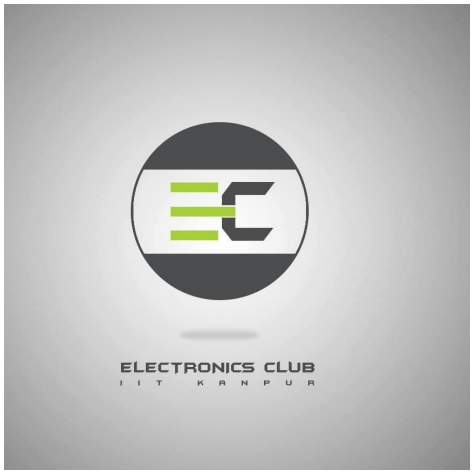

Electronics Club
IIT Kanpur

FPGA based Emulation of Quantum Algorithms

Final Documentation

Team Members:

- Milind Nakul.
- Chitresh Chaudhari.
- Abhinav Sharma.

Project Mentor:

- Apoorva Nandan

INDEX :

1.Acknowledgement

We are grateful to the Electronics club and Science & Technology council IIT Kanpur for providing us with the resources and motivation for this summer project. We would like to thank our mentors and co-ordinators of Electronics club:

- Apoorva Nandan
- Soumya Ranjan Dash
- Jay Mundra
- Mudit Agrawal
- Nitish Deshpande

to guide us throughout the project.

2.Introduction

This project aims to emulate the behavior of a real quantum system, capable of running quantum algorithms on FPGA(Field Programmable Gate Array) while maintaining their natural time complexity.

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. Physical realization of a quantum computer is proving to be extremely challenging. Only small scale small-scale quantum computation implementations have been achieved. This project aims to emulate these small scale quantum computations because of the significant improvements in time over classical computations.

3.Theoretical Background

In general, quantum algorithms obey the basic process flow structure. The computation process begins with a system set in a specific quantum state, which is then converted into superposition of multiple basis states. Unitary transformations are performed on the quantum state according to the required operations of the algorithm. Finally, measurement is carried out, resulting in the qubits collapsing into classical bits. While a classical bit is represented by an electrical voltage value or a wire current value, a quantum bit can take any linear combination of the two basic states 0 and 1 called a superposition state.

The computation process begins with a system set in a specific quantum state, which is then converted into superposition of multiple basis states. Unitary

transformations are performed on the quantum state according to the required operations of the algorithm. Finally, measurement is carried out, resulting in the qubits collapsing into classical bits. It can be shown that if quantum algorithms run on quantum computers, their processing speeds improve exponentially compared to their classical counterparts. However, due to the lack of quantum computers, circuit models of quantum algorithms are currently simulated using classical computers to verify their functionalities.

On the other hand, software simulation cannot use the intrinsic parallelism of quantum algorithms efficiently. To address the problem, the hardware emulation of quantum algorithms is considered here. To emulate quantum algorithms using FPGAs, a new representation for quantum bits is emulated that improves the emulation of quantum circuits considerably.

- **Quantum algorithms considered:**

Grover's algorithm is a quantum algorithm that finds with high probability the unique input to a black box function that produces a particular output value, using just O(sqrt N) evaluations of the function, where N is the size of the function's domain.

The Quantum Fourier transform is the classical discrete Fourier transform applied to the vector of amplitudes of a quantum state, where we usually consider vectors of length N equals powers of 2.

A quantum system which performs a specific operation on a selected set of qubits in a fixed period of time is called a quantum gate. It can be shown that the behavior of a quantum gate is always linear. Therefore, a quantum gate can be represented by a unitary matrix. Various quantum gates with different functionalities can be applied on superposition states to carry out quantum algorithms. FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be wired together, like many logic gates that can be inter-wired in different configurations. Complex logic blocks can be configured and then wired together to perform a variety of tasks. The FPGA configuration is generally specified using a hardware description language (HDL).

4.Steps Involved

- Learning to code in Verilog.
- Implementing basic circuits such as adders, multipliers,finite state ma- chines etc in Verilog.
- Learning about Grover's Search Algorithm and Quantum Fourier Trans- form.
- Implementation of these algorithms in C.

- Simulation of these algorithms in Verilog maintaining their natural time complexity.
- Emulation of these algorithms on FPGA.
- Comparison of time required for simulation in C and simulation in Verilog.

5.Verilog

Verilog is a hardware description language used for describing electronic circuits and signals.For emulating Quantum algorithms we need to write synthesizable verilog code for a piece of hardware. And hence detailed verilog knowledge, with its implementation on synthesis is required. Design experience on Xilinx will be useful while implementing Quantum algorithms on FPGA.

6.Our Approach

We want to emulate quantum algorithms using classical components. But the classical circuits cannot deal with parallelism in quantum circuits. Our approach is to use similarities in the quantum and classical circuits to design emulators of quantum algorithms on FPGA boards. We aim to emulate Grover's Search algorithm and Quantum Fourier Transform(referred to as QFT) using this approach. Quantum circuits are one convenient way of describing quantum algorithms. Such circuits comprise of analogues to digital bits and gates.

6.1 Modelling QFT for FPGA emulation

We will try to emulate QFT on FPGA using its mathematical model. The corresponding quantum circuits of the algorithm is shown in figure 1:

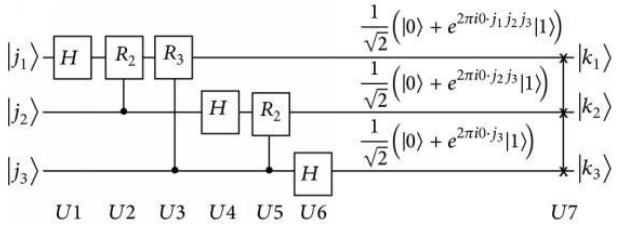

figure 1: Quantum Circuit of 3-bit QFT.

This circuits consists of Hadamard gates, Swap gates and the controlled

phase-shift gates.In order to model the effect of superposition and entanglement, derivation of each unitary transformation is made through the tensor product of individual quantum gate and identity matrix to form unitary matrix of equal dimension with the quantum state vector.

The matrix corresponding to QFT is as shown in figure 2:

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

Figure 2: Matrix for QFT

Here elements of the matrix are the nth root of unity.Since these quantum unitary matrices are mostly sparse matrices, we extract minimal number of useful arithmetic operations (due to nonzero elements in the matrix), resulting in an optimal realization of the model that can be mapped to an efficient FPGA emulation.

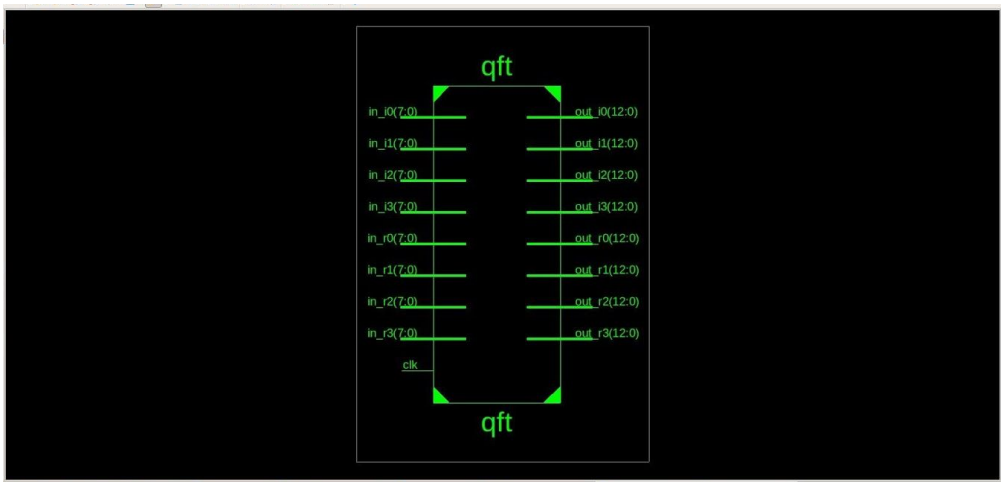

Figure 3: Schematic for QFT

6.2 Modelling Grover's Search for FPGA Emulation

We will try to emulate Grover's Search on FPGA using arithmetic operations.

We will utilize the computational resources available on FPGA such as comparators, adders/subtractors, and multiplexers for efficient emulation. This method mainly consists of phase inversion followed by inversion about the mean. Grover's search for a database with n elements mainly involves the processes of inverting the phase of target element, and performing inversion about mean on all elements. The arithmetic functions of the inversion about mean are derived through straightforward computations that involve summation, bit shift, and subtraction.

Mathematical modelling of Grover's Search is shown in figure 4.

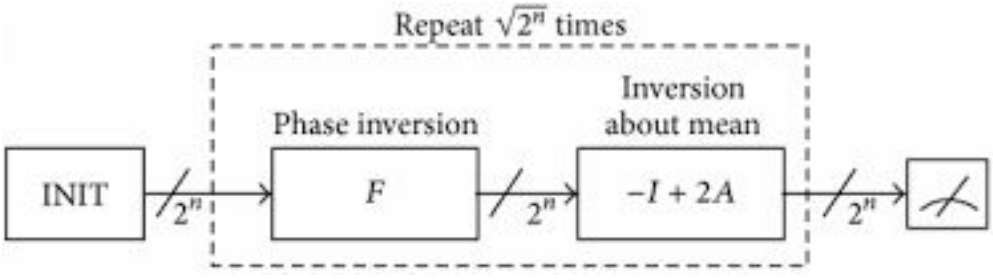

Figure 4: Mathematical Modelling of Grover's Search

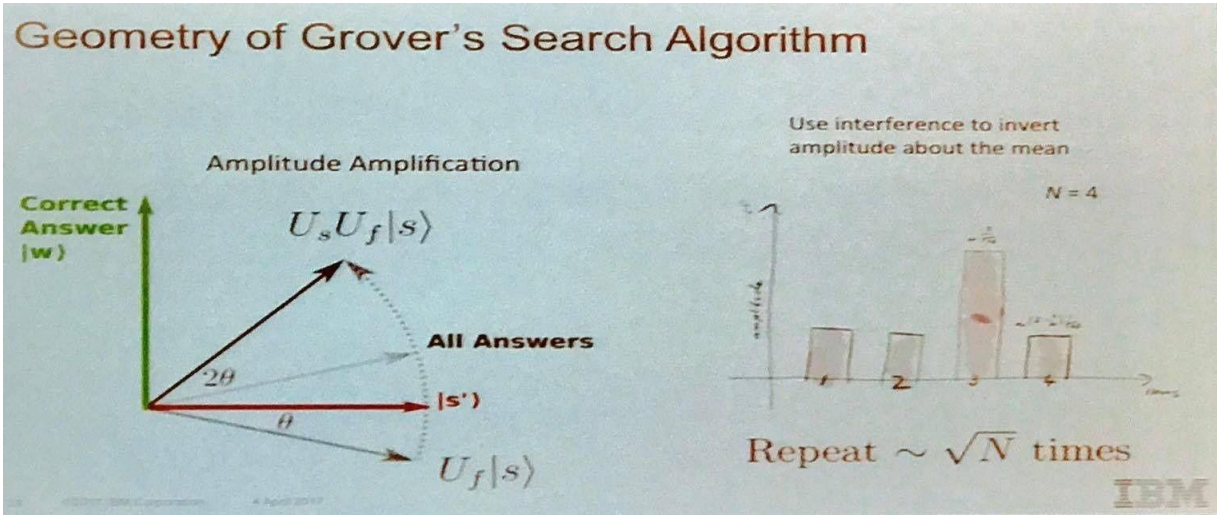

Figure 5: Basic Grover's Search

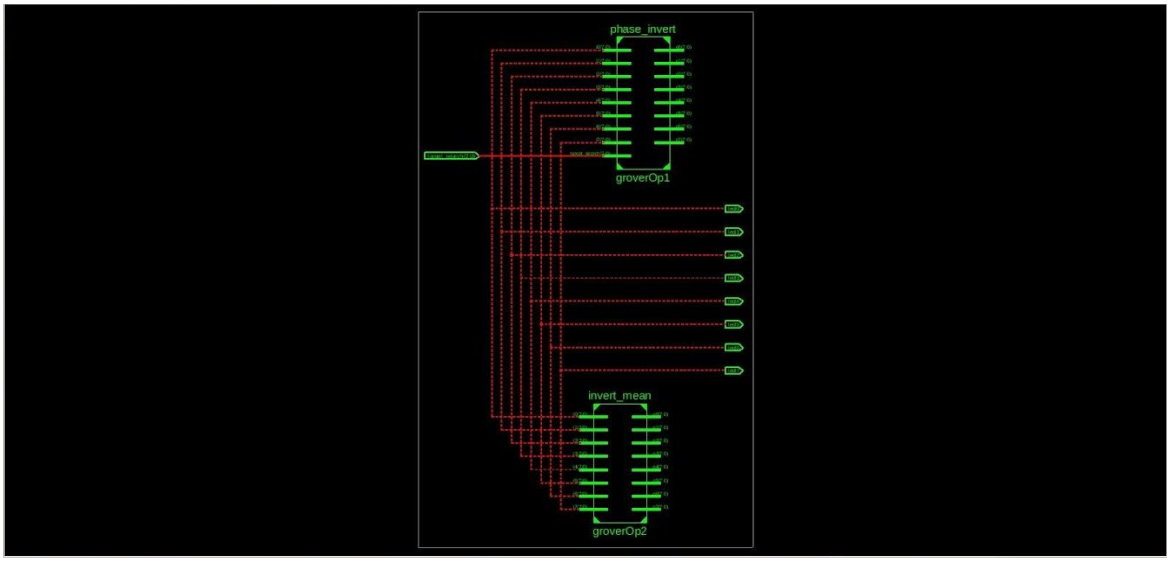

Figure 6: Schematics for Grover's Search

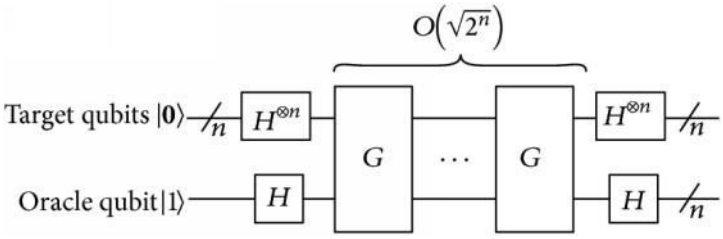

Figure 7: Quantum Circuit for Grover's Search

7.Simulation Results in Verilog

7.1 Grover's Search:

The simulation results for Grover's Search Algorithm is shown in figure 8, 9 and 10 :

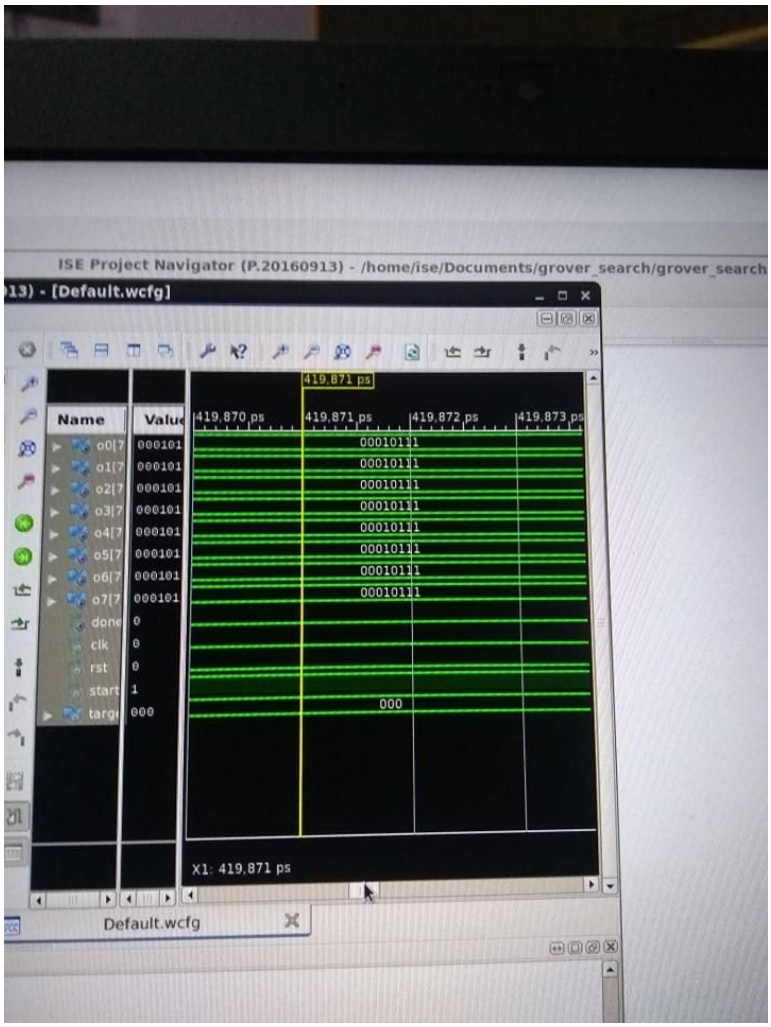

Figure 8: Starting with equal amplitudes for all elements

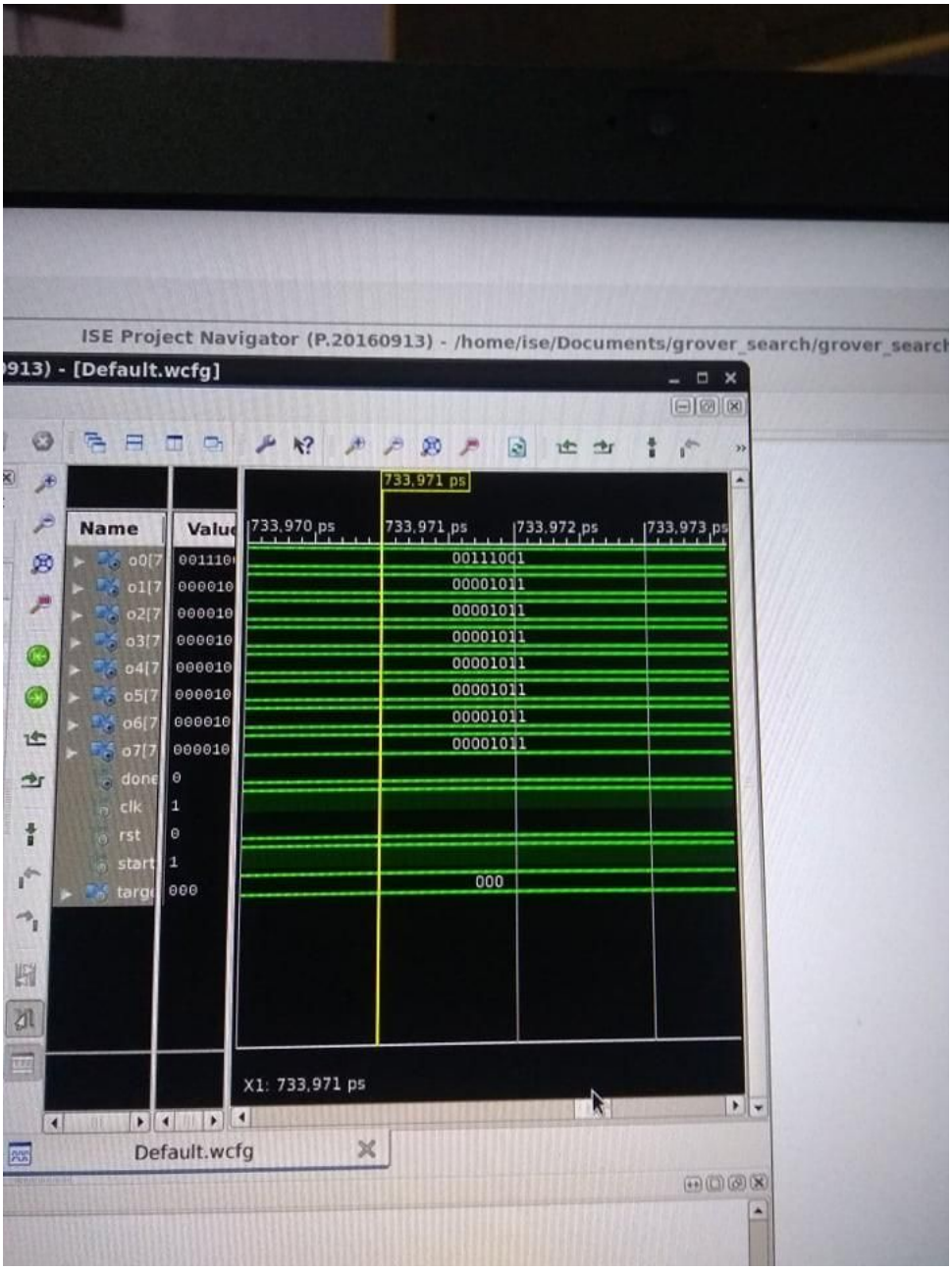

Figure 9: Amplitudes of elements after first iteration.

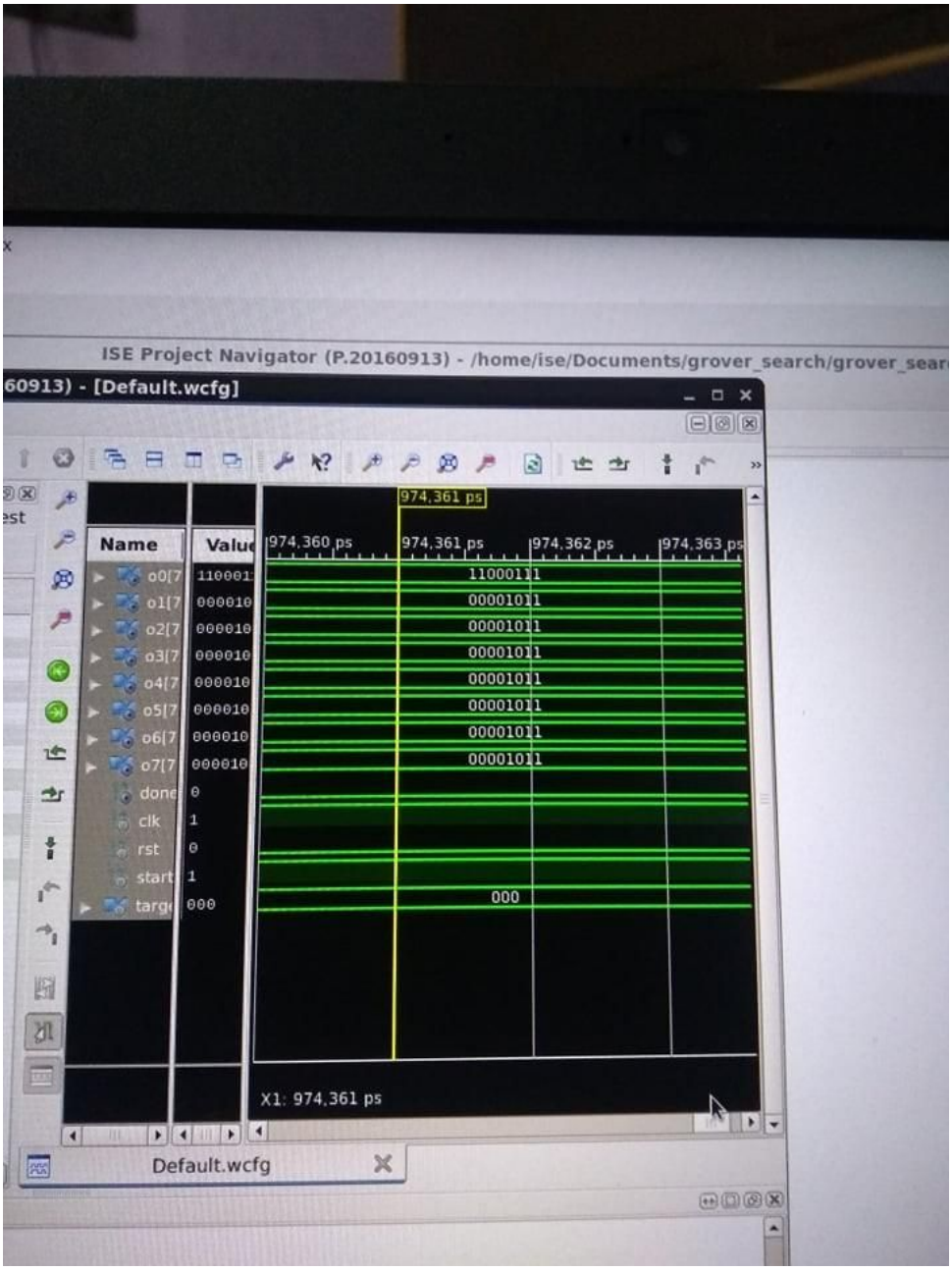

Figure 10: Amplitudes for elements after the final iteration

7.2 Quantum Fourier Transform

The simulation results for Quantum Fourier Transform are shown in figure 11:

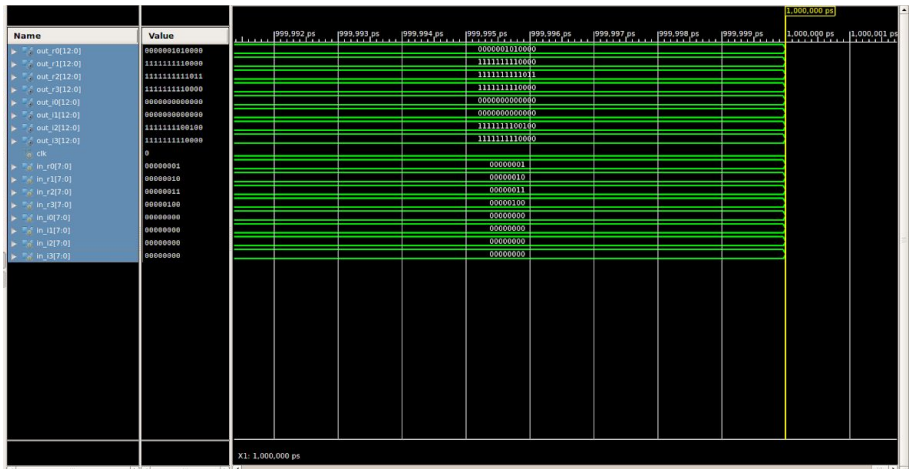

Figure 11: Simulation of QFT in Verilog

8.Results

In this project we have tried to maintain the natural time complexity of the algorithms. As we know that quantum algorithms are much faster than their classical counterparts we measured the time taken for simulation of these algorithms in C and their simulation in Verilog and the results are given below.

In C, to measure the time for simulation we used the time.h library.In Verilog, the simulator shows the time required for the simulation to complete. The results are shown in the following table.

8.1 Grover's Search Algorithm

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

8.2 Quantum Fourier Transform

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

9.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.