Advancing on-chip memory Implementations for Improved Performance in OpenROAD Flow Scripts

*Note: Sub-titles are not captured in Xplore and should not be used

Yimin Gao
ECE Dept.
University of Virginia
Charlottesville, United States
yg9bq@virginia.edu

Jun Yin
ECE Dept.
University of Virginia
Charlottesville, United States
htf6ry@virginia.edu

Bhupendra Singh Reniwal
ECE Dept.
University of Virginia
Charlottesville, United States
hfx3cu@virginia.edu

Mircea Stan
ECE Dept.
University of Virginia
Charlottesville, United States
mircea@virginia.edu

***Abstract*—This paper explores the use of OpenROAD Flow Scripts (ORFS) which automates the design implementation flow of OpenROAD, an open-source toolchain for RTL-to-GDSII digital implementation of System-on-Chip (SoC) design. In this paper We present our experience using ORFS on a RISC-V core and provide suggestions regarding optimizing on-chip memory implementations to improve the scripts for better performance.**

***Index Terms*—OpenROAD Flow Scripts, RTL-to-GDSII flow, open-source tools, automated design, on-chip memory**

I. Introduction

OpenROAD is an open-source toolchain used for RTL-to-GDSII digital implementation of System-on-Chip (SoC) design. The toolchain is complemented by the OpenROAD Flow Scripts (ORFS) which is a collection of TCL and python scripts that aim to automate the design implementation flow. Overall, our use of ORFS was a great experience. However, improvements could be made further to enhance the performance of a design.

II. ORFS Toolset Overview

The toolset is capable of automating various tasks in the RTL-to-GDSII flow without any issues. ORFS also includes features that allow an automated search for the best parameters for a specific design, and provides users with comprehensive statistics within the GUI after the flow, such as timing and power reports, etc.

Our use of ORFS for a two-stage pipelined RISC-V core ibex was a great experience overall. We went through the RTL to GDSII flow using ORFS for a 32-bit RISC-V core ibex. While the flow went through flawlessly, we also ran the Autotuner script for the design to find the best set of parameters of the design.

III. On-chip Implementations Improvement

However, improvements can still be made on top of the current ORFS Toolset to enhance the performance of the design. For most of the time, the large set of on-chip memory such as register file or cache memory is usually implemented as flip-flops in ORFS, which requires a significant amount of chip area.

Consequently, the optimized implementation using SRAM, flip-flop latch and mixed implementation of large on-chip memory has become increasingly necessary to further optimize the toolchain. In addition, incorporating the memory implementation assignment within Autotuner scripts can also enable the flow to achieve a more optimal balance of power, performance and area.

IV. Conclusion

Our experience using the ORFS Toolset for RTL-to-GDSII flow of a RISC-V core has been favorable. The tools are user-friendly and capable of finishing the tasks within a short period of time. To optimize the toolset, we suggest applying advanced memory implementation (SRAM, flip-flop, latch, and mixed implementation) in large on-chip memory such as register file and cache. In addition, a search of the best memory mixed implementation could be incorporated within the current AutoTuner toolset.

References

[1] Ajayi, Tutu and Blaauw, David, "Openroad: Toward a self-driving, open-source digital layout implementation tool chain," Proceedings of Government Microcircuit Applications and Critical Technology Conference, 2019

[2] McLean, C., Kujawinski, E. B. (2020). AutoTuner: high fidelity and robust parameter selection for metabolomics data processing. Analytical chemistry, 92(8), 5724-5732.

[3] M. Marinella, "The future of memory," 2013 IEEE Aerospace Conference, Big Sky, MT, USA, 2013, pp. 1-11, doi: 10.1109/AERO.2013.6507427.

[4] Mohammad, K., Tekeste, T., Mohammad, B., Saleh, H., Qurran, M. (2019). Embedded memory options for ultra-low power IoT devices. Microelectronics Journal, 93, 104634.

[5] Meinerzhagen, P., Roth, C., Burg, A. (2010, August). Towards generic low-power area-efficient standard cell based memory architectures. In 2010 53rd IEEE International Midwest Symposium on Circuits and Systems (pp. 129-132). IEEE.

[6] Nedovic, N., Oklobdzija, V. G. (2000, September). Hybrid latch flip-flop with improved power efficiency. In Proceedings 13th Symposium on Integrated Circuits and Systems Design (Cat. No. PR00843) (pp. 211-215). IEEE.