Digital Signal Processing Technology
Insight • Analysis • Advice

Using General-Purpose Processors for Signal Processing

ARM Developers' Conference 2004

Berkeley Design Technology, Inc.
http://www.BDTI.com

Parallelism

DSPs generally provide more parallelism than GPPs

- Instruction set style—individual instructions are
  - GPP: RISC—non-specialized, single operation per instruction
  - DSP: Compound—multi-operation per instruction, or RISC with multi-issue
- Multi-issue support—multiple instructions per cycle
  - GPP: Superscalar—hardware performs dynamic instruction grouping/scheduling
  - DSP: VLIW—programmer or compiler/assembler performs instruction grouping/scheduling
- SIMD
  - Both GPPs and DSPs provide SIMD support
  - SIMD is a primary means of parallelism in GPPs, secondary in DSPs

Parallelism

Key implications of differences

- Cycle efficiency
  - DSPs have advantage on signal processing tasks
    - But may require special software development strategies—like assembly level programming—to realize full advantage
- Memory use efficiency
  - Multi-operation instructions give DSPs advantage on signal processing tasks
  - But GPPs often better on non-signal processing tasks—which typically consumes most of the code space
- Compiler friendliness
  - GPPs generally have the advantage
  - SIMD difficult for compilers, whether GPP or DSP
    - Often requires assembly programming or use of high level intrinsics—both of which complicate software development

Memory Architecture

All DSPs and most GPPs use Harvard memory architectures—with separate instruction and data buses

- DSPs generally use SRAM with DMA controllers
- GPPs generally use cache memory
  - Caches can pose challenges for signal processing software
- DSPs typically have more on-chip data buses and greater data bandwidth than GPPs
- DSPs often have higher bandwidth off-chip memory interfaces
  - E.g., separate data and address buses vs. multiplexed
- GPPs often have larger address spaces than DSPs

Memory Architecture

Key implications of differences

- Cache related issues are paramount
  - Caches work by lowering *average* access time
    - They are effective at doing this in many applications
    - But access times vary significantly
  - Some applications are sensitive to *maximum* access time
  - Signal processing application access patterns tend to be predictable
    - Thus, DMA may be preferable to a cache
    - Some recent caches provide pre-fetching capability
- Signal processing tasks benefit from high bandwidth memory interfaces

Execution Time Predictability

Dynamic features are used heavily in high-end GPPs to boost performance

- Superscalar execution
- Caches
- Branch prediction
- Data-dependent instruction execution times

These features are occasionally used in DSPs too

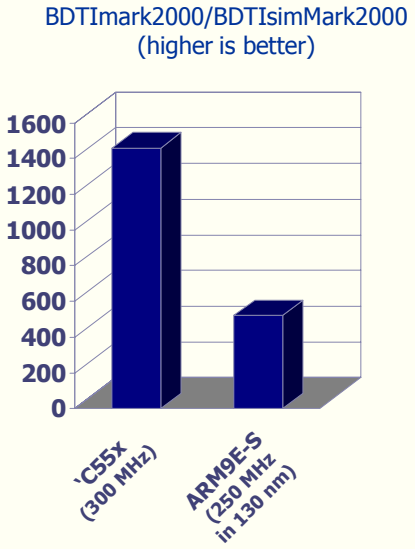
These features complicate software development for real-time signal processing applications

- Ensuring real-time behavior
- Optimizing code

Speed

BDTImark2000™ is a summary score of signal processing speed—based on the execution time of the BDTI™ Benchmarks

The DSP is about 3X faster, although its clock speed is only 20% higher—parallelism is the main differentiator

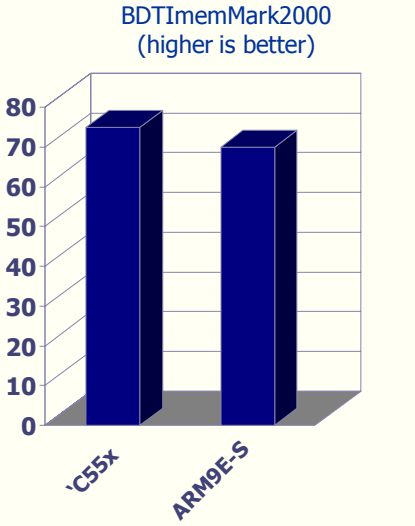


Memory Use

BDTImemMark2000™ is a summary score of total memory use—based on the BDTI™ Benchmarks, with emphasis on the Control Benchmark.

The DSP and GPP have similar scores.

Results are based on hand coded assembly—compiler efficiency is not reflected



Development Support

	DSPs	GPPs
Tools	Primitive to moderately sophisticated	Primitive to very sophisticated
Signal-processing-specific tool support	Good to excellent E.g., cycle-accurate simulators, DSP C extensions	Poor but improving E.g., general lack of cycle-accurate simulators
Signal processing software support	Poor to excellent	Limited but growing
Non-signal-processing software support	Poor Few to moderate RTOS options	Fair to extensive Few to extensive RTOS options

Development Support

- Software availability is key
  - Applications with extensive signal processing functionality may favor a DSP for software availability
  - Mainstream OS requirement may force the use of a GPP
  - Mainstream OS and heavy signal processing may suggest use of both DSP and GPP
- Compatibility between generations
- Shared architectures
  - Multiple chip vendors supplying same architecture

Conclusions

- Performance is not the only consideration
- Other prominent factors include
  - Cost
  - Integration
  - Energy efficiency
  - Compatibility, availability
    - Multi-vendor architectures
    - Licensable cores
  - Tools
    - Signal-processing-oriented
    - Other-oriented
- Software infrastructure is key
  - DSPs have the advantage for signal processing tasks
  - GPPs have the advantage for other tasks

For More Information…
www.BDTI.com

Free Information
- Inside [DSP]
  - BDTI and *EE Times* on digital signal processing
- Processor benchmarking
  - BDTImark2000™ scores for dozens of processors
- Pocket Guide to Processors for DSP
  - Basic stats on 40+ processors
- White papers/presentation slides on
  - Signal processing software optimization
  - Processor architectures and performance
- Article reprints on signal-processing-oriented processors and applications
- *comp.dsp* FAQ

2004 Edition