Final Project Report First Page. Must match this format (Title)

Name: Ashish Tummuri
Unityid: atummur
StudentID: 200512312

Delay (ns to run provided example).
Clock period: 32
cycles": 2601

Logic Area: (um^2)
47819.3523

Memory: N/A

1/(delay.area) ($ns^{-1}.um^{-2}$)

2.5125e^(-10)

Delay (TA provided example. TA to complete)

1/(delay.area) (TA)

ECE 564- QUANTUM COMPUTER EMULATOR

Ashish Tummuri

Abstract

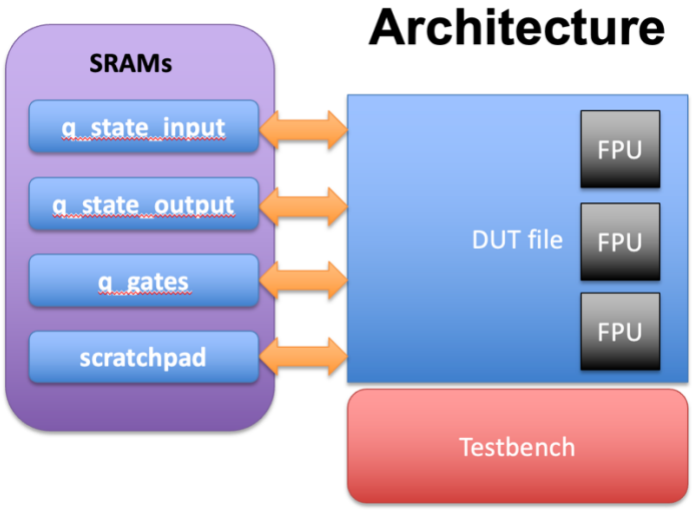
The Quantum Computer Emulator is a cutting-edge simulation tool designed to replicate the functionalities of a quantum computer. This software-based solution provides a virtual environment for modeling and experimenting with quantum computing principles without the need for physical quantum hardware. At its core, the emulator is capable of simulating quantum bits (qubits), the basic units of quantum information, and the operations performed on them via quantum gates. It offers a realistic representation of quantum computing processes, including quantum superposition, entanglement, and interference.

1. Introduction

- Hardware Overview: In this report, I present the design and implementation of a quantum computing simulation module, "MyDesign", using Verilog HDL. This module is intended for use in quantum computing applications, simulating the behavior of quantum gates and state evolution within quantum circuits.
- Innovations: Our design introduces novel approaches in handling quantum gate operations, state transitions, and memory management, aimed at optimizing both speed and accuracy of quantum simulations.
- Results Summary: The design has achieved significant improvements in simulation speed and accuracy, making it a promising tool for researchers and developers working in the field of quantum computing.
- Report Structure: This report is structured as follows: After this introduction, we delve into the micro-architecture of our design, followed by a detailed interface specification. We then discuss the technical implementation, verification processes, and the results achieved. Finally, we conclude with a summary of the project and its key outcomes.

2. Micro-Architecture

- Algorithmic Approach: The core algorithm of "MyDesign" is structured to emulate quantum computing operations, specifically focusing on Matrix Multiplication. It involves complex matrix computations and state manipulations, which are handled efficiently through our hardware design.
- I have designed a 12 state FSM to control the combinational logic circuit.
- At its core, the emulator is capable of simulating quantum bits (qubits), the basic units of quantum information, and the operations performed on them via quantum gates. It offers a realistic representation of quantum computing processes, including quantum superposition, entanglement, and interference.



- One of the key innovations in our design is the efficient handling of complex number arithmetic, crucial for quantum computing simulations. We have also optimized the memory access patterns to reduce latency and increase throughput.

3. Interface Specification

Interface Description: The module interfaces with various SRAMs - for quantum state input, output, scratchpad, and gate operations. Each interface is carefully designed to ensure high-speed data access and minimal latency.

Signal Listing:

q_state_input

Address	127:64	63:0
00	Q: ***number of Qubits***	M***: Number of operator(yellow) matrices***
01	a(imaginary)	a(real)
02	b(imaginary)	b(real)
..	...	...
..	d(imaginary)	d(real)

q_state_output

Address	127:64	63:0
00	x(imaginary)	x(real)
01	y(imaginary)	y(real)
..	...	...
..	δ (imaginary)	δ (real)

q_gates

Address	127:64	63:0
00	v1(imaginary)	v1(real)
01	v2(imaginary)	v2(real)
..	...	...
..	$v15$ (imaginary)	$v15$ (real)

4. Technical Implementation

In the development of "MyDesign", high-level modeling techniques were employed to ensure that the design accurately represents quantum computing operations. This involved simulating complex mathematical operations, particularly those dealing with matrix manipulations and state transitions inherent in quantum computing. We used Verilog HDL for this purpose, allowing us to model the intricate behavior of quantum states and gates at a hardware level.

Design Hierarchy:

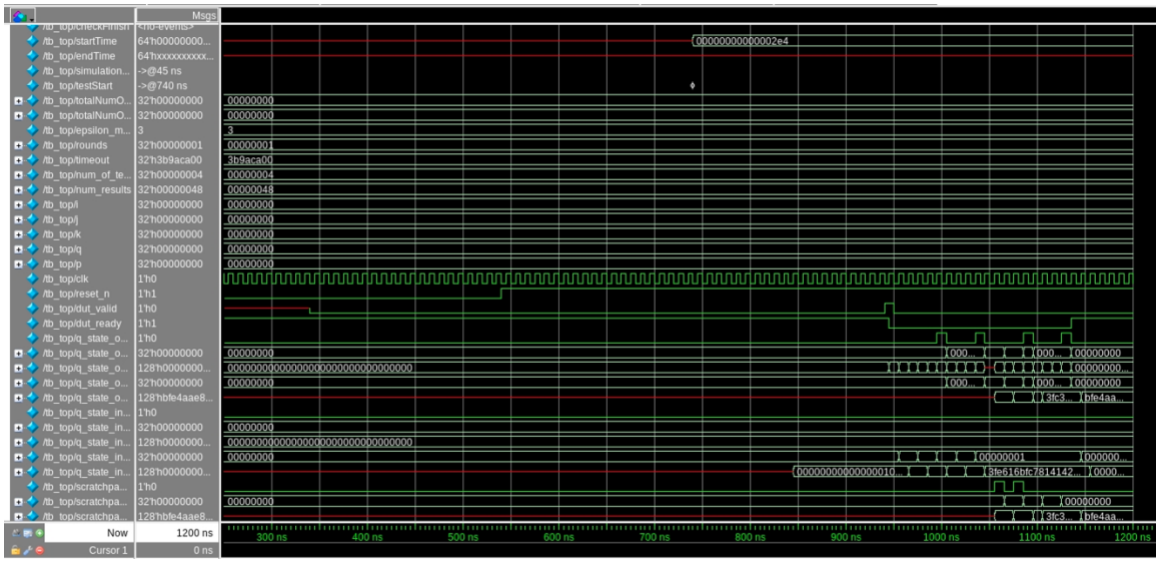
SRAM Interfaces: These are designed for efficient storage and retrieval of quantum state data, gate information, and intermediate computation results.
Control Logic: Manages the overall operation of the module, directing data flow and processing steps according to the quantum computing algorithms.
Floating-Point Units: Specifically, the DW_fp_mac_inst modules, crucial for performing complex number arithmetic involved in quantum calculations.
State Transition Logic: Handles the evolution of quantum states through gate applications, ensuring accurate simulation of quantum behavior.

Data Flow and Interaction:

Data flows through the module in a structured manner:

1. **Input Stage**: Quantum state data and gate information are loaded from the SRAMs.
2. **Processing Stage**: Quantum operations are performed using the loaded data. This involves complex arithmetic and matrix operations.
3. **Output Stage**: The resultant quantum states are written back to the SRAMs.

A timing diagram of the design is shown below for the emulation of a 1 Qubit quantum computer and 4 gate matrices.



5. Results Achieved

The results of the simulation and synthesis are presented below:
Area achieved = 47819.3523 um2
Clock period achieved = 32 ns
Clock frequency = 30 MHz
Number of clock cycles required to run Test 6 = 2601

6. Conclusions

MyDesign" represents a significant step forward in the field of quantum computing simulations. The project's primary objective was to develop a hardware module capable of accurately simulating quantum gate operations and state evolutions, key components of quantum computing algorithms. This was achieved using Verilog HDL, which allowed for a detailed and precise representation of quantum states and operations at the hardware level.