

# CAB402 Programming Paradigms

## Quantum Computing

Thanat Chokwijitkul n9234900

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Early History</b>	<b>3</b>
<b>3</b>	<b>Making a bibliography</b>	<b>4</b>
	<b>References</b>	<b>5</b>
<b>A</b>	<b>Example of operation</b>	<b>6</b>
A.1	Example input and output . . . . .	6
A.1.1	Input . . . . .	6
A.1.2	Output . . . . .	6
A.1.3	Another way to include code . . . . .	6

# 1. Introduction

The theory of computation has been extensively developed during the last few decades. Computers have provided reliable solutions for a myriad of community's seemingly unsolvable problems. Notwithstanding, various complicated problems have been continuously introduced to society as it never stops growing and becomes more complex. Even though technology nowadays has been steady advancing to approach the demands of society, whereas such steady progress has its limitation since many of those problems are intricate to model or appear to require time-intensive solutions. This phenomenon implies to the necessity of a new computing revolution since classical computation no longer has an ability to reach the increased demand.

According to Moore's law, the computational power of computers would dramatically increase approximately every two years [Thompson and Parthasarathy, 2006]. The theory can be proven real since the size of transistors has rapidly become smaller to a few nanometres [Nielsen and Chuang, 2000]. This results in a higher number of transistors mounted on an integrated circuit. The laws of classical physics do not function with objects with such small size. Thus, quantum computing becomes a next solution to deal with these complex problems.

Quantum computing is the revolution. It is a relatively new field of research which has the potential to introduce the world of computing to a new stage where certain computationally intense problems can be solved with a shorter amount of processing time. One can consider quantum computing as the art of utilising all the possibilities that the laws of quantum physics contribute to solving computational problems while classical computers merely use a minuscule subset of these possibilities. However, quantum computers are not a replacement for conventional computers as quantum mechanics only improve the computing efficiency for certain types of computation.

This research report delivers a brief glimpse into the world of quantum computers and the laws of quantum mechanics applied to this entirely new type of computer. It will also introduce the fundamental concepts of the field along with exploring quantum computing in practice using Language-Integrated Quantum Operations (LIQUi|)). This report concludes with experience evaluation in the quantum computing research in relation to the content of the unit (Programming Paradigms).

## 2. Early History

In the early 1980s, physicist Richard Feynman recognised that it was not possible for phenomena associated with entangled particles (quantum phenomena) to be efficiently simulated on classical computers [Deutsch, 1982]. Feynman was the first who suggested that quantum-mechanical systems might have higher computational power than conventional computers.

In the same decade, Paul Benioff proved that quantum-mechanical systems were at least as powerful as classical computers because Turing machines could be modelled with such system [Yanofsky and Mannucci, 2008].

In 1985, David Deutsch, of Oxford, published the paper "Quantum theory, the Church-Turing principle and the universal quantum computer" [Deutsch, 1985]. Deutsch claimed that the universal Turing machine has superior abilities compared with Turing machine, including random number generation, parallelism and physical system simulation with finite-dimensional state spaces.

Another of Deutsch's paper "Quantum computational networks" [Deutsch, 1989] was published in 1989. The article proved that quantum circuits are as powerful as the universal Turing machine. Deutsch also introduced the first truly quantum algorithm in 1990, namely Deutsch's algorithm which later was generalised to the Deutsch-Jozsa algorithm.

In 1993, Andrew Chi-Chih Yao extended Deutsch's paper "Quantum computational networks" [Yao, 1993] in the paper "Quantum circuit complexity" by addressing the complexity of quantum computation according to Deutsch's work. Yao's findings indicated quantum computing researchers to concentrate on quantum circuits instead of quantum Turing machine.

Peter Shor proposed another quantum algorithm in 1994. This algorithm uses the concept of qubit entanglement and superposition for integer factorisation [Bhunia, 2010]. In principle, executing the algorithm on a quantum computer would far surpass the efficiency of all classical computers.

The University of California at Berkeley, MIT, Harvard University, and IBM researchers conducted an experiment using nuclear magnetic resonance (NMR) to manipulate quantum data in liquids. The team also developed a 2-bit quantum computer with radio frequency as its input. Afterwards, a new quantum algorithm that executes on quantum computers was introduced by Lov Grover of Bell Laboratories in 1996 by the name Grover's quantum algorithm [Bhunia, 2010].

In 1998, researchers at the University Innsbruck in Austria put the idea of quantum teleportation [IBM, 2014], proposed in 1993, into practice. The theorem demonstrates the concept of entanglement and teleportation. This research is an implication for data transfer and network in the quantum system.

### **3. Making a bibliography**

# References

- [Bhunia, 2010] Bhunia, C. T. (2010). *Introduction to Quantum Computing*. New Age International.
- [Deutsch, 1982] Deutsch, D. (1982). Quantum computation. *Physics World*.
- [Deutsch, 1985] Deutsch, D. (1985). Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London, Series A*, 400(1818):97–117.
- [Deutsch, 1989] Deutsch, D. (1989). Quantum computational networks. *Proceedings of the Royal Society of London, Series A*, 425(1868):73–90.
- [IBM, 2014] IBM (2014). Quantum teleportation. [http://researcher.watson.ibm.com/researcher/view\\_group.php?id=2862](http://researcher.watson.ibm.com/researcher/view_group.php?id=2862).
- [Nielsen and Chuang, 2000] Nielsen, M. A. and Chuang, I. L. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press.
- [Thompson and Parthasarathy, 2006] Thompson, S. E. and Parthasarathy, S. (2006). Moore’s law: the future of si microelectronics. *Materials Today*, 9(6):20–25.
- [Yanofsky and Mannucci, 2008] Yanofsky, N. S. and Mannucci, M. A. (2008). *Quantum Computing for Computer Scientists*. Cambridge University Press.
- [Yao, 1993] Yao, A. C.-C. (1993). Quantum circuit complexity. *Proceedings of 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 352–361.

# A. Example of operation

An appendix is just like any other chapter, except that it comes after the appendix command in the master file.

One use of an appendix is to include an example of input to the system and the corresponding output.

One way to do this is to include, unformatted, an existing input file. You can do this using `\verbatiminput`. In this appendix we include a copy of the C file `hello.c` and its output file `hello.out`. If you use this facility you should make sure that the file which you input does not contain TAB characters, since  $\text{\LaTeX}$  treats each TAB as a single space; you can use the Unix command `expand` (see manual page) to expand tabs into the appropriate number of spaces.

## A.1 Example input and output

### A.1.1 Input

(Actually, this isn't input, it's the source code, but it will do as an example)

```
/* Hello world program */

#include <stdio.h>

int main(void)
{
    printf("Hello World!\n") ;
    return 0 ;
}
```

### A.1.2 Output

### A.1.3 Another way to include code

You can also use the capabilities of the `listings` package to include sections of code, it does some keyword highlighting.

```
/* Hello world program */

#include <stdio.h>

int main(void)
```

```
{  
    printf("Hello _World!\n") ;  
    return 0 ;  
}
```