EEL3701C
Jaiden Magnan
11198

Problem

1. Design a state machine that counts 0, 1, 2, 3, 4, 5, 6, 7, 0, …. Your state bits can also be your count.
2. If the count is less than (not equal to) 4 **and A is true**, then F will be true.
3. If the count is 3 or 7, then TS will be true.
4. TS and F cannot be true at the same time, TS takes priority over F.

Inputs	**Outputs**
A(L)	Count (H)
	F(H)
	TS(H)

You will implement all combinatorial logic for the state machine in VHDL, sample code is given below. You only need to show a simulation for this state machine, no physical circuit is necessary.

Submit - Lab Quiz 5 Phone

- An ASM chart describing the behavior of the sequence. Use Mealy and Moore outputs as appropriate, and label each state.
- A comprehensive truth table that includes next-state equations and equations describing the outputs of your controller. It should have every possible combination of inputs, use wildcards and dont-cares when needed.

Submit - Lab Quiz 5 Generic

- A screenshot of your VHDL code.
- A screenshot of your BDF showing all sequential logic.
- A screenshot of your Quartus Simulation **(make sure you run the simulation).**
- The Quartus Archive.

EEL3701C
Jaiden Magnan
11198

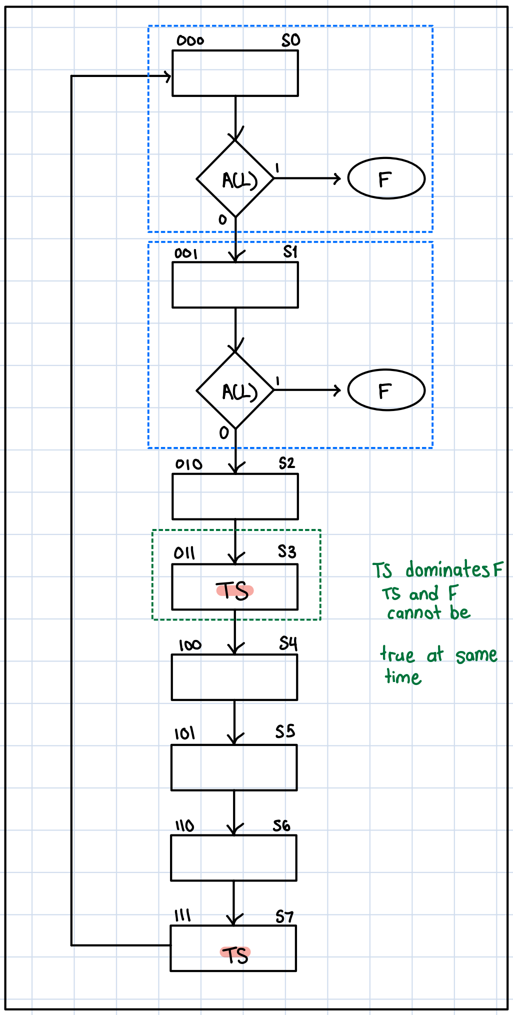
```
-- ex1.vhd from the Examples page of the course
website
library ieee;
use ieee.std_logic_1164.all;
entity ex1 is
port(
    A, B: in std_logic;
    Y : out std_logic);
end ex1;

architecture behav of ex1 is
signal Cout: std_logic;
begin
Y <= A or B;

end behav;
```

To declare a bus, use
`BUS_NAME: out bit_vector(1 downto 0);`
And access bit 1 with
`BUS_NAME(1)`

ASM Diagram



TS dominates F
TS and F cannot be true at same time

checks for F

Next State Truth Table

count	Q_2	Q_1	Q_0	A(L)	F	TS	Q_2^+	Q_1^+	Q_0^+	count$^+$
0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	1	0	0	0	1	1
1	0	0	1	0	0	0	0	1	0	2
1	0	0	1	1	1	0	0	1	0	2
2	0	1	0	–	–	0	0	1	1	3
3	0	1	1	–	–	1	1	0	0	4
4	1	0	0	–	–	0	1	0	1	5
5	1	0	1	–	–	0	1	1	0	6
6	1	1	0	–	–	0	1	1	1	7
7	1	1	1	–	–	1	0	0	0	0