Poster 11

System-on-Chip Based Emulation of Quantum Circuits

By Finlay Harris

Dr Louise Crockett

Introduction to Quantum Computing:

What is a Quantum Computer?

A **quantum computer** leverages **quantum mechanics** to process information. This is done using **qubits** instead of classical bits.

What are Qubits and Superposition?

Unlike classical bits, which exist as either **0 or 1**, **qubits** can exist in a **superposition of both states** simultaneously. A qubit's state can be represented mathematically as Equation 1:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

Figure 1 shows a Bloch Sphere representation of a qubit.

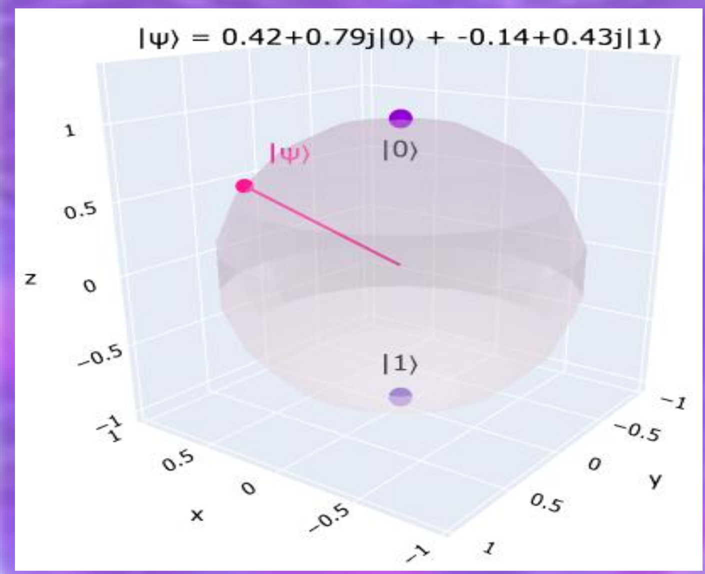

Figure 1: *The Bloch Sphere*

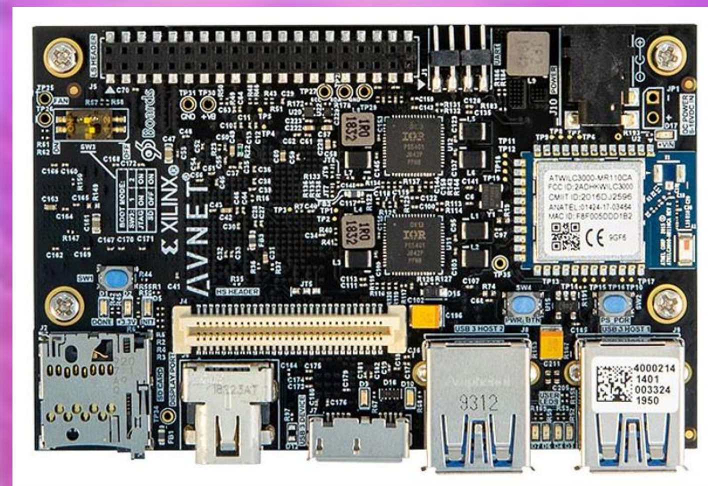

Figure 3: Ultra96-V2 Board

What are Quantum Gates and Circuits?

Quantum gates manipulate qubits.

A **quantum circuit** is a sequence of quantum gates applied to qubits

Table 1 Key Quantum Gates

Gate	Description
X/Y/Z Gate	Phase shift of π radians around X/Y/Z axis.
Hadamard	Applies a 90° rotation around the y-axis followed by a 180° rotation about the x-axis
CPhase (S and T)	Apply phase shifts (π/2 for S, π/4 for T) to the qubit's state based on the state of a control qubit.
SWAP	Swaps the quantum states of two qubits.

Quantum Fourier Transform (QFT) & Shors Algorithm:

QFT **maps a quantum state to its frequency domain**, creating a superposition of frequency components. It is key for periodicity detection, such as in Shor's algorithm for factoring, where it finds the periodicity of modular exponentiation.

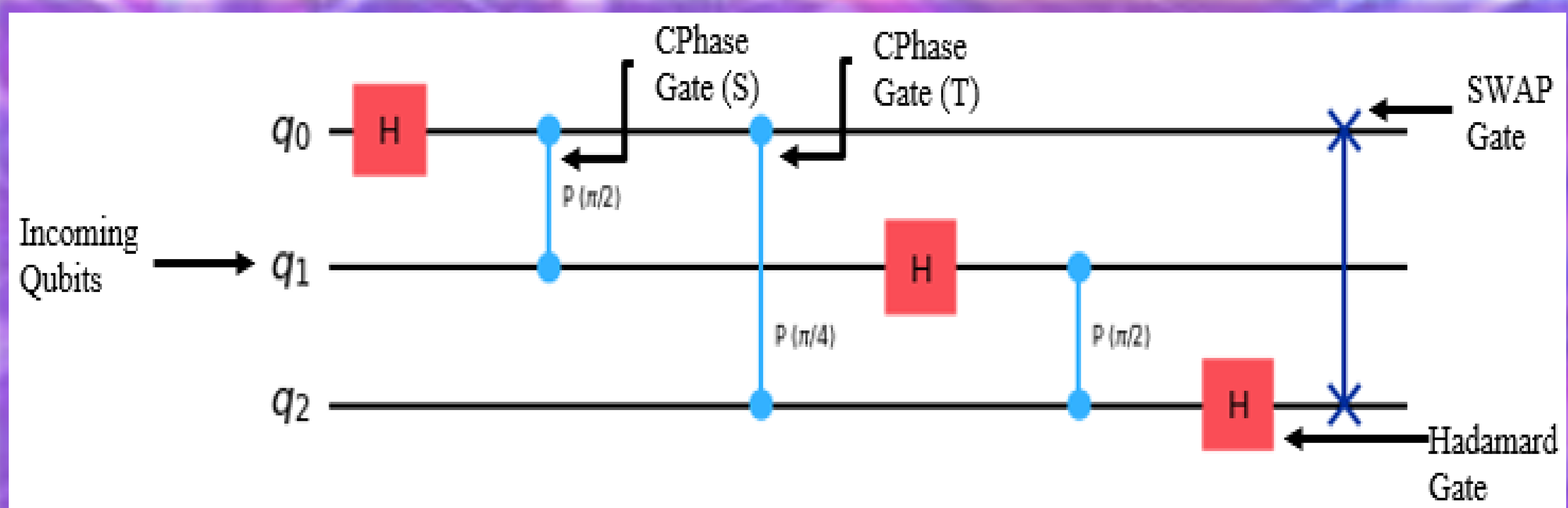

Figure 2: *A 3-qubit QFT Circuit Diagram*

The Design Process:

1. Model Composer Design
2. Create block diagram in Vivado.
3. Generate bitstream.
4. Add bitstream to PYNQ folder.
5. Write Python code to manipulate data into expected format for PL.
6. Transfer to PL and back to PS.
7. Display output data on Bloch Sphere .

An Introduction to the project:

This project **emulates quantum gates** and **algorithms** on the **Ultra96 MPSoC**. The project objectives include:

1. Emulate quantum gates and algorithms (QFT, Shor's) with Model Composer and Vivado.
2. Develop a user-friendly interface for data input, quantum circuit visualisation, and result display.
3. Create an AI model for quantum error correction and explore different techniques.

Why is this project important?

This work provides accessible quantum algorithms for education without quantum hardware, offering greater flexibility and scalability than traditional FPGA emulators.

System Design

Quantum gates and algorithms are emulated using the **Model Composer and Vivado FPGA Design Tools** for the hardware aspects**,** along with **PYNQ, and Python** for the software aspects.

Quantum Gates:

- Qubits are represented as **4 × 32-bit fixed-point numbers**.
- Each gate follows **mathematical operations on α and β values**.
- **AXI-Stream** template enables PS-PL communication
- **A bitstream** was generated using Vivado and used within PYNQ.
- **Python code** then converts the data out from the PL into floating point, used to **generate a Bloch sphere** to see the transformation of the gate.

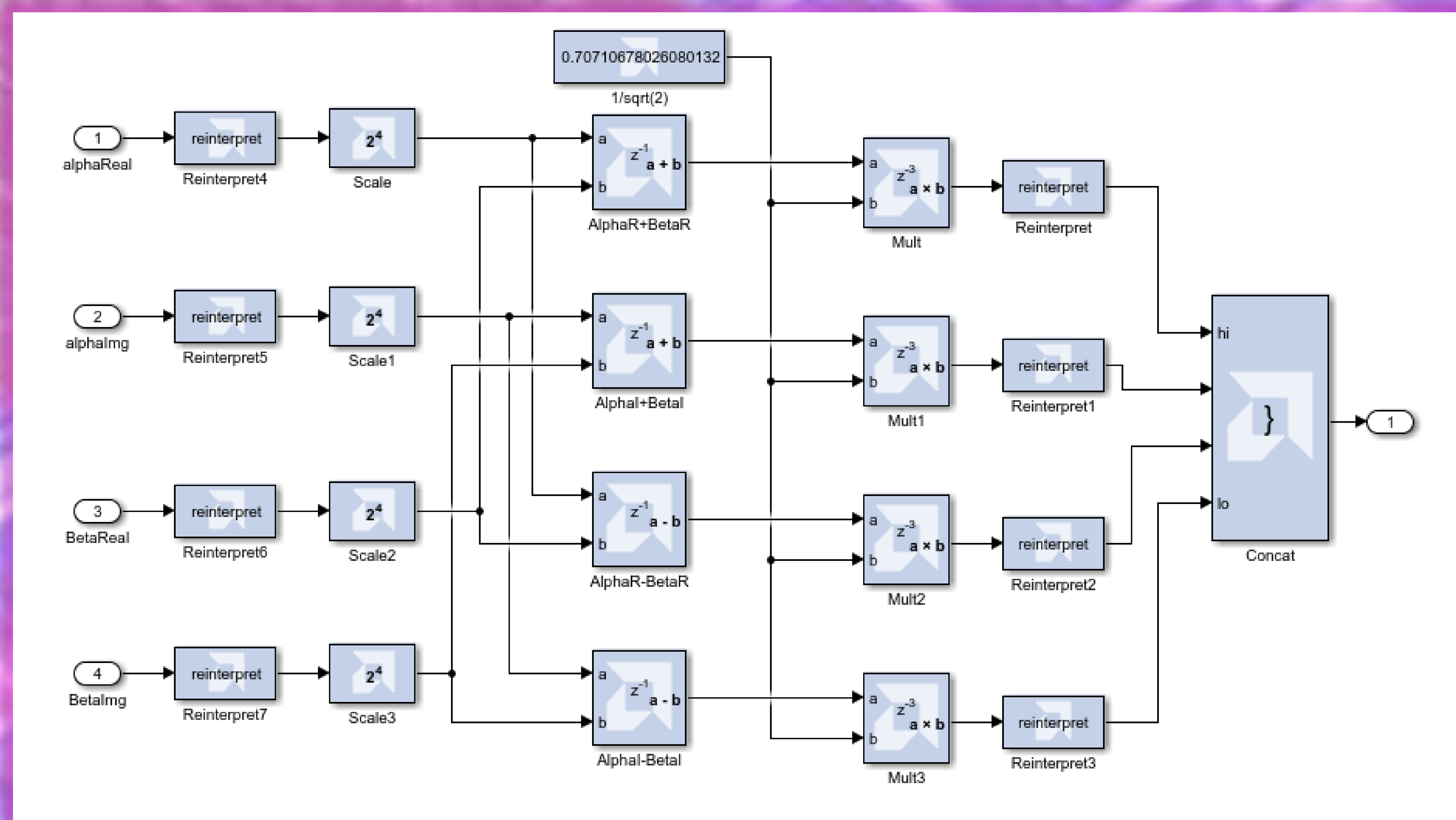

Figure 4: Model Composer Design of Hadamard Gate

3-Qubit QFT:

- The design uses the different gates designed previously.
- **3 Qubits** are input via AXI4-Lite and are represented as 32-bit unsigned fixed-point numbers.
- The **bitstream** was generated in Vivado and transferred to PYNQ where Python code is written to display the data.

Results and Progress:

- The emulation of quantum gates was highly accurate.
- Minor truncation issues in the S, T, and Hadamard.
- **3-qubit QFT**: Real differences ranged **0.0037 - 0.0116.**
- **Phase shifts:** Between **1.54 rad** and **π rad** due to a global phase shift.
- Shor's algorithm not achieved due to the need for more qubits in the QFT.
- GUI implemented to interact with project.

Error Correction:

- Research on error correction methods.

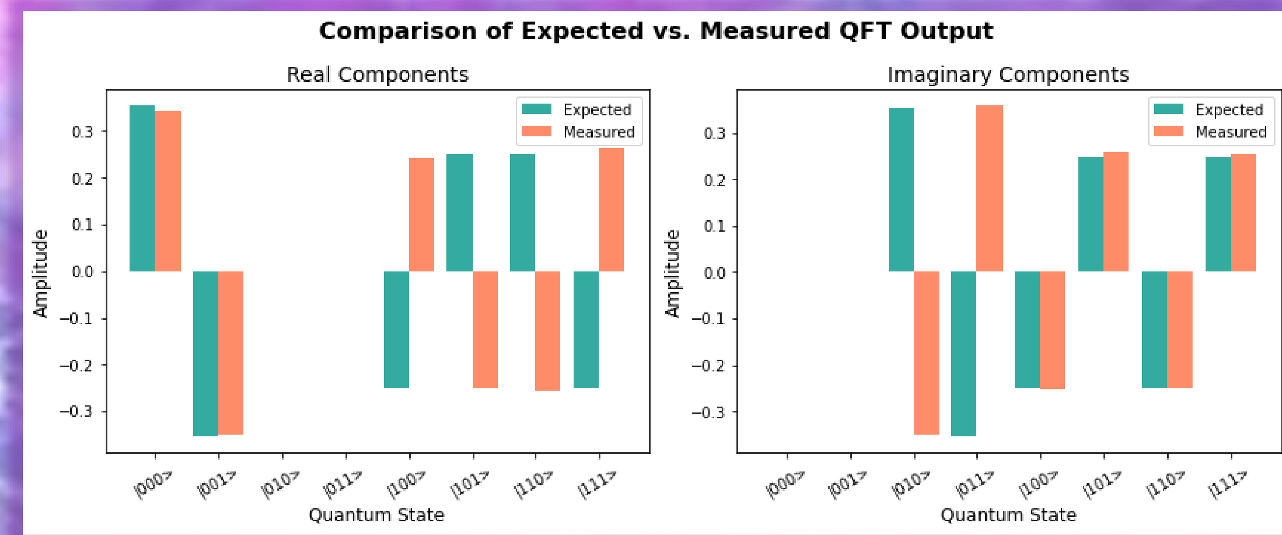

Figure 5: Graphical Comparison of QFT Results

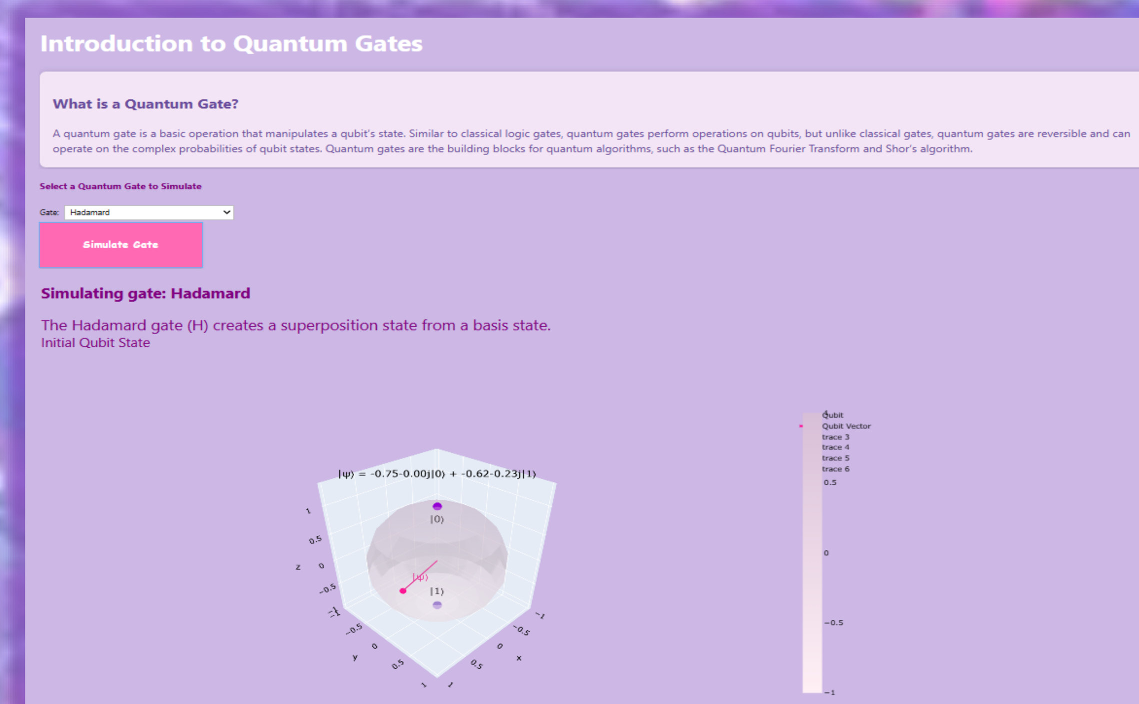

Figure 6: Screenshot of GUI

Conclusion:

Achieved:

- Successfully emulated all quantum gates.
- Successfully emulated a 3-qubit QFT.
- Set up a GUI so users can interact.

Challenges:

- AXI DMA communication limited to 128 bits, causing truncated results.
- The project's large scope prevented full exploration of error correction techniques.

Future Work:

- Implement a higher qubit QFT for Shor's algorithm.
- Implementation of error correction.