2017-12-04

Experiments in Sound and Music Quantum Computing

Kirke, Alexis

10.1007/978-3-319-49881-2_5
Springer International Publishing

Chapter 5

Experiments in Sound and Music Quantum Computing

Alexis Kirke and Eduardo R. Miranda

Abstract. This chapter is an introduction to quantum computing in sound and music. This is done through a series of examples of research applying quantum computing and principles to musical systems. By this process, the key elements that differentiate quantum physical systems from classical physical systems will be introduced, and what this implies for computation, sound and music. This will also allow an explanation of the two main types of quantum computers being utilized inside and outside of academia.

5.1 Introduction

5.1.1 Background

Max Plank was born in 1858 in Kiel, Germany, into a family of distinguished scholars in the fields of Theology (grandfather) and Law (father). He loved music and was an accomplished pianist. He had considered pursuing a career as a professional musician, but apparently the teenager was advised that he should better study something else. He ended up studying Physics at Munich University and came to be one of the most important scientists of the 20th century. Max Plank is one of the pioneers of Quantum Physics.

In the beginning of the 20th century Plank demonstrated that energy of electromagnetic radiation could only be emitted or absorbed as bundled packs of energy, which he coined as quantum, or quanta in plural. The notion of quantum energy was a breakthrough from the long-established notion that energy was emitted

or absorbed from matter continuously. Albert Einstein subsequently followed from Plank's work and proposed that light was not continuous either, but made of particle-like quanta, named as photons. The invention of the quantum opened a Pandora's box of strange concepts that define our microscopic physical world, such as the notion that the energy of an electron in an atom is a quantized particle, which also behave as a wave. Strangely, it has been demonstrated that an electron in an atom could be in one place and then reappear in another without being anywhere in between. And more, particles can be linked is such strange way that changes to one instantaneously affects the other, even if they are very far apart from each other: this phenomenon is referred to as entanglement. The microscopic, atomic and subatomic world is mind-boggling.

Metaphorically, think of the distinction that we often make between analogue and digital electronics: before the discovery of the quantum, the Universe was analogue. Now it is digital. It remains a mystery how the everyday world of Newton's macroscopic physics emerges from the bizarre microscopic, atomic and subatomic world. Nevertheless, although we do not fully understand the world of quantum mechanics, scientists have been looking into harnessing it to develop computers operating on quantum principles.

In the early 1980s Richard Feynman, from the California Institute of Technology, proposed the idea of developing a quantum mechanical computer. Subsequently David Deutsch, a visiting professor at Oxford University, laid the foundations of the quantum theory of computation, which effectively kick-started research into building such machines (Deutsch 1985).

Like classical computing, quantum computing works with bits – i.e. values of 0 and 1. In quantum computing these are called qubits. There are two main forms of quantum computing, which will be discussed in this chapter: traditional quantum computing and adiabatic quantum computing.

One system we will examine in this chapter is a logic gate with qubits represented by light waves. This is a good example for discussing some advantages of quantum computing. Quantum light waves act like particles as well as waves (Shadbolt et al. 2014); these particles are called photons. Most people are familiar with the light interference patterns seen during experiments at high school. This is normally explained as light wave peaks interfering with light wave troughs. However many experiments have also been done to show individual photons of the light build up interference patterns. The wave patterns seen in interference in fact summarize the *probability* of finding an individual photon at a particular point. The axioms of quantum mechanics say that one may only calculate the probability of particle being in a certain state. In fact before a particle is measured, it can be thought of as being in multiple possible states. It is this superposition of states and its implications that leads to the useful features of quantum computing. A qubit is actually a superposition of multiple bit-states, a property that leads of the speeding ups expected in traditional quantum computing (Shor 2006).

The speed up in adiabatic quantum computers is due to a different but related effect, which will be discussed later in the chapter.

5.1.1 Music and Quantum Physics

The natural world has always been a rich source of inspiration for music. Vivaldi's *Four Seasons* and Beethoven's *Pastoral Symphony* immediately come to mind as two examples of classical music inspired by nature. And of course there is Holst's *The Planets*, which is one of the most celebrated examples of music inspired by the solar system that I can think of. The ancient Greek philosophical maxim, that astronomy is for the eyes what music is for the ears, still inspires composers today. Indeed a plethora of approaches to composing music inspired by natural science has emerged since *The Planets* was composed a century ago, including approaches inspired and informed by Physics and indeed Quantum Physics.

Even though initiatives to explore the potential of quantum computing for music emerged only very recently, concepts pertaining to Quantum Physics have been making their way into music technology and indeed musical thinking for decades. For instance, in the late 1950s composer Karlheinz Stockhausen published an article entitled "… How Time Passes …" (Stockhausen 1959) where he introduces approaches to correlate different musical time scales. He proposed that the notions of pitch and rhythm should be viewed as one unified continuous time-domain phenomena. A number of musicologists objected this article. Nevertheless, despite its alleged musicological flaws, it introduces new ways of thinking about music, which were comparable with the way in which his contemporary physicists were looking into the natural world.

Another example is the remarkable work on to stochastic music introduced by Iannis Xenakis in his book *Formalized Music* (Xenakis 1971). Here he laid out principles for composing music using mathematical processes based on statistical mechanics. For instance, to compose the piece *Pithoprakta* he developed a method to render a model describing Brownian motion - the erratic dance of tiny particles - into musical notes. The mathematical representations of music developed by Xenakis open the doors for developing representations of music based on stochastic models of quantum mechanics (Davidson 2007).

A paper introducing the notion of acoustical quanta, published in 1947 by Hungarian-British physicist and Nobel Prize winner Dennis Gabor (Gabor 1947) has been very influential in the field of computer music, as it paved the way for a sound synthesis technique referred to as *granular synthesis* (Miranda 2002). Not surprisingly, this method for representing and synthesizing sounds has been featured in various chapters in this volume. Gabor proposed the basis for a representation method, which combines frequency-domain and time-domain information of sound. His point of departure was to acknowledge the fact that the ear has a time threshold for discerning

sound properties. Below this threshold, different sounds are heard as clicks, no matter how different their spectra might be. The length and shape of a wavecycle define frequency and spectrum properties, but the ear needs several cycles to discern these properties. Gabor called this minimum sound length as *acoustic quanta* and estimated that it usually falls between 10 and 30 milliseconds, according to the nature of both the sound and the subject. Gabor's theory fundamentally suggested that a more accurate representation of sound is obtained by 'slicing' the sound into very short segments (grains, or 'sound quanta'), and by applying a Fourier type of analysis to each of these segments. This granular representation of sound facilitates the development of models of music using the mathematical tools that are alike those used in Quantum Physics. For instance (Miranda and Maia Jr. 2005) introduced a method for granular sound synthesis using Walsh functions (Fine 1949) and Hadamard matrices (Wallis 1976).

There have been an increasing number of practice-based musical projects inspired or informed by quantum mechanical processes, but the great majority of these still are metaphorical, based on simulation and/or not directly connected to actual quantum effects.

For instance, the webpage "Listen to the Quantum Computer Music" is an interesting initiative (Weimer 2014). Two pieces of music are playable online through MIDI simulations. Each is a sonification of a well-known quantum computation algorithm. One is Shor's Algorithm (Shor 2006), which is the factorization algorithm mentioned earlier. The other is a database search algorithm known as Grover's algorithm (Grover 2001). The offline sonification of quantum mechanics equations have also been investigated in (Sturm 2000), (Sturm 2001) and (O' Flaherty 2009), with the third being an attempt to create a musical signature for the Higgs Boson at CERN before its discovery. Another paper defines what it calls Quantum Music (Putz and Svozil 2015), though once again this is by analogy to the equations of quantum mechanics, rather than directly concerned with quantum physics. Certain equations of quantum mechanics have also been used to synthesize new sounds (Cadiz and Ramos 2014). The orchestral piece "Music of the Quantum" (Coleman 2003) was written as an outreach tool for a physics research group, and has been performed on various occasions. The melody is carried between violin and accordion. The aim of this was as a metaphor for the wave particle duality of quantum mechanics, using two contrasting instruments.

We cite "Danceroom Spectroscopy" as an example of a quantum simulation performance (Glowacki 2012) where quantum molecular models generate live visuals. Dancers are tracked by camera and their movements treated as the movement of active particles in the real-time molecular model. Thus the dancers act as a mathematically accurate force field on the particles, and these results are seen in large scale animations around the dancers.

There have been performances and music that use real-world quantum-related data. However most of these have been done offline, rather than using physics occurring during the performance. These include the piece "Background Count": a pre-recorded

electroacoustic composition that incorporates historical Geiger counter data into its creation (Brody 1997). Another sonification of real-physics data done offline was the "LHChamber Music" project (Alice Matters 2014), which used a harp, a guitar, two violins, a keyboard, a clarinet and a flute.

One of the first real-time uses of subatomic physics for a public performance was "Cloud Chamber", which was discussed Chapter 2 in this volume. In "Cloud Chamber" physical cosmic rays are made visible in real-time, and some of them are tracked by visual recognition and turned in to sound. A violin plays along with this, and in some versions of the performance, the violin triggered a continuous electric voltage that change the visible particle tracks, and thus the sounds. Cloud Chamber was followed a few years later by a CERN-created system, which worked directly, without the need to a camera. Called "Cosmic Piano", it detects cosmic rays using metal plates and turns them into sound (Wired 2015).

The previous two musical performances were live, and the data ware not quantum *as such*. They were quantum-related in that the cosmic rays and cloud chambers are subatomic quantum processes. However, they do not incorporate actual quantum computation in their music.

As has been mentioned, this chapter will discuss two types of quantum computing: traditional (or gate-based) and adiabatic. Gate-based quantum computing involves implementing Boolean logic gates such as AND, OR and NOT, as quantum systems. The advantages of doing this are that before measurement, a quantum system is in a state of superposition, so a single physical quantum gate circuit will be represented as a superposition of a vast number of such circuits in parallel. With careful preparation it is possible for calculations to occur in this superposition state, thus leading to an immense speed up in calculations.

A photonic quantum computer will now be introduced followed by work on to music and quantum computation has been developed at Plymouth University's Interdisciplinary Centre for Computer Music Research (ICCMR).

5.2 Photonic Quantum Computing

The quantum computer set up utilized in this next section exactly simulates a quantum CNOT (Controlled NOT) gate. The CNOT gate acts on two quantum bits, or qubits; a qubit is the quantum-mechanical analog of a classical bit. CNOT is a two-bit gate where a control bit (referred to as C) stays the same but the other bit changes during an operation (i.e., from 0 to 1 or vice-versa) if C = 1 or stays the same if C = 0. In other words, a CNOT gate flips ($|0\rangle \rightarrow |1\rangle$) the state of the target qubit if and only if the state of the control qubit is $|1\rangle$. Various physical platforms for quantum computing have been proposed including ion traps (Kielpinski 2002) and superconducting qubits (Kelly et al. 2015). Here we consider a photonic quantum computer (Knill et al. 2001): a scheme for efficient quantum computation with linear

optics in which information is represented in the quantum state of optical-frequency photons.

In the hardware, photons are obtained by focusing a 404nm laser on to a piece of nonlinear crystal (Bisumuth Borate). This causes the crystal to probabilistically spit out 808nm photon pairs, in a process known as Type I spontaneous parametric down conversion. The chip, which performs several experiments that would each ordinarily be carried out on an optical bench the size of a large dining table, is 70mm by 3mm. It consists of a network of tiny channels, which guide, manipulate and interact single photons. Waveguides are made with a higher refractive index than their surroundings, so that photons can propagate along them by total internal reflection. The waveguides in the integrated optical device are made from silica and sit in a wafer of silicon, which allows things to be kept on a relatively small scale; the chip is 70mm x 3mm.

Using eight reconfigurable electrodes embedded in the circuit, photon pairs can be manipulated. A schematic is shown in Figure 1. The circles with numbers in them are known as phase shifters, and will be discussed later. They are able to change the phase of the photons. The points where the lines meet are called beam splitters, which will be explained below, and also enable further quantum effects to be added to the calculation.

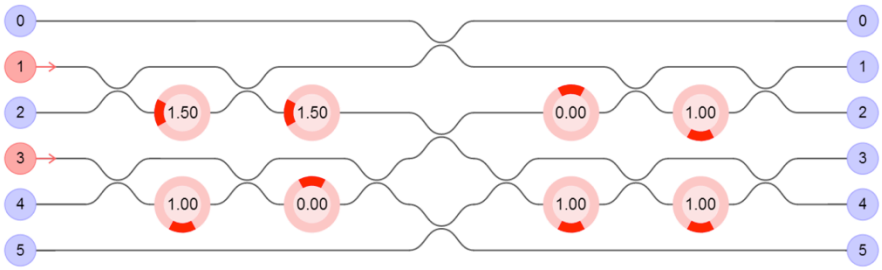

Figure 1: Schematic of the Photon Quantum Computer showing photons being input on (1) and (3) and various phase shifter settings in the pathways.

Table 1: Setting up inputs on the quantum CNOT

	Control	**Target**	**Inputs to send photon in to**
0	0	1, 3	
0	1	1, 4	

1	0	2, 3
1	1	2, 4

The key elements are the inputs marked 1 to 4 in Figure 1. In this CNOT the inputs are each represented by two photons. These allow the inputs of the quantum CNOT to be specified, as shown in Table 1.

The hardware and simulation systems are located at the University of Bristol and can be accessed with only a few seconds lag over the internet. A JSON web API is provided, which gives us full access to the CNOT. It can use any modern programming language (Mathematica, Python, Javascript, MATLAB ...) to talk to the Bristol servers through this API and get data. Below is an example API call, getting counts from the chip with all phases set to zero (i.e. the circles with floating point numbers in Figure 1 all set to 0). This is the Python code to make the call:

counts = urllib2.urlopen("http://cnotmz.appspot.com/experiment?
phases=0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0&accessToken=XXXXXXXXXXXXXXX").read()

The above calls will return data in the form:

{"counts": {"2,3": 0, "1,3": 80, "1,4": 0, "2,4": 28}, "max": 80, "sum": 108}

The call returns the number of photons detected across the output groups at the far right of Figure 1. It can be seen how these relate to qubit values using Table 1. In this example outputs 1 and 3 had a count of 80 simultaneous photons in the time segment. Changing the phase of the photons causes them to interfere destructively or constructively with each other.

We will now give a brief introduction to how photonic quantum computers function, as they are different to some other forms of quantum computing. Consider a subset of the sort of paths contained in the chip, as shown in Figure 2. The far left shows the inputs for the qubit: putting in a photon into (0) gives a qubit of value 0, an input in (1) gives a qubit of value 1. In the centre is a beam splitter, which splits the photon, and at the far right are the photon detectors that count the number of photons arriving in each path.

If a single photon is put in through (0) or (1) 2000 times, then we would expect to detect the photon half the time at the top detector and half the time at the bottom detector. Between the beam splitter and the detectors, the photon is in what is known as a superposition state, it is "blurred" across paths 0 and 1. Adding another beam splitter gives Figure 3. If a photon is sent into (0) then as a result of the extra beam splitter is will always be detected at the lower detector for the following reason. At the first beam splitter it blurs across both paths, and at the second beam splitter these blurred paths interfere with each other behaving like light waves. This interference

causes the probability of the particle being detected at the top detector to become zero. Thus the particle is always detected at the bottom detector. Technically this interference is happening to the spatial wave function.

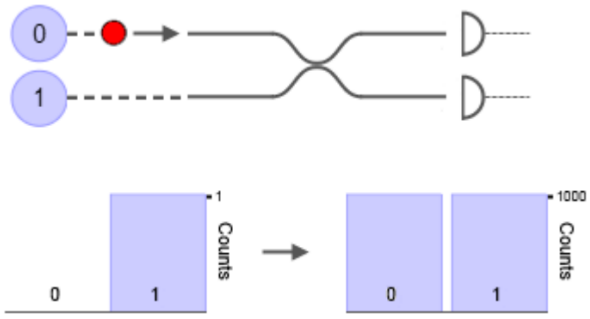

Figure 3: A simplified photon path and the result

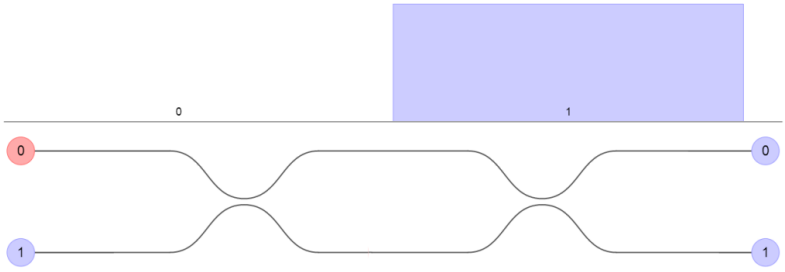

Figure 3: Photon system with an additional beam splitter.

This interference effect can thus be manipulated using the phase shifters in the waveguides. Figures 4 and 5 show what happens when a phase shifter is added. Figure 4 applies a phase shift of 0.5π radians to the "part" of the blurred photon in that waveguide (hence the number 0.5 in the circle). This causes interference effects at the second waveguide leading to photon detection happening at top and bottom detectors with equal probability.

The phase shifter in Figure 5 is set to 1π radians (hence the value 1 in the circle). This creates an interference effect in the second beam splitter that leads to the waves cancelling out for the bottom detector. Therefore, the photon will always be detected at 0. Different phase shifts would cause different probabilities of detecting the photon at different detectors. The demonstration of these interferences is a mathematical task, which – although not highly advanced – would require lengthy mathematical expansions; they will not be shown in this chapter.

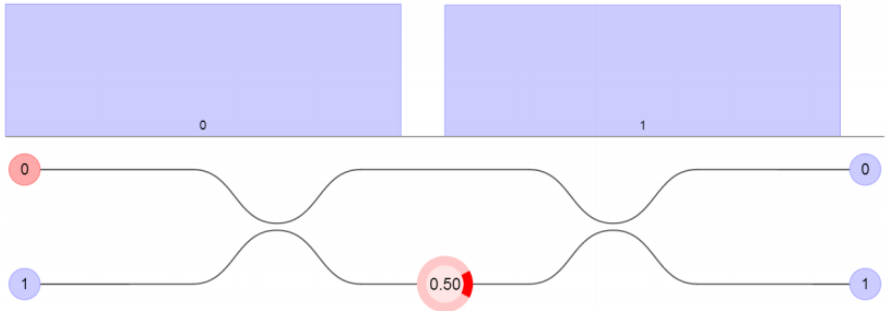

Figure 4: Photon system with an additional beam splitter.

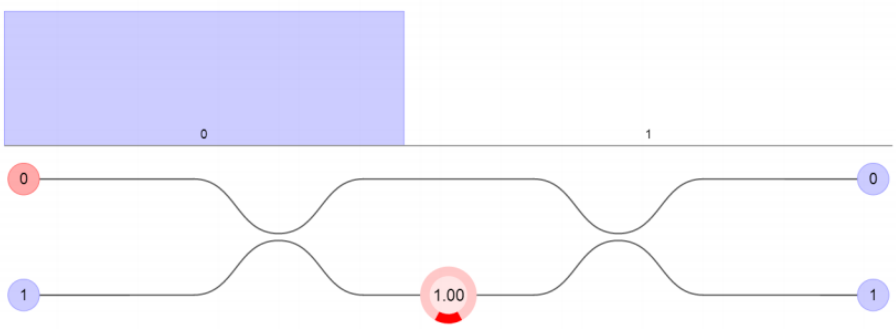

Figure 5: Photon system with an additional beam splitter.

However looking at Figure 1, and the brief description of the JSON Web API earlier, it can be seen that the phases can be set in various paths dynamically and photon counts returned, over the Internet.

The mathematical formulation of quantum mechanics – as supported by a multitude of experimental results for decades – has implications which cannot be explained in the way that we are used to explaining the physics we learned in high school and every day experience. One such implication, which most people are unaware of, is the question of what makes two particles separate from each other. For example what makes two light particles (photons) different objects? Consider the common sense notion of the separateness of say two wooden planks A and B, 100 metres apart. If you move plank A, plank B does not move. If you break plank B, plank A does not break. In fact most local actions and observations on planks A and B are independent of each other. Conversely, if you push the end of a plank, and another plank 100 metres away moves, you would assume they are connected – perhaps by a piece of string or a rod. They in effect become one object – the “plank plus string plus plank”

object. This is our common sense notion of things being separate independent objects, or one connected object.
Things are not so simple at the subatomic level. In fact it is possible to use a process to generate two photons that are as separated as it is possible for two particles to be. They could be a million light years apart, not influencing each other by force fields. Yet it can be shown by the mathematics of quantum mechanics that doing something to photon A would affect photon B. Because there is no known force field or interaction between the photons during this process, then by our common sense notion of separate objects, the photons are not entirely separate objects. But they are, clearly. They are a million light years apart with no interaction. This process of being separated particles but in some sense not separated was intolerable to Einstein; and the fact that the mathematics of quantum mechanics enabled this to happen, proved to him quantum mechanics was wrong. A methodology was found to quantify these issues as a testable inequality called Bells Inequality (Shadbolt et al. 2012). Amazingly, when experiments were done in the 1960s, it was found that this inequality could be violated, thus implying that the entanglement predicted by the mathematics happened in the real world, leading to an avalanche of philosophical debate, which still continues.

We do not wish to concern ourselves with this debate here, but wish to create a musical mapping from a quantum computer live whose results show the effects of entanglement. The quantum computer used here can generate entangled photons using beam splitters. Although the entangled photons are only separated by a tiny distance, from a physics point of view they are entirely "different planks". They have no detectable physical interaction. Yet statistically they behave as if they are connected, are part of one larger object. It is these statistics that are amplified through the computer music system.

Before explaining the mapping and control system, it is necessary to explicate an experiment that exhibits the effects of entanglement. Here is an analogical explanation of that experiment, the Prisoners' Postman: two soldiers Alice and Bob are caught and placed as prisoners of war in separate huts either side of a compound, outside of hearing range of each other. Their jailor Eve is a kindly person but likes playing games. She tells each soldier that if they can give her some wrapping paper and something to put in it each morning, then she will send it as a present to one of their families. There are gaps under the prison huts and each day there is always a 50/50 chance of Alice and Bob both finding either a stone or some old newspaper within a hands grasp. So once Alice and Bob have chanced across one or the other, Eve will ask each for the address of one of the families. Alice can give her own address or Bob's, and Bob can give his own address or Alice's (they know each other's because they are old comrades-in-arms). But neither can know what the other has said. As long as they don't both find only pebbles, and they both choose the same address, Jailor Eve will use one of the pieces of newspaper as wrapping and send the other item (be it pebble or scrunched-up newspaper) to that address. This as a sign to their family they're alive and okay. If they choose different addresses, Eve will not send the package, except… Jailor Eve is as bored of hanging around the compound as Alice and Bob, so she invents a twist to make the daily game more interesting.

Even if Alice and Bob provide different addresses, there is one case where Eve will still send a package. This is if Alice and Bob both fail to provide wrapping paper, i.e. they find only pebbles. In that case Eve will find some newspaper and wrap both pebbles for them and send the package randomly to one of the families. So to summarize: Alice and Bob get to send out a package to one of their families either if they both pick the same address and at least one of them finds wrapping paper, OR if they pick different addresses and they both only find pebbles. The question is: assuming that each has a 50/50 chance each day of finding a pebble or newspaper within reach, what strategy should Alice and Bob following in choosing addresses to increase the chance that at least one letter is sent? Oh and they do get a chance every so often to set a strategy, because extra prisoners come in in transit, so Alice and Bob are placed in the same hut for that one day, and then returned to isolation from each other.

If Alice and Bob pick a random strategy, i.e. they randomly select an address whether they find paper or pebble, on average they will send out 1 letter every two days – i.e. a 50% a day probability of success. If, after being in a hut together, they agree to select only Alice's address for 7 days, and then only Bob's address for 7 days, this will increase to a 75% a day probability of a parcel being sent. In fact both agreeing to select the same address at the same time is the optimal strategy. Over the years, if Alice and Bob try different strategies, they will still hit the upper maximum of 75%, because they cannot communicate before choosing the addresses. They are on different sides of the compound. It can be proved that without communication, the limit is 75% for any strategy because Alice's knowledge is local to her, and Bob's is local to him.

To understand how this relates to the entanglement experiment, consider two photons generated by a beam splitter so they are entangled. After the photons have separated – Alice performs an operation on the photon based on her chosen address and whether she's found a rock or a paper. Bob does a similar thing on his photon. So the state of the two photons now fully describes whether Alice and Bob can win and get a package sent. But neither photon can communicate or affect the other. When the photons are observed at the detectors, you would expect them to be in win state 75% of the time. In other words, there should be nothing Alice and Bob can do to win more than 75% of the time.

However, it turns out that if Alice and Bob do the right experiment on the photon, then the photons are in a win state 85.36% of the time (to be precise it can be shown to be $0.25*[2+2^{0.5}]$). There is a ten-point increase in the chance of Alice and Bob getting to send a package. If the entanglement is removed, or Alice and Bob go back to normal strategies, the probability goes down to 75%. In this case the photons are on a single quantum computer chip, but the experiment has been performed with photons on separate islands, and this increase has still been observed. The mathematics implies what Einstein called "spooky action at a distance" faster than the fastest possible speed in the universe (the speed of light).

The phenomenon of entanglement is at the heart of traditional quantum computing, and relates to instantaneous statistical correlations between measurements, even when they are physically separated and have no causal connection. One methodology used to quantify incidences of entanglement is by Bells Inequality (Shadbolt et al. 2012). What will be partially explained here is the CHSH inequality (Clauser 1969) a more practical form of Bell's ground-breaking work. The CHSH methodology defines four methods of measuring simultaneous photon events across outputs 1 to 4 in Figure 1. These four methods involve setting all the phase shifters on the right hand side of Figure 1 to four different sets of values (the values will be described later). For each set, an experiment is run. The four sets are usually labelled A, A', B and B'. CHSH says that if reality is local then for the two measurements with settings A, A' and B, B':

$$CHSH = E(A, B) + E(A, B') + E(A', B) - E(A', B') <= 2 \quad (1)$$

where E is the quantum correlation, defined for the qubits in Table 1 as:

$$E = (N_{00}-N_{01}-N_{10}+N_{11})/N_{Total} \quad (2)$$

where N is the count at the detectors of the detected qubits. To investigate this with the photonic quantum chip, the four measurement configurations in Equation 1 are activated by setting the last four phase shifters on the right hand side of the schematic in Figure 1. A and A' are the two settings for the top two, B and B' the settings for the bottom two. The quantum correlation in Equation 2 is then given by:

$$E = 9[P(1, 3)+P(2, 4)-P(1, 4)-P(2, 3)] \quad (3)$$

where:

$$P(x, y) = N(x, y) / [N(1, 3)+N(2, 4)+N(1, 4)+N(2, 3)] \quad (4)$$

Where $N(x,y)$ is the number of coincident photons counted at x and y detectors in Figure 1, in the same time segment. The reason for multiplying by 9 is "post-selection". The chip has 9 more output states that the qubit states, so we throw these away and multiply up the output states of interest.

Most phase values for A, B, A' and B' will lead to the CHSH value in Equation 1 being less than or equal to 2, thus satisfying local realism. However the following settings violate classical local reality (in other words lead to instantaneous correlations between physically separated and non-communicating qubits). First set the left hand side phases as shown in Figure 8: 1.5, 1.0, 1.5 and 0.0 (i.e., 1.5π, π, 1.5π and 0 radians).

Then for A set the top two phase shifters on the right hand side to 0.5 and 1.75, for A' set them to 0.5 and 1.25. For B set the lower two on the left hand side to 0.5 and 1.5, and for B' to 1.0 and 1.0. The settings for [A,B], [A,B'], [A',B] and [A',B'] are shown in Figures 9 to 12 respectively. Having set these up, if you actually run

sufficient experiments on the photonic computer and take the average value of the CHSH, it will be closer to 2.6 than to 2. The theoretical limit can be shown to be 2*sqrt(2).

In fact these detector settings give the maximally entangled states for the photons, i.e.; qubits. Figure 11 shows how adjust the phases on the right hand side of the schematic can move the CHSH values between the classical limit and outside the classical limit.

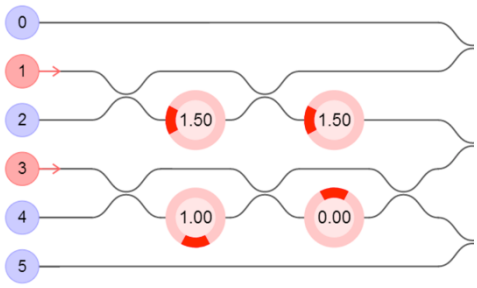

Figure 6: Fixed settings of first four phase shifter.

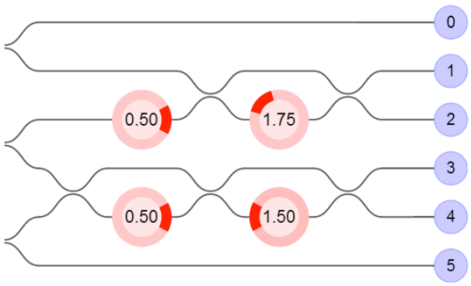

Figure 7: Phase shifter settings for E(A,B), in Equation 1.

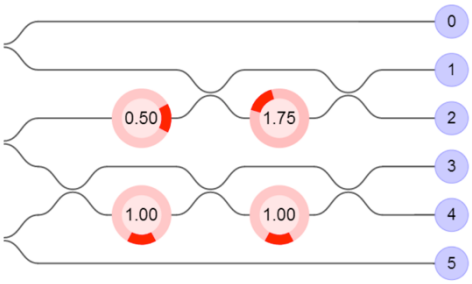

Figure 8: Phase shifter settings for (A,B') in Equation 1.

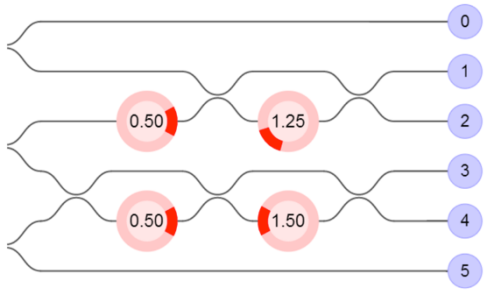

Figure 9: Phase shifter settings for E(A',B) in Equation 1.

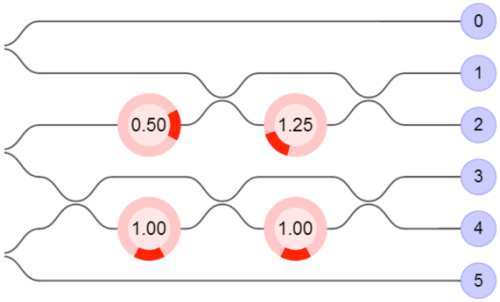

Figure 10: Phase shifter settings for E(A',B') in Equation 1.

There are a number of simulated quantum computers available on the Internet (Quantiki Wiki 2015). However the uniqueness of the Bristol computer is twofold. Firstly it is a photonic quantum computer, whose inputs and outputs are based on simultaneous photon arrival rates. This "tempo" of photon detection fits well with PMAP (Pulsed Melodic Affective Processing), which will be explained later in this next section. Secondly the simulator is actually designed - by hardware experiment and theory - to accurately reflect the hardware system and its interface.

Thus by designing any system using their online simulator, then once the hardware system has been re-commissioned, the code can simply have its call structure replaced and it will become a hardware QC application.

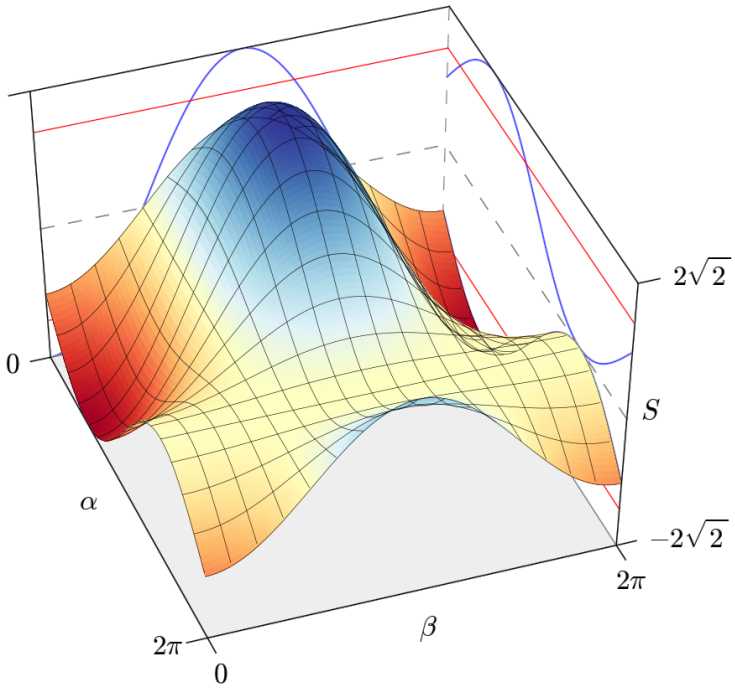

Figure 11: How phase changes effect the Bell CHSH number calculated

A number of musifications have been done of the simulated QC at Bristol. For example, an orchestral simulation that sonifies calculations based on simultaneous photon counts from the four outputs. An output of the musification algorithm is shown in Figure 12. The experimental parameters from which the lines are calculated are labelled 'BN' for Bell CHSH number, 'cb' for simultaneous photon count quantum expectation and 'c' for the simultaneous photon count. There are two Bell Number lines: one with shorter durations, the first line, and one with longer durations in the second line. These are both calculated from the same Bell Number, but differentiated to give the desired timbral, pitch and rhythmic effects.

For example the longer BN pitch is calculated as MIDI[1] pitch 42 minus the Bell Number for 5 bars, and then 40 minutes the Bell Number for the next 5 bars after that. The shorter BN pitches are calculated alternately as MIDI pitch 72 plus the Bell number times 4, and MIDI pitch 70 plus 4 times the Bell number, for all 10 bars. For each Bell number value, 10 bars of music are generated. Similar calculations are performed for the c and cb lines: e.g. the pitches for the c line is calculated deterministically from c13, c23, c14, and c24 in that order.

All pitches are moved to the nearest note in C major or minor before being finalised. The choice of major or minor depends on the values of c13, c23, c14, and c24. Within certain bounds of these values the minor mode is chosen, otherwise the major mode is chosen. An example of the effect of this sonification, as the QC is forced towards

entanglement through gradual phase adjustments, is available on SoundCloud (Kirke 2015a). Note that this particular example involves the manual orchestration of the four lines in Figure 12 across an orchestral simulator.

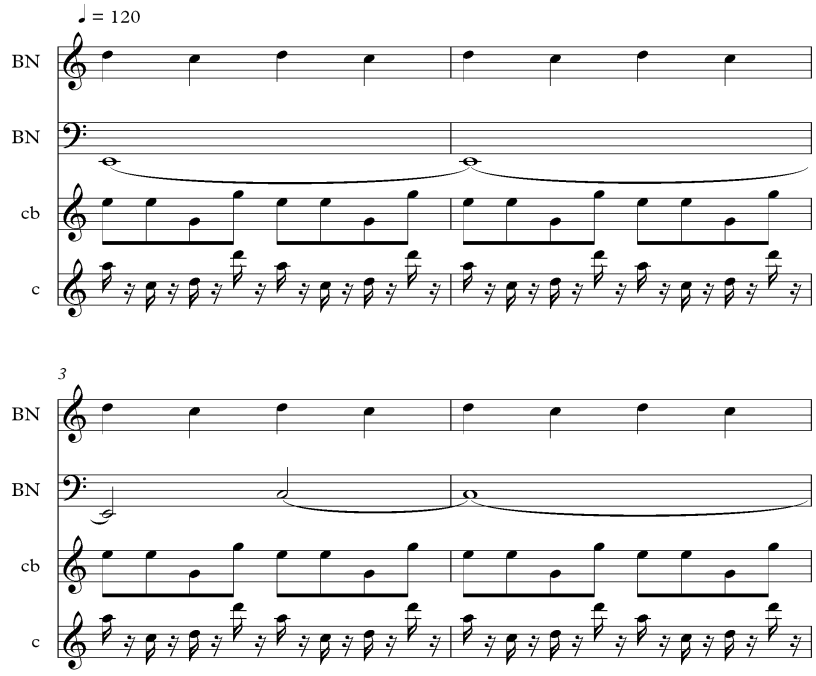

Figure 12: Example of the output of the musification algorithm.

5.3 Controlling the QC with Music

In addition to the musification, a virtual machine was designed that enabled the QC to driven towards entanglement using a form of computation based on music.

The concept of the Virtual Machine has been around for many decades (Goldberg 1974). The best-known virtual machines are probably the Java Virtual Machine (Freund and Mitchell 1999) and those that allow Apple users to run Microsoft Windows (VMWare Inc. 2015), and Docker for Linux (Seo 2014). These virtual machines allow the execution of software, which was either not designed for the supporting hardware or is not designed for any particular supporting hardware. Some of the most pervasive virtual systems are the server virtualization used in cloud computing services; this allows one computer to think it is multiple computers, each with their own OS (Guohui and Ng 2010). Some of the implementations of this approach use up server processing in the virtualization process (Huber et al. 2011), but this is felt to be outweighed by the advantages it brings. One recent and successful

example of a research virtual machine is the neural network spread across multiple machines in Stanford's Deep Learning system (Coates et al. 2013). The purpose of this virtual machine is to speed up operations on the neural network training at a lower cost.

The virtual machine referred to in this chapter is not aimed at reducing computation time. It is more in the philosophy of the Mouse and Graphical User Interface – also known as Window, Icon, Mouse and Pointer (WIMP). WIMP did not increase processing power on computers when they were added. In fact they reduced it (Garmon 2010) because of the requirements for bitmapped screens and windows and so forth. However there are many tasks now that would be unfeasibly slow for us without WIMP. Furthermore there are some tasks, which would be unimaginable without WIMP, for example modern digital photo manipulation. Similarly high level programming languages at first seemed much slower than machine code. However they opened up the world of software to powerful new ways of thinking which could never have been achieved by the machine code programming approach. Changing the mode of Human Computer Interaction opens up opportunities for increasing the usefulness of computers, even though it can on the lowest level slow it down.

Given the growth in virtual computing, unconventional computing has the opportunity to greatly expand its possible modes, limiting computation only by imagination; hence the field of Unconventional Virtual Computation (UVC). There has been a significant amount of work using simulation to run unconventional computation; however these have been designed to simulate a hardware or wetware system (Jones 2010; Bull et al. 2008; Spector et al. 1999).

An implementation of a hybrid unconventional computation system is described below: a virtual PMAP processor (Kirke and Miranda 2014) linked to the photonic quantum system. The PMAP processor will be used to control the Bristol photonic quantum system, driving it towards a maximally entangled output state.

5.4 HYBRID UVC AND QC

It has been shown previously that the Quantum Computer in the Cloud can achieve entanglement in the CNOT gate (Shadbolt et al. 2012) and also that PMAP can be used to control non-musical adaptive processes (Kirke and Miranda 1999; Kirke and Miranda 2014). The interest here is to examine how the two can be combined to create a form of Hybrid Computer which is part Unconventional Virtual Computation and part Quantum Computation. It is particularly of interest utilizing PMAP and a photonic QC. Being essentially spike-based, PMAP is particularly suited to dealing with data which is rhythmic and has a tempo. Simultaneous photon arrival counts have some of these properties. To examine this link in the first place, the simplest possible process will be created. A PMAP circuit will be designed which attempts to move the QC towards entanglement and keep it there. Although this in itself has no

explicit computational usage, it would be a demonstration of the ability of UVC (PMAP) to interface with work with a quantum computer. The circuit in Figure 13 is the basis for this. The design process behind this circuit will be explained.

During the following experiments, the input of the QC is held at simultaneous photons on 1, 3 (input 0, 0). The output counts of photon simultaneous arrivals are sampled every second and these become the counts S_{13}, S_{23}, S_{14}, and S_{24} seen in Figure 14. These arrival rates are converted by a simple linear transform into tempos for a PMAP stream. So the higher the output count, the higher the music tempo. The pitches of the PMAP streams do not vary in this calculation, and so are simply just a repeating figure consisting of middle C and middle E. So S13, S23, S14, and S24 will be PMAP streams of tempos proportional to C13, C23, C14, and C24 respectively, each with a pitch form [C,E,C,E,C,E, …] or [1, 4, 1, 4, 1, 4, …].

Looking at Equation 5, the combination of counts is replaced by musical ANDs (MANDs) done on the four PMAP streams. The negative signs in the combination are replaced by a musical NOT (MNOT) on the negative input of the MAND gate. Thus Equation 5 becomes:

$$E = MAND(S_{13},S_{24}, MNOT(S_{14}), MNOT(S_{23})] \quad (5)$$

which is what is implemented in the top part of Figure 1. However then the calculation of the entanglement measure requires four such calculations like Equation 5 to be performed with the different phase shifter settings in Figures 7, 8, 9 and 10.

The MSHIFT object is simply a musical shift register. So at each of the four calculation step it stores an output from the MAND gate being input to it. This is synchronized with the phase change process. So for the each of the four phase value sets required to calculate the Bell CHSH number (Figures 7, 8, 9 and 10) the melody from Equation 5 is stored in the MSHIFT register. Equation 1 (for calculating Bell CHSH) is approximated using the MAND for combination, and the MNOT for negative values, giving Equation 6.

$$CHSH\text{-}PMAP = MAND(MNOT(MSHIFT1), MSHIFT2, MSHIFT3, MSHIFT3) \quad (6)$$

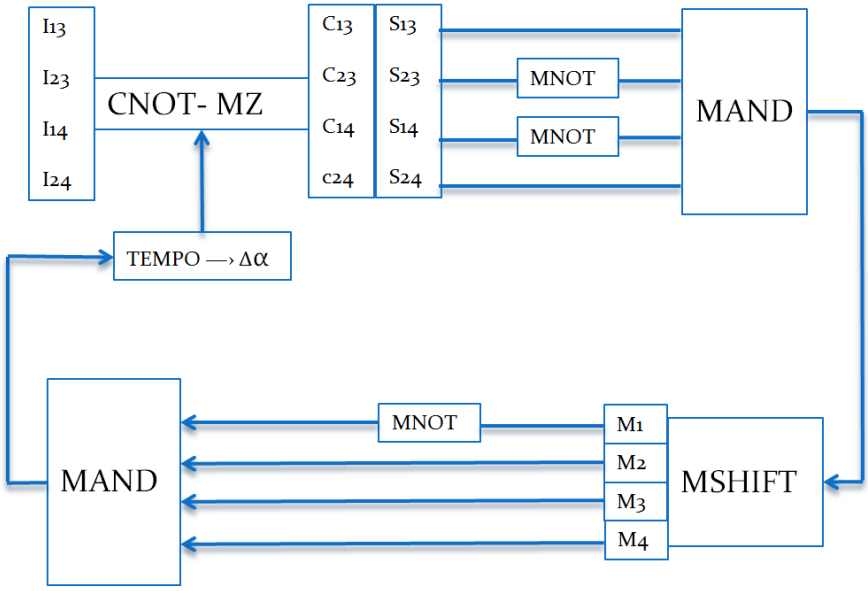

Figure 13: Circuit which is the basis for an example PMAP / QC hybrid process

This produces a PMAP output whose tempo is then used in the "Tempo → Δα" object to calculate a phase shifter setting for each of the four sub-experiments. The phase shifter which is adjusted by the PMAP circuit is that seen in the top left of Figures 7, 8, 9 and 10. This corresponds to α in Figure 11. It can be seen the optimal value is 0.5 (i.e. 0.5π) to maximize entanglement. So each time CHSH-PMAP is calculated, it will result in a melody whose tempo is converted into a change delta as follows. If the tempo of the current CHSH-PMAP stream is greater than the tempo of the previous CHSH-PMAP stream, then the delta is left unchanged. *If the tempo is less than the previous tempo*, then the delta is adjusted as in Equation 7.

$$delta \rightarrow -0.5delta \tag{7}$$

Then at each iteration we have:

$$\alpha \rightarrow \alpha + delta \tag{8}$$

If the PMAP circuit based on Equations 5 and 6 is somehow representing the CHSH calculation in Equation 1, then increases in tempo should be correlated to increases in entanglement and vice-versa. Thus the effect of Equations 7 and 8 should be to cause α to converge to the point of maximum entanglement, i.e 0.5π.

To explain this more clearly. Suppose that α is set to 0.6π and *delta* is set to 0.1, and the first calculation (where α = 0.6π) gives a tempo of T1 from Equation 6. Then the

delta is applied to give $\alpha = 0.7\pi$, i.e. $(0.6+delta)\pi$. Suppose that gives a tempo of T2 from Equation 6.

If T1 < T2, i.e. if the tempo is increasing, then the next value of α will be $\alpha = 0.8\pi$. But if T2 < T1, i.e. if the tempo decreases, then *delta* = -0.5**delta*, i.e. *delta* = -0.5*0.1 = -0.05. So the next value of the phase shifter will be $\alpha = (0.7-0.05)\pi = 0.65$. Then if the next tempo T3 is such that T3 > T2, *delta* will remain unchanged, so the next value will be $\alpha = (0.65-0.05)\pi = 0.6$. This is the algorithm that is being implemented by the "Tempo → $\Delta\alpha$" object in Figure 15. It is claimed that the combination of this object, and the PMAP and the QC system are sufficient to move the qubits to close to maximum entanglement, and to keep them there.

If the PMAP circuitry consistently approximates the Bell CHSH calculation, then the system will clearly always converge – as it is using a form of gradient descent with decreasing gradient; and Figure 11 shows there are no local maxima. The maximum that the simulation can reach is a Bell Value of approximately 2.54. It was indeed found that the QC / PMAP system always converges to near a Bell CHSH value of 2.54, with the α values close to 0.5 + 2.N for N = 0,1,2,…, because the maxima in Figure 11 actually occur at $\alpha = (0.5 + 2.N)\,\pi$. Thirty examples were run, each starting at a random α between 0 and 2π, and with a random initial delta between 0 and 1. Examples of 4 such runs are shown in Figure 14.

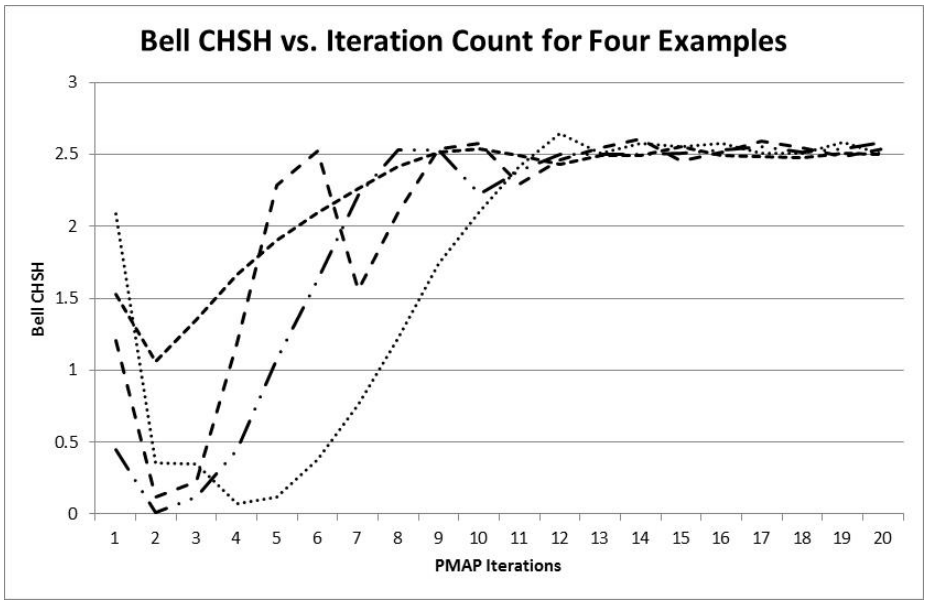

Figure 14: Four examples of Bell CHSH convergence using PMAP

It can be seen in Figure 14 that the examples are clearly converging to a value just over 2.5 (the maximum possible being approximately 2.54). This convergence is

driven by the changing tempo on the output of the PMAP circuit (the output of the second MAND in Figure 13). The tempos for these four examples can be seen in Figure 14. The second MAND output for eight iterations of an actual convergence example is available on SoundCloud (Kirke 2015b).

Looking at twenty eight of the thirty example runs, average errors during convergence are shown in Figure 16. The reason only 28 are included is that two of the runs became stuck in local maxima. Run 12 stuck at Bell CHSH value between 2.4 and 2.5, corresponding to phase alpha of around 0.6π rather than 0.5π. So this was still close to maximum entanglement. However run 13 became stuck in a local maximum a long way from the quantum region: a Bell CHSH of 0.7, which corresponded to a phase α of around 1.84π.

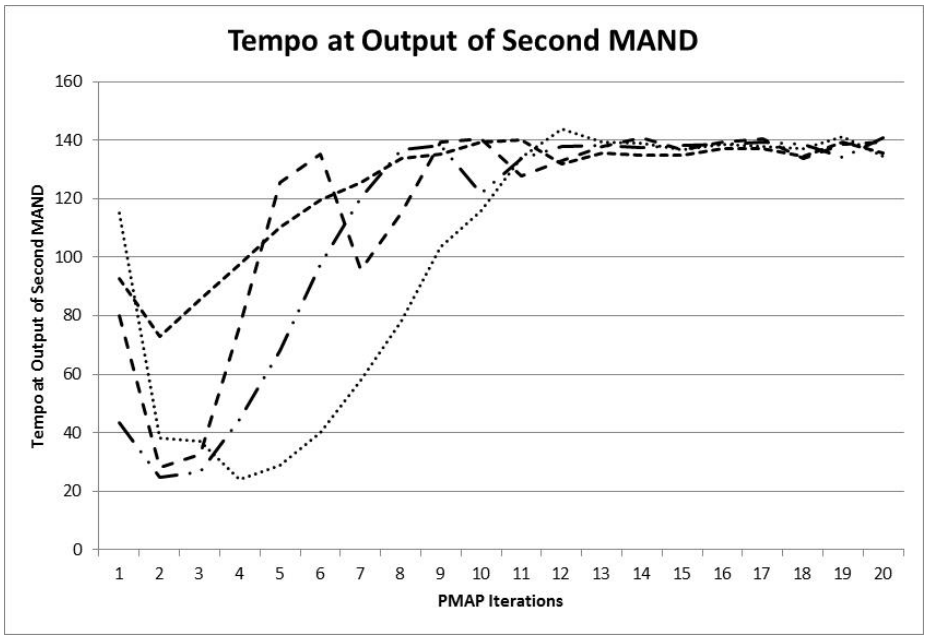

Figure 15: The four examples of Bell CHSH convergence using PMAP from Figure 14, shown with their tempo outputs on the second MAND in Figure 13.

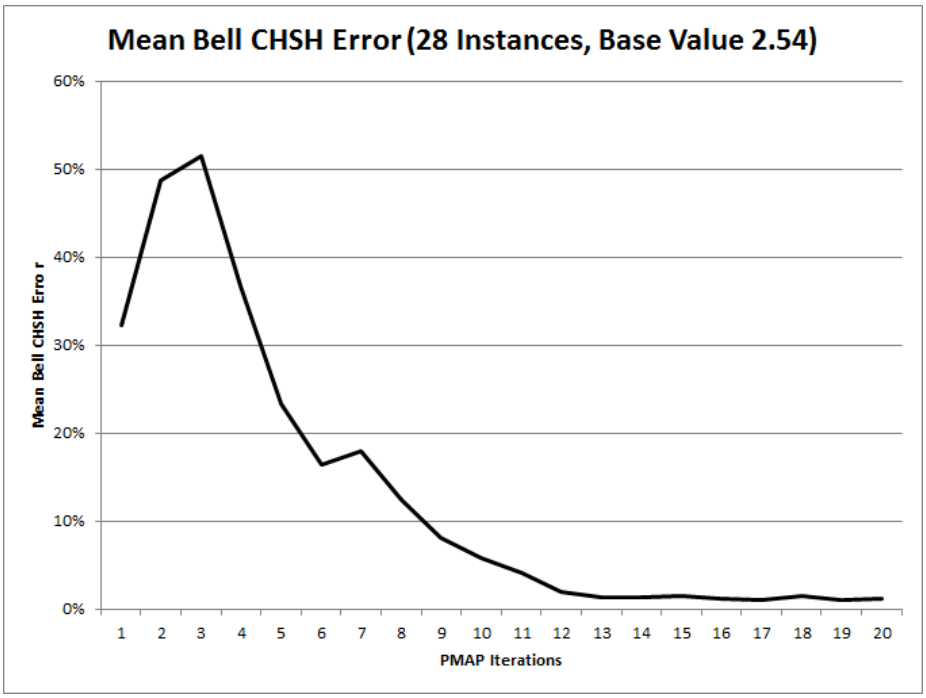

Figure 16: The mean error during convergence across 28 examples run, across twenty iterations.

These local maxima in the PMAP system, which are not present in the quantum computer, are not surprising, given how different the modalities are in the calculation: quantum versus PMAP. What is surprising is that only two of them were found in 30 runs, and that only one of them actually "broke" the functionality of the system. Even with the outliers included, the average Bell CHSH value across all 30 runs after 20 iterations was 2.45 and standard deviation 0.34, with most of the standard deviation coming from a single outlier.

A form of attractor diagram is show in Figure 17 – plotting the average phase alpha (divided by 2π) during convergence against the mean tempo on the output of the PMAP circuit – for the 23 examples that converged to 0.5π. The average starting point is around 0.7 and all then converge to an α of around 0.5π at a tempo just under 140. For five of the examples, the starting point and delta meant that it converged to 2.5π rather 0.5π, which is also the same maxima of the Bell CHSH curve. These are in shown Figure 20.

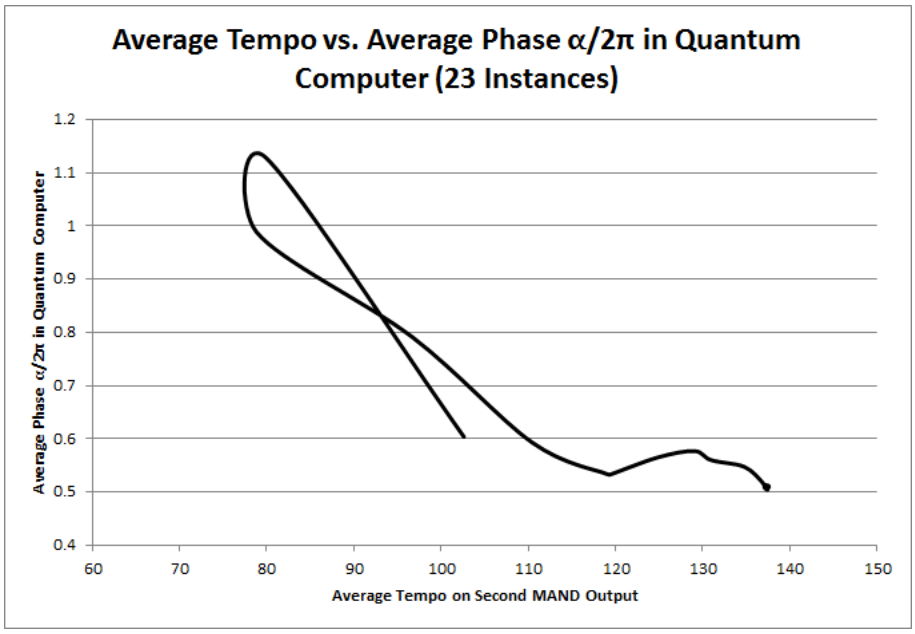

Figure 16: Mean Tempo vs. Mean Phase α/2π for 23 of the 30 examples

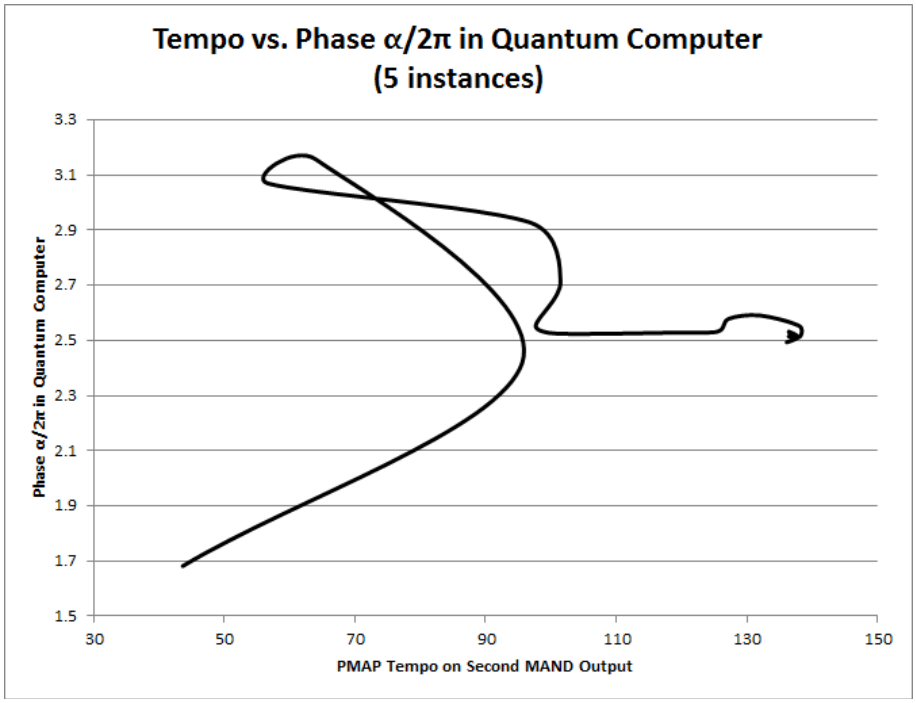

Figure 18: Tempo vs. Phase α/2π for 5 Examples which converge towards 2.5

Given the non-triviality of the Bell CHSH calculation, it is unlikely that the PMAP circuit is causing the convergence by chance. However to further examine this, various elements in the PMAP circuit were adjusted to confirm the system did not converge for any similar PMAP circuit. Firstly the MNOT after the MSHIFT register from Figure 13 was removed. This was then replaced, and the first MAND was changed to a MOR. Ten runs were done for each of these conditions. For the first the average final Bell value after 20 iterations was 0.85 with standard deviation 0.72. For the second it was average value 0.94 with standard deviation 0.89. This compares unfavourably with the 30 runs with the Figure 13 set-up – the average bell value (including the outlier) was 2.45 and standard deviation 0.34.

In terms of PMAP's originally envisioned functionality – giving insight into the computation process – do the PMAP melodies give insight into what is occurring? Suppose two virtual "probes" are placed into the circuit in Figure 13 at the output of the first MAND and the output of the second MAND gate. At maximum bell value achievable – 2.538 – the tempo on the output of the first MAND is 142, and the tempo at the output of the second MAND is also 142. Away from the maximum bell values these tempos are different. Also the further away the PMAP system is from entangling the photons the lower those tempos are. This means that listening to the PMAP data at these two points can have three effects:

- It gives an insight into the process development – the closer the system comes to entanglement, the more in synch the two PMAP streams will sound.

- It gives an emotional insight into what is occurring: higher tempo in a major key communicates higher "happiness" (Kirke and Miranda 2014). Thus the closer the system gets to fulfilling its aim (maximum entanglement of qubits) the happier it sounds. This fulfills one of PMAPs aims to give an emotional insight into calculations.

- It gives an insight into the process itself. A non-expert might observe that for the system to achieve entanglement requires the outputs of the two MANDs to be higher and more similar tempos. So it might start the non-expert thinking along the lines of what photon output counts would lead to this tempo. This provides another model of considering the nature of the entanglement equations, which may be more understandable to an individual unfamiliar with quantum computing.

Although this particular PMAP process is not a practical one – we know how to entangle qubits – it gives a demonstration of how more useful processes could also be given greater transparency thanks to unconventional virtual computing, as well as how UVC can be combined with other forms of computation to give consistent functionality.

The purpose of this section of the chapter has been to introduce the concept of Unconventional Virtual Computing (UVC). In particular a form of UVC has been introduced called PMAP. It has been applied in a simple hybrid system involving a UVC and a simulated photonic quantum computer. The UVC successfully kept the quantum computer qubits in a state of entanglement using gradient descent. It also showed how UVC can give insight into the computation going on, in novel ways.

It is interesting to note that this not the first time sound has been used in quantum computing: researchers at the University of Bristol Centre for Quantum Photonics have used a system involving photo-diodes, a tone generator and a loudspeaker to detect the location of Hong-Ou-Mandel dips (Hong et al. 1987) in hardware photon beam splitters. However – and putting aside any contributions to UVC - to build a music-based system which can interface with real single photons, real quantum systems which we can't see and which are actually entangled, is to our knowledge a new contribution.

This part of the chapter has been written from the point of view that with the increasing virtualization of computers, and the recognition that this year's virtual computers are as fast as the hardware computers of 10 years ago, it is becoming clear that we are only limited in our modes of computation by our imagination. Given that improvements in computer efficiency are not always due to increasing computation speed, UVC has the potential for speeding up working with computers by making their processes more human-understandable.

In terms of speeding up hardware computers themselves, the next section is about commercial quantum computers that are showing potential for large speed increases.

5.5 ADIABATIC QUANTUM COMPUTING

As of the time of publishing this chapter, there is only one company making quantum computers available commercially (Warren 2013). These computers are based on adiabatic quantum computing (Albash et al. 2015). Quantum computers discussed so far have focused on quantum implementation of Boolean logic gates such as AND, OR and NOT. An adiabatic quantum computer implements a different form of computation reminiscent of connectionist computing: what is known as an Ising model (Lucas 2013). Ising models were originally used to describe the physics of a magnetic material based on the molecules within it. As well as electrical charge, each of these molecules has a property known as spin; their spin can be +1 or -1. The total energy $E(\boldsymbol{s})$ of the a collection of molecules with spin $\underline{s_i}$ can is modeled by:

$$E(\boldsymbol{s}) = \sum_{(i,j)\in \text{neigh}} J_{i,j} s_i s_j + \sum_{i \in V} h_i s_i \qquad (9)$$

s_i is the spin value of molecule i, and h_i represents the external magnetic field strength acting on that molecule. J_{ij} represents the interactions between each molecule and its nearest neighbours. An adiabatic quantum computer attempts to find values of s_i to minimize the value of $E(\boldsymbol{s})$, for a given set of of h_i and J_{ij}. The user sets the values of h_i and J_{ij}. Such a minimizer can be implemented using non-quantum hardware. However significant speedups are expected through the use of quantum hardware. Such hardware is now being sold by the Canadian company D-Wave.

There is an ongoing debate about how the D-Wave adiabatic computer truly functions and what speedup it can provide; but recent results have suggested large speed increases for quantum hardware (Neven 2016). This is thought by some to be due to quantum tunnelling (Katzgraber 2015). When searching for low E(s) states, a quantum system can tunnel into nearby states. Quantum tunnelling allows physical systems to move to states in ways that would not be possible in the classical Newtonian view of the world. The systems "tunnels" through to the new, usually inaccessible, state instantly.

On the face of it, it may not seem significant that quantum computers can be built to solve only one this problem type. However over a period of 28 years, more than 10,000 publications came out in areas as wide as zoology and artificial intelligence on applications of the Ising model (Bian et al. 2010). Any problem that can be modeled using elements interacting pairwise with each other, and involves minimizing some measure of the interaction, has the potential for being modeled as an Ising problem.

To understand how to formulate problems on an adiabatic quantum computer, a simple musical 8 qubit problem. Looking at the Ising Equation 9, if the s_i are replaced by q_i (as is usual in D-Wave notation) then the equation would be written, for 8 qubits:

$$\begin{aligned} E(\boldsymbol{q}) = {} & h_1q_1 + h_2q_2 + h_3q_3 + h_4q_4 + h_5q_5 + h_6q_6 + h_7q_7 + h_8q_8 \\ & + J_{1,2}\,q_1q_2 + J_{1,2}\,q_1q_3 + J_{1,4}\,q_1q_4 + J_{1,5}\,q_1q_5 + \ldots \\ & + J_{2,1}\,q_2q_1 + J_{2,3}\,q_2q_3 + J_{2,4}\,q_2q_4 + J_{2,5}\,q_2q_5 + \ldots \\ & \ldots \\ & + J_{8,1}\,q_8q_1 + J_{8,2}\,q_8q_2 + J_{8,3}\,q_8q_3 + J_{8,4}\,q_8q_4 + \ldots + J_{8,7}\,q_8q_7 \end{aligned} \tag{10}$$

D-Wave computers use what is known as a chimera graph (Isakov et al. 2010), a portion of which shown in Figure 18. Looking at the bottom left of the Figure 18, an 8 qubit subset q_1 to q_8 can be seen. The complete connection equation for this subset of 8 qubits is actually:

$$\begin{aligned} E(\boldsymbol{q}) = {} & h_1q_1 + h_2q_2 + h_3q_3 + h_4q_4 + h_5q_5 + h_6q_6 + h_7q_7 + h_8q_8 \\ & + J_{1,5}\,q_1q_5 + J_{1,6}\,q_1q_6 + J_{1,7}\,q_1q_7 + J_{1,8}\,q_1\,q_8 \\ & + J_{2,5}\,q_2q_5 + J_{2,6}\,q_2q_6 + J_{2,7}\,q_2q_7 + J_{2,8}\,q_2\,q_8 \\ & + J_{3,5}\,q_3q_5 + J_{3,6}\,q_3q_6 + J_{3,7}\,q_3\,q_7 + J_{3,8}\,q_3\,q_8 \\ & + J_{4,5}\,q_4q_5 + J_{4,6}\,q_4q_6 + J_{4,7}\,q_4\,q_7 + J_{4,8}\,q_4\,q_8 \end{aligned} \tag{11}$$

There are no cross terms between the first four qubits or the last four.

Note there are three D-Wave models: the D-Wave One, D-Wave Two and D-Wave 2X - with 128, 512 and 1000+ qubits respectively. All results in this chapter are run on D-Wave 2X hardware. All three models use the same chimera graph structure.

5.6 Harmonising Music with D-Wave

A basic harmony tool will be presented to help demonstrate D-Wave programming. Called qHarmony, it will generate options for a set of white piano notes that can be constructed as a "reasonably" assonant chord, and which can harmonize a user-provided white piano note. This problem will be approached by mapping the notes of the scale of C Major to qubits,. The qubits connections in the D-Wave will be designed so that qubits representing notes that are closer together on the keyboard, contribute to a higher energy than qubits representing notes that are further away from each other on the keyboard. Before designing the connections, the chimera graph must be addressed. Because of the chimera graph, it is not so intuitive to consider a note-qubit mapping such as:

$$[c, d, e, f, g, a, b, C] \text{ <--> } [q_0,q_1,q_2,q_3,q_4,q_5,q_6,q_7]$$

The left hand side of this mapping refers to the eight-note C major scale, with 'C' referring to the note an octave above 'c'. In such a mapping it would be natural to think of qubits q_0 and q_1 as being "close" on the piano keyboard. However they are not far or close, they are not connected at all in the chimera graph. Qubits q_0 and q_4 are connected in the chimera configuration in Figure 19. Similarly qubits q_1 and q_2 are not connected, however qubit q_1 and q_5 are connected. Looking at Figure 19 it can be seen that q_0 connects to q_4, q_4 connects to q_2, q_2 connects to q_5 and so forth up to q_7. Thus the qubit to note mapping is done as follows:

$$[c, d, e, f, g, a, b, C] \text{ <--> } [q_0,q_4,q_1,q_5,q_2,q_6,q_3,q_7]$$

This mapping can be represented as what qHarmony calls an Adjacency Function $A(n)$:

$$A(n) = \begin{cases} 2n & ; n < 4 \\ 2(n-4)+1 & ; otherwise \end{cases} \quad (12)$$

In this equation, the n-th note in [c, d, e, f, g, a, c, C] will map to the $A(n)$-th qubit. Using an adjacency function allows a simpler equation to be written for the J values that are sent to the D-Wave. Consider the following settings for J:

$$J_{i,j} = 7 - 2*abs(A(i)-A(j)) \quad (13)$$

$$h_i = 0$$

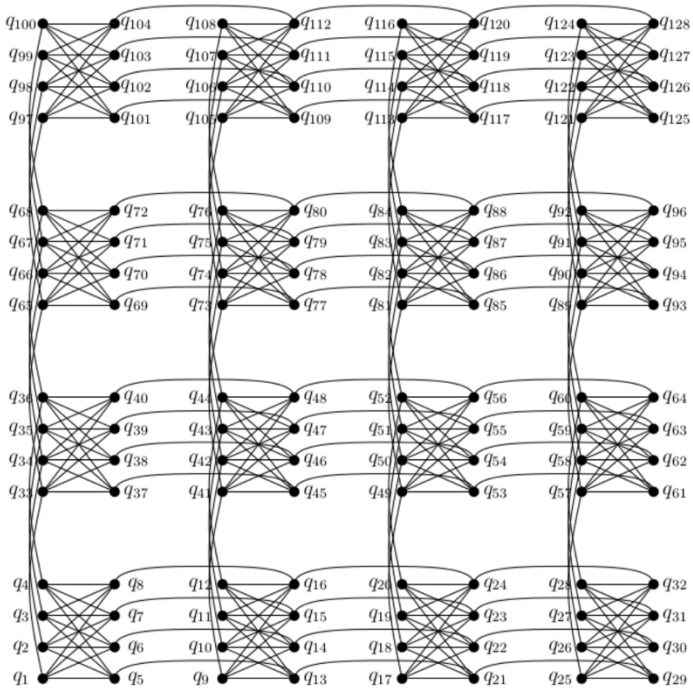

Figure 19: Part of a chimera graph for a D-Wave computer

To see the effect of the equations look at Table 2 and the resulting Table 3. If the qubits for notes c and d are both switched on, then by Equation 11, a value of 5 will be added to the energy. Whereas if note 0 (c) and note 7 (C) are switched on, a value of 7 will be subtracted from the energy. Now because the *hi* are all 0 and the system is trying to minimize the sum $E(q)$, it will "prefer" the interval [c, C] to [c, d], because it leads to a lower energy. The second interval would be considered less consonant by most than the first interval. Similarly it will "prefer" the interval [d, b] to [b, C] (most would consider the first interval to be more consonant than the second). This highlights the basis of the algorithm - attempting to create reasonably consonant chords by reducing the occurrences of too many notes that are only a tone or a semi-tone apart.

Table 2: Relative energy contributions when qubit pairs are switched on

	q_0	q_1	q_2	q_3
q_4	5	5	1	-3
q_5	1	5	5	1
q_6	-3	1	5	5

$\mathbf{q_7}$	-7	-3	1	5

Table 3: Mapping of Table 2 to musical note labels

	c	**e**	**G**	**b**
d	5	5	1	-3
f	1	5	5	1
a	-3	1	5	5
C	-7	-3	1	5

To actually use qHarmony system, some of the h_i values need to be set to non-zero before calling the D-Wave 2X. The values are set based on the notes that it desired for the D-Wave to harmonize with. Suppose you wish to harmonize with the *n*-th note of the scale, then the complete set of equations are:

$$J_{i,j} = 7 - 2*abs(A(i)-A(j))$$

$$h_{A(i)} = \begin{array}{ll} -7 & ; i = n \\ 1 & ; otherwise \end{array} \qquad (14)$$

The effect of this setting of $h_{A(i)}$ is to give the *n*-th note of the scale (the *A(i)*-th qubit) far more influence on the energy calculation. If the D-Wave 2X switches on the related qubit, it will potentially reduce the energy far more than switching on other qubits.

The *n*-th note selected in Equation 14 is called the Input Note. Different input notes lead to the D-Wave 2X outputs shown in Table 4. The D-Wave can be made to return multiple results in order of increasing energy, which is how they are shown in the table. Note that Table 4 has already been translated from qubits to notes, using the inverse of the adjacency function in Equation 12.

It is possible to convert the notes into chords. Although this is not a 100% objective process (some note sets can represent multiple chord symbols) it is interesting to notice is that the lowest energy output leads to chords in a minor key for 5 out of 8 of the input notes. Whereas taking the second lowest energy leads to chord progressions in a major key for 6 out of 8 notes. This observation is used to create a further control input for the user in qHarmony. They can request an increased chance of major or minor harmonization.

A prime motivation for looking at quantum computing for music is that a quantum computer has a non-deterministic output. The results can change from run to run, in an unpredictable but structured way. This is helpful in music generation systems to avoid lock-in, always generating the same material. It is actually possible to input more than one note for harmonizations by qHarmony. It is simply a case of setting the

weights of multiple note qubits to -7. This allows an extension of Table 4 to, for example, Table 5.

Table 4: Example chord outputs from the qHarmony algorithm

	Input Note:	***None***	***c***	***d***	***e***	***F***	***g***	***a***	***b***	***C***
Lowest energy output:		dfb	ceaC	dgb	ceaC	dfb	dgb	ceaC	dgb	ceaC
Other low energy outputs:		dgb	cefac	dfb	cefaC	dfgb	dfgb	cefaC	dfb	cefaC
		ceaC	cegaC		cegaC	cefaC	cegaC	cegaC	ceaC	cegaC
					degb	fgb	degb	dfab	dfgb	dgb
							dg		degb	dfb

Table 5: Example chord outputs with two note constraints from the qHarmony algorithm

	Input Note:	***cd***	***cf***	***de***	***cb***	***Ab***	***ga***	***bC***	***fg***
Lowest energy output:		cdfaC	cefaC	degb	cegbC	dfab	cegaC	cegbC	dfgb
Other low energy outputs:		dgb	cfaC	ceaC	dgb	ceaC	dgb	dgb	dfgb
		dfb	dfb	deb	dfb	cefaC	cegC	dfb	cegaC
		ceaC			ceaC	cdfab		dfgb	
		df						cdeC	
								cefabC	

The D-Wave 2X processor will now be used to generate a harmony for a simple tune. qHarmony is actually a hyrbid algorithm: part quantum computing, part classical computing. The classical algorithm takes as input a melody, and a list of mark-ups for where the melody is to be harmonized, the nnotes to be input to the quantum harmony generator, and whether the user would prefer major harmonies, minor harmonies or either. It then moves through the melody, calling the quantum algorithm described above at each harmony point. If a minor harmony is preferred, it selects the lowest energy output, if a major harmony the second lowest energy output. If neither is

constrained, it randomly selects one of the quantum computer's lowest energy results. Note that if more than one input note is used, then the system ignores the major / minor request, on the basis that its limited model 8 qubit is unable to deal usefully with too many constraints.

As an example consider the melody in Figure 20. Harmonies will be generated with the following constraints:

- Using the first note of the bar as the input note, with major / minor unconstrained
- A second run using the first note of the bar as the input note, with major / minor unconstrained - to show quantum fluctuations
- Using the first and last notes of the bar as the input notes, with major / minor unconstrained
- Using the first note of the bar as the input note, with bars 1-5 major harmonies are requested, bars 6-12 minor, 13 to the end major

The results, using the hardware quantum computer D-Wave 2X, are shown in Figures 21, 22, 23 and 24.

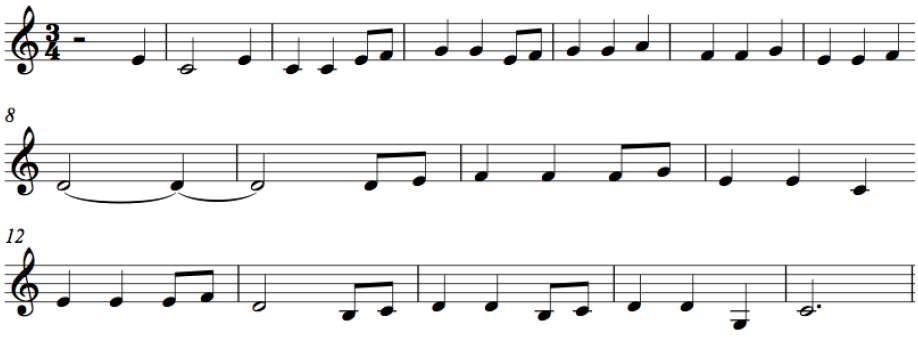

Figure 20: Example melody to send to hybrid algorithm

There are a few things worth noting about these figures. The most obvious is that the quality of harmonies produced is not particularly high. There are two main reasons for this. The first is that only an 8 qubit system was used (to simplify this introduction to adiabatic quantum programming). This limits the harmonies to white notes. Such a constraint is rare in most mainstream composition. However, introducing adiabatic quantum computing with a larger number of qubits (say 13 qubits) would have required much concentration on the issues of the chimera map. The connectivity of the D-Wave 2X outside of 8 qubit segments becomes more complex, as seen in Figure 20.

Furthermore, for introductory purposes, it was desired to use a simple harmony generation algorithm that didn't incorporate past results. qHarmony takes no account

of the context of previous harmonies. For example if a composer uses an *Am/C* chord to harmonize a melody segment, then that choice of chord will effect the next chord. Not so in this simplified example above. However, the example does fulfil its function - introducing the basics of the chimera configuration constraints, and showing how one should "think" when using an adiabatic QC (in terms of $E(s)$, h_i and J_{ij}).

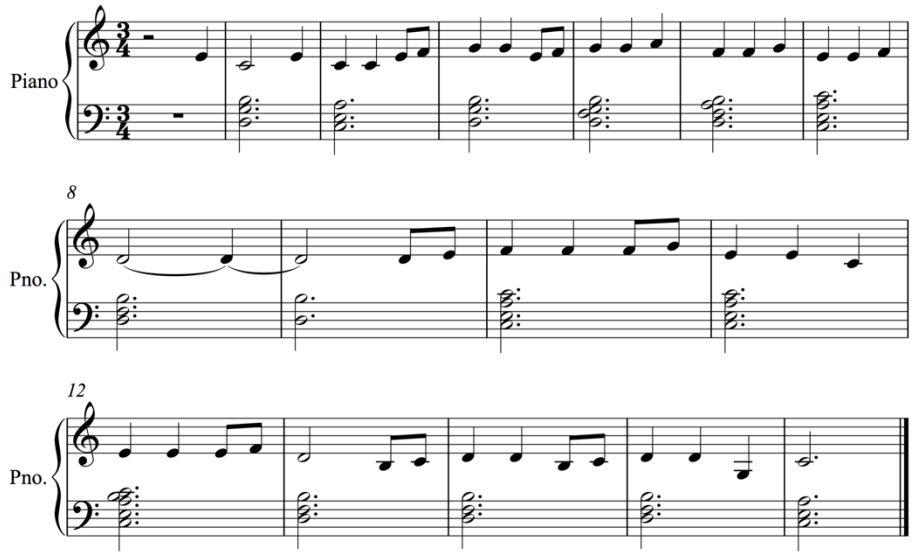

Figure 21: Melody in Figure 20 harmonized using the first note of the bar as the input note, with major / minor unconstrained.

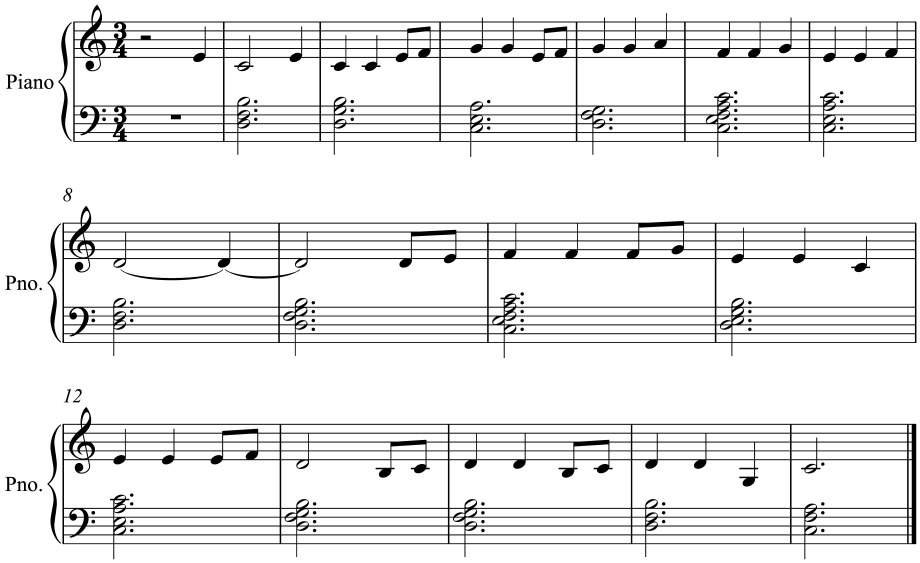

Figure 23: A second harmonization using the same method as Figure 22.

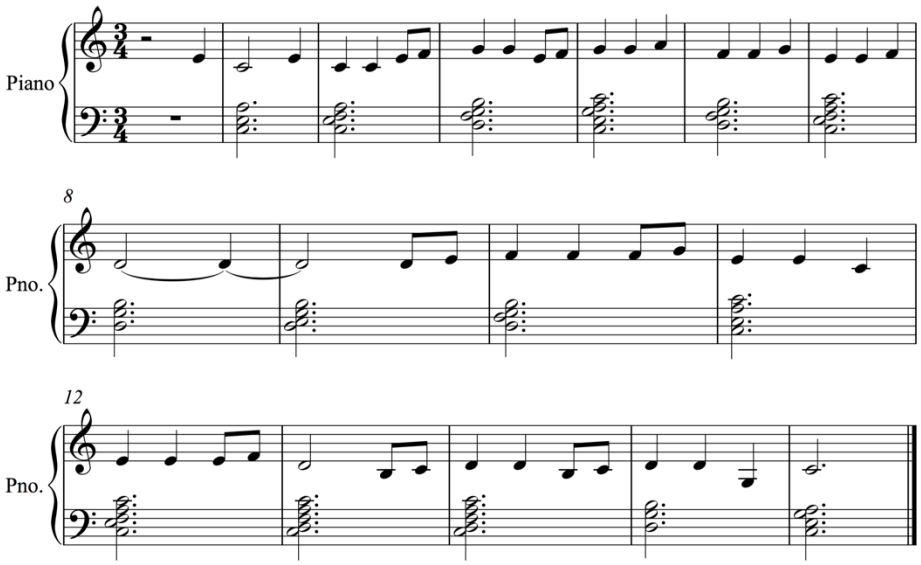

Figure 24: Harmonization of Figure 21 using the first and last notes of the bar as the input notes, with major / minor unconstrained.

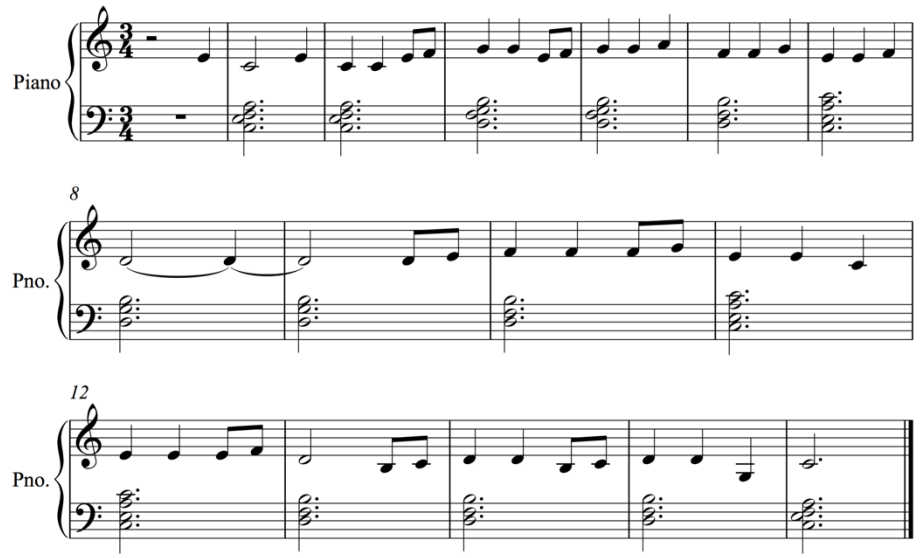

Figure 24: Harmonization of Figure 20 using the first note of the bar as the input note, with bars 1-5 major harmonies are requested, bars 6-12 minor, 13 to the end major.

The harmonization in Figure 21 (constrained by one note, and no key mode constraint) is a fairly acceptable, if not traditionally optimal, harmony. The harmonization in Figure 22 has the same constraints but provides an alternative presented by the quantum computer. These two harmonizations are sufficiently different to highlight alternative harmonic properties of the melody to a composer. Figure 23's harmonies sound comparatively unusual and unresolved. This is because the a second note is added to the constraints in each bar - and those second notes are mostly what would be described as passing notes in the melody - rather than significant harmonic notes.

The final example in Figure 24 returns to the one-note constraint, and adds a constraint of preferring certain bars to be minor and certain to be major. It is an attempted constraint because the major / minor division depends on what energy solutions are returned by the D-Wave 2X - and this itself is non-deterministic. The algorithm returns bars 2-5 as major (all correct); bars 6-12 are correctly minor except for bars 8 and 9; bars 13-15 are incorrectly minor but 16 is major as requested. This could be viewed as an "accuracy" of 66%. This is a misnomer as it is precisely the uncertainty that provides part of the motivation for quantum computer music.

qHarmony only takes advantage of one aspect of quantum computing: its non-deterministic nature and ability to return multiple results. However the quantum part of the algorithm is so simple that it does not require the potential speed-ups available from quantum computers. The D-Wave 2X has over 1000 qubits available, and enters states of superposition and entanglement during its calculations. Even the simple 8 qubit algorithm above will have utilized these quantum states in coming to the results in Figures 21, 22, 23 and 24.

However a much more complex and constrained problem would be required to utilize all advantages of quantum computing. Constraint-based and spectral composition, together with musical/sonic pattern matching algorithms are areas which may benefit from adiabatic quantum computing, due to their potential computational complexity. In essence, any complex musical problem that can be fully or partially modelled as an Ising system, could benefit from adiabatic quantum computation.

5.7 Final Remarks

This chapter has introduced research into quantum computing and music. This was done through a series of examples of research applying quantum computing and principles to musical systems. By this process, the key elements that differentiate

quantum physical systems from classical physical systems were introduced, and what this implied for computation. This allowed an explanation of the two main types of quantum computers being utilized inside and outside of academia: gate-based quantum computing and adiabatic quantum computing.

For gate-based computing, an example was given of an active sonification of a simulated CNOT-MZ gate and a hybrid system using musical computation and the same CNOT-MZ. For adiabatic quantum computing, an example was given of a simple harmony generation system using an actual hardware quantum computer, rather than a simulated one. The harmony algorithm made use of the non-deterministic multi-result nature of the quantum computing, but not of the potential speed up.

The idea of quantum computer music still is incipient. It is difficult to predict how it might develop from here. In this chapter we introduced a number of concepts and ideas, which are still in the early stages of development not only on the technical side but also on the musicological side. New musical concepts and compositional approaches need to be developed in terms of quantum thinking. New quantum mechanics-compatible representations of sound and compositional problem formulations need to be developed for a truly new approach to music: quantum computer music. One vein that seems to have greater potential for further development concerns granular sound synthesis. Gabor's notion of acoustic quanta is susceptible to the mathematical tools routinely used to describe the mechanics of quantum computing.

Did Max Plank ever envisage that his work might end up influencing the future of music too?

5.8 Acknowledgements

Parts of this chapter were published previously in the *International Journal of Unconventional Computation* (Kirke and Miranda 2014). Peter Shadbolt of the Controlled Quantum Dynamics Group at Imperial College London provided much help with insight into Bell's Theorem and CHSH, as well as Figure 11. Figures 1 – 10 were provided by both Peter Shadbolt and by Alex Neville of the Bristol Centre for Quantum Photonics. Daniel Lidar and his group provided much insight and support into using the D-Wave during the first author's residency at USC Viterbi School of Engineering.

5.9 Questions

1. What is one key difference between quantum and classical physics?
2. What properties of quantum systems provide the speed up advantage over classical computers?
3. Name two proposed substrates for traditional quantum computing.
4. What is wave-particle duality?
5. What property of photons is manipulated by a photonic quantum computer?

6. What is the difference between a NOT gate and a CNOT gate?
7. What is a beam splitter?
8. What is entanglement?
9. Why is the CHSH inequality used instead of the original Bells inequality?
10. What is the classical limit?
11. What is the quantum correlation?
12. How many experiments are required for each Bell's inequality calculation?
13. What is the difference between sonification and musification?
14. What is the difference between unconventional computation and unconventional virtual computation?
15. What form of calculation do traditional quantum computers use, as opposed to adiabatic quantum computers?
16. What was an Ising model originally designed to simulate?
17. What do the h_i and J_{ij} represent in an Ising model?
18. What does an adiabatic quantum computer attempt to do with the energy $E(s)$?
19. Calculate the energies of some example 8 qubit lists using the two note harmony system $E(s)$ equation.
20. How might you turn the two-note harmony system into a system that encourages dissonance?

5.10 References

Deutsch, D. (1985). “Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer”, *Proceedings of the Royal Society A* 400(1818):97–117.

Shadbolt, P., Mathews, K., Laing, A. O’Brien, J. (2014). “Testing the Foundations of Quantum Mechanics with Photons”, *Nature Physics* 10:278–286.

Shor, P. (2006). “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”, *SIAM Journal of Computing* 26(5):1484–1509.

Weimer, H. (2014). “Listen to Quantum Computer Music” *Quantenblog*. http://www.quantenblog.net/physics/quantum-computer-music, (Last accessed 31 Oct 2014)

Grover, L. (2001). “From Schrödinger's equation to quantum search algorithm”, *American Journal of Physics* 69(7): 769-777.

Sturm, B. (2000). “Sonification of Particle Systems via de Broglie’s Hypothesis”, *Proceedings of the 2000 International Conference on Auditory Display*. Atlanta, Georgia.

Sturm, B. (2001). “Composing for an Ensemble of Atoms: The Metamorphosis of Scientific Experiment into Music”, *Organised Sound*, 6(2):131-145.

O’ Flaherty, E. (2009). *LHCsound: Sonification of the ATLAS data output*. STFC Small Awards Scheme.

Putz, V. and Svozil, K. (2015). “Quantum Music”, *Soft Computing*, pp. 1-5. doi:10.1007/s00500-015-1835-x

Cadiz, R. and Ramos, J. (2014). “Sound Synthesis of a Gaussian Quantum Particle in an Infinite Square Well”, *Computer Music Journal* 38(4):53-67.

Coleman, J. (2003). *Music of the Quantum*. http://musicofthequantum.rutgers.edu/musicofthequantum.php (Last accessed 10 Sep 2016).

Glowacki, D., Tew, P., Mitchell, T., McIntosh-Smith (2012). S., *Danceroom Spectroscopy: Interactive quantum molecular dynamics accelerated on GPU architectures using OpenCL*. In *The fourth UK Many-Core developer conference* (UKMAC 2012), Bristol, UK.

Brody, J. (1997). *Background Count for percussion and 2 channel electroacoustic*. https://www.innova.mu/sites/www.innova.mu/files/liner-notes/314.htm (Last accessed 10 Sep 2016).

Alice Matters (2016) Scientists ‘sonify’ LHC data to Chamber Music, Alice Matters, 30 October 2014, http://alicematters.web.cern.ch/?q=content/node/776 Last accessed 2 June 2016.

Wired (2015). CERN's 'Cosmic Piano' uses particle data to make music, Wired, 8 Sept 2015.

Kielpinski, D., Monroe, C., Wineland, D. (2002). “Architecture for a large-scale ion-trap quantum computer”, *Nature* 412:709-711.

Kelly, J., et al. (2015). “State preservation by repetitive error detection in a superconducting quantum circuit”, *Nature*, 519:66-69.

Knill, E., Laflamme, R., Milburn G. (2001). “A scheme for efficient quantum computation with linear optics”, *Nature* 409:46–52.

Shadbolt, P., Verde, M., Peruzzo, A., Politi, A., Laing, A. Lobino, M., Matthews, J., Thompson, M., O'Brien, J. (2012). “Generating, manipulating and measuring entanglement and mixture with a reconfigurable photonic circuit”, *Nature Photonics* 6:45-49.

Shadbolt, P., Vértesi, T., Liang, Y. Branciard, C., Brunner, N., O'Brien, J. (2012). “Guaranteed violation of a Bell inequality without aligned reference frames or calibrated devices”, *Scientific Reports,* Vol. 2, Article 470.

Clauser, J., Horne, M., Shimony, A., Holt, R. (1969). “Proposed experiment to test local hidden-variable theories”, *Phys. Rev. Lett.* 23(15):880–884.

Quantiki Wiki (2015) List of QC Simulators, From http://www.quantiki.org/wiki/List_of_QC_simulators (Last accessed 01 Apr 2015).

Kirke, A. (2015a). Sound example. https://soundcloud.com/alexiskirke/the-entangled-orchestra-simulation-prototype (Last accessed on10 Sep 2015).

Goldberg, R. (1974). “Survey of Virtual Machine Research”, *Computing*, June, pp. 34-45.

Freund, S. and Mitchell, J. (1999). “A formal framework for the Java bytecode language and verifier”, *Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA '99)*. New York, NY, USA, pp. 147–166.

(VMWare Inc. 2015). “Understanding Full Virtualization, Paravirtualization, and Hardware Assist”. Available on-line: http://www.vmware.com/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html (Last accessed on 28 Sep 2016).

Seo, K., Hwang, H., Moon, I., Kwon, O. and Kim, B. (2014). “Performance Comparison Analysis of Linux Container and Virtual Machine for Building Cloud”, *Advanced Science and Technology Letters* 66:105-111.

Guohui W., Ng, T. (2010). “The Impact of Virtualization on Network Performance of Amazon EC2 Data Center”, *Proceedings IEEE INFOCOM*, pp.1-9, 2010.

Huber, N., von Quast, M. et al. (2011). “Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments”, *Proceedings of the 1st International Conference on Cloud Computing and Services Science (CLOSER 2011)*, Noordwijkerhout, The Netherlands, pp. 563 - 573, 2011.

Coates, A., Huval, B. et al. (2013). “Deep learning with COTS HPC Systems”, *Proceedings of the 30th international Conference on Machine Learning*, Atlanta, Georgia, USA.

Garmon, J. (2010). “What were the original system requirements for Windows 1.0? Available on-line: http://www.jaygarmon.net/2010/11/what-were-original-system-requirements.html (Last accessed on 28 Sept 2016).

Jones, J. (2010). “The Emergence and Dynamical Evolution of Complex Transport Networks from Simple Low-Level Behaviours”, *International Journal of Unconventional Computing* 6(2):125-144.

Bull, L., Budd, A., Stone, C., Uroukov, I., Costello, B. d. L., Adamatzky, A. (2008). “Towards unconventional computing through simulated evolution: Control of nonlinear media by a learning classifier system” *Artificial Life* 14 (2):203-222.

Spector, L., et al. (1999). "Finding a better-than-classical quantum AND/OR algorithm using genetic programming", *Proceedings of the Congress on Evolutionary Computation*. Vol. 3.

Kirke, A. and Miranda, E. (2014). “Pulsed Melodic Affective Processing: Musical structures for increasing transparency in emotional computation”, *Simulation* 90(5):606-622.

Kirke, A. and Miranda, E. (2014). “Towards Harmonic Extensions of Pulsed Melodic Affective Processing - Further Musical Structures for Increasing Transparency in Emotional Computation”, *International Journal of Unconventional Computation* 10(3):199-217.

Kirke, A. (2015b). Sound example: https://soundcloud.com/alexiskirke/quantum-pmap-convergence-example (Last accessed on 28 Sep 2016).

Hong, C., Ou, Z. and Mandel, L. (19870. "Measurement of subpicosecond time intervals between two photons by interference. *Phys. Rev. Lett.* 59 (18): 2044–2046.

Warren, R.H. (2013). “Numeric experiments on the commercial quantum computer”, *Notices of the AMS* 60(11).

Albash, T., Vinci, W., Mishra, A., Warburton, P.A. Lidar, D.A. (2015). “Consistency tests of classical and quantum models for a quantum annealer. *Physical Review A* 91(4): 042314

Lucas, A. (2014). “Ising formulations of many NP problems”, *Frontiers in Physics*, Vol. 2, id.5.

Neven, H. (2016). “Quantum Annealing at Google: Recent Learnings and Next Steps”, *APS March Meeting 2016*, abstract #F45.001.

Katzgraber, H. G. (2015). “Seeking Quantum Speedup Through Spin Glasses: Evidence of Tunneling?”, *APS March Meeting 2015*, abstract #L53.005

Bian, Z., Chudak, F., Macready, W. G. Rose, G. (2010). *The Ising model: teaching an old problem new tricks.* Available on-line http://www.dwavesys.com/sites/default/files/weightedmaxsat_v2.pdf (Last accessed on 28 Sep 2016).

Isakov, S.V., Zintchenko, I.N., Rønnow, T.F. aTroyer, M. (2015). “Optimised simulated annealing for Ising spin glasses”, *Computer Physics Communications* 192, pp.265-271.

Stockhausen, K. (1959). “… How Time Passes …”, *Die Reihe* (English edition), Vol. 3, pp. 10-40.

Xenakis, I. (1971). *Formalized Music*. Indiana University Press.

Davidson, M. P. (2007). “Stochastic Models of Quantum Mechanics – A Perspective”, *AIP Conference Proceedings* 889. Available on-line: https://arxiv.org/pdf/quant-ph/0610046.pdf (Last accessed on 30 Sep 2016).

Gabor, D. (1947). Acoustical quanta and the theory of hearing, *Nature*, 159, 591–594.

Miranda, E. R. (2002). *Computer Sound Design: Synthesis Techniques and Programming*. Elsevier/Focal Press.

Miranda, E. R. and Maia Jr., A. (2007). “Fuzzy Granular Synthesis Controlled by Walsh Functions”, *Proceedings of X Brazilian Symposium on Computer Music.* Belo Horizonte, Brazil. Available on-line: http://compmus.ime.usp.br/sbcm/2005/papers/tech-12479.pdf (Last accessed on 30 Sep 2016).

Fine, N.J. (1949). "On the Walsh functions". Transaction of the American Mathematical Society ***65****: 372–414.*

Wallis, J. S. (1976). "On the existence of Hadamard matrices". *Journal of Combinatorial Theory A*. **21** (2): 188–195.