Insight, Analysis, and Advice on Signal Processing Technology

Microprocessors vs. DSPs (S044)

Kenton Williston
Berkeley Design Technology, Inc.
Berkeley, California USA
+1 (510) 665-1600

info@BDTI.com
http://www.BDTI.com

Workshop Outline

Definitions
DSP Algorithms Shape DSPs
Comparing DSPs and GPPs
Comparing Performance
When to Use Which
Conclusions

Definitions

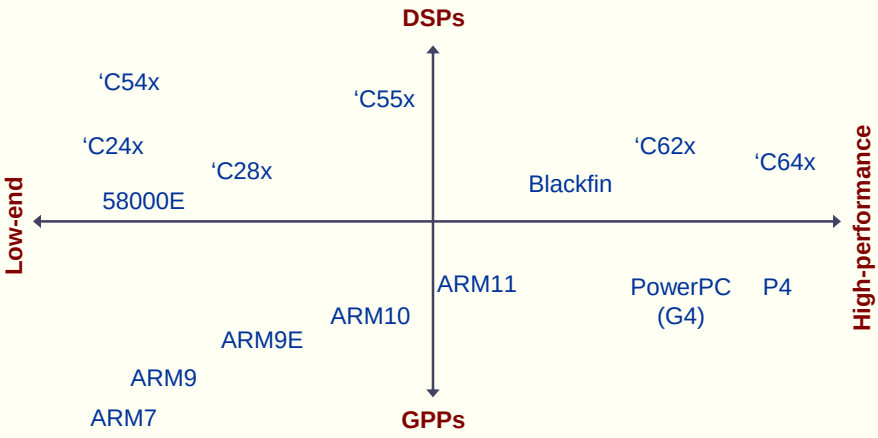
Microprocessors–General-Purpose Processors (GPPs)
- CPUs for PCs and workstations
  - E.g., Intel Pentium III
- 32-bit GPPs for embedded applications
  - E.g., ARM ARM7

Digital Signal Processors (DSPs)
- Microprocessors specialized for signal processing applications

Low-end DSPs and GPPs
- Architectures targeting extremely cost sensitive markets, often older architectures

High Performance DSPs and GPPs
- Architectures that use advanced techniques to improve parallelism, performance
- Usually have higher clock rates

Example Processors



DSP Algorithms Shape DSPs

How Signal Processing is Different From Other Tasks

- Very computationally demanding
- Requires attention to numeric fidelity
- High memory bandwidth requirements
- Streaming data—and lots of it
- Predictable data access patterns
- Execution-time locality
- Math-centric
- Real-time constraints
- Standards: algorithms, interfaces

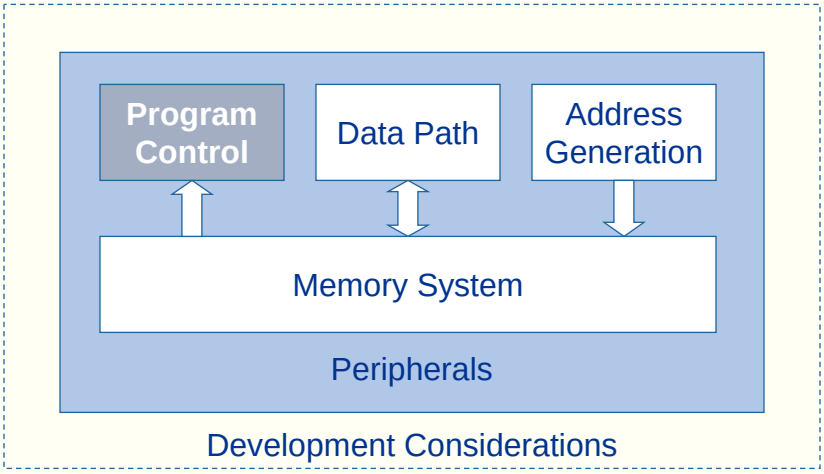
DSP Algorithms Shape DSPs

Computational demands	→	Multiple parallel execution units, hardware acceleration of common DSP functions
Numeric fidelity	→	Accumulator registers, guard bits, saturation hardware
High memory bandwidth	→	Harvard architecture, support for parallel moves
Predictable data access patterns	→	Specialized addressing modes, e.g., modulo, bit-reversed

DSP Algorithms Shape DSPs

Execution-time locality	→	Hardware looping, streamlined interrupt handling
Math-centricity	→	Single-cycle multiplier(s) or MAC unit(s), MAC instruction
Streaming data	→	Data memory usually SRAM, not cache; DMA
Real-time constraints	→	Few dynamic features, on-chip SRAM instead of cache
Standards	→	16-bit data types; rounding, saturation modes

Key Processor Attributes



Comparing DSPs and GPPs
Instruction Set

Low-end DSP	Low-end GPP
Specialized, complex instructions	General-purpose instructions
Multiple operations per instruction	Typically only one operation per instruction
Poor orthogonality	Good orthogonality

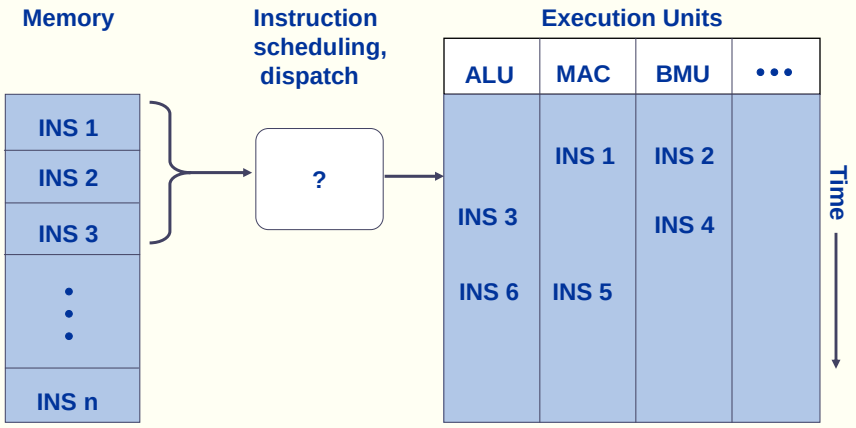
```
mac x0,y0,a x:(r0)+,x0 y:(r4)+,y0
```

```
mpy  r2,r3,r4
add  r4,r5,r5
mov  (r0),r2
mov  (r1),r3
inc  r0
inc  r1
```


Comparing DSPs and GPPs
Instruction Set

High-performance DSP	High-performance GPP
	Baseline:
Simple to moderately-complex instructions	Simple instructions
Moderate to excellent orthogonality	Moderate to excellent orthogonality
	With SIMD extensions:
	Moderately complex instructions
	Moderate to excellent orthogonality

Multi-Issue Approaches
VLIW vs. Superscalar



Comparing DSPs and GPPs
Architecture Type

Low-end DSP & GPP

Single-issue

- DSP
  - Compound instructions perform multiple operations, e.g., multiply + load + modify address register
  - Examples: 'C54x, 'C24x, 'C28x
- GPP
  - RISC instructions perform single operation, e.g., add, load, or store
  - Examples: ARM7, ARM9

High-performance DSP & GPP

Multi-issue

- DSP
  - Typically VLIW
  - Up to 8 instructions/cycle
  - Examples: 'C64x, SC140, TigerSHARC
- GPP
  - Typically superscalar
  - Up to 4 instructions/cycle
  - Example: PowerPC 74xx

Comparing DSPs and GPPs

Trade-offs: Superscalar vs. VLIW

Superscalar (high-performance GPPs, mostly)

- Increased hardware complexity
  - Silicon area, power consumption
- Dynamic behavior
  - Complex performance model, timing variability
- Increased performance with binary compatibility
- Decreased software complexity (programmer/compiler)

VLIW (high-performance DSPs, mostly)

- Decreased hardware complexity
- No dynamic behavior
- Binary compatibility difficult (downward direction)
- Increased software complexity

Comparing DSPs and GPPs

Program Control

Low-end DSP	Low-end GPP
Hardware looping	Software looping
Interrupts disabled during certain operations	Interrupts rarely disabled
Limited or no register shadowing	Register shadowing common
Simple pipelines • Often provide delay slots to hide branch latencies	Simple pipelines • No delay slots or branch prediction
May support fast interrupts	May support fast interrupts

Comparing DSPs and GPPs
Program Control

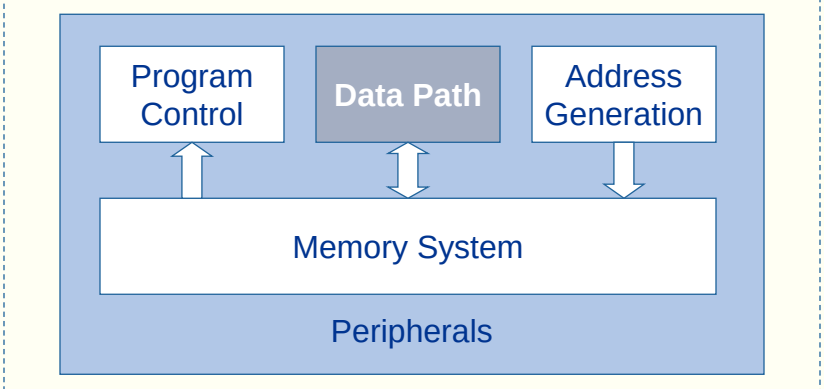
High-end DSP	High-end GPP
Usually support hardware looping	Software looping
Interrupts rarely disabled	Interrupts rarely disabled
May offer shadow registers	Register shadowing common
Complicated pipelines in some cases • May be non-interlocked • May have multi-cycle latencies • May use branch prediction	Moderately to extremely complicated pipelines • May have very long instruction latencies • Often use branch prediction
May support fast interrupts	May support fast interrupts

Comparing DSPs and GPPs
Branch Prediction: Strengths and Weaknesses

In many applications, branch prediction is very accurate
- This includes signal processing applications, where most branches are part of for-next loops

Complex branch prediction algorithms introduce timing uncertainty
- It can be difficult to predict whether the prediction will be correct at any given instant

Key Processor Attributes



Comparing DSPs and GPPs

Data Path

Low-end DSP	Low-end GPP
Dedicated hardware performs all key arithmetic operations in 1 cycle	Multiplies often take >1 cycle
	Multi-bit shifts often take >1 cycle
Usually 16-bit	Usually 32-bit, integer only
Hardware support for managing numeric fidelity • Guard bits, saturation, rounding modes, …	Saturation, rounding typically take extra cycles
Limited bit-manipulation capabilities	May have superior bit-manipulation capabilities

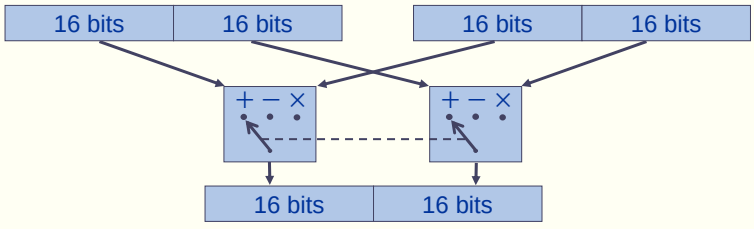
Comparing DSPs and GPPs

Data Path

High-performance DSP	High-performance GPP
Up to 8 arithmetic units	1-3 arithmetic units
Some specialized arithmetic units • E.g., MAC unit, Viterbi unit	General-purpose arithmetic units • E.g., integer unit, floating-point unit
Support multiple data sizes	Support multiple data sizes
Limited to excellent bit-manipulation capabilities	May have superior bit-manipulation capabilities
Hardware support for managing numeric fidelity	Saturation, rounding typically take extra cycles

SIMD

Single Instruction, Multiple Data



Performs the same operation simultaneously on multiple sets of operands

- Under the control of a single instruction

Some SIMD processors support multiple data widths (for example, 32-bit, 16-bit, and 8-bit)

Comparing DSPs and GPPs
SIMD Features

Low-end DSP & GPP

DSPs: very limited SIMD features
- E.g., dual add, subtract of 16-bit fixed-point data

GPPs: No SIMD support

High-performance DSP & GPP

DSPs: limited to extensive SIMD features
- E.g., TigerSHARC
  - 4 x 32-bit float
  - 4 x 32-bit integer
  - 8 x 16-bit integer
  - 16 x 8-bit integer

GPPs: extensive SIMD features
- E.g., PowerPC 74xx
  - 4 x 32-bit float
  - 4 x 32-bit integer
  - 8 x 16-bit integer
  - 16 x 8-bit integer

SIMD Challenges

Each instruction performs lots of work
- Data parallelism

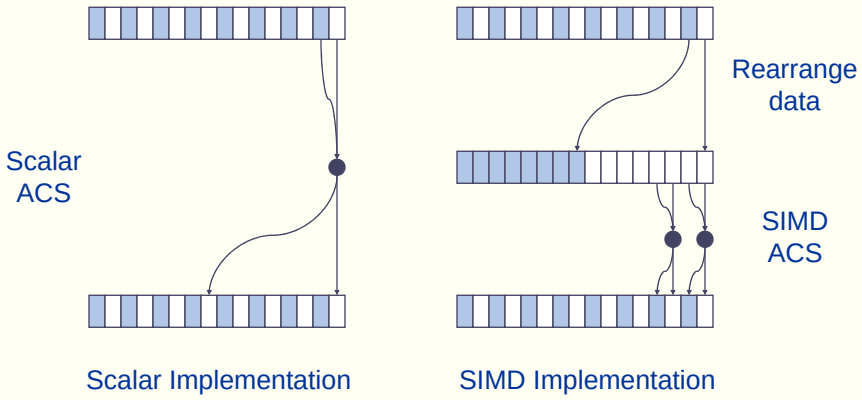
Algorithms, data organization must be amenable to data-parallel processing
- May require programmer creativity, alternative algorithms
- Data-reorganization penalties can be significant

Compilers generally don't use SIMD capabilities

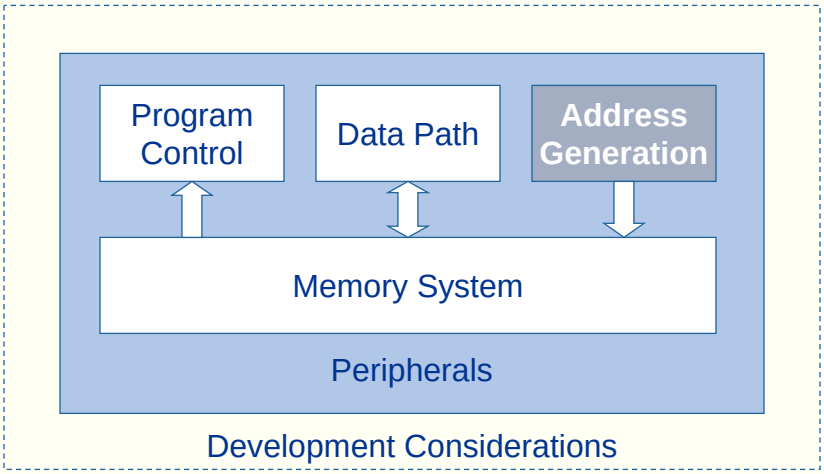
Most effective on algorithms that process large blocks of data

SIMD Challenges

Example: Viterbi Add-Compare-Select (ACS) Loop



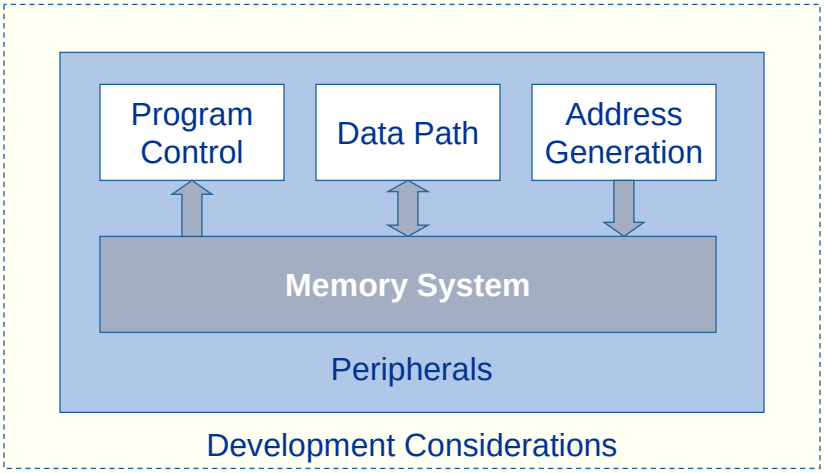
Key Processor Attributes



Comparing DSPs and GPPs
Addressing

Low-end and High-Performance DSP	Low-end and High-Performance GPP
Dedicated address-generation units	Often, no separate address-generation units
Specialized addressing modes • Autoincrement • Modulo (circular) • Bit-reversed (for FFT)	General-purpose addressing modes

Key Processor Attributes



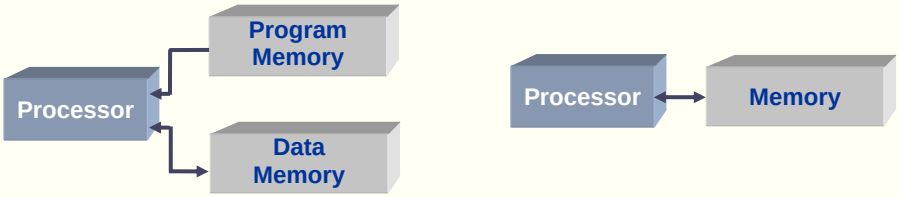
Comparing DSPs and GPPs
Memory Architecture

Low-end DSP

Harvard architecture
2-4 memory accesses per cycle
No caches; on-chip SRAM
DMA

Low-end GPP

Von Neumann architecture
Typically 1 access per cycle
Typically use cache(s)



Comparing DSPs and GPPs
Memory Architecture

High-performance DSP

Harvard architecture

Per cycle accesses:
- 1-8 instructions
- Two or more 16- to 64-bit data words

Sometimes caches, often lockable, configurable as SRAM

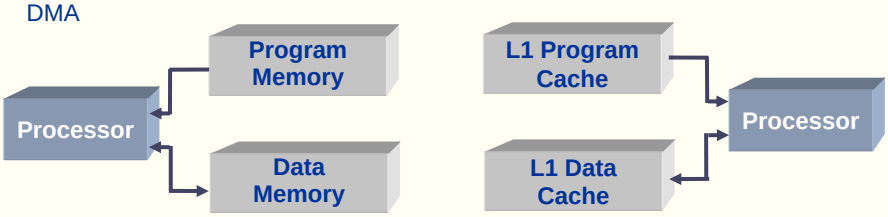
DMA

High-performance GPP

Harvard architecture

Per cycle accesses:
- 1-4 instructions
- ~Two 32- to 64-bit or one 128-bit data word

Usually use caches



Comparing DSPs and GPPs
Caches: Challenges

Caches work by lowering average access time
- They are effective at doing this in many applications
- But access times vary significantly

Some applications are sensitive to maximum access time (not average)
- E.g., many “hard-real-time” signal processing applications

Signal processing access patterns often predictable
- Thus, DMA may be preferable to a cache
- Some recent caches provide pre-fetching capability
- Some DSPs’ caches can be locked or configured as part cache, part SRAM

Comparing DSPs and GPPs
Dynamic Features

Dynamic features are used heavily in high-end GPPs to boost performance
- Superscalar execution
- Caches
- Branch prediction
- Data-dependent instruction execution times

These features are occasionally used in DSPs, too

These features complicate software development for real-time DSP applications
- Ensuring real-time behavior
- Optimizing code

Comparing DSPs and GPPs

Dynamic Features

Low-end GPPs and DSPs

GPPs:
- Dynamic caches common

DSPs:
- Rarely have dynamic features
  - Small "loop buffer" instruction cache exception

High-performance GPPs and DSPs

GPPs: Moderate to extensive use of dynamic features
- Dynamic caches standard
- Superscalar execution, branch prediction common

DSPs: Generally avoid dynamic features
- Dynamic cache is most common dynamic feature
- Superscalar execution rare
- Branch prediction sometimes used

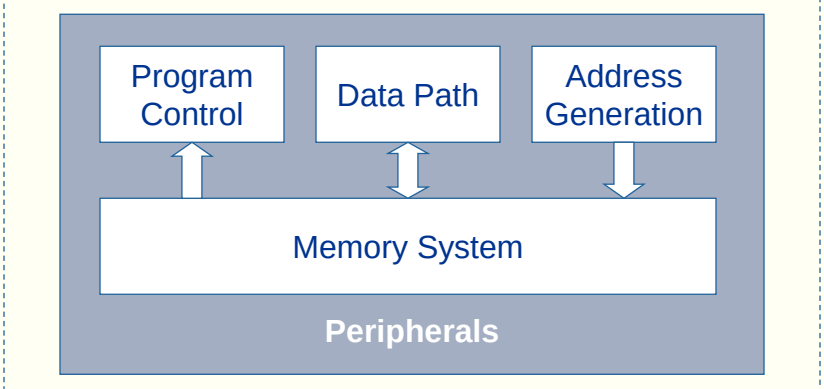
Parallelism: Implications of Differences

Cycle efficiency
- DSPs have advantage on signal processing tasks
  - But may require special software development strategies—like assembly level programming—to realize full advantage

Memory use efficiency
- Multi-operation instructions give DSPs advantage on signal processing tasks
- But GPPs often better on non-signal processing tasks—which typically consumes most of the code space

Compiler friendliness
- GPPs generally have the advantage
- SIMD difficult for compilers, whether GPP or DSP
  - Often requires assembly programming or use of high level intrinsics—both of which complicate software development

Key Processor Attributes

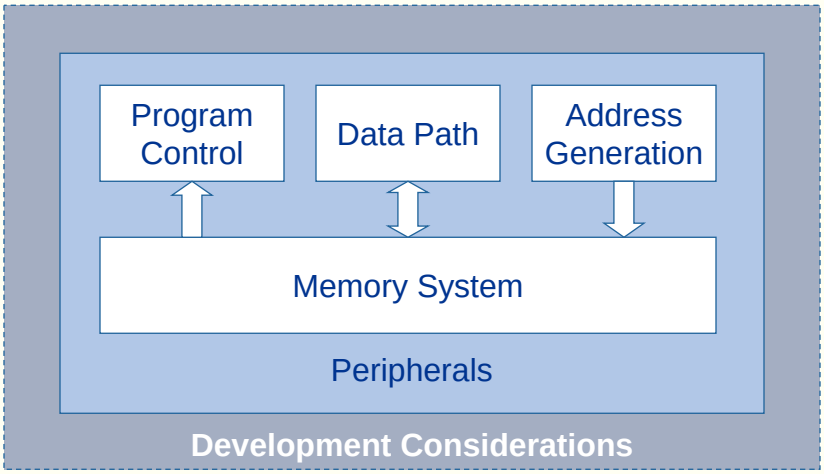


Comparing DSPs and GPPs

On-Chip Integration

Low-end GPPs and DSPs	High-performance GPPs and DSPs
Typically, wide range of on-chip peripherals and I/O interfaces	Moderate to extensive on-chip integration • PC CPUs offer very little on-chip integration
Often oriented towards consumer applications • E.g., video coprocessors, USB ports, …	Often oriented towards communications infrastructure • E.g., Viterbi decoding coprocessors, UTOPIA ports, …

Key Processor Attributes



Comparing DSPs and GPPs
Compatibility and Availability

Low-end DSP	Low-end GPP
Mostly proprietary architectures • I.e., one architecture, one vendor	Many shared architectures • I.e., one architecture, several (to many) vendors
Limited (at best) compatibility between successive generations	Often binary compatibility between successive generations
Occasionally available as licensable core	Often available as licensable core • E.g., ARM, MIPS

Comparing DSPs and GPPs
Compatibility and Availability

High-performance DSP	High-performance GPP
Mostly proprietary architectures • Exceptions: StarCore, ZSP	Mostly shared architectures • PowerPC, MIPS, ARM, x86
Sometimes binary compatibility between successive generations • E.g., 'C6xxx, StarCore, ZSP	Usually binary compatibility between successive generations
Sometimes available as licensable core • E.g., StarCore, CEVA-X, ZSP	Sometimes available as licensable core • E.g., ARM, MIPS

Comparing DSPs and GPPs
Development Support

	DSPs	GPPs
Tools	Primitive to moderately sophisticated	Primitive to very sophisticated
DSP-specific tool support	Good to excellent E.g., cycle-accurate simulators, DSP C extensions	Poor but improving E.g., general lack of cycle-accurate simulators
3rd-party DSP software support	Poor to excellent	Limited but growing
Non-DSP 3rd-party software support	Poor Few to moderate RTOS options	Extensive Few to extensive RTOS options
Links w/other high-level tools	E.g., MATLAB	E.g., GUI builders

Comparing Performance

When evaluating processors for signal processing, application-specific, product-specific considerations dominate

- Relative performance can vary dramatically depending on the benchmark

Vendor performance claims should be viewed skeptically

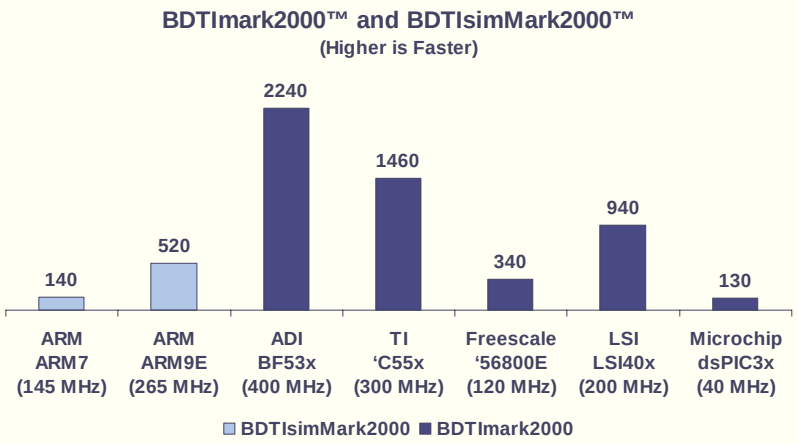
- "MIPS" = …
- Benchmarks are a sharp tool

Performance is more than speed

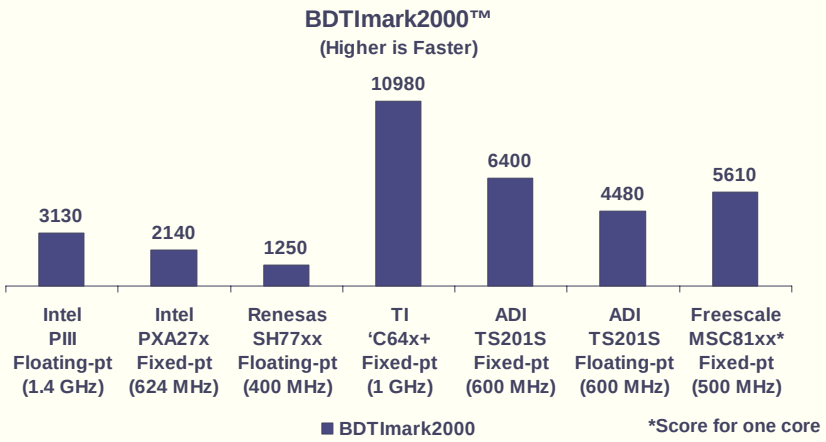
- Cost/perf, energy efficiency, memory use …

Comparing Performance

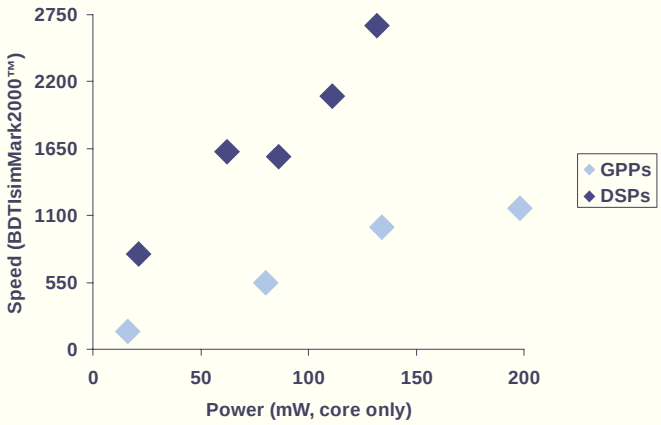
Speed: Low-end DSPs/GPPs (Below $10)



Comparing Performance
Speed: High-performance DSPs/GPPs



Comparing Performance
Speed vs. Power: Licensable Cores



All data assumes use of TSMC CL013G process and ARM Artisan SAGE-X library

Conclusions
When to Use Which

DSP	GPP
Heavy signal processing requirements	Modest signal processing requirements
Limited control processing	Extensive control processing • Especially if code density and portability are important
The DSP is incumbent	The GPP is incumbent
Software compatibility between generations not required—or can be achieved w/ DSP	Software compatibility between generations required
Multi-vendor architecture not desired	Multi-vendor architecture desired
DSP has better integration for application	GPP has better integration for application

Conclusions
Will DSP-Capable GPPs Render DSPs Obsolete?

No, but they will pose increasingly strong competition
- Why have GPP+DSP if GPP alone is good enough?

Demands of most communications and media-processing applications will continue to favor DSPs

Software infrastructure is key
- DSPs have the advantage for DSP tasks
- GPPs have the advantage for other tasks

For DSPs, the competitive field has become much larger
- Differentiating criteria are changing

For More Information…
www.BDTI.com

Inside DSP newsletter and quarterly reports

Benchmark scores for dozens of processors

Pocket Guide to Processors for DSP

- Basic stats on over 40 processors

Articles, white papers, and presentation slides

- Processor architectures and performance
- Signal processing applications
- Signal processing software optimization

comp.dsp FAQ

6th Edition

INSIGHT • ANALYSIS • ADVICE
ON SIGNAL PROCESSING TECHNOLOGY