FPGA-Based VLIW Processor for Compression Techniques Using Verilog

Adil Hydari
Rutgers University
Edison, NJ, United States
ahh68@scarletmail.rutgers.edu

ABSTRACT

Very Long Instruction Word (VLIW) processors offer parallel execution by explicitly encoding instruction-level parallelism into the Computer Architecture. By leveraging Field-Programmable Gate Arrays (FPGAs) for a VLIW processor implementation I can provide high reconfigurability as well as a testbench to verify RTL code. This proposal outlines a plan to design and evaluate an FPGA-based VLIW processor architecture optimized for advanced compression techniques using Hardware-Design-Languages (HDL). The proposed architecture aims to enhance processing efficiency, reduce power consumption, and improve system performance when executing compression and decompression tasks. By baking in data compression instructions into the microarchitecture, I hope to propose an efficient method for lossless data compression/decompression in low-power embedded scenarios.

KEYWORDS

FPGA, VLIW Processor, Data Compression, Computer Architecture, Verilog, Chisel, Parallel Processing, Instruction Level Parallelism

ACM Reference Format:
Adil Hydari. 2024. FPGA-Based VLIW Processor for Compression Techniques Using Verilog. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Data compression is a critical component in modern computing systems, enabling efficient storage, transmission, and processing of large volumes of data. As applications ranging from multimedia streaming to big data analytics continue to grow, the demand for high-performance and energy-efficient compression techniques intensifies. Traditional processor architectures often struggle to balance the computational demands of compression algorithms with the need for low power consumption and high throughput.

Very Long Instruction Word (VLIW) processors, known for their ability to exploit instruction-level parallelism (ILP), present a promising avenue for optimizing compression tasks. Implementing a VLIW architecture on Field-Programmable Gate Arrays (FPGAs) harnesses the low-power consumption and reconfigurability inherent to FPGAs, perhaps enabling customized hardware acceleration for compression algorithms. This combination of VLIW and may achieve performance gains while maintaining energy efficiency.

The proposed architecture seeks to enhance processing efficiency, reduce power consumption, and improve overall system performance by integrating compression-aware features, optimized instruction sets, and leveraging the reconfigurable nature of FPGAs through an Hardware-Design-Language (HDL) implementation.

2 BACKGROUND AND MOTIVATION

2.1 FPGA-Based VLIW Processor Architecture

Compression algorithms often involve multiple independent operations such as bit manipulation, arithmetic calculations, and table lookups. VLIW architectures can execute these operations simultaneously by packing them into a single long instruction word. Further, the compiler in a VLIW system statically schedules instructions, determining which operations can be executed in parallel. This is especially effective for compression algorithms, where the data dependencies are well-understood and can be analyzed at compile time.

An FPGA implementation was chosen for the accessibility of HDL languages, wide breadth of simulators, and the eventual RTL verification on real FPGA hardware. Specifically, alternatives are not easily implemented without a large amount of work or high development costs. ASIC development and tapeout is often costly and has development lifecycle, however, the end product is more power efficient and more processing throughput. Simulation in System-C may also be an option, but the author is more familiar with Verilog and Chisel.

2.2 Data Compression Techniques

Modern compression algorithms, such as Entropy coding, Lempel-Ziv (LZ77/LZ78), and Predictive coding, vary in complexity and computational requirements. These algorithms often involve repetitive patterns, symbol frequency analysis, and bit-level manipulations, which can benefit from parallel processing capabilities offered by FPGA-based VLIW architectures.

2.3 Need for Optimization

While VLIW processors offer inherent parallelism, standard architectures may not be optimized for the specific operations required by compression algorithms. Tailoring the processor to handle common compression tasks on an FPGA may lead to substantial improvements in speed and energy efficiency. This necessitates architectural modifications, specialized instruction sets, and FPGA-specific optimizations that align with the operational characteristics of compression techniques.

3 RELATED WORK

Several studies have explored the optimization of processor architectures for data compression, including FPGA-based implementations. Previous work includes:

- **Superscalar Optimization:** Enhancing superscalar processors to better handle compression tasks through dynamic scheduling, speculative execution, and multiple issue [3].
- **ASICs for Compression:** Designing application-specific integrated circuits (ASICs) tailored for specific compression algorithms, achieving high efficiency but lacking flexibility [2].
- **VLIW in Multimedia:** Applying VLIW architectures on FPGAs to multimedia processing tasks, which share similarities with compression operations in terms of data parallelism [1].
- **Verilog-Based Processor Designs:** Implementing processor architectures in Verilog on FPGAs to exploit hardware-level parallelism and customization [4].

However, there is a gap in research specifically addressing the optimization of FPGA-based VLIW architectures for a broad range of compression techniques, which this proposal aims to fill.

4 PROPOSED WORK

This research will focus on the following objectives:

1. **Architectural Analysis:** Analyze the computational patterns of various compression algorithms to identify opportunities for parallelism and hardware optimization within an FPGA-based VLIW framework.
2. **Instruction Set Extension:** Develop specialized instruction sets tailored to common compression operations, enabling more efficient execution within the VLIW architecture implemented on FPGA.
3. **Hardware Modifications:** Modify the VLIW processor design to incorporate compression-aware features, such as dedicated functional units for bit manipulation and pattern recognition, leveraging FPGA resources effectively.
4. **HDL Implementation:** Implement the proposed VLIW architecture and its extensions in Verilog/Chisel, ensuring synthesis and optimization for FPGA deployment.
5. **Performance Evaluation:** Deploy the implemented architecture on FPGA platforms (e.g., Xilinx, Intel) and benchmark it against standard VLIW processors to assess improvements in speed, energy efficiency, and resource utilization.

5 METHODOLOGY

5.1 Algorithm Selection

Select a representative set of compression algorithms, including lossless (e.g., Huffman, LZ77) and lossy (e.g., JPEG, MPEG), to ensure comprehensive architectural optimization. Analyze these algorithms to identify parallelizable components and hardware acceleration opportunities.

5.2 Architectural Design

Design the modified VLIW architecture tailored for FPGA implementation, incorporating specialized functional units and extended instruction sets. Utilize Verilog for hardware description, focusing on optimizing data paths and leveraging FPGA-specific features such as Look-Up Tables (LUTs) and Block RAMs.

5.3 HDL Implementation

Develop the processor architecture in HDL, ensuring modularity and scalability. Implement compression-aware modules, such as dedicated bit manipulation units and pattern recognition blocks, to accelerate compression operations. Optimize the design for resource utilization and timing constraints specific to the target FPGA platform.

5.4 Simulation and Benchmarking

Use simulation tools (e.g., ModelSim, Vivado) to verify the correctness of the HDL implementation. Deploy the synthesized design on FPGA development boards (e.g., Xilinx Zynq, Intel Cyclone) and execute benchmark suites that include compression and decompression workloads. Measure performance metrics such as execution time, power consumption (using FPGA power estimation tools), and resource utilization (LUTs, Flip-Flops, DSPs).

5.5 Comparative Analysis

Compare the results with baseline FPGA-based VLIW architectures and other processor types (e.g., superscalar, SIMD) to evaluate the effectiveness of the optimizations. Analyze trade-offs between performance gains and resource utilization, and assess the scalability of the proposed architecture for different compression algorithms.

6 EXPECTED RESULTS

The research is expected to demonstrate that an FPGA-based VLIW processor optimized for compression techniques can achieve significant improvements in:

- **Performance:** Reduced execution time for compression and decompression tasks through enhanced parallelism and specialized instructions implemented in hardware.
- **Energy Efficiency:** Lower power consumption by minimizing redundant operations and leveraging FPGA's reconfigurable fabric for optimized hardware acceleration.
- **Resource Utilization:** Efficient use of FPGA resources (LUTs, DSPs, Block RAMs) by implementing compression-aware features without excessive area overhead.
- **Scalability:** Ability to efficiently handle a wide range of compression algorithms, providing a versatile solution for various applications.

Additionally, the proposed architecture should maintain compatibility with existing VLIW compilation tools through compiler enhancements, facilitating ease of integration and deployment.

7 FUTURE WORK

7.1 Compiler Development

Extend existing VLIW compilers to support the new instruction sets and optimize instruction scheduling for compression tasks.

Ensure that the compiler can efficiently map high-level compression operations to the specialized hardware units within the FPGA-based VLIW architecture.

8 CONCLUSION

Optimizing FPGA-based VLIW processor architectures for compression techniques presents a promising pathway to achieving high-performance and energy-efficient data processing. By integrating specialized instruction sets, hardware modifications, and leveraging the reconfigurable nature of FPGAs through HDL implementation, the proposal aims to enhance the capabilities of VLIW processors. This makes them well-suited for the growing demands of data-intensive applications. The outcomes of this research could pave the way for more versatile and efficient processing solutions in fields such as multimedia, telecommunications, and big data analytics.

ACKNOWLEDGMENTS

Professor Striki

REFERENCES

[1] Vincent Brost, Charles Meunier, Debyo Saptono, and Fan Yang. 2011. Flexible VLIW processor based on FPGA for real-time image processing. In *Proceedings of the 2011 Conference on Design Architectures for Signal Image Processing*. 1–8. https://doi.org/10.1109/DASIP.2011.6136855

[2] Song A. Guo C. Huang, Y. and Y. Yang. 2023. ASIC design of LZ77 compressor for computational storage drives. In *Electron. Lett.* IET, 59.

[3] Yeager K.C. 2002. The Mips R10000 superscalar microprocessor. *IEEE Micro* 16, 2 (2002), 28–41.

[4] Dilshan Kumarathunga, Omega Gamage, Asitha Samarasinghe, Nipuna Saranga, Ranga Rodrigo, and Ajith Pasqual. 2019. VLIW Based Runtime Reconfigurable Machine Vision Coprocessor Architecture for Edge Computing. 103–106. https://doi.org/10.1109/ASAP.2019.00-22