The DARPA Boolean equation benchmark on a reconfigurable computer

D. A. Buell and Sreesa Akella and James P. Davis and E. A. Michalski and Gang Quan
Department of Computer Science and Engineering
University of South Carolina
Columbia, South Carolina 29208
{buell|akella|jimdavis|michalsk|gquan}@cse.sc.edu

Abstract

The Defense Advanced Research Projects Agency has recently released a set of six discrete mathematics benchmarks that can be used to measure the performance of high productivity computing systems. Benchmark five requires matching a short bit string (with don't care positions) against a very long bit stream, setting up systems of linear equations with $0-1$ coefficients, and solving the systems using Gaussian elimination. We describe the implementation of this benchmark on the SRC Computers reconfigurable computer and present results on performance. Since this is a reconfigurable machine with Field Programmable Gate Arrays (FPGAs) that can be used as processing elements, the implementation has many features of a special purpose hardware design as well as the load balancing and data access problems inherent in a software implementation.

1. Introduction

The Defense Advanced Research Projects Agency has recently released a set of six discrete mathematics benchmarks that can be used to measure the performance of high productivity computing sytems [1]. These benchmarks are intended to augment the DARPA floating point benchmarks as well as standard performance guides such as LINPACK. Described briefly, the six benchmarks (numbered zero through five by DARPA) are

0. random access to a very large shared memory array;
1. matrix multiplication with multiprecise modular coefficients;
2. a dynamic programming problem;
3. transposition of bits in a bit stream;
4. integer sorting;
5. matching of a bit string with a bit stream and solution of a derived system of linear boolean equations.

Benchmark five requires matching a short bit string (including don't care bits) against a very long bit stream, setting up systems of linear equations with $0-1$ coefficients, and solving the systems using Gaussian elimination. We describe an implementation of this benchmark on the SRC Computers SRC-6e reconfigurable computer [2], and we report on this implementation. We will provide raw performance information, an analysis of the SRC-6e for this application, and a more general discussion of the issues of load balancing of data movement and computation for problems similar to this benchmark.

2. Benchmark Five

Let $\{s_i\}$ be a stream of bits of length L. We are to search for all occurrences of a bit pattern P where bits in P can be specified as 1, 0, or "don't care" bits. Let j be the position in the bit stream immediately after an occurrence of P.

Beginning with bit j, we form N equations in N unknowns over $GF(2)$, solve the system of equations, and output either the unique solution or the information that no solution exists.

The systems of equations should be set up and solved for every occurrence of P in the bit stream.

Specific parameters are given as examples for benchmark five.

- The length L of the input bitstream is 10^7.
- An example bit pattern is

$$P = 000???100??110?10?1111,$$

  where ? indicates a "don't care" bit.
- We take $N = 700$, and thus we are required for every substring match to set up and solve the 700×700

matrix equation

$$\begin{pmatrix} s_j & s_{j+1} & \cdots & s_{j+699} \\ s_{j+701} & s_{j+702} & \cdots & s_{j+1400} \\ \cdots & & & \\ s_{j+489999} & s_{j+490000} & \cdots & s_{j+490698} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_{700} \end{pmatrix} = \begin{pmatrix} s_{j+700} \\ s_{j+1401} \\ \cdots \\ s_{j+490699} \end{pmatrix}$$

3. Software Implementation

These benchmarks are new to DARPA as benchmarks, but they have been in use for some time. However, the available software implementations are rather dated, which makes it difficult to make comparisons against prior art.

To ensure that we are in fact solving the benchmark problem as posed, we have rewritten code in C for the benchmark for an Intel Pentium 4 processor. We began with a naive implementation that, although slow, was simple, so we could accept the answers as definitive. A second step was efficient C code for a standard Pentium. Unfortunately, lacking a highly optimized version in C or Fortran, we can not make a complete comparison of the implementation on the SRC-6e. However, we hope that over time there will develop a collection of results that will permit these results to be put into context.

4. Implementation on the SRC-6e

Subsequent to the implementations in C is a VHDL version as a reference for a hardware implementation. The VHDL version can be included as code for an initial implementation on the SRC-6e to provide a working program. The next step, C code for the MAP on the SRC-6e, is an ongoing project. This would permit direct synthesis of logic for the Xilinx FPGAs and a comparison of the MAP-C against VHDL.

5. Conclusions

Our implementation is not yet complete, so we cannot offer a full comparison at this time. However, we expect performance on this benchmark will reflect the fact that the SRC-6e permits configuring the FPGAs to perform efficient computations on the bit strings of Benchmark Five. Although the extant two software implementations can work around the pattern lengths and equation solving that does not fit word lengths, they do so at some cost in both time and memory. In contrast, however, the VHDL implementation represents direct execution on the hardware and seems to promise significant performance increase over conventional processors, regardless of how efficiently they have been programmed.

References

[1] Defense Advanced Research Projects Agency. High productivity computing systems discrete mathematics benchmarks, 2003.

[2] SRC Computers, Inc. Web site. `www.srccomp.com`.