論文 / 著書情報
Article / Book Information

題目(和文)	C/C++によるシステムレベルRTLデザイン手法およびLLVMコンパイラインフラストラクチャを用いた設計効率の評価
Title(English)	C/C++ Based System Level RTL Design Methodology and Its Design Efficiency Using LLVM Compiler Infrastructure
著者(和文)	貞末多聞
Author(English)	Tamon Sadasue
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12388号, 授与年月日:2023年3月26日, 学位の種別:課程博士, 審査員:一色 剛,髙橋 篤司,本村 真人,劉 載勲,佐々木 広
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12388号, Conferred date:2023/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

（博士課程）
Doctoral Program

論 文 要 旨

THESIS SUMMARY

系・コース： Department of, Graduate major in	工学院 情報通信	系 コース	申請学位（専攻分野）： Academic Degree Requested	博士 Doctor of	（ 工学 ） Engineering
学生氏名： Student's Name	貞末 多聞		指導教員（主）： Academic Supervisor(main)	一色 剛	
			指導教員（副）： Academic Supervisor(sub)		

要旨（英文 800 語程度）
Thesis Summary（approx.800 English Words ）

As the scale of digital circuit design increases, design and verification using conventional RTL such as verilog-HDL and VHDL, has become a problem in terms of efficiency. We consider that conventional HDLs have design efficiency problems from three points of view: description efficiency problems, verification efficiency problems, and system-level design and verification problems. The inefficiency of conventional HDL descriptions is mainly due to the lack of aggregate types for interfaces, poor preprocessing, and poor parameterization capabilities without support for generic programming. Major problems with verification efficiency are that conventional HDL simulation takes much longer than software debugging and lacks interactive debugger environments such as GDB. The problem in system-level design and verification is that bus and communication protocol abstractions are not built into the design language, resulting in increased simulation time and reduced signal visibility.

To solve these problems of conventional hardware description language (HDL), a lot of methods have been proposed. One major approach is a high-level synthesis (HLS) method that generates hardware from a software program, and the other is a domain-specific language (DSL) method that specializes a high-level language to describe the hardware. However, previous approaches have advantages and disadvantages, and are not effective in all cases of hardware description.

C2RTL is a framework developed in Isshiki Laboratory against this background. Although it takes C language as input, unlike HLS, it enables description at the Register Transfer Level of abstraction, and provides the efficient hardware implementation and verification as well as the full control to hardware resources. However, the previous C2RTL did not support C++ code, so object-oriented programming and generic programming using templates were not available. Only IP level description was supported, system description was out of scope. Lastly, since the previous C2RTL was an in-house compiler, it is difficult to maintain compiler quality.

At this work, we introduced LLVM as a compiler platform and developed a methodology to convert LLVM-IR, an intermediate language of LLVM, into RTL-IR, an intermediate language of hardware structure. The verilog code and its equivalent C simulation code are generated from this RTL-IR. This mechanism ensures the compliance with the latest C/C++ standards and allows the application of LLVM's generic and customized optimization passes.

Next, by developing a two-step RTL description method for IP level and system level, we constructed a framework that can design and verify systems such as SoCs using only C/C++ language. Thus, by extending C2RTL to system-level descriptions, it is possible to complete the design and verification of the entire chip in C/C++ language and utilize an extensive software development environment such as debuggers and IDEs for hardware development.

In this paper, we clarify how LLVM-C2RTL converts input code to hardware representation via LLVM-IR, and then evaluated the design efficiency of LLVM-C2RTL through several studies.

In order to evaluate that the preprocessing function of LLVM-C2RTL improves the parameterization ability of hardware design, we implemented and evaluated a machine learning algorithm based on decision trees. We present a fully pipelined hardware implementation of Gradient Boosted Tree training with high performance and flexibility to realize highly parameterized machine learning models. This experimental result shows that our FPGA implementation achieves a 11- to 33-times faster performance and more than

300-times higher power efficiency than a GPU accelerated software implementation.

We evaluated the design efficiency of LLVM-C2RTL. As a result of comparing the amount of code required to create the AES encryptor and RISC-V core with other DSLs that enable register level description, we confirmed that the amount of code can be reduced by approximately 40%. In addition, the compile time of LLVM-C2RTL is about an order of magnitude faster than its counterparts, and the simulation time is comparable to verilator, one of the fastest verilog simulators.

As for the quality of the generated hardware, we verified that the synthesizable clock frequency is comparable to other approaches and can be improved by the automatic pipelining feature of LLVM-C2RTL, and the area of generated hardware is reduced through compiling with LLVM optimization.

These experimental results shows that LLVM-C2RTL framework offers an extremely efficient design and test environment because a complete system-level hardware design and debugging can be performed on powerful software development platforms and tool sets. The generated verilog code gives almost the same performance as hand-designed logic using the conventional HDL and automatic pipelining and system-level design functionality facilitate the trade-off search between performance and area size.