

Architectures for DSP: The Options Multiply

Berkeley Design Technology, Inc.

www.BDTI.com



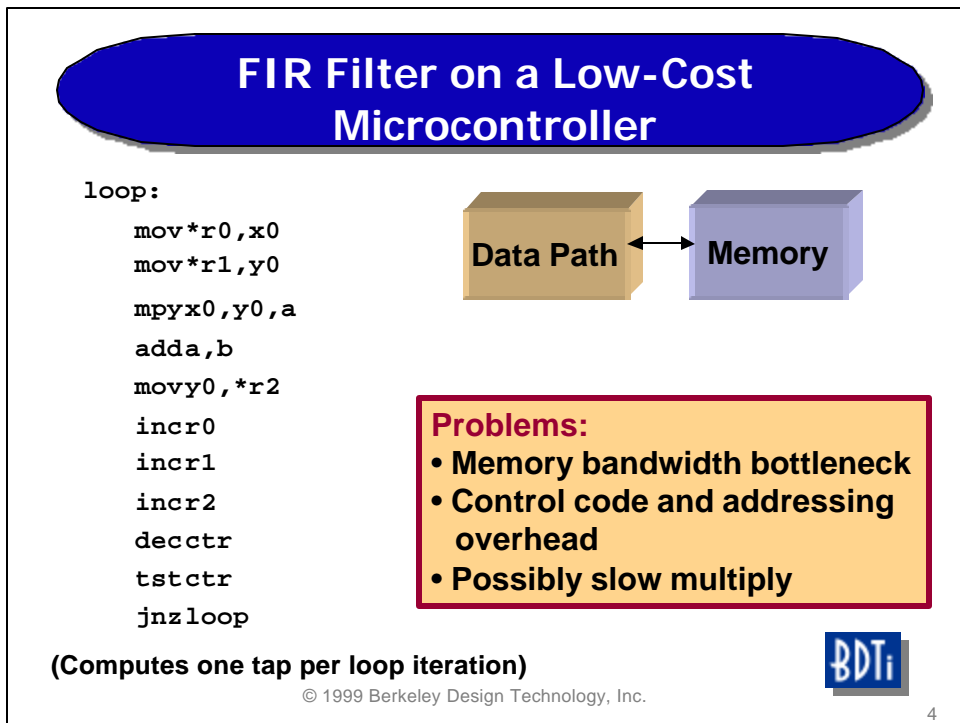
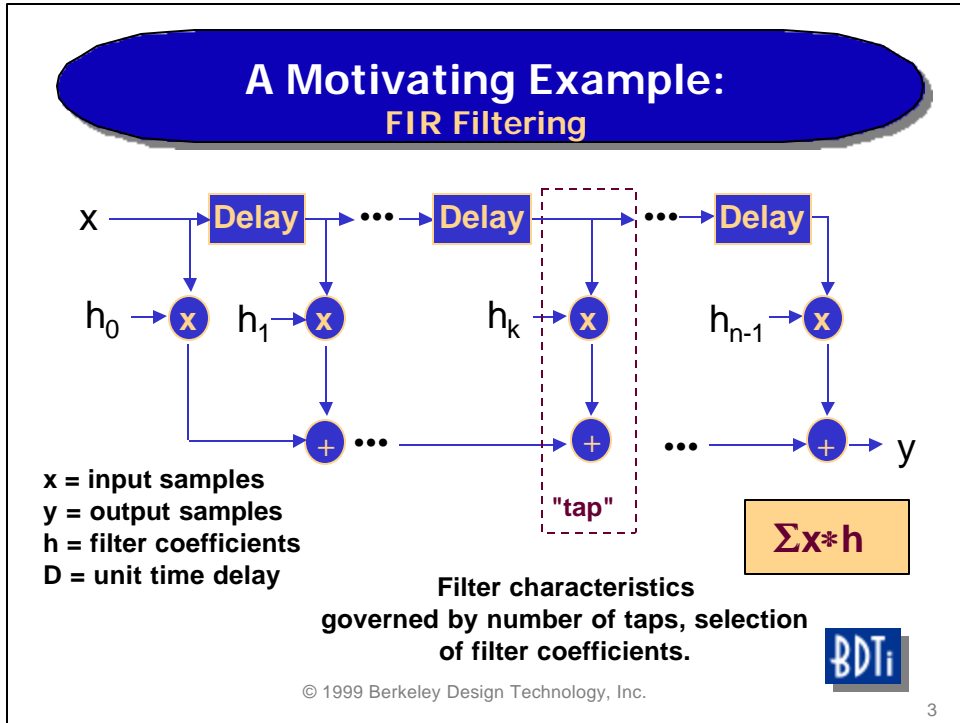
Copyright © 1999 Berkeley Design Technology, Inc.

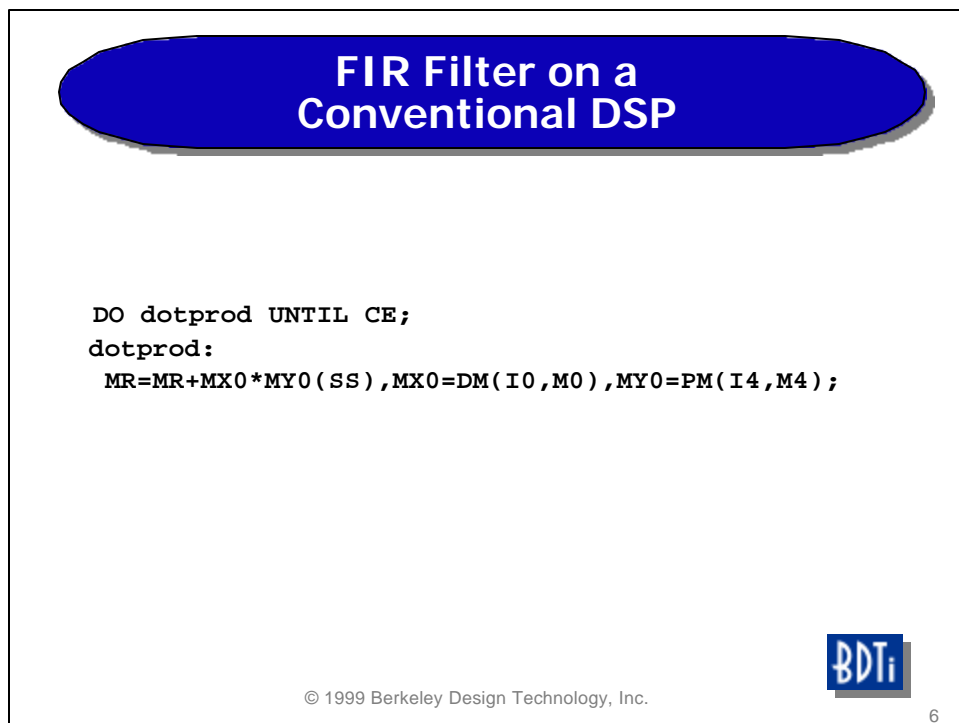
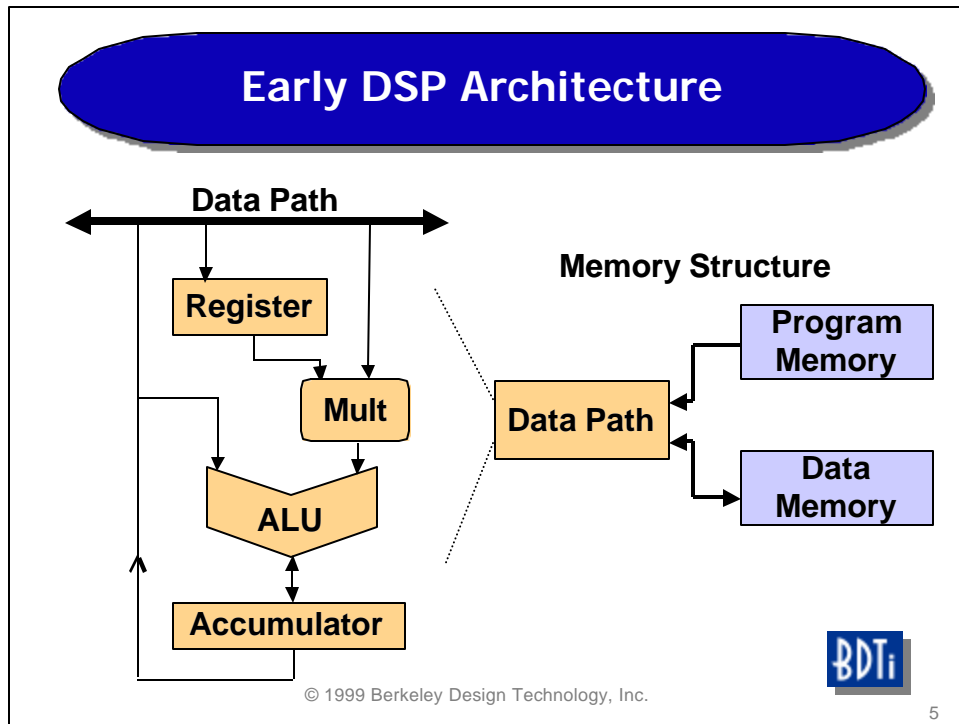
1

Outline

- ◆ DSP architectural basics
- ◆ Improved performance through increased parallelism
 - Allowing more operations per instruction
 - Enhanced conventional DSPs
 - Single-instruction, multiple-data (SIMD)
 - Issuing multiple instructions per instruction cycle
 - VLIW (very long instruction word) DSPs
 - Superscalar DSPs
- ◆ CPUs with SIMD extensions
- ◆ DSP/microcontroller hybrids







Baseline: "Conventional DSPs"

- ◆ Common attributes:
 - 16- or 24-bit fixed-point (fractional), or 32-bit floating-point arithmetic
 - 16-, 24-, or 32-bit instructions
 - One instruction per cycle ("single issue")
 - Complex, "compound" instructions encoding many operations
 - Highly constrained, non-orthogonal architectures



© 1999 Berkeley Design Technology, Inc.

7

Baseline: "Conventional DSPs"

- ◆ Common attributes (cont.):
 - Dedicated addressing hardware w/ specialized addressing modes
 - Multiple-access on-chip memory architecture
 - Dedicated hardware for loops and other execution control
 - Specialized on-chip peripherals and I/O interfaces
 - Low cost, low power, low memory usage



© 1999 Berkeley Design Technology, Inc.

8

Increasing Parallelism

- ◆ Boosting performance beyond the increases afforded by faster clock speeds requires the processor to do more work in every clock cycle. How?
- ◆ By increasing the processors' parallelism in one of the following ways:
 1. Increase the number of operations that can be performed in each instruction
 2. Increase the number of instructions that can be issued and executed in every cycle



© 1999 Berkeley Design Technology, Inc.

9

1. More Operations Per Instruction

- ◆ How to increase the number of operations that can be performed in each instruction?
 - Add execution units (multiplier, adder, etc.)
 - Enhance the instruction set to take advantage of the additional hardware
 - Possibly, increase the instruction word width
 - Use wider buses to keep the processor fed with data
 - Add SIMD (single instruction, multiple data) capabilities



© 1999 Berkeley Design Technology, Inc.

10

2. More Instructions Per Clock Cycle

- ◆ How to increase the number of instructions that are issued and executed in every clock cycle?
 - Use **VLIW** techniques
 - Use **superscalar** techniques
- ◆ VLIW and superscalar architectures typically use simple, RISC-based instructions
 - More orthogonal than the complex, compound instructions traditionally used in DSP processors

© 1999 Berkeley Design Technology, Inc.



11

New Architectures for DSP

- ◆ **Enhanced conventional DSPs**
 - Examples: Lucent DSP16xxx, ADI ADSP-2116x
- ◆ **VLIW (Very Long Instruction Word) DSPs**
 - Examples: TI TMS320C6xxx, Infineon Carmel
- ◆ **Superscalar DSPs**
 - Example: LSI Logic LSI401Z
- ◆ **General-purpose processors, hybrids:**
 - Examples: PowerPC with AltiVec, TriCore

© 1999 Berkeley Design Technology, Inc.



12

Enhanced Conventional DSPs

More parallelism via:

- ◆ Multi-operation data path
 - e.g., 2nd multiplier, adder
 - SIMD capabilities (ranging from limited to extensive)
- ◆ Highly specialized hardware in core
 - e.g., application-oriented data path operations
- ◆ Co-processors
 - Viterbi decoding, FIR filtering, etc.

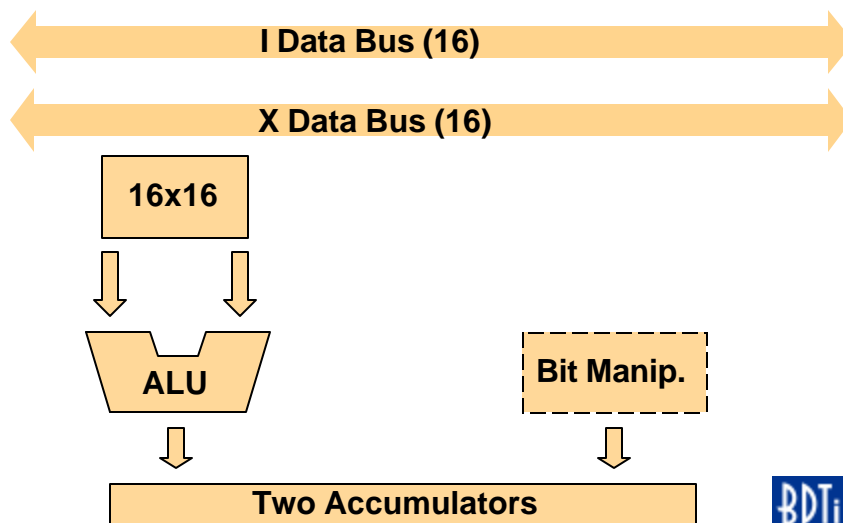
Example: Lucent DSP16xxx, ADI ADSP-2116x

© 1999 Berkeley Design Technology, Inc.



13

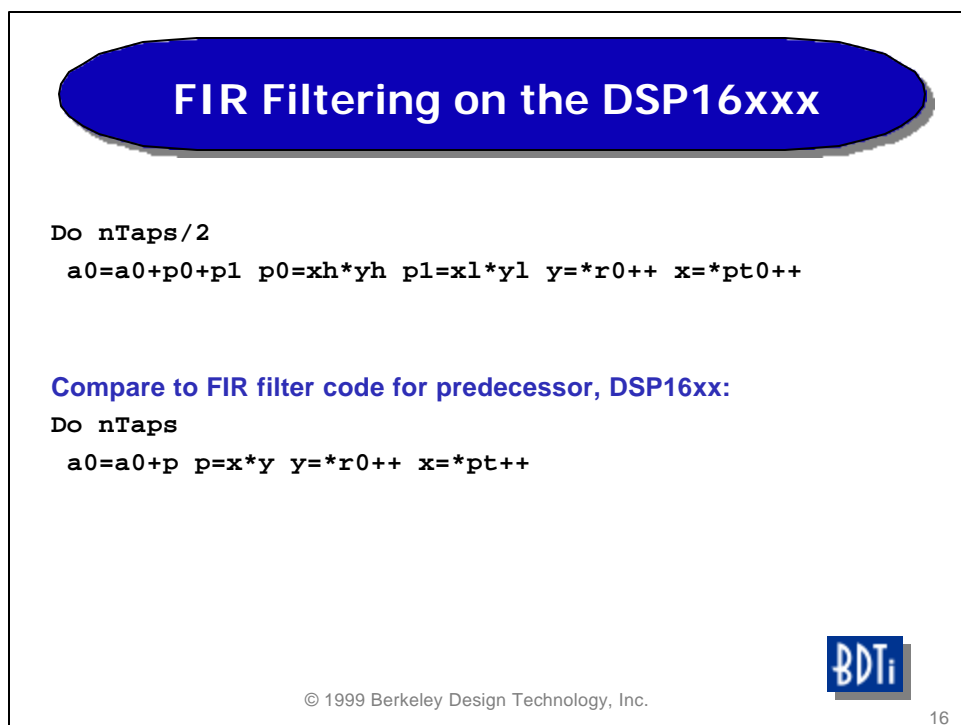
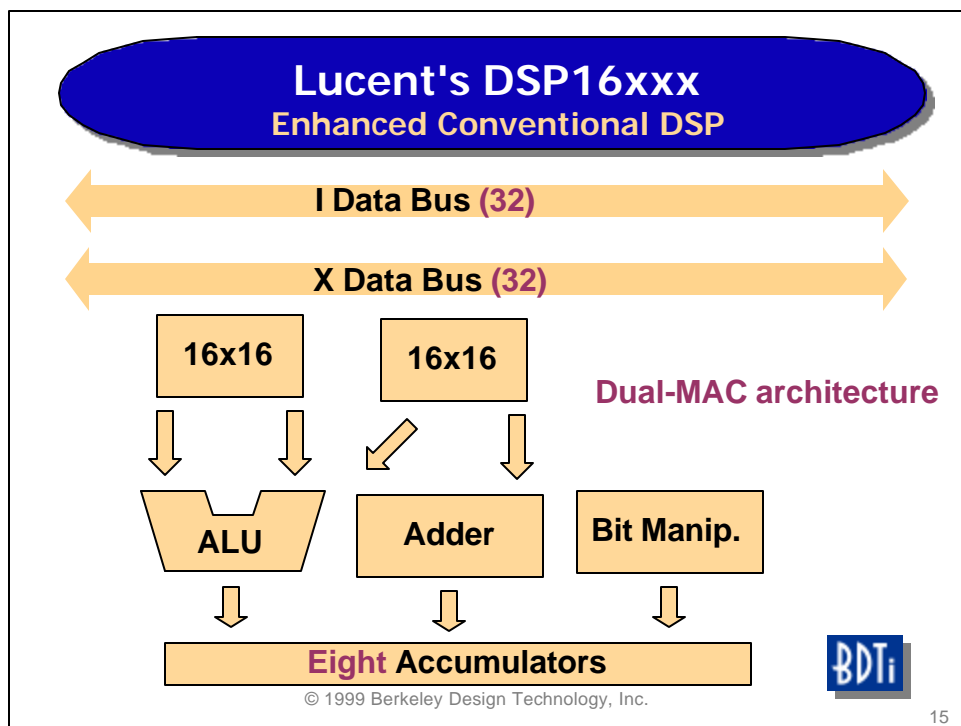
Lucent's DSP16xx Conventional DSP



© 1999 Berkeley Design Technology, Inc.

14

Architectures for DSP: The Options Multiply



Enhanced Conventional DSPs

- ◆ Advantages:
 - Allows incremental performance increases while maintaining competitive cost, power, code density
 - Compatibility is possible; similarity is likely
- ◆ Disadvantages:
 - Increasingly complex, hard-to-program architectures
 - Poor compiler targets
 - How much farther can we get with this approach?

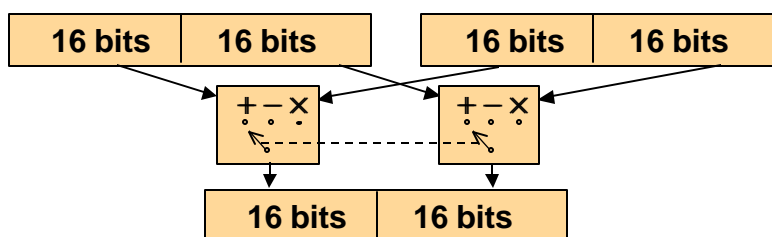


© 1999 Berkeley Design Technology, Inc.

17

SIMD

Single Instruction, Multiple Data



- ◆ Splits words into smaller chunks for parallel operations
- ◆ Some SIMD processors support multiple data widths (16-bit, 8-bit, ..)
- ◆ Examples: Lucent DSP16xxx, ADI ADSP-2116x, ADI TigerSHARC



© 1999 Berkeley Design Technology, Inc.

18

SIMD Extensions

- ◆ SIMD is becoming more and more common in DSP processors
 - Limited SIMD capabilities on the DSP16xxx
 - Full SIMD capabilities (enabled by dual data paths) on ADI's ADSP-2116x
 - "Hierarchical" SIMD capabilities on TigerSHARC



© 1999 Berkeley Design Technology, Inc.

19

SIMD Extensions

- ◆ SIMD extensions for CPUs are also common. Why?
 - Make good use of existing wide resources
 - Buses, data path
 - Significantly accelerate many DSP/image/video algorithms without a radical architectural change
 - Examples include Pentium with MMX/SSE, PowerPC G4 with AltiVec, ...



© 1999 Berkeley Design Technology, Inc.

20

SIMD Challenges

- ◆ Algorithms, data organization must be amenable to data-parallel processing
 - Programmers must be creative, and sometimes pursue alternative algorithms
 - Reorganization penalties can be significant
 - Most effective on algorithms that process large blocks of data—not very useful on single-sample algorithms

© 1999 Berkeley Design Technology, Inc.



21

SIMD Challenges

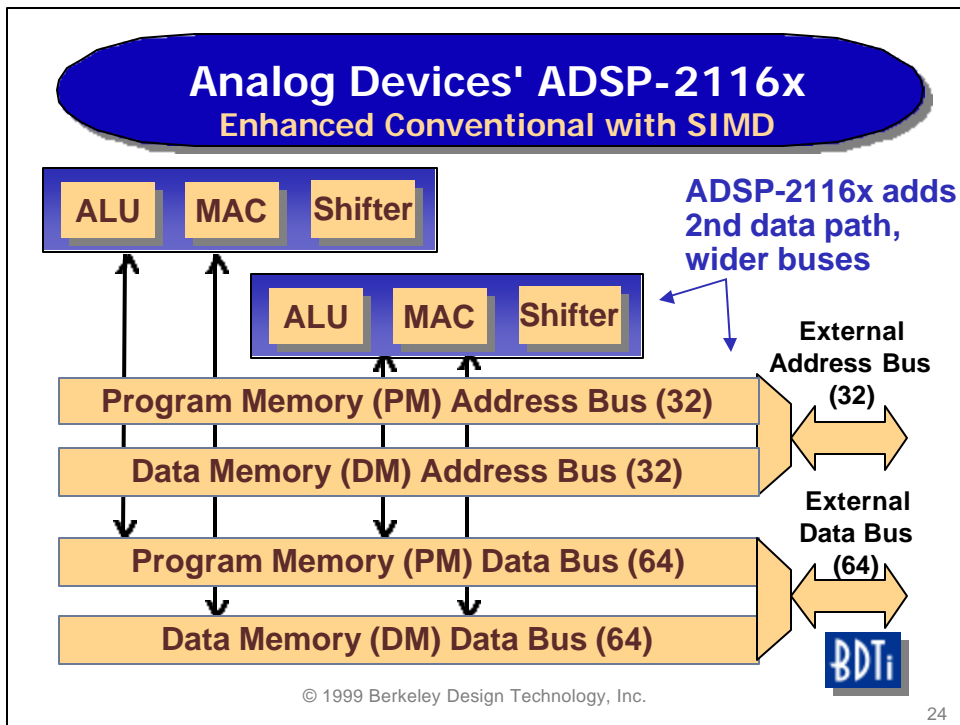
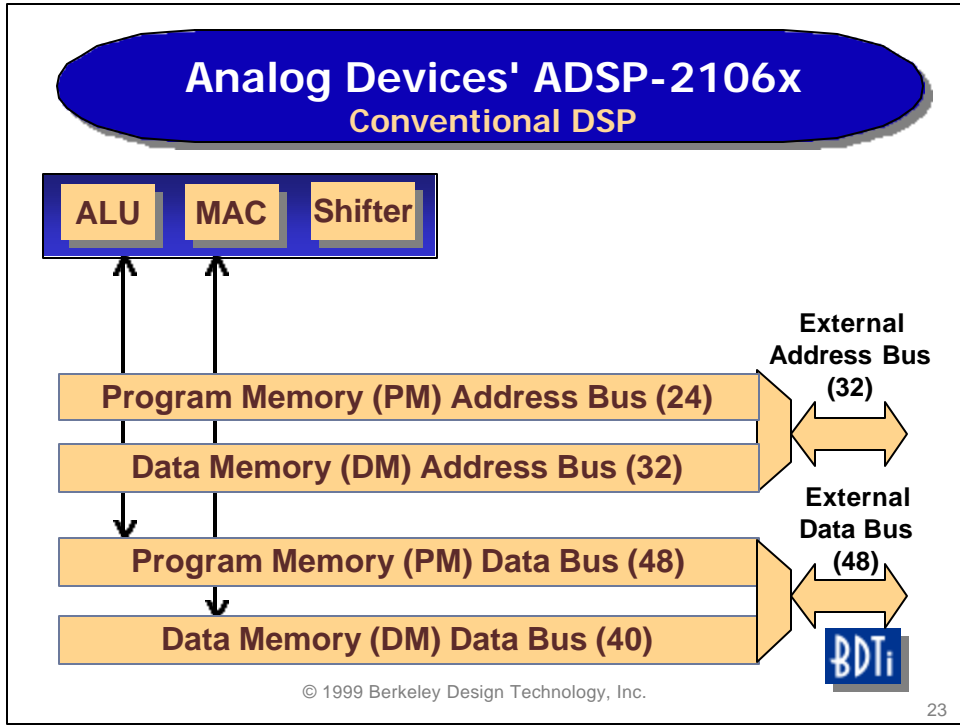
- ◆ Loss of generality
 - Each instruction processes N elements (typically $4 \leq N \leq 8$)
 - Loops often must be unrolled for speed
- ◆ High program memory usage
 - Loop unrolling
 - Re-arranging data for SIMD processing
 - Merging partial results
- ◆ Often, only fixed-point supported

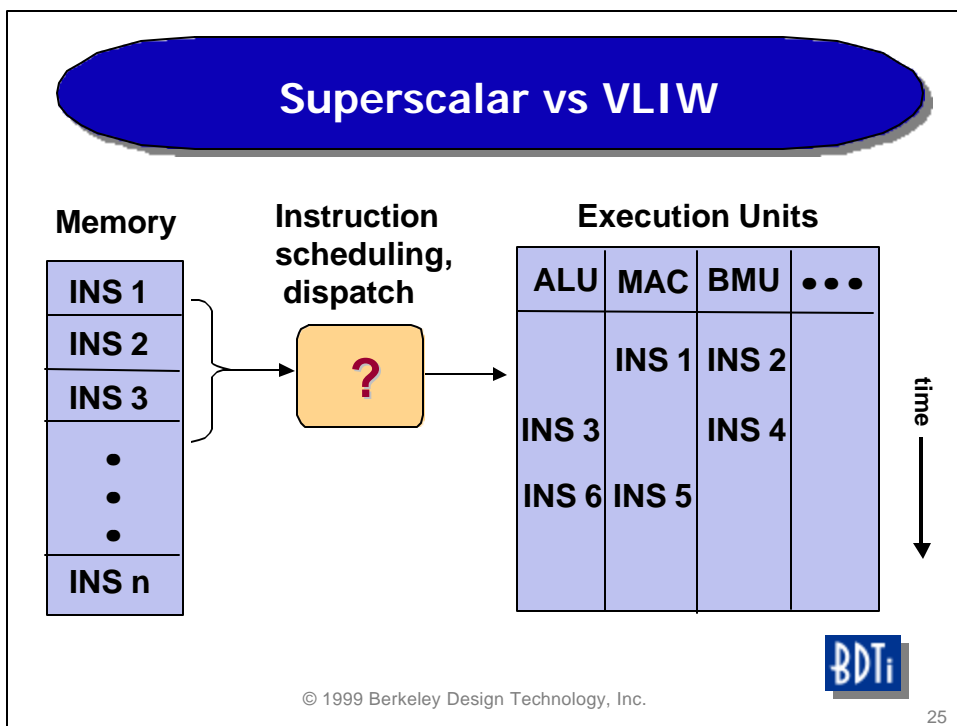
© 1999 Berkeley Design Technology, Inc.



22

Architectures for DSP: The Options Multiply





VLIW (Very Long Instruction Word)

Examples of current & upcoming VLIW-based architectures for DSP applications:

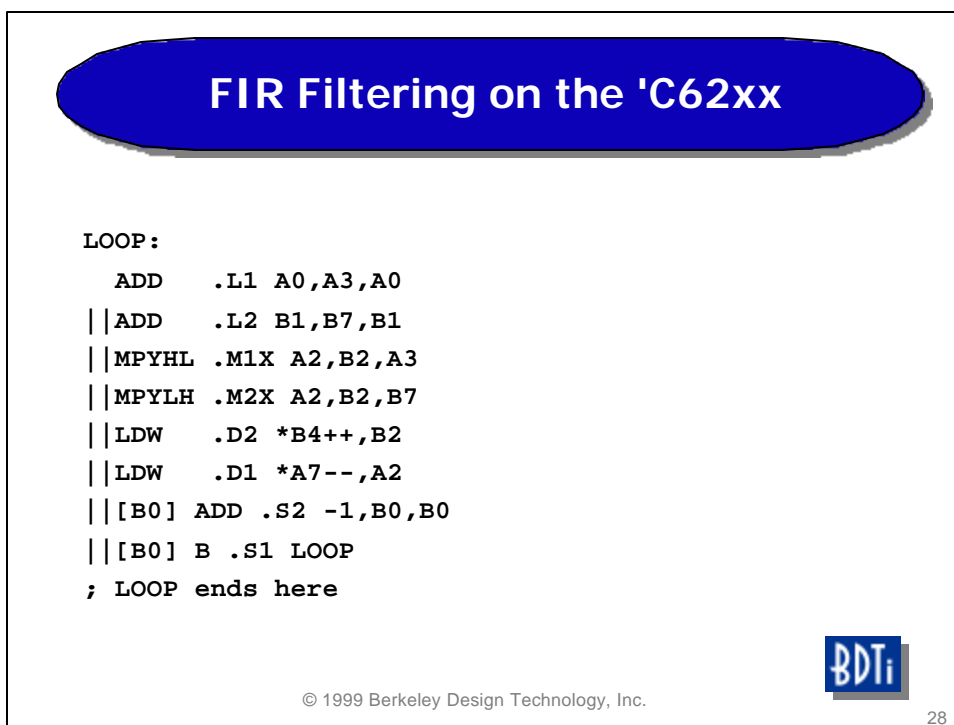
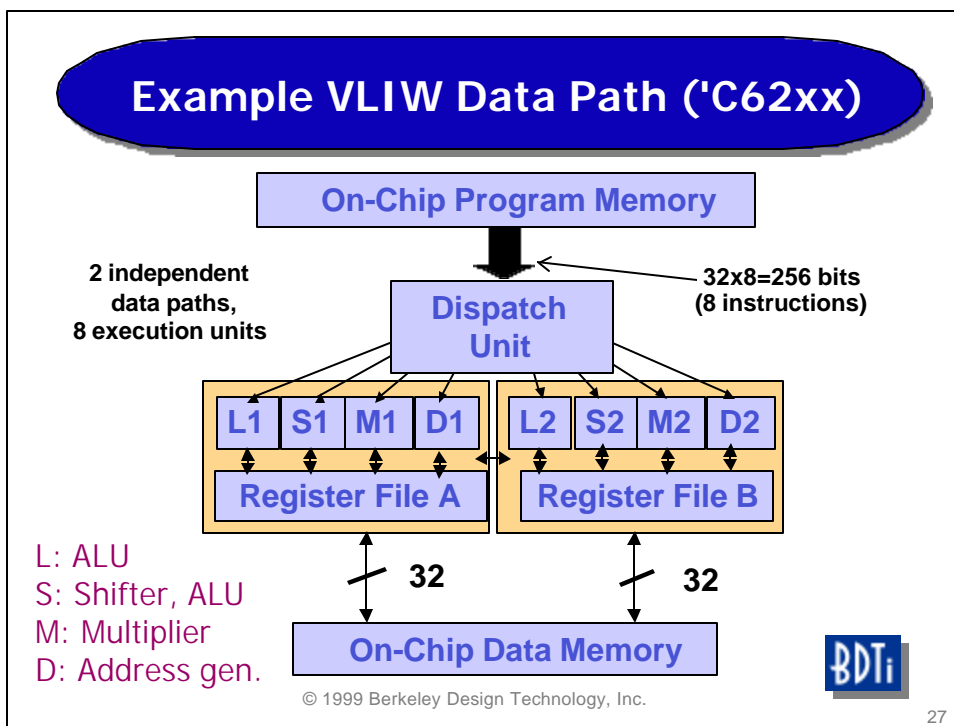
- TI TMS320C6xxx, Infineon Carmel, ADI TigerSHARC, StarCore SC140

Characteristics:

- Multiple independent instructions per cycle, packed into single large "super-instruction" or "packet"
- More regular, orthogonal, RISC-like operations
- Large, uniform register sets

© 1999 Berkeley Design Technology, Inc.

26



VLIW Architectures

◆ Advantages:

- Increased performance
- More regular architectures
 - Potentially easier to program, better compiler targets
- Scalable (?)

© 1999 Berkeley Design Technology, Inc.



29

VLIW Architectures

◆ Disadvantages:

- New kinds of programmer/compiler complexity
 - Programmer (or code-generation tool) must keep track of instruction scheduling
 - Some VLIW processors have deep pipelines and long latencies--can be confusing, may make peak performance elusive
- Code size bloat
 - High program memory bandwidth requirements
- High power consumption

© 1999 Berkeley Design Technology, Inc.



30

Superscalar Architectures

Current superscalar architectures for DSP apps:

- LSI Logic LSI401Z

Characteristics:

- Borrow techniques from high-end CPUs
- Multiple (usually 2-4) instructions issued per instruction cycle
 - Instruction scheduling handled in hardware, not by programmer/tools
- RISC-like instruction set
- Lots of parallelism



© 1999 Berkeley Design Technology, Inc.

31

Superscalar Architectures

◆ Advantages:

- Large jump in performance
- More regular architectures (potentially easier to program, better compiler targets)
- Programmer (or code generation tool) isn't required to schedule instructions
 - But peak performance may be hard to achieve without hand-tweaking
 - Code size not increased significantly



© 1999 Berkeley Design Technology, Inc.

32

Superscalar Architectures

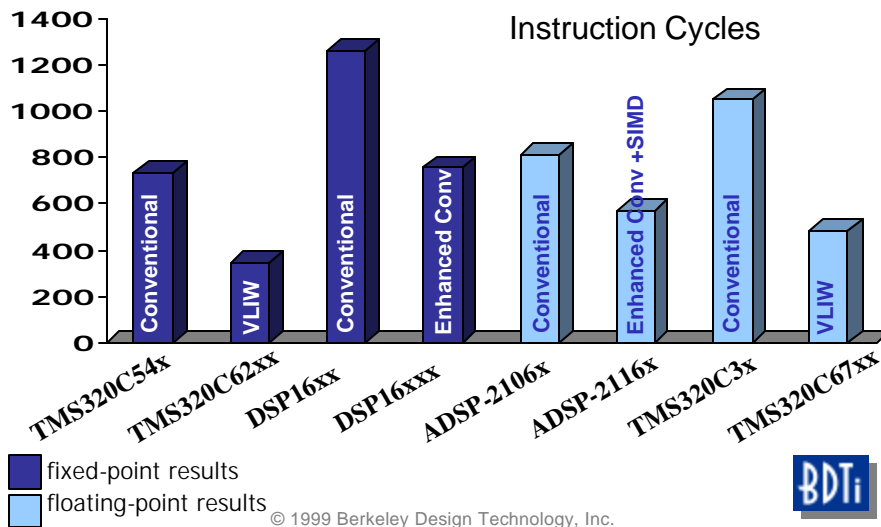
- ◆ Disadvantages:
 - Energy consumption is a major challenge
 - Dynamic behavior complicates software development
 - Execution-time variability can be a hazard—how to guarantee meeting real-time constraints?
 - Coding for worst-case behavior may leave a lot of a processor's performance untapped
 - Code optimization is challenging
 - If you can't tell how long a segment takes to execute, how can you tell if an optimization improved the performance?



© 1999 Berkeley Design Technology, Inc.

33

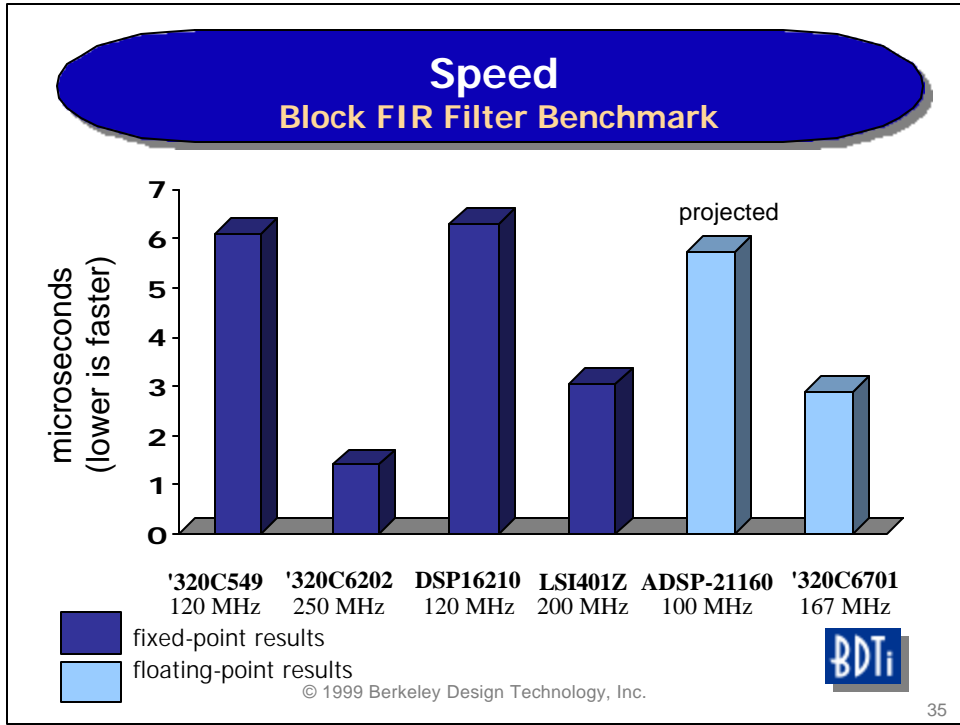
Comparing Architectural Efficiency Block FIR Filter Benchmark



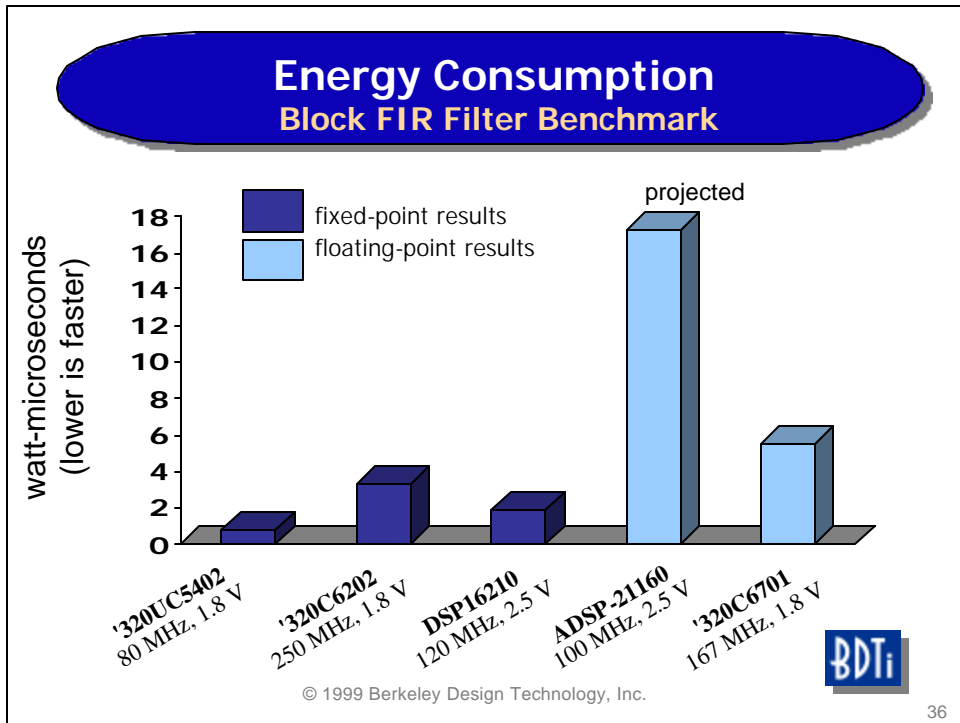
© 1999 Berkeley Design Technology, Inc.

34

Architectures for DSP: The Options Multiply

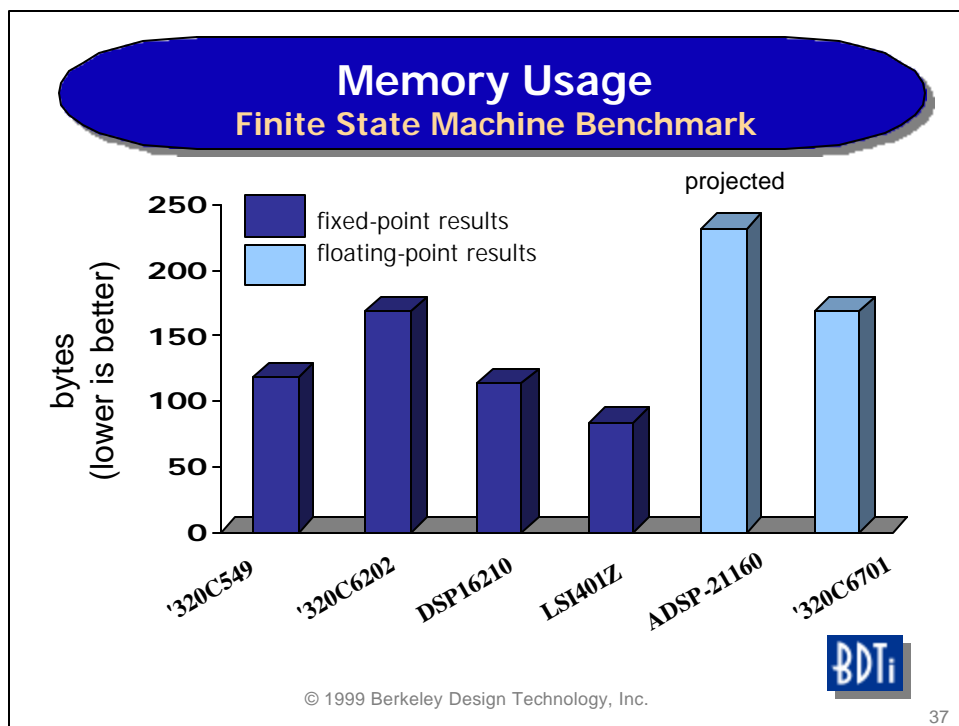


35



36

Architectures for DSP: The Options Multiply



37

High-Performance GPPs with SIMD

- ◆ Most high-performance GPPs targeting desktop applications are superscalar architectures
 - Pentium, PowerPC
- ◆ Often have many dynamic features to accelerate performance
 - Sophisticated, multi-level caches
 - Branch prediction
 - Speculative execution
- ◆ Most offer SIMD extensions to increase performance on DSP and multimedia applications (audio, video)
 - MMX/SSE, AltiVec

© 1999 Berkeley Design Technology, Inc. 

38

High-Performance GPPs with SIMD

- ◆ These processors can often execute DSP tasks faster than DSP processors
- ◆ So why do people still use DSPs?
 - Price
 - Power consumption
 - Availability of off-the-shelf DSP software, DSP-oriented development tools
 - DSP-oriented on-chip integration
 - Execution-time predictability is especially problematic with high-performance GPPs

© 1999 Berkeley Design Technology, Inc.



39

An Illustrative Example Execution-Time Predictability

Vector add 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?

© 1999 Berkeley Design Technology, Inc.



40

Hybrid DSP/Microcontrollers

- ◆ GPPs for embedded applications are starting to address DSP needs
- ◆ Embedded GPPs typically don't have the advanced features that affect execution-time predictability, so are easier to use for DSP
- ◆ There are a wide variety of approaches to combining DSP and microcontroller functionality

© 1999 Berkeley Design Technology, Inc.



41

Hybrid DSP/Microcontrollers Approaches

- Multiple processors on a die
 - e.g., Motorola DSP5665x
- DSP co-processor
 - e.g., Massana FILU-200, ARM Piccolo
- DSP brain transplant in existing μ C
 - e.g., SH-DSP
- Microcontroller tweaks to existing DSP
 - e.g., TMS320C27xx
- Totally new design
 - e.g., TriCore

© 1999 Berkeley Design Technology, Inc.



42

Hybrid DSP/Microcontrollers

Advantages, Disadvantages

- Multiple processors on a die
 - Two entirely different instruction sets, debugging tools, etc.
 - Both cores can operate in parallel
 - No resource contention...
 - ..but probably resource duplication

© 1999 Berkeley Design Technology, Inc.



43

Hybrid DSP/Microcontrollers

Advantages, Disadvantages

- DSP co-processor
 - May result in complicated programming model
 - May involve two instruction sets
 - Possible deadlocks
 - Possible resource contention
 - Substantial overhead may be consumed in transferring data between co-processor and host
 - May allow both cores to operate in parallel

© 1999 Berkeley Design Technology, Inc.



44

Hybrid DSP/Microcontrollers Advantages, Disadvantages

- DSP brain transplant in existing μC , microcontroller tweaks to existing DSP
 - Simpler programming model than dual cores
 - Constraints imposed by "legacy" architecture
- Totally new design
 - Avoids legacy constraints
 - May result in a cleaner architecture
 - Adopting a totally new architecture can be risky

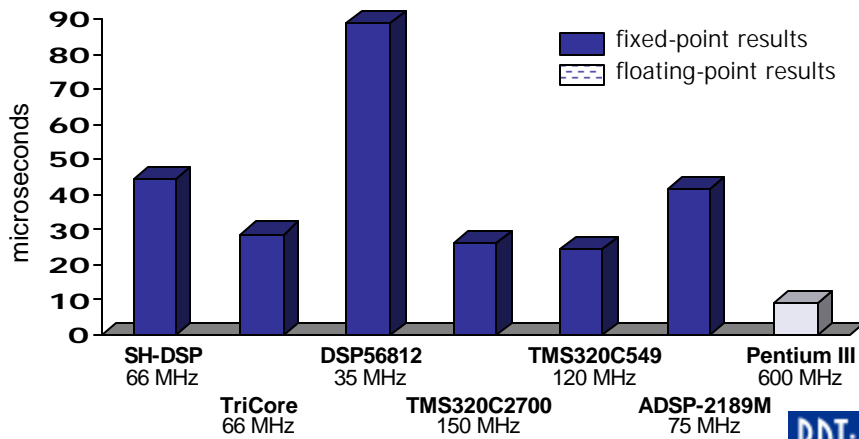


© 1999 Berkeley Design Technology, Inc.

45

Execution Times Complex Block FIR Filter Benchmark

(lower is better)



© 1999 Berkeley Design Technology, Inc.



46

Conclusions

- ◆ The variety, performance range of processors for DSP is exploding
 - Better selection, flexibility,...
 - ...but harder to choose the "best" processor
 - Architectures are interesting, but other factors may be more important

© 1999 Berkeley Design Technology, Inc.



47

For More Information...

Free resources on BDTI's web site,

www.BDTI.com

- *DSP Processors Hit the Mainstream* covers DSP architectural basics and new developments. Originally printed in *IEEE Computer Magazine*.
- *Evaluating DSP Processor Performance*, a white paper from BDTI.
- Numerous other BDTI article reprints, slides
- *comp.dsp* FAQ

© 1999 Berkeley Design Technology, Inc.



48

Join BDTI – We're Hiring!

BDTI is currently recruiting for the following positions:

- DSP Software Engineers *Both full-time and part-time*
- DSP Engineers/Analysts *positions are available.*

Why work for BDTI?

- ◆ Cool new technology: You will get to work with the hottest new processors and tools, often before they are publicly announced
- ◆ Diverse assignments: You will have the opportunity to use DSP technology for a wide range of applications, from audio to communications to ...
- ◆ High-caliber colleagues: We have a reputation for employing some of the sharpest engineers in the DSP industry



© 1999 Berkeley Design Technology, Inc.

49