TFE4141 Assignment2:

Task1:

1.2

First, each signal is given an initial value which can be declared explicitly or implicitly (with or without a sensitivity list).

An event occurs, which is a change in the signal value.

A delta-delay will be used to handle the new events (if no explicit delay time is specified).

If specified explicitly, the signal assumes its new value after that specified delay.

Each concurrent process will be executed.

All processes sensitive to those signals execute until they reach a wait statement.

Future events will be scheduled in an event queue.

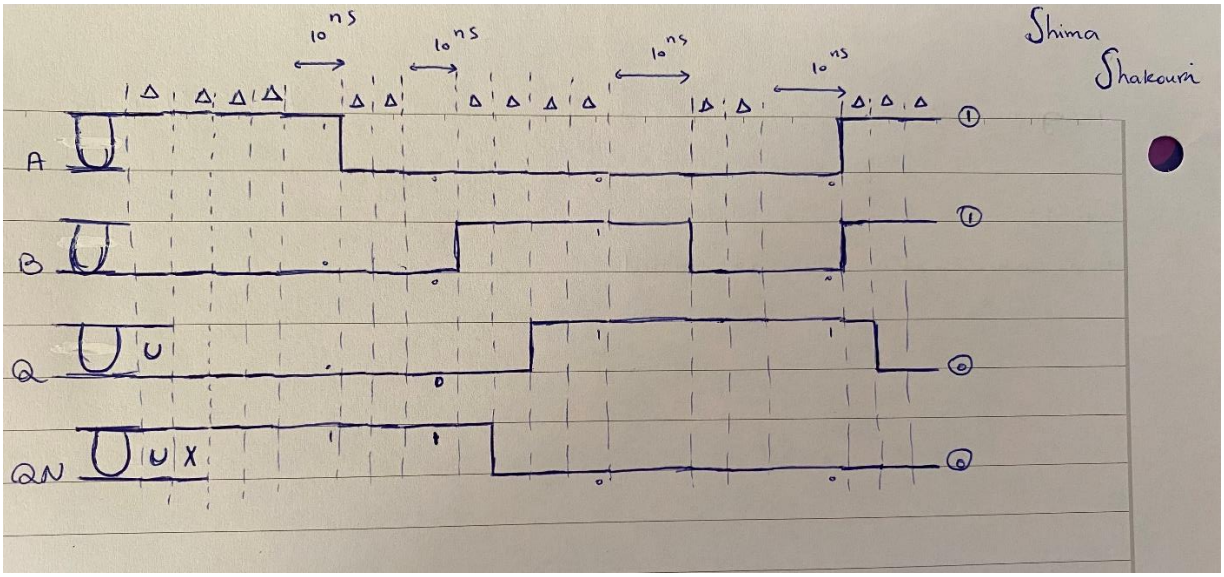
The cycle will continue until there is no more events.

1.3

If a signal is assigned value without an explicit delay, the implicit one which is delta_delay will be considered. If we have an explicit delay, sometimes because of that delay we will not be able to see some changes, cause for example an output should be evaluated after a certain delay (after changing of the signals) but in the time its waiting, the value of the signals may change again.

Task2

2.1



2.2

They will be suspended.

2.3

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Task_2 is
    Port ( A, B : in std_ulogic;
           Q, QN : inout std_ulogic);
end entity Task_2;

architecture behavior of Task_2 is
begin
    Q <= A nor QN;
    QN <= B nor Q;
end architecture behavior;
```

Test Bench:

```
library IEEE;
use IEEE.Std_logic_1164.all;

entity Task_2_tb is
end entity Task_2_tb;
architecture bench of Task_2_tb is
  signal A, B, Q, QN : std_ulogic;
begin
```

```
  dut: entity work.Task_2(behavior)
    port map ( A  => A, B  => B, Q  => Q, QN => QN );
  stimulus: process
  begin
  A <= '1' ; B <= '0' ;
  wait for 10 ns ;
  A <= '0' ;
  wait for 10 ns ;
  B <= '1' ;
  wait for 10 ns ;
  B <= '0' ;
  wait for 10 ns ;
  B <= '1' ; A <= '1' ;
    wait;
  end process stimulus;
end architecture bench;
```