Wavelet-based Image Compression on the Reconfigurable Computer ACE-V

Hagen Gädke and Andreas Koch

Tech. Univ. Braunschweig (E.I.S.), Mühlenpfordtstr. 23, D-38106 Braunschweig, Germany
koch@eis.cs.tu-bs.de

Abstract. Wavelet-based image compression has been suggested previously as a means to evaluate and compare both traditional and reconfigurable computers in terms of performance and resource requirements. We present a reconfigurable implementation of such an application that not only achieves a performance comparable to that of recent CPUs, but does so at a fraction of their power consumption.

1 Introduction

Accurately evaluating computer systems, both of the traditional and the reconfigurable kind, is not trivial. Too many characteristics can be measured in too many metrics. Implementations of benchmark applications are sometimes only subtly different, but no longer comparable (e.g., due to different quality of results).

To alleviate this, the Honeywell Benchmark Suite [1] uses so-called stressmarks to evaluate a broad spectrum of system characteristics. Each individual stressmark was developed specifically to test a subset of the interesting properties. All stressmarks are described by usage documents and sample implementations in C and sometimes also in VHDL. Minimum requirements on the quality of results support the comparability of measurements.

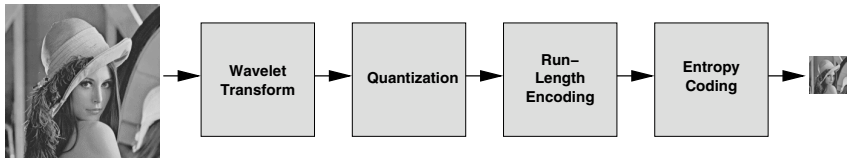
This paper examines an implementation of the versatility stressmark of the Honeywell suite. Since the VHDL reference code included in the Honeywell suite is not complete, the results described here can serve as baseline data for comparison with future realizations.

2 Versatility Stressmark

This application from the Honeywell suite [2] aims to evaluate how well the target hardware can perform several different functions using a single architecture. The stressmark is a wavelet-based compression algorithm [3] for square 8b gray scale images. Figure 1 shows the processing flow of the application.

While Honeywell includes a sample C implementation, the stressmark allows deviations. E.g., both true run-length encoding or zero length encoding may be used. For the entropy coding step, acceptable algorithms would include Huffman, Shannon-Fano and arithmetic coding.

Figure 1. Versatility structure



The "versatility" aspect of the stressmark considers the different implementation options for the algorithm. The choice we made is a so-called *high-complexity* implementation that executes all steps in hardware using a single configuration. Another angle not considered in the original stressmark definition is the different nature of the various stages: The wavelet transform and the quantization stage perform mostly arithmetic on multi-bit words, while the entropy coding step primarily performs bit-manipulations. With this mix, the capability of the target platform to handle these mixed computation styles can be evaluated.

The quality of results requirements for this stressmark are defined as minimum peak signal-to-noise ratios (specified in dB) at a maximum compressed bit rate (given in bits per pixel).

3 Stressmark Realization and Optimization

As in the sample C implementation, the wavelet transform itself is computed as a 3-step (2,2)-biorthogonal Cohen-Daubechies-Fouveau transform implemented in the Lifting Scheme [4].

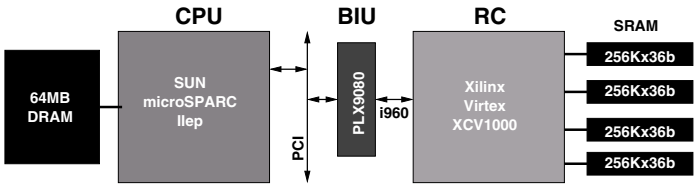
The first horizontal filtering pass processes 4 8b pixels simultaneously per clock, subsequent horizontal passes operate on 2 16b values simultaneously per clock (the width of the data expands during processing). All vertical passes process a single 16b value per clock. As in the sample C code, the three highest frequency blocks are assumed to contain only irrelevant (non-visible) details and are dropped entirely from further processing.

The quantization step can be started only *after* the minimum and maximum coefficient values of a block are known. Thus, a block can be quantized only after it has been completely wavelet-transformed. However, the quantization and both the following run-length and entropy encoding steps can be performed in a pipeline-parallel fashion.

While we have adhered to the C sample implementation of the stressmark to a large degree, some obvious inefficiencies were corrected beforehand (both in SW and HW):

- The memory requirements were reduced by sharing the memory for two buffers across all processing phases instead allocating dedicated areas.
- The Wavelet Transform fcdf22 does not explicitly copy the input data into a local buffer before the computation. Instead, the local copy is built and maintained on-the-fly during the calculation.

Figure 2. ACE-V architecture



Module	Slices	BlockRAMs
Wavelet	1075	
Quantization	1251	
Zero-Length Encoding	179	
Huffman Coding	258	3
Global Control & Muxing	1507	
Memory Streaming Engine (MARC) [6]	1971	6
Total	6241	9
Equivalent ASIC	270K Gates	

Table 1. Resource requirements

- The high-frequency coefficient blocks 7, 8, and 9 are discarded as early as possible and are neither stored nor processed further.
- Row-order traversal of memory is faster than column-order accesses, since our 32b system data bus can fetch four adjacent 8b pixels or two adjacent 16b wavelet coefficients in a single cycle.

4 Experimental Results

Our implementation of the application for the ACE-V platform (Figure 2, [5]) was formulated in RTL Verilog, synthesized using Synplicity Synplify 7.2.2 and mapped to the Virtex target using Xilinx ISE 5.2.03[1]. Table 1 lists the area requirements of the complete versatility stressmark.

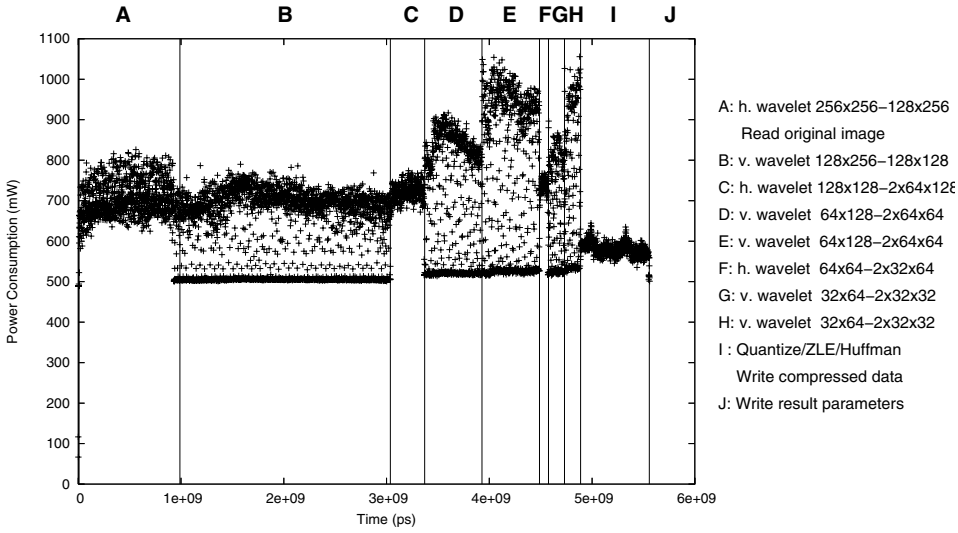
Table 2 gives an overview over the execution times of the stressmark when compressing 256x256 and 512x512 pixel Lena images on various platforms. In all cases, computation and in-memory data transfer operations were timed, but disk I/O was always omitted. Furthermore, for all reconfigurable platforms, configuration times were not included since our implementation does not need to reconfigure between processing phases. The last two lines in the table shows the clock frequencies achievable when targeting our design to recent 90nm commodity (Xilinx Spartan 3 series) and high-performance reconfigurable devices (Xilinx Virtex IIpro series). The hypothetical execution time on these RCUs is estimated by a simple scaling based on the increased clock frequency.

[1] Later versions of the tools have reproducible errors that lead to non-functional circuits.

Processor	Clock [MHz]	Execution Time [ms]		Power [W]
		256x256	512x512	
ACE-V RCU	30	6.6	17.5	1.1
Sun UltraSPARC III+	900	6.7	24	52.0
AMD Athlon	1333	6.0	131	63.0 . . . 70.0
AMD Athlon XP	1666	3.8	91	54.7 . . . 60.3
Xilinx XC3S1000-4 FPGA	*63*	*3.1*	*8.3*	*-?-*
Xilinx XC2VP20-7 FPGA	*105*	*1.9*	*5.0*	*-?-*

Table 2. Performance data

Figure 3. Power consumption profile for Virtex-based RCU



Even more interesting than the absolute performance data is the power consumption of the different processing units for the same task. For the traditional CPUs, the values are quoted from their data sheets. For the ACE-V Virtex 1000 RCU, the number shown is the peak power consumption as determined using the Xilinx XPWR power estimation program on a complete post-layout simulation trace (based on more than 30GB of data). Figure 3 shows the power consumption profile of the RCU with 1μs resolution over the entire simulated execution. Note the regular drops during the vertical processing phases, occurring when the end of a column has been reached and the read stream has to be reprogrammed and restarted. At those times, the wavelet transform units remain idle.

Table 3 shows the quality-of-results requirements from the original Versatility C implementation and the actual values achieved by our hardware when compressing im-

Image	b [bpp]	PSNR [dB]	
		Original	Hardware
Barbara	0.29	26.8	26.8
Goldhill	0.26	27.1	27.6
Lena	0.27	27.6	27.6

Table 3. Quality of results for $L = 256$ at default compression quality

ages with $L = 256$[2] at the stressmark's default quality value of $q = 128$. Our hardware implementation at least matches the software values. For the Goldhill image, an error in the reference software (loss of up to 7b after Huffman coding) that was corrected in the hardware version even improves the hardware-achieved quality.

5 Discussion and Conclusion

We have implemented the Versatility stressmark of the Honeywell suite on a Virtex-based adaptive computer system (ACS) and evaluated it in terms of resource requirements, performance and power consumption. While the ACE-V ACS with its slow 1998-vintage RCU (250nm process) is no longer competitive with more recent CISC CPUs, current reconfigurable devices will allow the realization of RCUs that reach or exceed CPU performance again. Even more promising is the low power consumption of the reconfigurable solutions. With the move to 90nm devices, higher power savings seem quite achievable.

References

1. Kumar, S., Pires, L., Ponnuswamy S., et al., "A Benchmark Suite for Evaluating Configurable Computing Systems - Status, Reflections, and Future Directions", *Proc. Eighth International Symposium on Field-Programmable Gate Arrays (FPGA)*, Monterey (USA), 2000

2. Honeywell Technology Center, "Versatility Stressmark", *Benchmark Specification Document Version 0.8 CDRL A001*, Minneapolis (USA), 1997

3. Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I.,"Image Coding using the Wavelet Transform", *IEEE Transactions on Image Processing*, No. 1, 1992

4. Uytterhoeven, G., Roose, D., Bultheel, A., "Wavelet Transforms using the Lifting Scheme", *Technical Report ITA-Wavelets Report WP 1.1, Katholieke Universiteit Leuven, Department of Computer Science*, Belgium, 1997

5. Koch, A., Golze, U., "A Comprehensive Prototyping Platform for Hardware-Software Codesign", *Proc. Workshop on Rapid Systems Prototyping*, Paris (F), 2000

6. Lange, H., Koch, A. "Memory Access Schemes for Configurable Processors", *Proc. Intl. Workshop on Field-Programmable Logic and Applications (FPL)*, Villach (AT), 2000

[2] Since the ACE-V has reliability issues performing memory transfers $>$ 64KB, no QoR data could be obtained on the actual HW for $L = 512$.