# A Study on Optimal Logic Circuits Combining RTL and Gate-Level Designs

ZHANG, Zhifei ( 1010754 )

School of Information Science,
Japan Advanced Institute of Science and Technology

February 8, 2014

**Keywords:** Logic Circuits, Field Programmable Gate Array (FPGA), Register Transfer Level (RTL), Verilog HDL, Schematic

It was a common way to use schematic editors/tools to implement logic circuits in ASIC and FPGA, where designers could directly design any logic structure they intended. However, with a great increase in logic sizes and the improvement of integrated circuit technology, the schematic design is reaching the limit. Recently, hardware description languages (HDL) such as Verilog HDL and VHDL have become the main trend. (This trend is gradually extending to higher-level language design, such as C language. Thanks to the hardware description languages, designers can significantly improve the (formerly cumbersome) processes. They can design without necessity of taking into account small circuit components, while they have only to consider the behaviors of their target hardware. Therefore, there is no doubt about efficiency in development. However, use of the hardware description languages requires logic synthesis processes that schematic design methods would not require. This means that the quality of the logic synthesis affects the performance of the design. For this reason, it is not easy to say that use of hardware description languages is always the best choice.

FPGA is an integrated circuit in which the internal circuits can be reconfigured and is now widely used in many electronic machinery fields.

This study targets FPGA designing, and the aim is to assess various design styles. The author used an integrated FPGA development software, ISE Design Suite 13.2, by Xilinx, Inc., which consists of simulation tools, schematic design input tools, logic synthesis, mapping, and place and route, etc. Use of HDL and schematic design can be evaluated as follows. The HDL description is synthesized and then mapped, placed, and routed by ISE tools. Similarly, the schematic design is mapped, place, and routed (without logic synthesis). Both designs are implemented into the Spartan-3E Starter Kit Board. Then the number of slices, LUTs, and the worst delay are reported. Comparing these reports, the generated circuites can be evaluated.

In this study, the following eight types of circuites are evaluated.

(1) Adder (2) Multiplexer (3) 7 segment decoder
(4) Tri-state (5) Shift register (6) Counter
(7) State- machine (8) CPU

In the evaluation, each type above is designed by different forms of HDL design and schematic design. The generated circuites are compared between the HDL design and schematic design. In addition, for large circuits, that is, CPU, circuits using HDL design and those using schematic design are mixed hierarchy.

After obtaining the results, it is found that, compared to schematic design, HDL design has an advantage of optimization by synthesis tools. The optimization process in synthesis tends to generate faster and smaller circuits. The main reason is that the optimization can utilize embedded carry logic elements, which are effective in high-speed circuits. This is the case especially for large scale circuits. As a whole, HDL can describe high abstraction level of circuits, which can well extract possibility of high optimization. On the other hand, for small circuits, Schematic desining is sometimes better. This means small components might have possibility of being well designed in the schematic style. Therefore, how to mix the two desining methods is worth considering.

Different HDL designs for the same logic had different results. This is because their abstraction levels are different and therefore high abstraction level provides rooms for optimization. As a result, elaborate optimization can exploit fast logic elements and generate better results. Similarly, different schematic designs had different result due to different logic structures. In addition, combination of the two design methods showed that the total

circuits including partial modules with better property had better results.

It is a promising idea that complicated and large-scale circuits should be divided into many modules and hierarchies. A small module has possibility of being designed by HDL and schematic style, which makes it possible to select better one. Combining modules with better property, the total circuits can be optimized and improved. Considering a development period and human (designer) resources, it is the most important to build a guide of which style (HDL or schematic) should be used for different types of complexity and scale of the target logic to shorten a design term, which is future work.