EE214 Digital Circuits Laboratory
Wadhwani Electronics Laboratory
Electrical Engineering IIT Bombay

Problem set: 7 HW | Date: Oct 1, 2023

Instructions:

1. Use **Dataflow** modeling for writing VHDL description
2. Perform RTL simulation using the provided testbench and tracefile.
3. Demonstrate the simulations to your TA
4. 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 will detect the occurrence of ***students*** word in a string of letters. The design accepts a sequence of letters coded in binary and outputs '1' if the required word is detected. The letters of **students** can be present anywhere in the string but in sequence.

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

For instance, "**l**s**l**p**t**l**u**l**k**d**l**e**l**i**n**l**t**l**s**l**h**" is the input text then the output sequence would be "00000000000000100". The state diagram below shows the flow of the FSM.

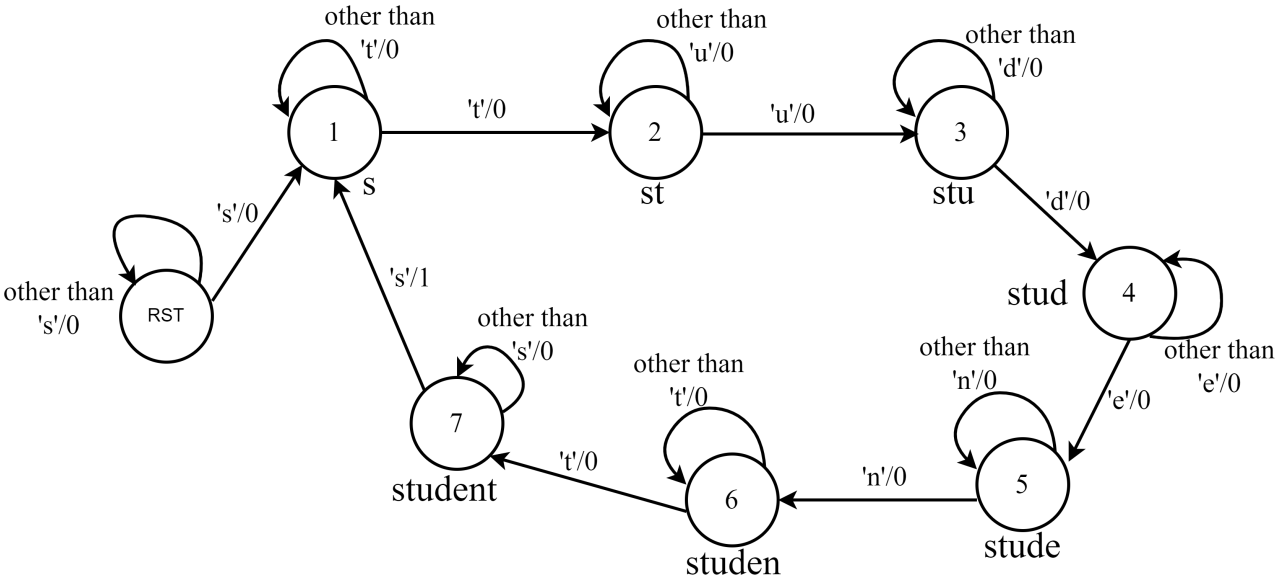

Figure 1: State diagram for detecting word "*students*" in the given string of alphabets.

In the state diagram, 8 states with one of its state being *RST* state is shown above. You have to fill the state table from the above state diagram.
Each state remembers the letters encountered so far of the word to be detected. If some other letter appears, it remains in the same state and waits for the correct input to arrive. For example, as shown in the figure(1) state-2 remembers "*st*" letters from the word "*students*". For any input other than "*u*", the next state is chosen as state-2. Similarly, state-7 remembers "*student*" letters from the word "*students*". For an input "*s*" when in state-7, the next state is chosen as state-1. And also outputs '1' as the entire string "*students*" is detected. For any other input other than "*s*", the next state is chosen as state-7 itself. This design shows an overlapping string detector as explained in the above example.

Reset	Input	Present state	Next state	Output
1	X	XXX	*RST*	0
0	's'	*RST*	1	
0	't'			
0	'u'			
0	'd'			
0	'e'			
0	'n'			
0	't'			
0	's'			1

Table 1: State transition table

2 Design Specification.

- Input: 5-bit input signal encodes blank-space and 26 lower-case characters (from a to z and where a = 1 to z = 26) , Reset, Clock
- TRACEFILE input format $5bitinput < nospace > Reset < nospace > Clock < space > Output < space > Maskbit$
- Output: 1-bit output

3 Lab Task

- Describe behavioral model of the string detector Mealy type FSM in VHDL.
- 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.

4 Code Snippet

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

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

architecture bhv of students 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(reset='1') then
                        y_present<=     -- 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)=19) then        --s has been detected
                                y_next<=    -- Fill the code here
                        else
                            -------------------------
        ---------Fill rest of the code here---------
        -------Similarly define output process after this which will give
        -------the output based on the present state and input (Mealy machine)
```