A DETAILED ANALYSIS OF MEDIABENCH

Benjamin Bishop Thomas P. Kelliher * Mary Jane Irwin

MicroSystems Design Lab
Department of Computer Science and Engineering
Penn State
University Park, PA 16802

Abstract - In this paper, we present a detailed analysis of the MediaBench benchmark suite. MediaBench consists of a number of popular embedded applications for communications and multimedia. MediaBench performance characteristics were examined by running MediaBench under the SimpleScalar simulation environment. Characteristics such as instruction mix, branch prediction accuracy, cache hit rates, memory usage, and integer bit utilization were considered. This information can be of use in designing embedded systems targeted at multimedia applications.

1 Introduction

As personal communications and multimedia devices become more commonplace, embedded processors should be designed in order to efficiently meet the needs of these applications. Recently, MediaBench [1] has been proposed as a benchmark set representative of multimedia and communications applications. The MediaBench paper argues that many existing benchmarks, SPECint, DSPstone, and Dhrystone for example, are not representative of embedded applications. In order to get a benchmark set that is closely related to current embedded applications, the authors have selected many current and popular multimedia applications. The performance characteristics of MediaBench were found to be quite distinct from SPEC, under the HPPA architecture for a number of metrics.

The SimpleScalar Tool Set [2] is a set of publicly available tools offering an environment for detailed simulations of modern microprocessors. SimpleScalar's instruction set architecture is based on that of the MIPS/DLX.

*Thomas P. Kelliher is with the Department of Mathematics & Computer Science, Goucher College, Baltimore, MD 21204.

The tool set provides several different simulators: functional, cache, profiling, and out-of-order, among others. Additionally, researchers can write new SimpleScalar simulators to test new microarchitectural ideas.

The contribution of this paper lies in three areas: We have ported MediaBench to SimpleScalar, providing the embedded systems design community with a set of tools for evaluating new designs. Our MediaBench port is available at [5]. We have extended the sim-profile simulator to provide integer bit utilization information, in order to give an approximation of the potential for reduced-precision SIMD operations (e.g. MMX). Finally, we report the results of profiling MediaBench on an embedded processor model, extending original MediaBench work by presenting a great deal of new execution characteristics. These characteristics include: branch prediction information, memory usage, instruction mix, and integer bit utilization.

In the following sections, we describe the MediaBench benchmarks and our benchmarking methodology. This is followed by the presentation and analysis of our results and a conclusion.

1.1 MediaBench

This section gives a brief description the the benchmarks simulated:

JPEG: Lossy compression for still images. (Encode/Decode)
MPEG: Lossy compression for video. (Encode/Decode)
GSM: European standard for speech transcoding. (Encode/Decode)
G721: CCITT voice compression. (Encode/Decode)
PGP: IDEA/RSA public-key encryption algorithm. (Encrypt/Decrypt)
PEGWIT: Elliptical curve public-key encryption algorithm. (Encrypt/Decrypt)
Ghostscript: Postscript interpreter.
Mesa: Public OpenGL clone. (Mipmap, Osdemo, Texgen)
RASTA: Speech recognition.
EPIC: Experimental image compression. (Encode/Decode)
ADPCM: Adaptive differential pulse code modulation audio coding. (Encode/Decode)

In addition to the benchmark executable, the MediaBench distribution also contains standard data sets. Please see [1] for more detail.

In order to facilitate a comparison with more conventional benchmarks, the SPECint95 benchmark set was also simulated. The reference data set was used for the SPEC simulations. Due to time constraints, SPEC simulations were limited to 5 million instructions.

2 Simulation Methodology

In order to generate all results of interest, two series of SimpleScalar simulations were run. Sim-profile was modified to report information on integer bit utilization during load/store operations. The authors believe that this gives a reasonable approximation of the potential for reduced precision SIMD operations. The Sim-profile simulations reported instruction mix and integer bit utilization. Sim-outorder was used in order to collect information about memory utilization (static and dynamic), cache hit rates, and branch prediction accuracy. There was no operating system overhead in either simulation.

The goal of these simulations was to model a low complexity embedded processor, and simulation parameters were chosen in order to resemble an embedded processor whenever possible. All simulations used single issue, in-order processor models. Please note that Sim-outorder was used not for its multiple issue/out of order abilities, but for its additional profiling options. The cache statistics were collected assuming StrongARM 110-like on-chip caches [3]. These are split level 1 instruction/data caches of size 16kB with 32 way associativity. Block size was set to be 32 words (in order to take advantage of expected spatial locality in data and loop unrolling in code). A random replacement policy was used. No off-chip caches were considered. In addition, a bimodal predictor [4] was used in order to collect branch prediction information.

3 Results

In this section, we present the results obtained using the simulation methodology described above. All figures show results for each MediaBench benchmark individually, as well as means for MediaBench and SPECint95. Due to large variations, Figures 2, 3, 4, 5, and 6 use a geometric mean. Figures 1 and 7 use an arithmetic mean.

Figure 1 gives the instruction probability by instruction group. Integer operations tend to be more common in MediaBench, store operations are noticeably less frequent. Figure 2 describes the branch prediction rate assuming a bimodal predictor. Branch prediction is more difficult in MediaBench. This could be caused by data dependent branches, which are harder to predict. Conditionals to compute an absolute value in MPEG encoding are an example of this. Figures 3 and 4 describe the cache assuming a StrongARM 110-like cache organization. The SPEC benchmarks showed much lower hit rates, but this could be due to higher startup costs (since the first 5 million instructions were issued), and larger data sets. Figure 5 shows the memory footprint of the benchmarks. Figure 6 shows static memory consumption by code or data information. Figure 7 gives a breakdown of bit utilization likelihood for granularities of 8 bits, 16 bits, and 32 bits. Note that, for both benchmark sets, a significant number of operations do not require 32 bits of precision.

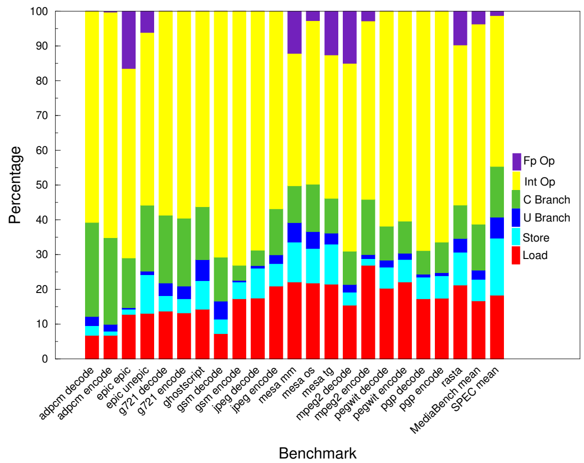

Figure 1: Instruction mix.

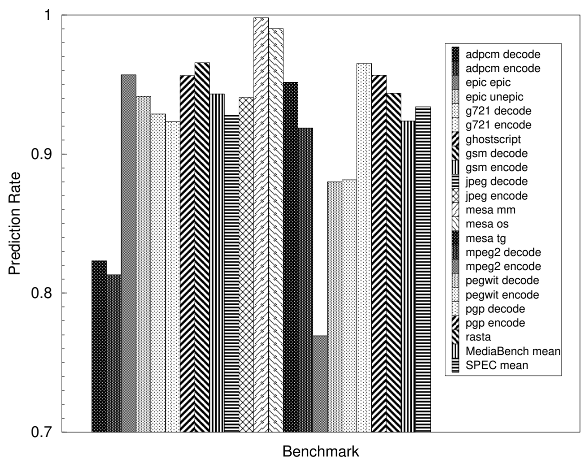

Figure 2: Branch prediction.

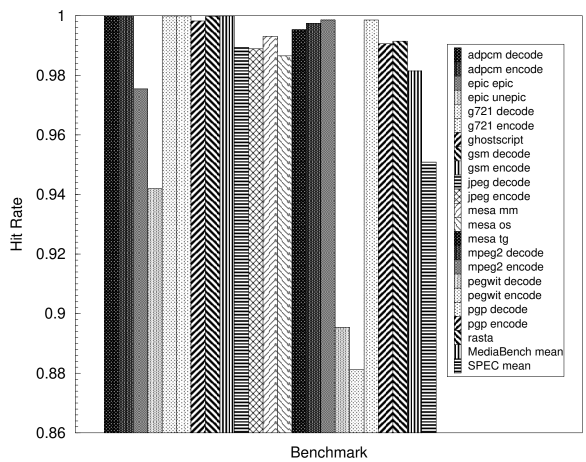

Figure 3: Data cache hit rate.

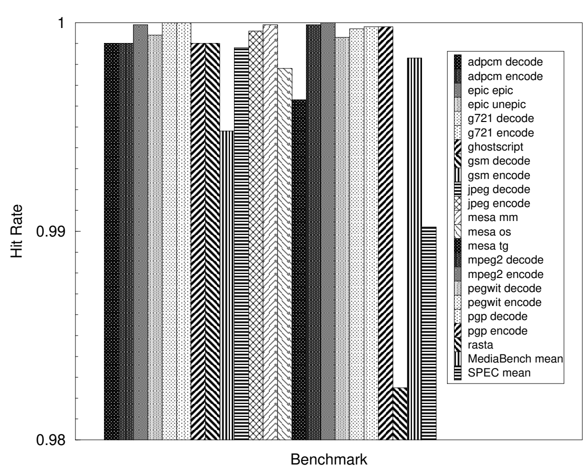

Figure 4: Instruction cache hit rate.

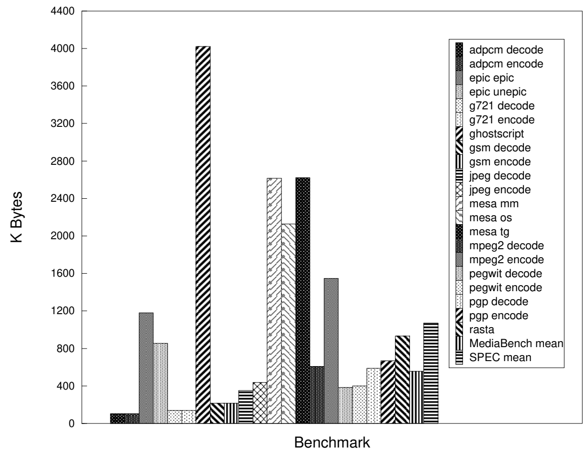

Figure 5: Memory utilization.

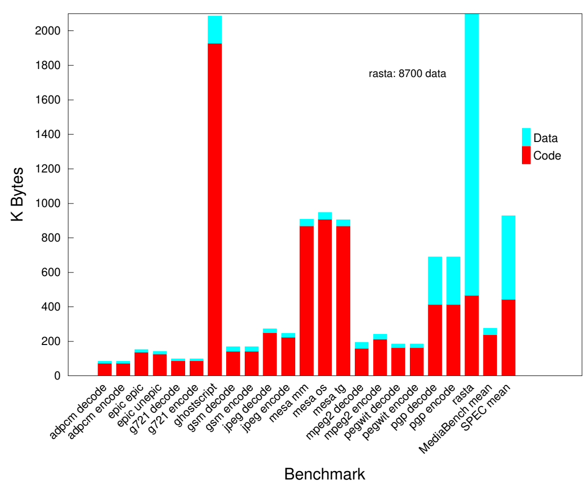

Figure 6: Data/Code static memory consumption.

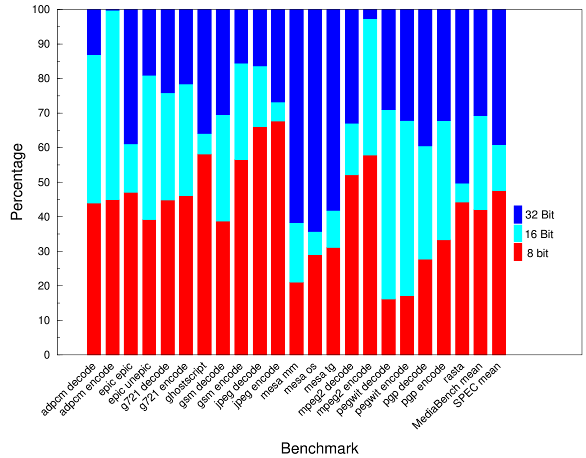

Figure 7: Integer bit utilization.

4 Conclusion

We have performed a detailed analysis of the MediaBench benchmark set. Using an extended version of the SimpleScalar simulator, we collected information about branch prediction accuracy, cache hit rates, memory usage, and integer bit utilization. This information was compared against mean SPECint95 results for the same simulation parameters. We found that branch prediction accuracy tended to be poor, due to unpredictable data dependent branches. Dynamic and static memory utilization was highly variable by benchmark. Additionally, most MediaBench benchmarks would benefit from reduced precision SIMD operations. Using the data contained in this paper, it is possible to better understand the hardware requirements of many popular multimedia applications.

References

[1] Lee, C., M. Potkonjak, and W. H. Mangione-Smith, "MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems," *Proc. 30th Annual International Symposium on Microarchitecture*, Dec. 1997, pp. 330–335.

[2] Burger, D., T. M. Austin, and S. Bennett, "Evaluating Future Microprocessors: The SimpleScalar Tool Set," Technical Report CS-TR-96-1308, University of Wisconsin-Madison, Jul. 1996.

[3] Burd, T., "Embedded Processor Information (16 & 32 bits)," `http://infopad.eecs.berkeley.edu/CIC/embed/local/`, Sep. 1996, accessed Apr. 1999.

[4] Smith, J. E., "A Study of Branch Prediction Strategies," *Proc. Eighth Symposium on Computer Architecture*, May 1981, pp. 135–148.

[5] `http://www.cse.psu.edu/~mdl/mediabench.tar.gz`