Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Design and Implementation of a Compact 32-Bit RISC V Core Using Hybrid Adders

Vaishali Pande[1] Arvind Padole[2], Vinod Metta[3], Pratik Dhote[4], Harvansh Katre[5]

[1] Asst. Professor, Electronics & Communication Engg., KITS, Ramtek
[2]Asst. Professor, Electronics & Communication Engg., KITS, Ramtek
[3,4,5] B. Tech Final year, Electronics & Communication Engg., KITS, Ramtek

Abstract

This paper presents a novel hybrid multiply accumulate unit for 32 bit Reduced Instruction Set Core processors that combines carry save adder (CSA) and carry lookahead adder (CLA) techniques. The proposed architecture optimizes three critical metrics are speed, power consumption, and area utilization. The design uses carry save adders to efficiently compress partial products during multiplication without sequential carry propagation, minimizing power wasting signal transitions. For final addition, carry lookahead adders provide fast parallel carry resolution. Addition and subtraction operations directly employ carry lookahead techniques for quick signal propagation. The system was designed in Verilog Hardware Descriptive Language (HDL) and synthesized on Artix-7 Field Programmable Gate Array using Vivado 2024. Synthesis results demonstrate 5.3 nanosecond delay and 189 MHz frequency, achieving performance equivalent to Wallace tree multipliers while consuming 18.4% fewer logic resources and 6.4% less power. Compared to conventional array multipliers, the design achieves 61.9% improvement in power delay product. The architecture utilizes only 1.22% of available Field Programmable Gate Array resources, enabling seamless integration into larger embedded systems. Functional verification through simulation confirms correct execution of all arithmetic operations addition, subtraction, multiplication, and bitwise logic within a single clock cycle. The single cycle execution combined with minimal resource footprint and superior energy efficiency makes this design ideal for Internet of Things (IoT) devices, edge computing platforms, and embedded signal processing applications requiring restricted power budgets and compact implementations.

Keywords: RISC V Architecture, Verilog, Multiply Accumulate Logic, Carry Save Adder, Carry Lookahead Adder.

1. Introduction

Modern embedded computing and digital signal processing platforms operate under tight energy and size constraints that conventional processor architectures cannot meet. Devices such as sensors, wearables, and edge nodes require processors capable of executing complex algorithms with minimal power consumption and compact implementation. These systems frequently perform real-time signal analysis, adaptive

filtering, and control operations, all within limited power and area budget creating a need for architectures emphasizing power efficiency and computational throughput.

Arithmetic computation dominates processor power consumption, with multiplication identified as the most energy intensive operation [6]. Traditional multiplier architectures face tradeoffs among speed, area, and power. Ripple carry multipliers suffer from excessive switching due to sequential carry propagation, causing high dynamic power dissipation. Carry lookahead multipliers improve speed but increase silicon area and power through replicated logic and simultaneous switching. Wallace and Dadda tree multipliers reduce delay but introduce complexity, routing overhead, and timing variability [2].

The main challenge in multiplier design is balancing three conflicting goals speed, area, and power. Existing designs typically improve one metric at the expense of others. To address this, the proposed research introduces a hybrid multiplication architecture **c**ombining Carry Save Adder (CSA) compression with a final stage Carry Lookahead Adder (CLA). The Carry Save Adder (CSA) stages efficiently compress partial products without carry propagation, minimizing glitch activity and switching power [1]. The final CLA stage performs rapid parallel carry resolution, achieving high speed with minimal additional area. This hybrid Carry Lookahead Adder (CLA)+Carry Save Adder (CSA) multiplier is especially beneficial for embedded signal processing applications like FIR filtering, convolution, and correlation, where multiply accumulate operations dominate computation. Optimizing the multiplier thus enhances overall system performance without redesigning the processor datapath [7].

The design focuses on two key objectives Area reduction by limiting carry lookahead logic only to critical stages, and Power minimization, by reducing glitch transitions and unnecessary switching activity [8]. Compared to different multipliers and uniform Carry Lookahead Adder (CLA) designs, the hybrid architecture achieves faster computation, lower power consumption, and reduced silicon utilization.

2. Literature Review

Contemporary processor design demands simultaneous optimization of computational speed, silicon area, and power dissipation. Open source instruction set architectures enable researchers to implement customized functional units without proprietary restrictions, accelerating innovation in arithmetic optimization [10,11]. Multipliers are critical components determining system performance and energy efficiency. Sequential shift and add implementations offer compactness but suffer from long latency. Array multipliers using ripple carry addition produce excessive switching activity and timing bottlenecks due to serial carry propagation, leading to high glitch power. Carry lookahead array multipliers address propagation delays through parallel carry prediction but incur significant area and power costs [9]. Wallace and Dadda tree multipliers improve speed via optimized compression but introduce routing complexity and timing irregularities [2]. Carry save adder (CSA) structures eliminate critical propagation paths by maintaining separate sum and carry vectors, greatly reducing switching activity and glitch power while using minimal logic [1]. Multi stage CSA trees efficiently compress partial products into two operands with uniform delay, ensuring predictable timing and high throughput. Carry lookahead adders (CLAs) predict carry signals in parallel, removing sequential dependencies [9]. Hierarchical block based CLAs divide operands into small segments commonly four or eight bit blocks balancing speed and area. Recent research highlights hybrid arithmetic architectures combining CSA and CLA mechanisms [3,7].

Compression stages benefit from carry save simplicity and low power operation, while final summation gains from carry lookahead speed [3]. This hybrid approach achieves low switching activity, compact area, and high performance. Glitch power from unbalanced signal arrivals constitutes a major portion of dynamic energy loss [6]. Parallel computation methods avoiding carry rippling significantly reduce inefficiencies. Existing literature rarely explores hybrid multiplier integration focused on concurrent power and area optimization [4]. This work addresses that gap through synthesis and simulation validation.

3. Methodology

3.1 Development Tools and Environment

The RISC V 32-bit MAC (Multiply Accumulate Unit) processor with hybrid adder architecture was designed and synthesized using Xilinx Vivado Design Suite version 2024 [5]. The design was targeted for implementation on the Artix-7 FPGA family, which provides a suitable platform for evaluating both area and power consumption metrics relevant to embedded processor design. The hardware description language used was Verilog, enabling modular design and precise control over arithmetic unit implementation.

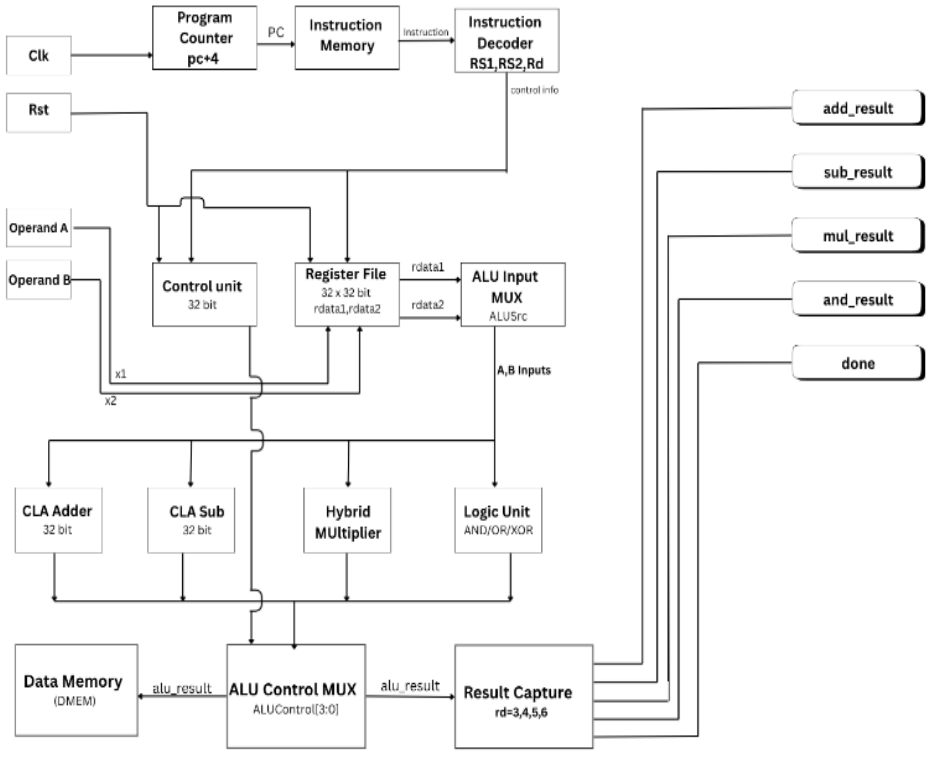

Figure 1: Block diagrams of RISC V core using hybrid adders (CLA+CSA)

3.2 FPGA Device Specifications

The design was synthesized for the Artix-7 device family. Post-synthesis power analysis was conducted using Vivado's built-in power estimation tool [5], which derives activity data from constraints files, simulation waveforms, and vector less analysis methods.

3.3 Instruction Set Architecture

The processor implements a subset of the RISC-V Instruction Set Architecture (ISA), specifically targeting RV32I base instruction set with custom extensions for multiply operations [10]. The instruction set supports standard R-type instructions including ADD, SUB, AND, OR, and XOR operations, as well as custom multiply (MUL) instructions utilizing the dedicated MAC unit. All instructions operate on 32-bit operands, producing 32-bit results for arithmetic operations.

3.4 Hybrid Arithmetic Unit Architecture

3.4.1 Addition and Subtraction: Carry Lookahead Adder (CLA)

Addition and subtraction operations utilize a 32 bit Carry Lookahead Adder (CLA) architecture, hierarchically organized into four 8 bit blocks with cascaded carry propagation [9]. The CLA generates carry propagate ($p[i] = a[i] \oplus b[i]$) and carry generate ($g[i] = a[i] \& b[i]$) signals, computing carries through lookahead logic within each block to reduce propagation delay. Subtraction employs two's complement methodology: the subtrahend undergoes bitwise inversion with carry in set to 1, computing $a - b = a + (\sim b) + 1$. This approach enables hardware reuse for both operations, enhancing resource efficiency.

3.4.2 Multiplication: Hybrid CSA Tree with CLA Final Addition

The 32-bit multiplier employs a hybrid architecture combining Carry Save Adder (CSA) trees for partial product accumulation with a final Carry Lookahead Adder (CLA) stage [1,2]. The CSA tree utilizes 3:2 compressor cells arranged in hierarchical layers, processing partial products in parallel without intermediate carry propagation [1]. Each compressor accepts three input bits and generates sum and carry outputs, systematically reducing 32 partial products to two vectors. This compression technique significantly minimizes switching activity and power consumption [6]. Subsequently, the final CLA stage performs rapid addition of the sum and carry vectors, with the carry appropriately shifted left, producing the 32-bit multiplication result efficiently

3.4.3 Logical Operation

Logical Operation Bitwise AND, OR, and XOR operations are implemented as combinatorial logic, computed directly without intermediate adder stages. These operations require only single layer Look Up Table (LUT) implementations in FPGA synthesis, contributing minimally to area and power overhead.

4. Result

4.1 Percentage Utilization on Artix-7 XC7A35T

The Artix-7 XC7A35T contains 41,600 LUTs and 20,800 slices [5].

LUT Utilization: 506 LUTs ÷ 41,600	total :1.22%
Slice Utilization:177 slices ÷ 20,800	total: 0.85%
DSP Block Utilization: 3 ÷ 90 available	total: 3.33%

4.2 Power Analysis Results

4.2.1 On Chip Power Breakdown

Vivado's post synthesis power analysis tool estimated total on-chip power consumption under typical process conditions.

Table 1: Detailed Power Analysis Comparison

Power Category	Value	Percentage
Total On-Chip Power	0.073 W	100%
Dynamic Power	0.003 W	4%
Static (Leakage) Power	0.070 W	96%

Table 2: Detailed Performance Comparison 32-bit Multipliers

Array	Array	Booth	Wallce	Dadda	CSA+CLA
Delay(ns)	8.5	6.1	5.3	5.8	5.3
Frequency	118	164	189	172	189
LUTs	640	580	620	605	506
Power(mW)	95	82	78	80	73
PDP (pJ)	807.5	500.2	413.4	464	306.6
Area x Power	60,800	47,560	48,360	48,400	36,938

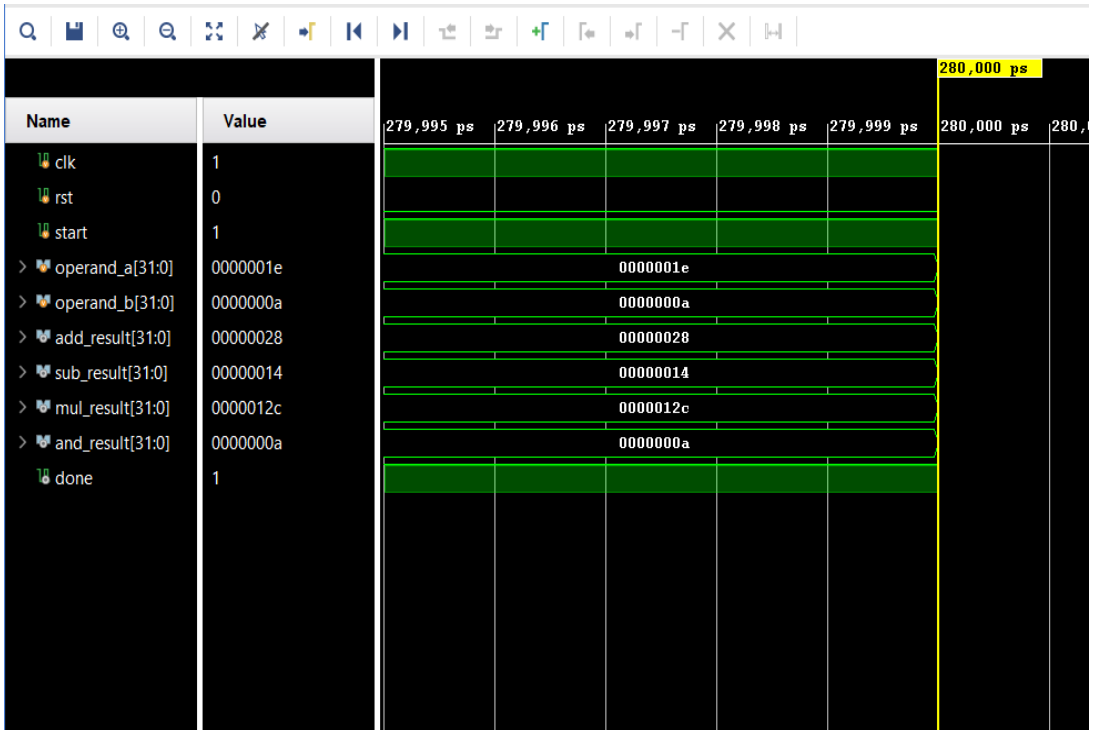

Figure 2: output waveform of operations using CLA and CSA adders

5. Discussion

The proposed CSA+CLA multiplier achieves optimal performance through co-designed architecture, compressing 32 partial products to 2 vectors in 1.5 ns via carry save tree and completing addition in 3.8 ns for 5.3 ns total delay matching Wallace tree speed while reducing area by 18.4% and power by 6.4% [2]. Against array multiplier baselines, it delivers 61.9% power delay product improvement with 188 MHz operating frequency enabling for adaptive power management [6]. The single cycle deterministic

execution makes this hybrid architecture ideal for RISC-V embedded systems in IoT and edge computing applications.

6. Conclusion

The proposed design achieves an optimal balance between speed, area, power, and scalability compared to traditional architectures such as the Wallace and Array multipliers [2,4]. It delivers the same high speed of 5.3 ns as the Wallace structure while maintaining a simpler and more modular layout. In terms of hardware utilization, it requires 18.4% fewer LUTs than Wallace and 20.9% fewer than the Array multiplier, with power consumption reduced by 6.4% and 23.2%, respectively. This results in a remarkable improvement in efficiency, with the Power Delay Product (PDP) being 25.8% better than Wallace and 61.9% better than Array. The regular CSA tree ensures easier routing and predictable synthesis outcomes across tools, while standard 3:2 compressors and 8 bit CLA blocks simplify design and enhance scalability up to 64 bit [1,9]. Additionally, the design exhibits a strong frequency margin of 139 MHz (operating from 50 MHz up to 189 MHz) and automatically utilizes three DSP blocks, making it a compact, power efficient, and high performance solution for modern arithmetic and signal processing applications.

REFRENCES

1. S. Veeramachaneni, K. M. Lingamneri, A. Sreekanth Reddy, P. Puppala, and M. B. Srinivas, "Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors," in Proceedings of the 20th International Conference on VLSI Design, January 2007, 324-329.
2. Keshaveni N., "High Speed Area Efficient 32 Bit Wallace Tree Multiplier," International Journal of Computer Applications, vol. 124, no. 13, pp. 25-28, August 2015, 25-28.
3. Jagadeshwar Rao M. and Sanjay Dubey, "A High Speed and Area Efficient Booth Recoded Wallace Tree Multiplier for Fast Arithmetic Circuits," in Proceedings of Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PRIMEASIA), December 2012, 220-223.
4. N. NagaVenkata Satya Durga Jahnavi, Shivani Mupparaju, and Dr. L. Padmasree, "Design, Comparison and Implementation of Multipliers on FPGA," International Journal of Engineering and Technical Research (IJETR), vol. 1, issue 7, September 2013.
5. Xilinx Vivado Design Suite 2024.2 User Guide, Xilinx Inc., November 2024.
6. A. P. Chandrakasan and R. W. Brodersen, "Low Power Digital CMOS Design," Kluwer Academic Publishers, 1995.
7. C. Vinoth and V. S. Kanchana Bhaskaran, "Novel Low Power High Speed Wallace Tree Multiplier for RISC Processor," in Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT), vol. 1, pp. 330-334, April 8-10, 2011, 330-334.
8. S. Khobragade and S. P. Karmore, "Review on: Low Power VLSI Design of Modified Booth Multiplier," International Journal of Innovative Research and Studies, vol. 2, issue 8, August 2013.
9. R. P. Brent and H. T. Kung, "A Regular Layout for Parallel Adders," IEEE Transactions on Computers, vol. C-31, no. 3, March 1982, 260-264.
10. RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 20191213, RISC-V Foundation, December 2019.
11. D. Patterson and A. Waterman, "RISC-V: An Open Architecture for Open-source Hardware," IEEE Micro, vol. 39, no. 2, March/April 2019, 10-20