Dissertation Plan – Enhancing a simulator for quantum stabiliser circuits

Alison Dauris – ad2160

**Introduction** - 500 - 1000

Project Summary

**Preparation** - 3500-4000

Quantum Computing

Qubits

Superposition and basis vectors

Multi qubit systems

Quantum circuits

Quantum gates

Quantum Measurement

State Preparation

Ancillas

Quantum Error Correction Codes

Quantum Errors

Noisy Quantum Processors

Noise Representation

Stabilisers

X Stabiliser

Z Stabiliser

Topological QEC

Planar codes

Logical qubits

Logical Operators

Syndrome analysis

Pauli Frame

Minimum Weight Perfect Matching (MWPM)

Threshold

Quantum Error Correction Code Examples - not sure, I might be covering some of these in the Interface design section.

Stim

Requirements Analysis

Model of software development

Licensing

Version Control and Testing

Starting Point

Summary

**Implementation** - 3500-4000

Interface design - Design decisions

Base coordinate system

Qubits involved selection

Representation of the error correction codes

Adding gates vs defining plaquettes

Quantum gates and measurement available to add

H, CNOT, MRZ, MRX - allowing code depth

System of having rounds

Not having pre-round and post-round gates

Adding numbering to CNOT execution

The importance of the CNOT ordering

Specifying the basis

Evaluation of the Interface Design

Table of codes representable

Noise Model Design

Noise Model Evaluation

Methodology used when designing interface

Frontend Implementation

Data Structures?

Backend Implementation

API endpoints

Stim Generation

Graph Creating

Threshold Calculation

Overview

Repository Overview

Language Choice

Software Implementation Choices

**Evaluation** - 2000 - 2500

Review of Project Requirements

Core Evaluation Metrics

Evaluating the effectiveness of the Stim Code generated - since I wrote code to write generic STIM rather than specific – not sure how practical?

Simulation run-time

Memory use

Usability - would do a user study

Task Completion time

Error rate in configuration

Learnability

Functionality

Generality - which QEC codes can be modeled

Stim Code accuracy - compare outputs of my interface to known benchmarks (generate graphs that have appeared in papers). Agreement with other models

Extensibility - eg Looking at using the interface to deal with having defects in chips

Comparison to Other works

Limitations

**Conclusion** - 500

Lessons Learnt

Future Work