

Accelerating FPGA-Based Wi-Fi Transceiver Design and Prototyping by High-Level Synthesis

Thijs Havinga, Xianjun Jiao, Wei Liu and Ingrid Moerman
IDLab, Department of Information Technology
Ghent University - imec
Ghent, Belgium
{firstname.lastname@UGent.be}

Abstract—Field-Programmable Gate Array (FPGA)-based Software-Defined Radio (SDR) is well-suited for experimenting with advanced wireless communication systems, as it allows to alter the architecture promptly while obtaining high performance. However, programming the FPGA using a Hardware Description Language (HDL) is a time-consuming task for FPGA developers and difficult for software developers, which limits the potential of SDR. High-Level Synthesis (HLS) tools aid the designers by allowing them to program on a higher layer of abstraction. However, if not carefully designed, it may lead to a degradation in computing performance or significant increase in resource utilization. This work shows that it is feasible to design modern Orthogonal Frequency Division Multiplex (OFDM) baseband processing modules like channel estimation and equalization using HLS without sacrificing performance and to integrate them in an HDL design to form a fully-operational FPGA-based Wi-Fi (IEEE 802.11a/g/n) transceiver. Starting from no HLS experience, a design with limited overhead in terms of latency and resource utilization as compared to the HDL approach was created in less than one month. The FPGA design generated by HLS achieved the same performance as compared to its HDL counterpart when deployed on a System-on-Chip (SoC)-based SDR, as verified by a professional wireless connectivity tester.

A System-on-Chip, consisting of a Central Processing Unit (CPU) and FPGA with high-speed interconnection, is a suitable platform for high-performance wireless systems, while preserving flexibility for prototyping. In this work we focus on the receiver baseband architecture of *openwifi* [1], an FPGA-based 20MHz bandwidth Single-Input Single-Output (SISO) Wi-Fi transceiver, originally designed in Verilog. We specifically targeted the channel estimator and equalizer, which are the core modules needed to remove channel impairments and sampling clock offset from the received signal, to be implemented using HLS. These modules have strict latency and throughput requirements to keep up with the incoming samples. Furthermore, the overall decoding latency should be limited such that an acknowledgement to a received frame can be sent within 16 μ s as required by the standard, which is not possible using host-based SDR.

As starting point we use a bit-true model of the HDL design which is implemented in MATLAB. This reference design is used to create C++ code in Vitis HLS, which leads to similar comprehensible sequential code, as opposed to the HDL code that needs to be aligned on a clock-cycle level.

This research was partially funded by the Flemish FWO SBO #S003921N VERI-END.com project.

TABLE I
HARDWARE UTILIZATION WHEN USING HDL OR HLS METHODOLOGY.

	LUT		FF		BRAM		DSP	
HDL	5,417	100%	10,073	100%	1	100%	19	100%
HLS	6,833	126.1%	7,208	71.6%	5	500%	34	178.9%

After inserting pragmas to optimize performance and resource utilization, C/RTL co-simulations are performed to confirm that the design meets the latency and throughput requirements. The design was successfully synthesized and implemented for a Xilinx ZedBoard, at a clock frequency of 100MHz. The hardware utilization of the modules using HDL or HLS are shown in Table I. In terms of flip-flops (FFs), the utilization is decreased. The increase in look-up tables (LUTs) is mostly due to the divider instances, for which in the HDL design dedicated IPs are used. The increase in BRAMs comes from the use of buffers instead of handling all samples in a streaming manner, and to store constants that are implemented using LUTs in HDL. Efficient sharing of digital signal processing (DSP) blocks in HDL lead to a lower utilization - it requires more effort to optimize this in HLS when this becomes a constraint.

For experimental validation, the FPGA design with HLS-generated channel estimator and equalizer is loaded on the ZedBoard with AD-FMCOMMS2-EBZ RF front-end and correctly interacts with the Linux mac80211 subsystem running on the on-chip CPU. It successfully received packets from a wireless connectivity tester and sends back an acknowledgment in time. From this, the receiver sensitivity at 10% packet error ratio was derived, which was measured to be similar to the HDL design for every modulation and coding scheme.

The source-code of the HLS-enhanced version is available at [2]. This approach already helped us to further enhance the design for better resistance to multipath fading and the implementation of Wi-Fi 6. It also speeds up the verification process using prototypes in a real-life environment, which is important for wireless research. Furthermore, we believe it opens opportunities for wireless system designers that are less familiar with digital hardware design.

REFERENCES

- [1] X. Jiao, W. Liu, M. Mehari, M. Aslam, and I. Moerman, "openwifi: a free and open-source IEEE802.11 SDR implementation on SoC," in *VTC2020-Spring*. IEEE, 2020, pp. 1–2.
- [2] <https://anonymous.4open.science/r/openofdm-17D7/>.