Acceptance Speech of

PROFESSOR PETER WILLISTON SHOR

Co-Winner of the 2002

King Faisal International Prize for Science

Your Royal Highness, Prince Sultan Ibn Abd Al-Aziz
Second Deputy Premier, Minister of Defence and Aviation
And Inspector General
Your Royal Highnesses
Your Excellencies
Distinguished Guests

It is a great honor to receive this award, and I would like to use this occasion to thank the King Faisal International Prize Committee, the professors who submitted my nomination, and the organizers who planned this event. I am receiving this prize in part for my discovery that a quantum computer, a hypothetical machine which I hope will be built sometime later this century, could factor large numbers into primes much more quickly than a conventional digital computer. As with most mathematical and scientific discoveries, my discovery of the factorization algorithm did not proceed from a vacuum, but depended on a' great number of previous discoveries. I would like to trace for you the history of one sequence of discoveries leading to my research. I do this both in the hope that it illustrates better the process of scientific discovery, and also in the possibly mistaken belief that mentioning at this ceremony the names of researchers whose work was essential to my discovery goes some little way towards compensating them for not being here.

I start with the concept of an algorithm. An algorithm is a step-by-step procedure that can be followed mechanically to perform a computation. As

computers have no real insight, it is necessary to first have an algorithm for a problem in order to program a computer to solve the problem. The English word algorithm is derived from the name of the great Arabic mathematician Muhammad Ibn Musa al-Khwarizmi, who introduced the decimal numerals in the ninth century. The word algorithm originally meant the procedures for performing arithmetic using decimal numerals, and later came also to mean procedures for performing other computations.

David Hilbert was a leading mathematician in the early twentieth century, one of whose gifts was for identifying problems which would be fruitful to attack. In 1928 he posed three problems in the foundations of mathematics, the last of which asked whether an algorithm existed which would determine whether a mathematical proposition was true or false. It was shown in 1936 that no such algorithm could exist; on the way to this result, four papers were written that drew a distinction between computable and non-computable functions. These papers, by Alonzo Church, Stephen Kleene, Emil Post and Alan Turing, contained three completely different definitions of what it meant for a function to be computable. It was soon shown that these three different definitions led to the exact same class of computable functions. This led Church and Turing to propose that this was the natural class of computable functions; this is now called the Church-Turing thesis.

After the first computers were built, it became evident that the distinction between computable and non-computable functions was much too coarse for use in practice. If one needs to obtain a solution to a problem, it is not much use to know that the solution is computable in theory, if the computation would take too long to ever be completed in practice. It gradually became clear that some means of characterizing efficiently computable functions was needed. In the 1960’s and 1970’s computer scientists, most particularly Stephen Cook and Richard Karp, arrived at functions computable in polynomial-time; computable functions seem to be computable reasonably efficiently in practice; additionally, the class of polynomial time as a good compromise between theory and practice. Most natural time - computable

functions had enough structure that interesting theorems could be proved about it. Of course, for the definition of functions computable in polynomial time to be universally applicable, it must be independent of the machine used for computation. This led to a strengthening of the Church - Turing thesis; which strengthening says that any function computable by any means can also be computed by a digital computer, while incurring only a polynomial amount of extra overhead. It was a great surprise to computer scientists that quantum computers appear to violate this strengthened Church-Turing thesis.

Until the investigation of quantum computation, it was not widely realized that the Church-Turing thesis, as well as its strengthening, are in truth statement about physics and not mathematics. To show that a digital computer can simulate any possible computer, one has to consider all computers which might exist in the physical world, and it is therefore the laws of physics which constrain the possible machines which might be built. If one were to look for a counterexample to the strengthened Church-Turing thesis, one should thus look for an area of physics which cannot be simulated efficiently on digital computers. One such area is quantum mechanics. It was observed by Yuri Manin, in 1980 in the Soviet Union, and by Richard Feynman, in 1982 in the United Sates, that it is extremely expensive to simulate quantum mechanics using a digital computer; and they proposed that quantum computers might be much more efficient at this task. In 1985, David Deutsch asked the question of whether quantum computers might be more efficient than digital computers for other computational tasks. This question was further addressed by Deutsch, Richards Jozsa, Ethan Bernstein, Umesh Vazirani, and finally Daniel Simon; these researchers found successively better examples of problems which quantum computers could solve more quickly than classical computers. None of these problems, however, was interesting in its own right. While examining Simon's paper, I realized that the key to his algorithm was periodicity structure certain functions he was considering. Since I knew that periodicity was related to the problem of factoring large numbers into the product of primes, this led me to start looking for the factoring algorithm on a quantum computer. The

difficulty of factoring is a crucial component of modern cryptography, so my discovery of the factoring algorithm launched the field of quantum computation, which had previously been a sideline studied by only a few people, into the spotlight.

What conclusion can be drawn from this brief history? I think the main conclusion is that the directions science will move in can rarely be foreseen. It appears that David Hubert initially thought that there would be an algorithm for determining the truth for propositions. David Deutsch started investigating quantum computing in relation to questions about the foundations of quantum mechanics which quantum computing has failed to shed much light upon. Daniel Simon discovered his quantum algorithm by trying to prove that quantum computers were no more powerful than classical computers. And when I started to look at quantum computing, I did not expect it would be related to prime factorization. This unpredictability of science makes it very difficult to answer one of the questions which I am asked most frequently: namely, when will useful quantum computers be built? It is also one of the things that makes science so interesting.

Thank you