Signals & Variables (4A)

Young Won Lim
05/02/2025

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Sequential Statement

```
process(clk) is
begin
    if rising_edge(clk) then
        a <= b;
        b <= c;
        c <= a;
        a <= c;
    end if;
end process;
```

https://electronics.stackexchange.com/qauestions/372181/process-statements-and-sequential-execution-in-vhdl

Sequential Statement

Within a process, statements are indeed carried out sequentially. However, values assigned to signals are not carried out immediately but scheduled to occur at the end of the process.

For example, when your third assignment c <= a; is carried out, the value of a is still the value a had at the start of the process.

This is because the first assignment a <= b; has not yet been carried out and a has not changed.

In fact, the first assignment will never be carried out because of the fourth assignment a <= c;, which will be scheduled to occur at the process end instead.

This behaviour reflects the behaviour of real logic circuitry, which is what VHDL was designed to do.

https://electronics.stackexchange.com/qauestions/372181/process-statements-and-sequential-execution-in-vhdl

Sequential Statement

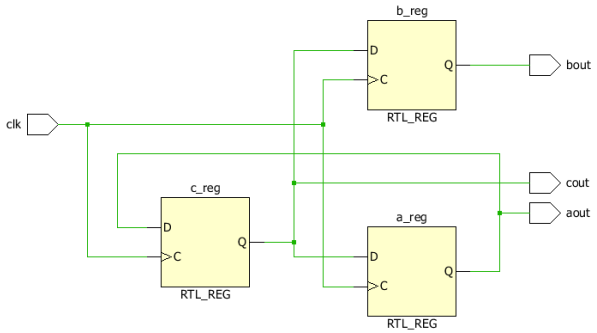
Reading your statements in reverse order:

a <= c -- the a register gets whatever was in the c register

c <= a -- the c register gets whatever was in the a register

b <= c -- the b register gets whatever was in the c register

a <= b -- this statement is ignored1.



1There can't be two drivers for a single FF input so, in VHDL, when multiple assignments to a signal occur within the same process statement, the last assigned value is the value which is propagated.

Notice also that b_reg and a_reg are exactly the same (i.e. they each clock the same signal in and out). If this simple example were to be synthesized, one of them would almost certainly be removed, and bout would be tied to aout. In fact, it took some keep attributes just to tell Vivado to not eliminate the b_reg for the RTL.

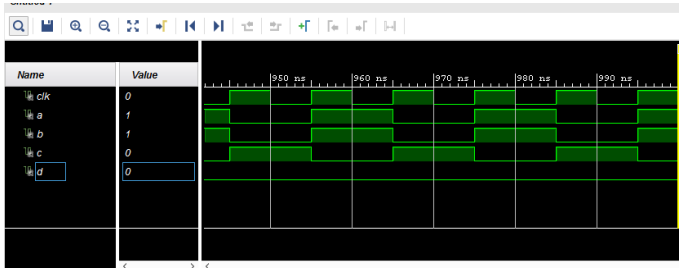
https://electronics.stackexchange.com/qauestions/372181/process-statements-and-sequential-execution-in-vhdl

Sequential Statement

In VHDL, statements in process execute sequentially. As you mentioned a, b, c and d are signals (if they were variables, they had different manner). assume these statements in process:

a <= b;

c <= a;



At the end of the process old value of b assigned to a. and old value of a assigned to c.

we can think in simpler way: statements in process executed sequentially, but the signals don't get the new values before end of the process. see this example that simulated in vivado:

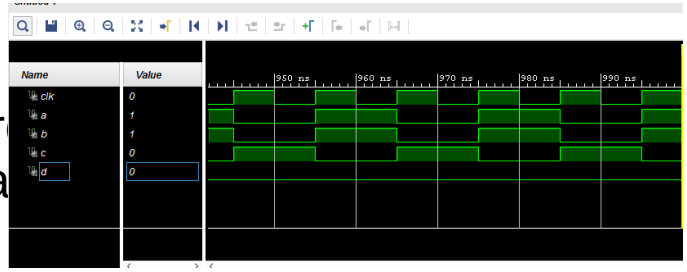
simulation

https://electronics.stackexchange.com/qauestions/372181/process-statements-and-sequential-execution-in-vhdl

Sequential Statement

you can see the initial value of the signal before rising edge of clock.

a=1; b=1; c=0;



In the rising edge of clock, the statements in the pr
of process the new value of signals (according to a
the question) are:

a=old c=0;

b=old c=0;

c=old a=1;

good luck.

https://electronics.stackexchange.com/qauestions/372181/process-statements-and-sequential-execution-in-vhdl

Sequential Statement

6.1.2. Sequential Signal Assignment Statement

Signals represent the interface between the concurrent domain of a VHDL model and the sequential do-

main within a process. A VHDL model is composed of several processes that communicate via signals. At simu-

lation, all hierarchy of the model is removed and only the processes and signals remain. The simulator alternates

between updating signal values and then running processes activated by changes of the signals listed in their

sensitivity lists

https://cse.usf.edu/~haozheng/teach/cda4253/doc/vhdl-stmt.pdf

Sequential Statement

6.1.2.1. Sequential Assignment Statement Execution

Signal assignment may be performed using a sequential statement or a concurrent statement. The se-

quential statement may only appear inside a process, while the concurrent statement may only appear outside

processes. The sequential signal assignment has a single form, the simple one, which is an unconditional as-

signment. The concurrent signal assignment has, in addition to its simple form (presented in Section 6.2.3.1),

two other forms: the conditional assignment (Section 6.2.3.2) and the selective assignment (Section 6.2.3.3). The

sequential signal assignment has the same syntax as the simple form of the concurrent signal assignment; the

difference between them results from context.

The sequential signal assignment has the following syntax:

signal <= expression [after delay];

https://cse.usf.edu/~haozheng/teach/cda4253/doc/vhdl-stmt.pdf

Sequential Statement

As a result of executing this statement in a process, the expression on the right-hand side of the assign-

ment symbol is evaluated and an event is scheduled to change the value of the signal. The simulator will only

change the value of a signal when the process suspends, and, if the after clause is used, after the delay speci-

fied from the current time. Therefore, in a process the signals will be updated only after executing all the state-

ments of the process or when a wait statement is encountered.

Typically, synthesis tools do not allow to use after clauses, or they ignore these clauses. The after

clauses are ignored not only because their interpretation for synthesis is not specified by standards, but also be-

cause it would be difficult to guarantee the results of such delays. For example, it is not clear whether the delay

should be interpreted as a minimum or maximum propagation delay. Also, it is not clear how the synthesis tool

https://cse.usf.edu/~haozheng/teach/cda4253/doc/vhdl-stmt.pdf

should proceed if a delay specified in the source code cannot be assured

Sequential Statement

A consequence of the way in which signal assignments within processes are executed is that when more

than one value is assigned to the same signal, only the last assignment will be effective. Thus, the two processes

in Example 6.7 are equivalent.

Example 6.7

```
proc7: process (a)
begin
z <= '0';
z <= a;
end process proc7;
```

https://cse.usf.edu/~haozheng/teach/cda4253/doc/vhdl-stmt.pdf

Sequential Statement

proc8: process (a)

begin

z <= a;

end process proc8;

In conclusion, the following important aspects should be taken into consideration when signal assign-

ment statements are used inside processes:

• Any signal assignment becomes effective only when the process suspends. Until that moment, all sig-

nals keep their old values.

• Only the last assignment to a signal will be effectively executed. Therefore, it would make no sense to

assign more than one value to a signal in the same process

https://cse.usf.edu/~haozheng/teach/cda4253/doc/vhdl-stmt.pdf

References

[1] http://en.wikipedia.org/
[2] J. V. Spiegel, VHDL Tutorial,
http://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html
[3] J. R. Armstrong, F. G. Gray, Structured Logic Design with VHDL
[4] Z. Navabi, VHDL Analysis and Modeling of Digital Systems
[5] D. Smith, HDL Chip Design
[6] http://www.csee.umbc.edu/portal/help/VHDL/stdpkg.html
[7] VHDL Tutorial - VHDL onlinewww.vhdl-online.de/tutorial/