DSP on General-Purpose Processors — An Overview
or
Can General-Purpose Processors Replace DSPs?

MicroDesign Resources DInner Meeting
Santa Clara, California
January 9, 1997

Jeff Bier

Berkeley Design Technology, Inc.
Berkeley, California
E-mail: bier@bdti.com
Web: www.bdti.com
Tel: (510) 665-1600

Why is this Important?

DSP applications are proliferating dramatically:

- **Wired and wireless comms., music, speech, video, motor control, noise cancellation, navigation, ...**
- **Many products are becoming DSP-intensive**

The DSP chip market is booming:

- **$2.3B (programmable) and $3.6B (non-programmable) in 1996**
- **Growth rate > 30% per year [Source: Forward Concepts]**

Many products contain DSPs and general-purpose processors (GPPs).

- **With increasing integration, one will tend to subsume the other**

Outline of Today's Presentation

Focus: DSP capabilities of general-purpose processors

- Applications and system architectures
- Processor architectural approaches
- Evaluation criteria
- Evaluating GPPs for DSP applications
- Conclusions and Trends

Not included:

- "Media" processors
- Non-programmable devices

Short Answer for the Impatient

Can general-purpose processors replace DSPs today?

Yes:

- **Especially in personal computers**
- **Especially for certain applications; e.g., telephony, videoconferencing**

But, you may not want to, because:

- **DSPs have an unbeatable combination of integration, cost-performance, low power, and infrastructure for many applications**
- **Ensuring strict real-time behavior on GPPs can be problematic**
- **DSP software development on GPPs can be difficult**

Requires application-by-application analysis

Applications and System Architectures

What's Special About DSP Applications?

Demands:

- Lots of number crunching
- High data bandwidth; limited data locality
- Real-time constraints
- Attention to subtle numeric effects (in fixed-point implementations)
- Specialized peripherals/interfaces

Applications

For this analysis, we divide DSP applications into two categories:

- Personal-computer-based
  - E.g., modems, speech compression/recognition/synthesis, music/sound synthesis, video compression
- Embedded
  - E.g., disk drive servo control, cellular phones, pagers, motor control, navigation, modem banks, answering machines

Both classes are candidates for implementation on general-purpose processors.

PC-based applications are receiving more attention now, but embedded applications are and will be far more numerous.

System Architectures

<u>Many</u> existing or emerging products:

- **Already contain a μP or μC**
- **Already contain a μP or μC <u>plus</u> a DSP**
- **Require μP/μC and DSP functionality**

Merging all programmable functionality into a single processor can be attractive:

- **High integration can reduce size, cost, power consumption**
- **Leverages existing software, tools, know-how**
- **Few modifications to existing system hardware**

Overview of GPP Architectural Approaches to DSP

GPP Architectural Approaches to DSP

General-purpose processor vendors have taken a variety of approaches to addressing DSP performance:

- **Baseline GPPs (moderate performance, no DSP features)**
- **High-performance GPPs with few/no DSP-oriented features**
- **GPPs with major DSP-oriented features**
  - **SIMD**
  - **DSP-processor-like**
- **DSP co-processors**

I: Baseline GPP Architectures

Example: Advanced RISC Machines' ARM7TDMI

Typical moderate-performance GPPs with no DSP features perform poorly on DSP tasks.

The main reasons for this are:

- **Poor multiplication throughput**
- **Limited memory bandwidth**
- **Loop overhead**
- **Address generation overhead**

Also, on fixed-point processors:

- **Lack of hardware support for fast overflow protection, convergent rounding, etc.**

II: High-Performance GPPs with No/Few DSP Features

Example: Pentium (P54C), PowerPC 604e, IDT R4650

These processors can perform very well on DSP tasks.

The main reasons for this are:

- **High clock rates (200+ MHz; 2-5 X those of typical DSPs)**
- **Single-cycle multiplication and arithmetic operations**
- **Good memory bandwidth**
- **Loop overhead reduced via branch prediction and multi-issue**
- **Address generation, other overhead reduced via multi-issue**

However, dynamic features complicate optimization of DSP code and real-time development.

III: GPPs with Major DSP Features (1 of 2)

Approach A: Single-instruction, multiple-data operations

Example: Intel MMX Pentium (P55C)

These processors achieve outstanding DSP performance by combining the features of "conventional" high-performance GPPs (group II) with new SIMD capabilities:

- **Partition existing data path (or add a new, partitioned one)**
- **Multiple operations/cycle on small data types (e.g., 4 multiplies)**
- **Single-cycle operations on various fixed-point data types**
- **Specialized instructions**

Integration of these features into a pre-existing architecture can be awkward.

III: GPPs with Major DSP Features (2 of 2)

Approach B: Integration of DSP-processor-like features

Example: Hitachi SH-DSP

These processors achieve good DSP performance by mimicking DSP processors.

To a conventional GPP architecture, they add:

- **A DSP-oriented data path, complete with dedicated registers**
- **Address generators, hardware looping, modulo addressing, saturation, etc.**

Integration of these features into a pre-existing architecture can be awkward.

IV: DSP Co-processors

Example: ARM Piccolo

These processors should achieve good performance on DSP tasks. None widely deployed yet.

Approach is similar to IV(B), but:

- **Programming can be more complicated**
- **More parallelism may be possible**

Contrast with DSP + GPP on one chip:

- **Motorola MC68356**
- **Texas Instruments TMS320C54x + ARM7**

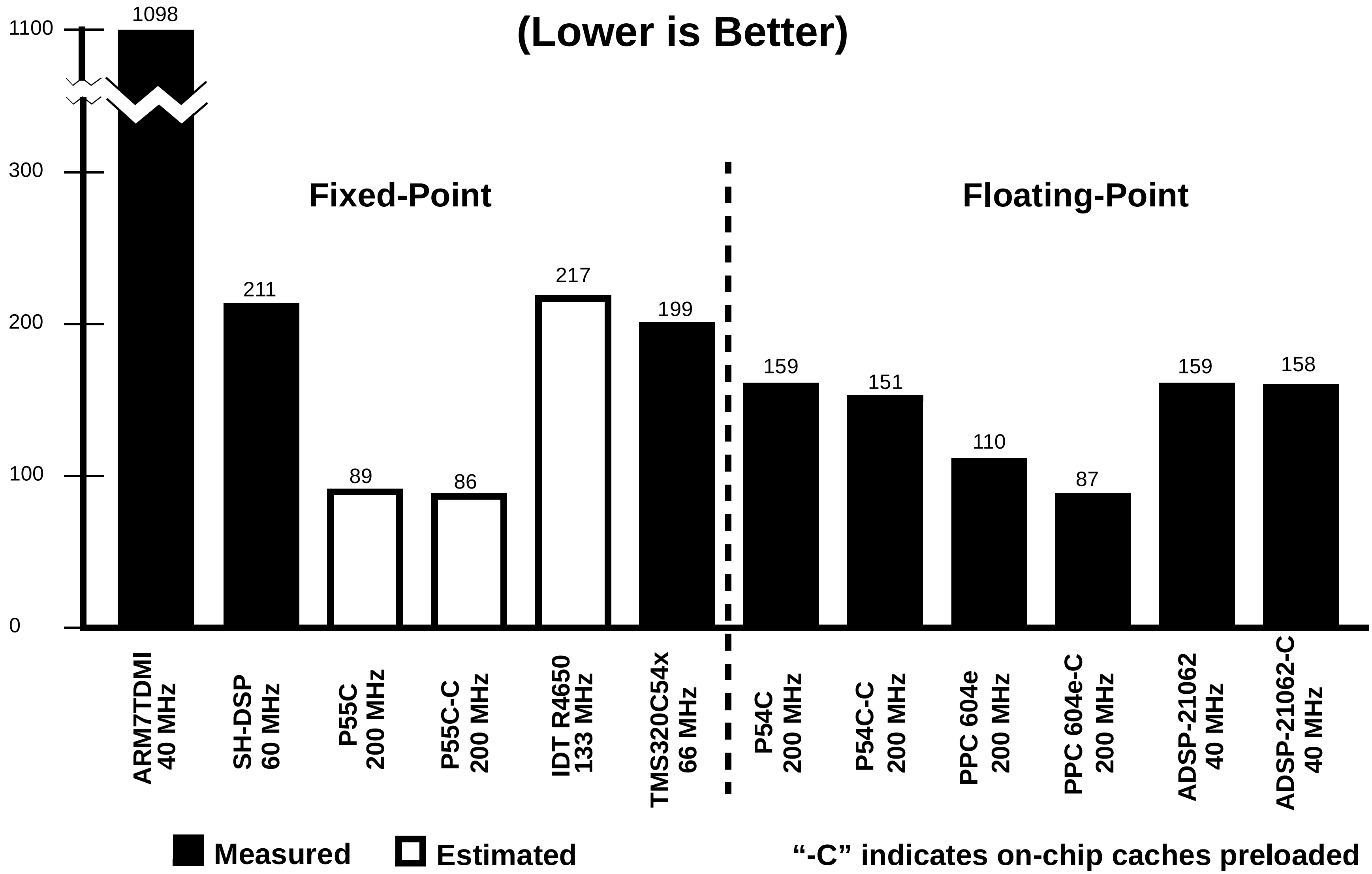
Evaluating GPPs for DSP

What’s Most Important?

- **DSP speed**
- **DSP numeric performance (can sometimes trade off vs. speed)**
- **Cost**
- **Cost/performance**
- **Power consumption**
- **Real-time suitability**
- **Product development time and cost**

Speed: BDTI FFT Benchmark Execution Time (μs)

(Lower is Better)



"-C" indicates on-chip caches preloaded

DSP Numeric Performance

DSP applications are often very sensitive to numeric effects. This is typically not a concern with floating-point processors.

On fixed-point processors, key issues include:

- **Selection of appropriate word widths**
- **Overflow protection**
- **Convergent rounding**
- **Multi-precision/block floating-point/floating-point support**

Lack of needed hardware support can be overcome with software, but the cost may be high.

Cost

Cost is surprisingly tricky to analyze.

- **Processor cost alone is often not very relevant.**

Need to compare the overall system costs resulting from processor choices.

- **E.g., may need to compare the cost of a DSP processor plus its own memory vs. the cost of upgrading to an enhanced GPP.**
- **Memory usage plays an important role.**

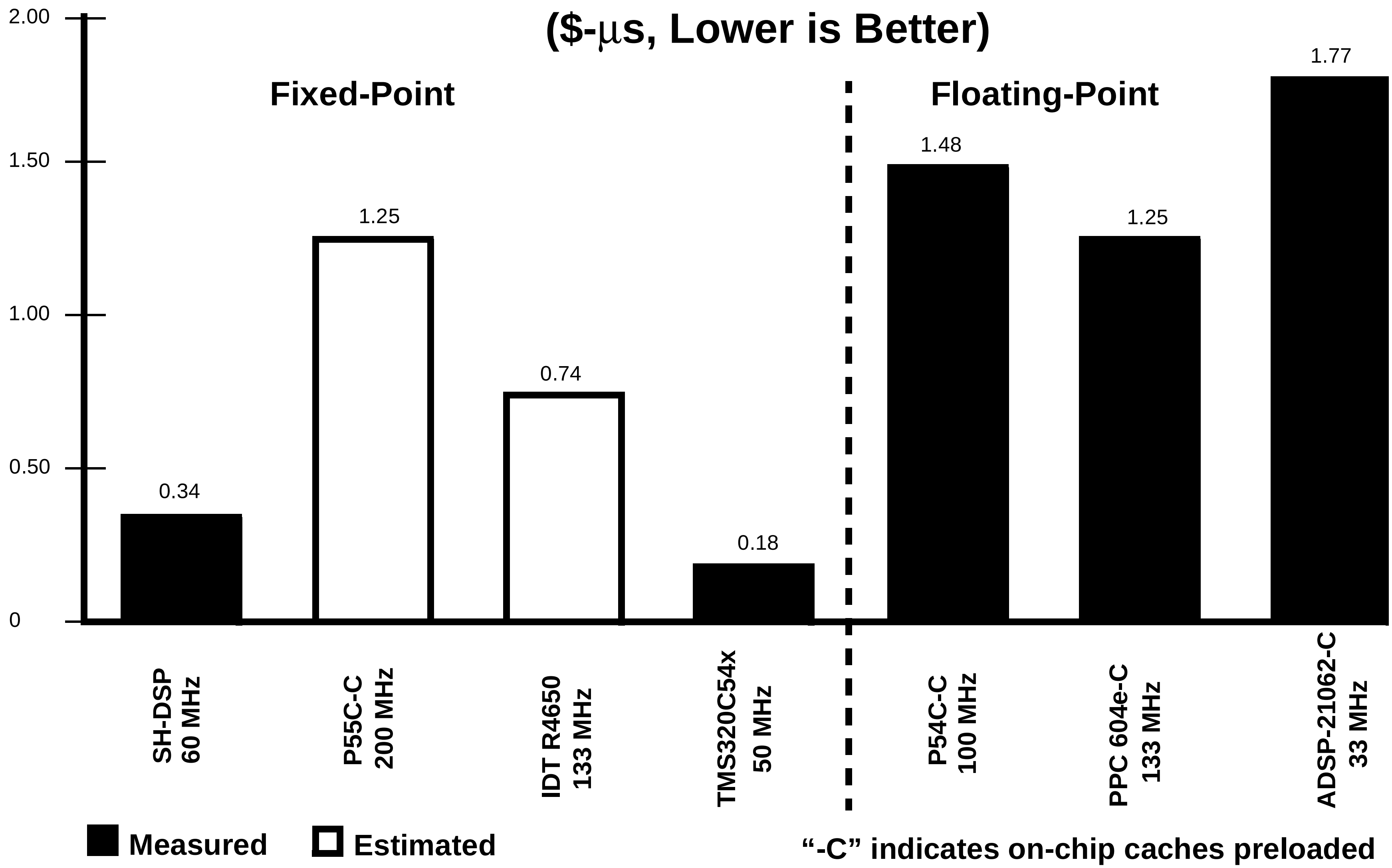
Pricing strategies are very different for PC-oriented GPPs vs. DSPs:

- **PC-oriented GPPs command > 2X price/performance <u>penalty</u> for the fastest versions. DSPs have their <u>best</u> price/perf. at high end.**

Example Cost

Type	Vendor	Processor	Speed (MHz)	Unit Price (Qty. 1,000)
Fixed-Point	Hitachi	SH-DSP	60	$45
	IDT	R4650	133	$63
		R4640	133	$34
	Intel	MMX Pentium (projected)	200	$550
	Texas Instr.	TMS320C548	66	$35
Floating-Point	Intel	Pentium	200	$509
			100	$106
	Motorola	PowerPC 604e	225	$620
			100	$173
	Analog Devices	ADSP-21062	40	$170

Cost-Execution Time Product, Block FIR Benchmark



"-C" indicates on-chip caches preloaded

Power Consumption

Power consumption is a key processor selection criteria in many important DSP applications.

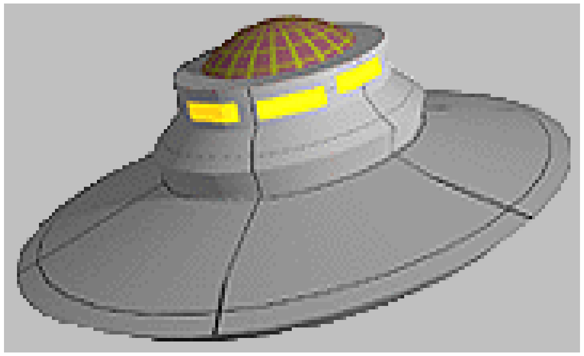
- **Today, only DSPs combine good DSP performance with very low power consumption and application-appropriate power management.**

Example data:

Vendor	Processor	Speed (MHz)	Voltage (V)	Typical Power Consumption (W)
Hitachi	SH-DSP	40	3.0	0.20 (est.)
IDT	R4650	133	3.3	2.1
Texas Instr.	TMS320C54x	50	3.0	0.11

Real-Time Suitability

Q: Why is running DSP code on general-purpose processors like alien abduction?



Real-Time Suitability

A: Both result in inexplicable gaps in time.

Real-Time Suitability

The most important DSP applications are real-time applications.

- **Many of these are "hard real-time" applications: failure to meet a real-time deadline creates a serious malfunction.**

High-performance GPPs make heavy use of dynamic features:

- **Caches, branch prediction, dynamic superscalar execution, data-dependent instruction execution times, etc.**

These features result in timing behavior that appears to be stochastic.

- **This seriously complicates development of DSP applications.**

PC applications are further complicated by the lack of real-time support in PC operating systems.

Product Development Time and Cost

Among the most important factors affecting development effort:

- **Breadth and quality of tools and documentation**
- **Processor ease of use**
- **Availability of off-the-shelf software libraries**

Developing DSP code for general-purpose processors requires using assembly language when efficiency is important.

- **High-performance GPPs are very difficult to program for DSP**
- **The most popular GPPs enjoy unparalleled tool support, but DSP-oriented tools are rare**
- **DSP software libraries for GPPs are few and far between**

Example of Optimization Challenge

Vector addition on PowerPC 604e:

```
@vec_add_loop:
 lfsu fpTemp1,4(rAAddr)       # Load A data, ptr. update
 lfsu fpTemp2,4(rBAddr)       # Load B data, ptr. update
 fadds fpSum,fpTemp1,fpTemp2  # Perform add operation
 stfsu fpSum,4(rCAddr)        # Store sum, ptr. update
bdnz @vec_add_loop            # loop
```

Q: How many instruction cycles per iteration?

Conclusions and Trends

Conclusions: Can GPPs Replace DSPs?

Today:

Yes:

- **In some PC DSP applications, the case is strong. Real-time behavior, OS support, and tools are weaknesses.**
- **In some embedded applications, especially where a μP or μC is already established, DSP algorithms are straightforward, and DSP performance needs are modest.**

And, no:

- **In many PC DSP applications, users needing the best quality and highest performance will benefit from DSPs and other specialized processors.**
- **In the most important embedded DSP applications, today's GPPs cannot compete: they have not pulled together all of the necessary attributes and infrastructure.**

Trends

- **DSP applications will continue to become increasingly important**
- **GPPs will continue to add and expand DSP-oriented enhancements**
- **DSP-oriented tools, software, and other infrastructure for GPPs will develop, but DSPs have a significant head-start**
- **DSPs will not stand still; there is fertile ground for architectural innovation, clock speed increases, etc.**
- **There will be an expanding diversity of processors; DSP and GPP family trees will mix**
- **Capabilities will become increasingly specialized for the wide range of important DSP applications**
- **GPPs will be suitable for an expanding range of DSP applications**

Further Resources

- BDTI technical reports:
  - *DSP on General-Purpose Processors* (just released)
  - *Buyer's Guide to DSP Processors*
- *Microprocessor Report* articles (especially 12/30/96, pp. 12-15)
- BDTI's web site: *www.bdti.com*
- Forward Concepts market research reports: *DSP Strategies 2000*
- DSP-oriented trade shows and conferences: ICSPAT, DSP World, etc.
- Join BDTI ... we're hiring