Paul Doornbusch
RMIT University
GPO Box 2476V
Melbourne 3001, Australia
pauld@iii.rmit.edu.au

Computer Sound Synthesis in 1951: The Music of CSIRAC

The Australian-built ''automatic computer'' initially known as the CSIR Mk1, and later known as CSIRAC, was one of the world's earliest stored-program electronic digital computers (Williams 1997). (See Figure 1.) Coincidentally, it may also have been the first computer to play music, even though later work done elsewhere in the 1950s is clearly the origin of computer music as we know the field today.

Developed in Sydney in the late 1940s by the Council for Scientific and Industrial Research (CSIR), the CSIR Mk1 ran its first program in November 1949. Geoff Hill, a mathematician and Australia's first real software engineer, programmed the CSIR Mk1 to play popular musical melodies through its loudspeaker starting in 1951, if not 1950. The CSIR Mk1 was moved to the University of Melbourne in June 1955 and renamed CSIRAC (McCann and Thorne 2000). It performed useful and trailblazing service there until 1964. During CSIRAC's time in Melbourne, the mathematics professor Thomas Cherry programmed it to perform music, developing a system and program such that anyone who understood standard musical notation could create a punched-paper data tape for CSIRAC to perform that music.

Although the music performed by the CSIR Mk1 may seem crude and unremarkable compared to the most advanced musical developments of the time, and especially to what is possible now, it is probably the first music in the world to be performed on a computer, and the means of production lay at the leading edge of technological sophistication at that time. These first steps of using a computer in a musical sense occurred in isolation, but they are still interesting, because the leap of imagination in using the flexibility of a general-purpose computer to create music and the programming ingenuity required to achieve it are significant. CSIRAC took some initial steps in that direction.

Computer Music Journal, 28:1, pp. 10–25, Spring 2004

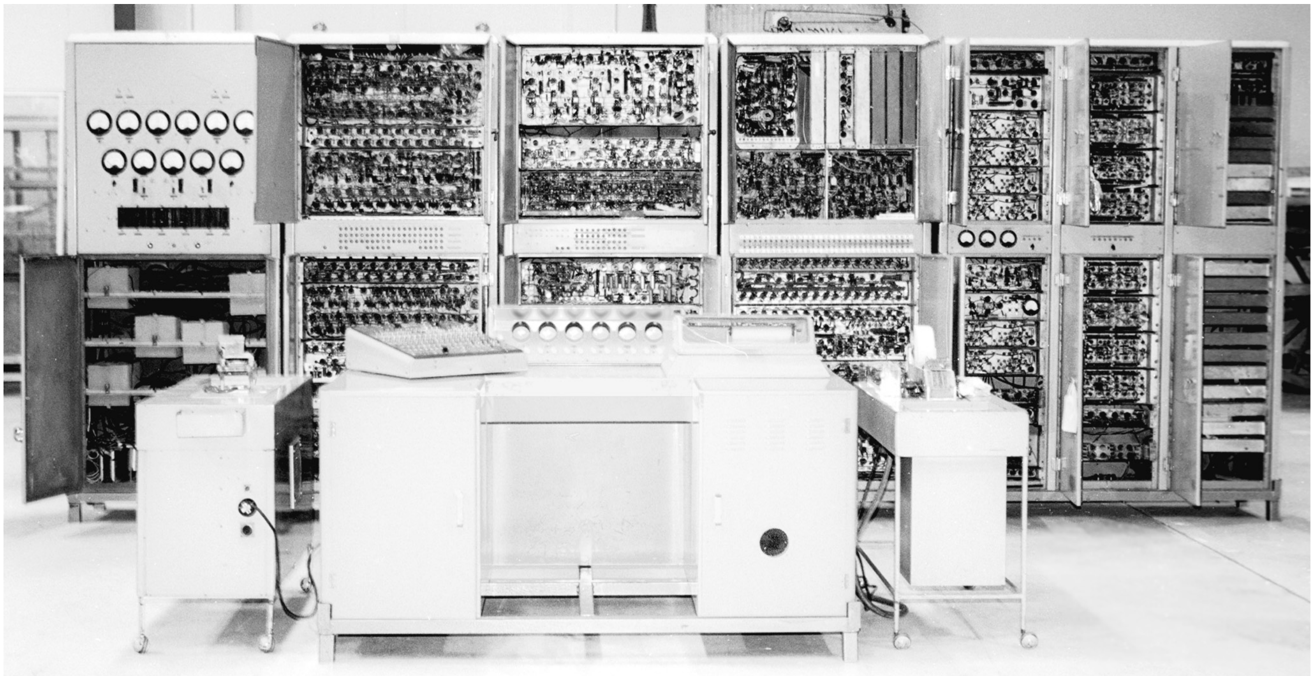
An Overview of CSIRAC

In the 1940s, modern physics had advanced to such a stage that the calculations required were enormous, manifold, and tedious. To address this problem, calculating machines had been developed, such as the ''linear equations machine,'' the ''differential analyzer,'' and the ''multi-register accounting machine.'' However, these calculating machines still required much human intervention, so there was a desire to build an automatic calculator with some sort of memory to store the data and also the instructions of what to do with the data. Two major technological advances of the time allowed the realization of an automatic calculator with memory. One was the thermionic valve (the vacuum tube), which was used as a switching device or as an electronic relay. The other was mercury delay-line ''memory,'' which had been used in radar systems during World War II. This memory system could be adapted for use in an automatic calculator.

The CSIR Radiophysics division was active during World War II, developing portable radar systems for Australian servicemen for jungle warfare. In 1941, Maston Beard joined the Radiophysics Laboratory, where, with Trevor Pearcey, he was instrumental in the development of Australia's first computer. This work was undertaken by the Radiophysics Laboratory as they were the leading experts in pulse technology and valve electronics (McCann and Thorne 2000).

Trevor Pearcey and Maston Beard officially began the ''Electronic Computer'' project in 1947. (Although the terms ''electronic computer'' and ''automatic computer'' may seem tautological today, in the 1940s to 1950s, the term ''computer'' typically denoted a secretary who operated a calculating machine.) In addition, in the same year the University of Pennsylvania published previously secret details of the computer known as ENIAC. In mid 1948, Pearcey completed and published the fundamental logical design of the computer in two papers. The

Figure 1. CSIRAC as displayed for its 50th birthday celebration, Museum Victoria, 25 November 1999. Note the speaker near the bottom of the right-hand door of the console.



overriding considerations of the logical design were engineering and programming simplicity, as this was intended to be a comprehensive prototype for a larger and more capable machine. The construction began in 1948. The first program, which simply multiplied two numbers, was run late in 1949, probably in November, but nobody recorded the exact date. Trevor Pearcey recalls, ''We all shouted 'Hooray!' and went back to work'' (Pearcey 1996).

The CSIR Mk1 was very different from today's computers because it was a serial machine. Data were sent around the computer from ''sources'' to ''destinations,'' one bit at a time. Modern computers typically move 32 or 64 bits in parallel. The CSIR Mk1's serial architecture had consequences for programming the machine, and it was especially significant for timing-critical programs such as those that may play music in real time.

It is important to appreciate that all operations were considered as serial transfers of numbers, or data, from a ''source'' to a ''destination.'' A source could be something like a register, a memory location, or the accumulator. A destination could be a memory location, a register, the paper tape punch, or the loudspeaker. Each 20-bit digital word was partitioned into a 5-bit destination, a 5-bit source, and a 10-bit data address.

During the transfer, the data could undergo transformation, such as being subtracted or added. The ten-bit data address, if it applied to the main (mercury delay-line) storage was further subdivided into two five-bit components: one to select which mercury delay line the data were in and another to select the position, or ''time,'' of the data in that delay line (Dean 1997). As the memory was a recirculating delay line and the machine architecture was serial, it was necessary for a program to *wait* until a particular memory location was available for reading. Two 1-msec major-cycles was the minimum time to execute an instruction, but if an opportunity was missed to access a memory address, then it could take 3 msec or 4 msec. This variable timing of instructions and memory access was critical to some applications, such as producing a repeatable sound, and this is the key to understanding how the music was produced. Each memory tube was a delay line, so the data in each position in a memory tube required a different time to ac-

Table 1. Facts and Figures Regarding the CSIRAC Computer

Architecture	Serial architecture; recirculating acoustic delay-line memory
Number of Tubes	Over 2000
Power Consumption	30,000 watts
Mass	7000 kg
Physical Dimensions	45 square meters
Memory	768 words of main memory (one storage ''word'' = two bytes)
Disk Storage	2048 words
Clock Frequency	0.001 MHz (1 kHz)
Speed	0.0005 MIPS (500 operations/sec)

cess. It was possible to calculate this time and determine how long after the start of a clock or access cycle the data were read. Numbers were placed in specific memory locations in such a way that when they were read out and sent to the loudspeaker, they were pulses with a pre-determined period. In this way, a predictable pitch was produced and used to create musical melodies (Ryan 2000).

Table 1 summarizes several facts and figures regarding the CSIRAC computer, and Figure 2 shows a diagram of the CSIR Mk1 architecture as it was originally conceived, ca. 1947 (Dean 1997). More detail about the architecture, construction, and technicalities of CSIRAC can be found online at www.cs.mu.oz.au/csirac.

The Loudspeaker

In common with several other first-generation computers, the CSIR Mk1 had a built-in loudspeaker. The Ferranti Mark I, derived from the Manchester Mark I, had a loudspeaker, and, like CSIRAC, played some very early music, as recorded in the British National Sound Archive on the tape with archive number H3942. Although I have seen no documentation about the Ferranti Mark I's means of producing music, this tape sounds very similar to the sounds produced by CSIRAC (according to Burton 2000).

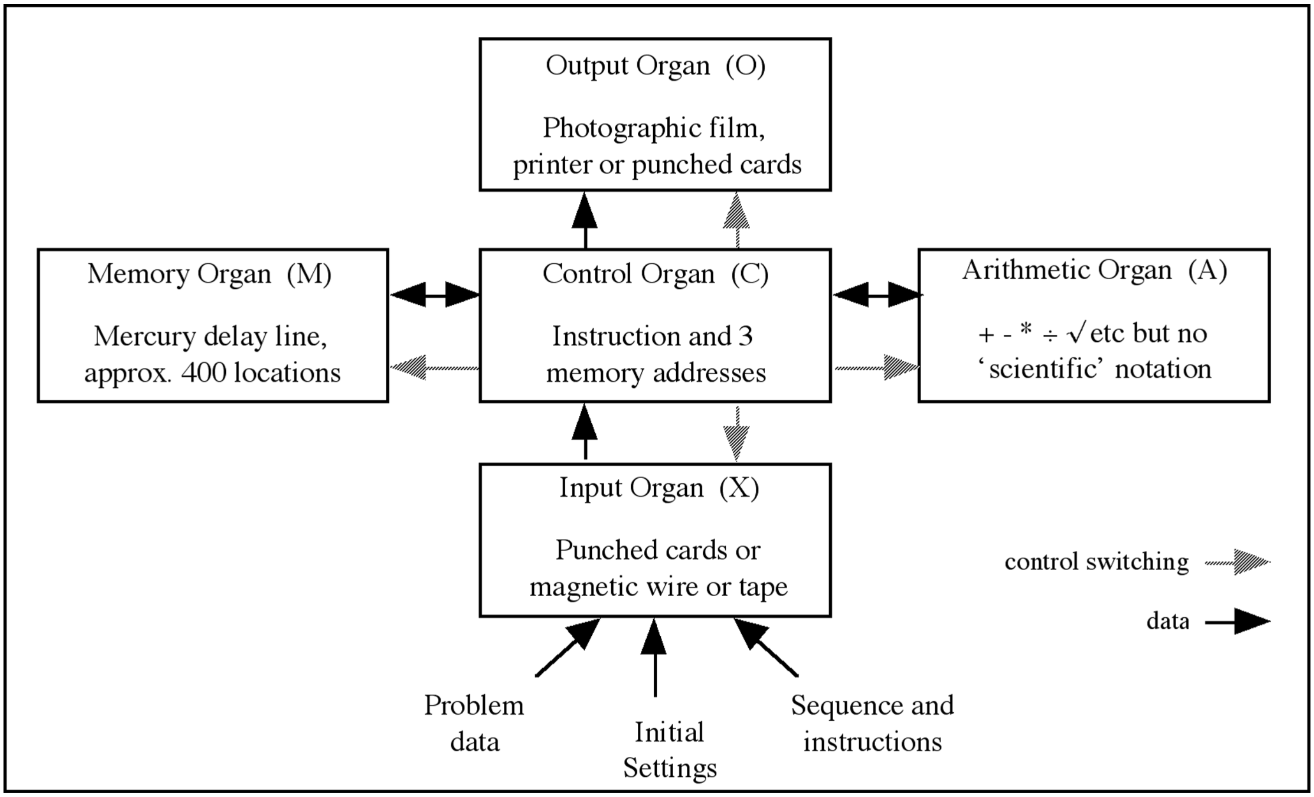
The CSIR Mk 1's loudspeaker, or ''hooter'' as it was called, was an output device used by programmers to signal that a particular event had been reached in the program. It was commonly used for warnings, often to signify the end of the program and sometimes as a debugging aid. With many of the earliest computers, owing to the lack of visual feedback (there was no display as is customary today), it was common to include a ''hoot'' instruction at the end of a program to signal that it had ended, or elsewhere if a signal was needed for the operator.

The loudspeaker on the CSIR Mk1 was built into the computer in such a way that it was a destination for data, effectively on a register of the machine, and it received the raw pulse data off the ''bus'' (Dean 1997). This worked to create some sort of sound, but it required more effort to accomplish a stable, pitched sound out of it as a single pulse would barely make an audible click. Therefore, multiple pulses would be required to achieve an audible result, possibly as several in-line ''P'' statements or as a short loop of instructions. The timing of the loop would have caused a change in the frequency of the sound from the ''hooter.'' Any programmer with an interest in sound or music would immediately see the potential available.

The sequence of instructions could look like this, for example:

```
 start of note
 set counter to N
 loop start, send pulse in memory
location M1 to speaker
 wait time t1
 send pulse in memory location M2 to
speaker
 wait time t1
 send pulse in memory location M3 to
speaker
 wait time t2
 send pulse in memory location M4 to
speaker
 wait time t3
 send pulse in memory location M5 to
speaker
 decrement N
 if N equal to 0 then exit, else return
to loop start
 end of note
```

Figure 2. Diagram of the CSIR Mk1 architecture as originally conceived, ca. 1947.



Music and Technology in the Time of CSIRAC

It is useful to place the activities of the CSIR Mk1 in historical context. There was significant activity in electronic music before the development of the CSIR Mk1 and prior to more experimental and adventurous musical developments after World War II. For example, the telharmonium, Theremin, sphärophon, dynaphone, ondes martenot, and trautonium achieved enduring reputations because major composers wrote works for them. All of them were capable of monophonic melodic output, but most of them were short-lived and were mostly used to perform traditional music. The Hammond organ was a different and commercially more successful development that was also used largely in a musically traditional manner (Chadabe 1997; Ernst 1977; Manning 1993).

There were other developments at this time that used electronics and that took a fresh and less musically restrictive approach to both sounds and music itself. For example, Hugh Le Caine's ''Coded Music Apparatus'' (1952) allowed the control of sound synthesis by five curves, one each for pitch and amplitude and three for timbre (Chadabe 1997). Similarly, Percy Grainger wrote *Free Music* for multiple Theremins and developed his own electronic musical instruments with the assistance of Burnett Cross (Bird 1999). However, most early electronic musical instruments were used to play electronic renditions of standard repertoire and not to create new music.

The real history and legacy of electronic music comes from developments about the same time that CSIRAC was being planned and built, against the background of the great artistic expansion after World War II. Against this background and with a spirit of freedom, reconstruction, and liberation that many artists felt after World War II, electronic music blossomed. In the field of electronic music, there were the two significant emerging developments: *musique concrète* and analog electronic music. In addition, John Cage, Pierre Boulez, Edgard Varèse, and others were writing advanced instrumental music, developing new composition theories and certainly becoming interested in electronic music (Cage 1967). It is against this background, but in isolation, that CSIRAC first played music.

Computer music developed quickly in other areas where composers were involved from the early developments. These developments took place at about the same time as Thomas Cherry developed his parameterized Music Programme for CSIRAC. The field of computer music at that time included two main streams of activity: computer-assisted composition and sound synthesis (Manning 1993). Computer-assisted composition (for traditional musical instruments, initially) developed from the early work of Lejaren Hiller and Leonard Isaacson, which started late in 1955 at the University of Illinois (Hiller and Isaacson 1959). Other composers quickly followed this work with developments of their own. In Paris from about 1957, Iannis Xenakis developed music programs that modeled his compositional processes, which used statistics and probability theory to choose musical parameters (Chadabe 1997; Roads 1996). Similar work was undertaken nearby a few years later by Pierre Barbaud and Roger Blanchard at Compagnie des Machines Bull, the French computer company (Hill 2000; Manning 1993).

In the early to mid 1960s in England, Stanley Gill, D. Champernowne, and D. Papworth were following comparable objectives. In Germany and the Netherlands during this time (from 1964), Gottfried Michael Koenig was developing his computer composition program Project 1. In 1957, at about the same time as the *Illiac Suite*, Max Mathews at Bell Laboratories was developing his Music I program, which produced completed music as a digital audio file that was played back with a digital-to-analog converter (Pierce 1995). By 1959 and with the help of composers, this work had evolved to allow timbral variation and modification of the sounds through sound synthesis.

For computing, however, it is not as simple to put activities into historical context as it would at first seem. There is a problem of definition: what combination of hardware and software capabilities constitutes a computer? Konrad Zuse created some early electromechanical calculators. His Z3 of 1941 (using relays) has been called the first electromechanical programmable digital computer. Alan Turing's Colossus, which was operational in 1943, has been called the first all-electronic programmable calculator as it had no memory and was driven by a punched paper tape. ENIAC was designed as a calculator but was later given programmable control (Williams 1997). However, if one accepts the definition of a digital computer as an all-electronic device capable of calculating and branching operations, where the data and instructions are held in rewritable memory, then the following series of events shown in Table 2 is a guide to when the first all-electronic digital computers first ran test programs.

Table 2. A Brief Guide Outlining When the First All-Electronic Digital Computers Ran Their First Test Programs

Date	*Machine*	*Institution*	*Country*
June 1948	MADM	Manchester University	UK
May 1949	EDSAC	Cambridge University	UK
August 1949	Binac	Electronic Control Company	USA
September 1949	Harvard Mark III	Harvard University	USA
November 1949	CSIR Mk1	CSIR Radiophysics	Australia
May 1950	Pilot ACE	National Physics Laboratory	UK
May 1950	SEAC	National Bureau of Standards	USA

When CSIRAC Played Music

There is a body of evidence that points to the CSIR Mk1 playing music in the very early 1950s. This largely comes from anecdotal sources, which is not surprising as it was an unofficial activity. This situation may change, as various other items of importance have been uncovered over the last several years. Doug McCann is the main historian who has studied CSIRAC and who has interviewed the original personnel involved with the project. He recalls hearing about the music:

> I was particularly interested in ascertaining if and when CSIRAC had played music. We questioned [Trevor] Pearcey about it. Pearcey was very lucid and said he clearly remembered it playing music in 1951. He tried to recall if it was any earlier than that because he said the computer started doing regular work in 1950 sometime. He said it definitely played music in August 1951 at the first Australian Computer Conference, and that it had first played it some time before that.
>
> Pearcey tried to give us a definite event to date it from and he was firm about the Australian Computer Conference. He did not appear to be the sort of person to exaggerate; if anything he was more likely to be conservative in his estimations for the sake of accuracy. If he was not sure of something he would say so. There was no hesitation in his assertions that the computer played music, at the latest, in August 1951. He was just not sure how he could date it any earlier. He said it was an early programming exercise and suggested it was first done sometime in 1950. (McCann 2000)

A fragment of the videotaped interview with Trevor Pearcey, which occurred several years after the one just recalled by Doug McCann, is transcribed below:

> I believe CSIRAC was a very early machine to provide tones through a loudspeaker . . . We played ''Girl with Flaxen Hair,'' ''Colonel Bogey,'' and one or two other things like that. . . . I had suggested that we record the tunes and get Frank Legg . . . to play it over the radio. However, Dr. Bowen, who was then chief, did not think this was good enough. I think he didn't realize the intellectual skill and effort that had gone into actually getting the machine to play specific musical sequences. This was in 1950 or '51; I cannot give a precise date. It was certainly a very early programming exercise. We played it at the conference. (Pearcey 1996)

Sadly, most of the people who were associated with the early days of the computer are no longer living, including Trevor Pearcey, Maston Beard, Geoff Hill, and Frank Legg, a radio announcer who was apparently willing to broadcast the music. However, some other people can verify that the CSIR Mk1 first played music in 1951. Reg Ryan, who started with the project in 1948, can recall the music at the 1951 conference: ''I can remember it playing music at the public opening, the conference, 1951 I think it was. I can't recall if it played music before that, it must have done so but I can't recall it now'' (Ryan 2000).

This is a fair corroboration that the CSIR Mk1 publicly played music at the first Conference of Automatic Computing Machines at Sydney University on 7–9 August 1951, but there is a little more information to back this up. Dick McGee, who started with the CSIR in April 1951, also remembers the music from that time:

> I can remember hearing the music soon after I started there. It was something simple they played. Something like ''Twinkle Twinkle'' I think. Anyway, I can remember that it was 1951, because I started in April. I went to one or two lectures at the conference. I don't remember the event where they played the music—they must have walked everybody over to the other building—but I can remember them talking about it soon after the conference, how surprised everyone was and so on. (McGee 2000)

Geoff Hill's Master of Science thesis from The University of Sydney, dated March 1954, mentions the music of the CSIR Mk1 on page 63: ''Extension to semantic analysis of language has led to design of 'language-translation' programmes. Sub-routines generating notes of the chromatic scale by sending pulses to a loud speaker [sic] can be used in an interpretive programme for playing music from a coded score. . . . [T]he techniques of music and 'translation' programmes were adapted as the basis of the powerful interpretive approach to complicated arithmetic'' (Hill 1954).

In addition to this, Ron Bowles (a CSIR Mk1 maintenance engineer) has examined the Sydney music punched paper tape and can establish the date for it as the first half of 1953. This is mostly owing to the particular type of tape reader needed for this tape, as the input and output devices on the computer changed several times. There are also other specifics of the machine, for example, the primary program or bootstrap loader and the implementation or detailed workings of some instructions, which varied from time to time and are therefore useful in dating programs. There are nine musical items on the Sydney music tape from early 1953. Each of these would have taken a considerable time to program. Given this and the secondary priority of the music, then it would suggest that the earliest items on this tape would have existed a significant amount of time before 1953 (Bowles 2000).

Taking into account all of the corroborating evidence, it is reasonable to accept the recollections of Trevor Pearcey, Reg Ryan, and Dick McGee that the CSIR Mk1 did publicly play music in the first week of August in 1951. It must have been playing music for some period before that notable date, but it is impossible to accurately determine a date for that now.

In Melbourne, circa 1957, Thomas Cherry wrote a music program that extended the pitch and dynamic range possible with the machine. Mr. Cherry generalized the structure of the music program such that it could accept a data tape of note pitch and duration data. With some simple instructions, someone could produce a data tape of some music so that it could be played by CSIRAC. Terry Holden, Ron Bowles, and Kay Thorne are the main source of the date for this. They say that it could have been late in 1956 that Mr. Cherry first had recognizable music coming from the machine, but it was most likely in 1957 (Bowles 2000; Holden 1997; K. Thorne 2000). The written records have only a little information that can date the work. Mr. Cherry used a form of notation in his private notes that was not used after 1958, so this supports the memories of Kay Thorne, Ron Bowles, and Terry Holden. In addition to this, a little support can be gleaned from the fact that Mr. Cherry used the Melbourne 12-hole paper tape exclusively, as CSIRAC had 5-hole equipment attached later, at about the middle of its service in Melbourne. This points to early work on the computer in Melbourne, as do Mr. Cherry's activities, because he was programming only very early in the Melbourne period, after which his duties were in administration.

The Music

As mentioned, the CSIR Mk1 had a ''hooter'' circuit that could be programmed to produce a variable frequency by sending pulses to the speaker at varying rates. It was a small step, then, to imagine that if one could control this process, then a controlled pitch would be the result. The first programmers of the CSIR Mk1 were Geoff Hill and Trevor Pearcey. Geoff Hill had perfect pitch and came from a musical family; both his mother and sister taught music throughout their lives (Hill 2000). Geoff Hill was the first person to program the CSIR Mk1 to play a musical melody. Initially, this was probably as a programming exercise and for his own interest.

The sound-production technique used on the CSIR Mk1 was as crude as is possible to imagine on a computer. The raw pulses of the computer's data words, the bit stream pulses, were sent directly to an audio amplifier with a loudspeaker attached. However, it is also worth remembering that this occurred when there was no such thing as digital-to-analog converters for playback, no techniques for working with digital audio, and little in the way of a complete theory of digital audio. In addition, the CSIR Mk1 produced music in real-time, which was necessary to overcome the limitation of the lack of mass storage. There was nothing such as magnetic computer tape to store digitized audio. There were enormous timing subtleties of the computer and sound generation process to be understood and addressed to achieve a stable, pre-determined frequency output.

This work took place in isolation and without prior example, eventually for the purpose of public demonstration and entertainment, but initially it was probably a personal interest. It was also possibly used as a significant programming challenge because of the timing and programming intricacies that needed to be negotiated. Engineers, not composers, undertook this musical endeavor, so the musical implications of the computer were not fully explored. Significant advances in computer sound generation theory and practice would have to wait for some further technical developments and the work of Max Mathews at Bell Telephone Laboratories (Roads 1980, 1996). John Pierce, who at the time was executive director of the Communication Sciences division at Bell Labs, said that there were computers making music with ''buzzes and squawks'' before the work of Max Mathews (Mathews 2000; Pierce 1995). Although it was little reported at the time, the CSIR Mk1 was one of these.

The musical pieces played by the CSIR Mk1, being popular tunes of the day, are not as musically significant as they might have been had composers been involved in creating the music. The achievement is important owing to the imagination of the practitioners, who conceived of using the flexibility of a digital computer to make music, and owing to the ingenuity required of the programmers to devise means to produce sounds from the computer. It is difficult to appreciate now just how skillful these people were. Only two of the best programmers were able to program the CSIR Mk1 to play music. Overcoming the technical issues in spite of a lack of prior practice to work from shows the ingenuity and skill that were needed to complete the task. Several people commented that when they first heard the music in the 1950s, it was perceived as something magical; it was ''astonishing.''

Thomas Cherry, in Melbourne, programmed CSIRAC to play music, and he also wrote some of the main arithmetic routines in the software library. The methods he adopted in the programming were modifications of Geoff Hill's practice in Sydney, and the range of notes and dynamics was extended. Hardware changes, providing speedier instructions, allowed the programming of some notes to extend the range. Mr. Cherry also designed the structure of the music program and generalized it such that it could accept a data tape of note pitch and duration data.

The generalization of playing music with the computer was probably the most significant contribution by Mr. Cherry to computer music practice with CSIRAC. In about 1957, he wrote a music performance program that would allow a computer user who understood simple standard music notation to enter it easily into CSIRAC for performance, without negotiating all of the timing problems normally required. The ''Music Programme'' could play music from memory or from punched paper tape. Playing from tape could allow playback of very long pieces, too large to fit into memory, but the speed of successive notes was limited by the speed of the paper tape reader. In practice, this was a small limitation, as it could play about ten notes a second. Playing from memory had the advantage that faster notes could be played, but the piece could not be large.

This work by Mr. Cherry is comparable to other computer music research of the time. The Music Programme, which allowed high-level (numerical) descriptions of music, has several similarities to Max Mathews's work from 1957 on Music I. Mr. Mathews was working at Bell Labs when he started his experiments of applying computers to musical goals. Neither the Music Programme nor Music I included a variable synthesis component (Roads 1980). The sound synthesis method used on CSIRAC was not structured in a way to allow a modification of the timbre. However, Music I was designed to output a tape of digital samples that would be played back later through digital-to-analog converter hardware, so it would soon include a synthesis component (Music II, in 1958) to allow modification of the sound waveform, and it did not need to deal with the problems of real-time performance of the work. The ability to output sound files with Music I and Music II was the result of several technological advances to which CSIRAC, being from a previous generation of computing technology, did not have access. To generate, store, and play back sound files requires: (1) on-line mass storage such as magnetic tape that can store sound files, (2) digital-to-analog converters to convert the sound files to audio, and (3) significant electronic memory capacity. The high-level, numerical specification for musical parameters was important for both Mr. Mathews's Music I and Mr. Cherry's Music Programme. It was also probably a natural step from other music technology such as punched paper tape or rolls for control of player pianos; however, it implies a degree of sophistication in the software to generalize the input data. The use of digital sound files was significant for Music I and later developments because it allowed for arbitrary timbral variation. Thus, Music II (1958) was capable of four independent parts with any of 16 timbres. This is an order-of-magnitude improvement over not being able to modify a timbre, although it did force a non real-time approach owing to the limits of computing power (Roads 1980). According to Max Mathews, composers' requirements drove these developments. The more recent developments of the ''Music N'' languages, for example Csound, did not achieve real-time output until about 1990; such is the processing power required. However, given all of the foregoing, the real significance of the work at Bell Labs were the concepts of the ''unit generator'' that was introduced in 1960 with Music III and the ''stored function,'' or ''function table,'' oscillator.

Both Mr. Cherry's Music Programme and Mr. Mathews's Music I allowed a similar high-level and numerical specification of musical events. However, what the Music Programme lost in timbral manipulation it made up for with real-time performance of a piece of music and limited run-time transposition (set via a register on the console) and tempo variation of the piece.

Music Reconstruction

When I first heard that CSIRAC played music in 1951, I was in disbelief. However, after a few electronic mail messages and discussions with some of the people previously mentioned, I became very curious. I hatched a plan to reconstruct the music, with the support of Peter Thorne and the University of Melbourne, but I was worried because I had no idea how to actually achieve this.

The music played by CSIRAC was unfortunately never recorded onto any audio storage format such as tape or disk. Little was known about the music, and although several people who had heard it in the 1950s and 1960s were still around to tell the tales, it was impossible to hear it any more. However, three key people who would be needed for the music reconstruction—Ron Bowles, John Spencer, and Jurij (George) Semkiw—were involved with a project at the University of Melbourne to thoroughly document CSIRAC, as it is now a museum piece and one of the oldest, intact, first-generation computers in the world. CSIRAC's circuit diagrams, manuals, and documentation still exist, as does all of its program library, which makes it one of the best-documented old computers. John Spencer was a programmer on CSIRAC. He not only remains highly skilled with CSIRAC programs, but he has also written a very comprehensive emulator for CSIRAC that runs on a modern personal computer. Ron Bowles and Jurij Semkiw were CSIRAC maintenance engineers who have intimate experience and undocumented knowledge of the internal workings of the machine. The unique skills of these three pioneers made the reconstruction of the music possible.

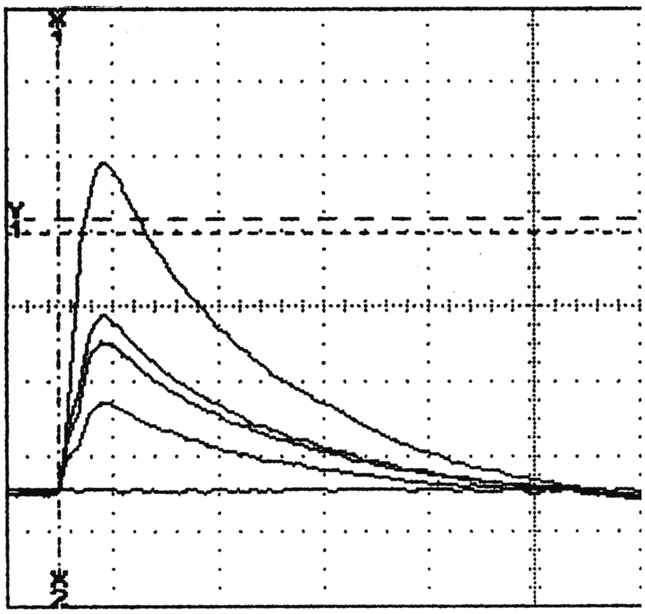
The first step, obviously, was to read the punched paper tapes of the Sydney and Melbourne music programs. The format of the CSIRAC punched paper tapes changed when the machine was shipped to Melbourne. Reading the program tapes was not difficult, but it was certainly tedious. Fortunately, there was a mechanical reader for the Melbourne tapes that was built by John Horvath of the University of Melbourne's computer science department. John Spencer was using this to store the contents of all of the paper tapes, the subroutine library, etc., as PC disk files, and he also ran the music program tapes through the reader. There were a few problems, mostly with paper tapes that had become torn on the edges, as these tended to tear more and jam as they went through the reader. The tapes were read multiple times to ensure an accurate reading. There was, however, no automated reading device for the Sydney paper tape format, so this was read by hand. John taught me how to do this, and we both undertook the task and compared the results. After a few corrections, we had an accurate reading of the Sydney tape and all of the music tapes as text files on a PC.

Somewhere along the way, the idea occurred to me that we could possibly split the reconstruction task into two components: (1) reconstructing the timing of the pulses sent to the speaker and (2) reconstructing the pulse shapes sent to the speaker. This turned out to be a key approach to solving the problem. The aim, simply stated, was to reproduce precisely the pulse stream and thus the sound that emerged from CSIRAC's loudspeaker. After some experimentation, we discovered that the best course of action would be to read the program and data tapes (and get them working in the emulator, a non-trivial matter), use several programs John Spencer developed from his emulator to generate the loudspeaker pulse timing data, reproduce with hardware the pulse shapes that appeared at the loudspeaker terminals, and then combine these to reproduce the pulse stream. This pulse stream could then be played through a loudspeaker and recorded if necessary.

The CSIRAC emulator that had previously been developed by John Spencer would now be used to run the music programs. The emulator could not output sound the way CSIRAC did owing to the limitations of the PC architecture, and it could not be modified to do so. It was decided that the simplest way to proceed was to develop programs from the emulator core that would write a file containing the timing of the loudspeaker pulses. This file contained details of when pulses were sent to the loudspeaker, which pulses were sent, and the interpulse timing details. The core CSIRAC emulator code was not designed for this, and Ron Bowles worked extensively with John Spencer to refine the timing of each instruction in the new program that would gather the correct timing details of the pulses sent to the loudspeaker. Ron Bowles still remembers the precise details of the timings of each memory location access, the timings of accessing the various sources and destinations, and their interdependencies. This is crucial information for a real-time activity, such as playing music, and it is not documented anywhere with any precision. Ron Bowles drew up tables of the exact timing of each memory location access, register access, and instruction, including how that varied with prevailing conditions within the machine. This information was then incorporated into the program developed from the core code of the emulator to gather the pulse timing data. After the timing details were refined, it was a relatively simple matter to put in place a data file output for the pulses being sent to the loudspeaker destination.

While the paper tapes were being read and the pulse timing programs were being developed, Jurij Semkiw was designing and building logic circuitry to reconstruct CSIRAC's pulse shape. The logic design allowed the repetition of a single word, with the exact timing of the bits as in CSIRAC's logic circuits. John Spencer had built an exact reproduction of the valve output amplifier (including an original output transformer) based on CSIRAC's circuits, and to achieve an accurate pulse shape, the output was sent through this amplifier with the same sort of loudspeaker attached that was used on CSIRAC. Pulses were played at various frequencies and with all combinations of bits turned on. This was checked with an oscilloscope to verify that the pulse shapes on the output matched the original pulses generated by CSIRAC according to traces found in the CSIRAC archive. It was also checked aurally with a loudspeaker to verify that it sounded the same as CSIRAC to listeners who remembered the machine's sound. This output was recorded onto a DAT recorder from across the loudspeaker terminals. The digitized pulses became data structures in another program, developed by John Spencer, that would read the file of loudspeaker pulse-timing data and apply the correct pulses in the time relationship specified in the timing file. This program wrote an output file of the pulses at

Figure 3. Oscilloscope screen image of the reconstructed speaker pulses. The four traces are for six bits on, four bits on, three bits on, and one bit on, from top to bottom.



the correct times, in effect a digital audio file of what CSIRAC would have produced at the time if a digital recording device were connected across the loudspeaker terminals. Figure 3 shows an oscilloscope screen image of the reconstructed speaker pulses.

We were interested in reproducing the music as exactly as possible, certainly with an error of less than one percent with respect to the waveforms that would have been heard at the time the pieces were originally played. The pulse shape as reproduced was indeed well within one percent tolerance of the CSIRAC pulse shape, but the pulse timing was at least ten times more accurate than that. This waveform accuracy would ensure a listening experience faithful to the original and would also ensure that any technical analysis of the waveform would be valid. The digital audio files of the music as played by CSIRAC are an extremely accurate representation of the pulse stream that was sent to the original loudspeaker and a very good representation of how CSIRAC would have sounded.

One further step would provide the greatest authenticity. The original loudspeaker used in CSIRAC, a 5-inch Rola model 5C, had a torn paper cone, so another was found that was in excellent condition and close in manufacturing serial number to the original. This loudspeaker was placed in the CSIRAC console door, and the music was played through it from a high-fidelity compact disc player and amplifier with a low output impedance to exert the minimum influence on the perceived sound and the waveform. The output was recorded to digital audio tape (DAT) with a microphone. When the sound of this recording is compared to the digital audio files that resulted from the reconstruction, it sounds more animated owing to the various resonances and noises from the speaker and console, and it is a more faithful representation of how it would have sounded originally.

Pitch and Sound Analysis

Upon hearing the music, a modern listener will be alerted to several tuning anomalies. Table 3 shows an analysis of some notes CSIRAC was programmed to play in both Sydney and in Melbourne. This table shows the standard note name (pitch class and octave number), the note number in Mr. Cherry's program, the frequency of the note in the equal-tempered scale in A440 tuning, and, for both Sydney and Melbourne, the fundamental frequencies of each note and the number of pulses used at each of those frequencies when CSIRAC played music. (Note that "× 1" is implied if no number of pulses is indicated.) These frequencies are based on the periods of the pulse loops in the music program tapes. Thus, they are fundamental frequencies for the fairly sawtooth-shaped waveform used to create the sounds, not the result of Fourier analyses of the notes, which is investigated later. There are many notes that use multiple pulse timings, often repeated, to approximate a note.

It can also be seen that some note loops combined up to five different pulse timings (see D4, Sydney) to try to produce a single note. Other note loops used up to six pulses in a loop but with several of the same period, see for example the notes

Table 3. Analysis of Some Notes CSIRAC was Programmed to Play in Both Sydney and in Melbourne

Pitch	*Note number*	*Equal-tempered frequency (Hz)*	*CSIRAC note frequencies and number of pulses, Sydney (Hz)*	*CSIRAC note frequencies and number of pulses, Melbourne (Hz)*
C2	0	65.41		65.1
F-sharp2	6	92.5		183.2, 92.1
G2	7	98		196.1, 97.5
A2	9	110	(108.7) × 2	(109.6) × 6
A-sharp2	10	116.5	115.7	115.7
B2	11	123.5	(122.5) × 2	(122.5) × 2
C3	12	130.8	(130.2) × 2	130.2
C-sharp3	13	138.6	138.8, 160.3	(138.9) × 2
D3	14	146.8	(144.9) × 4, 143.7	(146.2) × 7
D-sharp3	15	155.6	157.2, (155.7) × 2	(154.3) × 6
C4	24	261.6	(260.4) × 3	260.4
C-sharp4	25	277.2	(277.8) × 3, 219.3	(277.8) × 2, 138.9
D4	26	293.7	282.5, 287.5, 282.5, 142.5, 222.2	(292.4) × 5, 127.2
D-sharp4	27	311.1	248.8, (326.8) × 3, 166.7	(308.6) × 6, 154.3
F-sharp4	30	370		(370.4) × 2, 362.3, 370.4, 182.0
G4	31	392		(387.6, 793.7) × 3, 260.4

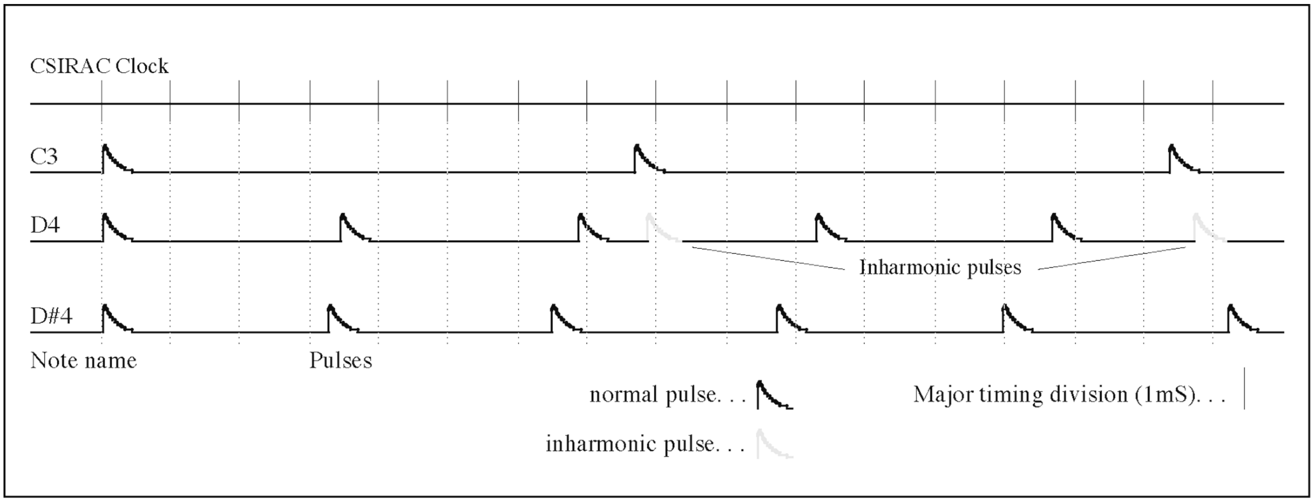
from D-sharp4 onwards in the above table. When several frequencies are listed for a note, a frequency that repeats several times does not necessarily mean that the pulses at that frequency all occur in a row; a pulse with another "period" can intervene. In other words, these are not necessarily sequential lists of pulse timings. The combination of multiple different pulse timings was used to approximate some notes owing to the troubles caused by CSIRAC's timing limitations. The variation in timing of speaker pulses was caused by the variation in memory access times, the machine architecture, and the timing granularity caused by the relatively low clock speed.

From Table 3, it can be seen that some notes, for example D4, have clearly inharmonic components. Particularly for the note D4 at Melbourne, one might think that if five pulses can be created in sequence that are very close to the desired frequency, then one more should be possible. However, the last pulse, providing a frequency of 127.2 Hz, has no harmonic relationship to the note D4. Careful analysis of Mr. Cherry's music program has revealed this is owing to the various machine limitations or idiosyncrasies. He must have realized through experimentation that the low frequencies were being produced and why, so he made adjustments to minimize their dissonance. With the note D4, he achieved the best result possible. The structure of the note loop is several pulses with the correct period, but the memory was limited, and there were insufficient memory locations with the appropriate timing to store another pulse, so an inexact delay had to be introduced, and exiting the loop earlier caused other problems. This caused the inharmonic pulse.

Figure 4 displays the pulse periods for three notes at Melbourne. It can be seen how different pulses in a note need a different timing offset, or delay, from a clock division. This is a graphical display of what the typical code for a note produces. The pulses that result in the inharmonic 127.2 Hz frequency for the note D4 at Melbourne can also be seen.

When listening to the sounds generated by CSIRAC, it becomes clear that there are pitches in the sound that do not appear in the table of how the notes were programmed. A spectral analysis of the notes was used to show these pitches, and help to investigate and explain the sound or timbre of the various notes. The spectrum graphs in Figures 5 and 6 were taken from the pulse stream as deliv-

Figure 4. CSIRAC major cycle timing with pulses for notes C3, D4, and D-sharp4 at Melbourne. This shows the various delays required from the onset of each instruction cycle to produce a pulse with a specific period that is not necessarily simply related to the machine cycle time. The different delays required between pulses were achieved by using different memory locations and instruction addresses.



ered to the speaker terminals. The speaker would filter this stream, mostly by reducing the higher partials. The vertical scale is linear amplitude, and the horizontal scale is logarithmic frequency with a range from 10 Hz to 10 kHz.

The spectrum for the note C3 at Melbourne (see Figure 5) shows the harmonics produced by CSIRAC's near-sawtooth pulse shape. This spectrum is highly representative of most of the notes that CSIRAC played, certainly from C2 to C4, the lower two octaves of the two-and-a-half octave range, as most of the notes have simple harmonic components.

The upper notes of CSIRAC's range were the most difficult to program, as discussed previously, which is clearly audible from listening to them. The note D4 at Melbourne (see Table 3 and Figure 6), with a fundamental frequency of 292.4 Hz, has a pulse timing that gives rise to an additional frequency component at 127 Hz. However, this does not explain all of the frequencies that appear in the output. The upper notes (as all others) sound like an accurate reconstruction to those who heard them when CSIRAC was operating. There are clearly audible lower frequencies in the output that sound dissonant or inharmonic. The spectrum shows, surprisingly, not only the expected frequencies at 292 Hz and 127 Hz and their associated harmonics, but also a frequency component at about 40 Hz. The 40-Hz component also appears to give rise to many odd side-band frequencies of all of the harmonics. This would explain why the note sounds quite inharmonic. However, the existence of the 40-Hz component took some investigation to explain. The difference between the planned fundamental of 292 Hz and the second harmonic of the spurious 127-Hz frequency (254 Hz) is 38 Hz. This is quite close to the mystery 40 Hz, but the spectral analysis shows not only a clear 40-Hz component but also clear harmonics of it at 80 Hz, 120 Hz, and so on. The precision of the measurement makes it unlikely that it is a difference frequency.

After considerable investigation, the reason for the 40-Hz component became clear. We found that for all notes, the program structure was a loop to send multiple pulses to the speaker at accurate periods, if sometimes not the desired period. This loop is executed a number of times to provide control over the duration of the note. For the note D4, the periods of the pulse loop are 3.42 msec, 3.42 msec, 3.42 msec, 3.42 msec, 3.42 msec, and 7.76 msec. The sum of the loop pulse periods is 24.96 msec, which gives a frequency of 40.06 Hz. The loop return causes a modulation of the pulses at the period of the execution of the loop. This gives the 40-Hz component in the output. The seven highest notes played by CSIRAC all have this structure, and they also have the spurious low-

Figure 5. Spectrum of note C3, fundamental 130.2 Hz. This is representative of most of the notes played by CSIRAC, such as the notes in the two octaves from C2 to C4.

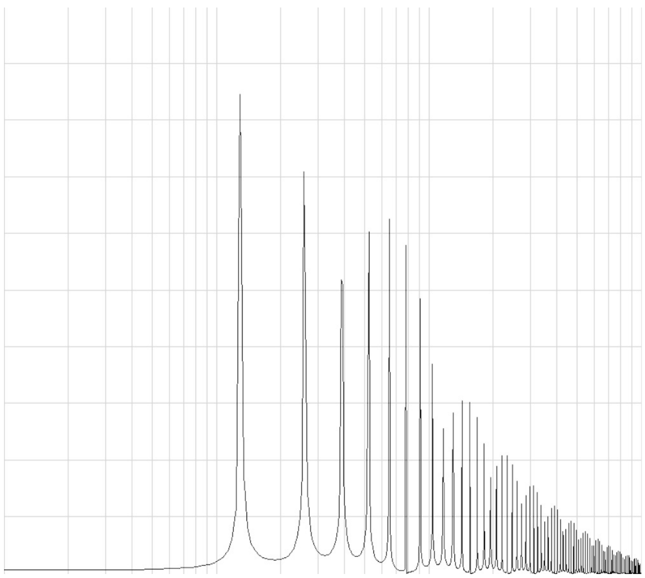
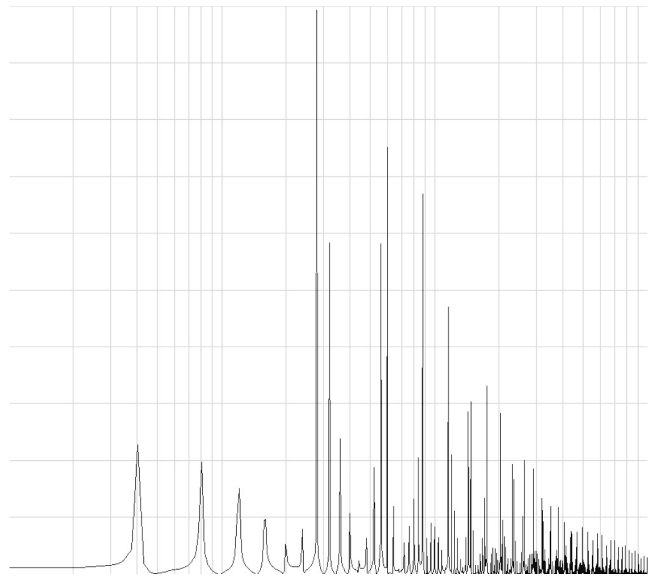


Figure 6. Spectrum of note D4, fundamental 292.4 Hz. Note the low-frequency artifacts, as discussed in the text.



frequency component in their output. With some of the other notes, it was possible to cause this low-frequency component to have some harmonic relationship to the desired note, thus reducing somewhat the dissonance created by its unavoidable presence.

Conclusion: What Was and What Was Not

CSIRAC was a unique technological achievement in Australia. It was also part of a unique musical achievement. CSIRAC, or the CSIR Mk1, was one of the very first computers in the world to make music, and possibly even the first. It accomplished this through a happy confluence of events and the very intelligent and diligent work of a few people. The technological feat was extraordinary, given the state of the hardware used to support the programming and produce the sounds. Similar activity was taking place slightly later in other parts of the world, for example England, and hopefully these activities will also be well documented in the future.

The music itself may now seem crude unless it is understood in the context of its creation. It was created by engineers who were not knowledgeable of the latest in musical composition practice and at a time when there was little thought of digital sound. The idea of using a computer, the world's most flexible machine, to create music was certainly a leap of imagination at the time. It is a pity that composers were not invited to use CSIRAC and discover how it could have solved several compositional problems as happened at Bell Labs. CSIRAC had much to offer composers: it could play almost any frequency within its range, so those interested in microtonal music or alternative scales and tunings would have found it invaluable. It could also possibly have been programmed to play more than one frequency at a time, although that would have posed a significant programming challenge. Additionally, with suitable programming, CSIRAC could have played any rhythm and could have produced rhythmically complex pieces. There was also the possibility of using it as a composition tool, in which some composers were interested, and which some of the programmers would have been happy to help implement. Perhaps we would have had very early examples of computer-based microtonal music and algorithmic composition.

Nearby composers were interested in these things. Percy Grainger was at the University of Melbourne at the time, and he was already known as an experimental composer who was interested in new sounds, ''free music,'' electronic music, and so on. Peter Thorne (2000) recalls: ''I can remember Percy Grainger walking past the Computation Laboratory at the time CSIRAC was running, actually walking down the alleyway between what would have been the cyclotron and [the] Physics [Building]. The others in the laboratory pointed out of the window and said, 'There's Percy Grainger.' He was going towards the Grainger Museum. He was that close. It must have been in about 1959. Grainger was at the University when CSIRAC was operating.''

It is indeed tempting to imagine what might have happened if some turn of events had nudged this solely physical proximity into a meeting and collaboration. This lack of professional musical input to the musical work, consistently the case for CSIRAC from 1949 to 1964, stands in stark contrast to the approach undertaken at Bell Labs. Max Mathews and John Pierce were both engineers, but they quickly recognized the need to involve musicians in their computer music work and they made a concerted effort to seek out appropriate people. The involvement of composers at Bell Labs led to some of the crucially important design decisions that in turn led to the development of current computer music. Lamentably, this did not happen with CSIRAC, perhaps because of its unfortunate isolation both geographically and culturally. CSIRAC had been programmed to play sounds, it had been developed as an instrument, and it offered new musical possibilities, but it was not used to evolve music as an art. If composers had become involved from the earliest development, the fundamental questions of ''Why would anyone want to use a computer to generate music?'' and ''What does the application of technology mean for the aesthetics of music?'' would have been addressed (Hiller and Isaacson 1959). Even if this activity did not lead to an historically significant, specific outcome or an understanding of the application of computers to music, the musical community would have been actively thinking about it and engaged with it. As it happened, this would have to wait until the events in the United States became public.

It appears that the musical programming of CSIRAC, both in Sydney and in Melbourne, was not as well appreciated as it could have been and thus not as well promoted as it deserved to be. With more enthusiasm and imagination, mostly from administrators, or with some joint research and development involving scientists and musicians, composers could easily have become involved with CSIRAC, and there could have been more exciting musical developments. As it stands, there is no enduring legacy of the music made with CSIRAC. It is notable owing to the leap of imagination required to conceive of using this very early computer to make music when there was no previous practice, or at least no known practice. It is also notable for the programming skill and cunning required to produce sound and music with this early machine.

Acknowledgments

It has been a pleasure to be involved with some of the original pioneers of computing in Australia. I hope that I have made clear the roles played by the people who worked with CSIRAC. I would like to thank all of them and the interviewees (Reginald Ryan, Terry Holden, Kay Thorne, Peter Thorne, Ron Bowles, Eileen Hill, Douglas McCann, and Dick McGee) for their generous time and assistance.

Much of the anecdotal and oral material was gathered over long periods of casual and personal contact with the people involved. For example, there were several pieces of the puzzle that came out during the CSIRAC 50th birthday celebrations in November 1999, and there were regular meetings on Tuesdays with John Spencer, Jurij Semkiw, Ron Bowles, Peter Thorne, Doug McCann, and others.

I am particularly grateful to Ron Bowles, Jurij Semkiw, and especially John Spencer for their gracious patience with my naïve questions, their generosity, and their skills, without which this project could never have progressed beyond the imagination stage. Many thanks.

The University of Melbourne Computer Science Department has been extremely helpful and supportive, providing access to people and facilities. Thanks are due especially to Peter Thorne and Leon Sterling for this assistance.

Special thanks to Paul Berg for reviewing this project.

This project was very generously assisted and funded by the Australian Commonwealth Government through the Australia Council, its arts funding and advisory body.

More information on this project is available online at www.cs.mu.oz.au/csirac.

References

Bird, J. 1999. *Percy Grainger*. Sydney, Australia: Currency Press.

Bowles, R. 2000. Interview with the author. 25 February.

Burton, C. P. 2000. Personal correspondence. 8 February–12 July.

Cage, J. 1967. *Silence*. Cambridge, Massachusetts: MIT Press.

Chadabe, J. 1997. *Electric Sound: The Past and Promise of Electronic Music*. Upper Saddle River, New Jersey: Prentice-Hall.

Dean, J. 1997. *CSIRAC: Australia's First Computer*. Killara, New South Wales, Australia: Australian Computer Museum Society.

Ernst, D. 1977. *The Evolution of Electronic Music*. New York: Schirmer.

Hill, E. 2000. Interview with the author. 2 March.

Hill, G. W. 1954. ''Programming for High-Speed Computers.'' MSc Thesis, University of Sydney, Australia.

Hiller, L. A., and L. M. Isaacson. 1959. *Experimental Music*. New York: McGraw-Hill.

Holden, T. 1997. Interview with the author recorded by Steven Pass. 7 June.

Manning, P. 1993. *Electronic and Computer Music*. Oxford, England: Oxford University Press.

Mathews, M. 2000. Personal correspondence. 31 August.

McCann, D. 2000. Interview with the author. 15 May.

McCann, D., and Thorne, P. 2000. *The Last of the First: CSIRAC: Australia's First Computer*. Melbourne, Victoria, Australia: University of Melbourne Computer Science and Software Engineering.

McGee, D. 2000. Interview with the author. 20 May.

Pearcey, T. 1996. Interview with Steven Pass, Doug McCann, and Peter Thorne. 1 October.

Pierce, J. 1995. ''Recollections by John Pierce.'' In liner notes to accompany *The Historical CD of Digital Sound Synthesis*. Mainz, Germany: Wergo.

Roads, C. 1980. ''Interview with Max Mathews.'' *Computer Music Journal* 4(4):15–22.

Roads, C. 1996. *The Computer Music Tutorial*. Cambridge, Massachusetts: MIT Press.

Ryan, R. 2000. Interview with the author. 25 January.

Thorne, K. 2000. Interview with the author. 21 February.

Thorne, P. 2000. Interview with the author. 16 February.

Williams, M. R. 1997. *A History of Computing Technology*. Los Alamitos, California: IEEE Computer Society Press.