

The Evolution of DSP Processors

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>



Outline

- ◆ DSP applications
- ◆ Digital filtering as a motivating problem
- ◆ The first generation of DSPs, with an example
- ◆ Comparison of DSP processors to general-purpose processors
- ◆ DSP evolution continues... later-generation DSPs and alternatives
- ◆ Conclusions

Who Cares?

- ◆ DSP is a key enabling technology for many types of electronic products
- ◆ DSP-intensive tasks are the performance bottleneck in many computer applications today
- ◆ Computational demands of DSP-intensive tasks are increasing very rapidly
- ◆ In many embedded applications, general-purpose microprocessors are not competitive with DSP-oriented processors today
- ◆ 1998 market for DSP processors: US \$3.2 billion



Example DSP Applications

- ◆ Digital cell phones
- ◆ Automated inspection
- ◆ Vehicle collision avoidance
- ◆ Voice-over-Internet
- ◆ Motor control
- ◆ Consumer audio
- ◆ Voice mail
- ◆ Navigation equipment
- ◆ Audio production
- ◆ Videoconferencing
- ◆ Pagers
- ◆ Music synthesis, effects
- ◆ Satellite communications
- ◆ Seismic analysis
- ◆ Secure communications
- ◆ Tapeless answering machines
- ◆ Sonar
- ◆ Cordless phones
- ◆ Digital cameras
- ◆ Modems (POTS, ISDN, cable, ...)
- ◆ Noise cancellation
- ◆ Medical ultrasound
- ◆ Patient monitoring
- ◆ Radar

And more to come...



This is Your Palm Pilot



This is Your Palm Pilot... On DSP



Hello, Dave. You have a meeting in 10 minutes.



Today's DSP "Killer Apps"

- ◆ In terms of dollar volume, the biggest markets for DSP processors today include:

- Digital cellular telephony
- Pagers and other wireless systems
- Modems
- Disk drive servo control

- Most demand good performance
- All demand low cost
- Many demand high energy efficiency

- Trends are towards better support for these (and similar) major applications.



DSP Tasks for Microprocessors

- ◆ Speech and audio compression
- ◆ Filtering
- ◆ Modulation and demodulation
- ◆ Error correction coding and decoding
- ◆ Servo control
- ◆ Audio processing (e.g., surround-sound, noise reduction, equalization, sample rate conversion, echo cancellation)
- ◆ Signaling (e.g., DTMF)
- ◆ Speech recognition
- ◆ Signal synthesis (e.g., music, speech)



What Do DSP Processors Need to Do Well?

Most DSP tasks require:

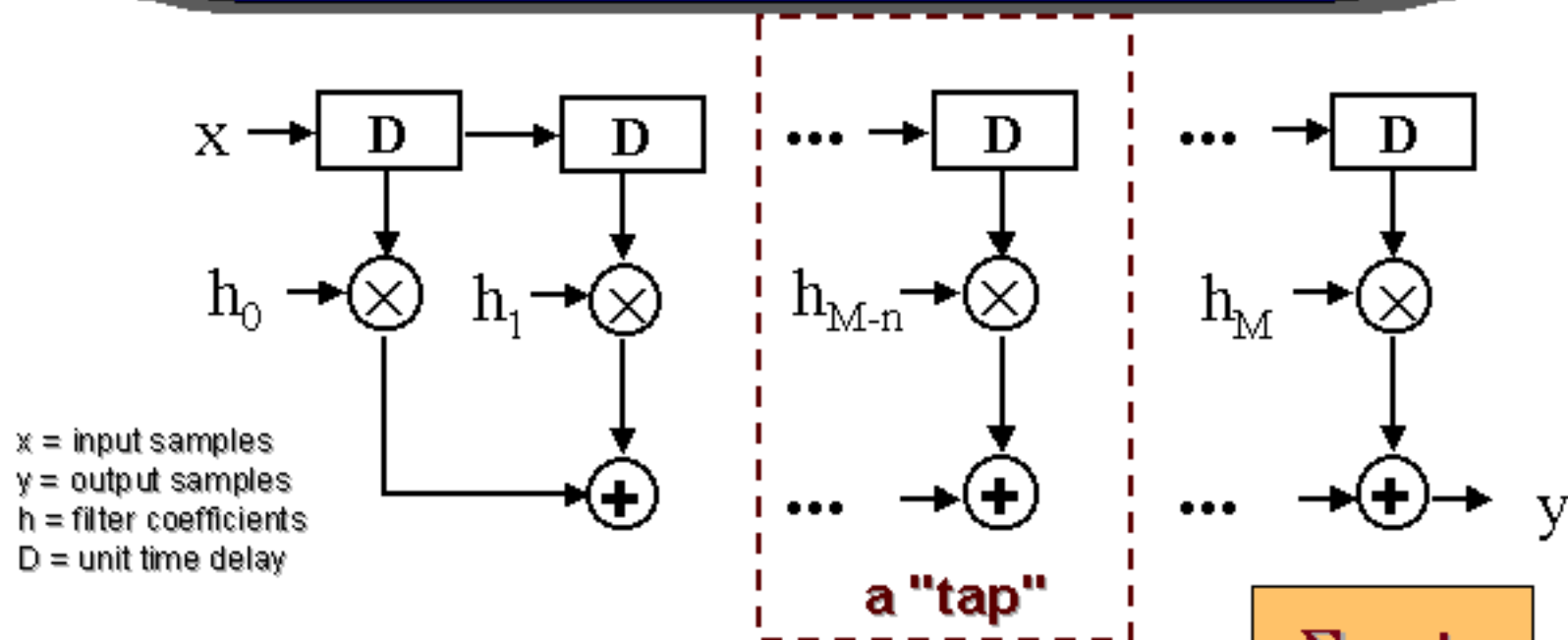
- Repetitive numeric calculations
- Attention to numeric fidelity
 - Fixed- vs floating-point
 - Standards
- High memory bandwidth
 - Streaming data
- Real-time processing

Processors must perform these tasks efficiently while minimizing:

- Cost
- Power
- Memory use
- Development time



A Motivating Example: FIR Filtering



Each tap ($M+1$ taps total) nominally requires:

- Two data fetches
- Multiply
- Accumulate
- Memory write-back to update delay line

FIR Filter on Von Neumann Architecture

```
loop:
    mov    *r0,x0
    mov    *r1,y0
    mpy    x0,y0,a
    add    a,b
    mov    y0,*r2
    inc    r0
    inc    r1
    inc    r2
    dec    ctr
    tst    ctr
    jnz    loop
```



Data Path ↔ **Memory**

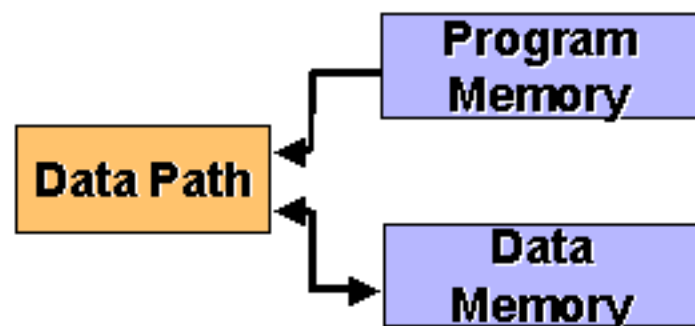
Problems:

- Memory bandwidth bottleneck
- Control code and addressing overhead
- Possibly slow multiply

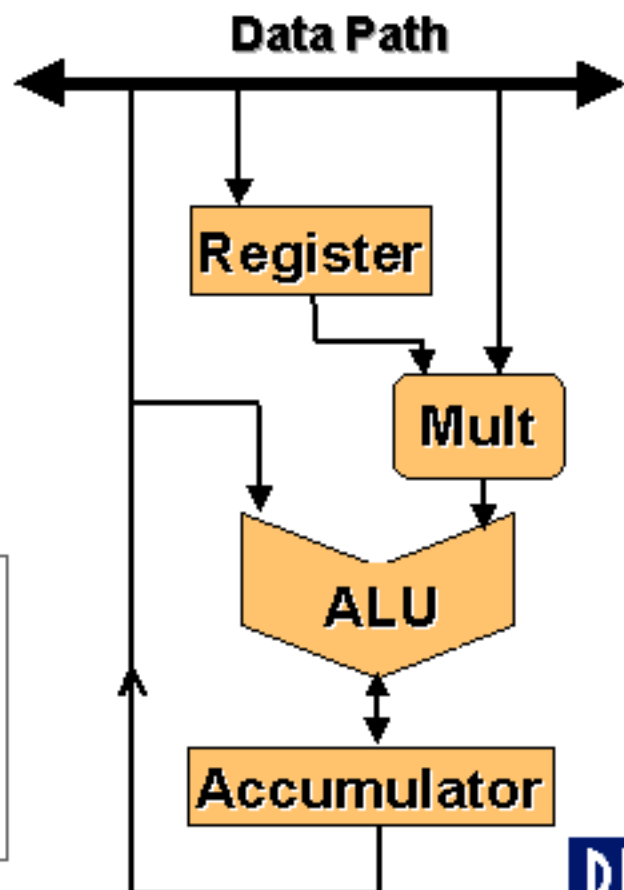
(Computes one tap per loop iteration)

First-Generation DSP (1982): Texas Instruments TMS32010

Memory Structure



- 16-bit fixed-point
- Harvard architecture
- Accumulator
- Specialized instruction set
- 390 ns MAC time (228 ns today)



TMS32010 Filter Code

```
LT      X4      ;Load T with x(n-4)
MPY     H4      ;P=H4*X4
LTD     X3      ;Load T with x(n-3);x(n-4)= x(n-3)
        ;Acc = Acc + P
MPY     H3      ;P=H3*X3

LTD     X2
MPY     H2

etc.
```

- ◆ Two instructions per tap, but requires unrolling

Features Common to Most DSP Processors

- ◆ Data path configured for DSP
- ◆ Specialized instruction set
- ◆ Multiple memory banks and buses
- ◆ Specialized addressing modes
- ◆ Specialized execution control
- ◆ Specialized peripherals for DSP

Data Path Comparison

DSP Processor

- ◆ Specialized hardware performs all key arithmetic operations in 1 cycle
- ◆ Hardware support for managing numeric fidelity:
 - Shifters
 - Guard bits
 - Saturation

General-Purpose Processor

- ◆ Multiplies often take >1 cycle
- ◆ Shifts often take >1 cycle
- ◆ Other operations (e.g., saturation, rounding) typically take multiple cycles

Instruction Set Comparison

DSP Processor

- ◆ Specialized, complex instructions
- ◆ Multiple operations per instruction

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

General-Purpose Processor

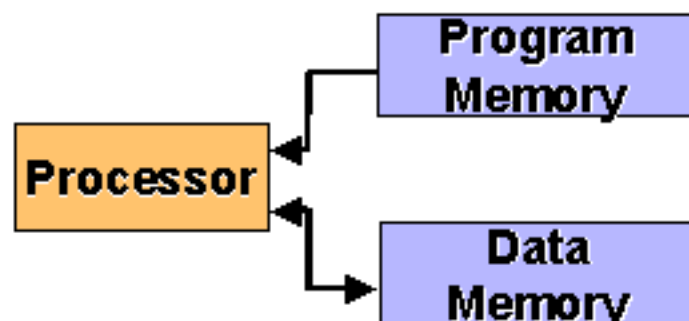
- ◆ General-purpose instructions
- ◆ Typically only one operation per instruction

```
mov  *r0,x0
mov  *r1,y0
mpy  x0,y0,a
add  a,b
mov  y0,*r2
inc  r0
inc  r1
```


Memory Architecture Comparison

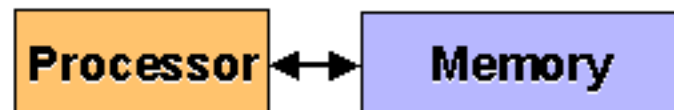
DSP Processor

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



General-Purpose Processor

- ◆ Von Neumann architecture
- ◆ Typically 1 access per cycle
- ◆ May use caches



Addressing Comparison

DSP Processor

- ◆ Dedicated address-generation units
- ◆ Specialized addressing modes
 - Autoincrement
 - Modulo (circular)
 - Bit-reversed (for FFT)
- ◆ Good immediate data support

General-Purpose Processor

- ◆ Often, no separate address-generation units
- ◆ General-purpose addressing modes

Execution Control

- ◆ Hardware support for fast looping
- ◆ "Fast interrupts" for I/O handling
- ◆ Real-time debugging support

Specialized Peripherals for DSP

- ◆ Synchronous serial ports
 - ◆ Parallel ports
 - ◆ Timers
 - ◆ On-chip A/D, D/A converters
 - ◆ Host ports
 - ◆ Bit I/O ports
 - ◆ On-chip DMA controller
 - ◆ Clock generators
- ◆ On-chip peripherals often designed for "background" operation, even when core is powered down.

Summary of DSP Attributes

- 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 addressing, bit-reversed addressing



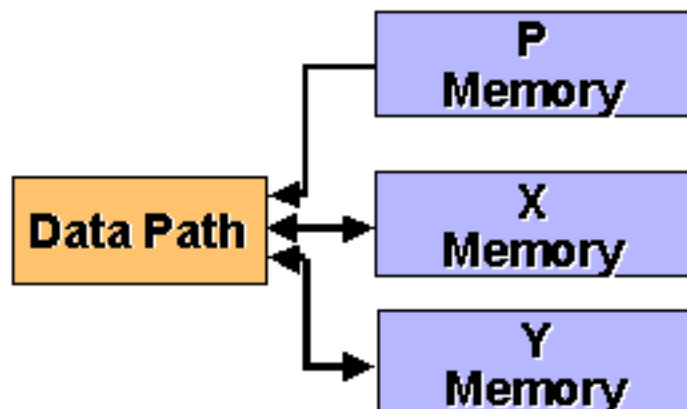
Summary of DSP Attributes

- Execution-time locality → Hardware-assisted zero-overhead looping, specialized instruction caches, streamlined interrupt handling
- MAC-centricity → Single-cycle multiplier(s) or MAC unit(s), MAC instruction
- Streaming data → No data cache; DMA
- Real-time constraints → Few dynamic features, on-chip RAM instead of cache
- Standards → Rounding, saturation



Second-Generation DSPs (1987): Motorola DSP56001

- ◆ 24-bit data, instructions
- ◆ 3 memory spaces (X, Y, P)
- ◆ Single- and multi-instruction hardware loops
- ◆ Modulo addressing
- ◆ 75 ns MAC (21 ns today)



```
move #Xaddr, r0
move #Haddr, r4
rep #Ntaps
mac x0, y0, a    x: (r0)+, x0    y: (r4)+, y0
```

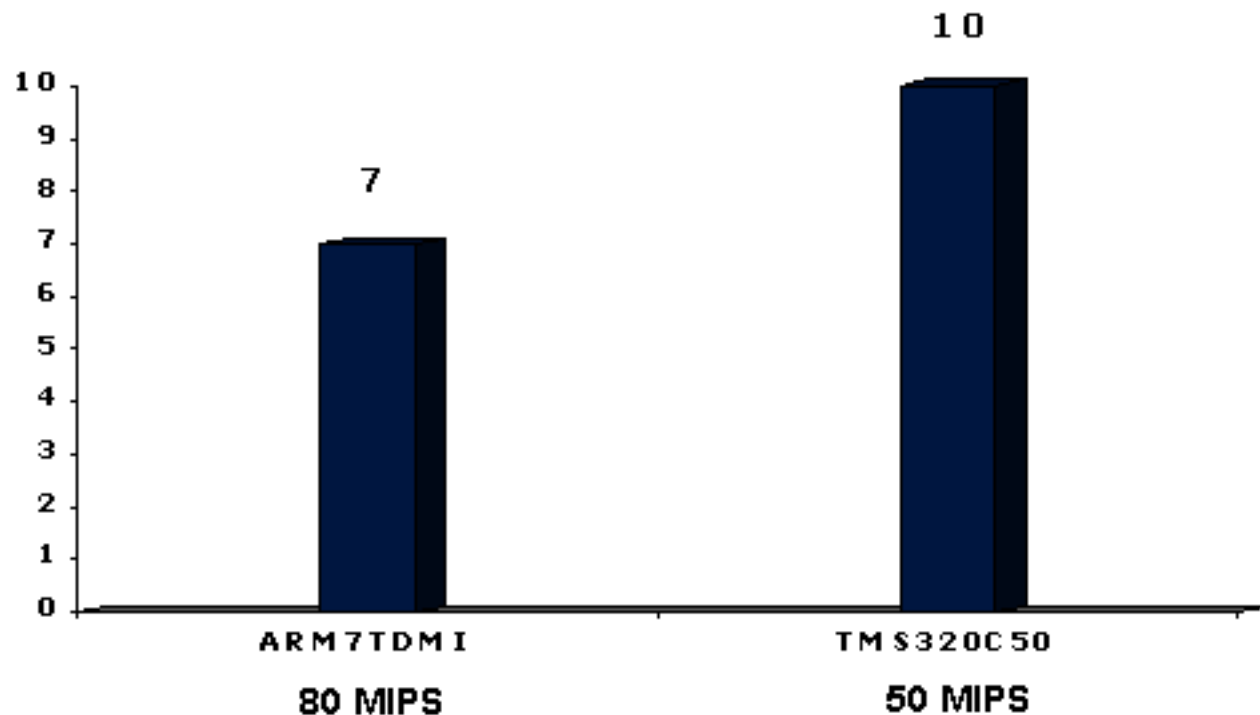
- ◆ Other 2nd-generation processors: Analog Devices ADSP-2100, TI TMS320C50



Low-cost GPP vs Low-Cost DSP

Speed (BDTImarks™)

**Note that
MIPS \neq Performance!**



Third Generation (1995): Ex: Motorola DSP56301, TI TMS320C541

- ◆ Enhanced conventional DSP architectures
- ◆ 3.0 or 3.3 volts
- ◆ More on-chip memory
- ◆ Application-specific function units in data path or as co-processors
- ◆ More sophisticated debugging and application development tools
- ◆ DSP cores (Pine, Oak from DSP Group, cDSP from TI)
- ◆ 20 ns MAC (10 ns today)
- ◆ Architectural innovation mostly limited to adding application-specific function units and miscellaneous minor refinements
- ◆ Also, multiple processors on a chip (TI TMS320C80, Motorola MC68356)



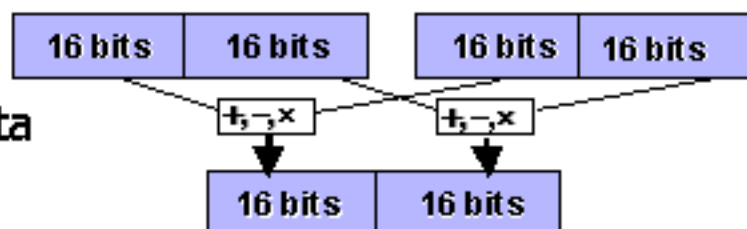
Fourth Generation (1997-1998):

Ex: TMS320C6201/6701, ZSP16401, MMX Pentium

Today's top DSP performers adopt architectures far different from conventional DSP processor designs:

◆ SIMD

- Single instruction, multiple data (e.g., MMX, AltiVec, MDMX)



◆ VLIW

- "Very long instruction word"
- Compile-time scheduling and parallel execution of multiple simple instructions (e.g., TMS320C6201/C6701)

◆ Superscalar

- Run-time scheduling and execution of >1 (usually 2-4) instructions per cycle (e.g., Pentium, PowerPC, ZSP164xx)

◆ User-defined instructions

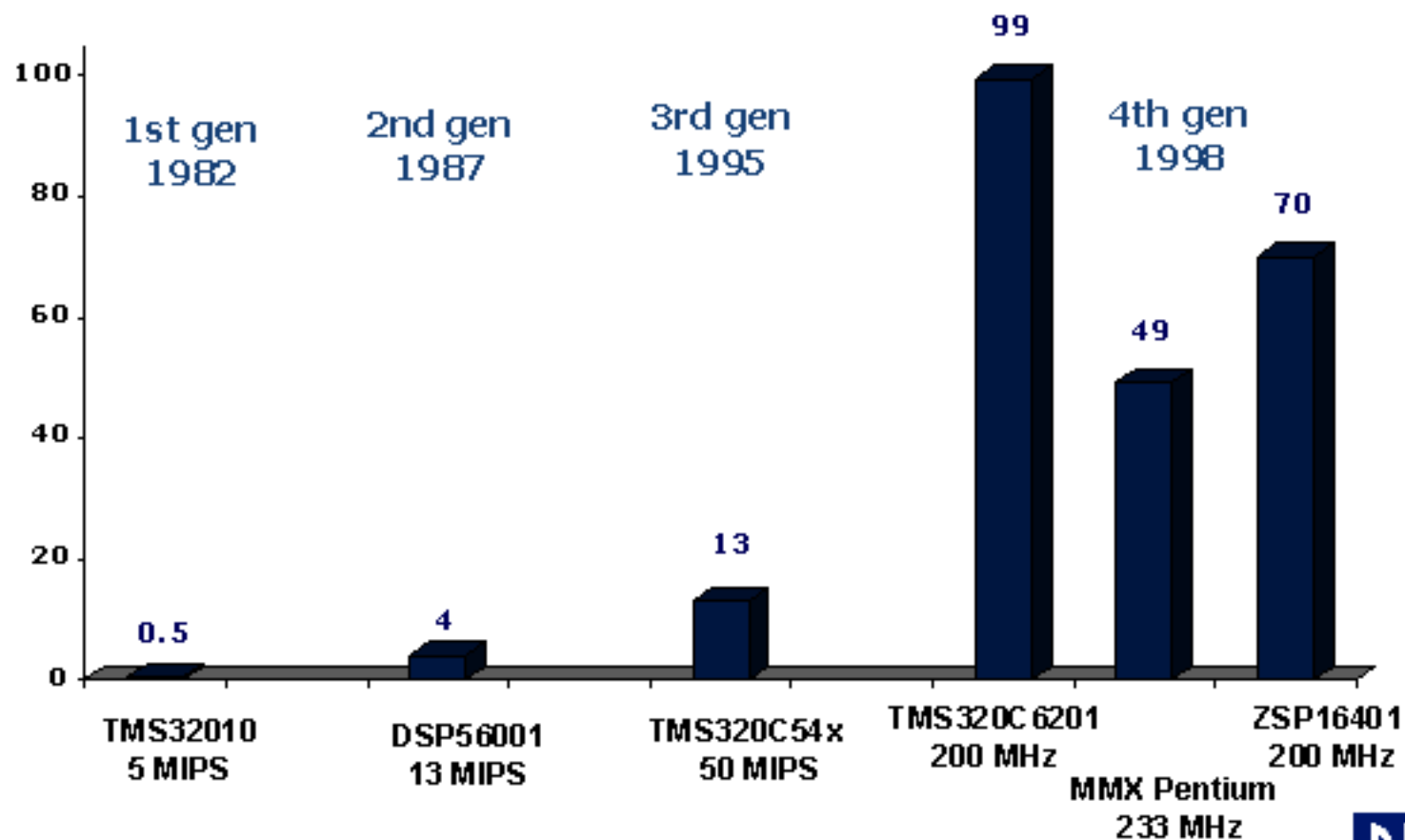


Fourth Generation Architectures

- ◆ Blazing clock speeds and superscalar architectures give some general-purpose processors, such as the PowerPC 604e, extremely good floating-point DSP performance *despite a lack of many DSP features*
- ◆ Adding SIMD extensions, such as MMX, MDMX, and AltiVec, yields strong fixed-point performance on GPPs
- ◆ *But strong DSP tools for general-purpose processors are lacking*

- ◆ VLIW-like and superscalar DSP architectures achieve top performance via high parallelism and increased clock speeds
- ◆ 3 ns MAC throughput... but expensive, power-hungry

Processor DSP Speed: BDTImarks



General-Purpose Processors Add DSP

"Go where the cycles are..."

General-purpose processors are increasingly adding DSP capabilities via a variety of mechanisms:

- Add SIMD capabilities
- Integrate a fixed-point DSP processor-like data path and related resources with an existing $\mu\text{C}/\mu\text{P}$ core (e.g., Hitachi SH-DSP)
- Add a DSP co-processor to an existing $\mu\text{C}/\mu\text{P}$ core (e.g., ARM Piccolo)
- Create an all-new, hybrid architecture (e.g., Siemens TriCore)



Conclusions

- ◆ DSP processor performance has increased by a factor of about 150x over the past 15 years ($\sim 40\%$ per year)
- ◆ Processor architectures for DSP will be increasingly specialized for applications, especially for communications applications
- ◆ General-purpose processors will continue to add DSP capabilities; GPP and DSP family trees will mix
- ◆ Users of processors for DSP will have an expanding array of choices
- ◆ Selecting processors requires careful, application-specific analysis



For More Information

<http://www.bdti.com>

Collection of BDTI's papers on
DSP processors, tools, and
benchmarking

<http://www.eg3.com/dsp>

Links to other good DSP sites

<comp.dsp>

Usenet group

Microprocessor Report

For info on newer DSPs

*DSP Processor Fundamentals,
BDTI*

Textbook on DSP processors

Or, join BDTI...We're Hiring!

