Understanding the New DSP Processor Architectures

Berkeley Design Technology, Inc.
+1 (510) 665-1600
info@BDTI.com

www.BDTI.com

1

Outline

- Baseline: conventional DSP processors
- 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

2

Baseline: "Conventional DSPs"

- Introduced in early 1980's, still volume leader today
- 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

3

Baseline: "Conventional DSPs"

- Common attributes (cont.):
  - Highly constrained, non-orthogonal architectures
  - 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

4

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 instruction cycle

5

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

6

2. More Instructions Per Clock Cycle

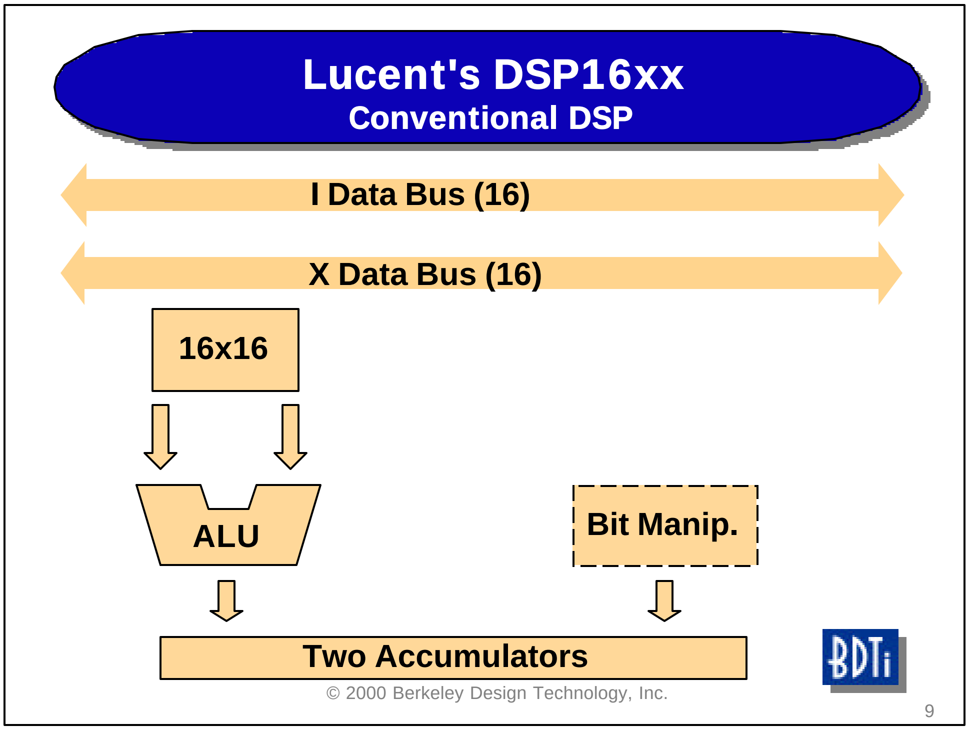
- How to increase the number of instructions that are issued and executed in every clock cycle?
  - Use VLIW techniques
  - Use superscalar techniques
- VLIW and superscalar architectures typically use simple, RISC-based instructions
  - More orthogonal than the complex, compound instructions traditionally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

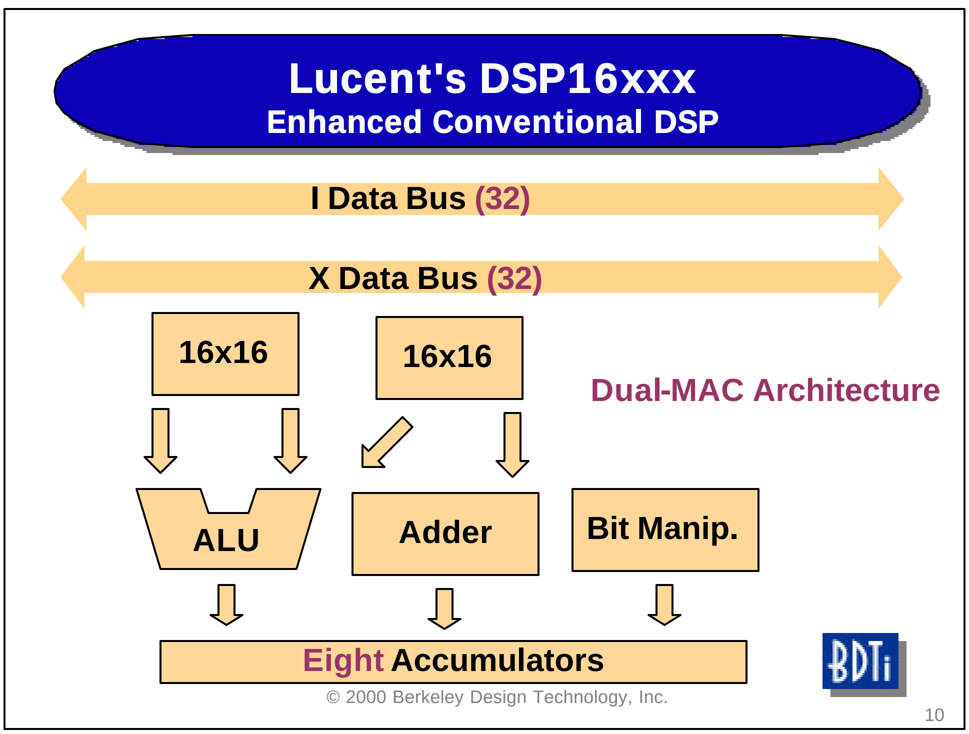
Enhanced Conventional DSPs

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


Lucent's DSP16xx
Conventional DSP
I Data Bus (16)
X Data Bus (16)
16x16
ALU
Bit Manip.
Two Accumulators
© 2000 Berkeley Design Technology, Inc.
9


Lucent's DSP16xxx
Enhanced Conventional DSP
I Data Bus (32)
X Data Bus (32)
16x16
16x16
Dual-MAC Architecture
ALU
Adder
Bit Manip.
Eight Accumulators
© 2000 Berkeley Design Technology, Inc.
10

FIR Filtering on the DSP16xxx

FIR filter inner loop code for predecessor, DSP16xx:

```
Do nTaps
 a0=a0+p p=x*y y=*r0++ x=*pt++
```

Compare to FIR filter inner loop code for DSP16xxx:

```
Do nTaps/2
 a0=a0+p0+p1 p0=xh*yh p1=xl*yl y=*r0++ x=*pt0++
```


11

Enhanced Conventional DSPs

- Advantages:
  - Allows significant 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?

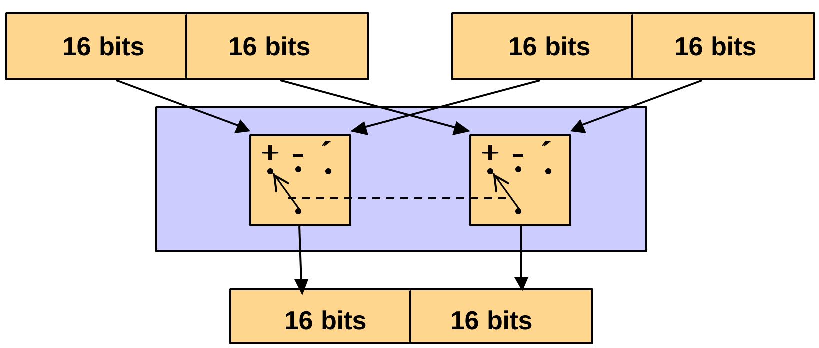
12

SIMD
Single Instruction, Multiple Data

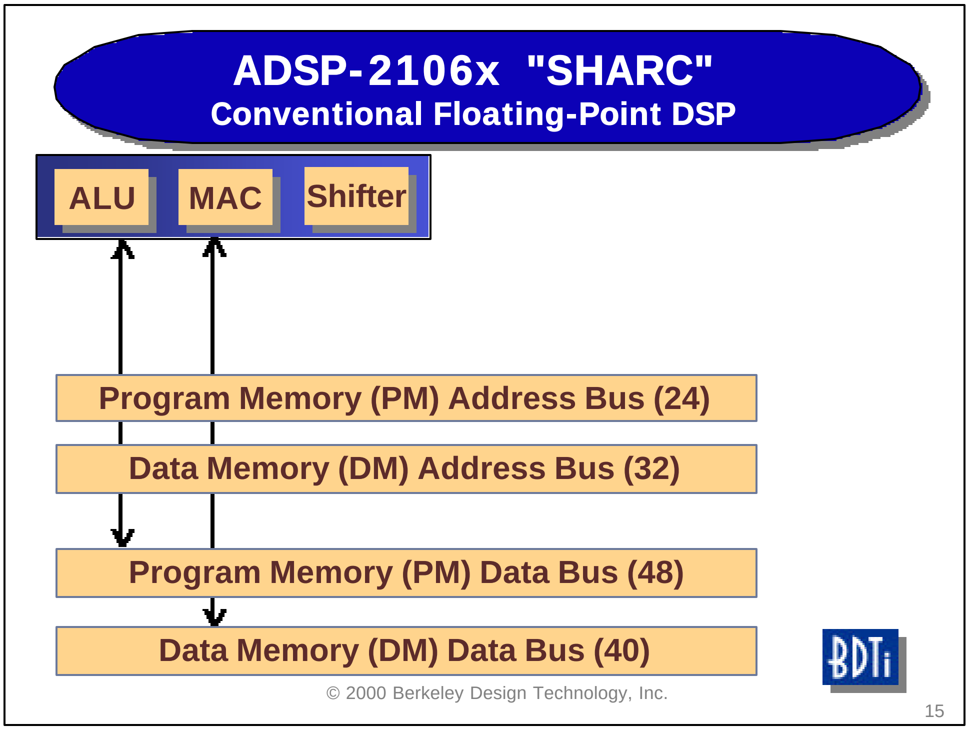
- One instruction performs the same operation on multiple (independent) sets of data
  - For each SIMD instruction, you can get 2x (or 4x, or 8x, …) the work
- Two ways to implement SIMD
  - Split execution units
  - Multiple execution units (or data paths) operating in lock-step

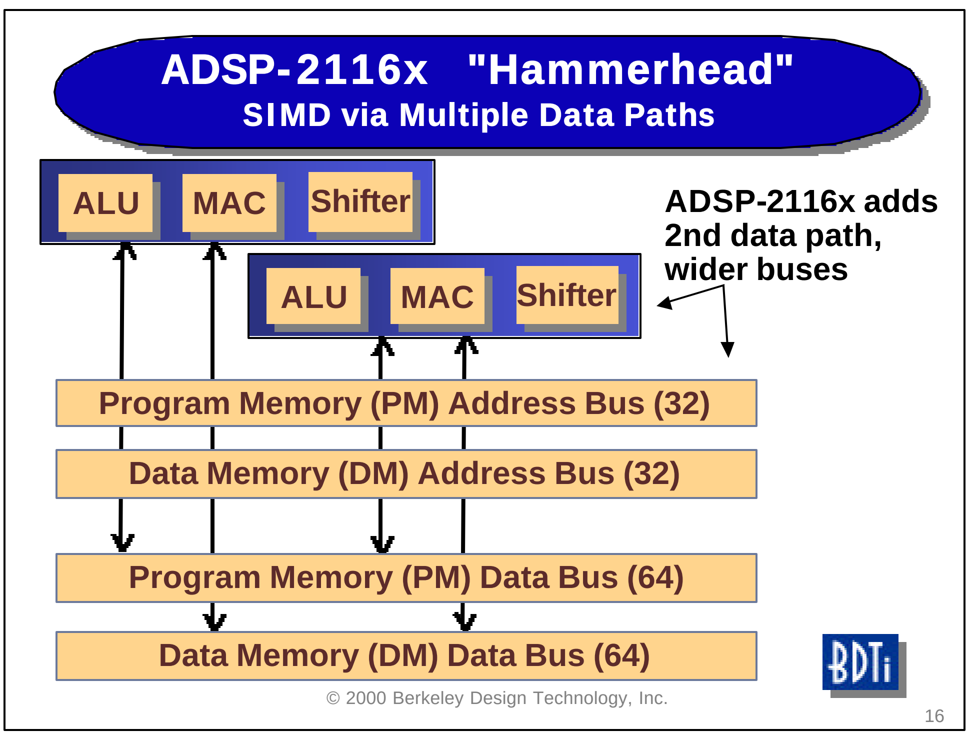
SIMD
Split Execution Unit

32-bit input register   32-bit input register



32-bit output register holds two results


ADSP-2106x "SHARC"
Conventional Floating-Point DSP
ALU
MAC
Shifter
Program Memory (PM) Address Bus (24)
Data Memory (DM) Address Bus (32)
Program Memory (PM) Data Bus (48)
Data Memory (DM) Data Bus (40)
© 2000 Berkeley Design Technology, Inc.
BDTi
15


ADSP-2116x "Hammerhead"
SIMD via Multiple Data Paths
ALU
MAC
Shifter
ALU
MAC
Shifter
ADSP-2116x adds 2nd data path, wider buses
Program Memory (PM) Address Bus (32)
Data Memory (DM) Address Bus (32)
Program Memory (PM) Data Bus (64)
Data Memory (DM) Data Bus (64)
© 2000 Berkeley Design Technology, Inc.
BDTi
16

FIR Filter Inner Loop
ADSP-2106x vs. ADSP-2116x

ADSP-2106x:

```
lcntr=TAPS-3, do macs until lce;
  macs: f12=f0*f4, f8=f8+f12, f0=dm(i0,m3), f4=pm(i8,m9);
```

One 16x16 multiply

ADSP-2116x:

```
bit set mode1 PEYEN; Turn on SIMD
lcntr=(TAPS-6)/2, do macs until lce;
  macs: f12=f0*f4, f8=f8+f12, f0=dm(i0,m3), f4=pm(i8,m9);
```

Two 16x16 multiplies: one in each data path

SIMD Characteristics

- Each instruction performs lots of work
- 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
- May support multiple data widths (e.g., 16-bit and 8-bit)

SIMD Challenges

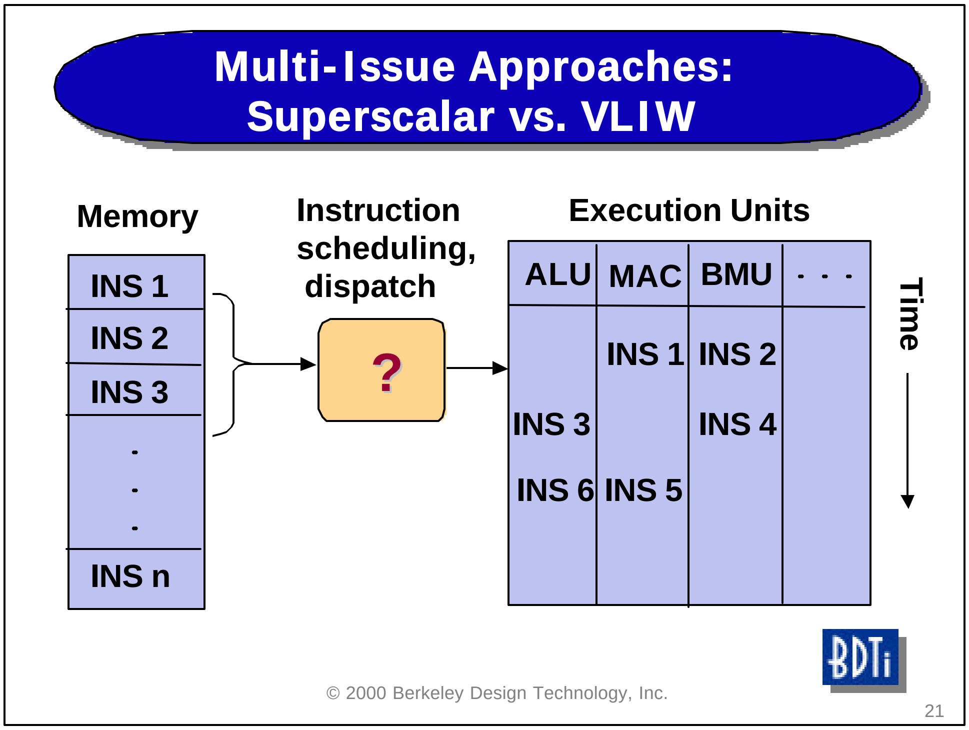
- Loss of generality
  - Each iteration of a loop processes N elements (typically $4 \leq N \leq 8$)
  - Amplified if loops are unrolled for speed
- High program memory usage
  - Re-arranging data for SIMD processing
  - Merging partial results
  - Loop unrolling
- Often, only fixed-point supported

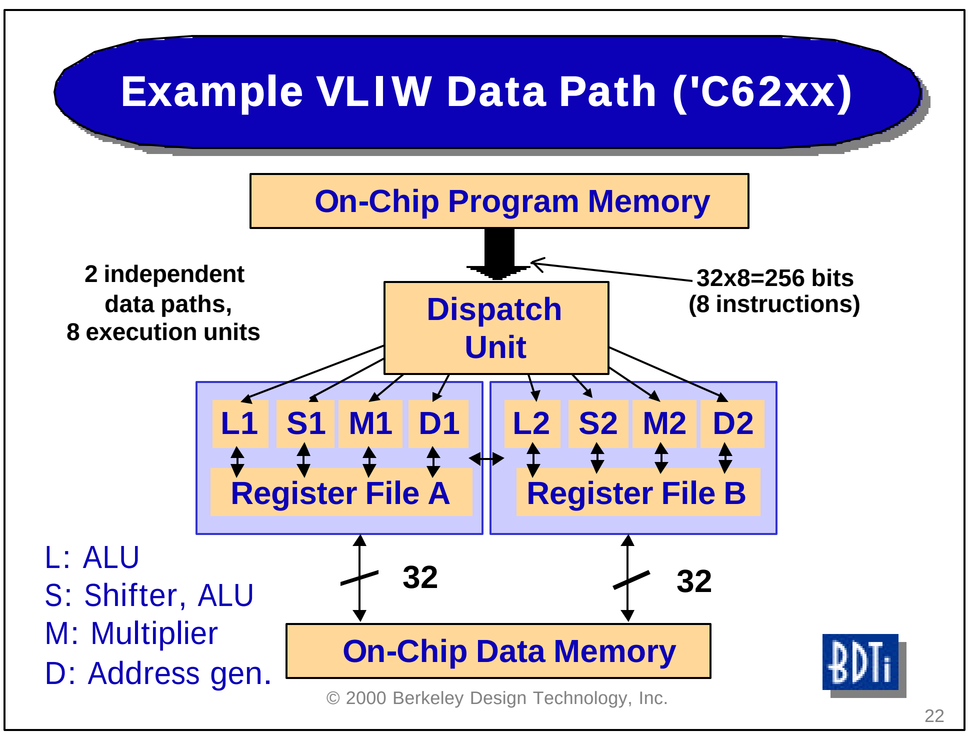
19

Multi-Issue Architectures: Why?

- Until ~1997, most DSPs were very similar
  - Specialized execution units
  - Specialized instruction sets
    - Difficult to program in assembly
    - Unfriendly compiler targets
  - One instruction per instruction cycle
- Multi-issue architectures are very different
  - They execute multiple instructions/cycle
  - They use simple, regular instruction sets
  - More parallelism, higher clocks -> faster processors
  - Better compiler targets

20


Multi-Issue Approaches:
Superscalar vs. VLIW
Memory
INS 1
INS 2
INS 3
INS n
Instruction
scheduling,
dispatch
?
Execution Units
ALU
MAC
BMU
. . .
INS 1
INS 2
INS 3
INS 4
INS 6
INS 5
Time
© 2000 Berkeley Design Technology, Inc.
21


Example VLIW Data Path ('C62xx)
On-Chip Program Memory
2 independent
data paths,
8 execution units
32x8=256 bits
(8 instructions)
Dispatch
Unit
L1
S1
M1
D1
L2
S2
M2
D2
Register File A
Register File B
L: ALU
S: Shifter, ALU
M: Multiplier
D: Address gen.
32
32
On-Chip Data Memory
© 2000 Berkeley Design Technology, Inc.
22

FIR Filter Inner Loop on TMS320C62xx

Can execute up to eight 32-bit instructions in parallel, 2 taps/cycle

```
LOOP:
  ADD   .L1 A0,A3,A0
||ADD   .L2 B1,B7,B1
||MPYHL .M1X A2,B2,A3
||MPYLH .M2X A2,B2,B7
||LDW   .D2 *B4++,B2
||LDW   .D1 *A7--,A2
||[B0] ADD .S2 -1,B0,B0
||[B0] B .S1 LOOP
```

Compare to a conventional DSP, 1 tap/cycle ...

```
dotprod:
 MR=MR+MX0*MY0(SS), MX0=DM(I0,M0),MY0=PM(I4,M4);
```


Advantages of VLIW Architectures

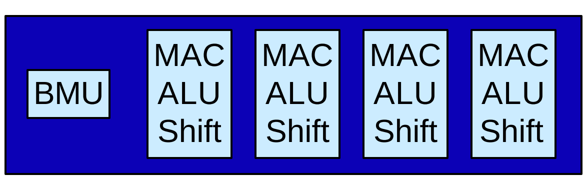
- Increased performance
- Better compiler targets
- Potentially easier to program
- Potentially scalable
  - Can add more execution units, allow more instructions to be executed in parallel as part of a VLIW instruction

Disadvantages of VLIW Architectures

- New kinds of programmer/compiler complexity
  - Programmer (or code-generation tool) must keep track of parallel instruction scheduling
  - In some processors, deep pipelines and long latencies can be confusing, may make peak performance elusive
- Increased memory use
  - High program memory bandwidth requirements
- High power consumption
- Misleading MIPS ratings

Another VLIW DSP: StarCore SC140 Core

- 16-bit fixed-point VLIW DSP core from Lucent/Motorola
  - Current development chip operates at 300 MHz
- StarCore claims it will scale the architecture
  - First VLIW architecture to target low-power apps as well as high-performance apps



Other SC100 cores may have different sets of execution units

StarCore SC140

- The SC140 addresses some of the weaknesses of the 'C62xx
  - Code density
    - Improved by using 16-bit instructions (instead of 32-bit) with 16-bit prefixes where needed
  - Programmability
    - The SC140 has a simpler pipeline than the 'C62xx (5 stages vs. 11), single-cycle latencies for nearly all instructions
  - Energy consumption
    - Narrower program bus, more efficient architecture, low-voltage process

27

Superscalar Architectures

Current superscalar DSP architectures:

- LSI Logic LSI401Z

Characteristics:

- Borrow techniques from high-end CPUs
  - Branch prediction, dynamic caching
- Multiple (usually 2-4) instructions issued per instruction cycle
- RISC-like instruction set
- Lots of parallelism

28

Superscalar Architectures

- Advantages:
  - Large jump in performance
  - More regular architectures (potentially easier to program, better compiler targets)
  - Programmer (or code generation tool) doesn't have to worry about instruction scheduling
  - Code size not increased significantly
  - Binary compatibility possible

Superscalar Architectures

- Disadvantages:
  - Energy consumption is a major challenge
  - Dynamic behavior complicates software development
    - Execution-time variability can be a hazard
    - Code optimization is challenging

Summary of DSP Architecture Types

Architecture	Issue Width	Instruction Scheduling	Instruction Type	When	SIMD	Typical Clock (MHz)
Conventional	1	Compile-time	Complex	1980-now	None to minimal	75-150
Enhanced Conventional	1	Compile-time	Complex	1996-now	Minimal to extensive	100-150
VLIW	2-8	Compile-time	Simple	1996-now	Minimal to extensive	100-300
Superscalar	2-4	Run-time	Simple	1997-now	Minimal	200

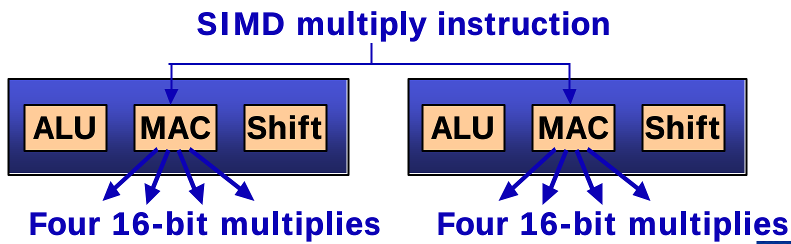
VLIW, Superscalar, SIMD

Instruction parallelism vs. data parallelism:

- SIMD uses data parallelism
  - Usually not useful for algorithms that process one sample at a time or contain tight feedback loops
- VLIW, superscalar use instruction parallelism
  - VLIW and superscalar techniques increase performance across a wider range of algorithms
- VLIW or superscalar can be combined with SIMD

ADI TigerSHARC
Combining VLIW with SIMD

- Combines VLIW with extensive SIMD to get massive parallelism
  - Using "hierarchical" SIMD, can perform eight 16x16-bit fixed-point multiplications per cycle (4X the 'C62xx)



33

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, enable higher clock speeds
  - 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

34

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

35

Hybrid DSP / Microcontrollers

- GPPs designed 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

36

Hybrid DSP / Microcontrollers
Approaches

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

37

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

38

Hybrid DSP / Microcontrollers
Advantages, Disadvantages

- DSP co-processor
  - May result in complicated programming model
    - Dual instruction sets
    - Possible deadlocks
  - Transferring data between the host and the co-processor may be time-consuming
  - Both cores can operate in parallel

Hybrid DSP / Microcontrollers
Advantages, Disadvantages

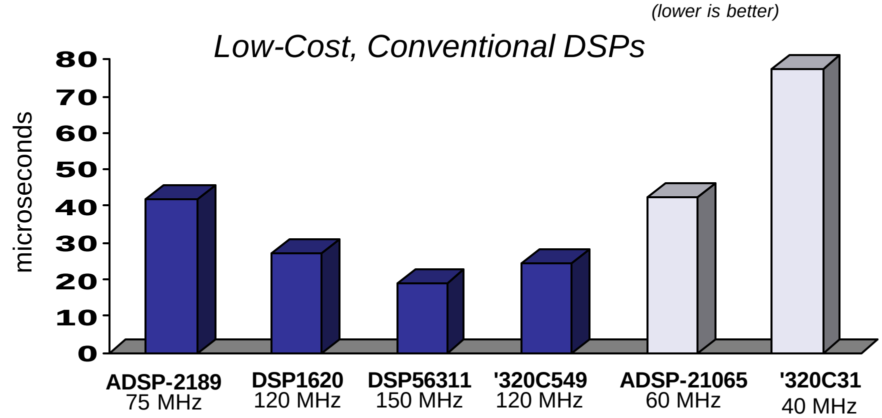
- DSP brain transplant in existing µC, microcontroller tweaks to existing DSP
  - Simpler programming model than dual cores
  - Subject to constraints imposed by "legacy" architecture
  - Allows code re-use
- Totally new design
  - Avoids legacy constraints
  - May result in a cleaner architecture
 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

Benchmark Results

41

Execution Times
Complex Block FIR Filter Benchmark

(lower is better)

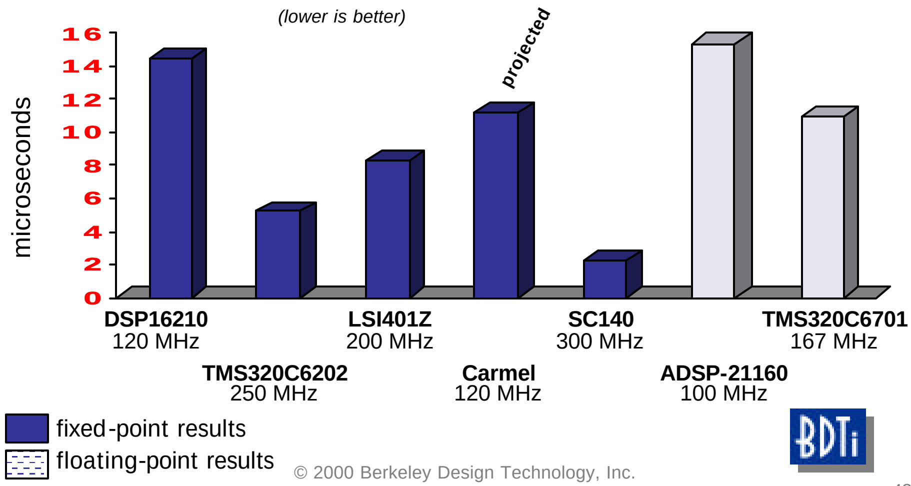


fixed-point results
floating-point results

42

Execution Times
Complex Block FIR Filter Benchmark

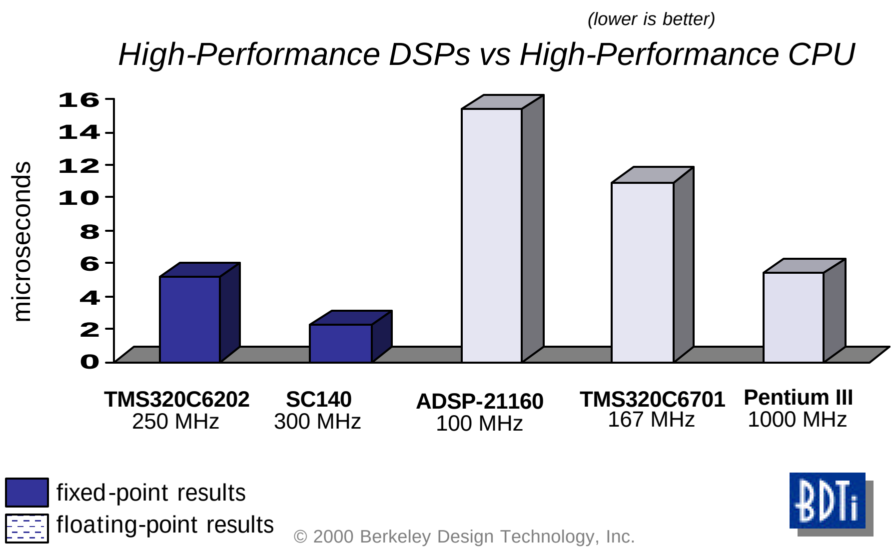
High-Performance DSPs



43

Execution Times
Complex Block FIR Filter Benchmark

High-Performance DSPs vs High-Performance CPU

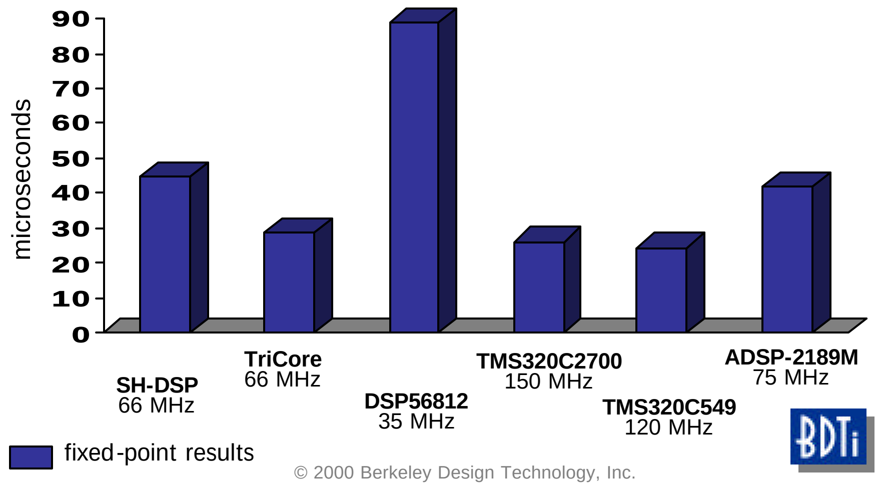


44

Execution Times
Complex Block FIR Filter Benchmark

(lower is better)

Low-Cost DSPs vs Low-Cost MCUs, Hybrids



45

Architecture Trends

- Multi-issue architectures dominate the field of new high-performance processors
  - But conventional DSPs still make up most of volume shipping today
- SIMD is becoming ubiquitous
- General-purpose processors increasingly tackling DSP, providing competition for dedicated DSP processors
- Shared DSP processor architectures emerging
- New emphasis on compatibility
- Compilability an increasingly important factor…
  - … as time-to-market pressures increase and applications become larger

46

For More Information...
www.BDTI.com

- White papers on DSP processor architectures and benchmarking
- Article reprints on DSP-oriented processors and apps
  - *Microprocessor Report*
  - *IEEE Spectrum*
  - *IEEE Computer* and others
- *comp.dsp* FAQ
- BDTImark2000 scores (coming soon)

47