RIVER: Reconfigurable Pre-Synthesized-Streaming Architecture for Signal Processing on FPGAs

Dominic Hillenbrand*, Christian Brugger*, Matthias Balzer*
*KIT - Karlsruhe Institute of Technology
Email: {dominic.hillenbrand, matthias.balzer}@kit.edu, christian.brugger@student.kit.edu

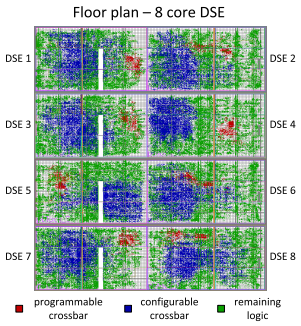

Figure 1. The figure shows the XC7V2000T-FLG1925-2 floor plan for an 8-core DSE system. Obviously, the configurable crossbar (blue) consumes a large percentage of the FPGA area. The programmable crossbar (red) is much smaller in comparison. Other components are more difficult to visualize since they are spatially more distributed. The remaining logic is shown in green.

I. Introduction

In embedded systems it is not uncommon to process high-bandwidth data streams in the range of gigabytes per second. FPGAs are capable of processing multiple data-streams in parallel, in real-time and at high data rates. Since FPGAs have become more pervasive in embedded systems it is desirable to make their capabilities accessible to a wider range of users. **Our novel contributions are**: **(1)** A scalable, run-time configurable and programmable signal processing architecture which covers a wide performance spectrum. **(2)** A framework which avoids lengthy compilation times by utilizing a repository of pre-synthesized FPGA configurations.

II. Design, Architecture

Our signal processing core is called "*Dynamic Streaming Engine (DSE)*". DSEs are run-time programmable, configurable and scalable by design. We achieve programmability through sequencers. Our sequencers have program counters, jump logic, instruction memory and support arbitrarily nested loops. Data is processed by one or more computational lanes that sit between two programmable crossbars. The crossbars feed data into computational lanes and forward intermediate results into a reduction stage. Individual DSEs can scale up- and down by changing: the number of stream input- and output ports, the number of computational lanes, the complexity and number of instructions, the complexity of the reduction stage, the number of buffers within computational lanes and the width of fixed point numbers. A "fat" DSE, for example, can support several smaller to medium sized convolutional 2D-filters (3x3-5x5) or a single but very large filter (7x7).

III. Design Flow

To avoid lengthy hardware compile times and achieve good coverage we pre-compile thousands of different variations of our parameterizable signal processor. This strategy becomes increasingly important since next generation FPGAs (Virtex-7) require tens of hours and gigabytes of memory to finish compilation. Our architecture, for example, needs 16 hours and 19 GB RAM for a Xilinx Virtex-7 target device. The generated FPGA configurations are collected in a repository for later retrieval. If necessary these FPGA configurations can be requested and configured at run-time. We have automated the Xilinx ISE tool flow and wrote a custom middleware to distribute Xilinx ISE compilation jobs to our cloud computing facilities. Currently, we have access to 40 nodes, 960 cores and a total of 2.8 TB memory. This powerful setup allows us to automatically compile up to one thousand FPGA configurations simultaneously.

IV. Application Mapping

It is, for example, possible to map 2D-filters with different kernel sizes to one or more DSEs. If enough resources are available then multiple pixels can filtered per cycle. For FPGA implementations this feature is crucial since they must compensate low clock speeds with data parallelism and pipelining.

V. Implementation

We have created a highly parameterizable Bluespec HDL implementation of our DSE architecture and synthesized it for the Virtex 7. All components of our DSE achieve frequencies in excess of 300 MHz. A complete DSE for a 3x3 filter with 4 pixels/cycle, for example, runs currently at 240 MHz. In addition to small DSEs we have created a four lane, 8-core DSE system where each lane has 8-MAC pipelines - see Figure 1. This "fat" DSE system has 1792 MACs and 8 MBit worth of buffers. The theoretical peak performance is 250 GMAC/s. A rough comparison between high- end processors: 4-core TI TMS320C6670 DSP 154 GMAC/s, Tensilica ConnX BBE64 128 GMAC/s, Fermi 589 GMAC/s. Our memory architecture is not cache- but stream-based for real-time processing. Thus if DSE systems and algorithms are well matched (e.g. #MACs) then we can achieve near peak performance.

VI. Conclusion

We introduced RIVER, a reconfigurable pre-synthesized streaming architecture and framework for real-time signal processing on FPGAs. Current state-of-the-art signal processing cores for FPGAs are specifically tailored towards a single task and must usually be compiled from HDL-sources, whereas our repository of pre-synthesized DSEs is instantly available, run-time reconfigurable and programmable on many levels. This feature becomes increasingly important since next generation FPGAs (Virtex-7) require tens of hours and gigabytes of memory to finish compilation. Another advantage of our approach is that DSE users do not require HDL-design skills and tools. Furthermore, we can accelerate application specific tasks by integrating specialized hardware features into computational lanes and reduction stages such as MAC-pipelines and adder-networks. Thanks to our flexible memory architecture, even fairly irregular data access patterns have been successfully realized in a streaming fashion. As a result our architecture *and* framework are a flexible platform for real-time signal processing on FPGAs.