Evaluation of a Low-Power Reconfigurable DSP Architecture

Arthur Abnous, Katsunori Seno†, Yuji Ichikawa‡, Marlene Wan, Jan Rabaey

Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
†Sony Corporation, Japan
‡SHARP Corporation, Japan

Abstract. Programmability is an important capability that provides flexible computing devices, but it incurs significant performance and power penalties. We have proposed an architecture that relies on dynamic reconfiguration of hardware resources to implement low-power *and* programmable processors for DSP applications. In this paper, we evaluate this architectural approach and compare it to other programmable architectures.

1 Introduction

Reducing power dissipation is a key design goal for portable computing and communication devices that employ increasingly sophisticated and power-hungry signal processing techniques. Flexibility is another critical requirement that mandates the use of programmable components. However, there is a fundamental trade-off between efficiency and flexibility, and as a result, programmable designs incur significant performance and power penalties compared to application-specific solutions.

Signal processing applications typically exhibit high degrees of parallelism and are dominated by a few regular kernels of computation that are responsible for a large fraction of execution time and energy. For these applications, we could potentially achieve significant power savings by executing the dominant computational kernels of a given class or *domain* of applications with common features on dedicated, optimized processing elements with minimum energy overhead. The result is a domain-specific processor whose design involves trading off the flexibility of a general-purpose programmable device to achieve higher levels of energy efficiency, while maintaining the flexibility to handle a variety of algorithms within the domain of interest. The Berkeley Pleiades architecture is based on this approach [1]. In this paper we analyze the energy-efficiency of the Pleiades architecture, and we compare it to other programmable architectures.

2 Architectural Evaluation

For the purposes of this study, we chose a set of programmable architectures ranging from general-purpose microprocessors to field-programmable gate arrays. With the exception of the XC4003A FPGA, the chosen architectures are highly optimized for low-power/low-voltage operation. The architectures were evaluated for a set of common DSP benchmarks that included Finite Impulse Response (FIR) filter, Infinite Impulse Response (IIR) filter, and Fast Fourier Transform (FFT).

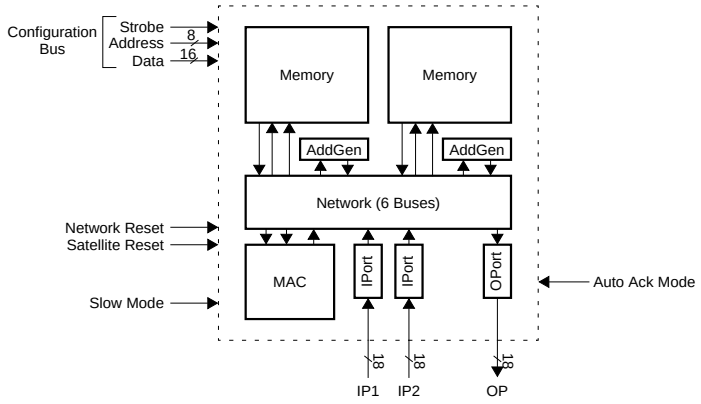

Fig. 1. Block Diagram of the P1 Prototype

2.1 Pleiades

To evaluate the effectiveness of the Pleiades architecture, we developed a test chip named P1. P1 was fabricated in a 0.6-μm CMOS process. Since our main design goal was to reduce power while achieving our target performance, we designed the chip to operate at 1.5V, the lowest supply voltage that provided reasonable performance for our process. The performance and power of P1 were evaluated by analyzing the extracted layout of the chip with the HSPICE circuit simulator and the PowerMill power analysis program.

The block diagram in Figure 1 shows the structure of P1. The main function of P1 is the computation of vector dot products, which are heavily used in speech coding algorithms based on linear predictive coding. All data and address values are 16-bit quantities, and are handled by 16-bit buses and datapaths. The processing blocks employed in P1 include a multiply-accumulate (MAC) unit, two SRAMs, two address generators, two external input ports, and an external output port. The core of the MAC processor is a full-custom design that implements a carry-save, radix-4 Booth array multiplier, a 40-bit accumulator, and an output shifter. Each SRAM block stores 128 16-bit words in a 64-row by 32-column array (two consecutive words per row). The design of the SRAM includes a Precharge Halt Access (PHA) mode that saves power when accessing the elements of a vector. When two consecutive accesses address the two words in the same row using the PHA mode, the bit lines are not precharged for the second access. The address generators are simple counter finite-state machines that provide an address stream for accessing a vector. The communication network consists of six 16-bit buses that can be programmed to connect to any of the processors. The functionality of each hardware resource is controlled by its configuration register(s). These registers are programmed via the global configuration bus by an external microprocessor.

The vector dot product kernel can be mapped onto P1 by programming a total of 11 configuration registers. After configuration, the address generators are triggered by a start token from the input port. This starts a thread that computes the dot products of two vectors residing in the SRAMs. The result is written to the output port. Table 1 show the power profile of this pipeline. The cycle time for this pipeline is 71.4 ns. Each cycle consumes 205 pJ of energy.

Th data for the dot product kernel was used to estimate the performance and energy of Pleiades for the FIR benchmark. The IIR and FFT results were obtained by extrapolating this data. The IIR processor consists of 4 address generators, 4 SRAMs, 2 MACs, and 1 ALU. The FFT processor consists of 3 address generators, 6 SRAMs, 4 multipliers, 6 ALUs, and 4 data-flow-control units.

Satellite	Power	Energy/Cycle
Address Generator	0.07 mW	5 pJ
SRAM (PHA Read)	0.39 mW	28 pJ
MAC	1.5 mW	109 pJ
Network	0.45 mW	32 pJ
Whole Chip	2.9 mW	205 pJ

Table 1. Power Profile for Dot Product on P1 (T_{cycle} = 71.4 ns)

2.2 StrongARM

General-purpose microprocessors represent the ultimate in flexibility. The StrongARM SA-110 microprocessor [2] is a highly optimized low-power processor, and we decided to evaluate it in our study. To measure the energy consumption of a benchmark kernel, we placed the code fragment for that kernel inside an infinite loop, and measured the average current drawn by the microprocessor core while executing that loop on the StrongARM evaluation board. All kernels fit completely in the on-chip caches, so there was no off-chip memory traffic while executing our benchmarks. The number of cycles spent in executing a given kernel was obtained from the StrongARM emulator.

2.3 Texas Instruments DSPs

Many DSP applications are implemented using programmable DSPs which typically resemble general-purpose microprocessors but have extra instructions and addressing modes that improve their performance for DSP algorithms. For this study, we chose two commonly used processors from Texas Instruments: the TMS320C2xx and the TMS320LC54x. The TMS320LC54x is an advanced signal processor that has been designed specifically for low-power operation.

Starting with assembly programs published by Texas Instruments in their application reports [3, 4, 5, 6], we created a set of benchmark programs in assembly language. All of these programs include initialization sections that were excluded in the performance and power calculations. A 3.0V supply voltage was used in these calculations. Energy values were calculated by adding the contributions of all instructions in a kernel using instruction-level power consumption data published by Texas Instruments [7, 8]. The same method was used to calculate the number of cycles spent executing a kernel.

2.4 XC4003A

Field-Programmable Gate Arrays (FPGA) have recently been used to implement high-throughput DSP applications that are beyond the reach of conventional signal processors. We evaluated the Xilinx XC4003A FPGA in this study. The XC4003A is too small for larger benchmarks, so we implemented smaller versions and extrapolated the results. We implemented an 8-bit, 5-tap FIR filter with constant coefficients on our evaluation board and measured its energy consumption directly. The measurements were then extrapolated to obtain energy values for a 16-bit implementation. Since filter coefficients were constant, add-and-shift multipliers were used in the design. This consumes much less energy than general multipliers used in the other architectures. The FIR design is fully pipelined, and produces an output result every cycle. For the IIR study, an 8-bit IIR biquad section was mapped onto a the FPGA and its energy was evaluated by an energy modeling tool for Xilinx FPGAs that has been developed in our group [9]. The input netlists were created using the Hyper synthesis system [10].

2.5 Normalization of Results

The architectures that we studied are implemented in different fabrication technologies and run at different supply voltages. Therefore, to make a meaningful comparison, we had to normalize the energy and delay values to a reference. We chose to normalize all values to our 0.6-μm process from MOSIS at a supply voltage of 1.5V.

Switched capacitance is normalized using Equation 1. Gate capacitance is assumed to represent circuit capacitance.

$$\textit{Capacitance} \propto \frac{A}{T_{ox}} \propto \frac{L^2}{T_{ox}} \tag{1}$$

A is gate area, L is minimum channel length, and T_{ox} is gate oxide thickness. T_{ox} was assumed to be proportional to the native supply voltage of a given process. Normalized energy is then computed using $E = CV^2$ with V_{dd} = 1.5V.

Delay is normalized using Equation 2.

$$\textit{Delay} = \frac{C \cdot V}{I} \propto \frac{L^2 \cdot V}{(V - V_{th})^{1.3}} \tag{2}$$

V is the supply voltage, C is gate capacitance, I is the MOSFET saturation current, and V_{th} is the threshold voltage. Process parameters for all architectures are listed in Table 2.

3 Results

Tables 3 and 4 show a comparison of the architectures we studied for the FIR and IIR benchmarks, respectively. The StrongARM has the worst performance of all because it takes many instructions and cycles to execute a kernel in a highly sequential manner. The lack of a single-cycle multiplier exacerbates this problem. The other architectures have more internal parallelism that allows them to have superior performance. Pleiades and the TI DSPs can execute one FIR tap in a single cycle. Pleiades does much better on the energy scale than the TI DSPs because the DSPs are general-purpose, and instruction execution involves a great deal of overhead. Pleiades, on the other hand, has the ability to create dedicated hardware structures tuned to the task at hand and executes operations with a small energy overhead. Features such as hardware looping reduce the instruction fetch overhead for the TI DSPs, but they still fall short of what Pleiades accomplishes. The XC4003A executes 5 taps in a single cycle. The XC4003A is not very energy efficient, but it has the ability to use optimized shift-and-add multipliers which the other architectures cannot.

Table 5 compares the efficiency of our reference architectures for the FFT benchmark. Compared to FIR and IIR, FFT is a more complex kernel. Pleiades outperforms the other processors by a large margin owing to its ability to exploit higher levels of parallelism by creating a dedicated parallel structure from its computational resources and flexible interconnect.

Processor	L_{min}	T_{ox}	V_{th}	$V_{dd,native}$	V_{dd}	Cap. Coeff.	Delay Coeff.
Pleiades	0.6 μm	9 nm	0.7 V	3.3 V	1.5 V	1.0	1.0
StrongARM [11]	0.35 μm	6 nm	0.35 V	1.5 V	1.5 V	1.96	4.7
TMS320C2xx	0.72 μm	14 nm †	0.7 V †	5.0 V	3.0 V	1.1	1.37
TMS320LC54x [12]	0.6 μm	9 nm †	0.7 V	3.3 V	3.0 V	1.0	1.97
XC4003A	0.6 μm	14 nm †	0.7 V †	5.0 V	5.0 V	1.56	2.7

Table 2. Fabrication Process Details (estimated values are marked with †)

Processor	StrongARM	TMS320C2xx	TMS320LC54x	XC4003A	Pleiades
Frequency(MHz)	169	20	40	6	14
Number of Multipliers	0.5	1	1	5	1
Throughput(cycles/tap)	17	1	1	0.2	1
Energy/tap(J)	21.1n	4.8n	2.4n	15.4n	205p
Capacitance/tap(pF)	9380	530	270	620	91
Capacitance/tap(pF) @0.6μm	16600	580	270	960	91
Energy/tap(J) @1.5V,0.6μm	37.4n	1.3n	600p	2.2n	205p
Frequency(MHz) @1.5V,0.6μm,V_{th}=0.7V	36	15	20	2.2	14
Energy·Delay/tap(J·s$\times 10^{-17}$) @1.5V,0.6μm,V_{th}=0.7V	1760	8.9	2.9	20	1.5

Table 3. FIR Benchmark Results

Processor	StrongARM	TMS320C2xx	TMS320LC54x	XC4003A	Pleiades
Frequency(MHz)	169	20	40	2.1	14
Number of Multipliers	0.5	1	1	9	2
Throughput(cycles/IIR)	114	20	13	1	8
Energy/IIR(nJ)	155	69	38	733	1.9
Capacitance/IIR(nF)	68.7	7.7	4.2	29.3	0.85
Capacitance/IIR(nF) @0.6μm	123	8.5	4.2	46	0.85
Energy/IIR(nJ) @1.5V,0.6μm	277	19.1	9.5	103	1.9
Frequency(MHz) @1.5V,0.6μm,V_{th}=0.7V	36	15	20	0.77	14
Energy·Delay/IIR(J·s$\times 10^{-17}$) @1.5V,0.6μm,V_{th}=0.7V	875	26.2	6.0	134	1.1

Table 4. IIR Benchmark Results

Processor	StrongARM	TMS320C2xx	TMS320LC54x	Pleiades
Frequency(MHz)	169	20	40	14
Number of Multipliers	0.5	1	1	4
Throughput(cycles/stage)	766	152	76	8
Energy/stage(nJ)	1040	478	197	13.3
Capacitance/stage(nF)	462	53.1	21.9	5.91
Capacitance/stage(nF) @0.6μm	831	58.4	21.9	5.91
Energy/stage(nJ) @1.5V,0.6μm	1870	131	49.3	13.3
Frequency(MHz) @1.5V,0.6μm,V_{th}=0.7V	36	15	20	14
Energy·Delay/stage(J·s$\times 10^{-17}$) @1.5V,0.6μm,V_{th}=0.7V	3970	137	18.5	0.759

Table 5. FFT Benchmark Results

4 Conclusions

We have compared the Pleiades architecture to other programmable processors that have various levels of programming granularity. Pleiades achieves better performance and energy efficiency because it can create dedicated hardware structures from optimized processing units that can execute a computational kernel with far less energy overhead and higher levels of parallelism than alternative approaches. At the same time, the Pleiades is flexible and can be programmed through its reconfigurable interconnect and parameterizable hardware blocks to execute a variety of tasks.

Acknowledgments

We would like to thank Varghese George and Hui Zhang for their assistance in designing the P1 test chip. We would also like to thank Erik Kusse for his assistance with the power estimations for the Xilinx XC4003A chip. This project was sponsored by DARPA grant DABT63-96-C-0026.

References

1. A. Abnous and J. Rabaey, "Ultra-Low-Power Domain-Specific Multimedia Processors," *Proceedings of the IEEE VLSI Signal Processing Workshop*, San Francisco, October 1996.
2. Digital Semiconductor, *Digital Semiconductor SA-110 Microprocessor Technical Reference Manual*, Digital Equipment Corporation, 1996.
3. *TMS320C5x General-Purpose Applications User's Guide*, Literature Number SPRU164, Texas Instruments, 1997.
4. T. Anderson, *The TMS320C2xx Sum-of-Products Methodology*, Technical Application Report SPRA068, Texas Instruments, 1996.
5. M. Tsai, *IIR Filter Design on the TMS320C54x DSP*, Technical Application Report SPRA079, Texas Instruments, 1996.
6. `ftp://ftp.ti.com/pub/tms320bbs/c5xxfiles/54xffts.exe`, 'C54x Software Support Files, Texas Instruments.
7. C. Turner, *Calculation of TMS320LC54x Power Dissipation*, Technical Application Report SPRA164, Texas Instruments, 1997.
8. C. Turner, *Calculation of TMS320C2xx Power Dissipation*, Technical Application Report SPRA088, Texas Instruments, 1996.
9. E. Kusse, Personal communication, 1997.
10. J. Rabaey *et al.*, "Fast Prototyping of Data Path Intensive Architectures," *IEEE Design & Test Magazine*, Vol. 8, No. 2, pp. 40-51, 1991.
11. J. Montanaro *et al.*, "A 160-MHz, 32-b, 0.5-W CMOS RISC Microprocessor," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 11, pp. 1703-1714, Nov. 1996.
12. A. Fischman and P. Rowland, *Designing Low-Power Applications with the TMS320LC54x*, Technical Application Report SPRA281, Texas Instruments, 1997.