Independent DSP Benchmarks: Methodologies and Results

Berkeley Design Technology, Inc.
2107 Dwight Way, Second Floor
Berkeley, California U.S.A.

+1 (510) 665-1600
info@BDTI.com
http://www.BDTI.com

1

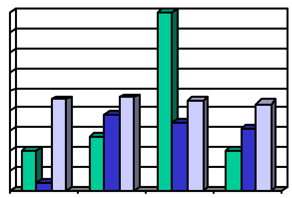
Outline

- Motivation for benchmarking
- DSP benchmarking approaches–pros and cons
- DSP benchmarks: what's available
- Benchmark performance of example processors
- The BDTImark™: what is it?
- Factors influencing benchmark results
- DSP benchmarking for general-purpose processors
- Conclusions

2

Motivation for Benchmarking

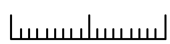
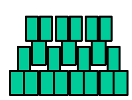
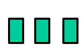
- Need quick and accurate comparisons of processors' DSP performance
- As architectures diversify, it becomes more difficult to compare performance
- There is a need for accurate comparisons of processors' DSP performance



 3

DSP Benchmarking Approaches

There are a number of DSP benchmarking approaches. The main candidates are:

- Simplified metrics (MIPS, MOPS, etc) 
- Complete DSP applications 
- DSP algorithm “kernels” 

 4

What's Wrong with MIPS?

Why not rely on MIPS, MOPS, MACs/sec, MFLOPS...?

These metrics are simple and easy to measure, but can be misleading. Questions to ponder:

- Just what is an “instruction” or “operation?”
  (or, when is 100 MIPS faster than 120 MIPS?)
- What's included in a MAC, and what if my application does something besides MACs?

Benchmarking Full Applications

Why not just use a full DSP application, like a V.90 modem or AC-3 decoder?

This approach is common in PC systems (e.g., SPEC) but is not appropriate for DSP benchmarking because:

- Applications tend to be ill-defined
- Hand-optimization usually required
  - Costly, time-consuming to implement
  - Evaluates programmer as much as processor
- Measures *system*, not just processor

What's an Algorithm Kernel?

- DSP algorithm kernels are the most computationally intensive portions of DSP applications.
- Example algorithm kernels include FFTs, IIR filters, Viterbi decoders, etc.

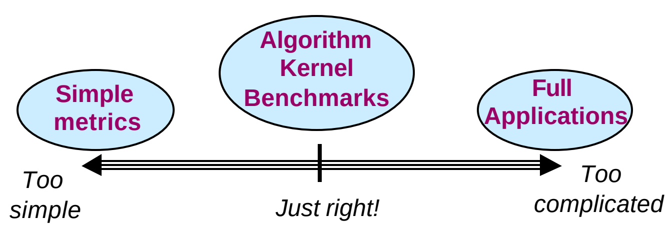
Application-relevant algorithm kernels are strong predictors of overall performance.

 7

Why Use Algorithm Kernels?

Algorithm kernels are good benchmark candidates because they are:

- Relevant
- Practical to specify and implement
- Relatively simple to optimize



 8

DSP Benchmark Landscape

- **Vendor benchmarks**
  - Most processor vendors provide DSP algorithm kernel benchmark results for their own processors
  - Benchmarks generally not standardized across vendors
  - Results not independently verified
- **EEMBC** (EDN Embedded Microprocessor Benchmark Consortium)
  - Consortium of semiconductor and IP vendors formed in 1998
  - Uses algorithm kernel benchmarks divided by application area (telecom, automotive, etc.)
  - Vendors implement benchmarks, EEMBC verifies results
  - Benchmarks implemented in C and optionally optimized assembly
  - Results publicly available

 9

DSP Benchmark Landscape

- **BDTI**
  - Independent DSP technology analysis and software development firm that developed proprietary set of DSP algorithm kernel benchmarks in 1994
  - Implements and/or verifies benchmarks in-house
  - Benchmarks implemented in optimized assembly following specification
  - Provides analysis of results; results and analyses available in published reports
  - Summary results published on web site

 10

BDTI Benchmarking Methodology

- Benchmarks are rigorously defined
- All implementations follow the same rules
- Benchmarks are hand-optimized in assembly
- Each benchmark is independently verified for:
  - Performance
  - Functionality
  - Optimality
  - Conformance to benchmark specs
- Benchmarks use processor's native data format

 11

BDTI Benchmarking Methodology

- Benchmarks are optimized for speed, then memory usage (except control-oriented benchmark, which is the other way around)
- BDTI's benchmarks reveal realistic performance, not necessarily fastest possible performance
- Benchmarks are architecture-independent; can be implemented on any processor (including general-purpose processors)

 12

BDTI Benchmark™ Suite

Composed of a wide variety of DSP algorithm kernels.
On each benchmark, we measure five quantities:

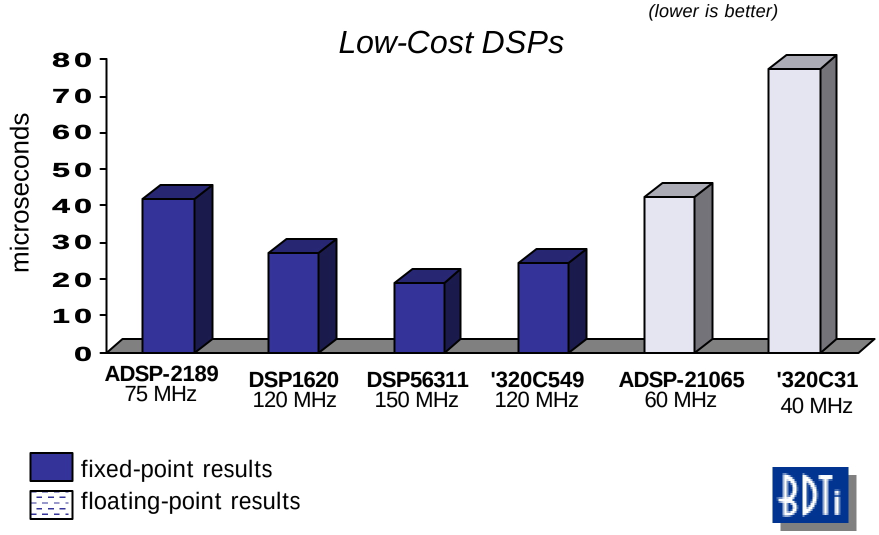
- Cycle count
- Execution time
- Cost-performance
- Energy consumption
- Memory use

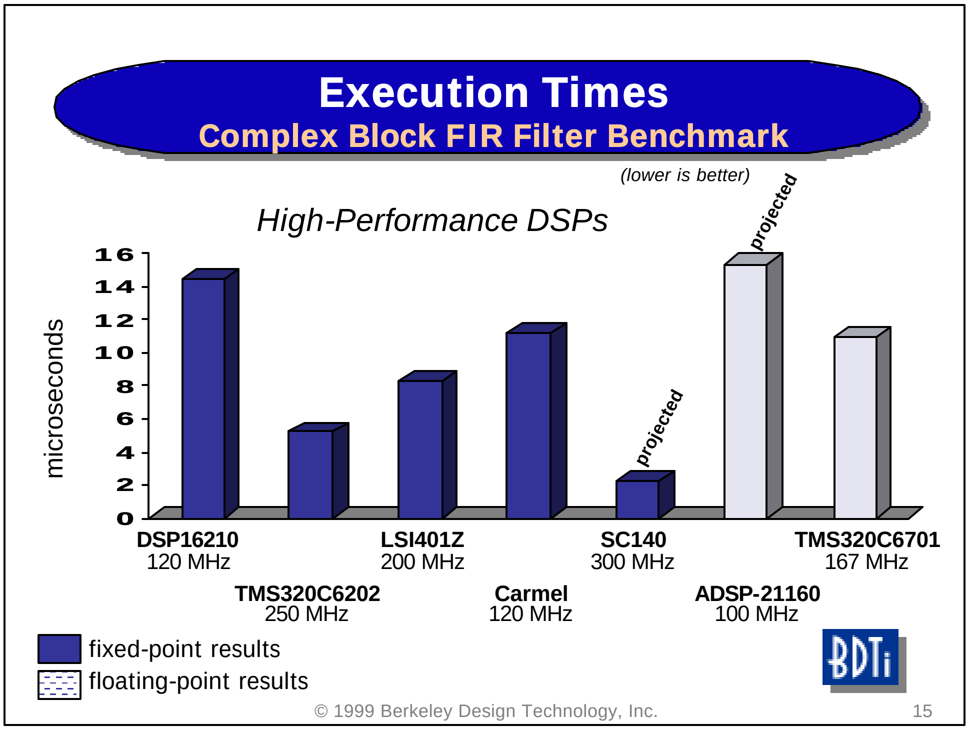
*Most benchmark results in this presentation are taken from BDTI's reports, *Buyer's Guide to DSP Processors 1999 Edition* and *DSP on General-Purpose Processors*

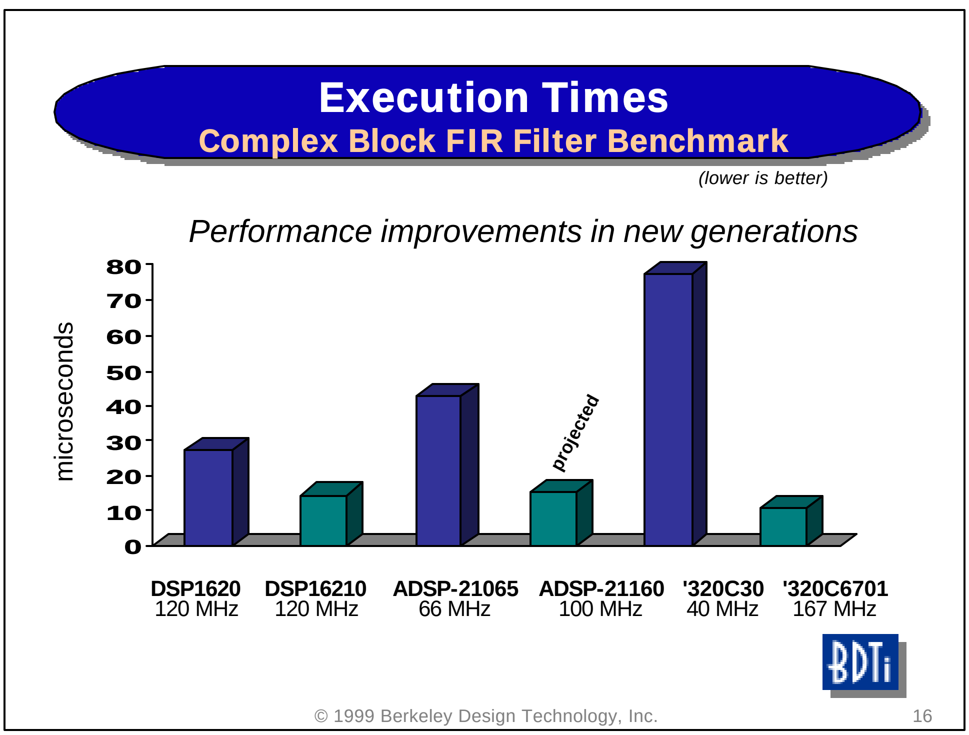
Execution Times

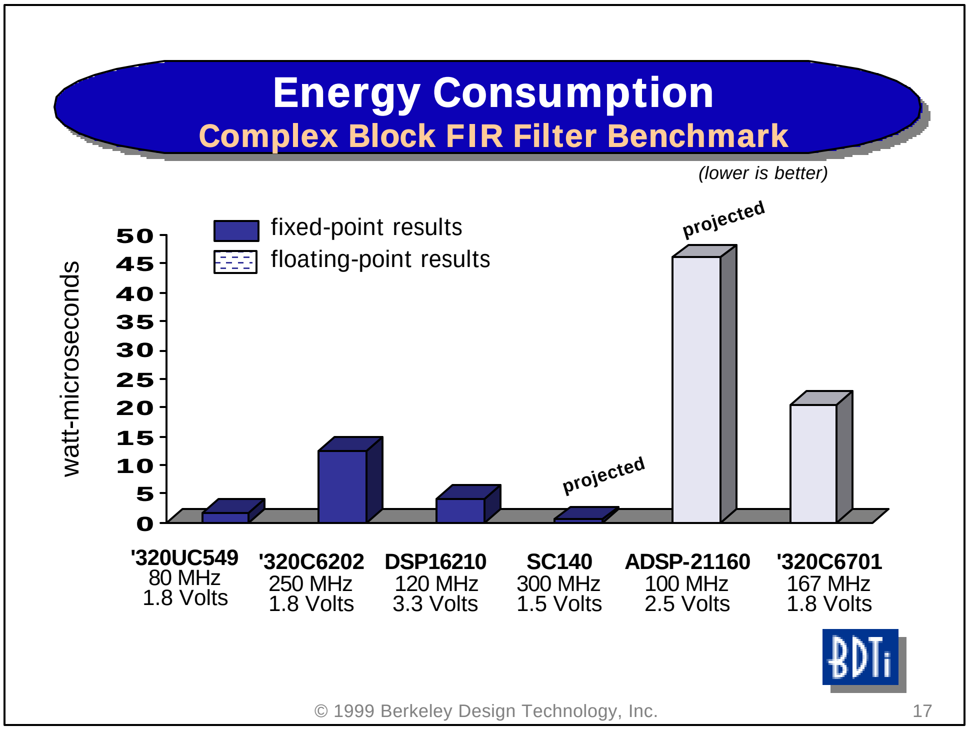
Complex Block FIR Filter Benchmark

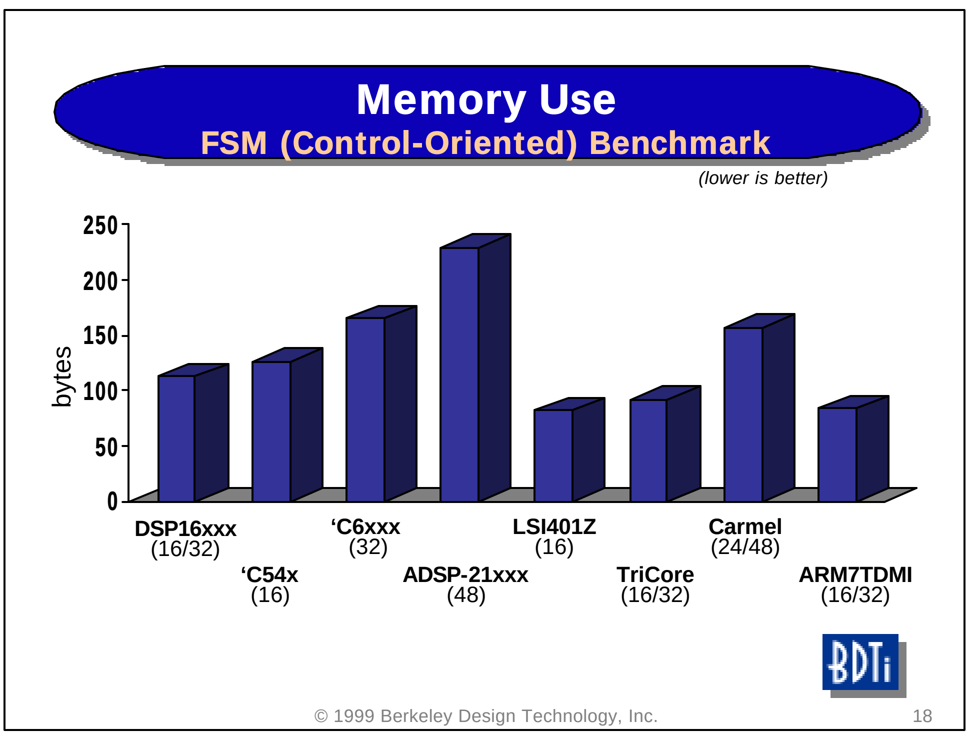
(lower is better)




Execution Times
Complex Block FIR Filter Benchmark
(lower is better)
High-Performance DSPs
microseconds
0
2
4
6
8
10
12
14
16
projected
projected
DSP16210
120 MHz
TMS320C6202
250 MHz
LSI401Z
200 MHz
Carmel
120 MHz
SC140
300 MHz
ADSP-21160
100 MHz
TMS320C6701
167 MHz
fixed-point results
floating-point results
BDTi
© 1999 Berkeley Design Technology, Inc.
15

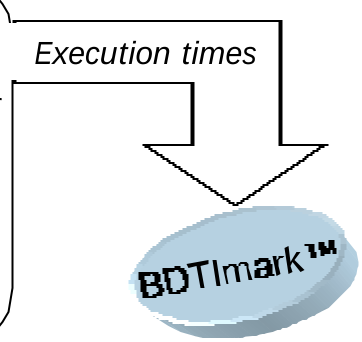

Execution Times
Complex Block FIR Filter Benchmark
(lower is better)
Performance improvements in new generations
microseconds
0
10
20
30
40
50
60
70
80
projected
DSP1620
120 MHz
DSP16210
120 MHz
ADSP-21065
66 MHz
ADSP-21160
100 MHz
'320C30
40 MHz
'320C6701
167 MHz
BDTi
© 1999 Berkeley Design Technology, Inc.
16


Energy Consumption
Complex Block FIR Filter Benchmark
(lower is better)
watt-microseconds
0
5
10
15
20
25
30
35
40
45
50
fixed-point results
floating-point results
projected
projected
'320UC549
80 MHz
1.8 Volts
'320C6202
250 MHz
1.8 Volts
DSP16210
120 MHz
3.3 Volts
SC140
300 MHz
1.5 Volts
ADSP-21160
100 MHz
2.5 Volts
'320C6701
167 MHz
1.8 Volts
© 1999 Berkeley Design Technology, Inc.
17


Memory Use
FSM (Control-Oriented) Benchmark
(lower is better)
bytes
0
50
100
150
200
250
DSP16xxx
(16/32)
'C54x
(16)
'C6xxx
(32)
ADSP-21xxx
(48)
LSI401Z
(16)
TriCore
(16/32)
Carmel
(24/48)
ARM7TDMI
(16/32)
© 1999 Berkeley Design Technology, Inc.
18

The BDTImark™

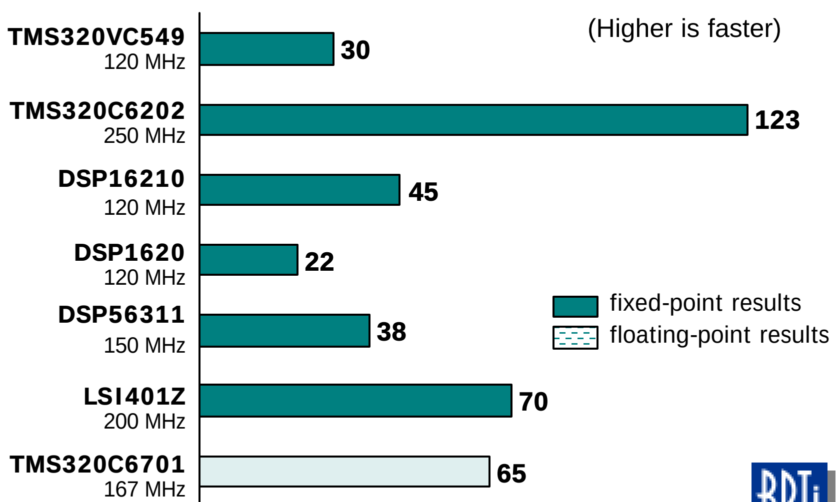
- Real block FIR filter
- Complex block FIR filter
- Single-sample real FIR filter
- Single-sample LMS-adaptive FIR filter
- Single-sample IIR filter
- Vector dot product
- Vector add
- Vector maximum
- IS-54 convolutional encoder
- Finite state machine
- 256-point FFT



Note: BDTI is currently updating its benchmark suite.

 19

Example BDTImark Results



 20

What Factors Influence Benchmark Results?

21

Factors Affecting Speed

- Clock rate
  - Pipeline
- More work per cycle
  - Parallel execution units
  - VLIW
  - Superscalar
  - SIMD capabilities
  - Hardware accelerators
  - RISC-like instructions vs complex, compound instructions
  - Memory bandwidth
  - Pipeline

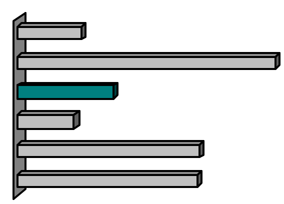
22

Case Study: The DSP16xxx

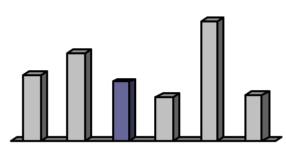
- Traditional DSP architecture, but with major additions
- Dual multipliers, wider memory buses yield 2 MACs/cycle
- Complex instructions, restrictions on parallel operations and register usage
- Simple pipeline

The DSP16210

- Good BDTImark score



- Moderate memory usage



- Good energy consumption

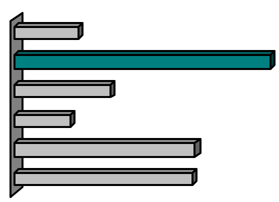
Case Study: The TMS320C62xx

- Radical new VLIW-like architecture
- Simple, RISC-like instructions with few restrictions
- 8 execution units (including 2 multipliers and 4 ALUs) produce 2 MACs/cycle
- Deep, complicated pipeline

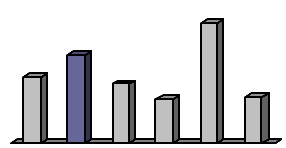
 25

The TMS320C6201

- Excellent BDTImark score



- High memory usage



- Moderate energy consumption

 26

GPPs for DSP

High-End GPPs for DSP

Today's high-end general-purpose processors outperform many DSPs *even on DSP applications.*

Why?

- Blazing clock speeds
- Superscalar execution
- Branch prediction, speculative execution
- Integrated DSP-oriented features

Drawbacks of High-End GPPs

Even when their performance is competitive, high-end GPPs don't usually replace DSPs because of

- Unpredictable execution times
- Poor cost-performance relative to fixed-point DSPs
- High energy consumption
- A lack of DSP-oriented development tools
- Integration difficulties

If a high-end GPP is incumbent, it may make sense to use it for DSP work. Otherwise, it's often better to use a DSP.

 29

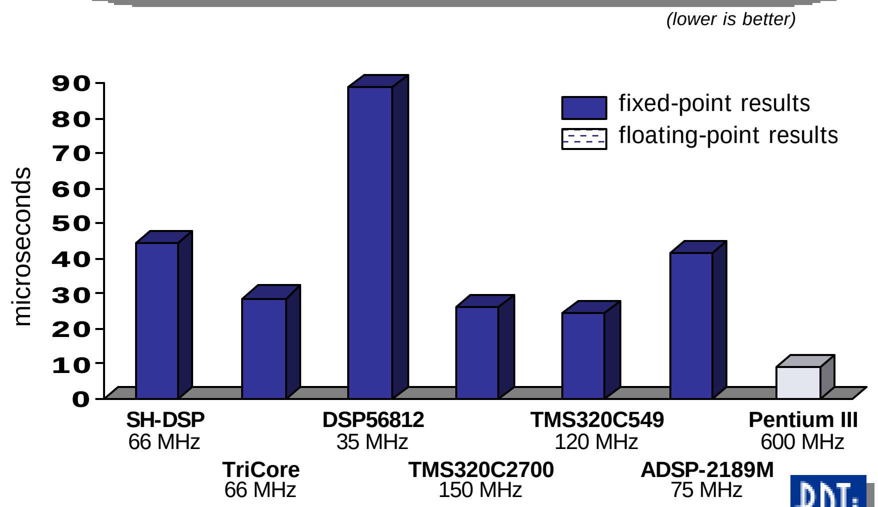
Embedded GPPs for DSP

- GPPs for embedded applications are starting to address DSP needs
  - Hitachi SH-DSP, ARM9E, Infineon TriCore
- These processors achieve reasonable DSP performance while maintaining relatively low cost and low energy consumption
- Embedded GPPs typically don't have the advanced features that affect execution time predictability, so are easier to use for DSP

 30

Execution Times

Complex Block FIR Filter Benchmark



 31

Conclusions

- Rigorous benchmark specs are essential
- The "best" processor depends on the application
- The fastest processor for a DSP task may not be a DSP
- Metrics other than execution speed may be most important
- Benchmarks don't tell the whole story

 32

Recent Developments

- New benchmarks
  - New FFT
  - Control – replaces FSM
  - Bit unpacking – replaces convolutional encoder
  - Viterbi decoder

 33

Work in Progress

- Work on new processors
  - TigerSHARC (Analog Devices)
  - Teak (DSP Group)
  - Palm (DSP Group)
  - Alpha 21264 (Compaq/Digital)
  - PowerPC G4 (Motorola)
  - SH-4 (Hitachi)

 34

For More Information...

Free resources on BDTI's web site,

http://www.BDTI.com

- *Evaluating DSP Processor Performance,* a white paper from BDTI
- *DSP Processors Hit the Mainstream* originally printed in IEEE Computer Magazine
- Numerous other BDTI article reprints, slides
- *comp.dsp* FAQ
- BDTImark scores

 35