A Survey of Multi-Core Coarse-Grained Reconfigurable Arrays for Embedded Applications

Justin L. Tripp, Jan Frigo, Paul Graham
Los Alamos National Labs
Email: {jtripp, jfrigo, grahamp}@lanl.gov

I. Introduction

Traditionally embedded computing tasks have been handled by DSPs as well as FPGAs. These provide two good alternatives to embedded processing in the areas of power, speed and configurability. However, they are still only two regions in a large multi-dimensional space. FPGAs provide low-level fine-grained parallelism with a degree of performance achievable through multiple levels of parallelism. DSPs have higher clock speeds and have lower power requirements, but lack the flexibility of an FPGA. Still, there are regions in the design space that can be explored that fit between DSPs and FPGAs. These are Coarse-Grained Reconfigurable Arrays (CGRAs).

CGRAs have several advantages over traditional DSP and FPGA approaches. First, most strive for ease of application development, such as a high-level language (HLL) approach for their tools. For example, rather than constructing a design in a hardware description language such as VHDL or Verilog, several of the described platforms use HLLs such as C or C++. Second, CGRAs are able to achieve high-performance and lower power per operation when compared to FPGAs and DSPs. CGRAs accomplish this through higher levels of raw parallelism as compared to DSPs and more efficient use of silicon per operation as compared to FPGAs.

CGRA systems are multi-core systems that fit into two different categories: processor-centric arrays or clusters, and hardware-centric medium-grained processor arrays (MGPA). Processor-centric arrays or clusters are made up of individual processors connected via programmable interconnect such as MIT's RAW [1], Clearspeed's CSX600 [2], IBM's Cell Broadband Engine and Ambric's AM2000 reconfigurable processing array (RPA) [3]. Hardware-oriented processing arrays or medium-grained processing arrays (MGPA) have an array structure with the basic level of computation at a higher level of abstraction than a gate. Examples of MGPAs are Elixent's reconfigurable arithmetic processors, MathStar's FPOA, and PACT's XPP.

Each of these approaches have particular strengths and weaknesses. The processor centric approaches can be programmed using HLLs such as C with some extensions. The MGPAs require tools for programming hardware, similar to an FPGA development chain. This is likely due to the more flexible nature of the individual processing elements. Next we will discuss several particular CGRA examples and how they were successfully applied to processing problems.

II. Processor Centric Arrays

Processor Centric Arrays are based on a processing element that operates on data sizes typically associated with microprocessors and are programmed in a manner similar to typical microprocessor development. For this class of CGRAs, we examined Clearspeed's CSX600, IBM's Cell Broadband Engine, and the Stretch S5 processor.

Clearspeed's CSX600 is a System-on-a-Chip architecture with multi-threading, SIMD, and Very Long Instruction Word (VLIW) parallelism. There are three concurrent processing units: a 32-bit RISC control processor, a SIMD processor array with 96 processing elements (PEs), and an I/O processor for transferring data between the control processor and the SIMD array. The chip operates at 250 MHz and dissipates 10 W.

The RISC control processor manages program flow and data transfer. Each SIMD processor is a VLIW core with a 4-stage 32-bit/64-bit multiply-add pipeline, a divide/square-root unit, and 6 KB of local SRAM. The PEs and RISC processor share a bus that has access to 128-KByte SRAM as well as 1 GByte of DDR2 SDRAM. The PEs can indirectly address into different memories using arrays.

The CSX600 was applied to a one-dimensional single-precision advection problem. The term *advection* describes the transport of a scalar quantity in a vector field such as the movement of silt in a river. On the CSX600, each PE is responsible for several cells in the vector field. In order to calculate an advection value, the values of its two neighbors from the previous time step are required which necessitates some level of communication between PEs. Taking advantage of the highest level of parallelism, the Clearspeed was on par with an AMD Opteron 275 (2.2 GHz). However, the Clearspeed board accomplished the same amount of work for only 25 Watts of power – one quarter the power of the Opteron.

The Cell Broadband Engine (BE) is another processor centric array with a SIMD-centered architecture with an IBM 64-bit PowerPC processor and eight optimized Synergistic Processing Elements (SPE). The Cell processor provides more than an $8\times$ improvement on compute capability compared to traditional processors due to the SIMD SPE engines. Each of the 8 SPE engines are dual-issue, have dedicated resources (for registers and DMA) and provide up to 4-way SIMD for utilizing data parallelism. The processor also contains a data ring for intra-processor and external communications. The system is integrated by a coherent on-chip element interconnect bus.

As shown in [4], the Cell was applied to a matched filter algorithm. The matched filter takes hyperspectral data and matches it against a particular signature. The signature is a vector of coefficients that represent the spectral reflection or transmission of a particular material. This algorithm mapped well to the Cell and by using the SPEs in parallel as well as 4-way SIMD floating-point instructions, it was able to achieve an 8 times speedup over a microprocessor. The Cell consumes about the same power as a high-end microprocessor (80 Watts), but provides higher performance.

The Stretch S5 uses a single specialized, high-performance RISC processor core—the 300-MHz Xtensa core with 16- and 24-bit instructions. The core supports a memory managed unit (MMU) with a translation look-aside buffer (TLB). The RISC processor has a Instruction-Set Extension Fabric (ISEF) that can implement a specific loop or set of instructions in parallel without the use of low-level assembly language. Stretch's Integrated Development Environment (IDE) tools suite is a graphical interface consisting of: a compiler, debugger, assembler, profiler, linker and editor. Stretch's C/C++ compiler programs the processor and automatically configures the ISEF with application-specific instructions.

We implemented a C/C++ image processing application for a high speed camera on the S5 processor. The algorithm included a high pass filter, interframe difference, and cross correlation calculation. We report the run-time performance and power consumed on the S5 development board. Maximum rated power for the S5 is 3 Watts.

III. Medium-Grained Processing Arrays

Medium-Grained Processing Arrays are reconfigurable and programmable arrays that operate on inputs larger than four bits. Most arrays target DSP applications and are able to achieve better power per operation due to specialized silicon.

The MathStar Field Programmable Object Array (FPOA) chip has 400 Silicon Objects—256 ALUs, 64 Multiply Accumulates (MACs), and 80 Register Files (RFs). The FPOA Silicon Objects can have a semi-autonomous nature. For example, each ALU has a program memory of eight instructions that can contain both ALU operations and communication instructions. The control path is bit-wise granular and guides program execution while data is moved using the 16-bit data path. Multiple objects can be combined to create wider data paths. Thus, instructions are the mechanism that tie the independent control and data paths together within the array.

Speed and power estimates for a dot product operator (as part of an FIR application) and the sum of an absolute value of a difference operator (as part of an image clustering algorithm) are reported. These designs mapped to the hardware running at 1 GHz. The power estimates are data dependent, but require less than 10 Watts for the two operators. The control states are implemented with ALU objects. The data-flow operations use a mix of ALU, register and MAC objects.

A major advantage of the FPOA is that the operating frequency is deterministic and can be as high as 1 GHz. The FPOA is clock-cycle based and its interconnect structure is deterministic—there is no physical analog timing closure required. Unused objects can be turned off to save power. Migration to a larger capacity or newer generation device requires no re-design, as long as the required object arrangement is a subset of the target device.

Some disadvantages of the architecture are that the design, simulation, and mapping to hardware is relatively time consuming. Mapping to hardware is a manual process, and thus time consuming. An FPOA is a medium grained device and, thus, it is not as flexible as a fine-grained device such as an FPGA. Also, it has limited local on-chip memory. The ALUs are 16-bit and thus must be cascaded to form larger data sizes—using up additional silicon resources.

IV. Summary

LANL has applications that range from low-power embedded sensor network nodes to high-performance embedded signal and image processors that are airborne or in orbit. Due to the wide spectrum of applications of interest, the four architectures described have specific strengths and weaknesses to consider. These are discussed below.

The Clearspeed architecture is best suited to applications with regular, predictable data access patterns. With hardware floating point units, it is capable of significant floating point computation as long as the data can be obtained and transferred into the SIMD processors' local memories. It does not perform well on large working sets and irregularly accessed data.

Similarly, the Cell is capable of floating point computation, but can more easily parallelize operations since the SIMD operations are four-way instead of 96 way. Also, very similar to the CSX600, the Cell requires data transfer to be micromanaged within the users code. In the power domain, Cell and CSX600 differ greatly, since the Cell requires an order of magnitude more in power.

The Stretch device provides good performance for a low-power device with an easy development flow. The next generation promises increased processing and I/O performance.

The MathStar FPOA is able to take advantage of high, deterministic clock rates to achieve higher performance and can reduce power by turning off unused objects. The speed advantage comes at a cost in flexibility and development time.

References

[1] M. Taylor et al, "The Raw Processor – A Scalable 32-bit Fabric for Embedded and General Purpose Computing," in *Hotchips XIII*. IEEE, August 2001.

[2] ClearSpeed Technology, *CSX600 Datasheet*, 2006, http://www.clearspeed.com.

[3] Ambric Reconfigurable Processor Arrays, *Ambric Data Sheet*, 2007, http://www.ambric.com.

[4] Z. K. Baker, M. B. Gokhale, and J. L. Tripp, "Matched Filter Computation on FPGA and Cell and GPU," in *Proceedings 2007 IEEE Symposium on Field Programmable Custom Computing Machines*. IEEE CS Press, April 2007.