Discrete Fourier Transform Compiler for FPGA and CPU/FPGA Partitioned Implementations[*]

Paolo D'Alberto Peter Milder Franz Franchetti
James C. Hoe Markus Püschel José Moura

Electrical and Computer Engineering
Carnegie Mellon University

Introduction

HW/SW partitioned implementations promise to offer the best of both worlds—the performance and efficiency of HW and the flexibility of SW. This remains an under-tapped paradigm due to its high design complexity, made worse by inadequate support in current tools and the lack of engineers trained in this design discipline. SPIRAL is a fully automatic design generation framework for the linear DSP transform domain. In the past, we have shown that SPIRAL automatically generates platform optimized DSP-transform software that is better than the best available hand-crafted vendor code [3]. Below, we briefly overview our current endeavor to extend SPIRAL to produce automatically high-quality HW only and HW/SW partitioned implementations.

Generating DFT IP blocks

Approach. Given the well-understood regular structure of linear DSP transforms, one can fully capture the available design space in a automatic design synthesis system. Previously, we developed parametrized DFT core generator ([1]) specific for the Pease DFT algorithms. Extending the SPIRAL formula generation and manipulation framework (originally for software only) we completed a general formula-to-hardware synthesis flow that produces, from different formula-level choices, a wide range of synthesizable RTL Verilog implementations, including latency-efficient iterative micro-architectures and throughput-efficient streaming micro-architectures.

Results. Fig. 1 shows area/latency tradeoffs of different Spiral generated HW implementations of DFT-1024, when targeting the Xilinx Virtex-II FPGA. (In the figure, points lower and further left is better.) The implementations differ in the degree of data-path reuse as well as in their underlying algorithmic structure; implementations corresponding to the same family of formulas are show in series. Our tool can produce a design with comparable area and latency as Xilinx LogiCore DFT-1024, as well as the entire range of tradeoff available on the Pareto front. Automatic space exploration makes it feasible to find a Pareto front that comprises implementations from different formula choices.

Generating hardware/software partitioned DFT implementations

With the above extension, Spiral is capable of generating both, software and hardware implementations of signal transforms. This makes it possible to tackle the more complex problem of generating and evaluating SW/HW partitioned implementations. We performed first experiments in this direction, again for the DFT, on a Xilinx Virtex-II-Pro FPGA with an embedded PowerPC fixed-point processor running at 300 MHz. The motivation for partitioning is in the potential of achieving SW-like flexibility while improving both, performance and energy efficiency.

Approach. The approach to SW/HW partitioning is in concept simple. We decide on one or several DFT cores, i.e., sizes, to be mapped to HW. The SW uses these cores throught a specific interface. Then we generate an entire library of DFTs (say, all two-power sizes) that may use the HW cores, if it is beneficial, and performs the (remaining) computation in SW.

[*] This work was supported by DARPA through the Department of Interior grant NBCH1050009 and by NSF through awards ACR-0234293 and ITR/NGS-0325687.

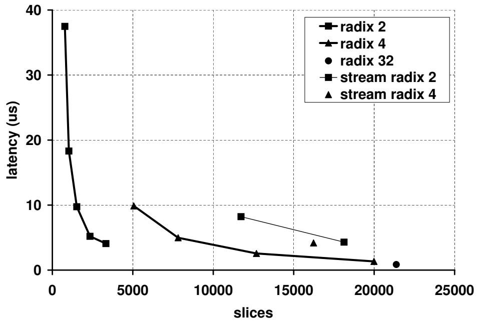

Figure 1: Area/latency trade-off of various different Spiral generated HW implementations of the DFT of size 1024.

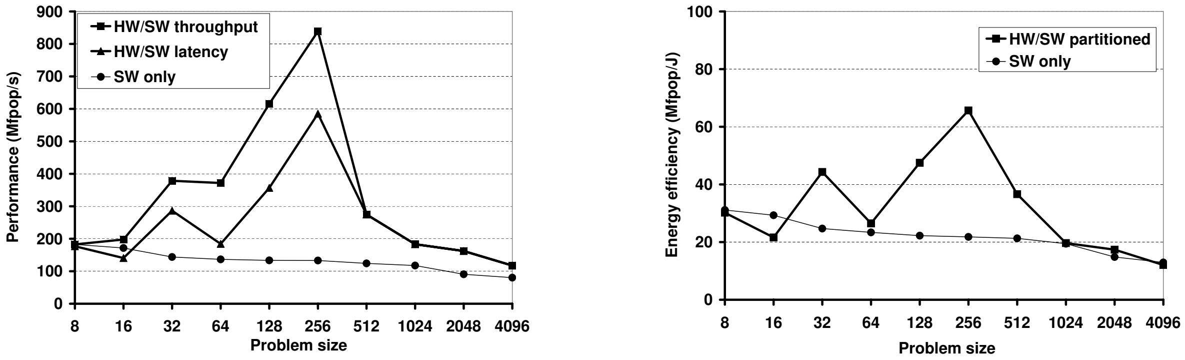

Figure 2: Evaluation of Spiral generated HW/SW partitioned DFT implementations for two-power sizes $4 \leq n \leq 4096$. Performance (left) and energy efficiency (right). Higher is better. The DFTs of size 32 and 256 are mapped to HW.

As HW-DFT basic blocks, we deploy throughput optimized cores due to the structure of the fast Fourier transform (FFT), which decomposes a large DFT into parallel stages of smaller DFTs. When these cores are pipelined in HW, expensive copy operations can be overlapped with computations for improve performance. Further, we allow the HW DFT cores to be also used for smaller sizes. This is possible because of the DFT algebraic properties and we produce what we call "virtual DFT cores."

Results. Fig. 2 shows preliminary experimental results with Spiral generated SW/HW partitioned DFT implementations. First, we generated a DFT SW library (16-bit fixed point) across a range of two-power sizes. The performance achieved is between 100 and 200 Mfpop/s[1] and the energy efficiency (physically measured) around 20 Mfpop/J).

Next we mapped HW DFT cores of size 32 and 256 to hardware and allowed the virtual core sizes of 16, 64, and 128. The SW/HW partitioned library subsequently generated has a 50–500better performance than the SW only solution. The energy efficiency is also improved as shown.

References

[1] G. Nordin, P. Milder, J. Hoe, and M. Püschel. Automatic generation of customized discrete Fourier transform IPs. In *Proceedings of the 42nd Annual Conference on Design Automation*, 2005.

[2] Markus Püschel, José M. F. Moura, Jeremy Johnson, David Padua, Manuela Veloso, Bryan W. Singer, Jianxin Xiong, Franz Franchetti, Aca Gačić, Yevgen Voronenko, Kang Chen, Robert W. Johnson, and Nick Rizzolo. SPIRAL: Code generation for DSP transforms. *Proc. of the IEEE*, 93(2):232–275, 2005. Special issue on *Program Generation, Optimization, and Adaptation*.

[3] Spiral web site. www.spiral.net.

[1] Mega fixed-point operations per second.