TFE4140 Modeling and Analysis of Digital Systems

Assignment 1

Stian Hvatum (hvatum)
MTDT

January 24, 2013

Contents

Task 1 2
1 2
2 2
3 2
Task 2 4
1 4
2 4
3 5

Task 1

1

2

As we see in figure 1, a VHDL simulation cycle begins with fetching the first statement. The first statement is the topmost line in the topmost process. When we evaluate the statement, if the left side, Y, of the statement changes its value, then we have to add all statements that depends on Y to a dynamic event queue. This new event is added at current time + some given delay.

As the event queue is sorted by event time, all events will be processed in order. When the system comes to a state where no more events occur, and the queue of events are empty, the simulation has come to an end.

3

When no delay is explicitly specified for a statement, it implies a delay of zero. Since this might introduce changes to the event queue in such a way that statement order matters, we add a so called *delta delay* to the time which event is queued for. The delta delay is an infinitesimal amount of time, such that the changes are reflected before the next real time step, and do not interfere the already queued events.

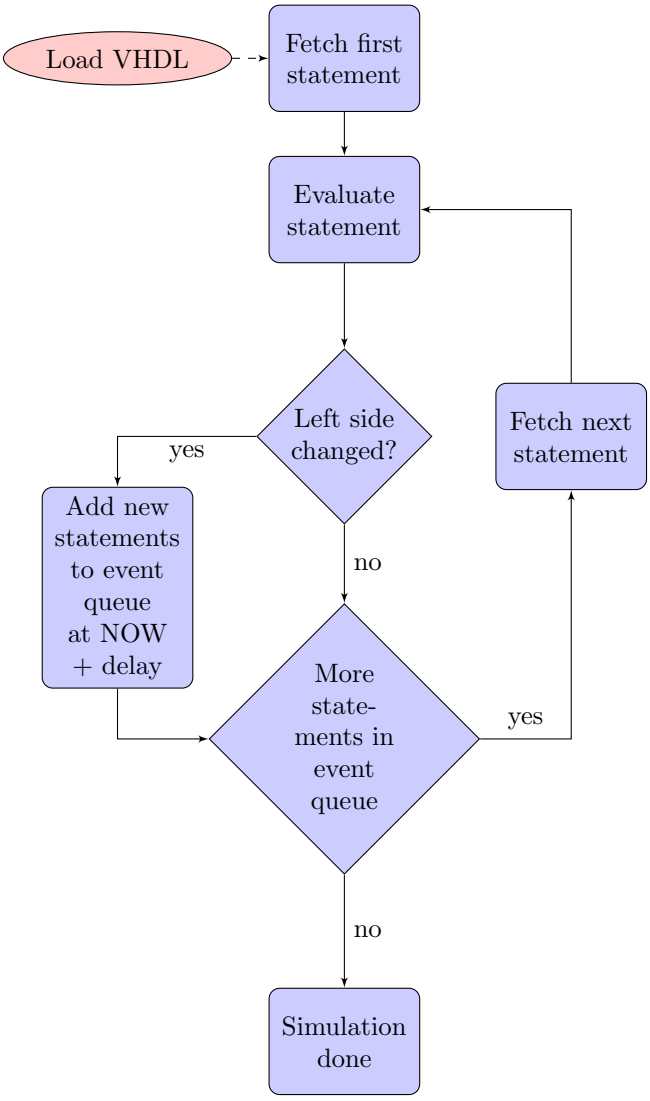

Figure 1: Simulation Cycle

Task 2

1

Time	A	B	Q	QN	Q+	QN+	Activated processes
0	U	U	U	U			Initializing, A,B activated
0+d	1	0			0	X	Q,QN
0+2d			0	X	(0)	1	Q,QN
0+3d			(0)	1		(1)	Q
0+4d				(1)			No change, suspend
10							A activated
10+d	0				(0)		No change, suspend
20							B activated
20+d		1				0	QN
20+2d				0	1		Q
20+3d			1			(0)	QN
20+4d				(0)			No change, suspend
30							B activated
30+d		0				(0)	QN
30+2d				(0)			No change, suspend
40							A,B activated
40+d	1	1			0	(0)	Q,QN
40+2d			0	(0)		(0)	QN
40+3d				(0)			No change, done

2

The simulation stops after all stimuli has been applied.

3

Implementation

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

ENTITY ab_latch IS
    PORT (A, B: IN std_ulogic;
          Q, QN: OUT std_ulogic);

    signal s_Q: std_ulogic;
    signal s_QN: std_ulogic;
END;

ARCHITECTURE behavior OF ab_latch IS
BEGIN
    s_Q <= A nor s_QN;
    s_QN <= B nor s_Q;
    Q <= s_Q;
    QN <= s_QN;
END;
```

Testbench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY test_ab_latch IS
END test_ab_latch;

ARCHITECTURE behavior OF test_ab_latch IS

    COMPONENT ab_latch
    PORT(
         A : IN  std_logic;
         B : IN  std_logic;
         Q : OUT  std_logic;
         QN : OUT  std_logic
        );
    END COMPONENT;

   signal A : std_logic := 'U';
   signal B : std_logic := 'U';

   signal Q : std_logic;
   signal QN : std_logic;

BEGIN
   uut: ab_latch PORT MAP (
          A => A,
          B => B,
          Q => Q,
          QN => QN
        );

   stim_proc: process
   begin
                wait for 100 ns;
                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' ;
   end process;
END;
```

Simulation

I ran my testbench in ISim. The results can be seen in figure 3

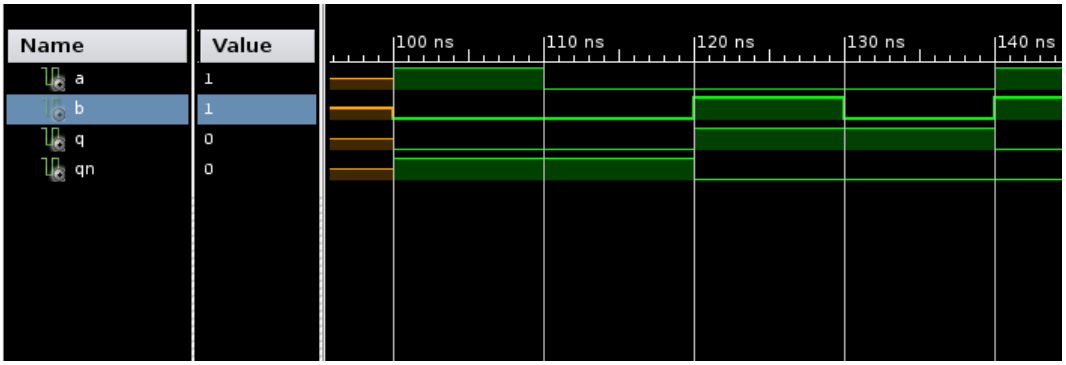

Figure 2: Simulating in ISim