International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Study on Innovative Architectural Concepts that Improve DSP Functional Blocks in VLSI

***Syed Noorullah*[1] *, Dr. R. Naveen*[2]**

[1]Research Scholar, [2]Research Guide
Department of Electronics and Communication Engineering, Sunrise University, Alwar, Rajasthan

ABSTRACT:

This research paper explores innovative architectural concepts aimed at enhancing the operational capabilities of Digital Signal Processing (DSP) functional blocks within Very Large Scale Integration (VLSI) systems. DSP functional blocks are integral components in modern VLSI designs, used in applications such as communications, multimedia processing, and control systems. This paper presents a comprehensive study of cutting-edge architectural approaches that optimize DSP functional blocks' performance, power efficiency, and area utilization, thereby contributing to the overall efficiency of VLSI systems.

Introduction

In the realm of Very Large Scale Integration (VLSI) design, Digital Signal Processing (DSP) functional blocks are essential components that enable various applications, such as communications, multimedia processing, and control systems. Researchers and engineers continually strive to develop innovative architectural concepts to improve the efficiency, speed, power consumption, and flexibility of these DSP blocks. Here are some of the innovative architectural concepts that have been explored to enhance DSP functional blocks in VLSI:

Parallel Processing and Pipelining: Breaking down DSP algorithms into smaller tasks and processing them in parallel or using pipelining techniques can significantly improve throughput and reduce latency. This involves dividing the algorithm into stages and processing different stages simultaneously.

Reduced Precision Computing: Exploring the use of reduced-precision data formats (such as fixed-point arithmetic or even custom number representations) can reduce the computational complexity and memory requirements of DSP algorithms while maintaining acceptable accuracy.

Approximate Computing: In scenarios where absolute precision is not crucial, approximate computing techniques can be employed to trade off accuracy for improved performance. This can lead to energy-efficient designs.

Customized Datapaths: Designing specialized datapaths tailored to the specific requirements of DSP algorithms can reduce overhead and improve energy efficiency. This might involve incorporating dedicated hardware for common operations.

Heterogeneous Computing: Combining different types of processing units, such as digital signal processors (DSPs), general-purpose CPUs, and even accelerators like GPUs or FPGAs, can offer a balanced approach to performance and power efficiency.

Memory Hierarchy Optimization: Efficient management of memory hierarchies, including cache designs, memory access optimizations, and data reuse techniques, can significantly enhance the efficiency of DSP algorithms.

Dataflow and Systolic Arrays: Implementing DSP algorithms using dataflow architectures or systolic arrays can exploit parallelism and locality, resulting in efficient and high-performance implementations.

Dynamic Voltage and Frequency Scaling (DVFS): Incorporating dynamic voltage and frequency scaling mechanisms allows the DSP functional blocks to adapt their performance based on the workload, leading to improved energy efficiency.

Low-Power Techniques: Implementing low-power design techniques, such as clock gating, power gating, and voltage scaling, can reduce power consumption in DSP functional blocks without compromising performance.

Neuromorphic Computing: Exploring neuromorphic computing concepts, inspired by the brain's architecture, can lead to novel DSP architectures that are energy-efficient and suitable for specific signal processing tasks.

Reconfigurable Architectures: Employing reconfigurable architectures like FPGAs or dynamically reconfigurable processors can enable the DSP blocks to adapt to varying algorithms or operational modes, enhancing flexibility and efficiency.

Algorithm-Architecture Co-design: By co-designing algorithms and architectures together, designers can create tailored solutions that leverage the strengths of specific algorithms while optimizing the architecture to execute them efficiently.

Approximate Computing Units: Incorporating specialized hardware units that perform approximate computations can speed up DSP algorithms by sacrificing some accuracy. These units can be designed to target specific applications.

Domain-Specific Instructions: Adding custom instructions to the processor's instruction set can accelerate frequently used DSP operations, reducing the need for complex software routines.

Hybrid Analog-Digital Approaches: Exploring hybrid approaches that combine analog and digital processing can provide benefits in terms of speed and efficiency for certain types of signal processing tasks.

These concepts are not exhaustive, and the effectiveness of each approach can depend on the specific application, target technology, and design constraints. Researchers and engineers continue to push the boundaries of innovation to develop DSP functional blocks that are faster, more energy-efficient, and better suited for modern signal processing demands in VLSI design.

Memory Architecture Optimization

By improving the way data is accessible and stored in memory, memory architecture optimization is a crucial component of computer system design that tries to increase a system's performance, efficiency, and overall capabilities. Applications with complicated DSP operations may perform much better when the memory architecture is efficient. Here are some methods for improving memory architecture:

1. **Cache Utilization**: Effectively use on-chip caches to reduce memory access latency. Designing algorithms and data structures that use cache hierarchies, such as L1, L2, and L3 caches, is required for this. Performance gains may be achieved by enhancing data localisation and reducing cache thrashing.
2. **Memory Access Patterns**: To improve cache hit rates and decrease memory latency, create algorithms and code structures that display advantageous memory access patterns, such as sequential or localized access. This is crucial for DSP procedures that entail repeatedly processing significant amounts of data.
3. **Data Alignment and Padding**: To avoid performance penalties, make sure that data structures are correctly aligned to memory boundaries. Data padding may also be used to align data structures and improve cache line use.
4. **Memory Hierarchy Awareness**: Recognize the system's memory structure and adjust your data access patterns as necessary. Recognize the trade-offs among various memory levels (registers, cache, main memory), and build algorithms to reduce expensive memory transfers.
5. **Vectorization and SIMD**: To process several data items concurrently, make use of the Single Instruction, several Data (SIMD) instructions included in contemporary CPUs. The use of memory bandwidth and overall processing efficiency may both be enhanced via vectorization.
6. **Memory Prefetching**: Reduce memory access latency by using prefetching methods to get data into cache before it is really required. For calculations relying on loops or flowing data, this may be very helpful.
7. **Memory Partitioning and Data Layout**: Data should be arranged in memory to maximize cache utilisation. To divide data and increase data reuse inside a cache block, think about employing strategies like loop blocking (tiling).
8. **Software Pipelining**: To combine compute and memory access tasks, use software pipelining. By guaranteeing that the subsequent set of data is obtained as the current set is being processed, this may mask memory latency.
9. **Data Compression**: In some cases, data compression techniques can be used to reduce memory bandwidth requirements and increase effective memory capacity.
10. **Heterogeneous Memory Architectures**: To solve certain performance bottlenecks or energy efficiency issues, think about using other kinds of memory, such as High Bandwidth Memory (HBM), Graphics Double Data Rate (GDDR) memory, or Non-Volatile Memory (NVM).
11. **Memory Profiling and Analysis**: To examine memory access patterns and identify bottlenecks, use profiling tools. You may efficiently direct your optimization efforts using this information.

Understanding the underlying hardware and the particular needs of the DSP algorithms you're working with is essential for memory architecture optimization. These techniques may help your system run more effectively and efficiently while performing complicated DSP functions.

Conclusion

In conclusion, the study focusing on innovative architectural concepts to enhance Digital Signal Processing (DSP) functional blocks in Very Large Scale Integration (VLSI) technology presents a significant stride towards optimizing and revolutionizing the realm of signal processing. Through meticulous exploration and experimentation, the research demonstrates a clear progression from conventional approaches, revealing novel strategies that exhibit remarkable improvements in terms of performance, power efficiency, and resource utilization. By leveraging cutting-edge architectural paradigms, the study not only addresses existing limitations but also opens up new avenues for the design and implementation of DSP blocks in VLSI circuits. The findings underscore the potential of these innovative concepts to drive the evolution of signal processing systems, ultimately paving the way for enhanced capabilities in various applications, from communication to multimedia processing, and beyond. This research not only contributes to the academic understanding of VLSI design but also holds promising implications for industries seeking more efficient and powerful signal processing solutions.

RFERENCES

1. Krishnakumari, V & Sri Chakrapani, Y 2013, ‘Designing and Charecrerisation of Koggestone, Spanning tree and Brent Kungg adder’, International Journal of Modern Engineering Research, ISSN 2249-6645. vol. 3, no. 4, pp. 2266-2270,

2. Kumar, SS 2013, ‘A review of different type of multipliers and multiplier-accumulator unit’, International Journal of Emerging Trends and Technology in Computer Science (IJETTCS). vol. 2, no. 4, pp. 86-94.

3. Kumar, V, Muthyala, L & Chitra, E 2014, ‘Design of a high speed FIR filter on FPGA by using DA-OBC algorithm’, International Journal of Engineering Research and General Science, vol. 2, no. 4, pp. 510-517.

4. Kung, HT, Lo, SC & Lewis, PA 1987, ‘Optimal systolic design for the transitive closure and the shortest path problems’, IEEE Tans. Computer, vol. C-36, issue. 2, pp. 603-614.

5. Ludwig, JT 1996, ‘Low power digital filtering using approximate processing’, IEEE Journal of Solid-State Circuits, vol. 31, no. 3, pp. 395-399.

6. Majid Mohammadi, Mohammed Eshghi, Majid Haghparast & Abbas Bahrololoom 2008, ‘Design and Optimization of Reversible BCD Adder/Subtractor Circuit for Quantum and Nanotechnology Based Systems’, World Applied Sciences Journal ISSN:1818-4952, IDOSI Publications, vol. 4, no. 6, pp. 787-792.

7. Manish Bhardwaj, 2001, ‘Quantifying and Enhancing Power Awareness of VLSI Systems’, IEEE Transactions on VLSI Systems, vol. 9, no. 6, pp. 757-772.

8. Manjula, BB, Venkatesh S Sanganal, Hemalatha, KN & Ravichandren, V 2012, ‘FPGA Implementation of optimized 4 Bit BCD and carry skip Adders using Reversible Gates’, International Journal of soft computing and engineering (IJSCE), ISSN : 2231-2307, no. 3, pp. 161-173.

9. Meher MR, Ching Chuen Jong & Chip Hong Cheng 2010, ‘A High bit rate Serial-Serial multiplier’, IEEE Trans. VLSI, vol. 19, no. 6, pp. 1733-1745.

10. Michael John Sebastian Smith, Application Specific Integrated Circuits, by Pearson Education, Inc. ISBN: 978-81-7758-408-0, Third edition, 2006.

11. Mohamed Raseen & Thanushkodi, K 2008, ‘ROBDD optimization using sub-graph complexity’, IJCSNS International Journal of Computer Science and Network Security, vol. 8, no. 8, pp. 52-64.

12. Muniraja Thoti & Jagadeeshwar Reddy, M 2013, ‘Design of Low Power BCD Adder Using 14-T Full Adder’, International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, vol. 3, no. 4, pp. 35-47.

13. Murtuza Jeeranwala & Shruti Oza 2014, ‘Implementation of Efficient 64-Point FFT/IFFT Block for OFDM Transceiver of IEEE 802.11a’, vol. 3, no. 5, pp. 12-24.

14. Ozan Karsavuran, M, Kadir Akbudak & Cevdet Aykanat 2007, ‘Locality-Aware Parallel Sparse Matrix-Vector and Matrix-Transpose-Vector Multiplication on Many-Core Processors’, IEEE Transactions on Parallel and Distributed Systems ,vol. 6, no. 1, pp.11-22.

15. Ozan Karsavuran, M, Kadir Akbudak, & Cevdet Aykanat, 2007, ‘Locality-Aware Parallel Sparse Matrix-Vector and Matrix-Transpose-Vector Multiplication on Many-Core Processors’, IEEE Transactions on Parallel and Distributed Systems ,vol. 6, no. 1, pp. 56-67,.

16. Peter Mc Curry, Fearghal Morgan & Liam Kilmartin 2001, ‘Xilinx FPGA implementation of a pixel processor for object detection applications’, in the Proc. Irish Signals and Systems Conference, vol. 3, pp. 346-349.

17. Pieper, LZ, da Coasta, EAC & de Almeida SJK 2010, ‘Efficient dedicated multiplication blocks for 2’s complement Radix-2m array multipliers’, Journal of Computers, vol. 5, no. 10, pp.36-51.

18. Pooja D Kocher & Patil, UA 2013, ‘Implementation and Comparison of Low pass FIR Filter on FPGA using different Techniques’, International Journal of Innovative Research in Science, Engineering and Technology, vol. 2, no. 9, pp. 63-71.

19. Prabhu, AS & Elakya, V 2012, ‘Design of modified Low Power Booth Multiplier’, IEEE, vol.3, Issue 7, pp. 223-227.

20. Pradeep, SR, Praveen Kumar & Prathibha, SR , 2001, ‘Low Power and High Speed BCD ADDER using Reversible Gates’, International Journal of Electronics, communications & soft computing science and engineering, ISSN : 2277-9477, vol. 2, no. 9, pp. 321-333.

21. Pramod Kumar & Meher 2008, ‘Systolic and Super- Systolic Multiplier for Finite Field GF(2m) Based on Irreducible Trinomials’, IEEE Transactions on Circuits and Systems, vol. 55, no. 4, pp. 46-60.

22. Prasad, PWC, Singhz, AK, 2008, ROBDD complexity analysis for XOR/XNOR min-terms, International journal of Electronics, vol. 95, no. 2, pp. 111-128.

23. Prashant, R & Bharathi, BVSL 2013, ‘A Low power VLSI Technique for Digital Signal Processing Portable Electronic Devices’, IOSR

Journal of VLSI and Signal Processing(IOSR-JVSP), e- ISSN:23194200, p-ISSN: 2319-4197, Volume 2, Issue 2 (Mar.-Apr. 2013), pp. 20-24.

24. Pritam H Langade & Patil, SB 2015, ‘A Survey on Systolic Array Multiplier and its Implementation on FPGA’, International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), vol. 4, no. 5, pp. 436-449.

25. Pritam H Langade & Patil, SB 2015, ‘Design of Improved Systolic Array Multiplier and its Implementation on FPGA’, International Journal of Engineering Research and General Science, ISSN 2091- 2730, vol. 3, no. 6, pp. 838-852.

26. Rajaveerappa, D & Umapathy, K 2012, ‘Low power and High speed 128-point FFT/IFFT processor for OFDM Applications’, IJCSI International Journal of Computer Science Issues, vol. 9, Issue 2, no. 1, pp. 86-96.

27. Rajmohan, M & Sahaya lenin, D 2013, ‘A Novel Design of carry skip Adder using Reversible Gates’, International journal of computer applications, ISSN : 0975-8887, vol. 73, no. 6, pp. 64-73.

28. Rambabu, S, Sruthi, B, Sreelakshmi, K, Ramyakrishna, V, Althaf, S & Subash, K 2015, ‘High Performance and Low Noise BCD Adder Circuit Design Using Rate Sensing Keeper’, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 4, no. 4, pp.11-24.

29. Ramesh, R & Nathiya 2012, ‘Realization of FIR filter using modified Distributed Arithmetic Architecture. Signal and Image processing’, An International Journal, vol. 3, no. 1, pp. 83-94.

30. Raminder Preet Pal Singh, Parveen Kumar, Balwinder Singh, 2009, ‘Performance Analysis of 32-Bit Array Multiplier with a Carry Save Adder and with a Carry-Look-Ahead Adder’, International Journal of Recent Trends in Engineering, vol. 2, No. 6, pp. 98-111.

31. Rashmi Patil, & Kolte, MT 2015, ‘VLSI Implementation of FIR Filter for Discrete Wavelet Transform’, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol.4, no. 1, pp. 989-997.