Preview Experience with the Amazon F1: A White Paper

Shep Siegel, CTO

Atomic Rules LLC

Auburn, NH

{shepard.siegel|eric.laforest|aaron.severance}@atomicrules.com

The Background

The writing was on the wall well before Amazon's announcement of the AWS F1 service in November 2016. The successes and progressive disclosures of what Microsoft's Catapult team had done internally for accelerating Bing's search had shifted opinions dramatically in just two years. By the summer of 2016, Microsoft held a hands-on Catapult workshop ahead of FPL2016, and this author was then completely convinced that Microsoft's early success was not a one-off.

Our team was fortunate: Long before attention was focused on F1; Atomic Rules had spent the spring of 2016 working with Xilinx on ES1 silicon of their UltraScale+ devices VU3P and VU9P. The VU3P broke new ground with first-functional Gen3x16 PCIe, hardened CMACs with FEC, and rock-solid 28 Gbps transceivers. The VU9P did the same, but due to its size, added the new challenge of the two Laguna trenches separating the three Super Logic Regions. This history is vital to the arc that led Atomic Rules to F1.

Atomic Rules provides FPGA design services as well as sells IP Cores as products. Well before our journey to Lausanne, we were running our codes on UltraScale+ ES1 silicon in our labs: At speed, instrumented, and to our engineer's and stakeholder's satisfaction. Knowing firsthand that the VU9P was functional removed a great deal of fear, uncertainty and doubt.

With the F1 announcement and invitation from Amazon to participate in the beta preview; we became privy to a tranche of information. Almost all of this was terrific: Standardized AXI interfaces. A capable and well-provisioned Xeon host. The VU9P that we know and love. There were, however, a few immediate surprises: No direct network interface to the FPGA. All network traffic would go through PCIe to the host. Amazon has their reasons for this.

The one F1 item that raised our team's eyebrows was Amazon's bet-the-product use of Partial Reconfiguration to separate the F1 Shell and Customer Logic. FPGA vendors have been promising Partial Reconfiguration for decades, with successes and failures in fits and starts. If the Partial Reconfiguration flows didn't work, F1 would fail. Xilinx has been in an extended stretch of exceptionally good execution; so the Atomic Rules team jumped on board with the F1.

The Development

Developing for the F1 poses new challenges, but also removes obstacles. Designers normally used to the usual Vivado tool chain now have to learn how to handle EC2 virtual machines and S3 storage, particularly around their access policies. However, Amazon has documented the process, which only needs to be set-up once. Designers are also freed of the initial investment in expensive FPGA development boards, and are provided tested, pre-defined interfaces to the FPGA and to multiple banks of DDR4 memory.

Amazon provides an FPGA Developer AMI to run its own machine instances. This AMI provides a complete Vivado setup with all the necessary licenses. However, it requires a beefy instance with a minimum of 32GB of RAM, as we found some TCL issues were causing Vivado failures on machines with only 15GB RAM, even if it was not exhausted. We reported the issue on Amazon's FPGA Development forum and got a prompt response: it was a known issue and being worked on (As of this writing, F1 is still in pre-release beta, with release just around the corner).

Being a hands-on "show-me" FPGA shop from the beginning, we were already equipped for on-premises F1 development with Vivado, and so we moved our development in-house. Our first try had us manually obtaining and installing the F1-required licenses from Xilinx. The F1 environment makes heavy use of HDL encryption, partial reconfiguration, and other special features. To avoid this problem, Xilinx eventually made available a single license specific to F1 development which contained all the necessary features.

Amazon's F1 is structured into two parts: the Shell and the Custom Logic. The Shell is the pre-defined interface on the FPGA to which our design, the Custom Logic, must connect to. The Shell isolates potentially malfunctioning (or even malicious) Custom Logic from the host computer by defining various AXI interfaces to the host, as well as to the FPGA's own dedicated banks of DDR4 memory, and provides clocks, resets, and miscellaneous monitoring bits.

Amazon's GitHub F1 code repository provides all the Shell documentation and IP and a basic framework for simulation and synthesis, with scripts meant to be altered by the user. We had no difficulty integrating Verilog, System Verilog, VHDL, and EDIF netlists within a single project. The code repository also provides an SDK with libraries to allow host-side applications to communicate with the Shell.

Atomic Rules uses Domain Specific Languages such as Bluespec SystemVerilog (BSV), Vivado HLS, and others, where appropriate, as the cardinal source language for many of our RTL IP components. These components are instantiated inside the F1's Custom Logic. The standardized AXI interfaces of the F1 Custom Logic made our choices, and we suspect others choices, essentially a NOP. We found that the use of Domain Specific Languages appeared neither amplified nor attenuated by the F1 experience.

We developed our initial prototype in simulation, complete with a C-language interface to the F1 Shell model. This allowed us to reuse our C test benches as the actual application running on the host and interacting with the FPGA. We had no difficulty using Xilinx's own XSIM, and the provided Bus Functional Modules for the Shell were quite faithful: we found our simulations to work in hardware on the first try. The well-defined Shell interfaces allowed us to easily co-design host-side client and server network code along with the FPGA processing logic.

Acknowledgements

Atomic Rules would like to thank the F1 team at Amazon for allowing us the privilege of participating in the F1 beta preview. We would like to thank Xilinx, especially the Vivado team, for supporting us and our "Birds-of-a-Feather" meetings with other F1 Preview teams. We could not have gotten the head-start we did without early-access support from BittWare; on whose VU9P-carrying XUPP3R board we did most of our early F1 development. And both GateForge Consulting Ltd. (Toronto, ON) and VectorBlox Computing Inc. (Vancouver, BC) who, without their expert competencies, this work would not have been awesome.

Atomic Rules provides consulting and implementation services for those looking to exploit the opportunities the AWS F1 provides. For more information, please email info@atomicrules.com or visit www.atomicrules.com