FPGA Based Emulation of Quantum Algorithms

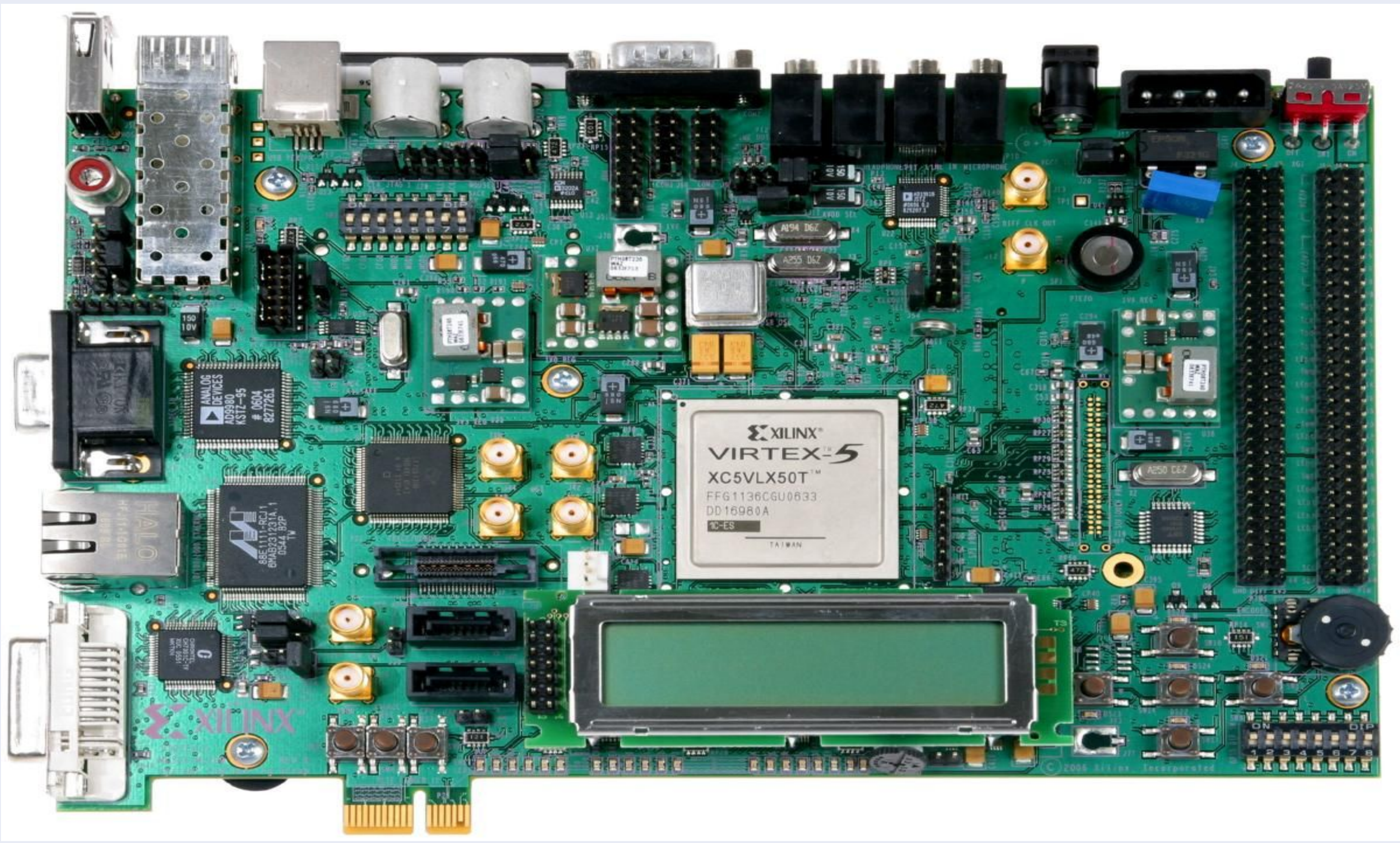
Chitresh, Milind, Abhinav

IIT KANPUR

Abstract

This project aims to emulate the behavior of a real quantum system, capable of running quantum algorithms on FPGA.

Quantum Fourier transform and Grover's search are chosen to be implemented in this project since they are the core of many useful quantum algorithms.



Hardware Used:

1) FPGA Board XUPV5 LX110T
2) USB JTAG

Methodology

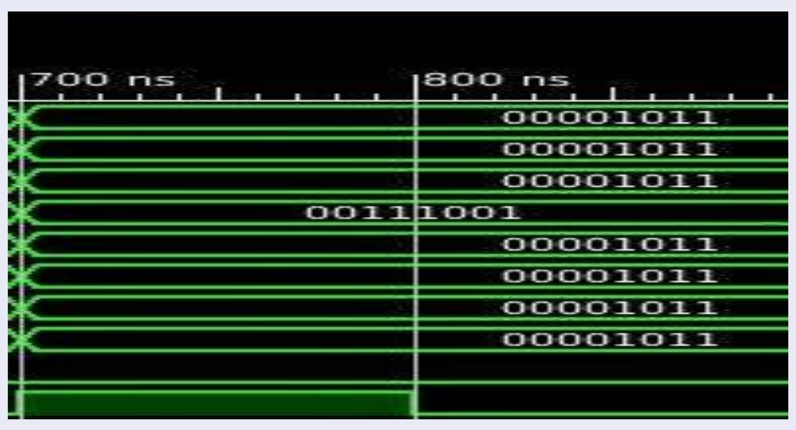
- **Grover Search:**

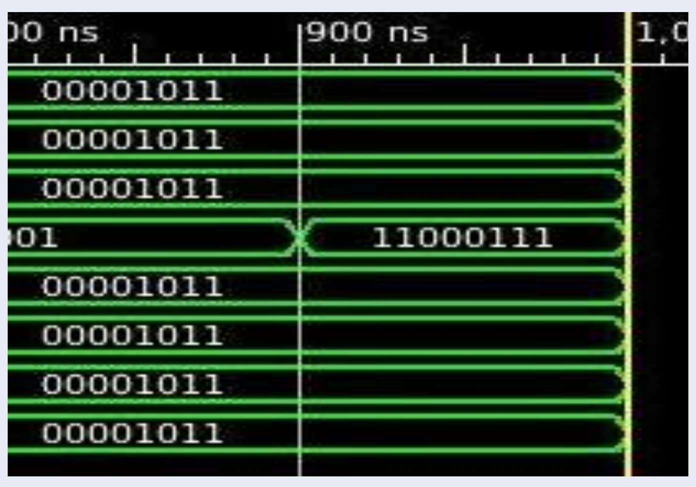
We utilized the computational resources available on FPGA such as comparators, adders/ subtractors, and multiplexers for efficient emulation.

Step 1:Phase Inversion

In this step phase of target element is inverted and all other phase are kept as it is using oracle

$U_f(x) = x*(-1)^{f(x)}$





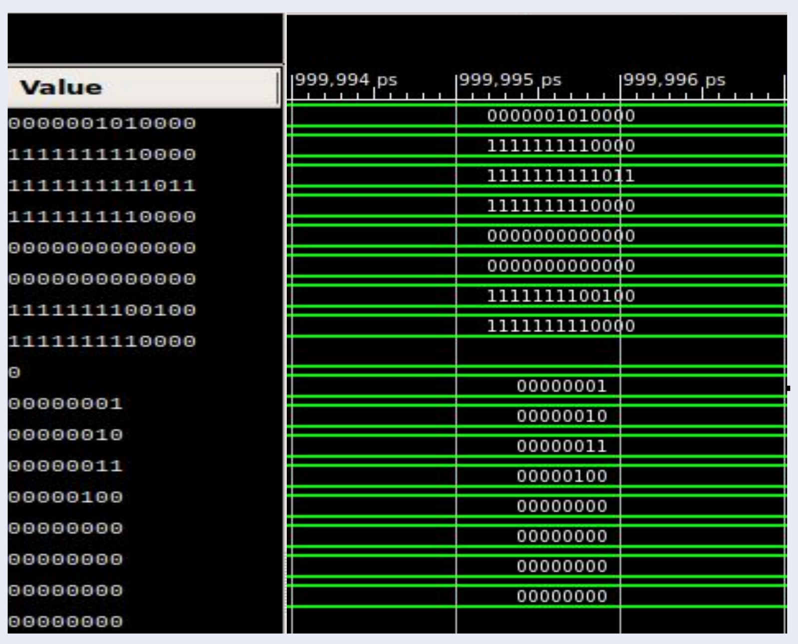
Step 2: Inversion about Mean

This results in amplitude amplification of desired element while shrinking the amplitude of other elements..This process is repeated sqrt(N) times. Here isolated number is our Target.

- **Quantum Fourier Transform:**

Quantum Fourier transform can be viewed as a unitary matrix acting on quantum state vectors.

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \omega_n^3 & \cdots & \omega_n^{N-1} \\ 1 & \omega_n^2 & \omega_n^4 & \omega_n^6 & \cdots & \omega_n^{2(N-1)} \\ 1 & \omega_n^3 & \omega_n^6 & \omega_n^9 & \cdots & \omega_n^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega_n^{N-1} & \omega_n^{2(N-1)} & \omega_n^{3(N-1)} & \cdots & \omega_n^{(N-1)(N-1)} \end{bmatrix}.$$



We did Tensor multiplication of above vector with quantum state vectors. Adjoint screenshots shows over state vectors of 3 Qubits.

Results

- **Grover Search:** We were able to implement Grover's Search algorithm on FPGA. We were able to maintain the natural time complexities of the algorithm. We compared the time required for the simulation in C and emulation in Verilog .

No of Qubits	Time for simulation in C	Time for simulation in Verilog
2	$1.077 * 10^{-3}$ s	$3.916 * 10^{-8}$ s
3	$2.517 * 10^{-3}$ s	$8.514 * 10^{-8}$ s
4	$5.415 * 10^{-3}$ s	$14.00 * 10^{-8}$ s

- **Quantum Fourier Transform:**

We also simulates Quantum Fourier Transform in Verilog and compared the corresponding times required for simulation in C and emulation in Verilog.

No of Qubits	Time for simulation in C	Time for simulation in Verilog
2	$0.966 * 10^{-3}$ s	$4.16 * 10^{-7}$ s
3	$1.161 * 10^{-3}$ s	$7.414 * 10^{-7}$ s
4	$2.566 * 10^{-3}$ s	$11.236 * 10^{-7}$ s

Conclusion

- We learned Verilog HDL. Using knowledge of verilog we synthesized and simulated adders, 8-bit comparator, LCD driver, finite state machine.
- We also learned basic microprocessor such as 8050.
- We learned quantum algorithms such as grover search algorithms and quantum fourier transform.
- From our observation we found that verilog simulation is nearly 10^5 times fast for grover Search and 10^4 times fast for QFT as compared to C.

Future Scope

Quantum algorithms can be used for solving many problems much faster than classical counterparts. The algorithms implemented in this project are the building blocks of much larger algorithms used for solving various complex tasks. We look for the implementation of these complex algorithms in the future.

We would also like to implement the basic algorithms for larger number of qubits. With the work progressing in the field of Quantum Computing we would like to find more efficient methods for FPGA based emulation of these algorithms.