Cairo University
Faculty of Engineering
Computer Engineering

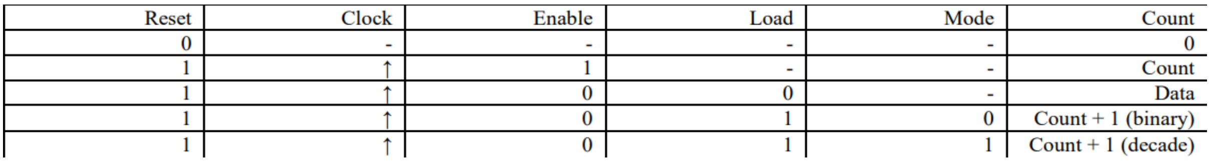
Computer Architecture

Lab 3

Q1) Write a VHDL code for a T flip flop, compile and simulate your code.

```
-- VHDL code for the T flip flop entity
Library ieee;
use ieee.std_logic_1164.all;
Entity tff is
        Port
        (T,rst,preset,clk : in std_logic;
        Q,Q_bar : out std_logic);
End tff;
```

Q2) It is required to design an 8 bit synchronous counter, with an enable, a parallel load, and an asynchronous reset. The counter loads or counts only when the enable is active. When the load is active, Data is loaded into count. The counter has two modes: binary and decade. In binary mode, it is an 8 bit binary counter. In decade mode, it counts in two 4 bit nibbles, each nibble counting from 0 to 9, and the bottom nibble carrying into the top nibble, such that it counts from 00 to 99 decimal.



Reset	Clock	Enable	Load	Mode	Count
0	-	-	-	-	0
1	↑	1	-	-	Count
1	↑	0	0	-	Data
1	↑	0	1	0	Count + 1 (binary)
1	↑	0	1	1	Count + 1 (decade)

Write a VHDL code for the previous circuit, compile and simulate. (you have to implement your own
full adder, don't use the + operator)

```
-- VHDL code for the sync_count entity

Library ieee;
use ieee.std_logic_1164.all;
Entity sync_count is
        Port
        (rst,clk,en,load,mode : in std_logic;
        Data : in std_logic_vector(7 downto 0);
        Count : out std_logic_vector(7 downto 0));
End sync_count;
```

Q3) write a testbench to your generic ALU submitted in the previous requirement. You can use the same test cases in your do file.