A Parallel Programmable Energy-Efficient Architecture for Computationally-Intensive DSP Systems

Bevan M. Baas
Department of Electrical and Computer Engineering
University of California, Davis

Abstract

An architecture that is well matched to DSP system workloads, enables high-throughput and high energy-efficiency, and is well suited for advancing VLSI fabrication technologies is presented. These processing systems consist of large numbers of simple uniform programmable processing elements communicating asynchronously through a configurable 2-D mesh network that connects adjacent processors at full clock rates. Early estimates predict an area density of 0.15 mm^2 per processor in 0.13μm CMOS. Results from mapping a 16-tap FIR filter over 85 design configurations show a factor of 9 variation in throughput per processor and validate the efficiency of the proposed processor granularity.

1 Introduction

As advancing semiconductor fabrication technologies enable more complex systems to be integrated onto a single chip, it is prudent to consider whether new processor architectures make better use of silicon resources. Application-specific processors for complex DSP system workloads normally make use of many parallel functional units, but common programmable DSP processors typically use computer architectures similar to those used by general-purpose processors.

The architecture presented here meets the needs of both high-throughput and energy-constrained computationally-intensive DSP workloads.

2 Primary Design Goals

Our proposed architecture targets four key goals.

Well matched with DSP system workloads. Many DSP system workloads are comprised of a cascading of various DSP tasks and may not be well suited to the classic processor-and-large-memory architectural style typically used in programmable DSP processors. An example of a computationally-intensive emerging application is the baseband processor for an 802.11a wireless LAN (5 GHz, 54 Mbps). Figure 1 shows an 802.11a transmitter data flow diagram which exhibits this serial concatenation of independent tasks. An architecture that more naturally maps target workloads will likely result in a simpler algorithmic mapping effort and a more efficient implementation.

High-throughput. General-purpose and DSP processors typically utilize a decreasing percentage of die area for datapath circuits. We seek an architecture that permits a significant proportion of the die to be dedicated to datapath circuits, and enables their effective utilization.

Energy-efficient. Many DSP algorithms require a relatively small amount of local memory. Because communication energy can often dominate computation energy, an energy-efficient architecture minimizes access and communication energy for data and instruction storage.

Address the opportunities and challenges of future VLSI fabrication technologies. In the near future, CMOS fabrication technologies will enable die with over 1 billion devices and clock rates over 10 GHz, and will require foundry NRE costs over $1 million per design. We desire an architecture that simplifies the design of a large processing system, eases the design of very high clock rate processors, and is programmable and potentially reconfigurable.

3 Key Features

Our proposed architecture consists of a large number of simple uniform processing elements operating asynchronously and connected through a recon-

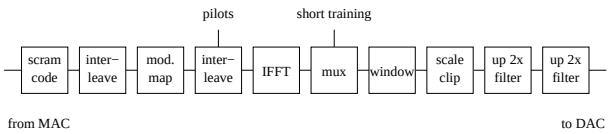

Figure 1: 802.11a wireless LAN transmit path dataflow diagram

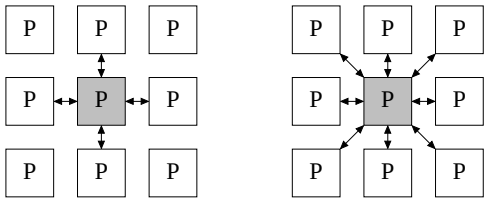

Figure 2: Processors in a 2-D mesh that connect to four (left) or eight (right) nearest neighbors

figurable network. It can also be viewed as a highly parallel MIMD system.

Simple processing elements. Atomic processing elements in existing parallel processing systems range in complexity from simple combinational logic blocks in FPGAs to complex processors and distributed memory in massive parallel supercomputers. Our design utilizes a single-issue processor with very small local memories, a high clock rate, and very small die area—these criteria affect every aspect of the design.

Reconfigurable 2-D mesh network. The architecture connects processors via a 2-dimensional mesh grid because it maps well to planar integrated circuits. To maintain link communication at full clock rates, interprocessor connections are made to nearest-neighbor processors only. We are currently evaluating whether overall system performance will be greater with connections to four or eight nearest neighboring processors, as shown in Fig. 2.

Completely asynchronous clocking. Each processor has its own digitally programmable clock oscillator. Such structures in previously fabricated chips were found to have quite modest area and power [2]. There are no global frequency or phase-related signals, and the system is globally asynchronous and locally synchronous (GALS) [3].

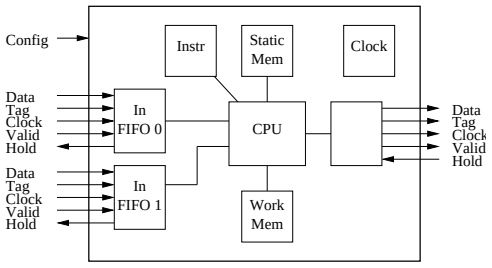

Figure 3: Processor block diagram

4 Processor Architecture

Figure 3 shows the major blocks inside each processing element. Two input interfaces and an output interface require asynchronous FIFO buffers to transfer and buffer data across processor boundaries.

The first implementation of the architecture has a 16-bit fixed-point datapath with a 16x16-bit multiplier-accumulator. General arithmetic instructions select two input operands from two of the following six sources:

- Working memory
- Static memory
- Accumulator
- Input FIFO buffer 0
- Input FIFO buffer 1
- Immediate instruction field

and can select one of the following four destinations:

- Working memory
- Static memory
- Accumulator
- Output FIFO buffer

Processors contain hardware support for several programmable address generators and zero-overhead looping. The processor's input interface allows it to receive data from two of four (or eight) neighboring processors. Output ports send data to any combination of the four (or eight) neighboring processors using a single FIFO.

5 Performance Estimates

Area and throughput estimates

Early area estimates predict each processor will occupy approximately 0.15 mm^2 in 0.13 μm CMOS. Because many features of the design are chosen to maximize clock rates, we expect processors to operate at frequencies among the highest possible for a digital system designed using a particular design approach and fabrication technology. With advancing semiconductor fabrication technologies, the number of processors will increase as the square of the scaling factor and clock rates will increase approximately linearly—resulting in a total peak system throughput that increases as the *cube* of the technology scaling factor. Table 1 summarizes area and performance estimates.

CMOS Tech	Processor Size	# Procs per Chip	Relative Clock	Rel Total Sys Thruput
0.130 μm	0.15 mm^2	540	1	540
0.090 μm	0.08 mm^2	1000	2	2000
0.045 μm	0.025 mm^2	3200	4	12,800

Table 1: Estimates for a 10 mm $\times$ 10 mm chip implemented in various semiconductor technologies

Results from mapping a 16-tap FIR filter to a grid of processors

To better understand how well algorithms map to the proposed grid of processors, a 16-tap FIR filter was mapped by hand using a number of topologies and 85 different design configurations [4]. Figures 4–7 show examples for 4-way connected processors using three major topology types. Figure 7 shows a "U" type topology utilizing an 8-way interconnection network.

In general, the 85 different algorithmic mappings of the 16-tap FIR resulted in unique combinations of numbers of required processors and achieved throughput. Although most of these mappings are parallelized at a much finer granularity than optimum, they provide a thorough exploration of the algorithmic space from a single processor at one extreme to 58 processors and one sample per clock cycle at the other extreme. The 85 data points are plotted in Fig. 8.

To examine the efficiency of the mappings, Fig. 9 shows the same data but with normalized throughput (throughput/processor) on the vertical axis. The single-processor case clearly makes the best use of hardware—it is a factor of nine times more effective than the least efficient case. This qualitatively shows the performance and efficiency gains possible when mapping algorithms onto small numbers of modest-complexity processors rather than the very fine granularity of previous systolic and wavefront array processors [5].

6 Challenges

The proposed architecture presents some significant challenges.

Algorithms requiring large memories. Some DSP algorithms require more working memory than the proposed several hundred words per processor. There

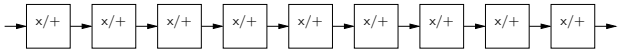

Figure 4: "I"-type FIR dataflow diagram

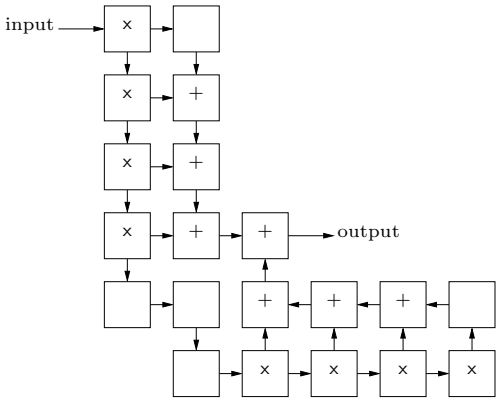

Figure 5: "L"-type FIR dataflow diagram

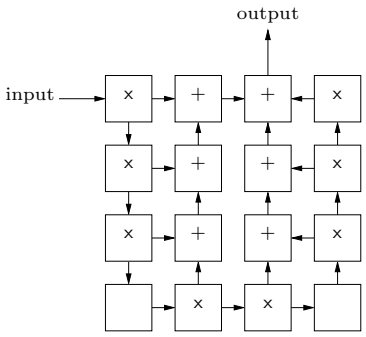

Figure 6: "U"-type FIR dataflow diagram

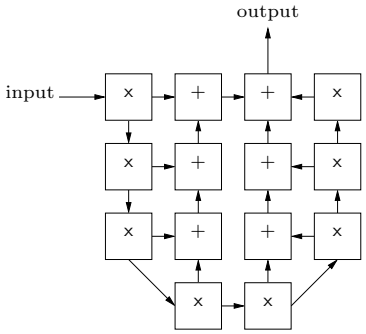

Figure 7: "U"-type FIR dataflow with an 8-way interprocessor connection network

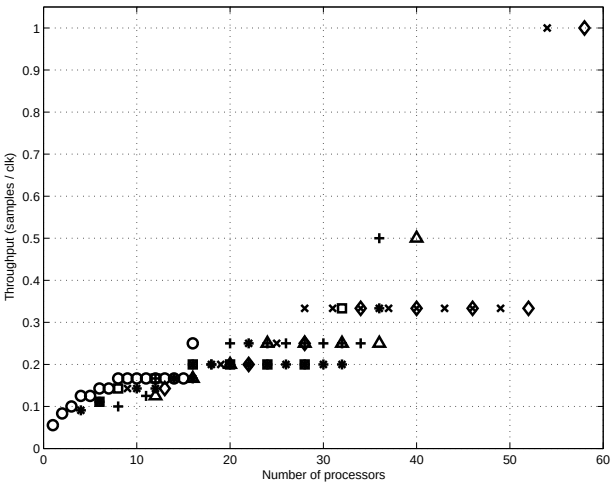

Figure 8: Number of processors vs. throughput for a 16-tap FIR filter

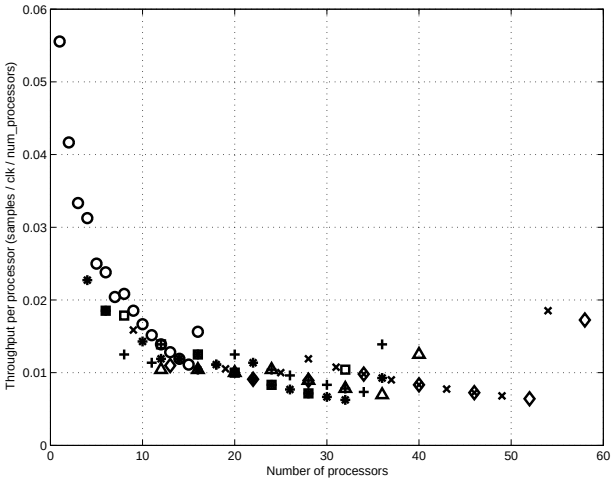

Figure 9: Number of processors vs. throughput per processor for a 16-tap FIR filter

are several potential solutions to this problem: 1) Redesign the algorithm to partition data so that multiple processors each contain relevant data over some period of the computation. Data is then flowed through the array and processors have access to needed data as it flows by. 2) Program processors to serve as memory decoders and memories; processors serving this purpose may be unable to perform other useful computation. 3) Embed discrete memory arrays in the grid of processors.

Parallelizing algorithms. Most DSP algorithms are parallelizable but may require significant effort to map.

Minimizing unused processors. Inter-processor network limitations can result in large numbers of processors (30-50% is common) being unusable for useful computation. This is an unavoidable downside of the proposed approach but the cost can be mitigated by the small size, great number, and high processor speed achieved by keeping the interconnection network simple.

7 Comparison With Similar Work

Systolic processors [6] contain synchronously-operating processors which "pump" data regularly through a processor array. Systolic processing elements receive and send data in a highly regular manner [7] which is decidedly different from the irregular and asynchronous communication in the proposed approach.

Wavefront array processors [8] are similar to systolic processors but rely on dataflow properties for inter-processor data synchronization. Previous designs were optimized for simple and regular single algorithm workloads such as matrix operations [5] and image processing kernels [9] [10]. Although the wavefront architecture is the previous work most similar to the proposed architecture, it differs in a number of significant ways including: the lack of optimization for complex DSP tasks; processing elements communicate through a fixed network, not a reconfigurable network; a very fine granularity for DSP task computation (generally at the multiply or add level); and very little discussion in the literature of asynchronously-operating processing elements.

Smart Memories [11] processing elements contain a 64-bit processor with two integer clusters, one FPU cluster, 128 KB of memory, and a dynamically-routed crossbar. Routing among tiles is packet-based through a dynamically-routed network. The area required per processor is equivalent to 10.6 mm^2 per processor in 0.13 μm CMOS.

The Oxygen project's RAW architecture [12] tar-

gets more general-purpose workloads, and specifies tiled processors with large instruction and data memories (32 KB IMEM and 32 KB DCache) and sophisticated network routers (including 64 KB SMem) in each processor. A fabricated chip in 0.15 μm CMOS occupies 330 mm^2 and yields 16 processing elements.

The Pleiades [13] chip contains a microprocessor and a variety of computational units connected by a hierarchical configurable network. It contains multiple computational granularity levels on the same die which results in a non-regular layout. It utilizes a GALS clocking scheme.

The Imagine [14] chip is organized as 8 clusters of 6 ALUs executing VLIW instructions that address a large hierarchical memory. A 256 mm^2 fabricated die contains 8 clusters in a 0.15 μm CMOS fabrication technology.

The datapath of the Rapid [15] processor comprises a 1-dimensional linear array of functional units connected through a programmable interconnect structure. Similarly to the proposed architecture, it also targets DSP applications and utilizes a fixed-point datapath.

8 Acknowledgments

This work was supported by a gift from Intel Corporation and by a UC Davis Faculty Research Grant.

References

[1] J. Thomson, B. Baas, E. M. Cooper, et al. An Integrated 802.11a Baseband and MAC Processor. In *IEEE International Solid-State Circuits Conference*, volume 45, pages 126–127, 451, 2002.

[2] B. M. Baas. A low-power, high-performance, 1024-point FFT processor. *IEEE Journal of Solid-State Circuits*, 34(3):380–387, March 1999.

[3] D. M. Chapiro. *Globally-Asynchronous Locally-Synchronous Systems*. PhD thesis, Stanford University, Stanford, CA, October 1984.

[4] H. C. Chang and B. M. Baas. Mapping an FIR filter to a 2-dimensional mesh of processors. Technical Report ECE-CE-2003-1, Computer Engineering Research Laboratory, ECE Department, University of California, Davis, February 2003. `http://www.ece.ucdavis.edu/cerl/techreports/2003-1/`.

[5] S. Y. Kung. VLSI array processors. In *IEEE ASSP Magazine*, pages 4–22, July 1985.

[6] H. T. Kung. Why systolic architectures? In *Computer Magazine*, January 1982.

[7] H. T. Kung. Systolic communication. In *International Conference on Systolic Arrays*, pages 695–703, May 1988.

[8] S. Y. Kung, C. E. Leiserson, et al. Wavefront array processor: Language, architecture, and applications. *IEEE Transactions on Computers*, C-31(11), November 1982.

[9] O. Menzilcioglu, H. T. Kung, and S. W. Song. Comprehensive evaluation of a two-dimensional configurable array. In *International Symposium on Fault-Tolerant Computing*, pages 93–100, June 1989.

[10] U. Schmidt and S. Mehrgardt. Wavefront array processor for video applications. In *IEEE International Conference on Computer Design*, pages 307–310, September 1990.

[11] K. Mai, T. Paaske, N. Jayasena, R. Ho, W. J. Dally, and M. Horowitz. Smart memories: A modular reconfigurable architecture. In *Proceedings of the International Symposium on Computer Architecture*, pages 161–171, June 2000.

[12] M. B. Taylor, J. Kim, J. Miller, et al. A 16-issue multiple-program-counter microprocessor with point-to-point scalar operand network. In *IEEE International Solid-State Circuits Conference*, pages 170–171, February 2003.

[13] H. Zhang, V. Prabhu, V. George, et al. A 1-V heterogeneous reconfigurable DSP IC for wireless baseband digital signal processing. *IEEE Journal of Solid-State Circuits*, 35(11):1697–1704, November 2000.

[14] B. Khailany, W. J. Dally, A. Chang, U. J. Kapasi, J. Namkoong, and B. Towles. VLSI design and verification of the imagine processor. In *IEEE International Conference on Computer Design*, pages 289–294, September 2002.

[15] D. C. Cronquist, C. Fisher, M. Figueroa, P. Franklin, and C. Ebeling. Architecture design of reconfigurable pipelined datapaths. In *Conference on Advanced Research in VLSI*, pages 23–40, March 1999.