VHDL notes

Fabien Le Mentec

October 17, 2015

Abstract

Set of practical VHDL notes and guidelines.

Contents

1 Reusability

1.1 General

as much as possible, standalone reusable components

1.2 Use unconstrained types

use unconstrainted arrays, propagate through hierarchy

1.3 Use type attributes

use type attribute as much as possible, esp. type'length and type'range

1.4 Use generics, not package constants

use generics for specialization. esp, think about template as non modifiable by the user.

2 Simulation

2.1 Per component test bench

use per component simulation test benches, not project wide. they are easier for the component user to understand, and force you to design stand alone components.

3 Synthesis

3.1 Inference

Usually, a VHDL developer does not explicitely indicate what hardware resource to use to implement logic. The synthetiser deduces that from its source code understanding (ie. signal netlist and operations). This process is known as inference.

Inference is very sensitive to the way code is written. For instance, the use of an additional signal to reset a shift register may prevent the synthetiser to infer a hardware shift register.

Thus, VHDL developers try as much as possible to write code in a standard way, that is known to be well understood by the synthetiser.

3.2 Explicit resource instanciation

Non portable but sure to instanciate the right resource.

3.3 Reset signals

Avoid reset signals. If not possible, make reset clock synchronous. The following construct helps:

```
process
begin
 wait until rising_edge(clk);

 if rst = '1' then
 end if;

end process;
```

3.4 Shift registers

XILINX FPGAs have hardware resources to implement shift registers.

4 Verification

4.1 Use assertions

use assertion to check data type lengths when unconstraints arrays

5 Documentation

5.1 Test benches as documentation

test benches also serve as a documentation for component users

6 Misc

6.1 Processes

6.2 Clocking

clear convention about how data passed to/from a component are clocked. by default, clocked using the component domain. idem for latching.

6.3 Typing

use right types (unsigned, slv ...), sizing and indexing from the beginning. it avoids further casting and simplifies the code.