CHAPTER 6

SUMMARY AND FUTURE WORK

6.1 Summary

VHDL code generator based on component diagram is an interesting subject with many and diverse programming languages, for example Java, JavaScript, PHP, HTML and VHDL as output languages.

Java was chosen as the programming language for several reasons. For one thing, it has a rich and well-documented API. Second, it means that the application can be run on different platforms (because this application was deployed as applet on web browser). The choice of Java was also motivated by the author’s personal curiosity regarding the Computer Aided Design (CAD) possibilities of the language.

Within the scope of the thesis, the functionality of the application (DiaHDL) is to be considered satisfactory. DiaHDL can be used to generate VHDL code based on a given component diagram. This code can be directly analyzed and synthesized. DiaHDL also provides a graphical user interface that allows the user to select a digital component, customize their parameter and generate the VHDL code.

Several libraries already been developed to show the functionality of DiaHDL. Those libraries are Ripple Carry Adder / Subtractor (with or without pipelining), Carry Look Ahead Adder / Subtractor (with or without pipelining), Signed Array Multiplier, Signed Carry Save Array Multiplier, and FIR Filter. All of these libraries are synthesizable and have already been tested on Xilinx ISE Design Suite 10.1.

This thesis documents the first phase of the development of a VHDL code generator based on component diagram. Currently, the application does rather well in controller circumstances, but to make it ready for typical usage
situations, more work is required. The following chapter gives some ideas of feature improvements.

6.2 Feature Improvements and Additions

Existing features can, of course, be improved upon in several areas, and there are also a number of new features which would be of benefit to general usability of the application. Some ideas and suggestions are mentioned below, in no particular order.

6.2.1 VHDL Library

Because the main focus of this research is to build the system from scratch then the VHDL library that can be provided is limited. Until the last time this report was written, there were 8 of VHDL code is included in the 5 components diagram. Those components can be used by the user to build a digital system. So the possibility to develop the library collection is still very wide.

6.2.2 Test Bench Implementation

The main focus of this research is to build the system that can generate VHDL code based on given diagram. As a complement of this program, in addition to generate the VHDL code, this application should also able to generate code for the Test Bench. Implementation of this Test Bench generator into DiaHDL is almost the same as the implementation of VHDL library.

6.2.3 Error Detection Improvements

In this initial version, development of DiaHDL is focused on the functionality of the system. Some of error detections were neglected. The future error checking should be focused on detecting error at design diagram.
6.2.4 Diagram Editing

In this early phase of development, the diagram editor that has been developed is very minimalist. Changing the parameters at configuration table does not appear in the component diagram. Additional editor features can be added to the tool. Achieving this will require deeper studies and extensive testing.

6.2.5 GUI

Graphical User Interface was developed which now only covers 4 main areas to meet the basic needs of the program. The view of GUI looks very simple. With the addition of the Look and Feel plug-in that is available on Java and open source community can make this view more splendid and interesting.