


# Microprocessors vs. DSPs

*Insight, Analysis, and Advice on Signal Processing Technology*




## Microprocessors vs. DSPs (ESC-223)

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

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

© 2006 Berkeley Design Technology, Inc.



## Workshop Outline

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

© 2006 BDTI

INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY

2

© 2006 Berkeley Design Technology, Inc.

# Microprocessors vs. DSPs

**BDTi**

## 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

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY      3

**BDTi**


## Example Processors

The diagram is a 2D plot with a vertical axis and a horizontal axis. The vertical axis is labeled 'DSPs' at the top and 'GPPs' at the bottom. The horizontal axis is labeled 'Low-end' on the left and 'High-performance' on the right. Processors are plotted as follows:

- Top-Left (High Performance, Low-end): 'C54x, 'C24x, 'C28x, 58000E
- Top-Center (High Performance, Mid-range): 'C55x
- Top-Right (High Performance, High-performance): Blackfin, 'C62x, 'C64x
- Bottom-Left (Low Performance, Low-end): ARM7, ARM9, ARM9E, ARM10
- Bottom-Center (Low Performance, Mid-range): ARM11
- Bottom-Right (Low Performance, High-performance): PowerPC (G4), P4

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY      4

© 2006 Berkeley Design Technology, Inc.




## 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

© 2006 BDTIINSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY5



## 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

© 2006 BDTIINSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY6

# Microprocessors vs. DSPs

**BDTi**

## 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

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      7


**BDTi**

## Key Processor Attributes

Development Considerations

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      8

© 2006 Berkeley Design Technology, Inc.



## Comparing DSPs and GPPs


### Instruction Set

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

<code>mac x0,y0,a x:(r0)+,x0 y:(r4)+,y0</code>	<code>mpy r2,r3,r4</code>
	<code>add r4,r5,r5</code>
	<code>mov (r0),r2</code>
	<code>mov (r1),r3</code>
	<code>inc r0</code>
	<code>inc r1</code>

© 2006 BDTI INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY 9

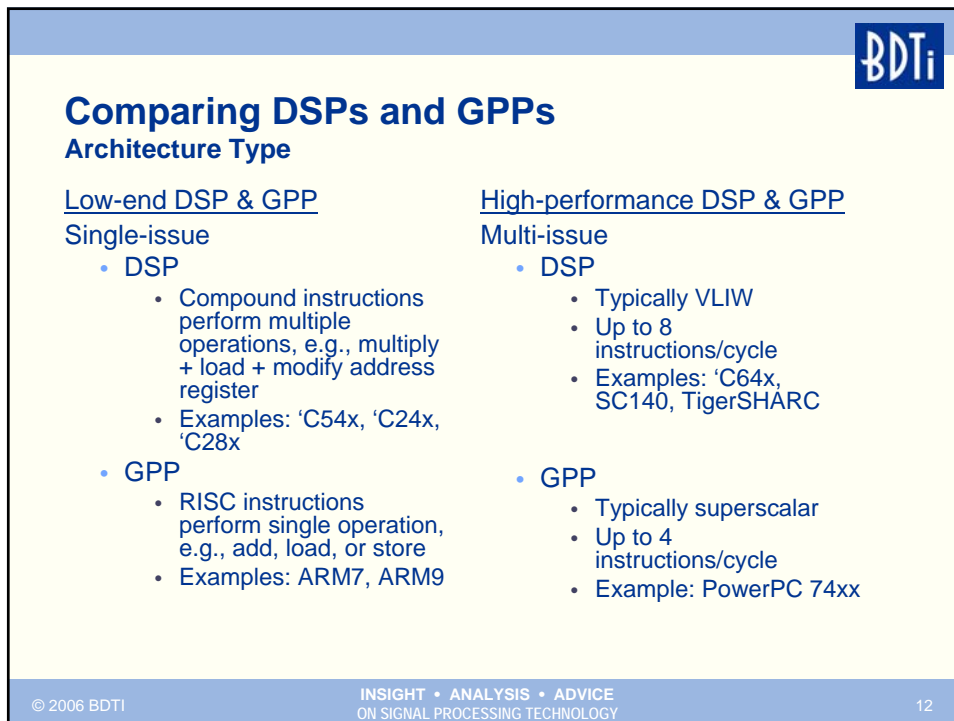
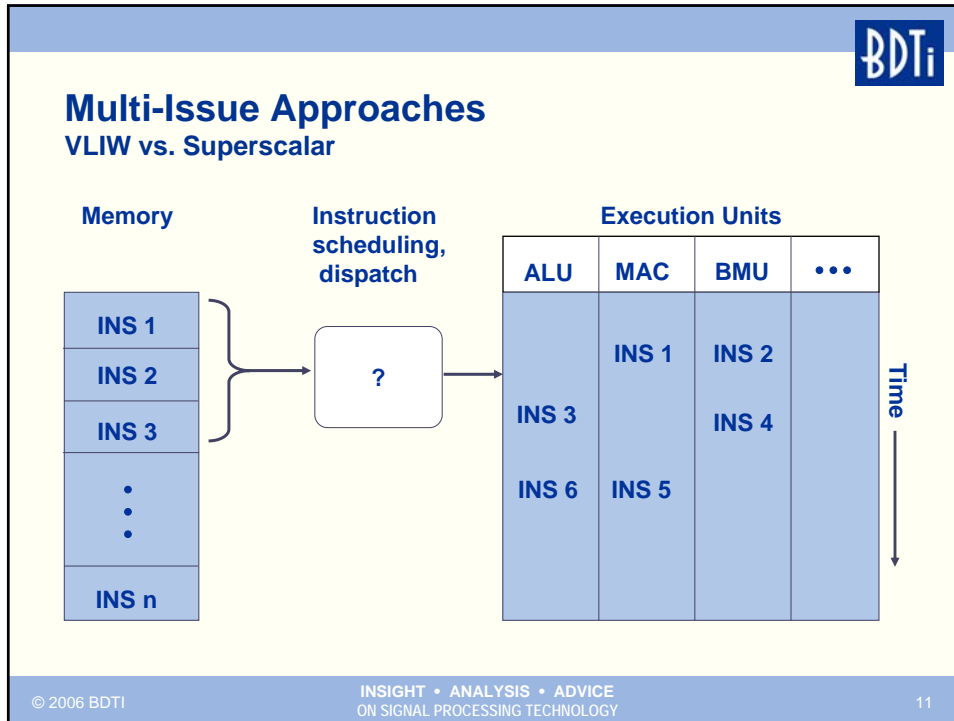


## Comparing DSPs and GPPs

### Instruction Set

<u>High-performance DSP</u>	<u>High-performance GPP</u>
Simple to moderately-complex instructions	<i>Baseline:</i> Simple instructions
Moderate to excellent orthogonality	Moderate to excellent orthogonality
	<i>With SIMD extensions:</i> Moderately complex instructions
	Moderate to excellent orthogonality

© 2006 BDTI INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY 10





## 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

Hardware looping  
Interrupts disabled during certain operations  
Limited or no register shadowing  
Simple pipelines

- Often provide delay slots to hide branch latencies


May support fast interrupts

#### Low-end GPP

Software looping  
Interrupts rarely disabled  
Register shadowing common  
Simple pipelines

- No delay slots or branch prediction

May support fast interrupts




## Comparing DSPs and GPPs

### Program Control

<p><u>High-end DSP</u></p> <p>Usually support hardware looping</p> <p>Interrupts rarely disabled</p> <p>May offer shadow registers</p> <p>Complicated pipelines in some cases</p> <ul style="list-style-type: none"><li>• May be non-interlocked</li><li>• May have multi-cycle latencies</li><li>• May use branch prediction</li></ul> <p>May support fast interrupts</p>	<p><u>High-end GPP</u></p> <p>Software looping</p> <p>Interrupts rarely disabled</p> <p>Register shadowing common</p> <p>Moderately to extremely complicated pipelines</p> <ul style="list-style-type: none"><li>• May have very long instruction latencies</li><li>• Often use branch prediction</li></ul> <p>May support fast interrupts</p>
--	--

© 2006 BDTIINSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY15



## 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

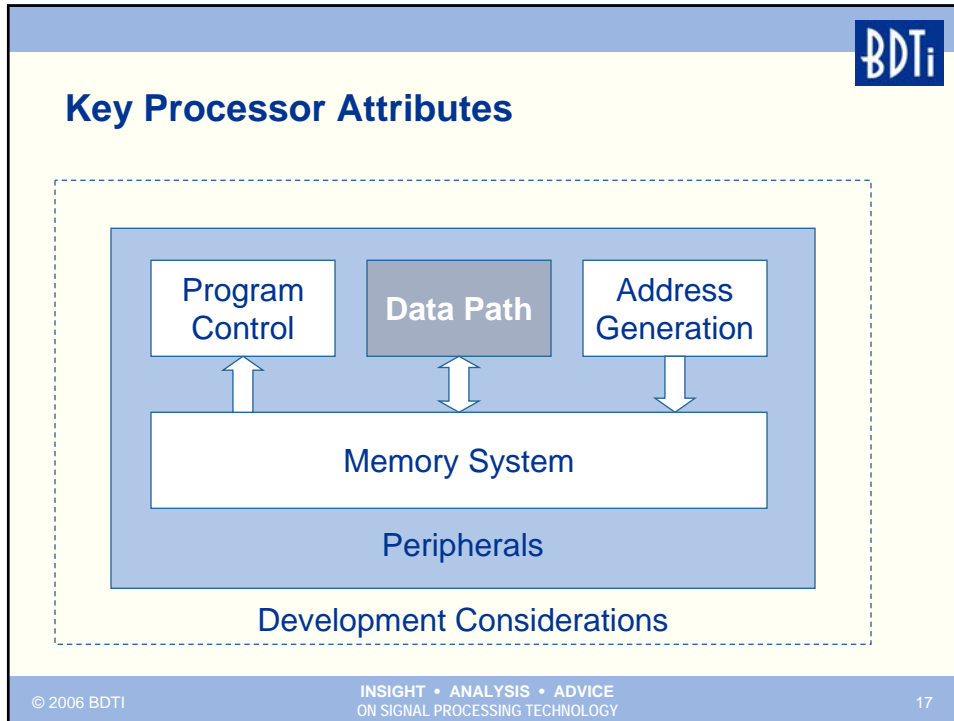
Complex branch prediction algorithms introduce timing uncertainty

- It can be difficult to predict whether the prediction will be correct at any given instant

© 2006 BDTIINSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY16



# Microprocessors vs. DSPs



© 2006 BDTI

INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY

17

The table, titled "Comparing DSPs and GPPs", compares the "Data Path" characteristics of a "Low-end DSP" and a "Low-end GPP". The BDTi logo is in the top right corner.

Low-end DSP	Low-end GPP
Dedicated hardware performs all key arithmetic operations in 1 cycle	Multiplies often take >1 cycle
Usually 16-bit	Multi-bit shifts often take >1 cycle
Hardware support for managing numeric fidelity <ul style="list-style-type: none"><li>• Guard bits, saturation, rounding modes, ...</li></ul>	Usually 32-bit, integer only
Limited bit-manipulation capabilities	Saturation, rounding typically take extra cycles
	May have superior bit-manipulation capabilities

© 2006 BDTI

INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY

18

© 2006 Berkeley Design Technology, Inc.

## Comparing DSPs and GPPs

### Data Path

#### High-performance DSP

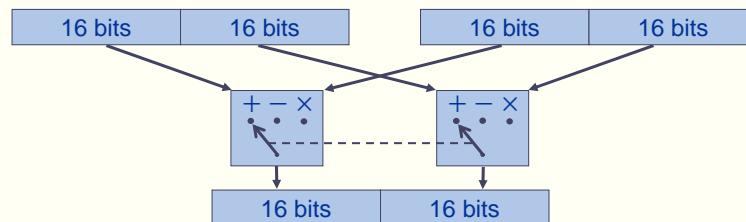
- Up to 8 arithmetic units
- Some specialized arithmetic units
  - E.g., MAC unit, Viterbi unit
- Support multiple data sizes
- Limited to excellent bit-manipulation capabilities
- Hardware support for managing numeric fidelity

#### High-performance GPP

- 1-3 arithmetic units
- General-purpose arithmetic units
  - E.g., integer unit, floating-point unit
- Support multiple data sizes
- May have superior bit-manipulation capabilities
- 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

<p><u>Low-end DSP &amp; GPP</u></p> <p>DSPs: very limited SIMD features</p> <ul style="list-style-type: none"><li>• E.g., dual add, subtract of 16-bit fixed-point data</li></ul> <p>GPPs: No SIMD support</p>	<p><u>High-performance DSP &amp; GPP</u></p> <p>DSPs: limited to extensive SIMD features</p> <ul style="list-style-type: none"><li>• E.g., TigerSHARC<ul style="list-style-type: none"><li>• 4 x 32-bit float</li><li>• 4 x 32-bit integer</li><li>• 8 x 16-bit integer</li><li>• 16 x 8-bit integer</li></ul></li></ul> <p>GPPs: extensive SIMD features</p> <ul style="list-style-type: none"><li>• E.g., PowerPC 74xx<ul style="list-style-type: none"><li>• 4 x 32-bit float</li><li>• 4 x 32-bit integer</li><li>• 8 x 16-bit integer</li><li>• 16 x 8-bit integer</li></ul></li></ul>
--	---

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      21



## 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

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      22

**BDTi**

## SIMD Challenges

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

Scalar ACS

Scalar Implementation

Rearrange data

SIMD ACS

SIMD Implementation

© 2006 BDTI

INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY

23

**BDTi**

## Key Processor Attributes

Program Control

Data Path

Address Generation

Memory System

Peripherals

Development Considerations

© 2006 BDTI

INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY

24

# Microprocessors vs. DSPs

**BDTi**

## Comparing DSPs and GPPs

### Addressing

<u>Low-end and High-Performance DSP</u>	<u>Low-end and High-Performance GPP</u>
Dedicated address-generation units	Often, no separate address-generation units
Specialized addressing modes <ul style="list-style-type: none"><li>• Autoincrement</li><li>• Modulo (circular)</li><li>• Bit-reversed (for FFT)</li></ul>	General-purpose addressing modes

© 2006 BDTI INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY 25

**BDTi**

## Key Processor Attributes

The diagram illustrates the key attributes of a processor. It features three main components at the top: Program Control, Data Path, and Address Generation. These are connected to a central Memory System. Below the Memory System is the Peripherals section. The entire system is enclosed in a dashed box labeled Development Considerations. Arrows indicate the flow of information: Program Control has an upward arrow to the Memory System; Data Path has a bidirectional arrow to the Memory System; and Address Generation has a downward arrow to the Memory System.

© 2006 BDTI INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY 26

© 2006 Berkeley Design Technology, Inc.

# Microprocessors vs. DSPs

**BDTi**

## Comparing DSPs and GPPs

### Memory Architecture

<u>Low-end DSP</u>  Harvard architecture 2-4 memory accesses per cycle No caches; on-chip SRAM DMA	<u>Low-end GPP</u>  Von Neumann architecture Typically 1 access per cycle Typically use cache(s)
---	--

```
graph LR; P[Processor] <--> PM[Program Memory]; P <--> DM[Data Memory];
```

```
graph LR; P[Processor] <--> M[Memory];
```

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      27

**BDTi**

## Comparing DSPs and GPPs

### Memory Architecture

<u>High-performance DSP</u>  Harvard architecture Per cycle accesses: <ul style="list-style-type: none"><li>• 1-8 instructions</li><li>• Two or more 16- to 64-bit data words</li></ul> Sometimes caches, often lockable, configurable as SRAM DMA	<u>High-performance GPP</u>  Harvard architecture Per cycle accesses: <ul style="list-style-type: none"><li>• 1-4 instructions</li><li>• ~Two 32- to 64-bit or one 128-bit data word</li></ul> Usually use caches
--	--

```
graph LR; P[Processor] <--> PM[Program Memory]; P <--> DM[Data Memory];
```

```
graph LR; P[Processor] <--> L1PC[L1 Program Cache]; P <--> L1DC[L1 Data Cache];
```

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      28

© 2006 Berkeley Design Technology, Inc.



## 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

<p><u>Low-end GPPs and DSPs</u></p> <p><b>GPPs:</b></p> <ul style="list-style-type: none"><li>• Dynamic caches common</li></ul> <p><b>DSPs:</b></p> <ul style="list-style-type: none"><li>• Rarely have dynamic features</li><li>• Small “loop buffer” instruction cache exception</li></ul>	<p><u>High-performance GPPs and DSPs</u></p> <p><b>GPPs:</b> Moderate to extensive use of dynamic features</p> <ul style="list-style-type: none"><li>• Dynamic caches standard</li><li>• Superscalar execution, branch prediction common</li></ul> <p><b>DSPs:</b> Generally avoid dynamic features</p> <ul style="list-style-type: none"><li>• Dynamic cache is most common dynamic feature</li><li>• Superscalar execution rare</li><li>• Branch prediction sometimes used</li></ul>
--	--

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      31



## 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

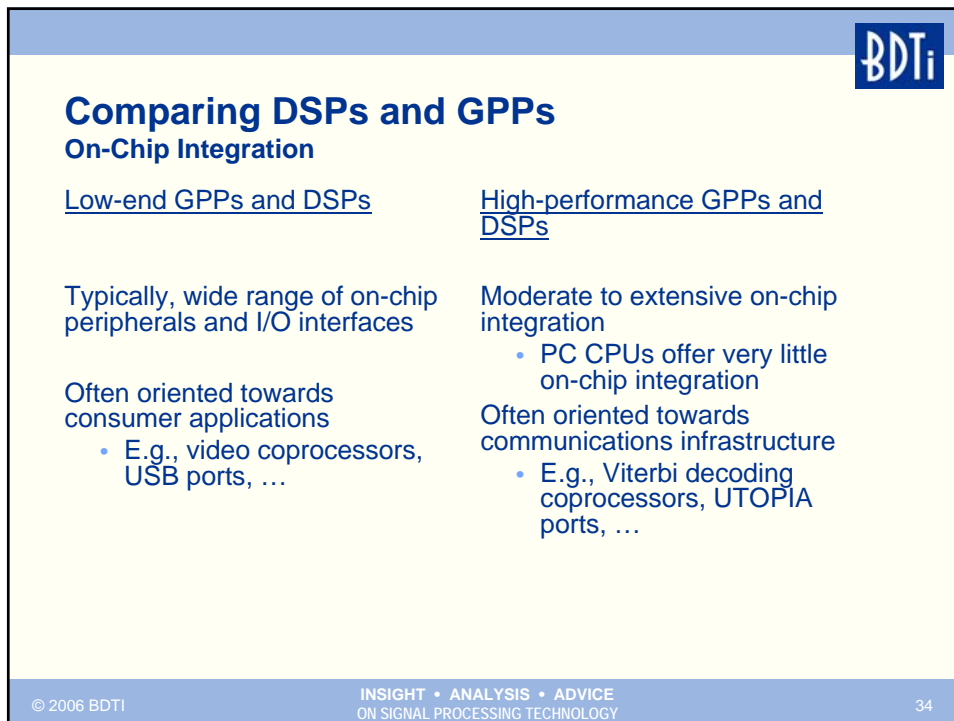
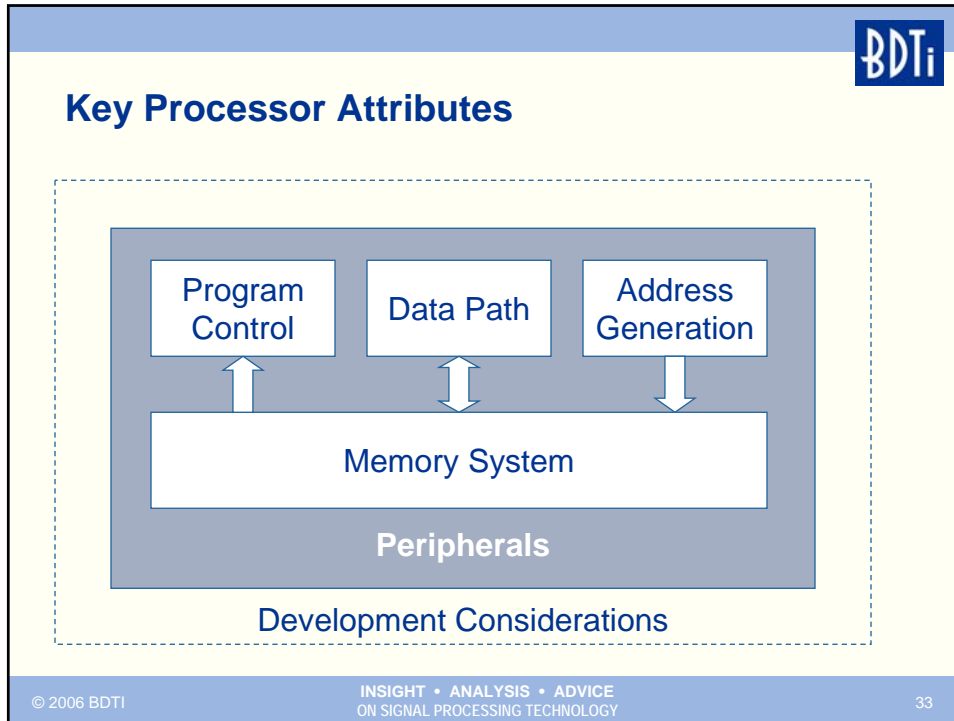
**Compiler friendliness**

- 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

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE ON SIGNAL PROCESSING TECHNOLOGY      32

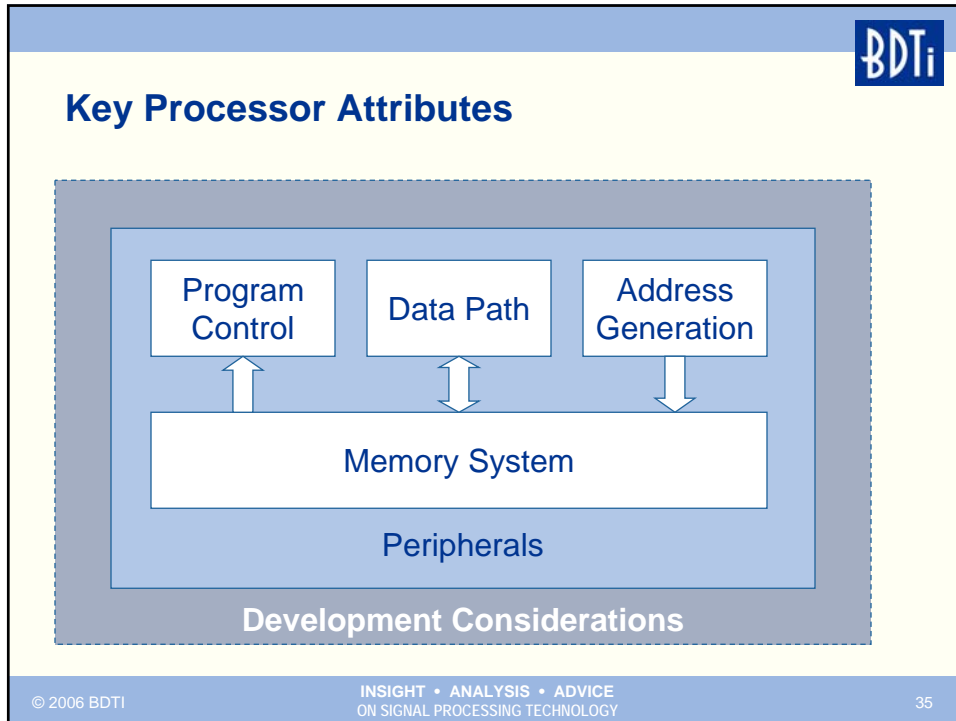


# Microprocessors vs. DSPs



© 2006 Berkeley Design Technology, Inc.

# Microprocessors vs. DSPs




The table, titled "Comparing DSPs and GPPs Compatibility and Availability", compares Low-end DSP and Low-end GPP. The BDTi logo is in the top right corner.

<u>Low-end DSP</u>	<u>Low-end GPP</u>
Mostly proprietary architectures <ul style="list-style-type: none"><li>• I.e., one architecture, one vendor</li></ul>	Many shared architectures <ul style="list-style-type: none"><li>• I.e., one architecture, several (to many) vendors</li></ul>
Limited (at best) compatibility between successive generations	Often binary compatibility between successive generations
Occasionally available as licensable core	Often available as licensable core <ul style="list-style-type: none"><li>• E.g., ARM, MIPS</li></ul>

© 2006 BDTI      INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY      36

© 2006 Berkeley Design Technology, Inc.




## Comparing DSPs and GPPs

### Compatibility and Availability

<p><u>High-performance DSP</u></p> <p>Mostly proprietary architectures</p> <ul style="list-style-type: none"> <li>• Exceptions: StarCore, ZSP</li> </ul> <p>Sometimes binary compatibility between successive generations</p> <ul style="list-style-type: none"> <li>• E.g., 'C6xxx, StarCore, ZSP</li> </ul> <p>Sometimes available as licensable core</p> <ul style="list-style-type: none"> <li>• E.g., StarCore, CEVA-X, ZSP</li> </ul>	<p><u>High-performance GPP</u></p> <p>Mostly shared architectures</p> <ul style="list-style-type: none"> <li>• PowerPC, MIPS, ARM, x86</li> </ul> <p>Usually binary compatibility between successive generations</p> <p>Sometimes available as licensable core</p> <ul style="list-style-type: none"> <li>• E.g., ARM, MIPS</li> </ul>
---	--

© 2006 BDTI
INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY
37




## Comparing DSPs and GPPs

### Development Support

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

© 2006 BDTI
INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY
38



## 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 ...

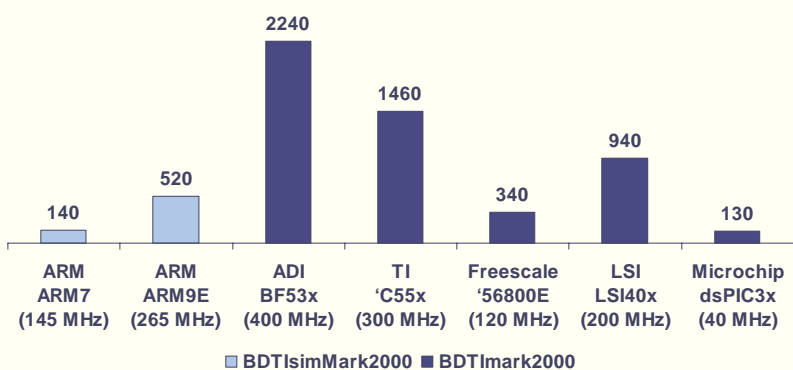
© 2006 BDTI
INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY
39



## Comparing Performance

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

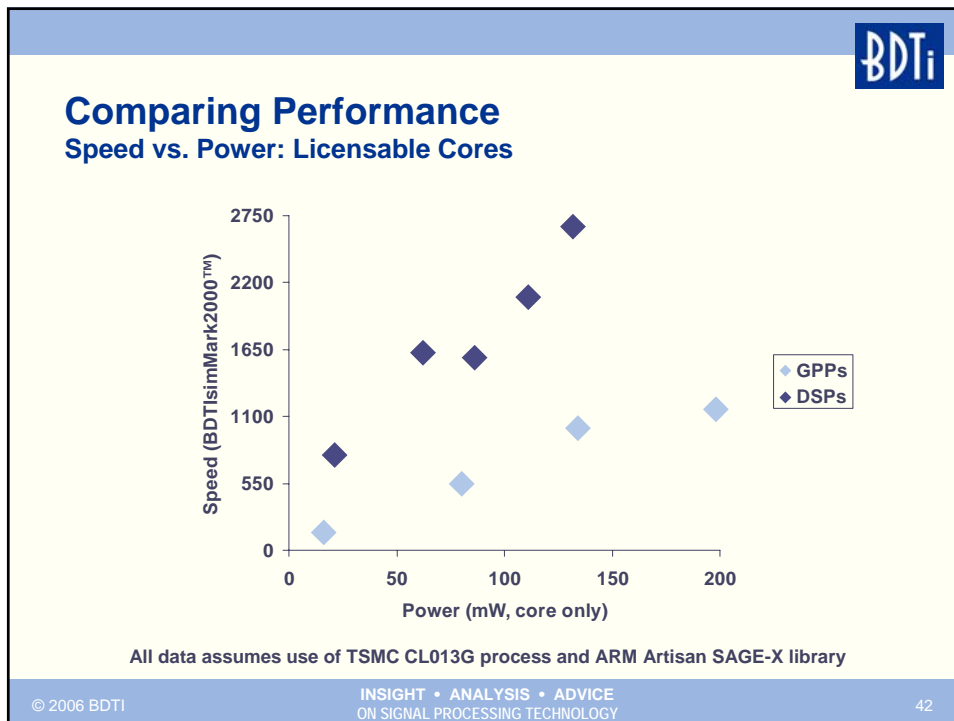
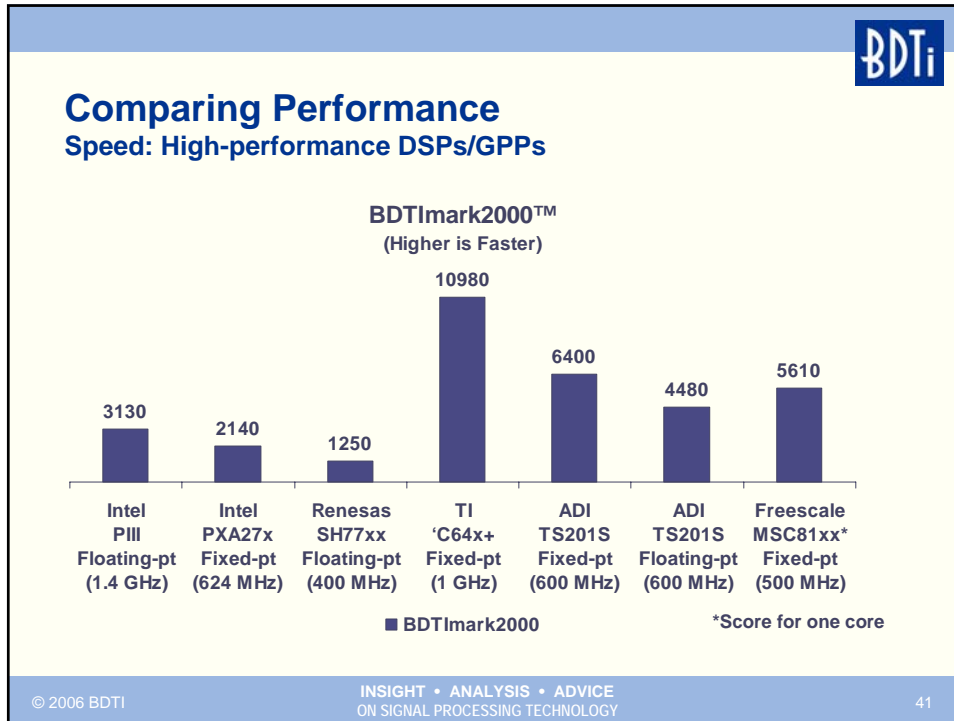
BDTImark2000™ and BDTIsimMark2000™  
(Higher is Faster)



Processor	BDTIsimMark2000	BDTImark2000
ARM ARM7 (145 MHz)	140	140
ARM ARM9E (265 MHz)	520	520
ADI BF53x (400 MHz)	2240	2240
TI 'C55x (300 MHz)	1460	1460
Freescale '56800E (120 MHz)	340	340
LSI LSI40x (200 MHz)	940	940
Microchip dsPIC3x (40 MHz)	130	130


© 2006 BDTI
INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY
40

# Microprocessors vs. DSPs



© 2006 Berkeley Design Technology, Inc.

# Microprocessors vs. DSPs




## Conclusions

### When to Use Which

<u>DSP</u> Heavy signal processing requirements Limited control processing	<u>GPP</u> Modest signal processing requirements Extensive control processing <ul style="list-style-type: none"><li>• Especially if code density and portability are important</li></ul>
The DSP is incumbent Software compatibility between generations not required—or can be achieved w/ DSP	The GPP is incumbent 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

© 2006 BDTI INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY 43



## 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

© 2006 BDTI INSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY 44

© 2006 Berkeley Design Technology, Inc.

## Microprocessors vs. DSPs



**For More Information...**  
**[www.BDTI.com](http://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

  
  
6<sup>th</sup> Edition

© 2006 BDTIINSIGHT • ANALYSIS • ADVICE  
ON SIGNAL PROCESSING TECHNOLOGY45