Title: Sustaining the Exponential Growth of Embedded Digital Signal Processing Capability

Authors:

Dr. Gary A. Shaw  (first author/corresponding author/presenting author)
MIT Lincoln Laboratory
244 Wood Street
Lexington, Massachusetts 02420-9185
Voice: 781-981-7994
Fax: 781-981-4608
E-mail: shaw@ll.mit.edu

Dr. Mark A. Richards
Georgia Institute of Technology
School of Electrical and Computer Engineering
Atlanta, GA 30332-0250
Voice: 404-894-2714
Fax: 404-894-0560
E-mail: mark.richards@ece.gatech.edu

Candidate Sessions (drawn from HPEC abstract submission web site):

- Algorithm Mapping to High Performance Architectures
- Future Program Office Needs for Embedded Computing Technologies
- Case Study Examples of High Performance Embedded Computing

Author’s Comments:

This proposed paper is less of a technical exposition than a retrospective and exhortation regarding the importance of sustained efforts in algorithm and software research and development. The paper compares and contrasts the contributions of hardware and algorithm improvements to the rate of improvement in high performance computing and signal processing. We conjecture that as IC shrinkage and attendant performance improvements begin to slow, the exponential rate of improvement we have become accustomed to for embedded applications will be sustainable only through a faster pace of improvement in algorithms and software.

---

1 This work was sponsored by the Department of the Air Force under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.
Sustaining the Exponential Growth of Embedded Digital Signal Processing Capability

Report Documentation Page

1. REPORT DATE
01 FEB 2005

2. REPORT TYPE
N/A

3. DATES COVERED
-

4. TITLE AND SUBTITLE
Sustaining the Exponential Growth of Embedded Digital Signal Processing Capability

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MIT Lincoln Laboratory 244 Wood Street Lexington, Massachusetts 02420-9185; Georgia Institute of Technology School of Electrical and Computer Engineering Atlanta, GA 30332-0250

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

<table>
<thead>
<tr>
<th>a. REPORT</th>
<th>b. ABSTRACT</th>
<th>c. THIS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
</tr>
</tbody>
</table>

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES
28

19. NAME OF RESPONSIBLE PERSON

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
Sustaining the Exponential Growth of Embedded Digital Signal Processing Capability

Gary A. Shaw and Mark A. Richards

The exponential improvement in IC device density and throughput observed over the past 40+ years, first articulated by Gordon Moore in 1965 and canonized as Moore’s Law, is well known and has had an immense impact on our society. Obvious, but less appreciated, is the fact that the computational capability of a digital signal processor, and more generally any high performance computing system, is the sum of its hardware capability and its software capability. By “hardware” we mean the physical implementation, which includes both individual ICs and the system architecture. “Software” is the computational procedure, which includes both the mathematical functionality and the particular algorithm by which it is implemented. For example, the discrete Fourier transform (DFT) is a particular mathematical function, while the fast Fourier transform (FFT) is a particular sequence of computations that implements the DFT efficiently.

Observed over long periods of time, software improvements, particularly fast algorithms, have contributed exponential rate improvements that equal or exceed the improvements accruing from faster hardware. The best known example in the DSP community is, of course, the FFT. Depending upon the problem order, the computational speedup afforded by the basic radix-2 FFT is equivalent to anywhere from one to 20+ years of device improvement. The FFT is not unique in terms of its impact on the acceleration of HPEC applications. In an example of long-term evolution of scientific computing algorithms, Bentley documents improvements in the solution of 3-D elliptic partial differential equations. He shows that from 1945 to 1985, operation counts for problems computed on an \(N \times N \times N\) grid were reduced through a succession of algorithmic improvements by a typical factor of \(N^4/60\). For a broadly representative problem size of \(N = 64\), the improvement is a factor of about \(3 \times 10^5\), or just under 1.4 orders of magnitude per decade. This rate of improvement is on par with that observed for IC devices. Other examples of long-term, as well as sudden, algorithmic improvements can be found in applications ranging from dial-up modems to computer chess. The improvements associated with these application areas will be illustrated in detail in the presentation.

---

1 This work was sponsored by the Department of the Air Force under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

While evidence exists for equal rate improvements from algorithms and hardware, there are nonetheless significant differences in the manner in which hardware and software improvements are manifested. Hardware throughput performance increases exponentially and predictably as a function of time. As long as Moore’s Law remains viable, we can count on the fact that a computationally complex algorithm not currently realizable in real time will eventually become realizable, and we can even predict approximately when it will be realizable! In contrast, while reduced complexity algorithms are discovered unpredictably in time, they increase execution speed exponentially and predictably as a function of problem dimension or order. Discovery of a fast algorithm thus acts in effect like the discovery of a “worm-hole” in time evolution of an application. Thus, rather than waiting for the time evolution of Moore’s Law hardware improvements, the discovery of a fast algorithm creates an instantaneous leap ahead in performance, with the “time-compression” benefit of the worm-hole increasing in proportion to problem order.

In comparison to the historically predictable payoff in performance associated with investments in IC development, the unpredictability of new algorithm discovery makes investing in signal processing capability somewhat more risky. However, the higher risk of algorithm investment is more than offset by its most fundamental payoff: the development of entirely new capabilities resulting from the application of new concepts and mathematics. Improvements in hardware speed enable application performance improvements by supporting more of the existing functionality within a given amount of time, space, power, or other resource. In contrast, breakthroughs in functionality add entirely new tools to the signal processing toolbox, and may change the entire nature of the signal processing problem and implementation complexity.

With regard to new algorithm concepts and capabilities, the scope of research opportunities is actually expanding. Many of the traditional examples of algorithm breakthroughs we have cited were new computational procedures that substantially reduced operation counts for a specific function; the solutions of PDEs, sorting, and the FFT are all examples. New mathematical techniques provide new opportunities for similar improvements. Algorithms that achieve speedups by clever matching of mathematical problem structure to computer architecture represent a very different avenue of attack. More fundamentally, emerging research focusing on knowledge-based and cognitive systems opens up to scrutiny entirely new types of both functionality and computational complexity.

The investment needed to maintain, and even expand, an active and robust embedded processing research community is a fraction of the investment needed to keep semiconductors on the Moore’s Law growth curve, primarily because algorithm research costs are dominated by salaries, which rise at a much slower rate than the cost of new fabrication facilities. Therefore, relatively modest increases in algorithm and software research are sufficient to maintain the total progress in signal processing performance. In particular, as we move into a future where the certainty of Moore’s Law hardware improvements is in question, and the investment gap between algorithm research and improved semiconductor fabrication facilities continues to widen, increasing our investment in algorithm research appears to be an attractive and proven means to sustaining exponential growth in embedded application performance.
Sustaining the Exponential Growth of Embedded DSP Capability†

High Performance Embedded Computing Workshop

28 September 2004

Dr. Gary Shaw
MIT Lincoln Laboratory

Dr. Mark Richards
Georgia Institute of Technology

† This work was sponsored by the Department of the Air Force under Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.
Elements Contributing to Embedded Processor Performance

- Signal Processor
  - Hardware
    - IC Devices
    - Computer Architecture
  - Software
    - Algorithms
    - Functionality
Outline

• Historical perspective – fulfillment of Moore’s Law
• Impediments to continued IC density growth
• Algorithms – the softer side of exponential growth
• Implications regarding sustaining exponential growth
• Summary and Conclusions
Moore’s Law: Prediction and Realization

John von Neumann: “Truth is much too complicated to allow anything but approximations.”

Gordon Moore: “If Al Gore invented the Internet, I invented the exponential.”

2x every year

1K 4K 16K 64K 256K 1M 4M 16M 64M 128M 256M 512M 1G 2G 4G

Source: Intel
Top 500 Computer Growth

Peak Performance (GFLOPS)

- N=1
- N=100
- N=500

2x every 1.1 year
Outline

• Historical perspective - fulfillment of Moore’s Law

• Impediments to continued IC density growth
  – Heat dissipation
  – Quantum effects
  – Production technology

• Algorithms – the softer side of exponential growth

• Implications regarding sustaining exponential growth

• Summary and Conclusions
# Performance Implications of Shrinking Feature Size

\[ \Delta \equiv \text{feature size} \]

<table>
<thead>
<tr>
<th>Performance Metric</th>
<th>Geometrical Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock Frequency</td>
<td>( 1/\Delta )</td>
</tr>
<tr>
<td>Transistor Power</td>
<td>( \Delta )</td>
</tr>
<tr>
<td>Transistor Density</td>
<td>( 1/\Delta^2 )</td>
</tr>
<tr>
<td>Total Device Power</td>
<td>( 1/\Delta )</td>
</tr>
<tr>
<td>Power Density</td>
<td>( 1/\Delta )</td>
</tr>
<tr>
<td>Energy/Instruction</td>
<td>( \Delta^2 )</td>
</tr>
</tbody>
</table>

![Graph showing Energy per Instruction over time](image)

-2x every 1.8 years
Moore’s Law Growth in Power Density

Year

Power Density (W/cm²)

- i386
- i486
- Pentium
- Pentium II
- Pentium MMX
- Pentium 4
- space shuttle tile
- nuclear reactor fuel cell
- hot plate

2x in 3.3 years
Moore’s Law is Dead, Long Live Moore’s Law!
Theory & Practice: Feature Size for MOSFET Devices

It's tough to make predictions, especially about the future. - Yogi Berra

<table>
<thead>
<tr>
<th>Year</th>
<th>Elect. Per Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Now</td>
<td>~100</td>
</tr>
<tr>
<td>+10</td>
<td>~10</td>
</tr>
<tr>
<td>+20</td>
<td>~1</td>
</tr>
</tbody>
</table>

Sources:
- Combined graph and original concept: Lance Glasser, former Director, DARPA/ETO
- Theory: Provided by Prof. David Ferry, Arizona State University
- Practice: The National Technology Roadmap for Semiconductors (SIA Publication, 1994)
Capitalization Cost Impediments

- Intel withdraws from DRAM market due to estimated ~ $400M capitalization cost.
- 12" fab $3-4B cost every 3 years.
- Extrapolation suggests potential risk for ROI.

Adapted from: MIT Lincoln Laboratory
Fulfillment and Impact of Moore’s Prediction

- Silicon CMOS IC fabrication technology

Naïve, low-order implementations → Scaling Rules → Process Innovations → Design Tool Innovations → Paradigm Shift?

Transistors

Year

1965

10^2

2005

10^9

2015?

Fundamental Limits

Biological Computing?

Quantum Computing?

- Examples of far-reaching impact

Altair 8800, 1975

Exponential Improvements In Computing at a Fixed Price Point

Itanium, 2005

Embedded Processors For Real-time Digital Signal Processing

Low-power Wireless Applications

Loosely-Coupled Hardware & Software Design Methodologies

Parallel Embedded Processor

System Controller

Node Controllers

Data Communication: MPL, MPART, DRI

Control Communication: CORBA, HP-CORBA

P0 P1 P2 P3

Computations: VSIPL, VSIPL++ | VSIPL++

MIT Lincoln Laboratory
Outline

- Historical perspective - fulfillment of Moore’s Law
- Impediments to continued IC density growth
  - Algorithms – the softer side of exponential growth
- Implications regarding sustaining exponential growth
- Summary and Conclusions
Elements Contributing to Embedded Processor Performance

- Signal Processor
  - Hardware
    - IC Devices
    - Computer Architecture
  - Software
    - Algorithms
    - Functionality
## Different Character of Hardware (IC) Vs. Algorithm Improvements

<table>
<thead>
<tr>
<th>Improvement Metrics</th>
<th>Hardware</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regularity</td>
<td>Predictable</td>
<td>Unpredictable</td>
</tr>
<tr>
<td>Dependent variable</td>
<td>Time</td>
<td>Order complexity</td>
</tr>
<tr>
<td>Impact on applications</td>
<td>Incremental</td>
<td>Leap-ahead</td>
</tr>
<tr>
<td>Useful lifetime</td>
<td>3 years or less</td>
<td>10 years or more</td>
</tr>
<tr>
<td>R&amp;D Cost growth</td>
<td>2x in 3 years</td>
<td>1.11x in 3 years</td>
</tr>
</tbody>
</table>
Computational Complexity Reduction Afforded by the FFT Over a Sum-of-Products DFT
Moore’s-Law Equivalent Years Required to Match FFT Computational Speedup

Years of Hardware Improvement Required for Equal Computational Speedup

Radix-2 FFT Length

FFT Length

5 10 15 20 25
4 8 16 32 64 128 256 512 1024 2048 4096 8192 16384
Exponential Improvement in Modem Rates

Downstream Data Rate (bps)

Year


2x every 1.8 years
Application Maturation Cycle

- **Application Concept**
- **Prototype**
- **Application Realization**
- **Faster Better Cheaper**

The cycle moves from application concept to prototype, then to application realization, and back, with an option to move to a different path.
Pulse-Doppler Radar Example

- Algorithmically naïve implementation

- Reduced-order implementation with digital I/Q
Pulse-Doppler Radar Algorithm Improvements

![Diagram showing the progress of Pulse Compression, Doppler Filtering, and Digital I/Q over years with the reference slope indicating Moore's Law](image-url)
Outline

- Historical perspective - fulfillment of Moore’s Law
- Impediments to continued IC density growth
- Algorithms – the softer side of exponential growth
- Implications regarding sustaining exponential growth
- Summary and Conclusions
IC Vs. Algorithm Development
(A Contrived but Useful Analogy)

• ICs

Naïve, low-order implementations → Scaling Rules → Process Innovations → Design Tool Innovations → Paradigm Shift?

Silicon lithographic fabrication technology

• Algorithms

Naïve, high-order implementations → Order Reductions → Architecture Innovations → H/W & S/W Codesign Tools → Paradigm Shift?

Linear algebraic representations and processing

MIT Lincoln Laboratory
Increased Emphasis on Codesign Methodologies

- Signal Processor
  - Hardware
  - Software
    - IC Devices
    - Computer Architecture
    - Algorithms
    - Functionality
Wafer-Fab Capitalization Cost Compared to Annual DSP Algorithm R&D Costs

- **Capital cost for state-of-the-art wafer fab facility**: Red line, increasing every 3 years.
- **Annual R&D support for entire IEEE SP Society membership (18,500 x $150K in 2001)**: Blue line, increasing every 3 years.


$B

1.11x every 3 years

2x every 3 years
Summary and Conclusions

• Fulfilling Moore’s Law
  – Enabled by diverse, innovative R&D aimed at realizing a common vision (ITRS semiconductor roadmap)
  – Continued improvements may be impeded by a combination of thermal, quantum, and capital cost limits

• Taking up the slack
  – Over same 40-year time frame as Moore’s Law, algorithm innovation has yielded exponentially improving performance as well
  – Algorithm innovation also enabled by diverse R&D, but without as clear of an industry-wide common vision
  – Algorithm R&D cost growth significantly lower than fab capital cost growth (1.1x vs. 2x every 3 years)

• Increasing the effectiveness of algorithm R&D
  – Develop better methods for quantifying the return on investment for algorithm R&D
  – Consider mechanisms for developing a broader industry vision and commitment to a long-term R&D roadmap

• Hardware/software codesign methods increasingly important