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
- Modern DSP-enhanced general-purpose processors
- Benchmark results
- 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 DSP applications, general-purpose microprocessors are not competitive with DSP-oriented processors today

Example DSP Applications
- Digital cell phones
- Automated inspection
- Vehicle collision avoidance
- Voice-over Internet
- Motor control
- Consumer audio
- Voice mail
- Navigation equipment
- Audio production
- Video conferencing
- Toys, games consoles
- Music synthesis, effects
- Satellite communications
- Seismic analysis
- Secure communications
- Tapeless answering machines
- Sonar
- Cordless phones
- Digital cameras
- Moderns (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"

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

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

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

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 consumption
- 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

```assembly
loop:    mov  *r0,x0
         mov  *r1,y0
         mpy  x0,y0,a
         add  a,b
         mov  y0,*r2
         inc  r0
         inc  r1
         inc  r2
         dec  cte
         tat  cte
         jnz  loop
```

(Computes one tap per loop iteration)

First-Generation DSP (1982): Texas Instruments TMS32010

- 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 loop 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
- Multiples 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

General-Purpose Processor
- General-purpose instructions
- Typically only one operation per instruction

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)

General-Purpose Processor
- General-purpose addressing modes
  - Favor compiler-generated code

- Good immediate data support
Execution Control

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

**General-Purpose Processor**
- Loops implemented in software
- Pipelines can increase cost of loops
- Interrupt overhead can be large for simple interrupts
- On-chip debug: usually not real-time

Specialized I/O for DSP

- Synchronous serial ports
- Parallel ports
- Timers
- On-chip A/D, D/A converters

- Host ports
- BI/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; powerful 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)

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

Low-cost GPP vs Low-Cost DSP

**Speed (BDTImarks) Note that MIPS = Performance!**

<table>
<thead>
<tr>
<th>Processor</th>
<th>Speed (BDTImarks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM7TDMI</td>
<td>7 ARM7TDMI</td>
</tr>
<tr>
<td>ADSP-218x</td>
<td>19 ADSP-218x</td>
</tr>
</tbody>
</table>

80 MIPS          75 MIPS
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)

Ex: TMS320C6201/6701, LSI401Z, 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

General-Purpose Processors Add DSP

SIMD
Single Instruction, Multiple Data
- Virtually all high-performance CPUs (and some modern DSPs) support SIMD operations
- One SIMD 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 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)
**SI MD 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

**High-Performance GPPs with SI MD**

- 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

**High-Performance GPPs with SI MD**

- 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

**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

### Approaches

- Multiple processors on a die
  - e.g., Motorola DSP5665x
- DSP coprocessor
  - e.g., Massana FILU-200
- DSP brain transplant in existing µC
  - e.g., 80-HDSP
- Microcontroller tweaks to existing DSP
  - e.g., TMS320C27xx
- Totally new design
  - e.g., TriCore

### 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
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

- 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

Processor DSP Speed: BDTI marks
(Higher is Better)

<table>
<thead>
<tr>
<th>1st gen</th>
<th>2nd gen</th>
<th>3rd gen</th>
<th>4th gen</th>
</tr>
</thead>
</table>

- TMS32010 5 MIPS
- TMS320C6201 250 MHz
- DSP56001 13 MIPS
- TMS320C54x 50 MIPS
- TMS320C6711 233 MHz

Speech
Audio
"2G" Wireless
Speech
Conclusions

- DSP processor performance has increased by a factor of about 150x over the past 15 years (~40% per year).
- Multi-issue architectures dominate the field of new high-performance processors.
  - But conventional DSPs still make up most of volume shipping today.
- General-purpose processors increasingly tackling DSP, providing competition for dedicated DSP processors.
- Users of processors for DSP will have an expanding array of choices.
- Compiler-friendliness is an increasingly important factor...
  - ...as time-to-market pressures increase and applications become larger.
- 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.eq3.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! (see www.BDTI.com)