

## SQASM: Simple Quantum Assembler

Ryan Watkins

BSc Computer Science r.watkins@student.reading.ac.uk

### ABSTRACT

New quantum programming languages need to be derived for when full-scale quantum systems become available. SQASM is a simple quantum assembler presented as a language other higher order QPL (Quantum Programming Language) implementations can compile down towards. The language is designed for the ease of the programmer to write code close to hardware also. SQASM ties into a complimentary quantum simulator with the help of parsing and lexical analysis from YACC and LEX respectively. The quantum simulator is written in Python. The idealized architectural model chosen is quantum random access machine (QRAM). One may perform local quantum algorithms and arithmetic with compilation down to gate descriptions. Full adherence of quantum mechanics to provide true quantum simulation. The simulator contains implementations of a quantum ripple carry adder [3], Quantum Cost efficient quantum multiplier [4] and also Deutsch's algorithm [2]. Both the simulator and programming language can be used as individual components to perform quantum computational simulation or work in tandem. Wrapper functions within the simulator allows SQASM to call certain functions to perform quantum computation.

```
INITIALIZE R 2
...
ADD 3 2 R2
MEASURE R2 RES
```

**Figure 1: SQASM compilation input**

```
--- ADD TERM TERM ---
Trying to call Quantum Simulator...
Elapsed time for add: 0.031271s
SUCCESS: ADD
SUCCESS: Python Simulator function called
SET HASH BIN: 50
--- MEASURE TERM TERM ---
Trying to call Quantum Simulator...
Amount of amplitudes in register: 65536
Selection range: [0,15]
RES: [1,0,1,0,0,0,0,0,0,0,0,0,0,0,0]
SUCCESS: Python Simulator function called
SET HASH BIN: 70
```

**Figure 2: SQASM compilation output**