Per Amund Roaulsson Marton & Dewan Abu Md. Arefin

16/09/18
DDS GROUP 4

TFE4141 Design of digital systems 1

Assignment 2: Understanding the Delta-delay

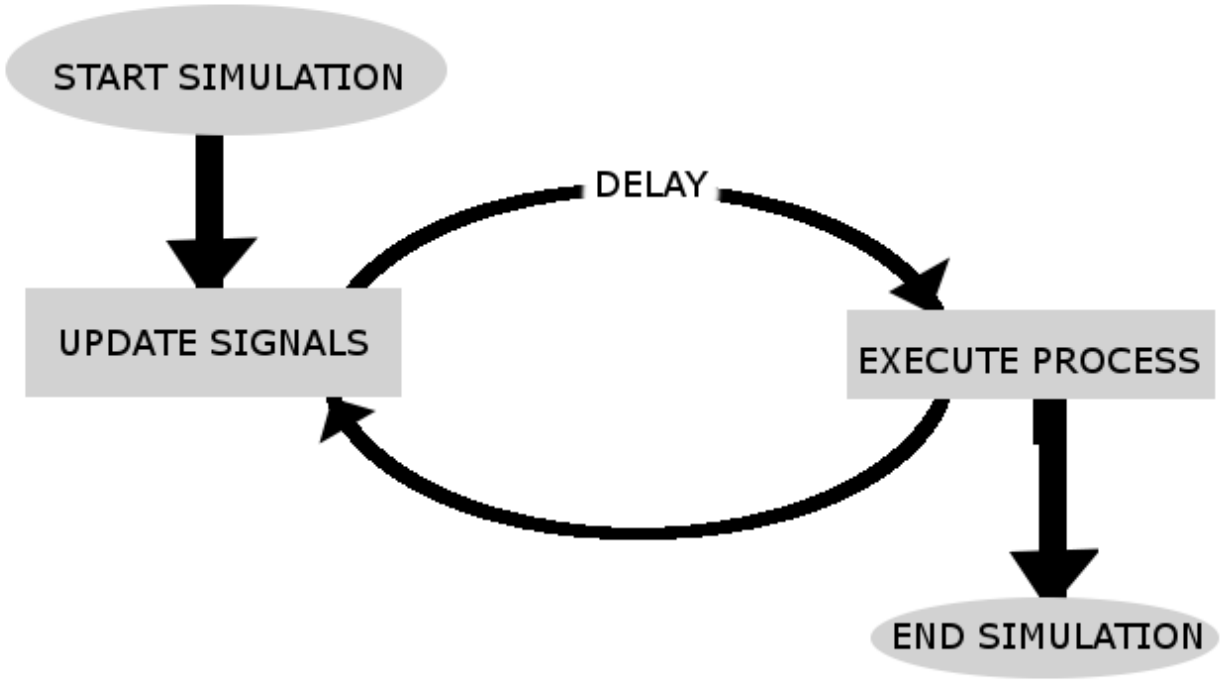
Task 1

1. Read the paper: Time and delta-delay in VHDL (found under Lectures/VHDL on it's learning).

   Answer: DONE

2. Explain in your own words what happens in a simulation cycle. If you want you may illustrate your explanation with a flow diagram. Keep your answer short.

   Answer: When a simulation starts, there is an update of a signal, then it will take some time (this is a delay) and then execute the process and update the output. This will continue in loop until the simulation stops.



3. Explain what happens to signals that are assigned values with and without **explicit delays**.

   Answer:

   With: The value updates after the specified time

   Without: The value updates after a delta-delay (smallest possible time delay)

Task 2

Given the following two concurrent (samtidige) processes in an architecture:

```
Q <= A nor QN;
QN <= B nor Q;
```

Given the following stimuli:

```
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' ;
```

1. Use the type **std_ulogic** as defined on the next page. Make a timing diagram for A, B, Q, and QN. Make sure that you include all delta-cycles.

Answer:

Time	A	B	Q	QN	Action	Activated Process
0ns	U	U	U	U	A <= '1' ; B <= '0'	Initalizing, A and B activated
0ns+Δ	1	0				Q, QN
0ns+2Δ			0	X		Q, QN
0ns+3Δ			(0)	1		Q
0ns+4Δ			(0)			No new change, Q is suspended
10ns					A <= '0'	A deactivated, visible next cycle
10ns+Δ	0					Q
10ns+2Δ			(0)			No new change, Q is suspended
20ns					B <= '1'	B activated, visible next cycle
20ns+Δ		1				QN
20ns+2Δ				0		Q
20ns+3Δ			1			QN
20ns+4Δ				(0)		No new change, QN is suspended
30ns					B <= '0'	B deactivated, visible next cycle
30ns+Δ		0				QN
30ns+2Δ				(0)		No new change, QN is suspended
40ns					B <= '1' ; A <= '1'	A and B activated, visible next cycle
40ns+Δ	1	1				Q, QN
40ns+2Δ			0	(0)		QN
40ns+3Δ				(0)		No new change, QN is suspended

2. Does the simulation stop after all stimuli have been applied and the two processes Q and QN are suspended, or do you experience oscillations?

Answer: Yes, it stops

3. Write VHDL-code with entity and architecture that implements the processes Q and QN. Write a testbench which applies the given stimuli to this entity.

Use the type std_ulogic which is defined as follows:

```
--
TYPE std_ulogic IS ( 'U',  -- Uninitialized
                     'X',  -- Forcing Unknown
                     '0',  -- Forcing 0
                     '1',  -- Forcing 1
                     'Z',  -- High Impedance
                     'W', -- Weak Unknown
                     'L',  -- Weak 0
                     'H',  -- Weak 1
                     '-',  -- Don't care );
--
```

Note: To make this type available you must include the following in you code:

***library** ieee ;*
***use** ieee.std_logic_1164.**all** ;*

These code lines must be written above the entity and architecture declaration.
When a simulation starts, a signal not given a specific initialization value, will be assigned the first value in the list, in this case '**U**' – (**U**ninitialized). Later on the signal value will typically change into '**X**'**, '1**' or '**0**'**.**

Answer:

```
-- VHDL CODE:

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity RS_nor is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Q : inout STD_LOGIC;
           QN : inout STD_LOGIC);
end RS_nor;

architecture Behavioral of RS_nor is

begin
    Q  <= A nor QN;
    QN <= B nor Q;
end Behavioral;




-- TESTBENCH

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

```
use IEEE.Numeric_Std.all;

entity RS_nor_tb is
end;

architecture bench of RS_nor_tb is

  component RS_nor
      Port ( A : in STD_LOGIC;
             B : in STD_LOGIC;
             Q : inout STD_LOGIC;
             QN : inout STD_LOGIC);
  end component;

  signal A: STD_LOGIC;
  signal B: STD_LOGIC;
  signal Q: STD_LOGIC;
  signal QN: STD_LOGIC;

begin
  uut: RS_nor 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 ;
     A <= '1' ; B <= '1' ;
    wait;
  end process;
end;
```