"Close-to-Concept Coding for Configurable Computing"

Henk Spaanenburg, Joe Thompson
Advanced Principles Group, Inc.
1045 Elm Street, Suite 201, Manchester, New Hampshire 03101

Abstract:

Breakthroughs in Field-Programmable Gate-Array (FPGA) technology can provide orders of magnitude improvements in computing performance, power, and cost over current state-of-the-art super-computing systems. This, in turn, has increased the need for programming tools[1] to support the development of performance sensitive applications. **A visualized macro-parallel-pipelines model of computation has been identified by APG to be extremely useful for transparent technology update/upgrade, as well as for performance in embedded-super-computing-class signal processing systems. We will describe an approach to design and develop the appropriate visual (close-to-concept) programming interface to "virtual" Programmable Signal Processing (PSP) middleware. The parameterized "virtual" PSP architecture will be mapped effectively (once) into the underlying FPGA technology. The system application programmer will, in essence, only see a graphical close-to-concept interface to the "virtual" PSP middleware. This approach will result in a more conventional, and therefore more productive, signal processing programming environment for embedded super-computers.**

Presentation Overview:

FPGA-Based Embedded High Performance Computing. The introduction of Field-Programmable Gate Arrays (FPGAs) during Sony Corporation's Dr. Tsugio Makimoto's defined third wave makes "softer" flexible hardware possible. Makimoto's third wave is triggered by the use of reconfigurable hardware as a basis for new computational paradigms[2]. Reconfigurable computing has come of age for High-Performance Computing (HPC). SRC Computers with their MAPstation, Cray (after the acquisition of OctigaBay Systems Corp.) with their XD-1, and SGI with their Reconfigurable Application-Specific Computing (RASC) technology have entered the HPC "Super-Computing" domain. APG has applied FPGA technology in the development of embedded super-computers based on its DOE-sponsored Halcyon RCB.

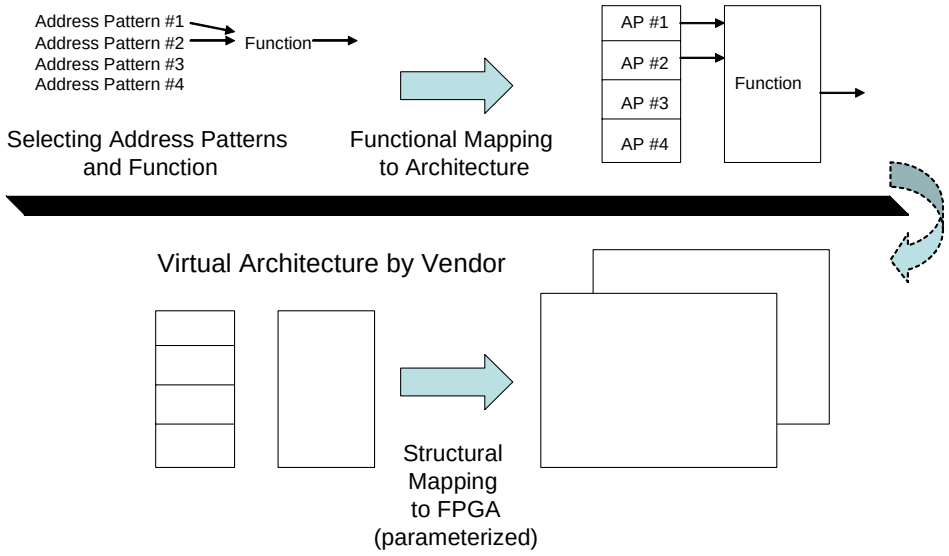
Programmable Signal Processor (PSP) Architecture. Previous experience in developing a Programmable Signal Processor (PSP)[3] has shown that for signal and image processing a macro-parallel-pipelines organization is sufficient and highly effective. The original GE PSP implementation utilized board-level processing elements in a parallel pipelines architecture to provide high throughput, real-time signal processing for complex multimode radar systems. The PSP was unique in its reliability aspects and visual programming approaches. APG is updating the PSP concept in light of dramatic functional density advances since the 1980's.

---

[1] Morris, K., "Saving Supercomputing with FPGAs, What We'll Do When We Hit the Wall", FPGA and Structured ASIC Journal (www.fpgajournal.com), November 22, 2005
[2] Manners, D., T. Makimoto, "Living with the Chip", Chapman & Hall, 1995
[3] H. Spaanenburg, "PSP/SVP - An Advanced VLSI Programmable Systolic Signal Processor", IEEE International Conference on Communications, June 1986 (Invited paper).

Virtual PSP Architecture Middleware. In the specific case of the "virtual" PSP, the selected (macro-parallel-pipelines) computational architecture represents a form of virtual processor architecture middleware into which the signal processing system application will be mapped. Most of the APG developments will take place in performing the highly efficient mappings of a parallel pipelines model of computation (MOC) as a virtual processing architecture onto the selected underlying FPGAs. The APG's "virtual" PSP middleware approach will produce an effective implementation of an architectural configuration[4] that will lend itself perfectly to embedded super-computing processing realizations. In addition, it will also be in line (updateable, upgradeable) and compatible with most FPGA-based device architectures.

Visual Flow Programming. By graphically (see below) representing the data flow through a parallel pipelines system, by indicating the selected addressing patterns for the respective buffers and by indicating accesses to corner-turn memories, etc., we are developing a programming environment for state-of-the-art FPGA-based high performance super-computers that is conceptually closer to traditional signal processing flows. This visualization approach will be more manageable than current C-based programming approaches where most of the data flow information has been obfuscated. From a system development point-of-view, we have removed the inefficiencies of C-based FPGA programming by providing a much simpler middleware interface, and by hiding the act of careful mapping onto FPGAs of the "virtual" PSP architecture.



Technical Program Areas of Interest:

- Automated Design Tools
- Advanced Middleware and Software Technology
- Mapping and Scheduling of Parallel and Real-Time Applications
- Signal Processing Techniques
- ASIC and FPGA Advances
- Parallel and Distributed System Architectures

[4] Ennis, W.G., H. Spaanenburg, "High Performance Programmable Signal Processors", NAECON'85, May 1985