1 DFIANT HDL

DFiant [3] is a hardware description language (HDL) and one of the LEGaTO programming models. For a long time, the dominating HDLs have been Verilog, System Verilog, and VHDL, and they all provide the same register transfer-level (RTL) hardware design abstraction. An RTL language burdens designers with explicitly clocked constructs that do not distinguish between design functionality and implementation constraints (e.g., timing, target device). For example, VHDL and Verilog constructs require designers to explicitly place a register, regardless if it is part of the core functionality (e.g., a state-machine state register), an artifact of the timing constraints (e.g, a pipeline register), or an artifact of the target interface (e.g., a synchronous protocol cycle delay). These semantics narrow design correctness to specific timing restrictions, while vendor library component instances couple the design to a given target device. Evidently, formulating complex portable designs is difficult, if not impossible. Finally, these older languages do not support modern programming features that enhance productivity and correctness such as polymorphism and type safety.

High-level synthesis (HLS) tools such as Vivado HLS [5], and high-level HDLs such as Bluespec SystemVerilog [2] and Chisel [1] attempt to bridge the programmability gap. While these tools and languages tend to incorporate modern programming features, they still mix functionality with timing and device constraints, or lack hardware construction and timed synchronization control. On one hand, Chisel and Bluespec constructs explicitly pipeline designs. And on the other hand, Vivado HLS C++ constructs cannot directly support a simple task as toggling a led at a given rate. Such tools and languages, therefore, fail to deliver a clean separation between functionality and implementation that can yield portable code, while providing general purpose HDL constructs.

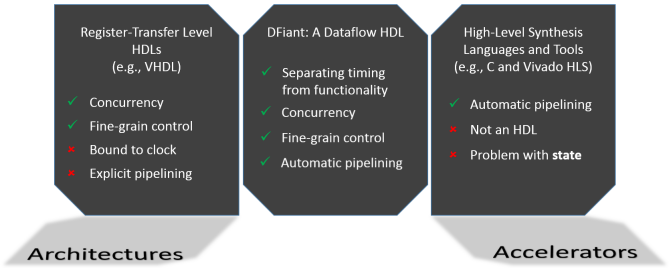

Figure 1: DFiant fills in the programmability gaps

Figure 1 summarizes the primary programmability pros and cons of RTL and HLS languages with their targeted programming domains, architectures and accelerators, respectively. DFiant aims to bridge over the programmability gaps by combining constructs and semantics from software, hardware and dataflow languages. DFiant is *not* an RTL language, nor is it a sequential HLS language such as C. Instead, the DFiant programming model accommodates a middle-ground approach between low-level hardware description and high-level sequential programming.

DFiant is a modern HDL whose goal is to improve hardware programmability and designer productivity by enabling designers to express truly portable and composable hardware designs. DFiant decouples functionality from timing constraints (in an effort to end the *"tyranny of the clock"* [4]). DFiant offers a clean model for hardware construction based on its core characteristics: (i) a clock-agnostic dataflow model that enables implicit parallel data and computation scheduling; and (ii) functional register/state constructs accompanied by an automatic pipelining process, which eliminate all explicit register placements along with their direct clock dependency.

DFiant is implemented as a Scala library and relies on Scala's strong, extensible, and polymorphic type system to provide its own hardware-focused type system (e.g., bit-accurate dataflow types, input/output port types). The library performs two main tasks: the frontend compilation, which translates dataflow variable interactions into a dependency graph; and the backend compilation, which translates the graph into a pipelined RTL code and a TCL constraints file, followed by a hardware synthesis process using commercial tools. DFiant can be used in any Scala-compatible IDE, including the LEGaTO eclipse-based tool-chain.

REFERENCES

[1] Jonathan Bachrach, Huy Vo, Brian Richards, Yunsup Lee, Andrew Waterman, Rimas Avižienis, John Wawrzynek, and Krste Asanović. 2012. Chisel: Constructing Hardware in a Scala Embedded Language. In *ACM/EDAC/IEEE Design Automation Conference (DAC)*.

[2] Rishiyur Nikhil. 2004. Bluespec System Verilog: efficient, correct RTL from high level specifications. In *ACM/IEEE Intl. Conf. on Formal Methods and Models for Co-Design*.

[3] Oron Port and Yoav Etsion. 2017. DFiant: A Dataflow Hardware Description Language. In *Intl. Conf. on Field Programmable Logic and Applications*.

[4] I Sutherland. 2012. The tyranny of the clock. *Comm. ACM* 55, 10 (2012), 35–36.

[5] Xilinx. 2015. Vivado High Level Synthesis User Guide. (2015).