

AMRITA SCHOOL OF ENGINEERING, BENGALURU

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

**B. TECH. VI SEMESTER AY 2024-2025**

**19ECE384 OPEN LAB**

**Report**

|  |  |
| --- | --- |
| **Registration numbers** | **Name of Students** |
| **BL.EN.U4ECE22021** | **E.Gokul** |
| **BL.EN.U4ECE22022** | **E.Sandeep Reddy** |
| **BL.EN.U4ECE22125** | **Lasya Godavari** |

**Submitted by**

FACULTY EXAMINER SIGNATURE

WITH EVALUATION DATE

INTRODUCTION

Design and Implementation of an Efficient Complex Binary Number System (CBNS) based Divider.  
  
PROBLEM DESCRIPTION

In modern digital systems, arithmetic operations involving complex numbers are essential, particularly in fields such as digital signal processing (DSP), wireless communication, image processing, and control systems. Among these operations, division is the most computationally expensive and hardware-intensive. Traditional binary and floating-point representations of complex numbers often involve separate handling of real and imaginary components, resulting in increased hardware complexity, higher power consumption, and longer computation times.  
The Complex Binary Number System (CBNS) has emerged as a promising number system where complex numbers are represented in binary form using a specific base (e.g., -1+j). This system eliminates the need to treat real and imaginary parts separately, potentially allowing more compact and parallelized hardware implementations. Despite its advantages, the implementation of efficient arithmetic units such as dividers in CBNS is still under active research and development due to challenges in algorithm design and hardware mapping.

MOTIVATION   
The motivation behind this work stems from the need to develop more efficient arithmetic units for complex number operations in hardware. As the demand for real-time, low-power computation grows—especially in portable and embedded devices—there is a pressing requirement for optimized architectures that can perform complex arithmetic directly without translating between multiple number systems.

CBNS provides an opportunity to simplify hardware circuits by representing complex numbers inherently, but current designs for complex division still rely on traditional methods or inefficient mappings that do not exploit CBNS’s full potential. Therefore, designing a hardware-efficient divider that operates natively in CBNS will contribute significantly to the performance and power efficiency of digital systems.

Relevance and Importance from a Hardware Perspective

From a hardware perspective, the division operation is known to consume more resources than addition, subtraction, or multiplication. In systems where complex number division is frequently performed—such as in FFT-based processors, MIMO systems, and modulator/demodulator circuits—having an optimized divider is crucial.

Implementing division directly in CBNS can result in:

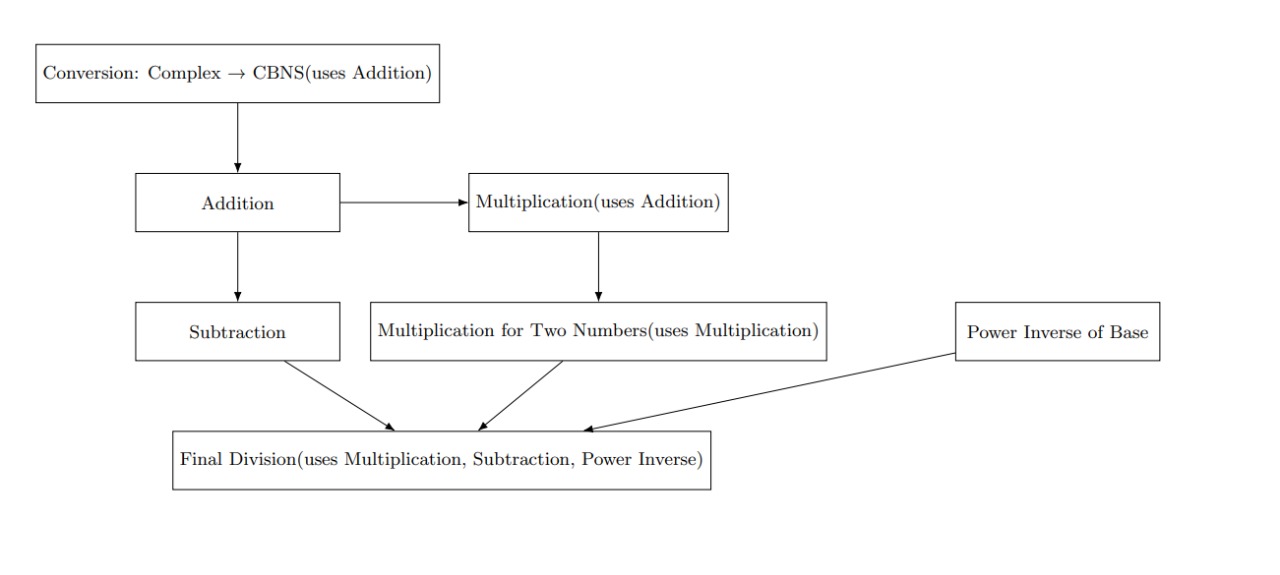
* Reduction in circuit area due to compact representation
* Lower power consumption through simplified logic
* Improved processing speed due to parallel computation capabilities
* Reduced latency in pipelines and arithmetic processing units

Thus, this project aims to design and implement a CBNS-based divider that is not only mathematically sound but also hardware-efficient, making it suitable for real-time applications where performance and power are critical constraints.

LITERATURE SURVEY:

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference** | Contribution | Advantages | Limitations |
| **Santosh, Sudia Sai, et al. (2021) "Complex Binary Number System-based Co-Processor Design for Signal Processing Applications."** | Proposed a CBNS-based co-processor for signal processing applications. Implemented modular CBNS arithmetic (addition, multiplication) on FPGA. | - Demonstrated reduced latency in CBNS arithmetic.- Proved CBNS is feasible for hardware co-processors.- Focus on power-efficient design. | - No specific focus on CBNS division.- Limited to basic arithmetic operations.- Did not address carry propagation challenges extensively. |
| **Jamil, Tariq, et al. (2021) "Nibble-size Multiplier Circuit Designs and their FPGA Implementations for Complex Binary Number System."** | Designed and implemented FPGA-based CBNS multipliers. Compared resource utilization & speed against traditional methods. | - Achieved better area-time efficiency.- Demonstrated scalability on FPGA platforms.- Optimized partial product handling in CBNS multiplication. | - Focused only on multipliers.- Did not provide insights into division algorithms.- Carry handling limited to multipliers. |
| **Jamil, Tariq & Al-Abri, S.S. (2010) "Design of a divider circuit for complex binary numbers."** | Proposed an early approach to CBNS division using an iterative algorithm for reciprocal approximation. Applied Newton-Raphson method for division. | - First to address CBNS division systematically.- Provided mathematical model for reciprocal computation.- Highlighted potential of Newton-Raphson in CBNS. | - High hardware complexity due to iterative nature.- Required multiple clock cycles for convergence.- No FPGA implementation or hardware optimization. |
| **Jamil, Tariq (2011) "An introduction to complex binary number system."** | Introduced CBNS fundamentals, carry propagation rules, and basic arithmetic rules. Established CBNS theory and representation. | - Foundation for all CBNS-based designs.- Defined addition, subtraction, and carry rules.- Enabled further exploration of CBNS-based systems. | - Theoretical focus, limited hardware implementation.- Did not address complex hardware challenges.- No division/multiplication focus. |
| **Das, Kaushik, et al. (2024) "Area–time–energy efficient architecture of CBNS‐based fast Fourier transform."** | Proposed an optimized CBNS-based FFT architecture. Focused on reducing area-time-energy product for real-time applications. | - Achieved significant reduction in energy consumption.- Optimized for high-speed DSP applications.- Demonstrated CBNS scalability in large systems. | - Focused on FFT, not arithmetic units.- Division challenges were not addressed.- Complex carry handling was not detailed. |

DESIGN



Feasibility

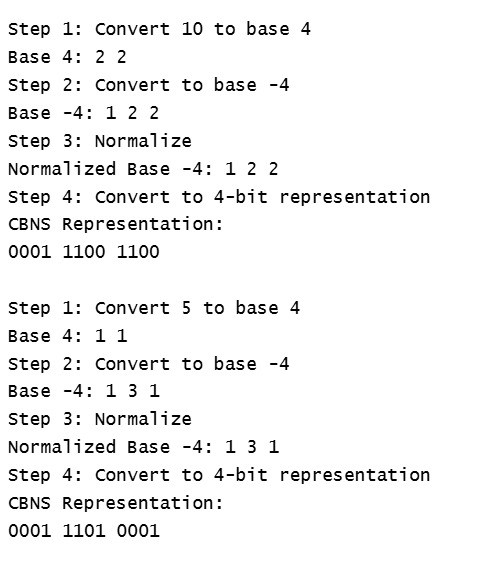
The implementation of complex number multiplication using the Complex Binary Number System (CBNS) is feasible and practical for digital hardware. CBNS enables compact representation of complex numbers, simplifying logic design and reducing hardware complexity. The successful realization of the multiplication unit confirms that CBNS-based arithmetic can be efficiently implemented using standard digital components.

Novelty

This approach is novel as it applies CBNS—a relatively unexplored number system—to hardware-based complex arithmetic. Unlike traditional methods, it handles complex numbers as a single entity, avoiding separate real and imaginary computations. This unique representation leads to simpler, faster, and potentially more power-efficient designs, setting a foundation for future development of a full CBNS-based divider.

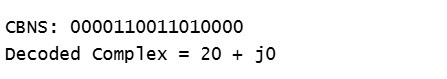
Simulation results

**Conversion of  number into CBNS format**



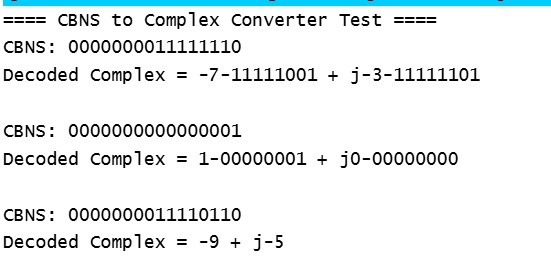
**Multiplication of two complex numbers**

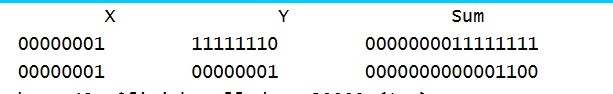


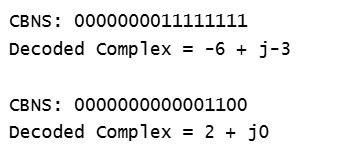
****

**Subtraction of two complex numbers**



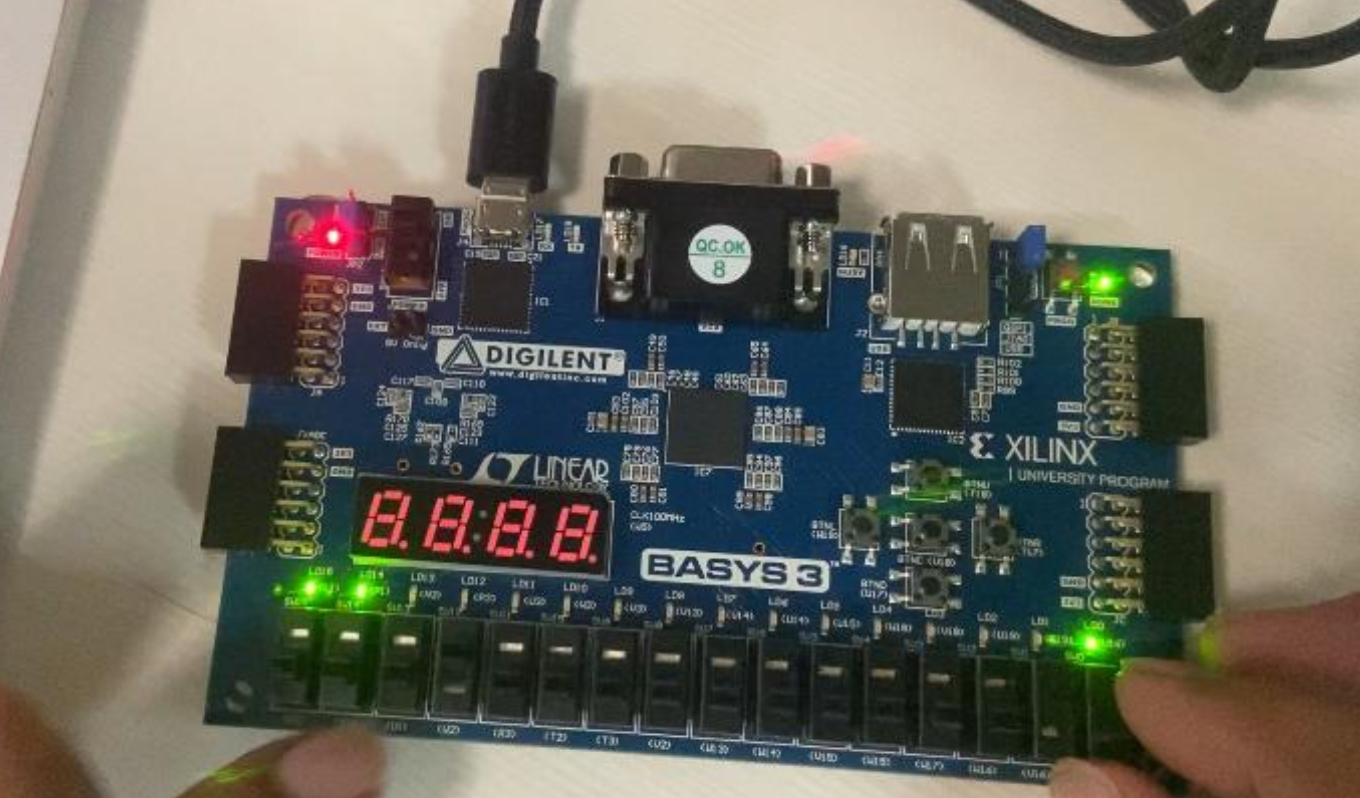


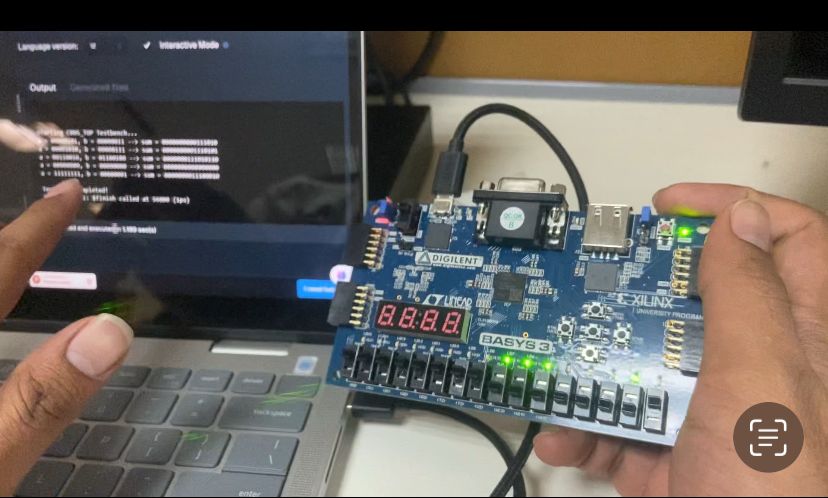


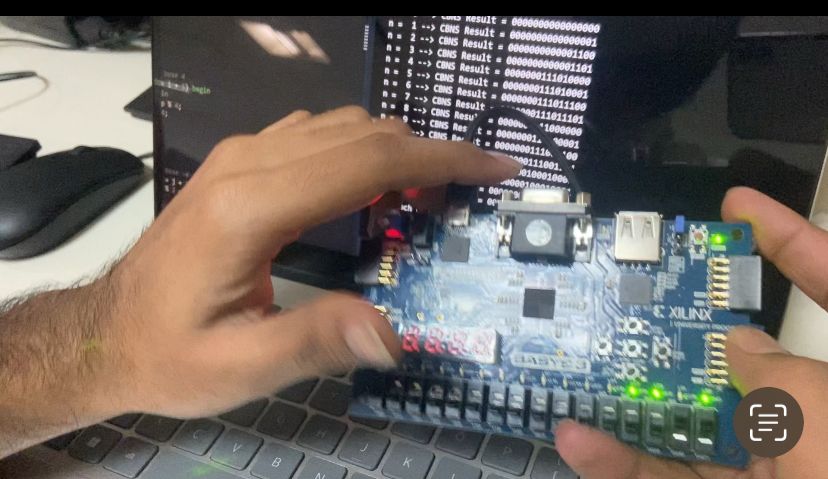


HARDWARE IMPLEMENTATION

|  |  |  |
| --- | --- | --- |
| COMPONENT | PURPOSE | AVAILABILTY /COST |
| BASYS 3 XILINX ARTIX-7 | Hardware implementation | Available in lab |
| PC (Laptop) | HDL Coding, simulation and synthesis | Personal |

Pictures of actual hardware implementation  
  


RESULTS  




CONCLUSION   
  
This report focused on the design and partial implementation of an efficient arithmetic unit based on the Complex Binary Number System (CBNS), with specific emphasis on multiplication. The motivation was to explore the potential of CBNS in simplifying complex number operations and reducing hardware complexity in digital systems.

The multiplication operation was successfully designed and implemented using the CBNS framework, demonstrating reduced hardware overhead and the possibility of parallel computation. The results suggest that CBNS offers a promising approach for complex arithmetic, especially in applications requiring high-speed and low-power computation.

Although the final goal is to implement a complete CBNS-based divider, this phase of the project lays a strong foundation by validating the multiplication unit and understanding the architecture necessary for extending the design to division. Future work will focus on developing and optimizing the division operation within CBNS to complete the arithmetic unit.

CHALLENGES FACED  
  
Challenges/Roadblocks Identified

1. Carry Overlapping in Addition
   1. Some numbers generate multiple overlapping carries, causing incorrect results.
   2. Needs recursive carry propagation and pattern recognition for efficient handling.
2. Borrow Handling in Subtraction
   1. Borrow propagation must align with CBNS rules.
3. CBNS Division Complexity
   1. Reciprocal approximation requires iterative convergence, adding computational load.
4. FPGA Resource Utilization
   1. Optimizing area and power for implementation remains a future challenge.

FUTURE SCOPE

* Integration of the CBNS divider into **larger DSP systems** (e.g., FFTs, filters, modulation).
* Extension of the architecture to support **floating-point complex arithmetic**.
* Application of **low-power optimization techniques** (e.g., clock gating, voltage scaling).
* **FPGA/ASIC implementation** for performance validation and real-world deployment.
* Exploration of CBNS arithmetic in **complex-valued neural network accelerators**.