A 1024-core 70 GFLOP/W Floating Point Manycore Microprocessor

Adapteva, Inc.
1666 Massachusetts Ave, Lexington, MA, USA
Email: {andreas, roman, oleg}@adapteva.com

Abstract

This paper describes the implementation of a software programmable floating point multicore architecture scalable to thousands of cores on a single die. A 1024 core implementation at 28nm occupies less 128mm^2 and has a simulated energy efficiency of 70 GFLOP/Watt with a peak performance of 1.4 TFLOP. The aggressive claims are supported by a 65nm silicon proven 16-core version of the same design with measured efficiency of 35 GFLOPS/Watt.

Architecture Overview

The multi-core architecture proposed in this work was designed to accelerate signal processing kernels requiring floating point math, such as large FFTs and matrix inversions. Examples of embedded applications requiring floating point math include: synthetic aperture radar, ultra sound, cellular antenna beam forming, and graphics processing. Figure 1 shows a block diagram of the architecture, with 16 processor tiles arranged as a 4 x 4 array.

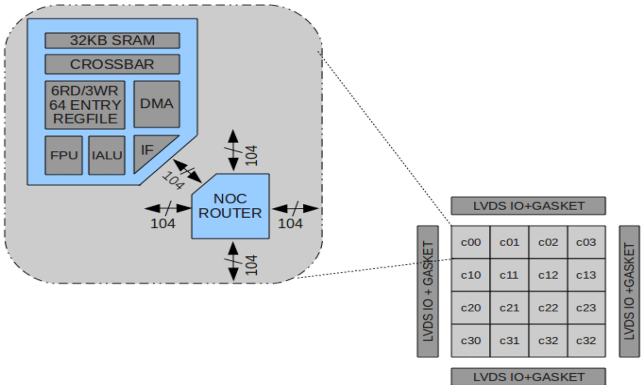

Figure 1: Multi-core Floating Point Accelerator Architecture

The tiles are connected through a 2D mesh network. Each processor tile contains a full routing cross-bar, a custom dual issue floating point RISC CPU, a DMA engine, and 32KB of multi-bank SRAM. All cores are ANSI-C programmable and share a single unified 32 bit flat address map. The processor cores can be programmed and run completely independently of each other or can work together to solve larger problems. An important architectural decision was the replacement of the traditional power hungry cache hierarchy with a distributed flat memory model that offers a total memory bandwidth of 32GB/s per processor core. The high bandwidth memory architecture and the flat unprotected 32 bit memory map lets up to 4096 cores communicate with each other directly with zero startup communication cost.

Network-On-Chip

The performance of a Network-On-Chip depends on a number of different factors such as: network topology, routing algorithms, packet strategy, buffer sizes, flow control, and quality of service support [6]. The proposed mesh NOC, shown in Figure 2, takes advantage of spatial locality and an abundance of short point-to-point on chip wires to send a source address, destination address, and data in parallel on every clock cycle.

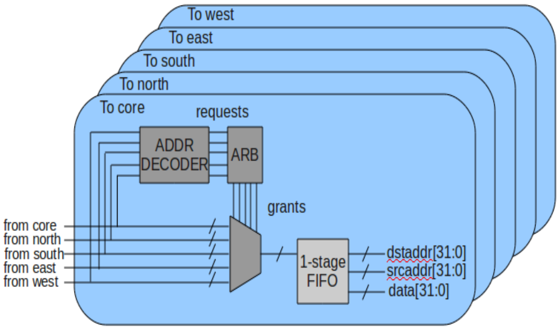

Figure 2: Network-On-Chip Architecture

The address inefficiency overhead of sending an address on every transaction was compensated for by a significantly simpler NoC router design and smaller FIFOs. On write transactions 32GB of data can flow into and out of each routing node on every clock cycle. The mesh throughput is balanced with the load/store throughput of the core, allowing the processor core to store data from its register file directly into adjacent cores memory without stalling the CPU pipeline. Round robin arbitration at each crossbar node ensures fairness in bandwidth allocation and together with the single cycle transaction design guarantees that the NoC is free of deadlocks. The effectiveness of the Network-On-Chip was tested in implementing multicore versions of 1024 point FFT and variable size matrix multiplication routine (SGEMM).

Chip Implementation

A chip product based on the above proposed architecture was implemented in a 65nm triple-Vt high speed CMOS process, and contains 40 million transistors, staggered pad-ring wire bonding, and is packaged in a 324 ball 15x15mm BGA package. Figure 3 shows the silicon evaluation platform for the 65nm chip. The evaluation platform hooks up to a standard GNU debugger based tool chain running on a Linux distribution. The chip daughter card is located at the right side of the picture with two external power supplies, a 1V core supply and a 2.5V IO supply.

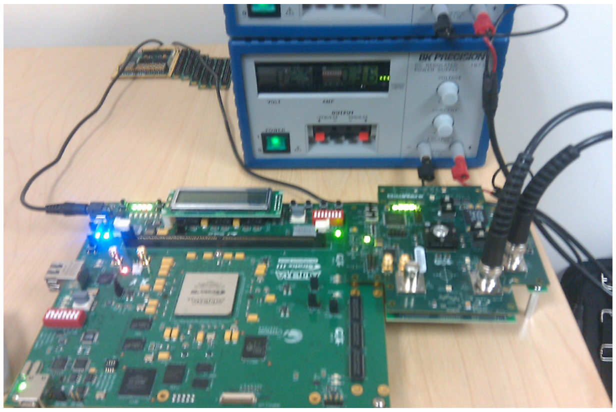

Figure 3: 65nm Silicon Evaluation Platform

Comparison to State of the Art

Table 1 compares this work to previously published work, demonstrating the merits of this multicore architecture approach for low power floating point applications. Table2 demonstrates the transistor efficiency advantage of this NoC compared to previous publications.

	[2]	[3]	[4]	[5]	This work
Process (nm)	65	90	45	45	65
Frequency (MHz)	3130	850	2000	648	1000
Cores	80	64	48	8	16
Area (mm^2)	275	n/a	567	16	11.5
Transistors(Millions)	100	615	1300	n/a	40
Performance(GFlops)	1000	n/a	n/a	36	32
Core Power (W)	200	10	125	0.85	0.35@500Mhz
Area/Core (mm)	3.43	n/a	11.8	2	0.5
Watt / (GHZ*Cores)	0.8	0.18	1.3	0.16	0.04

Table 1: Comparison of floating point multicore processors

	[7]	This work
Process	45nm	65nm
Frequency (GHz)	2.0	1.0
Transistors (Thousands)	640	67
Bandwidth (Tb/s/router)	1.28	0.32
Router Latency	4 cycle	1 cycle
Message Startup Cost	n/a	0
Area Efficiency (Tb/s Per Million Transistors)	2	4.8

Table 2: Comparison of state-of-the-art Network-On-Chip

A 1024 Core Microprocessor Implementation

The design discussed in the previous sections was ported to a low power 28nm process technology and completely implemented in layout. Figure 4 shows the layout of the 1024 core microprocessor array and the detailed layout of a single tile within the processor array. The design operates at 700MHz, occupies less than 128mm^2 of silicon area and has an estimated power consumption of 20W. In the lab measurements will be available November 2011.

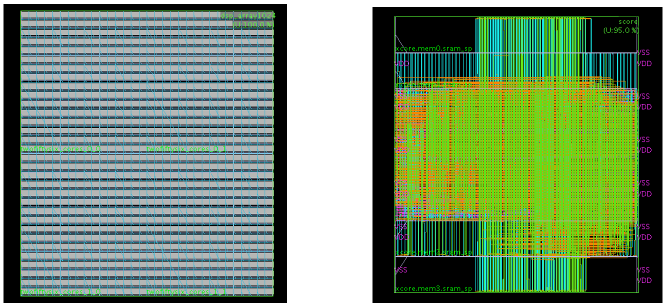

Figure 4: A 1024 core implementation in 28nm

Table 3 shows the power consumption numbers for a number of different processor array sizes at the 28nm process node.

	E16	E64	E256	E1K	E4K
Cores	16	64	256	1024	4096
Frequency MHz	700	700	700	700	700
Max Performance GFLOPS	22	90	350	1400	5700
Max Power	0.3W	1.2W	5W	20W	80W
Total Area (mm^2)	2	8	32	128	508

Conclusions

This paper presents the implementation of a 1024 core C-programmable floating point microprocessor array with an energy efficiency of 70 GFLOP/Watt.

References

[1] A. Olofsson et al., "A 25 GFLOPS/Watt Software Programmable Floating Point Accelerator", ISSCC Dig. Tech. Papers, 2004.

[2] S. Vangal, et al, "An 80-tile 1.28TFLOPS Network-on-Chip in 65nm CMOS", ISSCC Dig. Tech. Papers,2007.

[3] S. Bell et al., "TILE64 Processor: A 64-Core SoC with Mesh Interconnect", ISSCC Dig. Tech. Papers, 2008.

[4] J. Howard et al., "A 48-Core IA-32 Message-Passing Processors with DVFS in 45nm CMOS", ISSCC Dig. Tech. Papers, 2010

[5] Y. Yuyama et al., "A 45nm 37.3 GOPS/W Heterogeneous Multi-Core SoC", ISSCC Dig. Tech. Papers, 2010

[6] Dally W. Principles and Practices of Interconnection Networks, Morgan Kauffman 2003

[7] P. Salihundam et al., "A 2TB/s 6x4 Mesh Network with DVFS and 2.3Tb/s/W router in 45nm CMOS", 2010 Symposium on VLSICircuits