of letters.

For this experiment, letter 'a' is encoded as "00001", 'b' is encoded as "00010" and so on.

For example suppose the input string is: *Bring UNO cards from my bag*

This string contains the subsequence "run" "cry" and "broom".

Thus, the input and output sequences when lined up will look like:

Bring UNO cards from my bag

0000000100000000000010010000

**Note:** Consider characters in this problem as case insensitive.

## 2 Design Specification.

- Design separate FSM for the words(run, cry, broom). You can take "OR" of the FSMs output to get the final output.
- Input: 5-bit input signal encodes blank-space and 26 lower-case characters (from a to z and where a = 1 to z = 26, and blank-space = 0) , Reset, Clock.
- In this problem Reset is synchronous.
- Tracefile format < 5 bit input > < Reset > < Clock > < Output > < Maskbit >
- Output: 1-bit output

### 3 Lab Task

- Describe behavioural model of the string detector Mealy type FSM in VHDL. [3\*5M = 15 Marks]
- First draw the state diagram for the detection of the desired words "run" "cry" and "broom". [3\*5M = 15 Marks]
- Perform RTL and Gate-level simulation using the provided testbench and tracefile.
- Demonstrate the simulations to your TA.
- Perform scan-chain and demonstrate to your TA. [3\*5M = 15 Marks]

### 4 Code Snippet for single word detection:

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

entity word_detection is
port(  inp:in std_logic_vector(4 downto 0);
      reset,clock:in std_logic;
      outp: out std_logic);
end word_detection;

architecture bhv of word_detection is

-----Define state type here-----
type state is (rst,s1,s2.....); -- Fill other states here
-----Define signals of state type-----
signal y_present,y_next: state:=rst;

begin
clock_proc:process(clock,reset)
begin
    if(clock='1' and clock' event) then
        if(...) then
            y_present<=
                --Here Reset is synchronous
                -- Fill the code here
        else
            -- Fill the code here
        end if;
    end if;

end process;
state_transition_proc:process(inp,y_present)
begin
    case y_present is
        when rst=>
            if(unsigned(inp)=18) then          --r has been detected
                y_next<=      -- Fill the code here
            else
                -----
                -----Fill rest of the code here-----
            end if;
        end case;
    end process;

output_proc:process(y_present, inp)  ----- output process after this which will give
-----the output based on the present state and input (Mealy machine)
begin
    case y_present is
        when rst=>
            outp<='0';
        end case;
    end process;
```

-----  
-----*fill*-----  
-----

-----