Towards Understanding Architectural Tradeoffs in MEMS Closed-Loop Feedback Control

Greg Hoover
University of California, Santa Barbara
Engineering I
Santa Barbara, California 93106
ghoover@ece.ucsb.edu

Forrest Brewer
University of California, Santa Barbara
Engineering I
Santa Barbara, California 93106
forrest@ece.ucsb.edu

Timothy Sherwood
University of California, Santa Barbara
Engineering I
Santa Barbara, California 93106
sherwood@cs.ucsb.edu

ABSTRACT

Micro-Electro-Mechanical Systems (MEMS) combine lithographically formed mechanical structures with electrical elements to create physical systems that operate on the scale of microns. However, the physical scale of MEMS devices can make controlling them computationally challenging because the time constants involved are often several orders of magnitude faster than macro-scale devices and because they often require very low power operation. In this paper we begin an examination of the suitability of two different digital signal processors to the high-speed closed loop control problems faced by this new and growing domain. Working with domain experts in the area we characterize the classic tight feedback control loops required by these next generation MEMS devices, we explore the sources of overhead when using existing programmable systems, and we compare these approaches to an application-specific approach of our own design. In the end we demonstrate that this nature of this problem, both in terms of the required performance and the nature of the working datasize, results in a significant gap that could perhaps be filled by more programmable designs carefully crafted to this domain.

Categories and Subject Descriptors

C.1 [**Computer System Organization**]: Processor Architectures; C.3 [**Computer System Organization**]: Special-Purpose and Application-based Systems; C.4 [**Computer System Organization**]: Performance of Systems

General Terms

Design, Performance

Keywords

MEMS control, embedded architecture

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CASES'07, September 30–October 3, 2007, Salzburg, Austria.
Copyright 2007 ACM 978-1-59593-826-8/07/0009 ...$5.00.

1. INTRODUCTION

For as long as we have been able to build systems that interact with the physical world, we have had a need to control these systems in a stable and precise way. Of course complex dynamical systems usually require some sort of feedback control, and the mathematical tools required to reason about such systems is the classic preview of control theory. Since the majority of interfaces to be controlled are tied to actual *physical* objects (levers, actuators, motors, etc.), increases in processor performance over the past 30 years have resulted in the ability to carefully control increasingly complex systems. The time constants involved in moving an aileron or dampening a spring simply don't change very much over time. Thus, for many years we have been operating in a domain where real-time control was limited primarily by the ability to effectively *reason* about control, rather than the ability to effectively *implement* the controller. However, we are finding that in the domain of MEMS devices this no longer holds true.

Combining lithographically formed mechanical structures with electrical elements to create physical systems that operate on the scale of microns, Micro-Electro-Mechanical Systems (MEMS) are pushing the boundaries of physics, mechanics, and control theory. The physical scale of MEMS devices can make controlling them computationally challenging because the time constants involved are often several orders of magnitude faster than macro-scale devices and because they often require very low power operation. In fact, the control requirements of these systems have the potential to overwhelm the capabilities of even fairly high performance digital systems, a problem further exacerbated by the tight power requirements placed on MEMS used in sensor applications. The reality is that many system developers have resorted to fully analog control to meet timing [24]. Analog feedback control, that is, implementing the controller as an active analog feedback circuit, used to be the primary method of implementing controllers before digital systems of sufficient performance became widely and cheaply available. Unfortunately this presents numerous challenges since analog systems are more difficult to design (requiring careful control of all active components), implement (especially in modern low-cost processes optimized for digital systems), and maintain (there is not possibility of "patching" your system). Furthermore, the physical complexity of modern mechanical devices may require modal control that cannot be implemented by a single analog controller. These are just a

few issues that drove designers towards software-based control in the first place, but without adequate performance, systems cannot be expected to meet timing constraints for software-based closed-loop control.

In this work we explore the capabilities of commercial digital signal processors (DSP) in the context of high-throughput closed-loop feedback control. Our aim is twofold:

- Examine the design space for closed-loop control solutions through evaluation of COTS DSPs at the low power and high-performance ends of the spectrum.
- Evaluate the potential for novel architectures to fill the gap left by COTS devices, providing solid foundations for performing software-based control that simultaneously increase control performance (bandwidth) and reduce power consumption.

In working with MEMS developers on a variety of applications, we have seen that the state-space analysis technique [29, 17] is being applied across a wide variety of devices and so, for this initial work we concentrate specifically of characterizing its performance. State-space analysis is commonly used because of the relative ease of determining stable control equations because it forms the basis for many more complex control techniques, and because its behavior follows somewhat intuitively from the simple equations:

$$S' = A * S + B * x$$
$$y = C * S + D * x$$

where the next state (S') and output (y) vectors are computed by multiplication of the current state (S) and input (x) vectors with coefficient matrices A, B, C, and D. To characterize the closed-loop performance of existing DSP solutions, we model parameterized state-space algorithms and apply them to power and throughput models of the respective DSP devices. Of course an analysis of the DSP code by itself would leave little intuition of how well we *could* do. In order to address this questions we need to develop a custom architecture specifically designed to attack this problem, and in fact we have done just that. We will demonstrate that the nature of this problem, both in terms of the required performance and in the nature of the working datasize, results in a significant gap between what can be filled by currently available programmable designs carefully crafted to this domain and custom designs. This demonstrates that a custom, but strictly digital, solution is completely valid and useful in this domain, and hints to the idea that a processor specifically targeting this domain might be able to provide both the required performance and programmability.

The remainder of this paper is organized as follows: Section 2 discusses the motivation for this work. Section 3 presents our evaluation of DSP devices at performing closed-loop control. Section 4 our implementation of a partitioned vector machine optimized for feedback control and compares it to industry-leading DSPs. Section 5 presents an overview of prior work in the area. Finally, we summarize our findings and future work in Section 6.

2. MOTIVATION

While the feasibility of digital (as opposed to analog) control in MEMS devices might seem like an obvious step to many computer engineers, in fact this is an idea that faced very significant resistance from several different MEMS groups (both academic and especially industry) who simply did not believe the required performance could be delivered. In this section we would like to provide a little background, both in terms of the need for research into MEMS controllers and in terms of the related research projects ongoing in other areas.

Control systems fill a large space ranging from multi-redundant avionics systems to arm positioning motors in hard drives. In recent years, it has been possible to add more than sufficient bandwidth to control systems by simply selecting higher performance digital signal processors (DSP), or adding additional computation resources. Today, the reality is that control systems are bound by power and size constraints that dictate use of lower performance devices. For instance, distributed sensor networks can't be deployed with system implementations that require the space of a shoe box and several amps of continuous power. This is particularly true in the field of Micro-Electro-Mechanical Systems (MEMS), where sensors and actuators operate on the scale of microns. These devices are paving the way for low cost, tightly integrated solutions that combine high performance processing and ultra-accurate sensing. In addition to consuming less power and space than their macro-scale counterparts, these machines can be highly integrated with electrical components [11] and fabricated in bulk to reduce cost.

The hallmark of MEMS is the ability to produce extremely small structures capable of movement. As with macro-scale devices, these parts require control systems to influence their motion or orchestrate the operation of the device as a whole. Just as shrinking electronic features sizes has continued to provide increased performance and higher device density, process advancement has similar scaling benefits with MEMS, where smaller physical size allows greater device sensitivity and more devices to be fabricated on a single die. With smaller physical size, comes lower tolerance for error, and therefore the necessity for higher control precision. Additionally, these miniature devices are capable of extremely fast movement requiring very high control bandwidth to ensure their stability [10, 9]. With time constants several orders of magnitude faster than their non-MEMS counterparts, many MEMS devices require control bandwidth and accuracy exceeding the ability of conventional digital solutions.

While, at first, this may seem like a problem that is squarely in the domain of a programmable DSP, it turns out that many of today's micro-machined systems are limited by the availability of robust control (both theory and implementation) rather than fabrication techniques [12], and those MEMS that absolutely require on-chip control are currently constructed with analog controllers. While specially crafted analog control systems can meet such requirements, in our experience the difficulty of design, lack of flexibility to adjust to unknown device parameters, and complications that arise in system integration, are all important factors that severely hamper their effective use. As the field advances, increasingly complex control algorithms are required to ensure device stability [13, 35] and coordinate on-chip activities [14]. These requirements have begun to over-step the capabilities of analog controllers and, just as in the macro-scale world, this is an excellent fit for the capabilities of software-based control. Furthermore, the ineffectually of available control

solutions promotes design space decisions that may result in suboptimal or even unusable MEMS designs.

Exemplifying the growing gap between conventional control solutions and the needs of high control bandwidth are MEMS devices such as tunneling accelerometers [27], gyroscopes [32], and micromirrors used in optical network switches [23, 7]. Constructed from little more than a proof mass suspended on springs, MEMS accelerometers provide solutions to a wide range of applications from automotive airbags to data protection in hard disks and even offer the potential to correct injured or defective balance in people [30]. However, with physical elements separated by less than 1nm, adequate control requires significant bandwidth [34]. MEMS gyroscopes on the other hand, have lower bandwidth requirements but larger control complexity [6, 8, 16]. Novel designs is this space offer over 15 times the bandwidth at a fraction the size of their macro-scale counterparts, but require 2-dimensional control often requiring ASIC and DSP/FPGA hybrid systems that clearly defeat the advantages of micro-scale devices [32, 6, 16]. One of the earliest commercialized MEMS devices, the Digital Micromirror Device (DMD) is used in millions of Digital Light Projection (DLP) [20] television and theater projection systems to provide excellent fidelity. In optical switching applications, micromirrors are used to route data between fiber-optic links without the overhead of optical-electrical conversion. In these applications, mirror positioning accuracy is on the order of $100\mu rad$ [13]. Control frequencies of 10-20kHz are sufficient for micromirror control, however switches of 256x256 ports require sophisticated multi-DSP control systems with more than 25 DSPs [13, 5]. In these applications, system efficacy as a whole is determined by the speed and accuracy of mirror positioning. These applications typify a new characteristic of this domain of control systems whereby accuracy and control bandwidth trade off. Thus it is of growing significance that digital control solutions offer increased performance for closed-loop feedback tasks.

3. DSP EVALUATION

As the workhorse of digital control, the DSP embodies architectures targeted at fast computation of multiply- accumulate (MAC) and other common signal processing operations. While one might assume that DSPs are a natural fit for high-rapidity control, the generality in DSP architectures often adds unacceptable latency to computations. To date, conventional DSPs have provided more than adequate bandwidth, with consistent performance gains from feature scaling and added architectural parallelism. However, for this new class of control applications, the latencies inherent to many DSP architectures will limit the achievable *control bandwidth* [1]. Further, MEMS systems, by nature, are small in size and complexity, resulting in simple control computations that must be performed at very high rates [34]. For small control systems, the benefits of added parallelism in high performance DSPs fails to mitigate the complexity and performance overhead of the architecture.

Here we evaluate the performance and power tradeoffs of conventional DSP devices for performing closed-loop feedback control. As a basis for comparison, we have selected two DSP families from Texas Instruments as representative of the low power and high performance device markets. To provide a foundation for the comparison, we generalize models of the state-space analysis technique and apply them to power and performance models of the architectures.

3.1 Methodology

As the basis for our characterizations, we use the computation model extracted from the state-space control technique. There are a number of variables that influence the dimensions of each matrix in the control computation: the number of state variables, the density of the matrices, and the numbers of I/O. In order to provide a consistent and intelligible comparison, all modeling was performed with the assumption that all matrices are dense, and therefore require complete matrix multiplication (as is often the case), and that the control system has 2 inputs and 2 outputs. This allows comparison where control complexity is directly proportional to the dimension of the state vector (or number of state variables) – the dimension of the state vector will subsequently be simply referred to as 'dimension'. Hand-optimized cycle execution models are used as the foundation for our characterizations throughout this work. Power and performance characteristics for comparison devices are supplied by Texas Instruments [28, 19, 1, 2]. Since nondeterministic architectural latencies arising from prioritization of interrupts, bus contentions, and instruction mix can lead to significant execution inconsistencies, we assume that both DSP devices exhibit only static architectural latencies, ignoring the potential for dynamic uncertainties.

3.2 Comparison

The Texas Instruments TMS320C55x DSP family is the lowest power DSP available [1, 28]. Unlike many of its competitors, it provides low power execution in addition to low power standby modes. The TMS320C64x DSP family is arguably the highest performance DSP available, with an 8-way VLIW datapath and clock speeds up to 1GHz, it can execute as many as 8000 MIPS [2]. By comparison, the C55x family has a maximum clock frequency of 300MHz. In Figure 1 we present the maximum achievable control bandwidth as determined using these clock frequencies and the respective execution models over varying complexity control systems. Figure 1 also compares our optimized vector architecture to be presented in Section 4. It is immediately evident that the added parallelism available in the C64x family allows greater control bandwidth over the C55x with nearly an order of magnitude difference across all scales. The ability to perform 4 16-bit MAC operations per cycle greatly increases the throughput of the C64x. As complexity scales, however, the computational advantage is outweighed by the cost of moving data between local register files and main memory as seen by the exponential fall off in control bandwidth. It is important to note that computation complexity grows quadratically with control complexity due to the n^2 multiplications in a matrix multiply.

In Figure 2, we show how power scales for varying complexity control systems at fixed a 100kHz bandwidth. To model a constant control bandwidth, we scale the clock frequency of the device as necessary across the range of control complexities. This allows power calculations against models provided by the manufacturer. As expected, the C55x architecture is lower in power consumption that its high-

[1]control bandwidth refers to the throughput of the controller, however as each state space time step is dependent on the last, the latency is a critical aspect

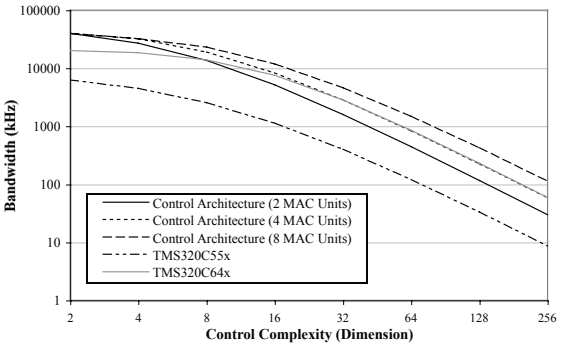

Figure 1: A plot of the control bandwidth of the TMS320C55x and TMS320C64x DSP families as control algorithm complexity is scaled, as compared to the capability of an optimized vector architectures.

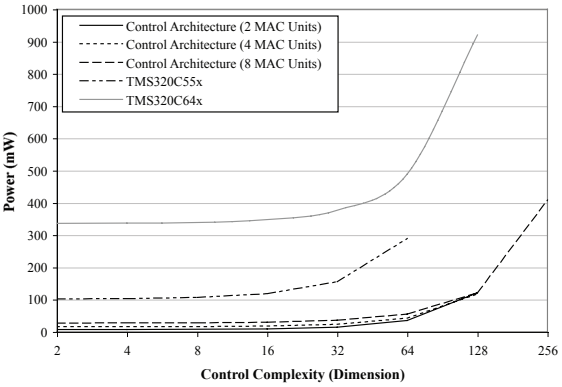

Figure 2: A plot of the power of the TMS320C55x and TMS320C64x DSP families as control algorithm complexity is scaled for 100kHz control bandwidth, as compared to the capability of an optimized vector architectures.

performance competitor, but fails to be a viable solution as complexity scales. It is interesting to note how compensating for performance using clock frequency scaling drives power consumption at an accelerated rate after the knee at dimension 32 and 64. At the high end, the 64x scales in terms of bandwidth, but with power consumption of nearly 1 amp, is impractical for use in small embedded systems.

While Figures 1 and 2 clearly show how bandwidth and power degrade at accelerated rates beyond a certain control complexity, the reasons are not immediately evident. In the case of the low power C55x family, limited architecture parallelism tends to serialize execution, resulting in numbers that linearize rather quickly. Hence, bandwidth and power numbers follow proportionally to control complexity. By contrast, the high-performance C64x family can perform many simultaneous operations through vector operations, dual pipelines, and multiple functional units. But added parallelism only helps to a point. Consider a matrix multiply algorithm for the C64x that loads rows of the corresponding matrices using vector load operations, then multiplies them using packed dot product instructions, and finally accumulates end values. In this algorithm, the C64x spends as much time loading data as it does performing computation. This, coupled with unavoidable data dependencies in the final accumulation stage, cause the performance of the C64x to drop off rapidly when the size of the matrices exceed that of the internal register files. This is clearly seen in the accelerated drop off in Figure 1.

4. PARTITIONED VECTOR ARCHITECTURE

Our goal is to create a reference design for high bandwidth closed-loop feedback control. To this end, we have designed an optimized functional pipeline and novel interface architecture. Here we describe the vector architecture that we have tuned specifically for closed-loop feedback control. The architecture aims to alleviate many of the bottlenecks of general-purpose processors through observations about state space control computation. To justify our decisions, we model the performance, area, and power of our architecture and compare it that of the DSPs evaluated above. We additionally characterize a range of implementations to showcase the scalability of the architecture. A diagram of our closed-loop feedback architecture supporting two MAC units is shown Figure 3.

4.1 Architecture

Observation of computation patterns in the state analysis control technique shows that a very limited number of operations are ever performed. The majority of computation comes in the form of several matrix multiplies, with some limited support required for data movement. Our architecture takes advantage of this by providing a functional data path supporting only addition, multiplication, and multiply-accumulate operations. We implement this functionality in a 6 stage, vector pipeline, where we determined that 4 execution stages was optimal via design exploration. The benefits of alleviating control flow, comparison, and general arithmetic instructions can be directly seen in the resulting performance, area, scalability, and simplicity of the design.

An optimized datapath is of no benefit without efficient access to supporting data. While access through a host processor interface is possible, such a solution adds to the complexity and uncertainty of the implementation. Priority interrupts and bus contention also add to the nondeterminism encountered in typical coprocessor or single processor solutions. Observation of data access patterns in the control algorithm allows memory sizes and organization to be optimized for area, power, and performance. Rather than support a single general-purpose register file, our architecture distinguishes between the functions of data commonly encountered in closed-loop calculations. By separating coefficient values from state and input data into multiple register files the aggregate bandwidth of any single register file can be reduced, further reducing both area and power requirements. This organization scales nicely, supporting a range of applications through simple resizing of respective register files.

Configuration and control occur via a host processor interface consisting of memory-mapped control registers. Execution is controlled using a series of timers that are accessible

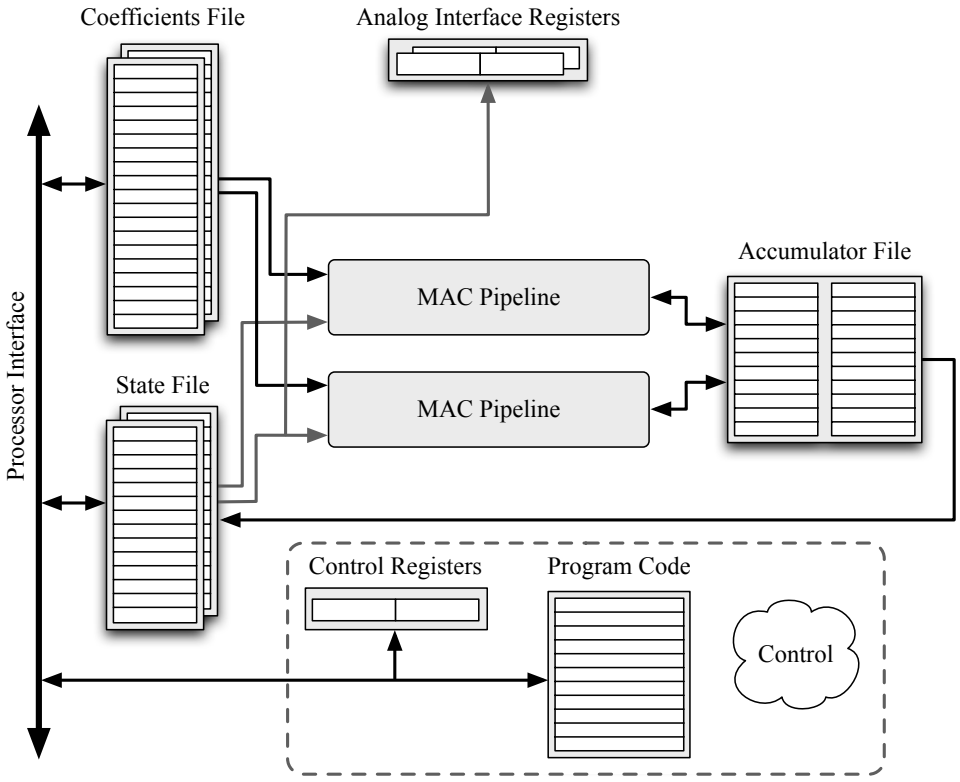

Figure 3: Scalable closed-loop feedback control architecture.

both internally and via the host processor interface. This allows reliable, static control timing while simultaneously reducing control overhead. Support for process introspection is provided via memory-mapped access to state value memory, allowing a variety of functionality including modal changes. Configurations include the control algorithm and coefficient values which remain static or change infrequently (due to modal changes). This allows access to coefficient memory to be optimized, requiring read-only access to the control architecture and write access via the host processor interface, substantially reducing the implementation overhead that would result from multi-porting memory. Further, non-linear systems benefit from the ability to uniformly select between coefficient memory banks, providing seamless migration from one set of coefficient values to another. In order to maintain the integrity of current state values, we opted to bank the state register file such that new values can be written without incurring latency from moving data between temporary storage.

Another aspect of our optimized memory organization can be seen in the analog interface, where analog-to-digital and digital-to-analog component interfaces are directly connected. Direct connectivity removes the host processor and any shared interconnect from the execution loop, allowing an autonomous solution and static timing boundaries. Interface registers are implemented as shadowed pairs to allow simultaneous input and output without the need for temporary storage. Direct interface coupling improves achievable control bandwidth and obviates the need for dynamic real-time scheduling.

Parallelism inherent in matrix multiplication suggests that significant performance improvements can be had by providing increased architectural parallelism. This can be seen in Figure 1, where implementations with greater parallelism achieve higher bandwidths for medium to large scale control systems. The cost of this scaling is presented in Figures 5 and 6, which additionally highlight the cost of supporting 40-bit precision in calculations, rather than 32.

4.2 Comparison

Characterization of our control architecture is performed through synthesis of the functional components of the datapath and memory modeling using the model presented by Agrawal [4]. The datapath implementation draws extensively from the Synopsys DesignWare library and is synthesized in 90nm TSMC using Synopsys Design Compiler. Memories smaller than 64 bytes are synthesized to flip-flop implementations in the same technology. The maximum clock frequency of our architecture is 1.2GHz in 90nm and is limited by the logic delay of the 6-stage datapath. Earlier we discussed the performance and power tradeoffs in Figures 1 and 2 as they pertain to the C55x and C64x architectures. Here we show how our vector architecture stacks up against the DSP solutions. To provide a more objective comparison to the TMS320C64x, we show characterizations of our architecture at 1GHz, rather than the maximum clock frequency.

In Figure 1, we can see that implementations of the optimized vector architecture offer consistently better than 2x the performance of the DSPs across all control scales, with performance drop off occurring at a slower rate. At the

low end, the parallelism afforded by vector instructions does well to match the performance of the high-end C64x. At the high-end, data access and storage become the predominant factors in performance, and optimizations here successfully reduce these latencies. This can be seen by similarities in the performance curves as parallelism is added to the control architecture. Our architecture shows potential for a wide range of applications, supporting control bandwidths as high as 40MHz in small systems and well over 200kHz in large control systems. The area impact of this scaling, shown in Figure 5, is minimal, requiring less than 2x overhead for a 14x performance improvement. Furthermore, the area cost of all linear control implementations is under $1mm^2$. With regard to power, it is interesting to note that even the control architecture with 16 MAC units requires less than the C64x DSP, as shown in Figures 2 and 6.

While the efficacy of our approach has been demonstrated through bandwidth and power numbers presented above, the advantage is more clearly shown when we relate bandwidth and power directly. To this end, Figure 7 shows how power scales with bandwidth for a fixed control complexity. The benefit is clearly shown over all bandwidths, with as much as 14x power reduction over the low power C55x. And while neither of the TI DSPs are capable of 100kHz bandwidth for a control system of dimension 256, an implementation of our architecture with 8 MAC units achieves this while requiring less power than the C64x and less than 50% more power than the C55x. While one might point out that our architecture is incomplete and requires a host processor that inevitably increases both the implementation area and power, direct connection of analog interfaces in our architecture facilitates autonomous execution allowing the host processor to idle in lower power modes such as that of the C55x requiring only 0.12mW [3].

5. RELATED WORK

There is a substantial body of work that aims to enhance system performance and power through synthesis of custom processors targeting a variety of applications. Digital signal processing is an area that is commonly targeted due to the inherent parallelism of many of the algorithms. Through extraction of execution parallelism, custom architectures can be automatically generated and synthesized [31, 22, 26, 18].

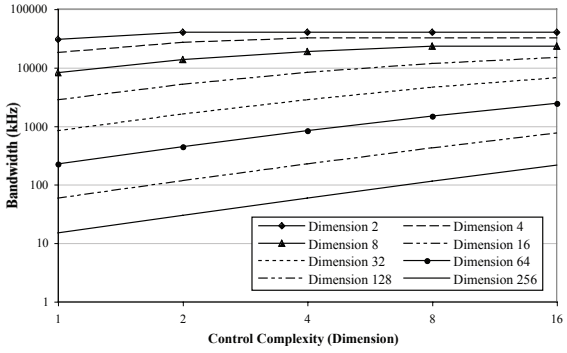

Figure 4: A plot of the bandwidth of our architecture at 1GHz as we scale the total number of MAC units in the design.

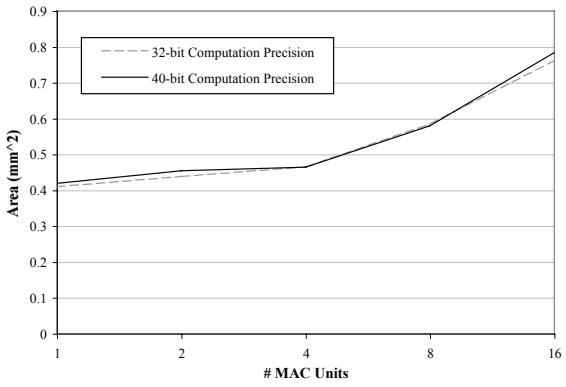

Figure 5: A plot of the area of our architecture as we scale the total number of MAC units in the design.

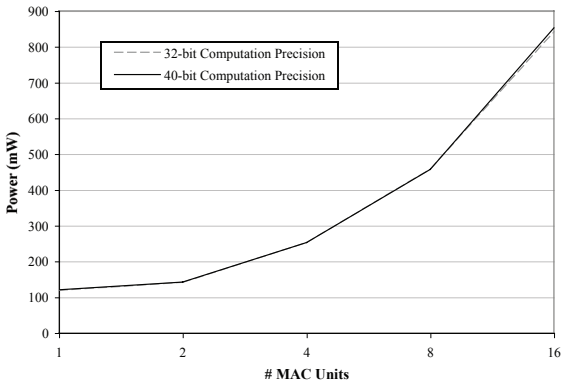

Figure 6: A plot of the power of our architecture as we scale the total number of MAC units in the design.

These accelerators are subsequently added to the processor datapath and accessed via instruction set extensions [33]. This, however, has the effect of increasing processor complexity and typically requires extensibility in the base processor design [18, 26].

At the system level, application engines can be integrated through well defined interfaces without adding to the complexity of the processor pipeline. The increasing prevalence of system on chip (SoC) solutions has made this a common avenue for many systems [15, 25]. While the flexibility of this approach serves to reduce design time and increases component reuse, it is only feasible in systems with suitable latency tolerances – offloading computation to remote components adds communication latency.

Though customized DSP cores have the potential to increase performance and even lower power [22, 21], they are ill-suited for small control systems where computation latency is primarily affected by architecture overhead. Furthermore, nondeterminism in the execution of many modern processors makes static timing boundaries difficult to determine and limits reliable bandwidth. To this end, we have provided an optimized datapath with direct communication to analog interfaces. This approach allows us to

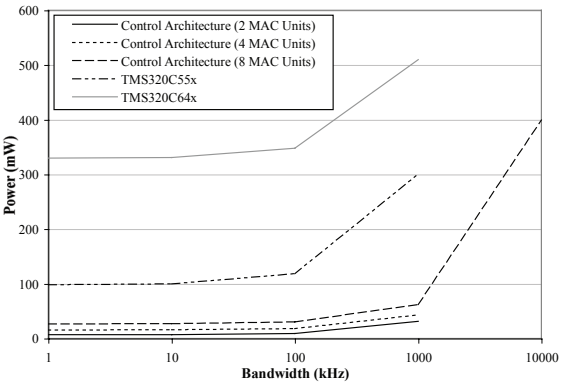

Figure 7: A plot of the power of the TMS320C55x and TMS320C64x DSP families as control bandwidth is scaled for a control system of dimension 16, as compared to the capability of an optimized vector architectures.

achieve the highest possible performance from our datapath architecture and creates, what we believe to be, the first autonomous control engine.

6. CONCLUSIONS

Integrated solutions that couple mechanical and electronics systems in micrometer spaces will revolutionize almost every application category by providing microscopic, low-cost solutions to currently impossible problems. The breadth of MEMS applications knows potentially limitless bounds, but realizing implementations will require novel solutions capable of supporting potentially millions of microscopic mechanical devices. While analog control systems are capable of addressing this requirement, the design difficulty, and lack of flexibility of, analog solutions makes them infeasible for exploratory research and commercialized products in this cutting edge field.

We have shown that there is a space of opportunity for novel architectures to address the growing gap between the capabilities of conventional control solutions and the demands of future devices. By rethinking the design choices in digital architecture, it is possible to create control systems with adequate bandwidth, negligible area overhead, and reasonable power margins. As justification for this, we have shown that vector architectures can be optimized to provide higher bandwidth and lower power than state-of-the-art industry DSP devices. We believe that by reevaluating the architectural tradeoffs in digital control it is possible to open the door for creation and commercialization of many new MEMS devices.

7. REFERENCES

[1] Tms320c55x dsp cpu reference guide. Technical report, Texas Istruments, Feb 2004.

[2] Tms320c64x/c64x+ dsp cpu and instruction set reference guide. Technical report, Texas Istruments, Aug 2006.

[3] Dsp resource guide, 2007.

[4] B. Agrawal and T. Sherwood. Guiding architectural sram models. In *Computer Design, International Conference on, Proceedings of*, Oct 2006.

[5] R. Allan. 3d mems-based optical switch handles 80-by-80 fibers. *Electronic Design*, Oct 2003.

[6] H. R. D. S. P. P. M. C. M. B. N. H. G. Andersson. A novel architecture for digital control of mems gyros. In *Sensors, 2004. Proceedings of IEEE*, volume 3, pages 1403–1406, Oct 2004.

[7] D. J. Bishop, C. R. Giles, and G. P. Austin. The lucent lambdarouter: Mems technology of the future here today. In *IEEE Communication Magazine*, volume 40, pages 75–79, Mar. 2002.

[8] Y.-C. C. R. M. T. T. B. Blaes. A control and signal processing integrated circuit for the jpl-boeing micromachined gyroscopes. *Control Systems Technology, IEEE Transactions on*, 13:286–300, March 2005.

[9] B. Borovic, A. Q. Liu, D. Popa, H. Cai, and F. L. Lewis. Open-loop versus closed-loop control of mems devices: choices and issues. *Journal of Micromechanics and Microengineering*, 15(10):1917–1924, 2005.

[10] J. Bryzek and E. Abbott. Control issues for mems. In *Proc. of the 42nd IEEE Conference on Decision and Control*, pages 3039–3047, 2003.

[11] J. Bryzek, A. Flannery, and D. Skurnik. Integrating microelectromechanical systems with integrated circuits. *IEEE Instrumentation and Measurement Magazine*, 7(2):51–59, 2004.

[12] K. F. BËĘhringer, B. R. Donald, N. C. MacDonald, G. T. A. Kovacs, and J. W. Suh. Computational methods for design and control of mems micromanipulator arrays.

[13] P. Chu, I. Brener, P. Chuan, L. Shi-Sheng, J. I. Dadap, P. Sangtae, K. Bergman, N. H. Bonadeo, T. Chau, C. Ming, R. Doran, R. Gibson, R. Harel, J. J. Johnson, C. D. Lee, D. R. Peale, T. Bo, D. T. K. Tong, T. Ming-Ju, W. Qi, W. Zhong, E. L. Goldstein, L. Y. Lin, and J. A. Walker. Design and nonlinear servo control of mems mirrors and their performance in a large port-count optical switch. *Journal of Microelectromechanical Systems*, 14(2), Apr. 2005.

[14] J. I. Dadap, P. B. Chu, I. Brenner, C. Pu, C. D. Lee, K. Bergman, N. Bonadeo, T. Chau, M. Chou, R. Doran, R. Gibson, R. Harel, J. J. Johnson, S. S. Lee, S. Park, D. R. Peale, R. Rodriguez, D. Tong, M. Tsai, C. Wu, W. Zhong, E. L. Goldstein, L. Y. Lin, and J. A. Walker. Modular mems-based optical cross-connect with large port-count. *IEEE Photonics Technology Letters*, 15(12):1773–1775, Dec. 2003.

[15] A. Dutta; S.; Jensen; R.; Rieckmann. Viper: A multiprocessor soc for advanced set-top box and digital tv systems. *Design and Test of Computers, IEEE*, 18(5):21–31, Sep-Oct 2001.

[16] M. F. D. K. K. H. B. B. C. P. K. Y. E. M. D. Foor. A hardware platform for tuning of mems devices using closed-loop frequency response. In *Aerospace, 2005 IEEE Conference*, pages 1–7, March 2005.

[17] B. Friedland. *Control system design: an introduction to state-space methods*. McGraw-Hill, 1986.

[18] R. Gonzalez. Xtensa: a configurable and extensible processor. *Micro, IEEE*, 20(2):60–70, Mar/Apr 2000.

[19] T. Hiers and M. Webster. Tms320c6414t/15t/16t power consumption summary. Technical report, Texas Istruments, Aug 2004.

[20] L. Hornbeck. Digital light processing^{TM}: A new mems-based display technology. In *IEEJ 14th Sensor Symposium, Technical Digest of*, 2006.

[21] Y.-L. T. M. H. T. J.-X. T. S.-J. Jou. Parameterized and low power dsp core for embedded systems. In *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, volume 5, pages V–265 – V–268, May 2003.

[22] C. L.-Y. M. K. R. K. Dsp data path synthesis for low-power applications. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 2, pages 1165–1168, 2001.

[23] J. Kahn, R. Katz, and K. Pister. Emerging challenges: Mobile networking for smart dust, 2000.

[24] J. Kim, C. Nuzman, B. Kumar, D. Lieuwen, J. Kraus, A. Weiss, C. Lichtenwalner, A. Papazian, R. Frahm, N. Basavanhally, D. Ramsey, V. Aksyuk, F. Pardo, M. Simon, V. Lifton, H. Chan, M. Haueis, A. Gasparyan, H. Shea, S. Arney, C. Bolle, P. Kolodner, R. Ryf, D. Neilson, and J. Gates. 1100/spl times/1100 port mems-based optical crossconnect with 4-db maximum loss. pages 1537 – 1539, 2003.

[25] J.-T. Kong. Soc in nanoera: challenges and endless possibility. In *Design, Automation and Test in Europe, 2005. Proceedings*, volume 1, Mar 2005.

[26] T.-J. Lin and C.-W. Jen. Cascade - configurable and scalable dsp environment. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, volume 4, pages IV–870 – IV–873, 2002.

[27] P. H. F. B. S. M. K. T. N. MacDonald. Single mask lateral tunneling accelerometer. In *Micro Electro Mechanical Systems, 1998. MEMS 98. Proceedings., The Eleventh Annual International Workshop on*, pages 340–344, Jan 1998.

[28] G. Martinez. Tms320vc5501/02 power consumption summary. Technical report, Texas Istruments, Jan 2004.

[29] K. Ogata. *State-space analysis of control systems*. Prentice-Hall, 1967.

[30] A. Pas. The sls, an artificial balance organ: Modeling and improving postural control. Master's thesis, Eindhoven University of Technology, March 2005.

[31] S. Ramanathan, V. Visvanathan, and S. Nandy. Synthesis of configurable architectures for dsp algorithms. *VLSI Design*, 00:350, 1999.

[32] C. A. S. E. A. M. Shkel. Concept, implementation, and control of wide bandwidth mems gyroscopes. In *American Control Conference, Proceedings of*, June 2001.

[33] C. G. E. M. P. Y. Y. K. B. tels; and S. Vassiliadis. Automatic selection of application-specific instruction-set extensions. In *Compilers, Architecture, and Synthesis for Embedded Systems, 2006. Proceedings (CASES '06). International Conference on*, Oct 2006.

[34] M. K. L. O.-R. K. Turner. Robust feedback control design of an ultra-sensitive, high bandwidth tunneling accelerometer. In *American Control Conference, 2005. Proceedings of the 2005*, volume 6, pages 4176– 4180, June 2005.

[35] G. Zhu, J. Levine, and L. Praly. Improving the performance of electrostatically actuated mems by nonlinear control. In *Proc. of the 44th IEEE Conference on Decision and Control*, Dec. 2005.