The Music of CSIRAC

AUSTRALIA'S FIRST COMPUTER MUSIC

Paul Doornbusch

The Music of CSIRAC

AUSTRALIA'S FIRST COMPUTER MUSIC

Paul Doornbusch

First published in 2005 by Common Ground Publishing, Australia.

The National Library of Australia Cataloguing-in-Publication data:

Doornbusch, Paul, 1959- .
The music of CSIRAC : Australia's first computer music.

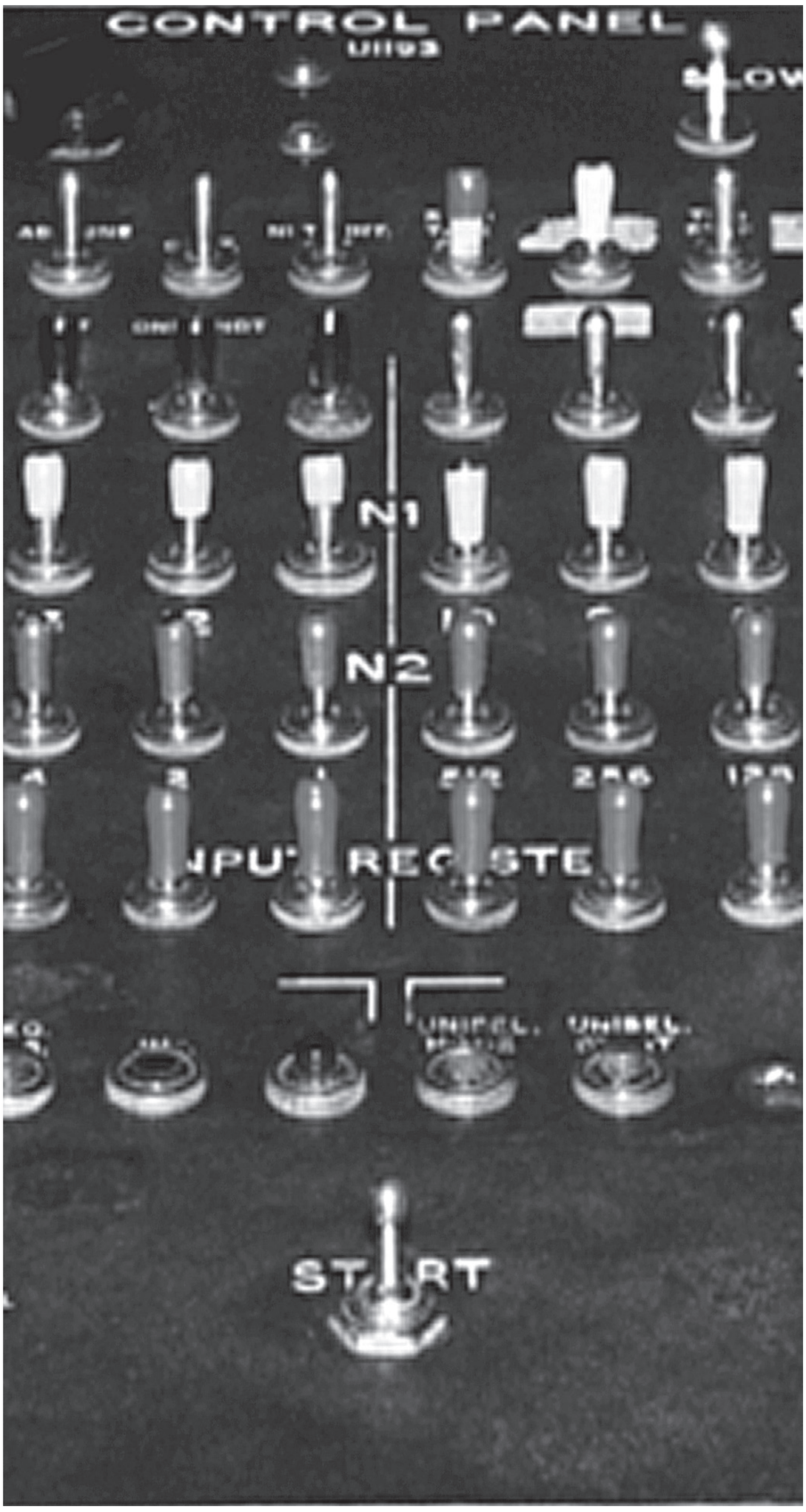
Bibliography.
Includes index.
ISBN 1 86335 569 3.

1. CSIRAC (Computer). 2. Computer music - Australia. I. Title.

786.76

Pearcey
FOUNDATION

Contents


CONTROL PANEL
N1
N2
START

Preface

It was in 1998 when, being a good expatriate Melburnian, living in The Hague and working at the Royal Conservatory of Holland, I was in my office reading The Age (Melbourne newspaper) on-line and I read the obituary of Trevor Pearcey. Interesting pieces, obituaries–a summary of a life in a few paragraphs. However, Trevor Pearcey's obituary was more extensive than usual and more like an article, as he was the designer (with Maston Beard) of Australia's first computer, the CSIR Mk1 (later called CSIRAC).

I had heard of this machine eons ago when I was an undergraduate at The University of Melbourne studying Computer Science. The most remarkable thing I remembered about this computer was that it had mercury delay-line memory. The idea of storing information as waveforms in liquid metal struck me as completely bizarre then, as I was weaned on silicone (although I had worked with ferrite core memory). However, in the obituary or article was a single comment about CSIRAC that leapt out at me: "It played music in 1951." Being a specialist composer in computer music, and working in one of Europe's most prestigious electronic and computer music facilities, I *knew* that computer music started in about 1957 with the work at AT&T Bell Labs in America. The 1951 date must have been a 'typo'. I let it pass. Nevertheless, curiosity got the better of me and as I was an alumnus of the institution and department involved and I knew the people there, I sent off some email. Time passed. There were return mail messages pointing to various people and eventually I emailed Peter Thorne and Doug McCann. They were adamant and confident that CSIRAC had played music in 1951, there was evidence…I was quietly astonished. I discussed it with my colleagues and they were either somewhat astonished or fairly sceptical.

However, unfortunately I was told the music had never been recorded. There were several mentions in the emails, in the weeks and months that followed, of the music tapes being "around somewhere", but nobody knew where. Eventually I

understood this to mean not magnetic recordings of CSIRAC but *punched-paper program tapes*. I would have to get used to the terminology. But apparently the program tapes were *somewhere*, that is, they existed, so I had more email with Peter Thorne about the feasibility of reconstructing the output of these music programs. Peter was highly sceptical. The communication went back and forth between Melbourne and The Hague and I started to learn more about CSIRAC, its internal workings and so on. I was convinced that somehow it must be possible to reconstruct the music played by CSIRAC—anything is possible if we apply ourselves enough.

Peter Thorne was less convinced, but he agreed to be a partner in a grant application to fund the attempted reconstruction. We won the grant and I returned to Australia and started the project. I still had no idea *how* to reconstruct the music, but shortly before I returned one of the music program tapes was found so it was looking positive. As the obvious tasks for the project were undertaken, the way forward eventually became clear. It was much more difficult than I had first thought in Holland. Perhaps this has made it all the more worthwhile and remarkable. The story unfolds in what follows.

I would like to dedicate this book and project to all of those men and women who worked on CSIRAC during its 15 or so years of activity, they are a remarkable crew, and to all of those brave and pioneering souls who dreamed of making music with computers.

Acknowledgments

Many people have helped with the re-creation of this work and have provided encouragement to me throughout this project. It has been a great pleasure to be involved with some of the 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 for their generous time and assistance. 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, you made this project possible and it was only by standing on your shoulders that I have been able to see further. The interviewees are; Reginald Ryan, Terry Holden, Kay Thorne, Peter Thorne, Ron Bowles, Eileen Hill, Doug McCann and Dick McGee, all of whom gave very generously of their time and knowledge.

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.

Thanks to Museum Victoria, especially Catherine Lovelock, Fiona Kinsey and David Demant who allowed me unfettered access to CSIRAC, now a prized museum object, to achieve the most faithful re-creation possible of the music. Thanks is also due to the Board of Museum Victoria for assisting in the publication of this work.

I am very grateful to the many other people who have helped 'behind the scenes', providing their time, encouragement, expertise and sometimes equipment. These People are; John Crawford, Douglas McCann, Judith Hughes, the Technical Services staff of the Department of Computer Science & Software Engineering department of The University of Melbourne. I am also very grateful to Steven Pass who graciously allowed me access to his video archive to provide a video excerpt on the CD, and the Pearcey family

for permission to use this.

Chris Burton very kindly provided unique information about early British computers through private correspondence and his connection with the Computer Conservation Society, which has been very useful and helpful.

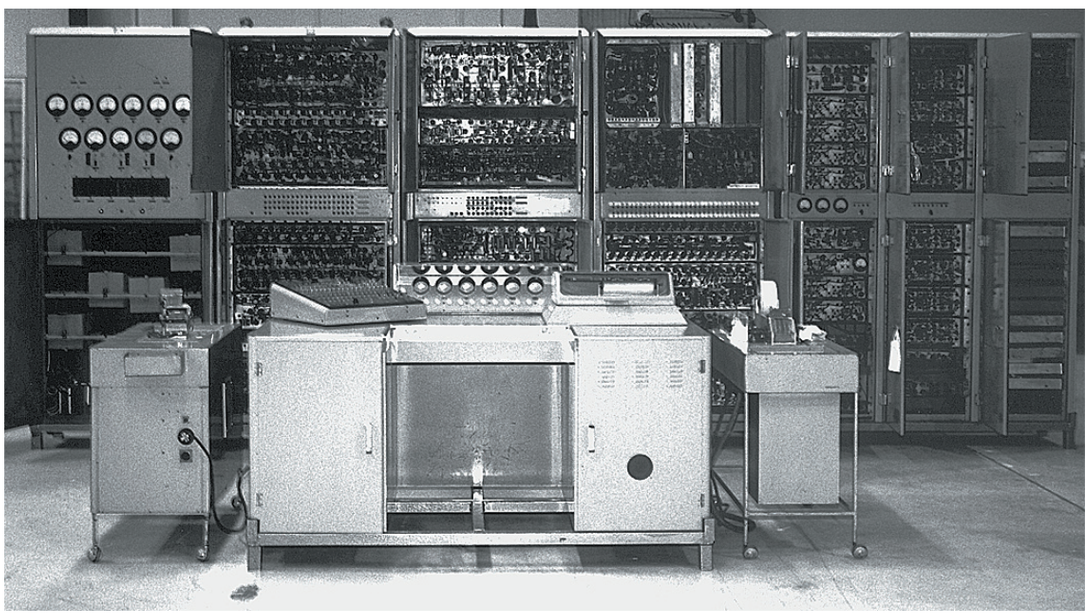
Thanks also to Gus Gollings for his tireless work, understanding and assistance in the editing, layout and publication.

Many thanks to Sarah Kenderdine for being so supportive and for the inspired ideas in the final preparation of this work which helped tremendously to bring it to fruition.

Special thanks to Paul Berg for many years of inspirational mentorship and for reviewing this project.

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

In memory of Ron Bowles.



CSIRAC as displayed for its 50th birthday celebration, Museum Victoria, 25 November 1999.

Introduction

Computer music has always been a rather ill-defined concept. It has something to do with music and something to do with computers. Just like beauty, it is in the ears of the beholder.

The use of computers for music production and reproduction is now considered commonplace, both by artists and consumers. For the former, computer music may still retain the aura of a separate field of academic or artistic endeavor. For the latter, it is a fact of life.

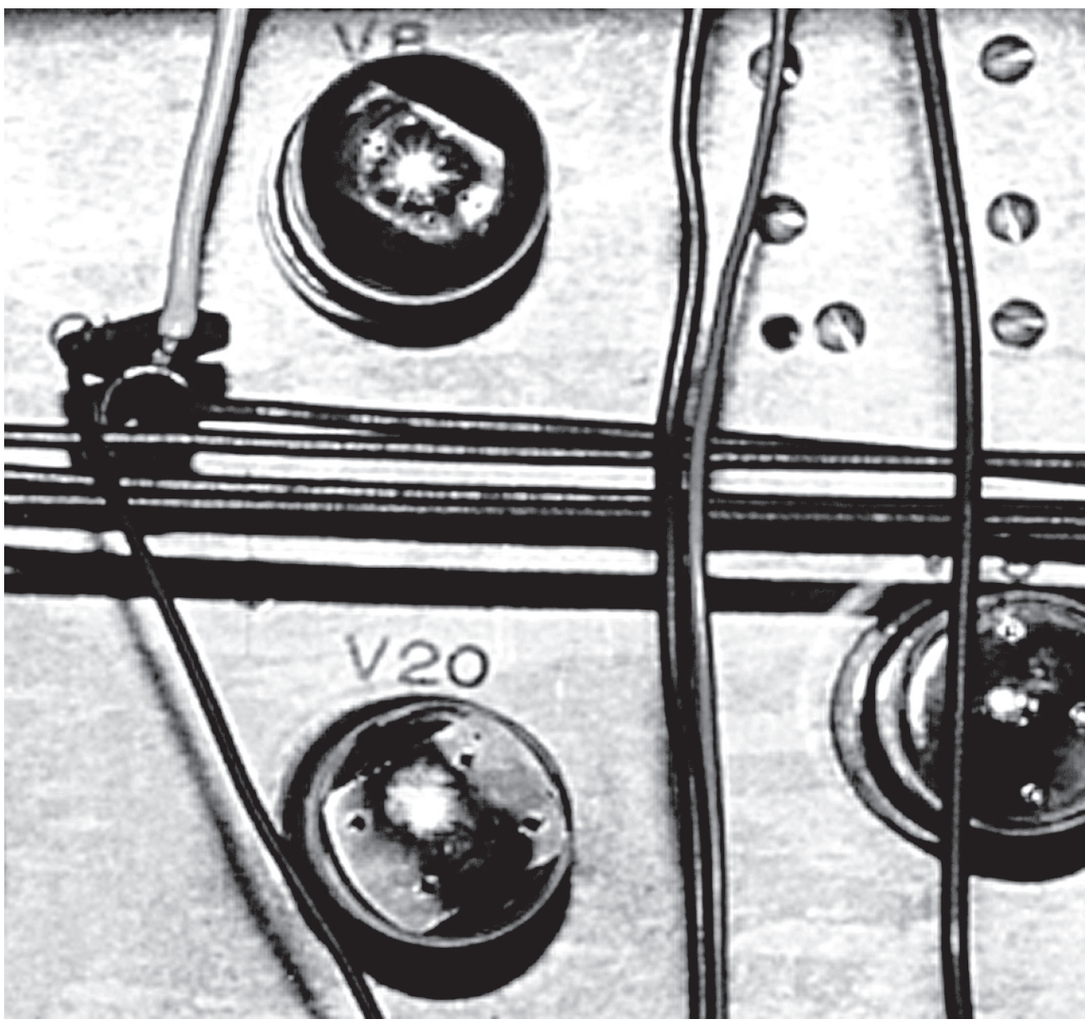
In both cases, the path that led to this usage is the result of many very small steps. Steps that were neither coordinated nor goal-oriented. Steps that were not labeled merely scientific, consumer-oriented, or artistic. Ideas concerning microtonality, experimental music, complex compositional systems, musical instrument design, and the enthralling power of a computation machine all contributed to a climate where something called 'computer music' was created.

Based on these ideas, an historical narrative could be produced, joining all of these activities in a compelling story of inevitable advances. But perhaps an equally viable narrative would be the chronicle of engineering achievements that accidentally became musically useful. Achievements that a musician or perhaps an engineer realised could also have some musical application. Sine wave generators, pulse generators, noise generators, and filters are examples of laboratory equipment that were contraindicated for musical use. Digital computers are another example. In the case of CSIRAC, the hooter circuit could be added to this list. Having a loudspeaker driven by pulses allowed people with imagination to make musical experiments.

The technical challenges faced by all pioneers of computer music were enormous. Surmounting these challenges was a contribution to what is now a dominant musical activity. An artistic history might discount some of these 'buzzes and squawks'. But a history cognisant of the current artistic, scientific, and consumer-oriented reality, should take note of the effort and dedication that now can be seen as a piece

of the puzzle, part of the grand musical adventure of the Twentieth Century: the use of computers in music.

Paul Berg *Institute of Sonology, Royal Conservatory, The Hague*



Fade Up

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. 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. Trevor Pearcey, an English radio physicist, and Maston Beard, a researcher at the CSIR Radiophysics Laboratory in Sydney designed the CSIR Mk1. The first 'programmer' or real software engineer to work with the CSIR Mk1 was Geoff Hill, a mathematician who assisted with the logical design. Hill, who came from a musical family, programmed the CSIR Mk1 to play popular musical melodies from as early as 1951. The CSIR Mk1 was moved to Melbourne in June 1955 and renamed CSIRAC. It performed useful and trailblazing service at the University of Melbourne until 1964. During CSIRAC's time in Melbourne the mathematics professor Thomas Cherry programmed CSIRAC 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. Whilst the music performed by the CSIR Mk1 may seem crude and unremarkable compared to the most advanced non-computer generated musical developments of the time and what is possible now, it is amongst the first computer music in the world and the means of production was at the leading edge of technological sophistication at the time. These first steps of using a computer in a musical sense occurred in isolation. The decision to create music using the flexibility of a general computer required programming ingenuity and a leap of the imagination. CSIRAC took some initial steps in that direction.

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 overcome this problem calculating machines had been developed, such as the 'linear equations machine', the 'differential analyser', and the 'multi-register accounting machine'. However, the calculating machines still required a lot of 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.

There were two major technological advances of the time that allowed the realisation of an automatic calculator with memory. One such advance was the thermionic valve (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, such as the Electronic Delay Storage Automatic Computer (EDSAC) or the CSIR Mk1.

Trevor Pearcey worked in England on advanced radar systems from 1940 to 1945. During this time he became well-acquainted with the (electro) mechanical calculators that were available. His work finished after World War II and he travelled via North America to Australia to take up a position with the Australian Council for Scientific and Industrial Research (CSIR) in their Radiophysics Laboratory. While in America he had contact with other radio researchers, some of whom were using large university calculators in their work. He also had a chance meeting with Howard Aiken, who had recently completed a machine called the Automatic Sequence Controlled Calculator (ASCC and also known as the Harvard Mk1) for IBM at Harvard University. The ASCC was a very large and complex calculator, using relays and partially controlled by a punched paper tape. Having left England Trevor Pearcey in the last few months of 1945 arrived in Australia on 26 December 1945.

The CSIR Radiophysics division was active during World

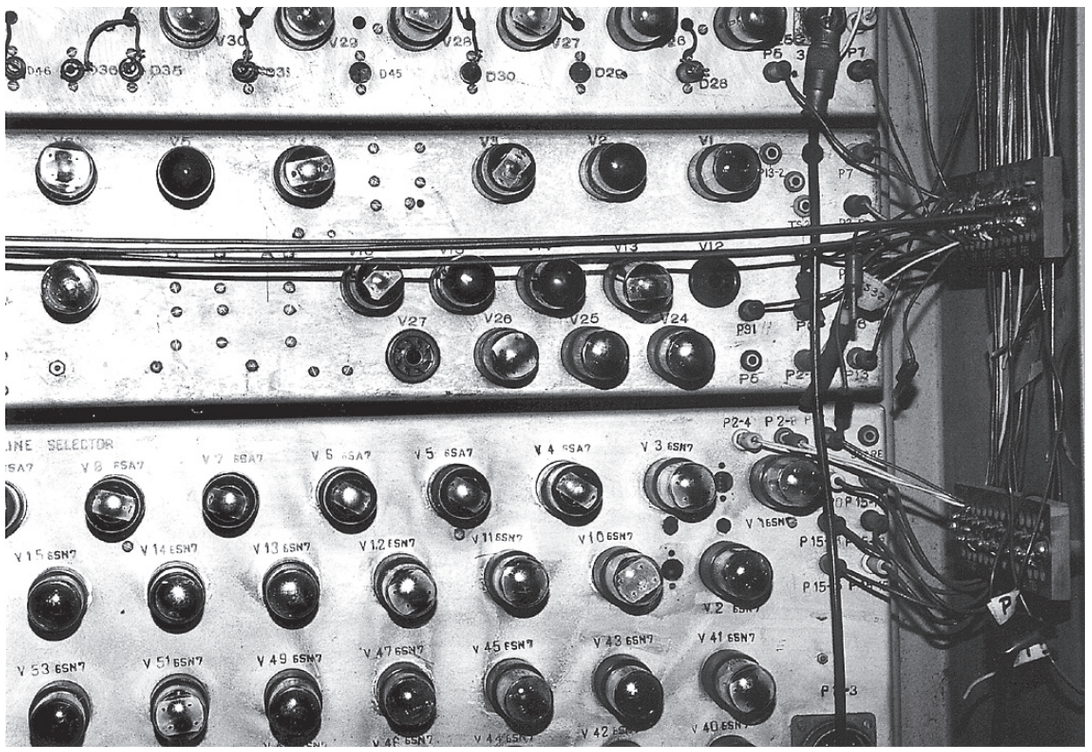
War II, developing radar systems for Australian servicemen in unique situations, such as portable radar for jungle warfare. In 1941, a talented electronics engineer, Maston Beard, joined the Radiophysics Laboratory. He would be instrumental in the development, with Trevor Pearcey, of Australia's first computer. After the war, the Radiophysics Laboratory had to decide what tasks it should pursue in peacetime. The management decided that Radiophysics should concern itself with Radio Astronomy, Rain Physics and Radio Propagation. Pearcey convinced the management to drop Radio Propagation and pursue Electronic Computation. This may seem now to be a strange area for the Radiophysics department to be involved in, but they were the leading experts in pulse technology and 'valve' electronics.

Trevor Pearcey began to assemble the basic concepts for building an electronic calculator that operated at high speed compared with available technology. Details of other electronic calculators of the time, for example the Electronic Numeric Integrator and Calculator (ENIAC), were secret and unavailable to Pearcey, but their existence proved that such technology was possible. For memory storage, mercury acoustic delay lines could be used. Instructions could be input to the calculator via a punched paper tape, as used by the ASCC, allowing the execution of a pre-defined sequence of operations. This was however, inflexible, but some recent theories and work pointed to a solution. In 1936 at Cambridge University, Alan Turing produced a new theory of what, hypothetically, could be computed. A 'Turing machine' can theoretically perform any calculation. John von Neumann, in about 1945 at Princeton University, provided a practical solution for Turing's theory where the commands and data (numbers) were coded in the same manner and stored in the same memory. This provided maximum efficiency, speed and flexibility, delivering the best solution for Trevor Pearcey. In early 1947, the 'Electronic Computer' project was officially commenced by Trevor Pearcey and Maston Beard. In addition, in this year the University of Pennsylvania published the details of ENIAC.

In April and June 1948, Trevor Pearcey completed and

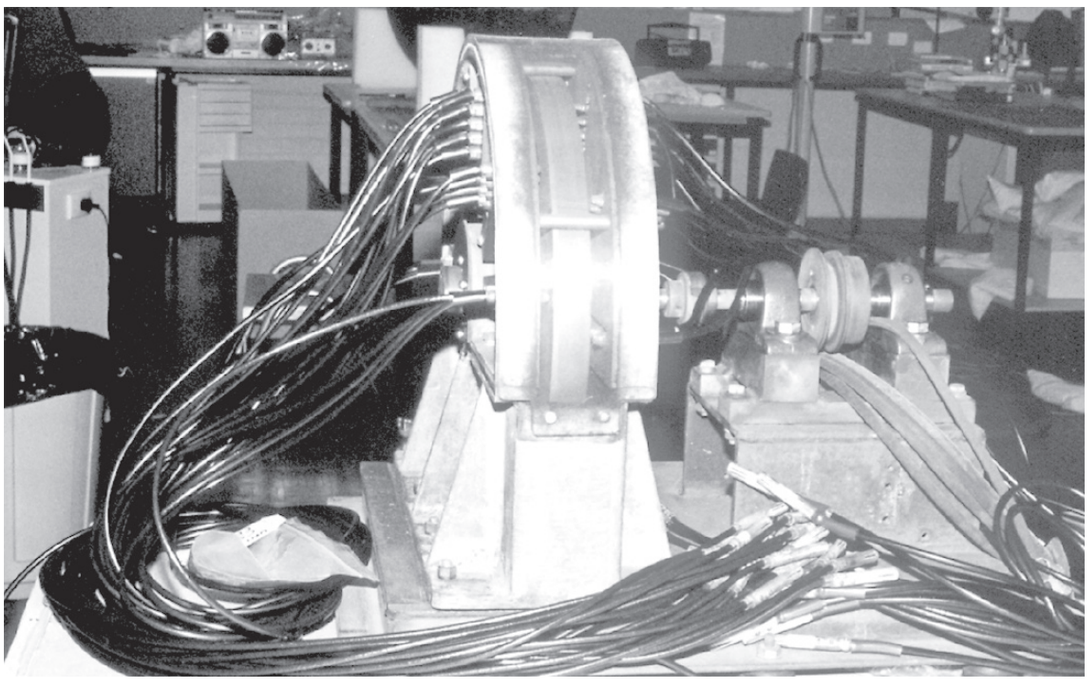
published the fundamental logical design of the computer in two papers. The overriding considerations of the logical design were engineering and programming simplicity, as this was intended to be a comprehensive prototype for a larger and more capable machine. Thus, the designers decided on a relatively small 20-bit word length, a serial architecture and a moderate clock frequency. The construction of the Automatic Computer (initially known as the CSIR Mk1, later known in Melbourne as CSIRAC) commenced in 1948, directed by Maston Beard who designed the electronic circuits to implement the logical design. The first program, a simple multiplication of 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.'

The CSIR Mk1 used electronic valves or vacuum tubes as its switching and logic elements. Only a few types of valves were used, mostly 6SN7, 6V6, 6SJ7, 6AC7, EA50 and KT61. Reginald Ryan added the primary memory store, which could contain up to 32 recirculating mercury acoustic delay lines. Each delay line was a five foot long special metal tube coated with lacquer and filled with mercury, where information could be stored as a sequence of pulses of 10MHz sine waves, 10 cycles for 1 bit. The original memory could store 16 instructions or 'words' (each 20 binary digits, or bits, long) per memory-tube, providing a maximum memory of 512 words. Ryan later doubled this by interleaving two 16 word pulse streams in each tube to give a maximum of 1024 words of memory. However, all of the memory-tubes were never completely functional or installed, although the reliability was improved in Melbourne by using internally polished stainless steel tubes. According to the original engineers, in typical operation the machine had 768 words of functional memory.



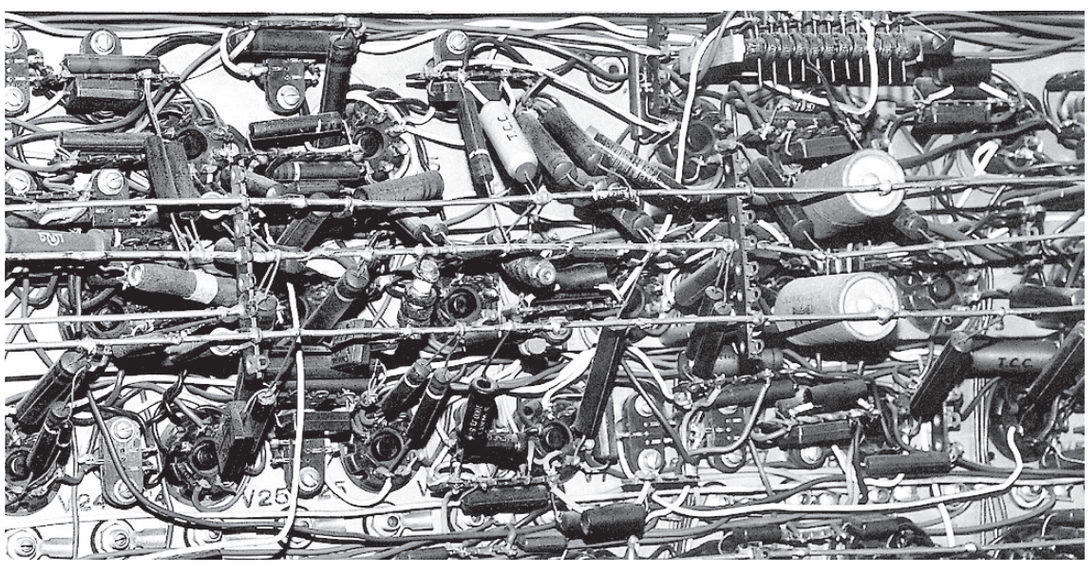
Detail of the inside of one of the racks from the side of the valves.

Brian Cooper started the development of a magnetic drum memory in 1950. It was running in 1952 with a capacity of 1024 words, an average access time of 5 milliseconds and a rotational speed of 6000 RPM provided by an aircraft gyroscope motor. The CSIR Mk1 operated at 500 operations a second with a clock frequency of 1,000Hz, a fair speed in its day but not as fast as some other machines of the time, as speed was sacrificed for design elegance and simplicity. However, the CSIR Mk1 was very different from today's computers because it was a serial machine. Data was sent around the computer from 'sources' to 'destinations', for example to and from memory, one bit at a time. Modern computers typically move around 32 or 64 bits in parallel. This difference in the CSIR Mk1's architecture had consequences for programming the machine and it is especially significant for timing critical programs such as those that may play music in real time.



The Melbourne disk, also called the drum, being cleaned for display. Note the washing machine drive belt and pulley on the right hand side, it was driven by a washing machine motor.

The CSIR Mk1 was notable because of the flexibility and economy of its instruction set. This was achieved through an instruction partitioning scheme that was of medium complexity, being more complex than the EDSAC or the Telecommunications Research Establishment Automatic Computer (TREAC), but less so than the EDVAC or the ACE. The simplicity and elegance of the instruction format resulted in a flexibility and economy of instructions, which benefited the development of complex programs that performed useful and complex tasks in a minimum of memory space.

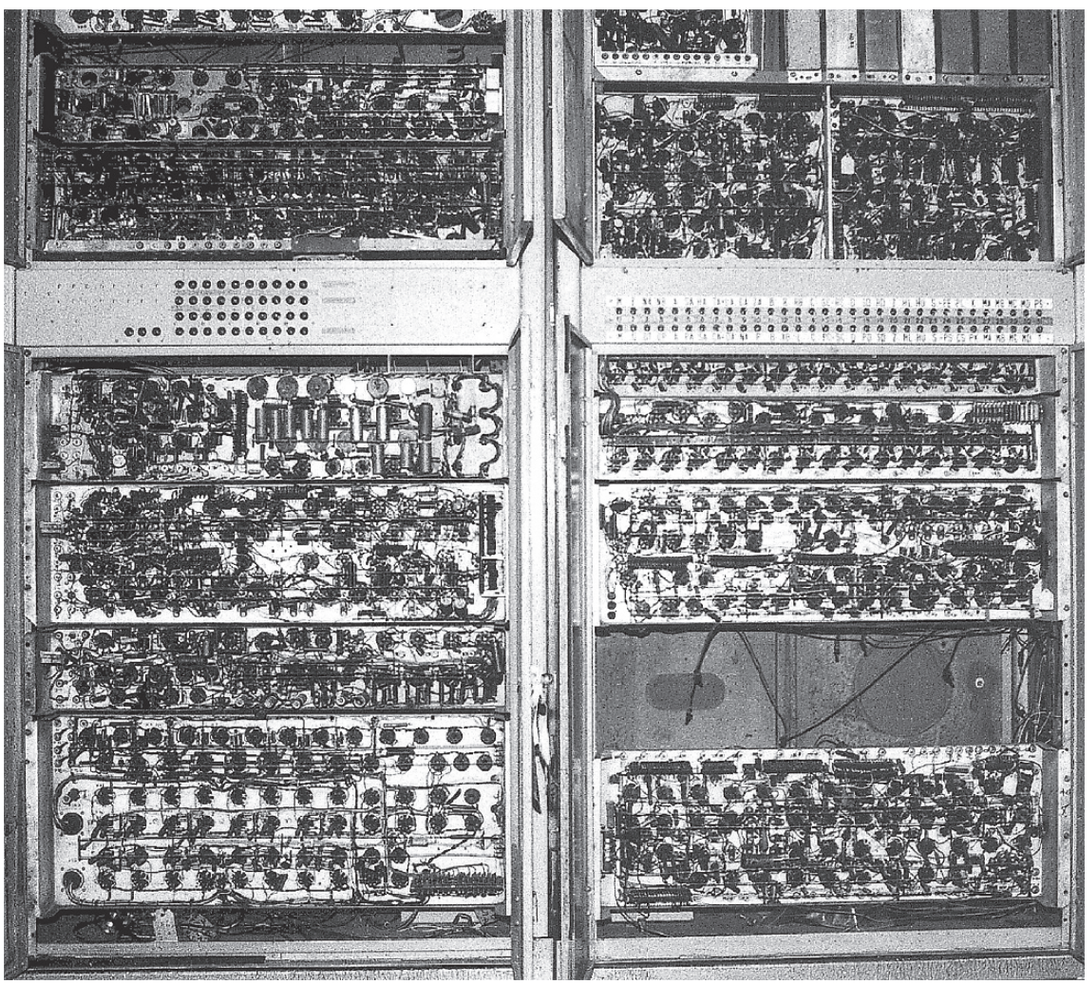


Detail of a part of the racks showing the construction techniques.

To understand something of the operation of the CSIR Mk1, 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 (there were 16 general purpose or 'D' registers), a memory location, the accumulator and so on. A destination could be a memory location, a register, the paper tape punch or the speaker, and so on. During the transfer the data could undergo transformation, such as being subtracted, or added. The instruction set partitioned the 20-bit digital words into three components: a 5-bit 'destination', a 5-bit 'source' and a 10-bit data address. The 10-bit data address, if it applied to the main (mercury delay line) store was further sub-divided into two 5-bit components; one to select which mercury delay line the data was in and another to select the position, or time, of the data in that delay line. Because the memory was a recirculating delay line and the whole machine architecture was serial, it was required to wait until a particular memory location was available for reading. The short word length was limited in its mathematical precision, but it was good enough for most of the engineering calculations that the CSIR Mk1 was to perform. For further precision, multiple word and floating point arithmetic representations and routines were developed. These occupied more of the limited memory space and took longer to compute than the simpler system.

Because of the serial architecture, there is no simple equivalent in the CSIR Mk1 to the common modern computer concept of 'clock frequency', which is usually the highest frequency of data transfer or operation. There were several timing clocks, or pulses, used in the CSIR Mk1. Most important for the purposes of the memory access times and sound production is the 1mS (really 960μS or almost 1000Hz) major-cycle pulse. This is the time required for one word to transverse the memory delay line. For the CSIR Mk1, at a minimum, one major-cycle was required to fetch a data word and another major-cycle was required to execute or transfer it. Thus, two major-cycles is the minimum time to execute an instruction, but if an opportunity is missed to access a memory address then it could take 3mS or 4mS.

As a result of the CSIR Mk1's serial architecture, the timing of instructions and memory access could vary and this was critical to some applications, such as producing a repeatable sound. Understanding the machine timing issues 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 access. It was possible to calculate this time and determine how long after the start of a clock or access cycle the data was read. Numbers were placed in specific memory locations in such a way that when they were read out and sent to the speaker, they were pulses with a pre-determined period. In this way a predictable pitch was produced and used to create musical melodies.



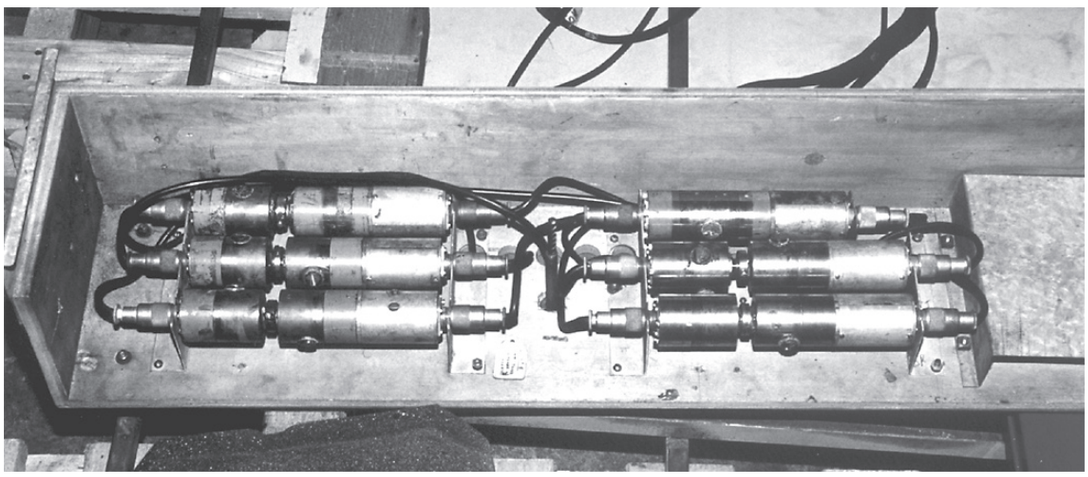
Two of the CSIR Mk1 racks opened to show the wiring.

Ron Bowles, a long-time maintenance engineer and the person who accompanied the CSIR Mk1 to Melbourne, knows most about the memory and machine timing:

> 'Well, something we called "speed-up" occurred at various times in the processing or fetching of data or instructions. Speed-up occurred in both halves of the computer cycle. The cycle was split in half. In the first half, the instruction was extracted from the memory and the decoding process was set up. In the second half of the cycle the instruction was decoded to determine what was required to be done and the instruction was executed… In the first half of the cycle, the position of the instruction in the 16 different possible *timing* positions in the memory-tube determines how far along in the computer cycle the machine must be before the instruction is available. Remember that CSIRAC is a serial computer. If the instruction occurs early enough in the memory, the fetching of it can be completed in 1mS instead of 2mS. Also, while decoding and executing the instruction there may be speed-up because within the instruction itself is an address … that could be the memory or a register. Now depending on that address, if it's early enough in the computer's cycle, then the instruction will be finished in 1mS. (Actually, it's 960µS, but we always said 1mS.) Thus, there's a total time of 2mS to fetch, decode and execute an instruction, which is as fast as the computer could operate. A computer cycle could thus take 2mS, 3mS or 4mS to complete before the next instruction could be undertaken … So the main thing affecting the timing is where the information occurs in memory, in which of the 16 positions. Thus, the way to vary the time in which the instruction actually takes place is by varying the address part of the instruction for whatever you are using for the source. So, even with a constant time for the whole operation, the time for a pulse to be sent to the speaker could be varied backwards and forwards. Sometimes, successive P (speaker) instructions would be in a loop with different addresses so that combined together they gave the desired result. This is how the high notes were achieved with any precision at all … The notes produced by the speaker are determined by the time intervals between successive speaker instructions rather than the computer cycle times in that set of instructions. Of course, these time intervals are intimately associated with the computer cycle times.'

The three arithmetic registers, two control registers and the half (H) register, shown below in their four-foot long wooden box, were also mercury acoustic delay lines. Each

one stored a single 20-bit word, except the H register, which stored only 10 bits.

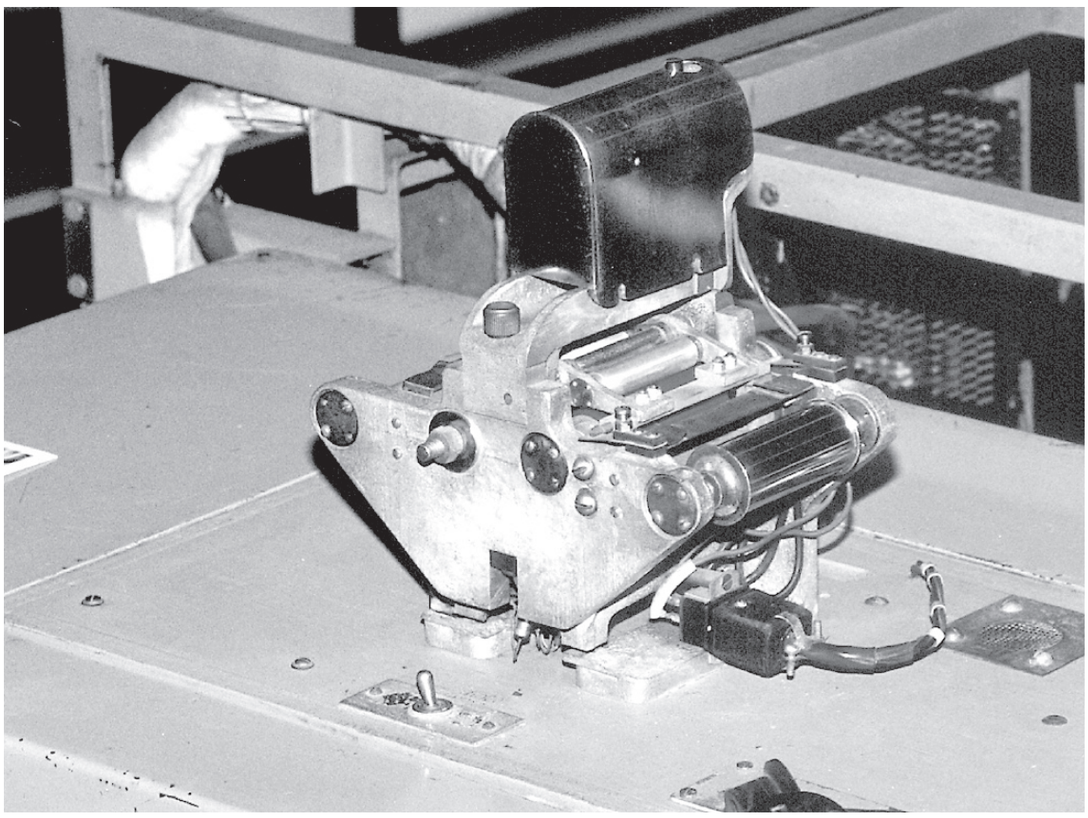


The arithmetic registers, control registers and H register mercury delay lines.

There were about 2000 electronic valves used in the CSIR Mk1, which formed its logic circuits. These electronic devices were not as reliable as the solid-state devices used today. Their mean time between failure was about 2000 hours. It is easy to see that in theory the CSIR Mk1 was not likely to run for more than an hour without a breakdown. In practice, it was considerably better than would be expected. This was largely due to an elaborate system of automatic power-on delays in the start-up sequence. Occasionally, the first time the machine was turned on in the morning, there would be one or more faulty valves. The machine would be powered up and stress tested by varying the main voltages on the valves. When the engineers had finished with the morning diagnostics, it would provide several hours of fault-free service. However, as it would sometimes break down during the day, it was common to run programs more than once to check the results. Important programs were run several times. Luckily, the maintenance engineers became very skilful and adept at quickly diagnosing and fixing the problems. The CSIR Mk1 was never left on overnight to improve reliability because it would have been unattended for twelve hours and have posed a fire hazard, besides which it was unnecessary due to the controlled power-on. The ambient temperature and the heat generated by the CSIR Mk1, or CSIRAC, affected the length

of the memory delay-line tubes and thus the propagation, or circulation, time of the tubes. To help overcome this problem there was a special additional delay line that held one pulse in the thermally insulated main memory box (called the coffin) which was used for clock timing adjustment. When there was variation in the memory propagation time as a function of temperature, this delay line, kept at the same temperature as the memory, would compensate for it and adjust the clock frequency. This kept the rest of the machine in sync with the timing of the memory delay-lines.

The CSIR Mk1 was, in many ways, operated like a modern-day personal computer. The operator sat at the computer console, pressed switches, and watched displays. There was a switch panel that contained three registers of switches, start and stop switches and so on. On CSIRAC's greatly enhanced console, post-1955, there was also a row of CRT displays that showed such things as the state of various parts of the machine and any bank of 16 words in the main memory. The speaker had various locations; in Melbourne it was situated in the door of the console, in Sydney it was ultimately situated in the much smaller console, but in the 1949–1951 period it was attached to a rack. Of course, the CSIR Mk1 had no operating system. When it was switched on one had a very blank machine, and a few ingenious solutions to the problem of how to bootstrap the machine were developed. These involved variously, entering codes manually on the console, a telephone exchange uniselector, a specialised tape instruction that could be stepped through from the console and the like.



Twelve hole punched paper tape reader, used to load programs and data into CSIRAC.

A few CSIRAC facts and figures:

— Used over 2000 valves
— Required 30,000 watts
— Weighed 7000kg
— 768 words of main memory (one storage 'word' is in effect equal to two bytes)
— 2048 words of disk storage
— 0.001MHz major-cycle frequency, achieving 0.0005MIPS
— Occupies 45 square metres

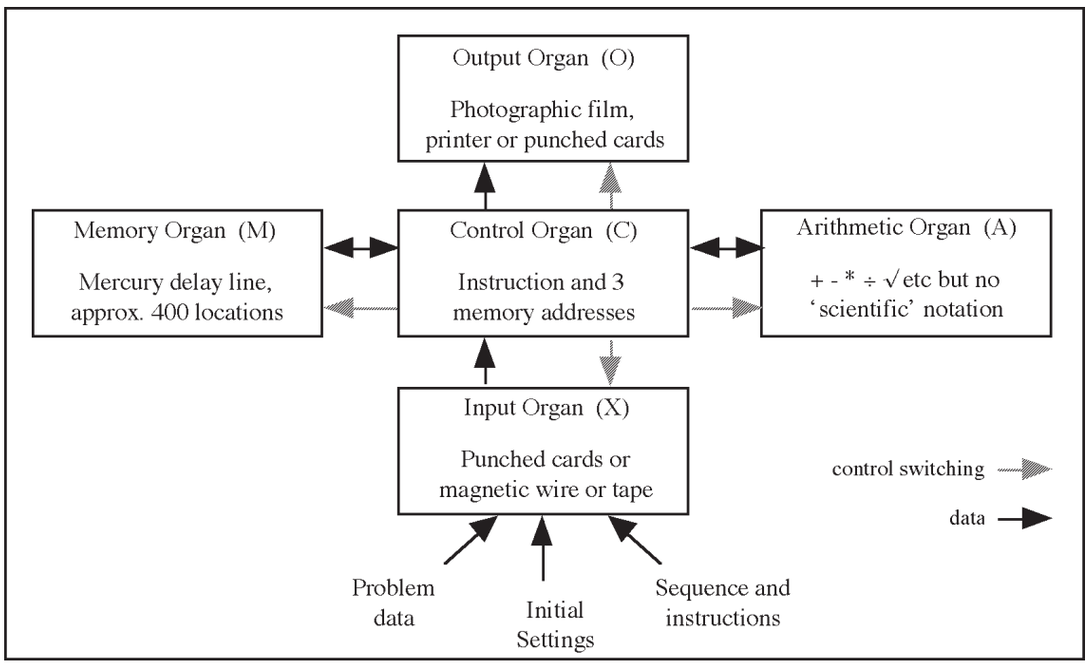


Diagram of the CSIR *Mk1 architecture as originally conceived, c. 1947. Much of this was never implemented, such as hardware for mathematical division and roots, magnetic wire input and photographic film output.*

Music and Technology in the Time of CSIRAC

It is useful to put the activities of the CSIR Mk1 into historical context. There was significant activity in electronic music before the development of the CSIR Mk1 and before the more experimental and adventurous musical developments after World War II. At about the turn of the century, Thaddeus Cahill created the first major electronic instrument, the 'Telharmonium'. His instrument, played via a keyboard, was quite large and immobile. The Telharmonium generated sine tones with rotating electric-dynamos and it distributed the output, without amplification, over telephone lines to speakers in hotels, restaurants and private homes.

During the 1920s in Moscow, Leon Theremin (who had been working as a KGB agent), developed the instrument named after him, which was played by the performer moving his hands near two antennae. The tone generation technique of the Theremin was a fairly sinusoidal valve-oscillator, which sounded something like a continuous viola, clarinet or vocal tone. It was capable of continuous pitch change because the capacitive coupling of the hands to one antenna controlled the pitch of an oscillator. The other antenna controlled volume. The Theremin was the first electronic instrument with a unique performance technique.

Throughout the 1920s and 1930s, many other electric and electronic instruments were developed such as the 'Sphärophon' (1927), the 'Dynaphone' (1927–1928) and so on. Two that were popular at the time are the 'Ondes Martenot' (1928) and the 'Trautonium' (1930). These achieved some enduring reputation because major composers composed works for them, or works that included these instruments. The Theremin is still popular, but that is possibly because of its unique performance technique. However, apart from the Theremin, these instruments were predominantly keyboard oriented. All of them were capable of monophonic melodic output, but most of them were short-lived. They were also mostly used to perform traditional music. The 'Hammond Organ' (1935) was a different and commercially more

successful development which was also used largely in a musically traditional manner.

The 'Electronic Sackbut' was developed by Hugh Le Cain in Canada, just after World War II. It had a pressure-sensitive keyboard that would allow the pitch to be varied, or smoothly slide, from one note to another, over a range of up to an octave. There was also the facility with the electronic Sackbut to add various buzzing and other noises to aid musical and timbral variety. CSIRAC was similar to most of these instruments in that it was also used in, but not restricted to, a musically traditional sense because it played melodies from the standard repertoire of instrumental music.

There were other developments at this time that used electronics, which took a fresh and less musically restrictive approach to both sounds and music itself. Percy Grainger, after writing 'Free Music' for multiple Theremins and because of his interest in free music, developed his own electronic musical instruments with the assistance of Burnett Cross. The Cross-Grainger 'Free Music Machines' (one was called the 'Kangaroo-Pouch'), developed in the early 1940s through to the end of the 1950s, typically used graphs drawn on a continuous roll of plastic or cut out of a roll of paper to control eight oscillators synchronously. These could play any pitch and any rhythm within their range. Similarly, Hugh Le Cain's 'Coded Music Apparatus' (1952), allowed the control of sound synthesis by five curves, one each for pitch and amplitude and three for timbre.

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 which happened at about the same time that CSIRAC was being planned and built and against the background of the great artistic expansion after World War II. Just before that war, in 1937, the influential American composer and musical thinker John Cage, delivered a speech which articulated the ideas and aesthetic which would shape much of music, including electronic and computer music, from that time and particularly after the end of World War II. In part, Cage prophetically said,

> 'I believe that the use of noise to make music will continue and increase until we reach a music produced through the aid of electrical instruments which will make available for musical purposes any and all sounds that can be heard. Photoelectric, film, and mechanical mediums for the synthetic production of music will be explored. Whereas, in the past, the point of disagreement has been between dissonance and consonance, it will be, in the immediate future, between noise and so-called musical sounds. The present methods of writing music, principally those which employ harmony and its reference to particular steps in the sound field, will be inadequate for the composer, who will be faced with the entire field of sound.'

Against this background and with the spirit of freedom, reconstruction and liberation artists felt after World War II, electronic music blossomed. In the field of electronic music, there were two significant emerging developments. 'Musique concrète' was establishing itself in the late 1940s through the activities of Pierre Schaeffer and Pierre Henry. French Radio (*Radiodiffusion Télévision Française*, RTF) set up the first electronic music studio in 1950 for use by Schaeffer and Henry. Musique concrète uses microphone-recorded sounds, compared to the synthetically generated sounds of pure electronic music. These everyday sounds, such as voices, noises, instruments and the like, are often used with modifications, including being played backwards, at different speeds, or with filtering. A montage of the sounds is recorded to tape for performance. This can be seen as a pre-cursor of modern day musical 'sampling'. Initially, the technology used to record and manipulate the sounds was direct disc-cutting lathes. Tape recorders became more readily available in the late 1940s and musique concrète took great advantage of the flexible editing and manipulation that they made possible. Other composers, such as Darius Milhaud, Paul Hindemith, Percy Grainger and Ernst Toch had also used manipulation of recorded sounds in compositions and in concerts as early as the 1920s. They had only variable-speed phonographs to work with. The technology of the day, such as sound recording equipment, made the techniques of musique concrète and these artistic developments possible.

Analog electronic music was the other area of intense interest where, as distinct from musique concrète, sounds are generated purely by electronic means. Using oscillators, filters and electronic modification of the developed sounds, music is created and recorded to audio tape, often using montage techniques. Sine wave generators were the initially favoured source of signals and one frequency would be recorded to each track of a multi-track tape recorder. This was then combined through a mixer onto a single track of another tape recorder. Other effects and modifications, such as filtering, reverberation and so on could be added later. Sounds created in this way were then combined with editing and mixing techniques to create a musical work. Some of the main composers who pioneered this work are Gottfried Michael Koenig, Karlheinze Stockhausen, Karel Goeyvaerts, and Herbert Eimert. The Cologne electronic music studio was in development at Westdeutcher Rundfunk (WDR) circa 1950, and similar experiments were taking place in most technologically advanced countries. Such a musical development was similarly dependent on the available technology of electronic oscillators, filters and tape recorders.

Musical sequence playing, which is effectively what CSIRAC did, has a history that dates back to the 1200s with mechanical carillons. This continued through other instruments such as the mechanical organ (for which Mozart wrote several pieces), clockwork flutes, elaborate music boxes and even some instruments, for example the 'Panharmonicon', to reproduce an entire orchestra. Late in the 19th century, perforated paper rolls were used to control various instruments, including the player piano, to very good effect.

A synthesiser which had some similarities to CSIRAC was the RCA 'Electronic Music Synthesiser' of the mid 1950s, which was controlled with two punched paper tapes. It was reported that the information was encoded by a fiendish combination of 4-bit binary switches. The RCA synthesiser was room-sized, had 750 valves, could play four-part polyphony and had controls for portamento and tremolo.

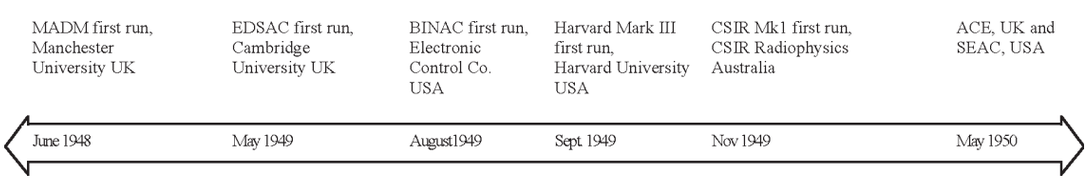
The punched paper tape control allowed for the precise control and variation of not only pitch and time, but also other parameters such as timbre.

In addition, as an artistic musical development at the time CSIRAC was being designed and built, John Cage, Pierre Boulez and others were writing advanced instrumental music, developing new composition theories and certainly becoming interested in electronic music. It is against this background, but in isolation, that CSIRAC first played music. While the musical output of CSIRAC was unimaginative compared to many of the musical developments emerging during its early years, there was considerable imagination required to use a general computing machine to play music and there was a great deal of ingenuity required in devising the techniques and programs to play it.

Computer music developed quickly in other areas where composers were involved from the early developments. Interestingly, these developments took place at about the same time as Thomas Cherry developed his Parameterised Music Programme for CSIRAC. Computer music had two main streams of activity: computer-assisted composition and sound synthesis. Computer-assisted composition, where the computer produces (data for) a score, developed from the early work of Lejaren Hiller and Leonard Isaacson which started late in 1955 at the University of Illinois. This resulted in the landmark piece 'Illiac Suite for String Quartet' in 1957. Other composers quickly followed this work with developments of their own. In Paris from about 1957, Iannis Xenakis developed music programs that modelled his compositional processes, which used statistics and probability theory to choose musical parameters. Similar work was undertaken nearby a couple of years later by Pierre Barbaud and Roger Blanchard at Compagnie des Machines Bull, a French computer company. 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 Matthews at Bell Laboratories was developing his Music 1 program, that produced completed music as a digital audio file with a triangle waveform. This was played back with a digital-to-analog converter. By 1959, and with the help of composers, this work had evolved to allow timbral variation and modification of the sounds through sound synthesis.

From a computing point of view, 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? Charles Babbage created some early mechanical calculators. Konrad Zuse created some early electromechanical calculators. His Z3 of 1941, because it used relays, has been called the first electromechanical programable calculator and digital computer. Alan Turing's 'Colossus' was operational in 1943 and it 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 programable control. Wallace Eckert's SSEC, in 1948, was a configurable calculator which could execute programed instructions. This list could continue and fill several volumes. (There are excellent internet references for those interested.) However, if one accepts the general definition of a digital computer as an all-electronic device capable of calculating and branching operations, where the data and instructions are held in some sort of rewritable memory, then the following series of events is an approximate guide to when the first all-electronic digital computers ran their first test programs.



Time line of computing events.

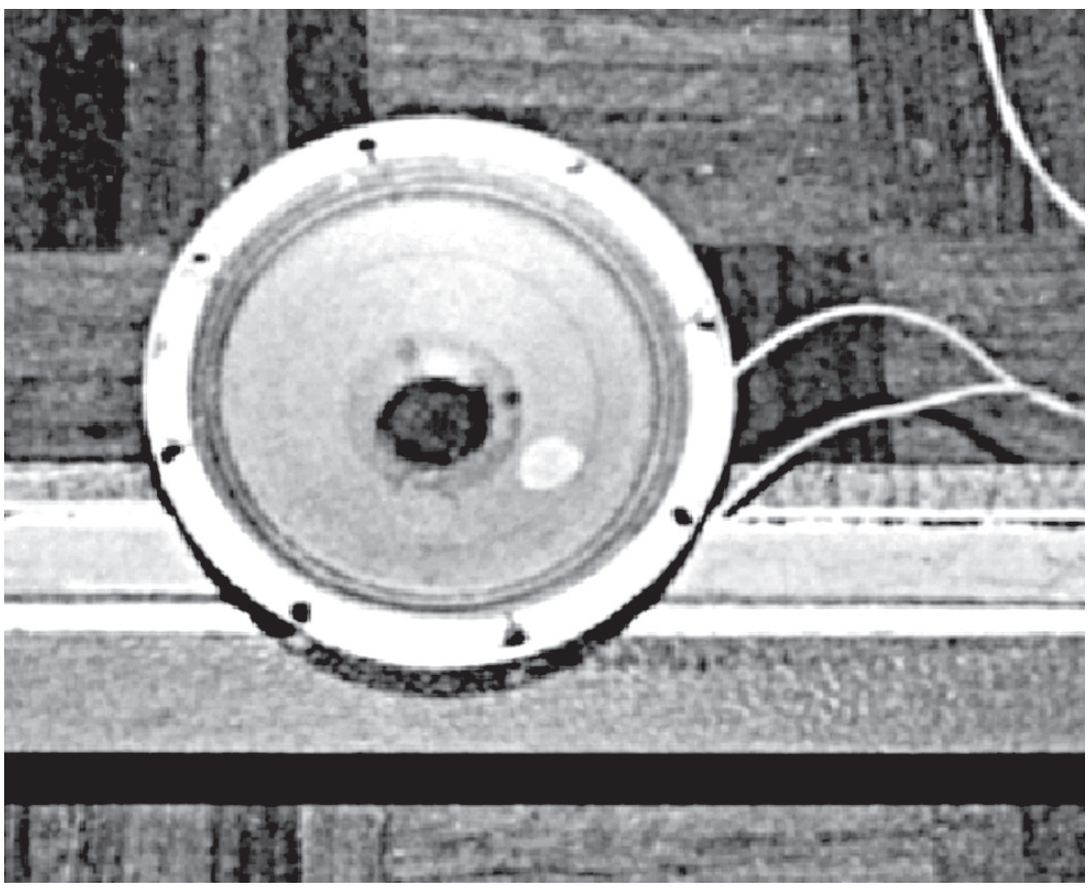
The Speaker

In common with several other first generation computers, the CSIR Mk1 had a built-in speaker. The speaker, or 'hooter' as it was known, 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. This was not unique to CSIRAC as several computers of that era, for example the Manchester Mark I and Cambridge University's EDSAC, had speakers that were used in this way. With many of the earliest computers, because of the lack of visual feedback (there was no display as is normal 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. This sounded something like a rougher or more distorted version of the 'beep' which is still common on computers today. With early computers, it was also common to keep a radio handy, because if the machine stopped working the operator could tell by the lack of interference on the radio. The hooter was a 'destination' for data within the machine and it was effectively treated as a register. The design of the CSIR Mk1 had several 'sources' and 'destinations' for data under programatic control, as shown in the original CSIR Mk1 architecture diagram in 'An Overview of CSIRAC'.

The 1948 design of the CSIR Mk1 makes no mention of a speaker or hooter. Careful checking of the original design documents show that the register for the speaker (destination 10 or symbol P) is not in the original list of destinations. It is recorded that Trevor Pearcey travelled to England in November and December of 1948 after completing the design. Whilst in England he saw other computers which certainly had hooters, for example the Manchester Mark I and EDSAC. However, some other first generation computers apparently did not have programmable frequency hooters, it was a register that triggered a fixed frequency oscillator connected to a speaker. The frequency of the oscillator was not under programatic

control. Documentation for the CSIR Mk1 speaker appears in papers from about 1950. It is possible that Trevor Pearcey added the speaker to the CSIR Mk1 after seeing them on the early computers in England, but this probably would have been some time in 1949, so it is still considered part of the original machine.

It is difficult to know now why the CSIR Mk1 was given a hooter that could produce a programmable frequency. It is unclear if the hooter on the Manchester Mark I (MADM) could produce a variable frequency. Certainly the hooter on the 1951 Ferranti Mark I, the production version of the MADM, was similar to that on the CSIR Mk1, but Pearcey saw the MADM, or its predecessor the SSEM, late in 1948 so he did not see the Ferranti machine. With a desire to get the machine finished, it is possible that it was decided to give the CSIR Mk1 a simple, but ultimately programmable-frequency, hooter. Alternatively, it could have been because of the design philosophy adopted, to create a simple, elegant and very flexible machine. This explanation probably best fits the evidence and personnel involved.



The original speaker on the console frame.

The speaker 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'. This would work to create some sort of sound, but it would require a little more effort on the part of the program to get a sound out of it as a single pulse would barely make a 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 of instructions 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 x
        loop start, send pulse to speaker
                wait time t1
                send pulse to speaker
                wait time t1
                send pulse to speaker
                wait time t2
                send pulse to speaker
                wait time t3
                send pulse to speaker
                decrement x
        if x equal to zero then exit, else return to loop start
end of note
```

Although the musical developments with the CSIR Mk1 were accomplished in isolation and with no precedence, it was not unique at that time. The Ferranti Mark I had a hooter arrangement very similar to that on the CSIR Mk1 and it was used to play music in 1951. The music played by the Ferranti Mark I was recorded by the BBC in an Outside Broadcast van for a children's radio show in September 1951. In 1951, Frank Cooper was an engineer working on the Ferranti Mark I installed in Manchester University. Christopher Strachey, a mathematician, wrote the program to play music under the tutelage of Alan Turing, who had supplied Strachey with a programming manual. The first music played by the Ferranti Mark I was the English national anthem ('God Save the King'),

and the programming techniques used were apparently similar to that described above which were used on the CSIR Mk1 in Sydney and Melbourne. News of this development spread. The recording crew of the BBC Children's Hour heard about the Manchester computer playing music and arrived to record it on 7 September 1951. Frank Cooper operated the computer for that recording and the BBC recorded several pieces of music. Similarly with the CSIR Mk1, the music was created by engineers and it took the form of popular melodies of the time. The original recording by the BBC was directly to acetate disk. Frank Cooper asked if he could have a copy of the recording, but the BBC would not provide a copy for copyright reasons. However, the BBC recording crew very kindly re-ran their recording apparatus with a scrap disk for Mr Cooper on the spot and presented him with it as a gift. The computer was not as reliable as would have been desirable and the re-run of the music program often stopped because of machine problems. The BBC disk has very unfortunately been lost. Luckily, the (British) Computer Conservation Society is now the custodian of the disk originally owned by Frank Cooper and it has been transcribed to DAT by the National Sound Archive (archive number H3942), so that this early example of a computer playing music, unlike with the CSIR Mk1, was not lost.

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, as searches of the archives of the CSIR and the individuals involved at the time have not yet turned up any written documents. This 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 the CSIR Mk1 and who has interviewed most of the original personnel involved with the project. He recalls hearing about the music:

> 'I was doing research on a topic in the history of Australian science when Peter Thorne asked me to become involved in a project on CSIR. As an historian of the natural sciences, I didn't know if I would be able to do the topic justice without having a background in the technical side of computing. I was intrigued though, because even though I had worked for a couple of years at the Museum of Victoria where CSIRAC is housed, I had never even heard of it. When I found out about it from Peter Thorne, I was really astounded. Why was such an important item so little publicly known?
>
> I first became aware of the possibility that CSIRAC played music when I saw it for the first time during a visit on 11 April 1996 to a Museum store in Maribyrnong. I was with others, including Edwin Parsons, a collector of early computer hardware and software. Ed was well versed in computer history and immediately noticed the speaker on the computer console. I could see he was rather surprised by it and he started asking questions about it. He informed me of the early computer music developments in the United States. According to answers he elicited from George (Jurij) Semkiw, it appeared likely that CSIRAC played the music much earlier than this, but he was unable to give us definite dates. I remember Ed saying 'We must talk to Trevor Pearcey about this.' I had already made preliminary contact with Pearcey so together Ed and I went to have an informal interview with

> him. I still have notes from that interview where the music is mentioned; the date of the interview is 19 April 1996.
>
> Ed was particularly interested in ascertaining if and when CSIRAC had played music. We both questioned 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 August1951 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. We interviewed Pearcey several times, once on video and he was consistent and very definite about the fact that CSIRAC played music in August 1951 at the latest, and probably somewhat earlier than that.

A fragment of the videotaped interview with Pearcey, which occurred several years after the one just recalled by Doug McCann, is transcribed below. In a later section, there is a more extensive transcription. It confirms McCann's recollection of what Pearcey could remember about the music and also adds some more detail. From the videotaped interview with Pearcey in October 1996, Trevor Pearcey said:

> 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 who was then on 2BL (radio station) 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 realise 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.

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 who was 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.

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 in 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.

Geoff Hill first programmed the CSIR Mk1 to play music. Mrs Hill, the wife of Geoff Hill, can remember hearing about the music in 1951 before she first met Geoff in 1952:

> It was while I was a resident at Sancta Sophia (student) College at Sydney University that I first heard about the

> computer playing music. I didn't know Geoff at the time, but Mary Thurling, also a resident at the college knew him. She and Geoff were good friends and university colleagues. Other students staying at Sancta Sophia also knew Geoff. The Division of Radiophysics where Geoff worked was in the university grounds and since at the same time he was in the process of writing his MSc thesis (Programming for High Speed Computers) he was very much involved in university life. Mary stated one night at the dinner table, 'Did you know that Geoff Hill has that machine playing music!' That was my first knowledge of it. I remember remarking at the time, 'Who's Geoff Hill?' So it must have been in 1951 because by 1952 I knew him. I did not meet Geoff until 1952 when we were both on the committee of the International Club at the university.

Mrs Hill still has Geoff Hill's Master of Science thesis from The University of Sydney, dated March 1954. On page 63 of the thesis the music is mentioned:

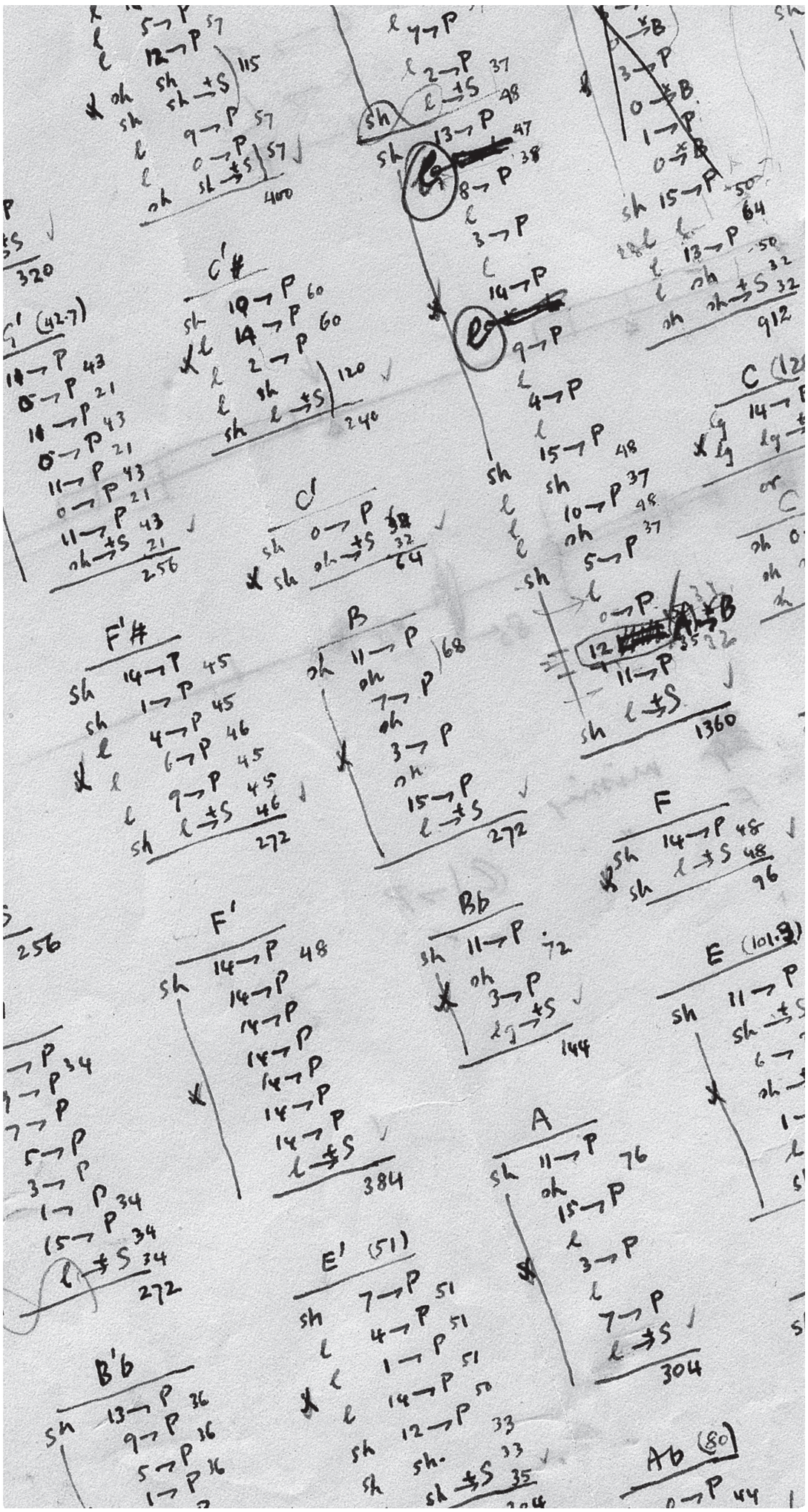
> Extension to semantic analysis of language has led to design of 'language-translation' programmes. Sub-routines generating notes of the chromatic scale by sending pulses to a loud speaker can be used in an interpretive programme for playing music from a coded score … the techniques of music and 'translation' programmes were adapted as the basis of the powerful interpretive approach to complicated arithmetic.

In addition to this, Ron Bowles has examined the Sydney music punched paper tape and can establish the date for it as the first half of 1953. This is mostly because of the particular type of tape reader needed for this tape. The input and output devices on the computer changed several times and these included punched cards, 12 hole tape with edge sprockets, 12 hole tape without edge sprockets that self synchronised and 5 hole tape. 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 to date 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, it would seem that the earliest items on this tape would have existed a significant amount of time before 1953.

Taking into account the corroborating evidence of Mrs Hill and Geoff Hill's 1954 thesis, the date of the surviving punched paper tape of the music from Sydney, the correlation of the recollections and also that most of these people have often not communicated at all since 1955, it is reasonable to accept the recollections of Pearcey, Ryan and McGee that the CSIR Mk1 did publicly play music in the first week of August in 1951. Similarly with the music played on the Ferranti Mark I a few weeks later, the computer 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. Cherry generalised 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 can remember it at about that time and as they say, it could have been late in 1956 that Cherry first had recognisable music coming from the machine, but it was most likely in 1957. The written records have only a little information that can date the work. 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 Cherry used the Melbourne 12 hole paper tape exclusively, as CSIRAC had 5 hole equipment attached at about the middle of its service in Melbourne. This points to early work on the computer in Melbourne, as does Cherry's activities because he was programming only very early in the Melbourne period, after which his duties were in administration.



The Music

The CSIR Mk1 had a hooter circuit that, unlike some other computers of the day, could be programmed to produce a variable frequency by sending pulses to the speaker at varying rates. Because the speaker was sent raw pulses through an amplifier, from the data bus, several pulses would need to be sent either as multiple in line (P) statements or in a loop. It is 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 very musical family, both his mother and sister taught music throughout their lives. 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. It was played publicly for the first public exhibition of the computer between 7–9 August in 1951, at the inaugural Conference of Automatic Computing Machines in Sydney. Reg Ryan, who joined the computing team in 1948 and the engineer responsible for the memory system on the CSIR Mk1, can remember the music from his earliest days:

> I always thought that the flashing lights on the console, when debugging the machine or running a program, may have given Geoff the idea to make music with it. The lights always reminded me of music when they were going. 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.

The first public demonstration of the CSIR Mk1 took place during the first Conference on Automatic Computing Machines. There was something of a political tussle at the time between the analogue-computing protagonists and the digital-computing protagonists. Trevor Pearcey quite probably knew that he would need to attract some positive attention to the CSIR Mk1. Geoff Hill, possibly in collaboration with

Pearcey, created a program to play 'Colonel Bogey', which was used at that demonstration according to the recollections of Pearcey. However, the digital world must have lost on that day because newspaper reports of the conference mostly mention the analogue-computing machine, a differential analyser and barely acknowledge the existence of the CSIR Mk1. Independent eyewitness accounts, however, confirm that this event did take place, as described previously and below.

Trevor Pearcey recalled the music of CSIRAC in an interview in October 1996:

> I believe CSIRAC was a very early machine to provide tones through a loudspeaker. The loudspeaker, of course, was initially installed so that with a particular instruction (as the loudspeaker was a destination) coded into a program, a pulse could be sent to the loudspeaker when that instruction was executed and thereby provide the user with an indication as to where the program was at any particular moment by providing him with appropriate clicks. The user could more or less follow the running of a program so that when the program went wild, because a spurious electronic fault occurred, he would then be warned by a change in the rhythm of the loudspeaker noise. From there of course it was not a very great extension to program regular tones and we managed to provide a program which would pass impulses to the loudspeaker in a diatonic scale and thereafter to extend the program to controlled execution of tones and the playing of what might be called music. This was of course a purely experimental program to see whether this could be done. This was in fact done successfully over a fairly small range of a couple of octaves of a diatonic scale and the first tunes were played after a few faulty passes and mistakes. We played 'Girl with Flaxen Hair', 'Colonel Bogey' and one or two other things like that. I remember calling the Chief of the division at that stage to come and listen to it and I had suggested that we record the tunes and get Frank Legg who was then on 2BL (radio station) 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 realise 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. The loudspeaker was mounted on one of the cabinets. Of course at that stage all the cabinets were open and the doors had not been put on.

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 speaker attached. However, it's also worth remembering that this occurred when there was no such thing as digital-to-analog converters, there was no digital audio practice and little in the way of complete digital audio theories at this time. In addition, the CSIR Mk1 produced music in real-time. The limitation and lack of mass storage also forced this real-time sound production approach, as there was nothing such as magnetic computer tape to store digitised audio and no digital-to-analog converters for playback. There were enormous timing subtleties of the computer and sound generation process to be understood and accommodated 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 endeavour, 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 Matthews at Bell Telephone Laboratories. John Pierce, who at the time was executive director of the Communication Sciences division at Bell Labs, says in the booklet accompanying 'The Historical CD of Sound Synthesis' that there were computers making music with 'buzzes and squawks' before the work of Max Matthews. Although it was little reported at the time, the CSIR Mk1 was one of these.

It is now possible to hear again this particular, tentative, beginning of the application of computing technology to music. The musical pieces played by the CSIR Mk1 are not as musically inspiring as they might have been if composers had been involved in creating the music. The computer had a lot to offer composers of the time, any frequency and any rhythm could have been programmed so the many composers interested in microtonal works, or music with no rhythm or very complex rhythms, could have created some very interesting music. As it stands, the music is most interesting from the point of view of the application of computers to music as a general principle and as an example of the early practice of computer programming to create music. The achievement is significant because of the imagination of the practitioners, to conceive of using the flexibility of a digital computer to make music and because of the ingenuity required of the programmers to devise means to produce reliable sounds from the computer. It is difficult to appreciate now just how skilful these people were. It is significant that it was only two of the best programmers who managed to program the CSIR Mk1 to play music. These programmers were the ones who wrote the software library routines, the boot or primary sequences and so on. The programming techniques, timing, and machine issues involved in programming the CSIR Mk1 to play music have been discussed in general and are discussed later in more detail. Overcoming these issues, when there was no prior practice to work from, is what shows the ingenuity and skill needed to complete the task, and demonstrates the significance of the programming achievement. Several people have commented that when they first heard the music, in the 1950s it was perceived as something magical, it was astonishing.

Thomas Cherry, in Melbourne, also programmed CSIRAC to play music and he also wrote some of the main arithmetic routines in the software library such as the division and trigonometric routines. 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 allowed the programming of some notes

to extend the range. Cherry also designed the structure of the music program and generalised it such that it could accept a data tape of note pitch and duration data so that others could create data tapes of music to be played.


20% 10% 5% 0 5% 10% 20%
– VOLTS +
210V. REG.
250
300
100 200 300 400 500
VOLTS
DC

Sydney Music, 1949–1955

Because the CSIR Mk1 had a career split between two cities with largely different personnel, it is convenient to separate the musical activities along the same division. The CSIR Mk1 operated in Sydney, Australia, from about November 1949 to June 1955. Geoff Hill was the main programmer at that time and he used the machine to play musical melodies. These melodies, mostly from popular songs, were: 'Colonel Bogey', 'Bonnie Banks', 'Girl with Flaxen Hair', 'Auld Lang Syne', a Chopin March, 'Thanks for the Memory', the 'Saul' aria from Handel's 'Messiah', some scales and a humorous variation on 'Saul', titled 'Saul with Blurts'. There are a few reported pieces that have not survived, such as 'Greensleeves'. The music reconstructed from this time comes from a punched paper program tape that dates from early 1953. This tape appears to have been a work-in-progress as there are a few corrections and notes on the paper tape, but then perhaps it was possible to endlessly 'tweak' these music programs. This program tape contained a selection of melodies that could be played, apparently as a collection of pieces programmed up until that time. It was labelled simply 'Music Tape', suggesting that there was probably only one, although experiments to program other pieces would possibly be on other program tapes. The 'Music Tape' was a collection of pieces for performance. The operator could select which piece was played by setting switches on the console.

All of these pieces use loops of pulses sent to the speaker to create the pitches. The dynamics or volume of each note could be varied by the number of pulses or 'on-bits' in the data word that was sent to the speaker, from a maximum of five bits to a minimum of one bit, zero bits on being a rest. Each word sent to the speaker became a single pulse repeated at audio frequencies, so the number of 'on-bits' in the word changed the amplitude of the resultant pulse to the speaker and thus the volume. This sound generation technique circumvented the need for a digital-to-analog converter, which was useful because there were not any at

this time. The pulse shape is fundamentally a 'sawtooth' triangular wave, which is now a familiar waveform in sound synthesis. Although the timbre or waveform of the note could not be varied, the pitch, duration and volume of a note were under programatic control. This is not dissimilar to Max Matthews' Music I program which used an equilateral triangle waveform. The range of notes available in Sydney was almost two octaves.

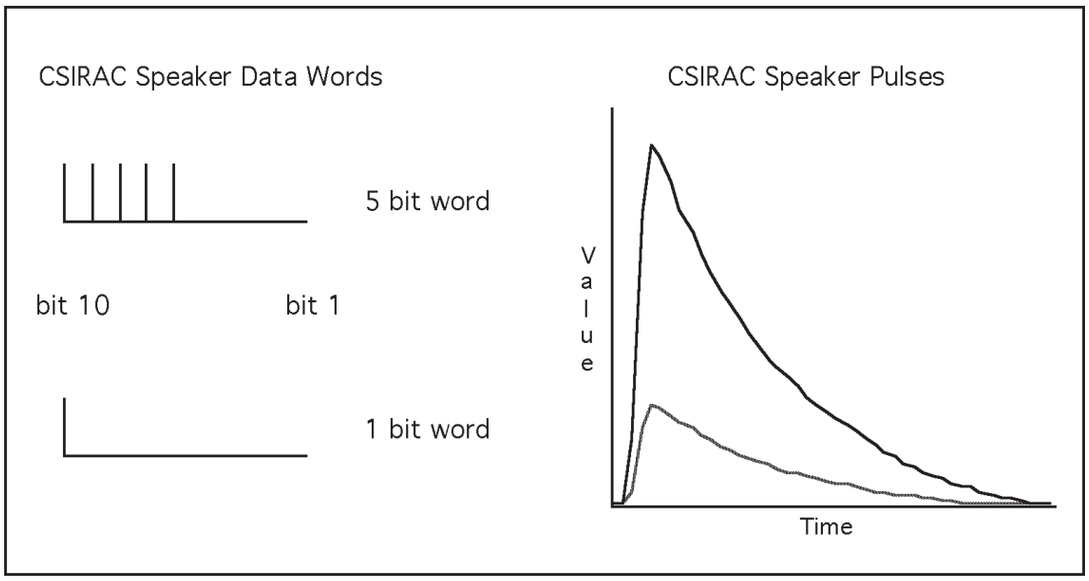


Diagram of CSIR Mk1 speaker-pulse amplitude variation.

It is speculation to ask why these particular melodies were chosen, but anecdotal evidence suggests that 'Colonel Bogey' was chosen because it was such a popular favourite and Trevor Pearcey was aware of the need to make a positive impression at the public unveiling of the CSIR Mk1 in 1951. As Reg Ryan recalls:

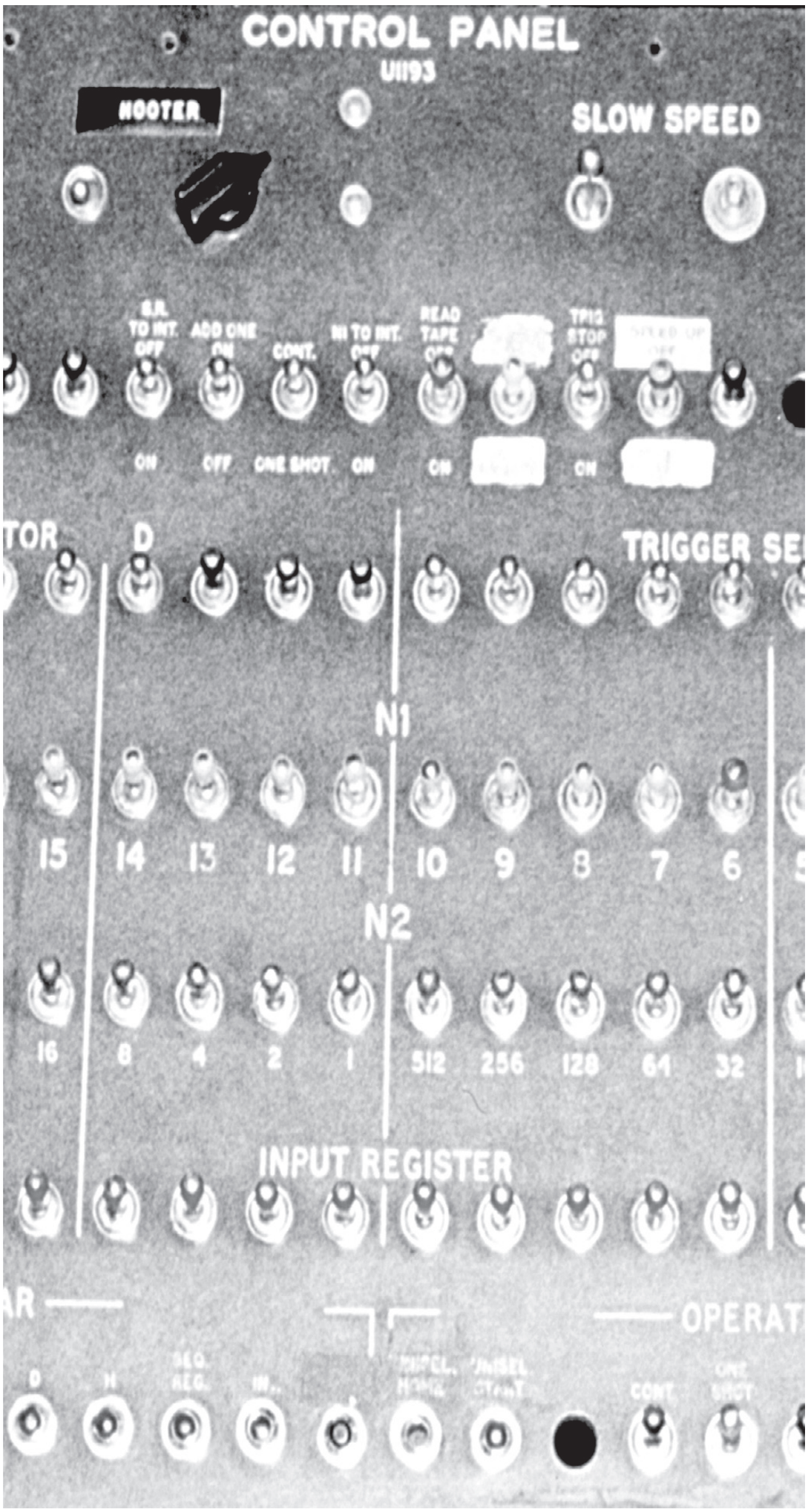
> Everybody remembered hearing 'Colonel Bogey'. It was popular and it had people intrigued that the computer could play music, it was something they could grab hold of.

'Bonnie Banks' was apparently programmed for the then Prime Minister of Australia, Sir Robert Menzies, on his request to hear something Scottish, according to the recollections of Geoff Hill's family. Pearcey wanted the Prime Minister to be interested in the computing project and support it. The upgrading of the CSIR Mk1's memory was supposedly

commemorated with 'Thanks for the Memory' and 'Auld Lang Syne' was eventually used for the decommissioning in Sydney but it was programmed well before that.

There are many stories surrounding this work. Mrs. Eileen Hill remembers, recalling the music and stories surrounding it after she first heard about it in 1951:

> … Some time later, Mrs. Hill (Geoff's mother) told me that Geoff was apparently working on the song 'Greensleeves' late one night very early on in his musical experiments with the computer, probably because it had mostly small steps between successive notes and it should have been easy. Geoff had perfect pitch, but he was not sure of the accuracy of one of the notes. To get a second opinion, he telephoned his mother to have a listen. She was a bit annoyed about being called so late and Geoff said with no preliminaries 'Can you please have a listen to this and check the pitch of one of the notes?' while he held the telephone handset to the speaker for her to hear the piece. When the piece had finished and he asked if it had any wrong notes she replied, 'That sounds like tissue paper being played through a comb. It's a bit late to be fooling around–what's going on there?' 'No, it's the computer.' He told her. Geoff told me later that the piece of music he played on that occasion was 'Greensleeves' and I believe that it may well have been the first music to be programmed for the machine.


CONTROL PANEL
U1193
HOOTER
SLOW SPEED
S.R. TO INT. OFF
ADD ONE ON
CONT.
N1 TO INT. OFF
READ TAPE OFF
TP13 STOP OFF
ON
OFF
ONE SHOT.
ON
ON
ON
TRIGGER SE
N1
15
14
13
12
11
10
9
8
7
6
N2
16
8
4
2
1
512
256
128
64
32
INPUT REGISTER
OPERAT
CONT.
ONE SHOT

Melbourne Music, 1955–1964

The CSIR Mk1 was dismantled mid-1955, placed under a tarpaulin on a large articulated truck and driven 1000km to its new home, the University of Melbourne Physics Department. There it was under the direction of Frank Hirst and it was renamed CSIRAC. Geoff Hill did spend quite some time in Melbourne with CSIRAC, but apparently he did not add to the musical repertoire of the machine during this period. However, there was a Professor of Mathematics named Thomas Cherry, later Sir Thomas Cherry FRS, who had a great interest in programming and music and he created music with CSIRAC. In Melbourne the practice of how CSIRAC was programmed for music was altered and refined somewhat. For example, to increase the dynamic range of the tones, all of the bits of the data word sent to the speaker were used, instead of only five as in the Sydney music. The data word sent to the speaker could vary in range from having all 12 bits on to only one bit on, with no bits on being a rest. The pitch range was also extended, although the accuracy of the highest notes left a lot to be desired.

Cherry programmed several pieces for CSIRAC. The Melbourne University song 'Gaudeamus Igitur' was apparently programmed and performed in 1956, possibly for the official university opening of CSIRAC. This program tape no longer exists and it is more likely that it was programmed by Geoff Hill. Although Cherry had mentioned during the opening ceremony that CSIRAC could play music, he would probably have been too unfamiliar with CSIRAC at that early stage to program it for that. Kay Thorne agrees:

> Yes, Geoff probably did program 'Gaudeamus'. It was just like his sense of humour to do that, with it going from the CSIRO to the university. That would be very 'Geoff' to do that, he had quite a sense of humour.

A newspaper report of the occasion vividly described it as sounding, 'Like a refrigerator defrosting in tune.'

The program tapes for a couple of test scales still exist, along with the popular melodies 'So early in the Morning' and 'In Cellar Cool', which was a popular drinking song–it appears that the pursuit of computer music and social drinking have been intimately linked since the earliest years.

Cherry also programmed a piece of music, a theme and variations for bassoon, for CSIRAC to perform. This must have been by 1958 at the latest, or possibly earlier but unfortunately there is no firm date. The notation used by Cherry, for example using the ' P' symbol, dates it to 1958 or earlier as this notation was no longer used by the start of 1959. The theme and variations, called 'Lucy Long', was chosen by Cherry because apparently the timbre and range of CSIRAC reminded him of the bassoon. The significant point of this work is that Cherry had generalised the requirements of playing music with CSIRAC and had written a program for it with 'Lucy Long' as the data for that program.

The generalisation of playing music with the computer was probably the most significant contribution by Cherry to computer music practice with CSIRAC. In about 1957 Cherry 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 having to negotiate all of the usual timing problems. The 'Music Programme' could play music from memory or from punched paper tape. Each had advantages and disadvantages. Playing from tape allowed the performances 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 because of the limited size of the memory the piece could not be large.

CSIRAC had a musical range of about two and a half octaves as programmed by Cherry and each 'note' was assigned a number. Similarly, each duration was assigned a number. Hence, there was a numerical description of the music to be played. The procedure to have CSIRAC play a piece of music was for a user to punch a paper tape according to a set of

instructions, reading the music notation and entering the correct numbers for the pitch, duration and dynamic (volume or intensity) for each note after looking them up in a table. An intensity number was punched (denoted by a Y punch on the paper tape), then note and duration numbers. The intensity would apply to all note pitch-duration numbers that came after it. This was appropriately called a 'pianola tape'. When the appropriate Music Programme, 'play from tape' or 'play from store', was loaded into the computer and executed, it would stop, waiting for the data tape. The pianola tape was loaded and stepped into position. If it was to play from tape, execution could begin and the piece heard. If it was to play from store, the tape would be read into memory and the program would again stop. Then execution could begin to play the music data from the mercury delay-line memory.

This work by Cherry bears some comparison with other computer music research of the time. The Music Programme, which allowed for a high-level (numerical) description of the music, has several similarities to Max Matthews' work from 1957, called Music I. Matthews was working at Bell Labs then, when he started his experiments of applying computers to musical goals. There is no variable synthesis component in either program. 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 achieve a synthesis component (Music II, 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 soundfiles with Music I and Music II was the result of several technological advances that CSIRAC, being a previous generation of computing technology, did not have access to. To generate, store and play back soundfiles requires: online mass storage that can store soundfiles such as magnetic tape; digital-to-analog converters to convert the soundfiles to audio; significant electronic memory capacity; and, a high enough bandwidth digital-bus

to transfer the data to the digital-to-analog converters. The high-level, numerical specification for musical parameters was important for both Matthews' Music I and Cherry's Music Programme. It was also probably a natural step from other music technology such as punched paper tape, or roll, for control of player pianos, however it implies a degree of sophistication in the software to generalise the input data. The use of digital sound files was significant for Music I and later developments because it allowed for arbitrary timbral variation, of which CSIRAC was not capable. 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 because of the limits of computing power. 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 significant innovations 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. The unit generator concept is similar to the concept of building a sound in an analog electronic studio or synthesiser, where different modules or universal building blocks, for example oscillators, filters and envelope generators, are connected together to create a sound. With a Music III unit generator this is accomplished with a file of text instructions to the program. The stored function oscillator directly synthesises a waveform by sampling a stored function that represents a single cycle of the sound and it is similarly controlled by parameters in a text file. These concepts, which were developed largely in response to the requirements of composers working with the tools, permitted the flexibility that composers wanted to generate new musical structures and they continue to influence computer music today.

Both Cherry's Music Programme and Matthews' Music I allowed for a similar high-level and numerical specification of musical events. However, what Cherry's program 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 speed variation of the piece.

As with the CSIR Mk1 playing music in Sydney, there are several people who remember events surrounding CSIRAC playing music in Melbourne. Terry Holden, an early user of CSIRAC in Melbourne remembers Cherry's music program:

> The most successful music program was written by Professor Tom Cherry. You've got to remember that this machine was fairly slow, with a millisecond sort of repetition rate. Tom Cherry wrote a music program that he went to a lot of trouble to make quite sophisticated so that you could change (transpose) the pitch and so on. However, the sort of noise that he managed to get out of it reminded him of a bassoon. It was a fairly low pitch. So he programmed a bassoon solo called 'Lucy Long', which I remember him playing. It was a mournful sort of dirge. It was good fun really. I think his music program was probably written in about 1957 or '58, it was quite early, it might have been earlier. I actually found a copy of the instructions for the music program and found such gems as, 'If transposition beyond these limits is attempted, the computer will be asked for notes that are not in its repertoire and it will go berserk.' Which is a beautiful way of putting things I think.

Kay Thorne, who worked daily with CSIRAC from March 1959, recalls several events concerning the music:

> The music was played quite a lot on university open days, but after a while we stopped doing it as much because of the difficulty of explaining it to people. If you go back to the mind set in those days, people felt that computers were going to be all-powerful things. There was no knowledge in the general public of how they worked. So there was a feeling that you could ask the computer anything. People used to ring up on the 'phone to ask the computer questions. We even had people ringing up with crossword clues to ask the computer, like 'Can you ask the computer what is a plant name with six letters that starts with A'. So with that perception, if we played the music on an open day for the general public, who mostly didn't know anything about computers, you had to

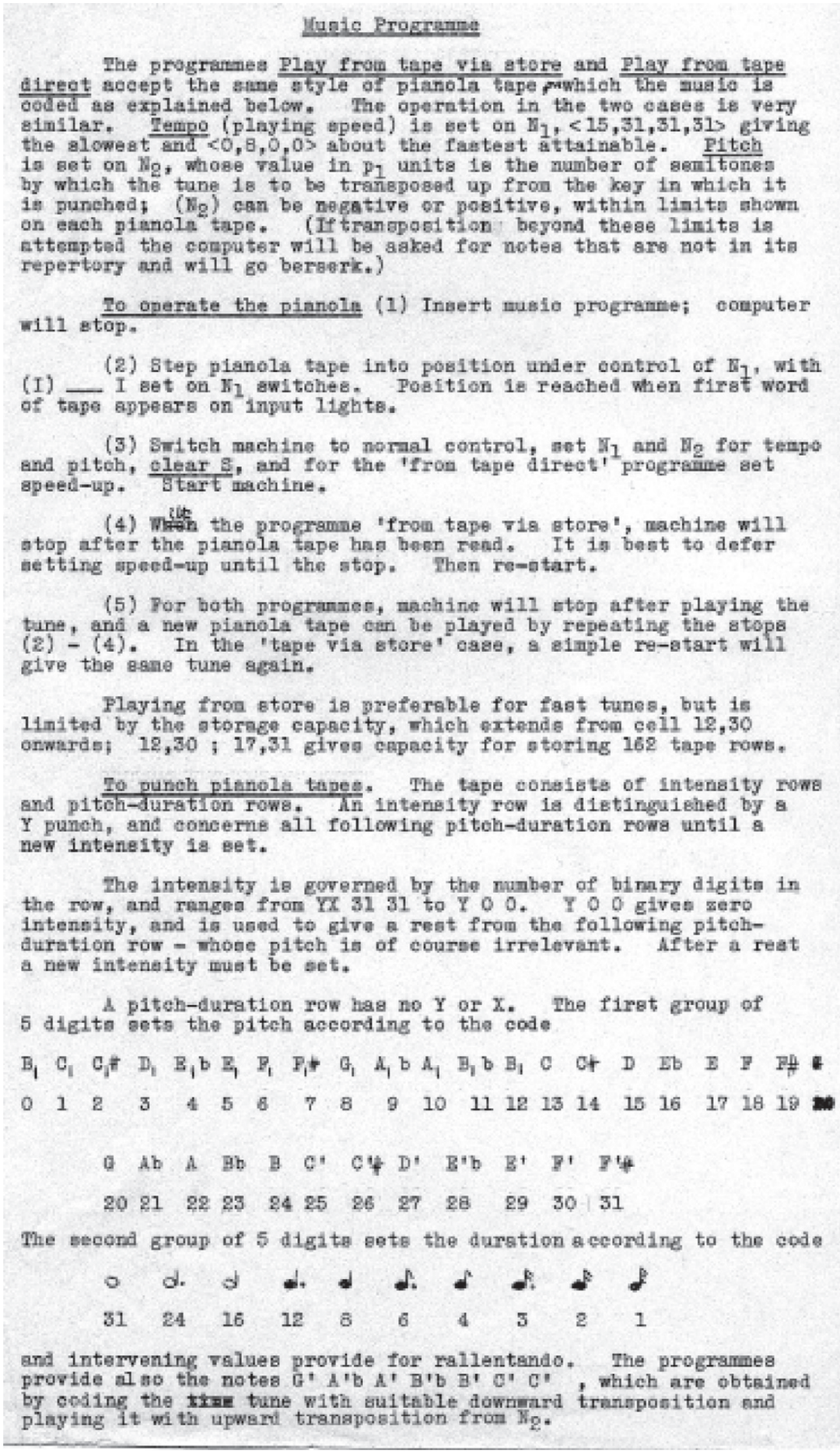
> try and explain why the computer was playing music. The skill involved in getting the computer to play music just went straight over their heads. So although we used to play it for open days and people often requested it and then we would play it, after a while we only played it for people who had some knowledge of programming and some knowledge of the machine who could appreciate what was involved. But if you worked there as I did you were familiar with that music. It was one of those things that was played, not once a year, but probably a couple of times a month. It was fairly frequent. The reason was that we were doing a tremendous amount of training of university people, industry people, even members of the public. So we had groups of visitors through the place quite frequently. Then each one of the programmers would bring along a group or friends and so on, so there were always visitors around.
>
> It's Tom Cherry who every one will remember for the music, he was the one who was most proud of it, but I have a feeling there *may* have been one or two others who also did it. I believe his music program was already functioning when I started there in March 1959. It was quite early. Tom Cherry did most of his programming in the early days because he was one of the group that ran the place. He didn't do a lot of hands on work by the stage that I was there. I think it was done in about '57 or so. The music was just part of our lives and it was just 'there', it was something we never thought about. It always had a rather nice sound I thought. It did! It was odd music but it was quite nice to have that because there was no 'piped' music or anything like that, so if the music started up every one would turn around and look and listen.

The document below contains the instructions for operating the Music Programme and punching pianola tapes, as typed by Cherry. The Music Programme did not take up much space in memory. This is important to remember for later analysis of the note frequencies, because there is not much memory space for the note pulse data. The whole music program and 'primary' (bootstrap loader), without data, occupied about 382 words in memory. The first 16 words were taken up with the 'primary', with the music program starting

at position 20. There was space in memory for another 162 tape rows. With this sort of memory restriction, it is easier to appreciate why short musical pieces were the norm and how much more of an achievement the Music Programme and 'Lucy Long' were.

Peter Thorne recalls hearing the music in Melbourne:

> I first saw CSIRAC in the late 1950s on a school visit. I next saw CSIRAC in about 1959 and I'm sure I first heard the music at a University open day. It played music every open day until it was closed down. People came in and we would play the music. It was almost invariably the long piece that Cherry programmed. It was the party piece and it was the demonstration piece, but most of the time the computer would be doing other things. So my main recollections of the music is largely 'Lucy Long', largely open days and largely people standing around with a half bemused smile on their faces. People didn't react to it that well, I don't think people were as impressed with the music as they were with the mathematical capabilities. Because, I think they looked at that and thought that a music student could play that. Whereas you could look at the mathematics being performed and realise that it was much faster than any people who could do figures and also much faster than a person with a desk calculator. It was much faster than that. So that was extraordinary, whereas to play music was not extraordinary. People of course did not realise how clever it was to play music and I actually think that at the time, if you think how many people knew how clever it was to do that, to have any understanding of the process involved in generating digital music, it was probably very few, it was limited to the people who were programmers of CSIRAC. Probably more people now realise how clever it was than was realised at the time. It's not obvious how clever it was then and people were more impressed by the calculation of loan repayments. By this time, for the music, it was perceived as doing something that a phonograph could do.



Music Programme

The programmes Play from tape via store and Play from tape direct accept the same style of pianola tape, which the music is coded as explained below. The operation in the two cases is very similar. Tempo (playing speed) is set on N_1, <15,31,31,31> giving the slowest and <0,8,0,0> about the fastest attainable. Pitch is set on N_2, whose value in p_1 units is the number of semitones by which the tune is to be transposed up from the key in which it is punched; (N_2) can be negative or positive, within limits shown on each pianola tape. (If transposition beyond these limits is attempted the computer will be asked for notes that are not in its repertory and will go berserk.)

To operate the pianola (1) Insert music programme; computer will stop.

(2) Step pianola tape into position under control of N_1, with (I) ___ I set on N_1 switches. Position is reached when first word of tape appears on input lights.

(3) Switch machine to normal control, set N_1 and N_2 for tempo and pitch, clear S, and for the 'from tape direct' programme set speed-up. Start machine.

(4) When the programme 'from tape via store', machine will stop after the pianola tape has been read. It is best to defer setting speed-up until the stop. Then re-start.

(5) For both programmes, machine will stop after playing the tune, and a new pianola tape can be played by repeating the steps (2) - (4). In the 'tape via store' case, a simple re-start will give the same tune again.

Playing from store is preferable for fast tunes, but is limited by the storage capacity, which extends from cell 12,30 onwards; 12,30 ; 17,31 gives capacity for storing 162 tape rows.

To punch pianola tapes. The tape consists of intensity rows and pitch-duration rows. An intensity row is distinguished by a Y punch, and concerns all following pitch-duration rows until a new intensity is set.

The intensity is governed by the number of binary digits in the row, and ranges from YX 31 31 to Y 0 0. Y 0 0 gives zero intensity, and is used to give a rest from the following pitch-duration row - whose pitch is of course irrelevant. After a rest a new intensity must be set.

A pitch-duration row has no Y or X. The first group of 5 digits sets the pitch according to the code

B_1	C_1	$C_1\sharp$	D_1	$E_1\flat$	E_1	F_1	$F_1\sharp$	G_1	$A_1\flat$	A_1	$B_1\flat$	B_1	C	C♯	D	Eb	E	F	F♯	G
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

G	Ab	A	Bb	B	C'	C'♯	D'	E'b	E'	F'	F'♯
20	21	22	23	24	25	26	27	28	29	30	31

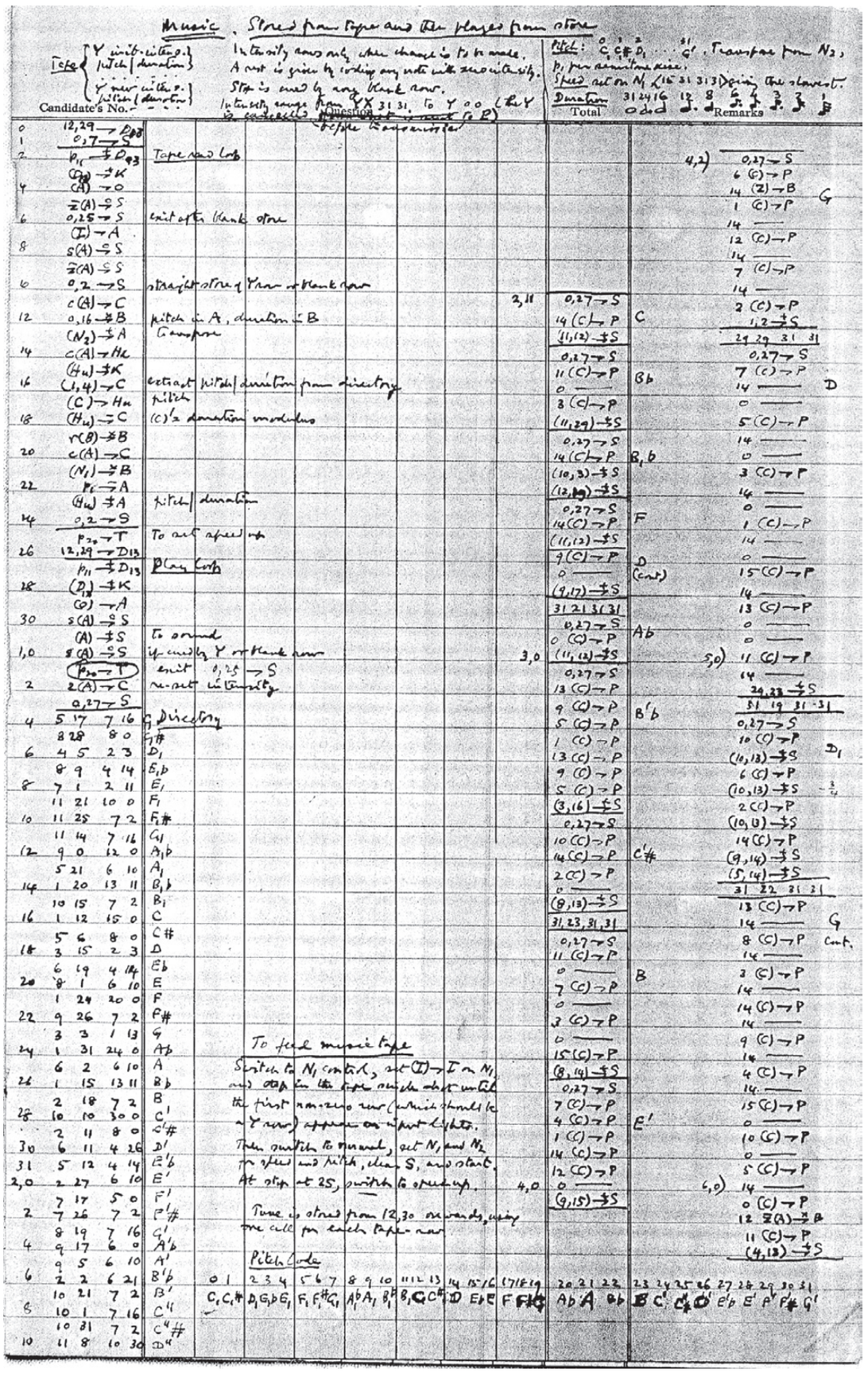
The second group of 5 digits sets the duration according to the code

𝅝	𝅗𝅥.	𝅗𝅥	♩.	♩	♪.	♪	𝅘𝅥𝅯.	𝅘𝅥𝅯	𝅘𝅥𝅰
31	24	16	12	8	6	4	3	2	1

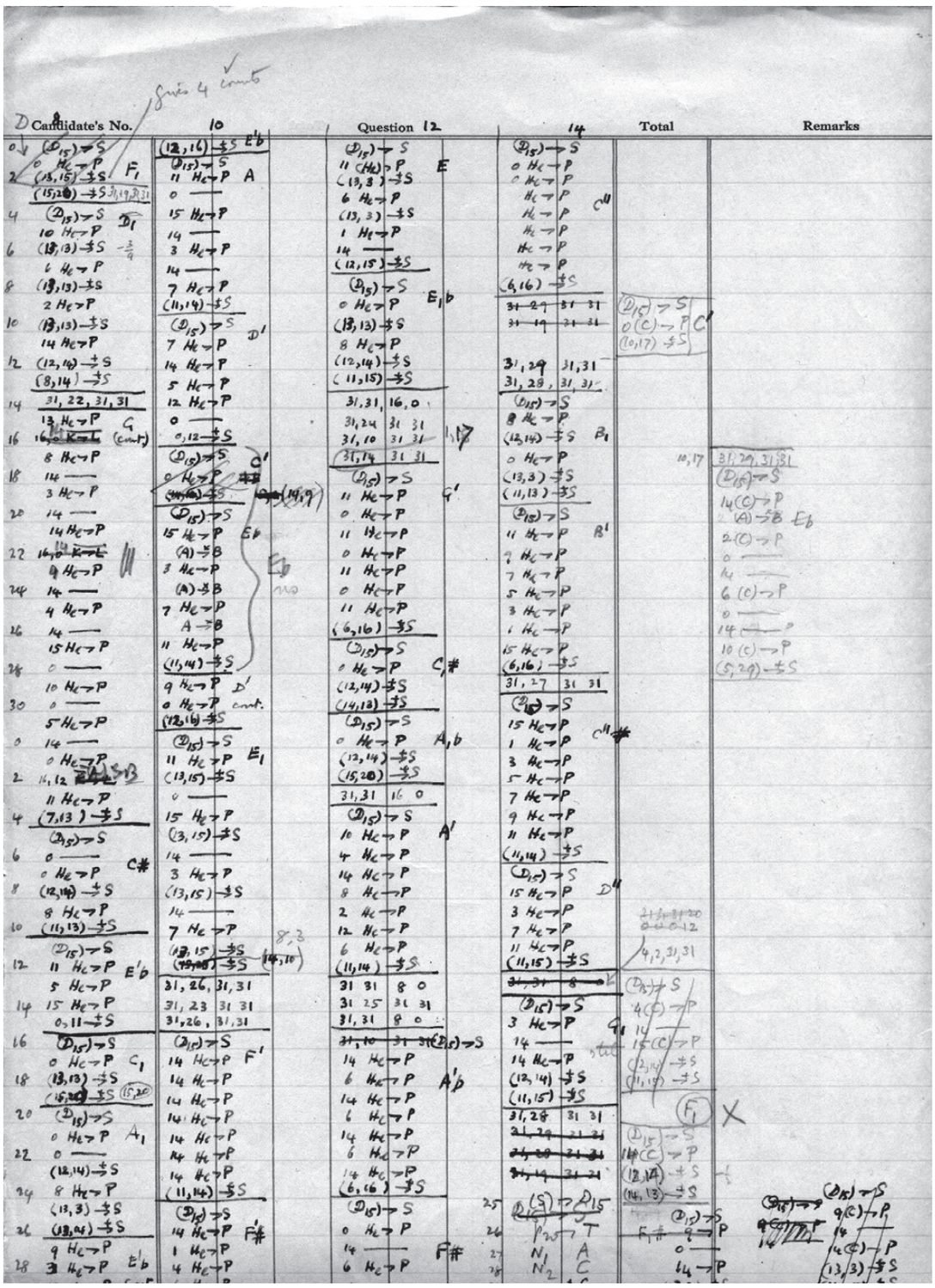
and intervening values provide for rallentando. The programmes provide also the notes G' A'b A' B'b B' C' C' , which are obtained by coding the tune with suitable downward transposition and playing it with upward transposition from N_2.

Cherry's instructions for the Music Programme. N1, N2 and I are registers that are set with console switches.

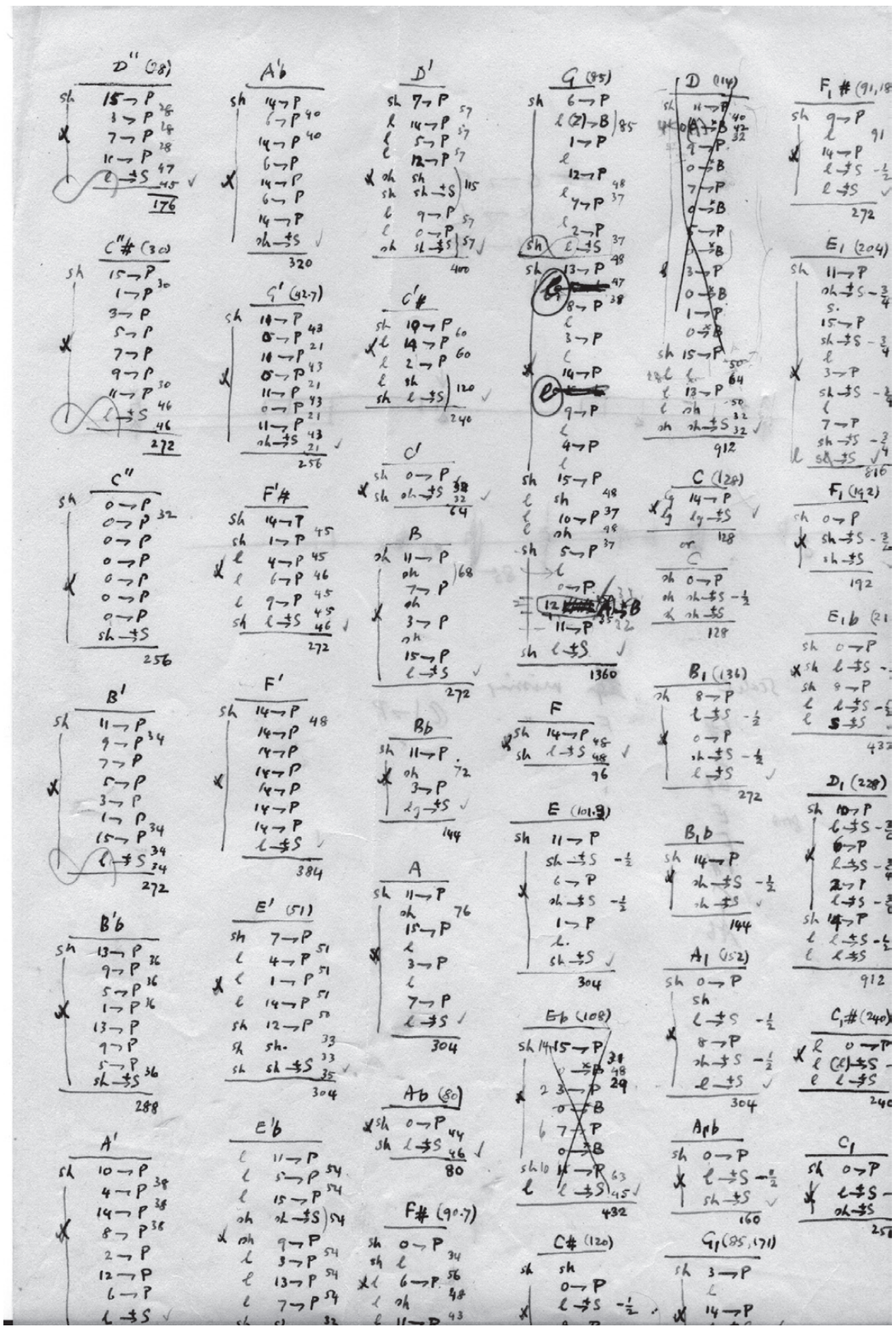
While reconstructing the music, Tom Cherry's original sketches and notes were found for the Music Programme. The program listing found amongst Cherry's notes was used to check the accuracy of the program from the punched paper tape and they were found to be the same. Below are a few extracts from his handwritten notes. It is interesting to observe the many corrections to the details of some of the musical pitches. Several of the pitch annotations also have numbers counting the minor-cycles (bit transfer time), thus representing the periods and delays of the instructions used in the pulse loops. Other numbers, before the instructions, represent memory locations used in the program for storage and data transfer and so on. Recent analysis, after the music was accurately reconstructed, revealed that different frequencies resulted from those that were expected. This is examined in detail later in the section on reconstructing the music.



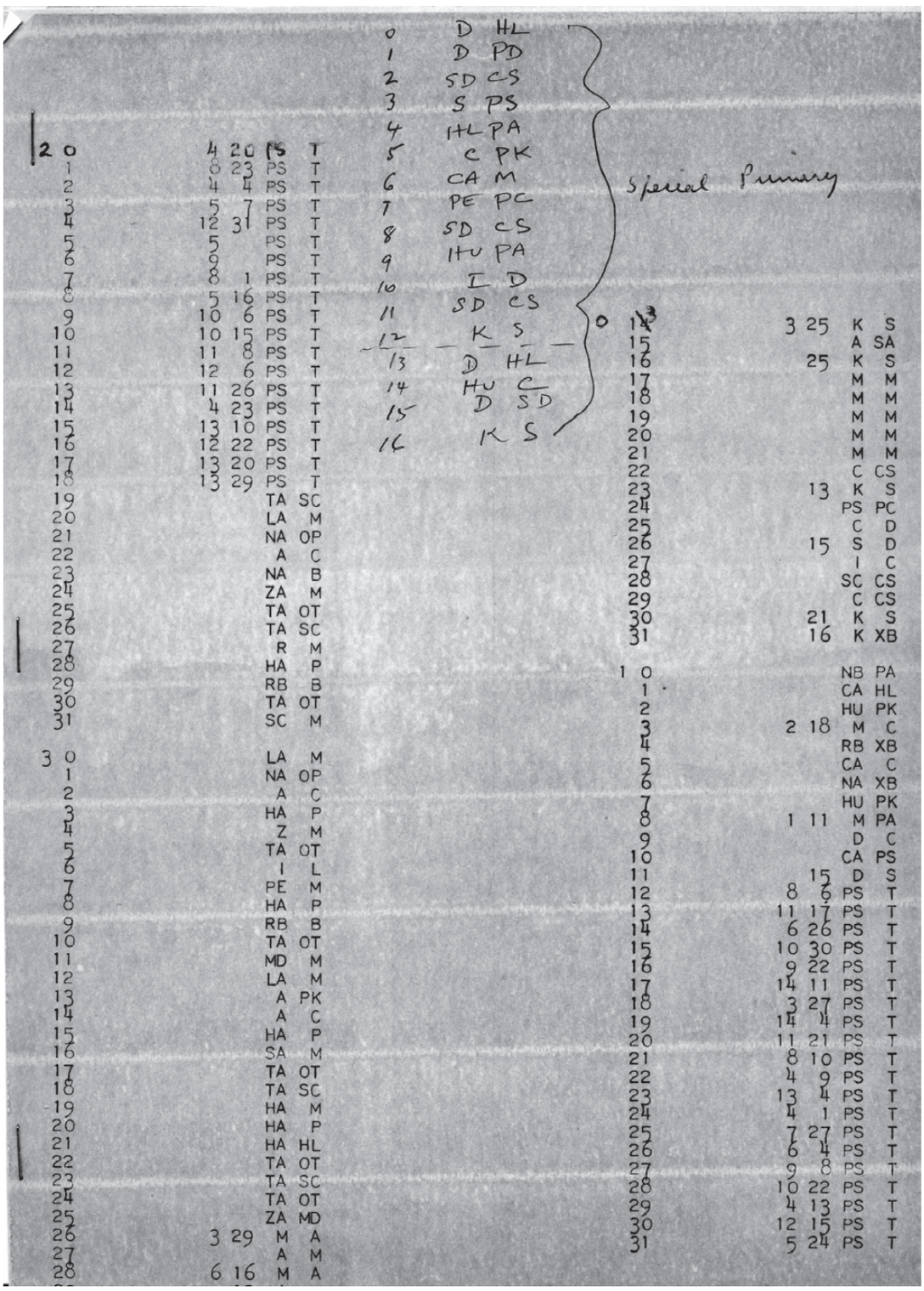
The first page of Cherry's notes for the Music Programme. Part of the program is on the left and the note data is on the right hand side of the page. The list of notes at the bottom (Pitch Code) is correct, but in the previous instructions for the program they are transposed.



Early note data structures for the speaker pulses. The symbol P means 'send the pulse in the source', here it is register HL, 'to the speaker'. This is an early example of the program as later Cherry found it better to use register C as the source for the data to be sent to the speaker.



A separate sheet of note data structures detailing the timing of the various loops of pulses and the instructions to produce the notes. The numbers show the count of minor-cycles for the instruction and thus the timing.



A listing of the first part of the Music Programme. The handwritten section 'Special Primary' is the special bootstrap loader. There are two columns of code.


P87
86-1

Music Reconstruction

The music played by CSIRAC was never recorded onto any audio storage format such as tape or disk. Little was known about the music, and several people who had heard it in the 1950s and 1960s were still around to tell the tales, but it was impossible to hear it any more. This was unfortunate as it seemed that a part of computer music history, which had occurred in isolation in Australia, had been lost. Also, an isolated piece of Australian music history was lost. Therefore, a plan was developed to reconstruct the music. As fortune would have it, three key people who would be needed for this endeavour, Ron Bowles, John Spencer and Jurij (George) Semkiw, were currently 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. Most other first generation computers were often 'cannibalised' for parts or simply dumped, although there is now a reconstruction of the Manchester Mark I. CSIRAC was in service long enough that its value was appreciated when it was decommissioned in 1964. CSIRAC's circuit diagrams, manuals and documentation still exist, as does all of its program library, which makes it not only one of the sole surviving first generation computers, but one of the best documented old computers. John Spencer was a programmer on CSIRAC. To this day, 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 PC. Ron Bowles and George 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.

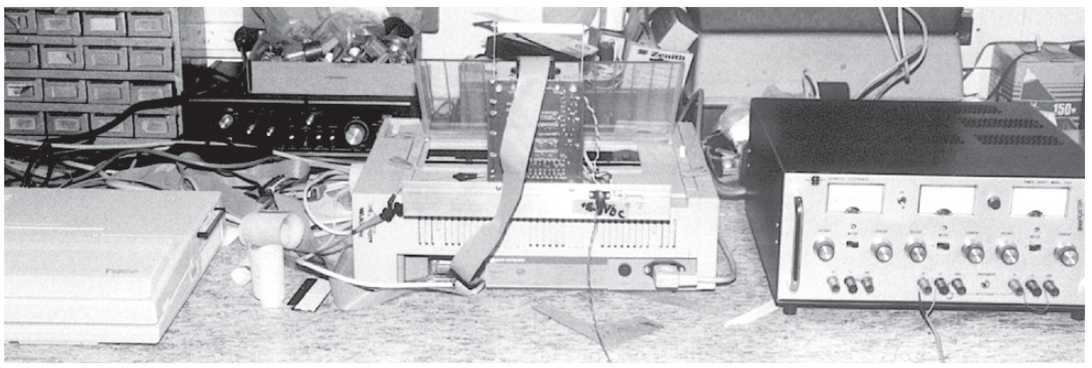
Ron Bowles, John Spencer and Jurij Semkiw examining the Sydney music punched-paper tape for the first time.

Initially, until just before the project to reconstruct the music had begun, only a few of the Melbourne music punched paper program tapes had been located. There were no program tapes of the Sydney music. Fortunately for the project, a couple of boxes of artefacts turned up that had been forgotten in an obscure archive. One box contained articles collected by Trevor Pearcey and another had items that belonged to Tom Cherry. Amongst other things such as manuals and documents, there was an early program tape of the Sydney music in Trevor Pearcey's box and there were some music program tapes from the Melbourne time in the CSIRAC program library. This was a very lucky turn of events.

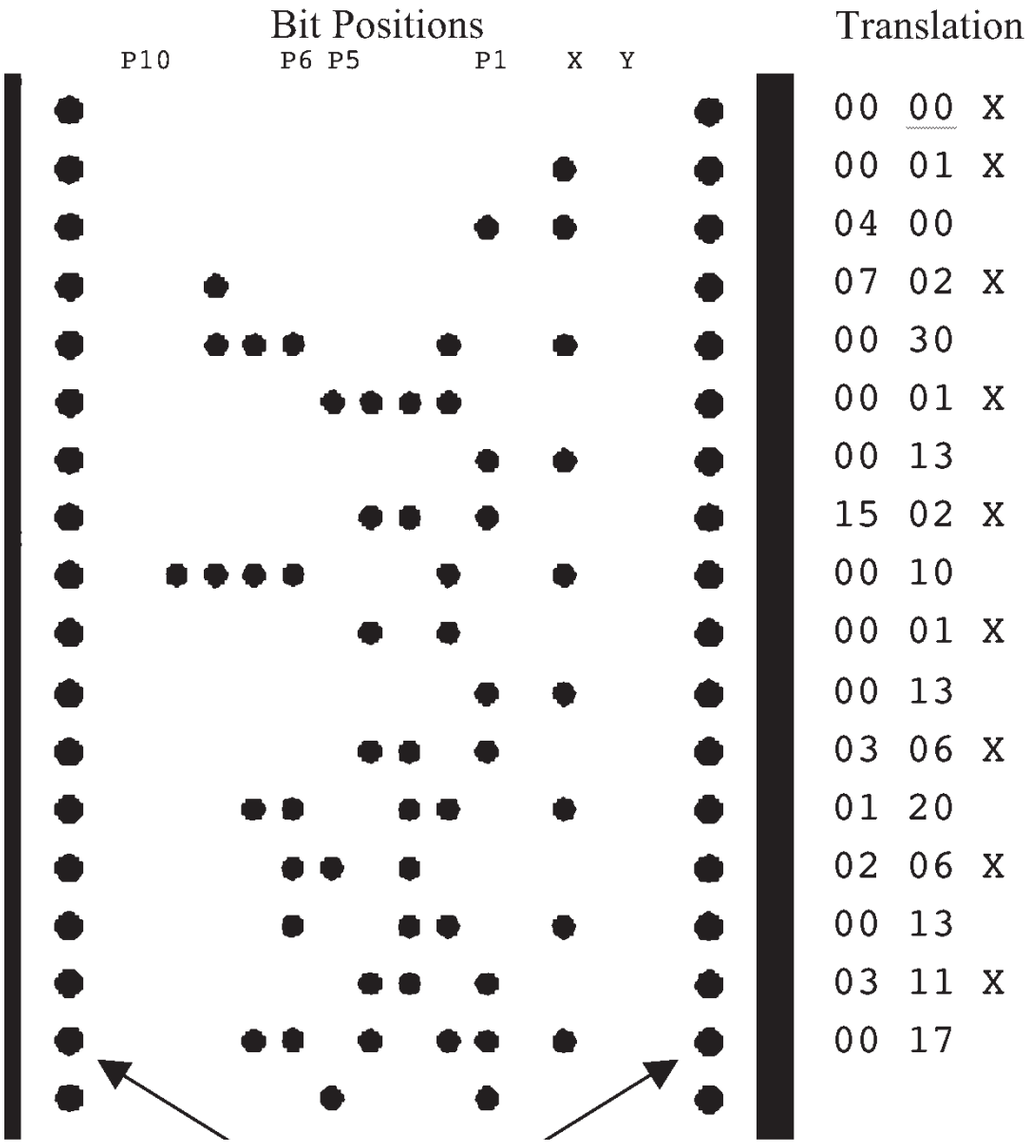
We were interested in reproducing the music as exactly as possible. Certainly to within an accuracy of better than one per cent of the waveforms that would have been heard at the time the pieces were originally played. The pulse shape as reproduced was well within one per cent accuracy of the CSIRAC pulse shape, but the pulse timing was at least 10 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. Several streams of activity were taking place in parallel and a good plan emerged to tie them together. It became increasingly clear that the best reconstruction of the music would be achieved by separating the reconstruction of the timing of the pulses from the reconstruction of the pulse shapes. The aim, simply stated, would be to reproduce precisely the pulse stream and thus the sound that emerged from CSIRAC's speaker. After some experimentation it emerged that the best course of action would be to read the program and data tapes (and get them working, a non-trivial matter), use several programs John Spencer developed from his emulator to generate the speaker pulse timing data, reproduce with hardware the pulse shapes that appeared at the speaker terminals, and combine these to reproduce the pulse stream. This pulse stream could then be played through a speaker and recorded if necessary.

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 tedious. Fortunately, there was a mechanical reader for the Melbourne tapes that was built by John Horvath of the Melbourne University Computer Science Department. John Spencer was using this to store the contents of all of the paper tapes, the subroutine library and so on, as 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. We prepared for the task of reading the Sydney paper tape by hand. To protect the program tape it was photocopied with a black background, this made the holes in the paper tape easily visible. We decided that we should each decode the 30 or so pages of dots and then compare our 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.



The reconstructed reader for the Melbourne tapes. The reading mechanism is attached to an old printer to pull the tapes through smoothly. The PC on the left controls the reader and stores the data as an ASCII file.



Example of punched paper tape as used in Sydney (arrows point to sprocket holes).

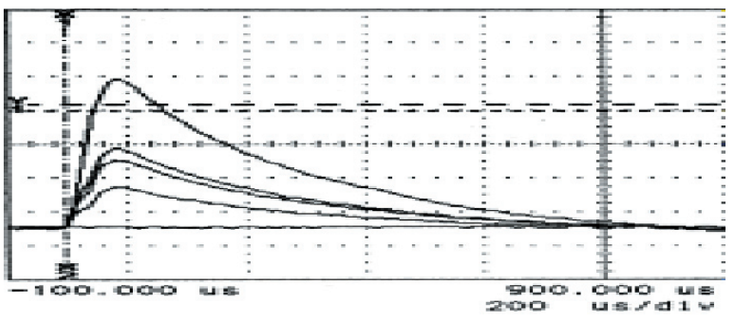
The CSIRAC emulator that had previously been developed by John Spencer was now going to be used to run the music programs. The emulator could not output sound the way

CSIRAC did because of the limitations of the PC architecture and it could not be modified to do so. However, while it could be modified to produce a file of speaker pulses this was an unwieldy solution and it was decided that the simplest way to proceed was to develop programs from the emulator core that would write a file of the *timing* of the speaker pulses. This file contained details of when pulses were sent to the speaker, which pulses were sent and the inter-pulse timing details. The core CSIRAC emulator code was not designed for this and Ron Bowles worked extensively with John Spencer on the new program to refine the timing of each instruction in the new program that would gather the correct timing details of the pulses sent to the speaker. Ron still has in his head the precise details of the timing of each memory location access, the timings of accessing the various sources and destinations and their interdependencies. This crucial information for a real-time activity, such as playing music, 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 these 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 speaker destination.

While the tapes were being read and the pulse timing programs developed, Jurij (George) Semkiw was designing and building some logic circuitry to reconstruct CSIRAC's pulse shape. The logic design allowed for the repetition of a single word, with the exact timing of the bits as in CSIRAC's logic circuits. Each of the bits themselves could be turned on or off with individual switches. The repetition of the data word was controlled by an oscillator. John Spencer had built an exact reproduction of the valve output amplifier based on CSIRAC's circuits and to achieve an accurate pulse shape, the output was sent through this amplifier with the same sort of speaker attached as was used on CSIRAC.

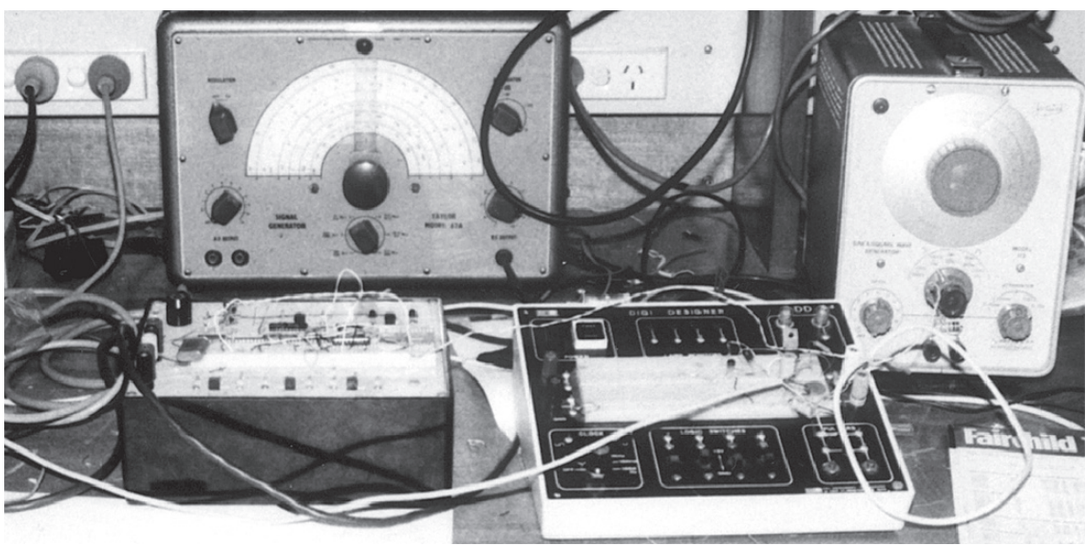
Pulses were played at various frequencies and with all

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. It was also checked audibly with a speaker, by listening to the pulse stream to check that it sounded the same as CSIRAC. This output was recorded onto a DAT (digital audio tape) recorder from across the speaker terminals.

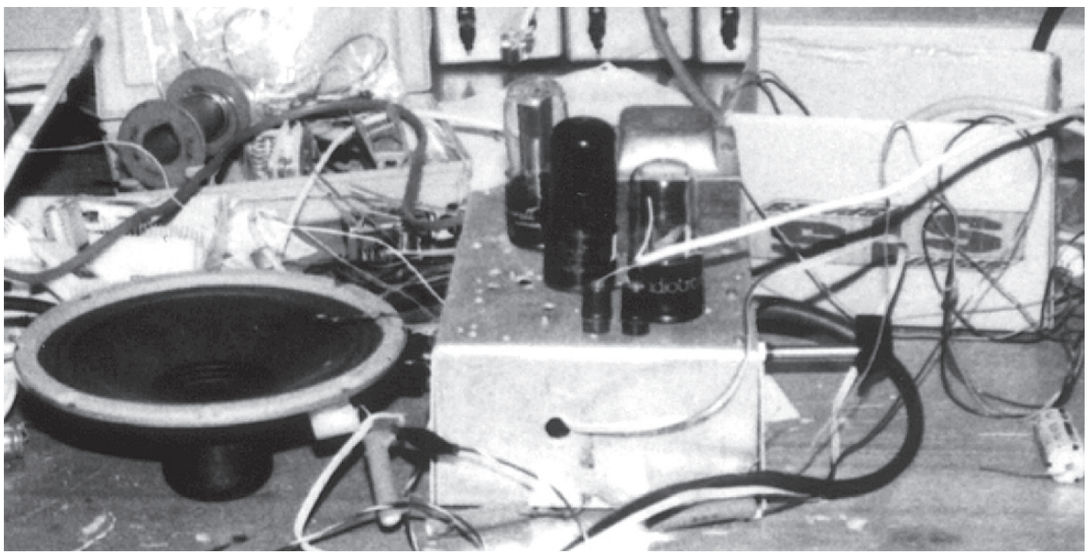


Oscilloscope screen image of the reconstructed speaker pulses as programmed in Melbourne. The four traces are for six bits on, four bits on, three bits on and one bit on.

The digitised pulses were accurately transferred to a computer via a digital link and the pulse waveforms examined in a digital audio editor. The pulse shapes for any given combination of bits had little variation from one moment to another, which was apparently the same for CSIRAC when it was operational according to the engineers. For each combination of bits in a word, a single pulse was isolated.



The logic circuitry used to recreate CSIRAC*'s pulse shapes.*

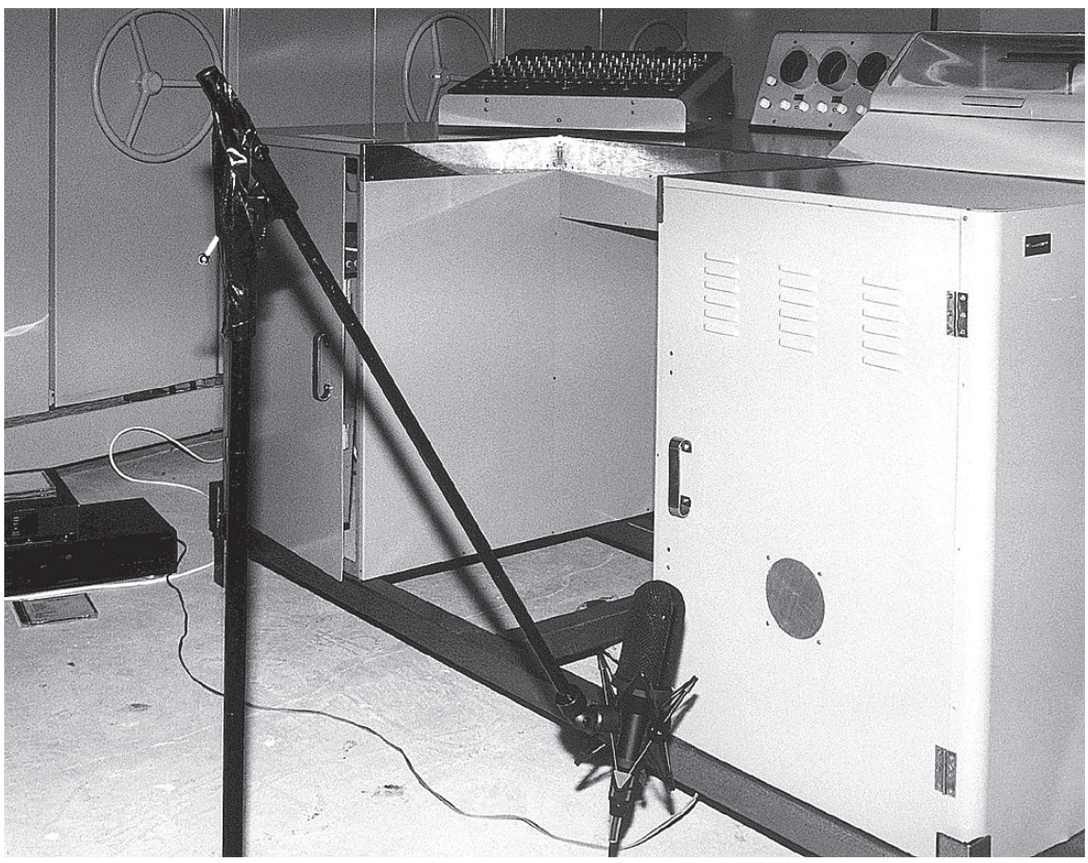


The valve amplifier, built to the original design and used to recreate the speaker pulses.

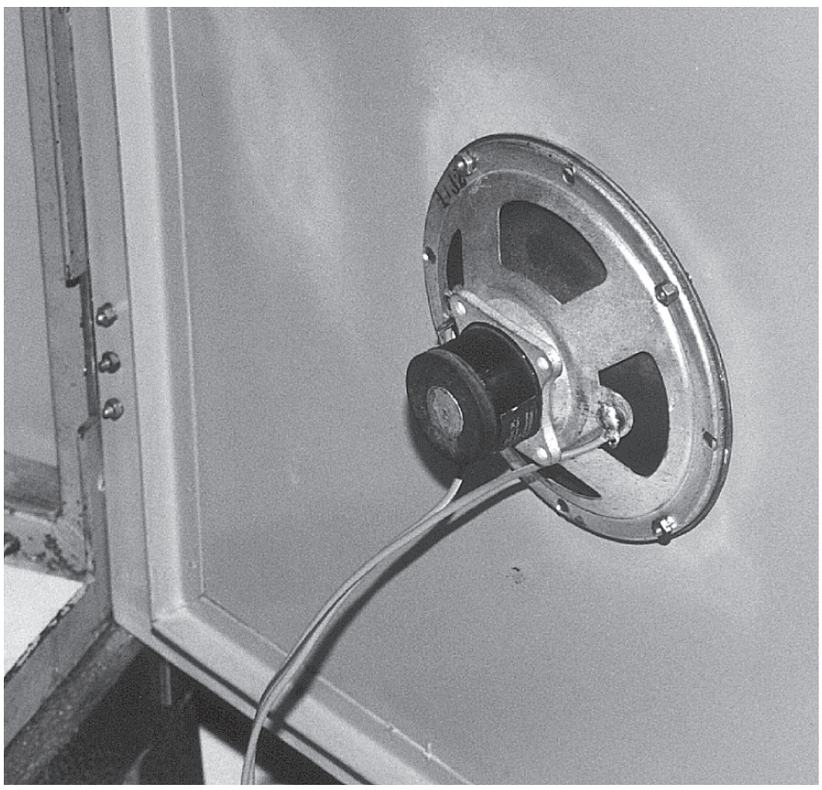
John Spencer wrote a program that contained the digitised pulses as data arrays. It would read the file of speaker pulse-timing data and apply the correct pulses in the time relationship specified in the timing file. This program wrote an output file of the pulses at the correct times, in effect a digital audio file of what CSIRAC would have produced at the time if a digital recording device were connected across the speaker terminals.

These files of the music as played by CSIRAC are an extremely accurate representation of the pulse stream that was sent to the speaker and a very good representation of how CSIRAC would have sounded. This was checked by playing the pieces to people who had heard CSIRAC play music and they all agreed that it sounded the same. However, one further step would provide the greatest authenticity. Museum Victoria, which now owns CSIRAC, agreed to allow access to CSIRAC and the console so that the music could be played through the speaker and recorded. The original speaker used in CSIRAC, a 5 inch Rola model 5C, had a somewhat torn paper cone so another had to be found. After calling many radio repair businesses it was discovered there were five available of the same model and vintage. After examination, a replacement speaker was selected that was closest in manufacturing serial number to the original and was also in excellent condition. This speaker was placed in the CSIRAC console door and the

music was played through it from an excellent quality Hi-Fi CD player and amplifier with a low output impedance so as to have the minimum influence on the sound and the waveform. The music was recorded with a Soundfield microphone, in a narrow cardioid configuration, directly to DAT. When the sound of this recording is compared to the digital audio files which resulted from the reconstruction, it sounds more alive and animated because of the various resonances and noises from the speaker and console.

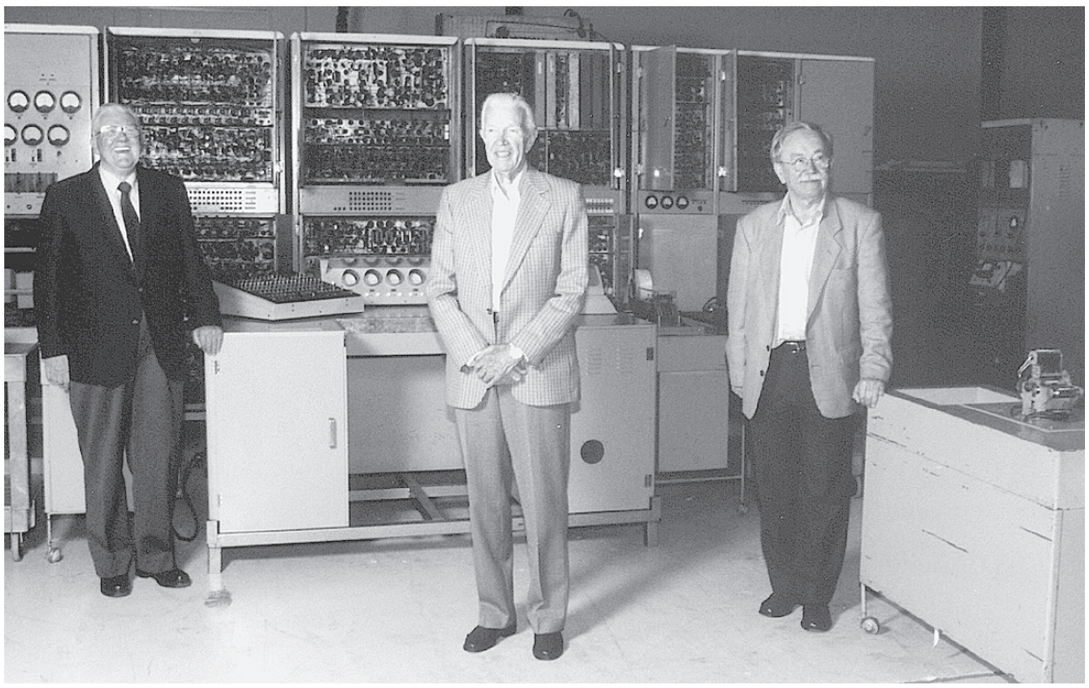


CSIRAC *again playing music and the recording of it. The storeroom was a close acoustic approximation of the Computation Laboratory. The CD player and amplifier are on the left.*



The replacement speaker as used in the recording, mounted in the console door.

The ultimate test of this process came in November 1999 when there was a celebration in Melbourne for 50 years of computing in Australia. People came from all over the country to be there, many of them had originally worked with CSIRAC and some of them had not seen it since 1955. It was gratifying to witness their astonishment when they heard the music. Several commented that they had not heard it since the 1950s and that they did not think it had been recorded then. They were surprised to find the reconstruction so convincing. One person also commented that even the out of tune or 'off' notes sounded authentic. This was reassuring as the notes that did not sound perfectly in tune were questioned, but the process had been checked many times for accuracy and no errors could be found.



John Spencer, Ron Bowles and Jurij Semkiw at the CSIRAC *50th birthday celebration, November 1999. (Photo courtesy Museum Victoria)*

Sound, Pitch and Note Analysis

The sounds made by CSIRAC were the result of data words being sent directly to the speaker. The timing of the data words, or the time periods between them, determined the frequency, or frequencies, of the resultant sound. Everything about the tuning and pitches in the notes is related to the machine architecture. Because of the machine architecture, the timing of instructions could vary. Understanding how and why this instruction timing occurs is crucial to understanding how the sounds were made and why the programming was so difficult, as this was critical for producing a repeatable sound. Ron Bowles, the longest serving maintenance engineer knows more about this than anyone:

> Well, something we called 'speed-up' occurred at various times in the processing or fetching of data or instructions. Speed-up occurred in both halves of the computer cycle. The cycle was split in half. In the first half, the instruction was extracted from the memory and the decoding process was set up. In the second half of the cycle the instruction was decoded to determine what was required to be done and the instruction was executed. The source gate of the information was opened and the destination gate was opened for the information to go to. In the first half of the cycle, the position of the instruction in the 16 different possible *timing* positions in the memory-tube determines how far along in the computer cycle the machine must be before the instruction is available. Remember that CSIRAC is a serial computer. If the instruction occurs early enough in the memory, the fetching of it can be completed in 1mS instead of 2mS. Also, while decoding and executing the instruction there may be speed-up because within the instruction itself is an address (all instructions contain one even if it's not needed) that could be of the memory or a register. Now depending on that address, if it's early enough in the computer's cycle then the instruction will be finished in 1mS. (Actually, it's 960µS, but we always said 1mS.) Thus, there's a total time of 2mS to fetch, decode and execute an instruction, which is as fast as the computer could operate. A computer cycle could thus take

> 2mS, 3mS or 4mS to complete before the next instruction could be undertaken. But, there's more, also, within the timing of the instruction itself, while it's being done. Each instruction is a 20-bit word and in each of those there is 3μS between each successive digit, so it can take 60μS for that to go through. So you have that 60μS to play around with again, but this could not have been used in the music program. So the main thing affecting the timing is where the information occurs in memory, in which of the 16 positions. Even when an instruction could take 2mS to complete, the timing of exactly when the instruction takes place in the second half (1mS) of that, could still vary because the address of the instruction could be in any one of 13 different time positions in memory and that could vary the timing by 0.72mS. Thus, the way to vary the time in which the instruction actually takes place is by varying the address part of the instruction for whatever you are using for the source. So even with a constant time for the whole operation, the time for a pulse to be sent to the speaker could be varied backwards and forwards. Sometimes, successive P (speaker) instructions would be in a loop with different addresses so that combined together they gave the desired result. This is how the high notes were achieved with any precision at all. There were three positions in the memory and the instruction addresses that always took 2mS to access. Those were the memory positions 12, 13 or 14 mod 16. The notes produced by the speaker are determined by the time intervals between successive speaker instructions rather than the computer cycle times in that set of instructions. Of course, these time intervals are intimately associated with the computer cycle times.

The tuning of the notes played by CSIRAC was as close to the diatonic, equal-tempered, scale as it was possible to make them at the time. However, being as close as possible may not always mean being accurate enough to sound perfectly in tune, especially for the highest notes, which were notoriously difficult to get right. Upon hearing the music, a modern listener will be alerted to several tuning anomalies. Below is an analysis of the 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
C#2	1	69.3		69.4
D2	2	73.42		(73.1) x 4
D#2	3	77.78		(77.2) x 2
E2	4	82.41		(81.7) x 4
F2	5	87.31		86.8
F#2	6	92.5		183.2, 92.1
G2	7	98		196.1, 97.5
G#2	8	103.8	104.2	104.2
A2	9	110	(108.7) x 2	(109.6) x 6
A#2	10	116.5	115.7	115.7
B2	11	123.5	(122.5) x 2	(122.5) x 2
C3	12	130.8	(130.2) x 2	130.2
C#3	13	138.6	138.8, 160.3	(138.9) x 2
D3	14	146.8	(144.9) x 4, 143.7	(146.2) x 7
D#3	15	155.6	157.2, (155.7) x2	(154.3) x 6
E3	16	164.8	(163.4) x2 , 166.7	(165.0) x 2, 163.4
F3	17	174.6	175.4, 171.8	(173.6) x 2
F#3	18	185	183.2, 222.2, 185.2	185.2, (183.2) x 2
G3	19	196	(196.1) x 2, 193.8	(196.1) x 16
G#3	20	207.6	(208.3) x 3	208.3
A3	21	220	(219.3) x 4	(219.3) x 4
A#3	22	233.1	(231.5) x 4	(231.5) x 2
B3	23	246.9	(245.1) x 4	(245.1) x 4
C4	24	261.6	(260.4) x 3	260.4
C#4	25	277.2	(277.8) x 3, 219.3	(277.8) x 2, 138.9
D4	26	293.7	282.5, 287.5, 282.5, 142.5, 222.2	(292.4) x 5, 127.2
D#4	27	311.1	248.8, (326.8) x 3, 166.7	(308.6) x 6, 154.3
E4	28	329.6		(326.8) x 3, 333.3, 165.0
F4	29	349.2		(347.2) x 6, 173.6
F#4	30	370		(370.4) x 2, 362.3, 370.4, 182.0
G4	31	392		(387.6, 793.7) x 3, 260.4

The table shows the standard note name which includes the pitch name and the octave, the note number in Cherry's program, the frequency of the note in the equal-tempered

scale and for Sydney and Melbourne it shows the fundamental frequencies in each note and the number of pulses used at that frequency when CSIRAC played music. 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 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. Also, there is anything from one to sixteen pulses in a loop to produce a pitch.

It can also be seen that some note loops combined up to five different pulse timings to try to produce a single note. Other note loops used up to sixteen pulses in a loop but with several of the same period, see for example the notes from D3 onwards in the above table. The note G3, as programmed in Melbourne, uses sixteen identical pulses to produce the note. The combination of multiple different pulse timings was used to approximate some notes because of 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. Thus, this approach to producing the notes was forced upon the programmer. Several of the notes use harmonics or sub-harmonics of the desired frequency. Some others, such as F3 as programmed in Sydney, have two components which are a little each side from the desired frequency, which was possibly an attempt to psychoacoustically trick the ear into believing it was hearing the pitch between the two that existed. Unfortunately, this will not work as it will lead to a perception of dissonance because different tones will be mixed with the two frequencies. There is also frequent use of sub-harmonics, typically at half of the fundamental frequency required. As those involved at the time report that there was no attempt to modify the timbre of CSIRAC, it was decided that the use of sub-harmonics was forced on the programmer by machine limitations. The CSIRAC pulse waveform has strong harmonic components at integer multiples of the fundamental frequency (see later spectral graphs), so sub-harmonics an octave below the

required frequency have strong components of the required frequency.

Within Cherry's program and also from his handwritten notes, there are indications of extremely deep knowledge of the timing of CSIRAC and several clever tricks to produce pitches at the correct frequency. By inventive and resourceful programming, it could be ensured that a single instruction could be executed multiple times without jumps. This technique was useful because of the memory limitations, but it was unavailable in Sydney because of a difference in the workings of the sequence (S) register. When the S register was incremented (M PS) and it carried from the lower half-word to the upper half-word, it could change the program flow by repeating itself, thus providing a timing delay and saving memory space. The main use of this, counting in the lower half of the S register, was to allow the functions of incrementing, testing and jumping to be combined in a single instruction. This advance in CSIRAC's design was well exploited by Cherry and it is probably the main reason for the extended range of the machine in Melbourne. In Sydney, four instructions would have been needed (increment, test, jump to exit, jump to start) for the same result and this would have taken too long for the higher notes to be produced. The root of the problem is that while it is usually possible to create a sequence of appropriately timed pulses, there must be a loop return, or exit, at the end of the group of pulses. The timing of the loop return can be influenced mostly by its position in memory, the instruction address and slightly by a preceding pulse (P) instruction. However, it will inevitably add an extra delay of at least 2mS (see previous interview with Ron Bowles) between the last P instruction in the loop and the first P instruction. Remarkably, with careful and resourceful programming, Cherry was often able to juggle the timing of instructions, their position in memory and so on such that the time interval between the last pulse in a loop and the first pulse, including the loop return instruction, was exactly equal to the time interval between the P instructions within the loop.

From the preceding table, it can be seen that some notes,

for example D4, have clearly non-harmonic components. Particularly for the note D4, one would think that if five pulses can be created in sequence that are very close to the desired frequency then one more should be possible, but an understanding of the machine shows that it was not possible. However, the last pulse, providing a frequency of 127.2Hz, has no harmonic relationship to the note D4. Careful analysis of Cherry's music program reveals why he did this. It is because of the various machine limitations or idiosyncrasies. He must have realised through experimentation that the low frequencies were being produced and why, so he made adjustments to minimise 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 this caused the inharmonic pulse. There were also other significant obstacles to be avoided with the timing granularity and memory limitations.

Within Cherry's Music Programme there is an apparently unused data section of upward and downward glissando. As programmed, this would have been too fast to be of much use. However, it shows that there had been work on glissando or attempts at more expressive output, which is required to avoid the music sounding 'mechanical', or to more accuratley simulate a person playing an instrument. The Music Programme could reportedly also provide for rallentando. It is most probable that the in-harmonic frequencies in some notes are a direct result of purely programming artefacts forced on the programmer by the hardware idiosyncrasies. The explanation for some of the very low frequencies observed in the higher notes (see later spectral analyses) is that for some notes CSIRAC was running pulse loops to produce notes close to the clock frequency and there was no fine timing control over the return or loop instruction. This would modulate the pulses and result in a frequency that had a period that was the sum of the delays of the pulses in the loop. This would unquestionably result in inharmonic

frequencies. There were certainly cases where it was simply unavoidable that some inharmonic pulses were sent to the speaker. It is clear from the information in the table that for the upper notes there must have been many problems in producing an accurate pitch. Today, there are much better program analysis tools than at that time. A complete analysis is readily available now because of the tools available, whereas at the time it would have been a time consuming procedure to produce one by hand. Examination of Cherry's handwritten notes from the development of the Music Programme, show that probably such a detailed analysis was not readily available then and that frequencies resulted other than those that were expected. Ron Bowles commented during the music reconstruction:

> I don't know why the Prof. has programmed these couple of notes like this (referring to F#2 and G2). I have worked out a better way to program them that would result in a pure tone, but I suppose you're more interested in authentically reproducing the original.

It is worth examining the improvements developed by Ron Bowles, as it shows how non-trivial this is. Below is a comparison of the original code by Tom Cherry and the improved code by Ron Bowles for the note F#2. Ron Bowles spent many hours developing this and managed to fit the improvements for both notes into the Music Programme, without increasing its size, by slotting them into some unused space. The programming of CSIRAC cannot be fully described here. However, in the code fragments below, comments come after a double forward slash (// - a modern convention), the first two columns of code represent the memory storage location, the next two columns represent the high and low parts of the instruction address and the next two columns are the source and destination respectively. The comments explain what the code means and they also give the contents of the sequence register at important points.

```
5 5 15 D S // exit when outer loop ends
5 6 9 C P // entry point, send to speaker
5 7 14 M M // move memory, ie wait
5 8 14 C P // send word in C to speaker
5 9 12 14 M PS // (12,14)' holds 31 31 16 0
5 10 11 13 M PS // (11 13)' holds 31 26 31 31
```

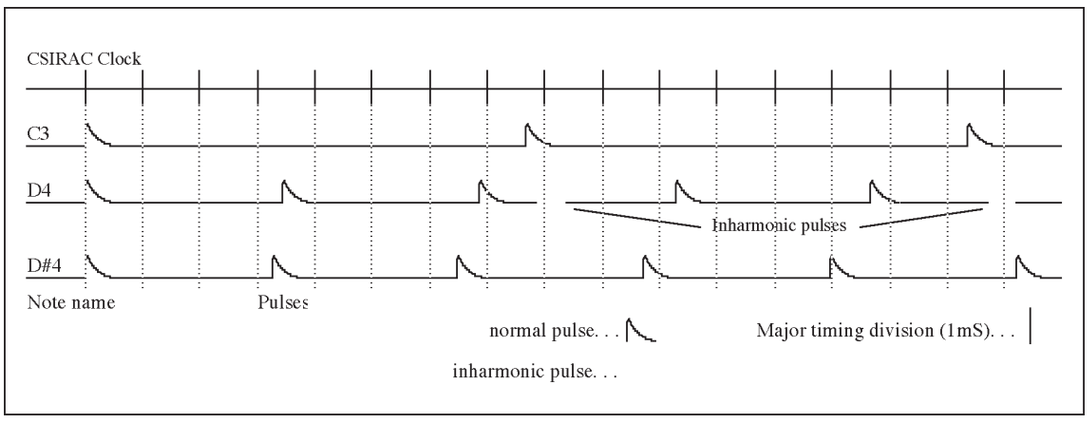
Tom Cherry's code for note F#2, c1957.

```
4 0 15 D S // exit when outer loop ends
4 1 1 C P // entry point for note 6
4 2 12 14 M PS // (12,14)' holds 31 31 16 0
4 3 14 M M // move memory, ie wait
4 4 13 C P // send word in C to speaker
4 5 12 14 M PS // add M contents to S register
4 6 14 M M // move memory, ie wait
4 7 9 C P // send word in C to speaker
4 8 12 14 M PS // add M contents to S register
4 9 14 M M // move memory, ie wait
4 10 5 C P // send word in C to speaker
4 11 12 14 M PS // add M contents to S register
4 12 8 3 M PS // (8,3)' holds 31 19 31 31
```

Ron Bowles' code for note F#2, 2000.

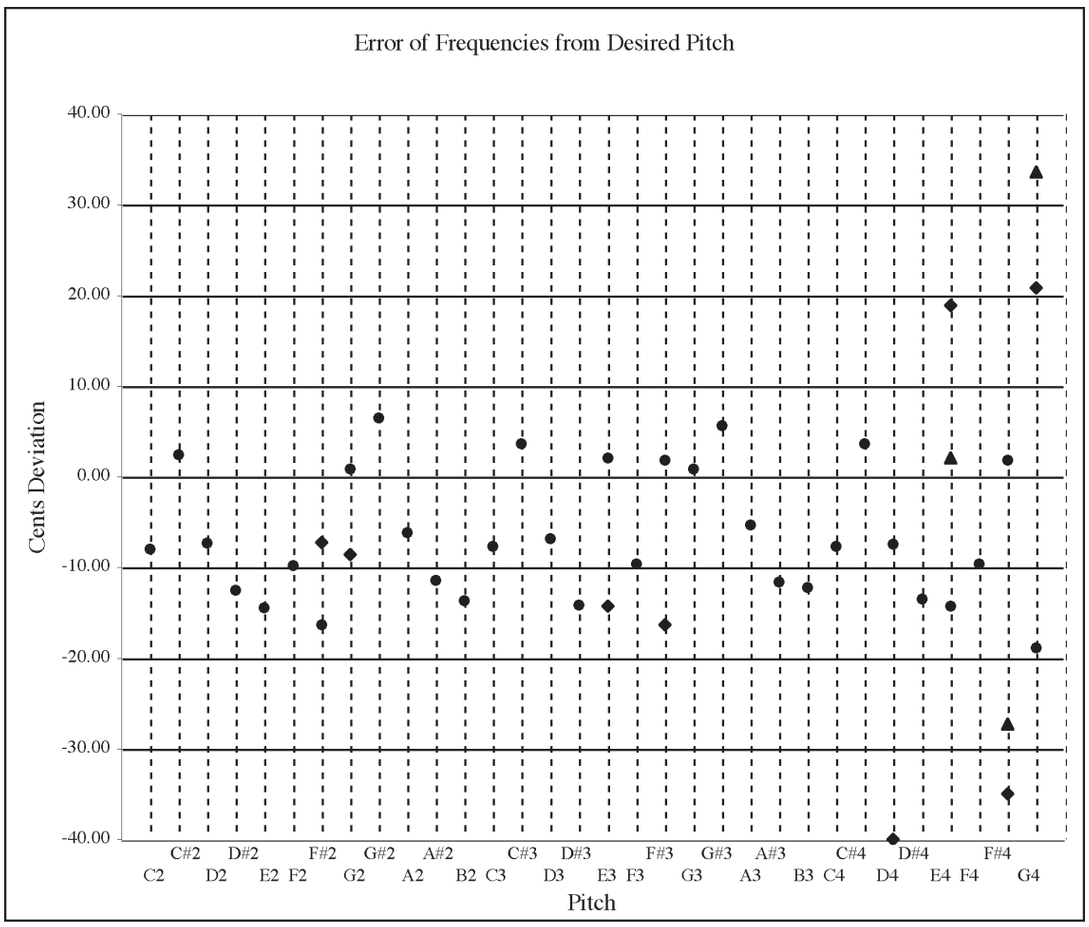
The code fragments above show that it was not a simple task to produce a pure note from CSIRAC, even within the frequency range where it required least effort. The many addresses used when moving memory (waiting) and for sending data to the speaker indicate the complexity of the task. Different access times and pulse timings are achieved through different memory addresses. The timing of all instructions was refined by Ron Bowles in the programs that gathered the pulse timing data. The changes to produce a pure tone for the note G2 are similarly complex.

The diagram below displays the pulse periods for three notes. It can be seen how, as described above, 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 127.2Hz frequency for the note D4 can also be seen.



CSIRAC *major cycle timing with pulses for notes C3, D4 and D#4. 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.*

To show the errors in the tuning of the notes, below is a graphical representation of the data in the previous table of pulse frequencies. This shows the error of the frequency, or frequencies, of the note from the desired pitch. The error is expressed in cents (hundredths of a semitone) from the desired note. It is difficult to say how much deviation is perceived as 'out of tune' as the perception depends heavily on the context and the waveform involved. However, as discussed, several of the notes have multiple dissonant frequencies, which would certainly be perceived as not in tune either with the note itself or with the scale. In addition, there are a few notes, for example E2, B2, D#3 and D#4, that while pure in pitch and without dissonant artefacts are probably out of tune enough with the scale to be noticeable, especially by a trained listener.

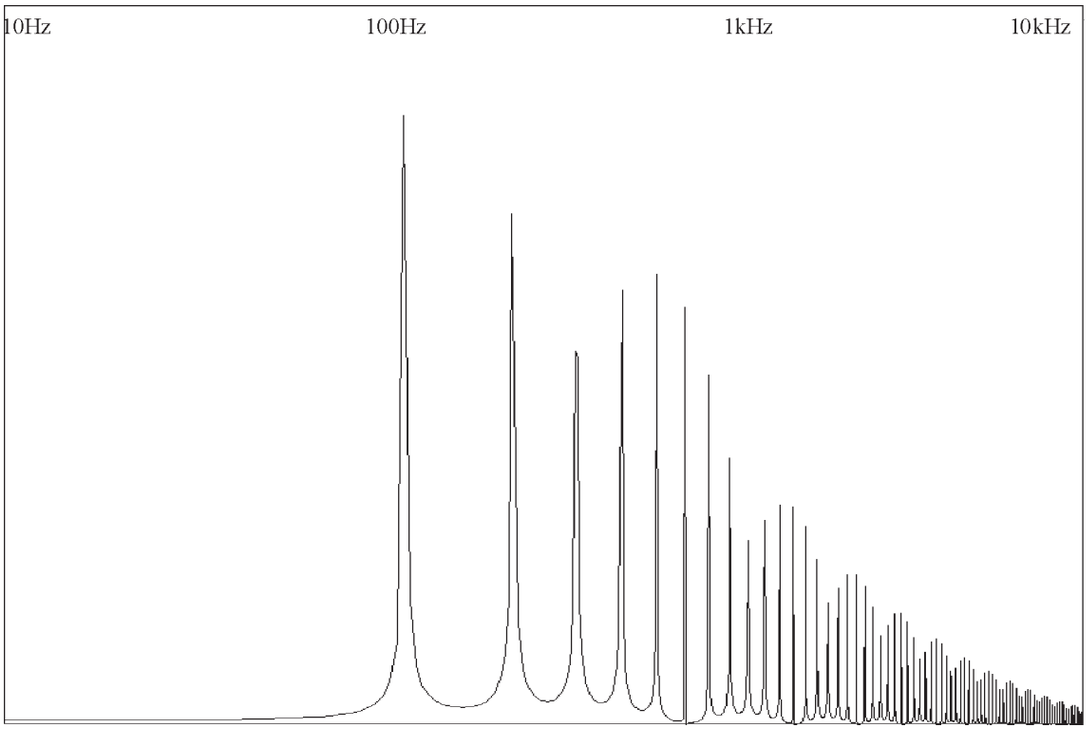


Accuracy of note tuning expressed in cents. Multiple vertically aligned points show notes with multiple frequency components. For notes with multiple parts, the first component is marked with a circle, the second with a diamond and the third with a triangle. Octave displacements have been collapsed to a single octave.

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 spectrum analysis of the notes can show these pitches and help to explain the sound or timbre of the various notes. The spectrum graphs below were taken of the pulse stream as delivered to the speaker, that is, as delivered to the speaker terminals. The speaker would filter this stream, mostly by reducing the higher partials. The vertical scale is linear amplitude, the horizontal scale is logarithmic frequency with 10Hz on the left axis and 10kHz on the right axis.

The harmonics for the note C3 show the spectrum 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

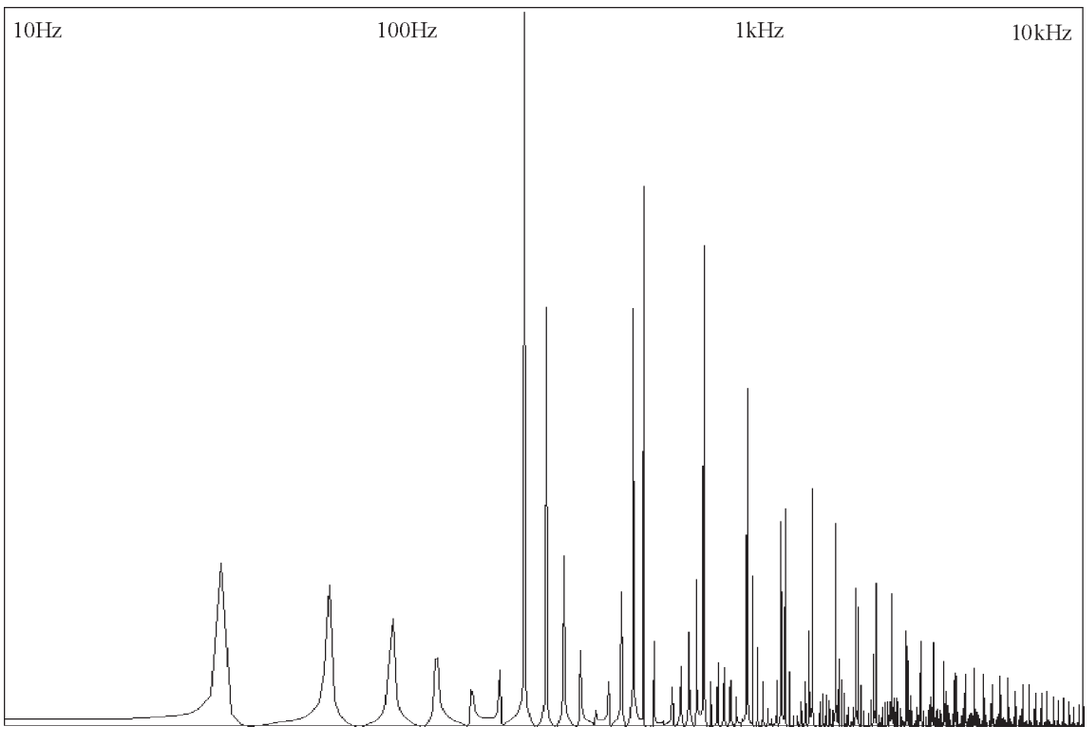
harmonic components. There are strong harmonics to over 10 times the fundamental frequency. This is somewhat different from the spectrum of a classic sawtooth waveform which has a regular harmonic distribution of 1 x 1F, 1/2 x 2F, 1/3 x 3F, 1/4 x 4F and so on, where F is the fundamental frequency. It can be seen that the upper harmonics are considerably stronger than would be expected from a sawtooth waveform. This would mostly be caused by the nonlinear decay of the pulse, making it more like a short impulse with the resulting increase in higher harmonics. The formant-like bands, centred at approximately the 5th, 11th, 17th, 23rd and so on harmonics are a function of the wave shape and remain in such a harmonic relationship in other notes.



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

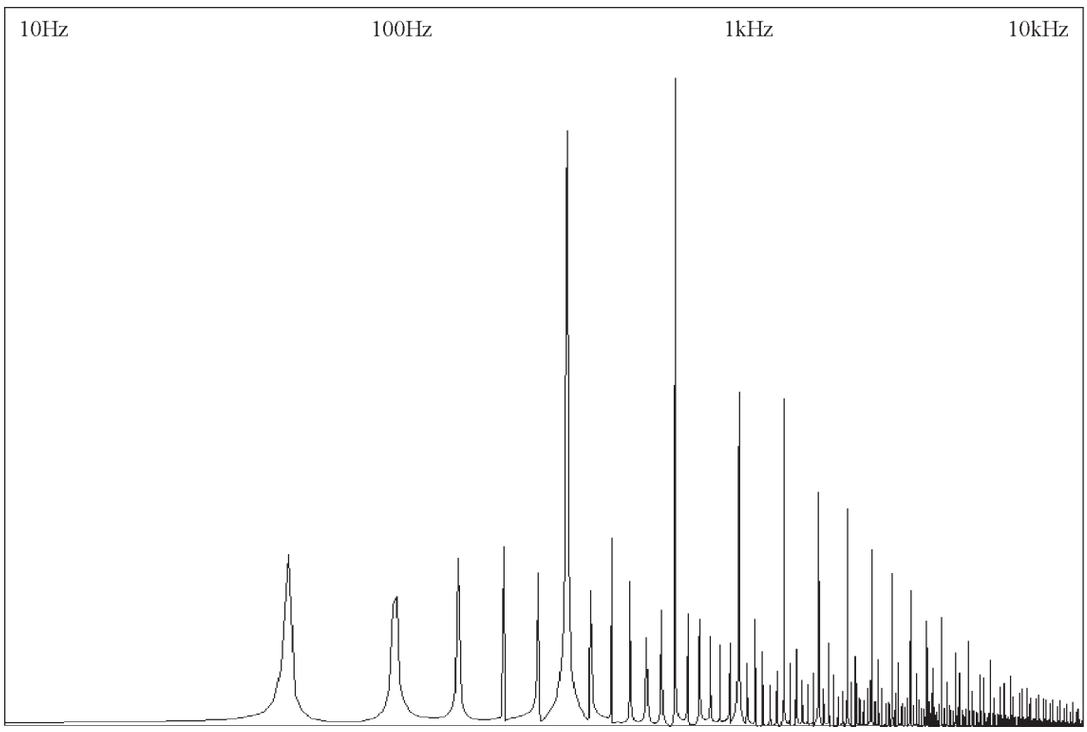
The upper notes of CSIRAC's range were the most difficult to program (as discussed previously), which is clearly audible from listening to them. The note D4, with a fundamental frequency of 292.4Hz, has a pulse timing that gives rise to an additional frequency component of 127Hz. However, this does not explain all of the frequencies that appear in the output. The upper notes (as all others) sound like a very accurate

reconstruction to those who heard them when CSIRAC was operating. There are clearly audible lower frequencies in the output, which sound dissonant or inharmonic. The spectrum shows, surprisingly, not only the expected frequencies at 292Hz and 127Hz and their associated harmonics, but also a frequency component at about 40Hz. The 40Hz component also appears to give rise to many odd sideband frequencies of all of the harmonics. This would explain why the note sounds quite 'rough' and inharmonic. However, the existence of the 40Hz component took some investigation to explain. The difference between the planned fundamental of 292Hz and the second harmonic of the spurious 127Hz frequency (254Hz) is 38Hz. This is quite close to the mystery 40Hz but the spectrum analysis shows not only a clear 40Hz component but also clear harmonics of it at 80Hz, 120Hz and so on. The precision of the measurement makes it unlikely that it is a difference frequency. After considerable investigation, the reason for the 40Hz component became clear. It emerged 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 3.42mS, 3.42mS, 3.42mS, 3.42mS, 3.42mS, and 7.76mS. The sum of the loop pulse periods is 24.96mS, which gives a frequency of 40.06Hz. The loop return causes a modulation of the pulses at the period of the execution of the loop. This gives the 40Hz component in the output. The seven highest notes played by CSIRAC all have this structure and they also have the spurious low 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.



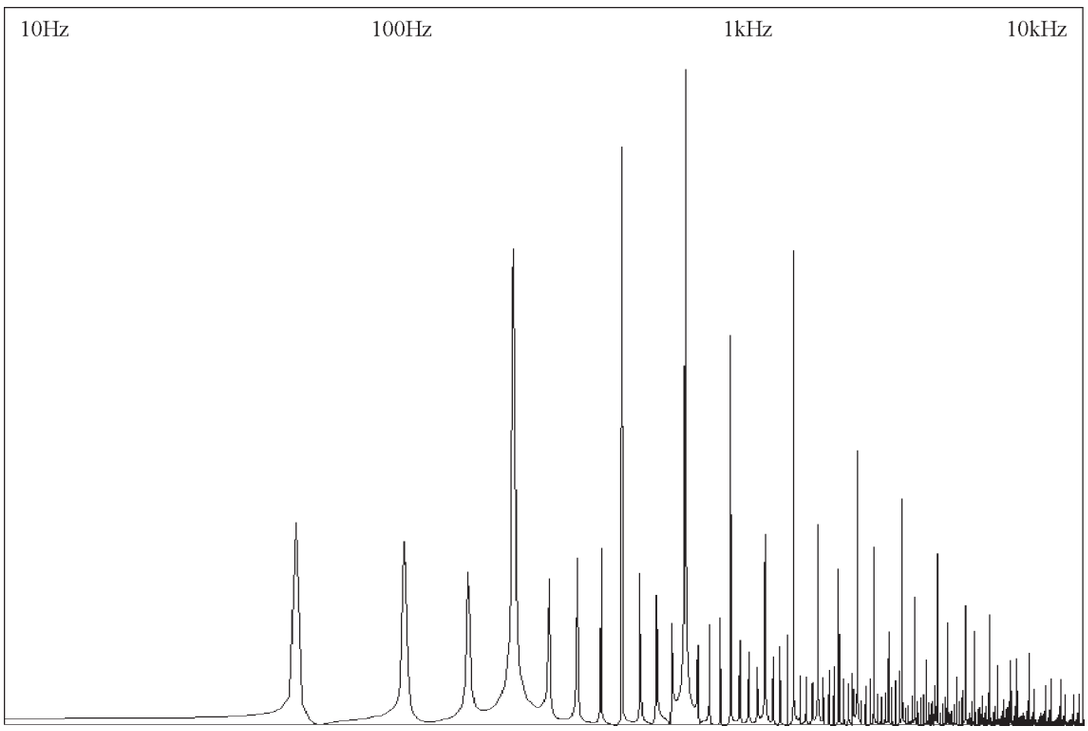
Spectrum of note D4, fundamental 292.4Hz. Note the low frequency artefacts as discussed above.

The two highest notes, F#4 and G4 share some common elements. The second harmonic is strongest in both notes and the formant character is missing from both. This is probably because the pulses do not get the time to decay, as in the lower pitched notes, before another pulse is generated and the formants are most likely caused by the nonlinear decay of the pulses. F#4 has the expected peak at 370Hz and a slightly stronger second harmonic at 740Hz. Beyond this, the amplitudes of the upper harmonics drop off more rapidly than the lower notes. As with all seven of the highest notes playable by CSIRAC, F#4 also has a strong low frequency component, caused by the sum of the periods in the pulse loop of the note. This is, however, unlike that of D4 in that it is harmonically related to the note. The low frequency component is based at about 62Hz. This is close enough to the note B1 to make F#4 sound not quite as dissonant as D4 does with the inharmonic low frequency component. There is also a string of harmonics that appear to be 62Hz sum-and-difference sidebands of the main frequencies. These sound very dissonant in the mix of frequencies.



Spectrum of note F#4, fundamental 370.4Hz. The second harmonic is the strongest.

The two strongest frequencies for G4 are 520Hz and the 775Hz second harmonic. The fundamental of 387Hz is low in amplitude and the second harmonic is the main frequency indicating the pitch G4. The intentionally programmed frequency at 260Hz is harmonically related to G4, it being the note C4, and the frequency of 520Hz is the second harmonic of that. The reason for the requirement of the 260Hz component is, however, something of a mystery. Probably the best explanation is that it was useful to adjust the low frequency component to a less dissonant pitch. The low frequency component, which is a result of the note programming structure as in D4, is 65Hz and it is also harmonically related to the fundamental. The low frequency component is almost precisely the pitch C2, two octaves and a fourth below the fundamental of the note. Again, there appears to be a number of sidebands about the main harmonics caused by the low frequency component, which would add dissonance to the sound.



Spectrum of note G4, fundamental 387.6Hz.

The spectrum analysis of the notes is useful for explaining the sound and timbre of the various notes CSIRAC played. The low frequency components of the upper notes appear to have been a programming artefact that was unavoidable because of the timing granularity and the loop structure of the notes, as described above. They do not appear in Cherry's handwritten planning of the program. In addition, when Geoff Hill was in Sydney and first writing music programs, he appears to have been forced into a similar situation to get higher notes from the hardware. The low frequency components appear to be mostly artefacts of the machine. Cherry was able to manage a compromise, amongst the myriad that were required, to get many of the spurious low frequency components to have at least some simple musically harmonic relationship to the intended note. This is probably an aspect of his programming skill that is difficult to appreciate now.

To assess the achievement of the programmers we need to remember the state of technology at the time to appreciate the challenge. CSIRAC was first operational when there was only a handful of computers in the world and there was no suitable digital-to-analog or analog-to-digital conversion technology. There was no computer music practice to base their work on. Every problem appeared original and had to be conquered for

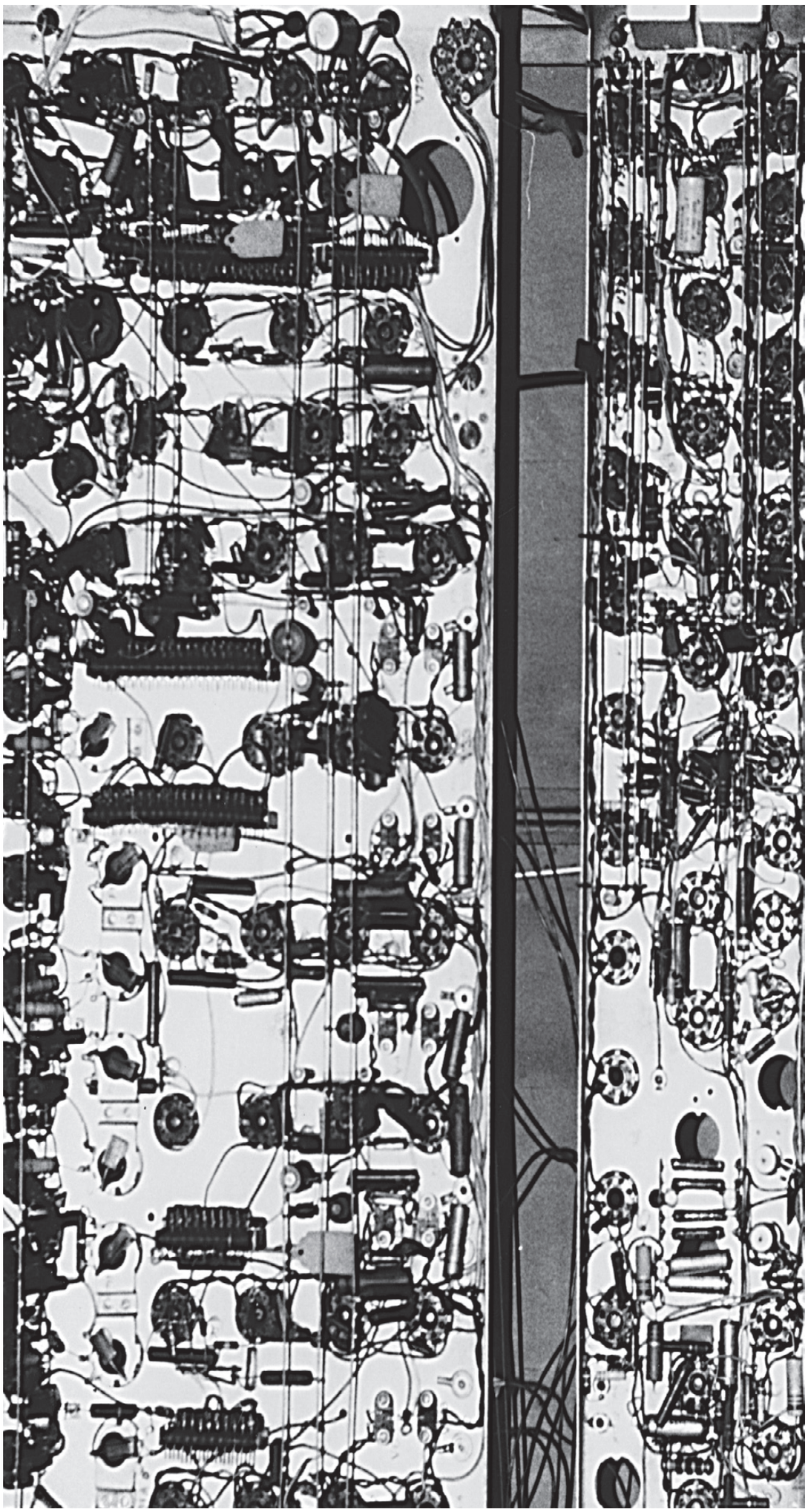
the first time. The task of generating a consistent tone with pulses sent to a speaker and controlling this to play music is theoretically simple, but in practice it proved enormously difficult. It is in the details of the programming that the skill of the programmer emerges. The first problem is that there are only 768 memory locations available for use. Also, the access time of various memory locations varied. This is what made it possible to send pulses with varying periods. The previous pulse timing diagram graphically displays the problem. Pulses were required to be sent to the speaker with a regular period that had no relationship to the period of the clock. A closer examination of the previous code example will also help. For example, a sequence of instructions to produce a note of medium complexity could look like this:

```
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
```

In this example, M1 to M5 are different memory locations with different timing delays. It is often a lack of memory locations with suitable timing that causes the artefacts in the notes. Time waiting, say t1 to t3, also uses memory access for the instruction execution and occurs at variable times. The time waiting was often difficult to juggle with the available memory and the memory timing required. This is what would lead to having several pulses with the correct period and one or two pulses with different periods. Then there was the test and loop section, which was required to play the note for the specified duration (N). As this determines the period between the last pulse in the loop and the first, it is crucial to program the timing of this accurately for the note to sound correct. It was very difficult to juggle this instruction section such that it executed in the correct time and sometimes there

was a compensating delay added before the first pulse in the loop. Such a structure was needed for every note and there was also the need to implement a program to control the execution of the notes, read the tape and so on, all in 768 words of memory. This helps to show the skill involved in the programming.

It is noteworthy that proficient programmers could program CSIRAC to design studies for large parabolic antennas, perform calculations for astronomical functions and ionospheric refraction, perform analyses of cloud droplet dynamics, make calculations for aircraft design and optics, perform X-ray crystallography analyses, perform the calculations for major engineering projects such as the Snowy Mountains Hydroelectricity project and perform calculations for large building thermodynamics. However, programming CSIRAC to play music was an altogether more difficult task, which required specialised programming skills and knowledge of the machine. That is why, in the very 1950s, the computer playing music was astonishing not only to the public but to other programmers too.



Finale

CSIRAC was a unique technological achievement in Australia. Unrecognised until now, 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, possibly even the first. It accomplished this through a happy confluence of events and the very intelligent and diligent work of a couple of people. The technological feat was extraordinary given the state of the hardware used to support the programming and produce the sounds.

The music itself may now seem very crude unless it is understood in the context of its creation. It was created by engineers who were not knowledgable 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. CSIRAC had a lot to offer composers. It could play almost any frequency within its range, so those interested in microtonal music, or alternate 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 it would have been a significant programming challenge. Additionally, with suitable programming, CSIRAC would have played any rhythm and could have produced rhythmically complex pieces or even arhythmic music. There is also the issue of using it as a composition tool, which some composers were interested in and some of the programmers would have been happy to help with the programming. Perhaps we would have had very early examples of computer-based microtonal music and algorithmic composition. There were composers nearby who were interested in these things. Percy Grainger was at Melbourne University 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 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 Physics. 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.

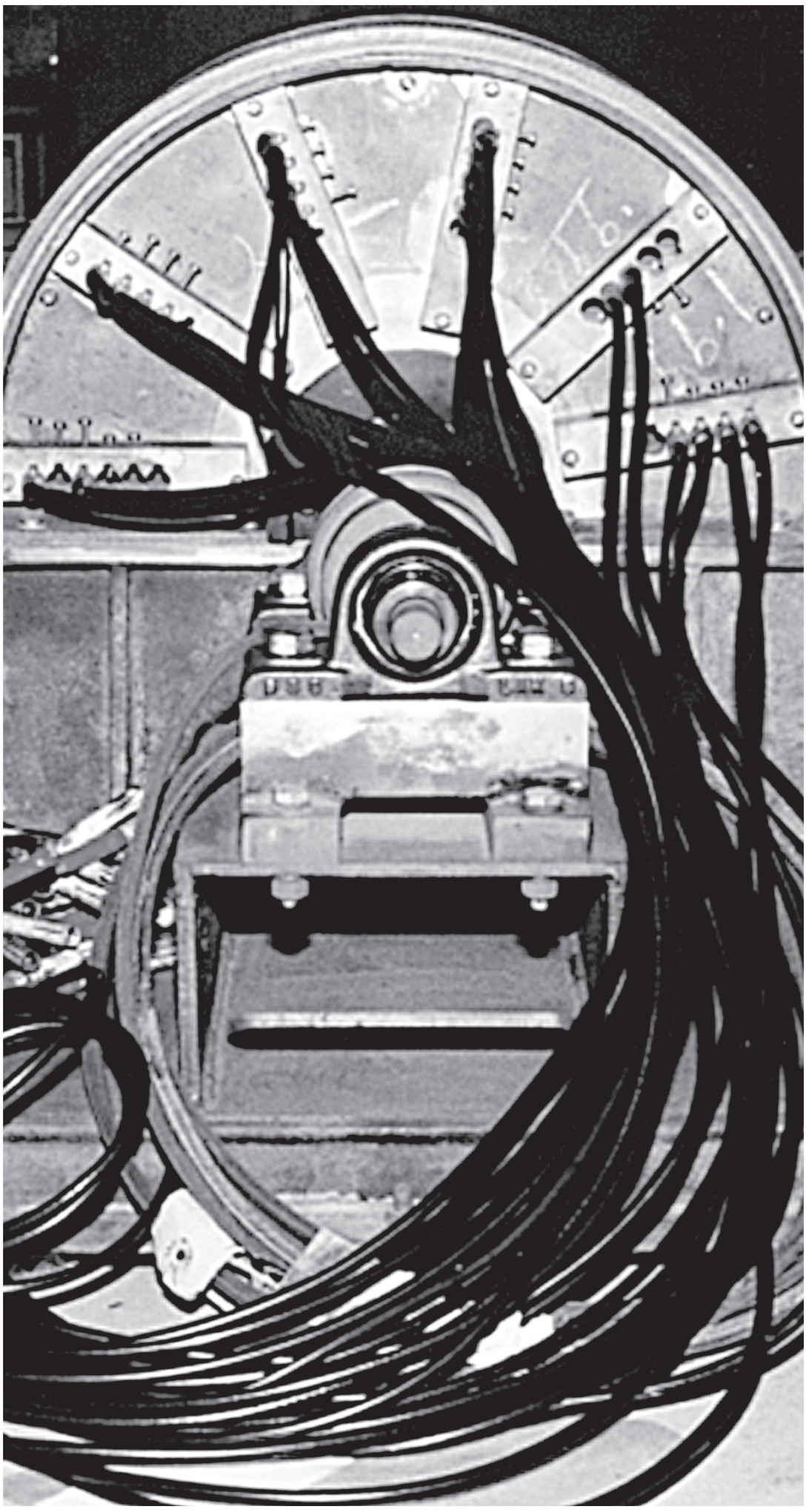
This lack of professional musical input to the musical work, consistently the case for CSIRAC from 1949 to 1964, is in stark contrast to the approach undertaken at Bell Labs. Max Matthews and John Pierce were both scientists but they quickly recognised the need to involve musicians in their computer 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 have 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. It had been shown that CSIRAC could be 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 anyone would want to use a computer to generate music and what the application of technology means for the aesthetics of music, would have been addressed. Even if there was no 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 USA became public.

It appears that the musical programming of CSIRAC, both in Sydney and in Melbourne, was not as appreciated as it could have been and thus not as well promoted as it deserved to be. With some more enthusiasm or imagination from mostly some administrators, but also the public, or with

some joint research and development involving scientists and musicians, composers could easily have become involved with CSIRAC and there could have been some more exciting musical developments, with a more enduring musical contribution.

As it stands, there is no enduring legacy of the music made with CSIRAC. It is notable because of the leap of imagination that was 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 required to produce the music. Overcoming the many technical problems along the way to that goal showed the ingenuity and creativity of those who took these first steps. CSIRAC always suffered from being isolated. It was politically isolated within the CSIR with few other departments knowing much of it or using it, which helped lead to the project being cancelled. It was also culturally isolated, even though its service was run as an 'open shop' in Melbourne, achieving comparatively little publicity. It was probably the isolation that was responsible for composers, such as Grainger, not becoming involved with the computer and using it to create significant musical developments. The isolation was also most probably responsible for the music of CSIRAC being lost until now, so that it is a blip in Australian musical history and the history of computer music. The music is, however, an interesting and very early blip.

The application of computers to the arts in general, not just the auditory or musical arts, continues to be one of the greatest challenges confronting artists, the public and computer scientists.



Appendix A: Newspaper reports

CSIRAC occasionally made news headlines. Below are the two articles that mention the music. They show a different attitude to computers than is common today. It was common at the time to refer to computers as 'electronic brains' and attribute aspects of real intelligence to them. These photographs also show how CSIRAC was operated with a person sitting at the console.

Professor Teaches Electronic Brain to Hum Bathroom Ditty and Play Games

By a Staff Correspondent.

CSIRAC — the University's giant electronic brain — has LEARNED TO SING!

Scientists have re-christened it "the brain with the beat," as it hums, in bathroom style, the lively ditty, Lucy Long.

CSIRAC's song is the result of several days' mathematical and musical gymnastics by Professor T. M. Cherry.

In his spare time Professor Cherry conceived a complicated punched-paper programme for the computer, enabling it to hum sweet melodies through its speaker.

"CSIRAC was really no more made for music than a merry-go-round was made for mathematics," Professor Cherry says.

"But the brain makes a better job of music than a merry-go-round would make of maths."

Speaking Hopes

A bigger computer, Professor Cherry says, could be programmed in sound-pulse patterns to speak with a human voice.

"But it would be a laborious process, serving little apparent purpose," he adds.

Plays Games

CSIRAC (when not engaged in serious computation) plays games with the massive cunning of a two-ton child.

Mr. G. W. Hill, who helps tend the brain in the University's physics department, invited me to match wits with the computer in a match game.

Ten matches were placed on a table.

CSIRAC reproduced them as points of eerie blue light on one of its tiny TV screens.

The winner would be he/it who picked up the last match.

My match moves were relayed to the brain via a signal switch.

In reply, the brain blotted up with darkness several blue light points on its screen.

We played many match games—but the brain, with devilish scientific cunning—won every time.

Then Mr. Hill showed me how to beat the brain.

He threw a switch—and a second TV screen lit up.

This screen showed, in pin-pricks of light, the advance moves CSIRAC was planning.

By peeping, unfairly, into the brain's brain, we won every game.

Despite its songs and games, CSIRAC has been taught to reject silly trivialities.

Ask it a riddle, such as "When is a door not a door?" and it will coldly reply (on punched paper): INSUFFICIENT DATA AVAILABLE.

The brain refuses, too, to be bothered with piffling tax problems.

Asked to calculate 5 per cent. of £1578, it replied: STORAGE CAPACITY EXCEEDED — although there were no other problems going through at the time.

Mr. Hill explained: "The brain isn't human enough to make excuses.

"But it simply can't distinguish between an utterly trivial question (which it regards as zero) and infinite overload."

CSIRAC is a polite brain. But, says, Mr. Hill, it could be taught to be rude to careless or trivial questioners.

Some brains have been programmed to reply: "WHY THE DEVIL DON'T YOU TAKE YOUR SENSELESS QUESTIONS ELSEWHERE?"

DR. HURST (nearest camera) and Mr. Hill at the control board of the computor in the Physics Department at the University.

From The Age, *Wednesday 27 July 1960.*

CSIRAC – the University's giant electronic brain – has LEARNED TO SING!

Scientists have re-christened it 'the brain with the beat,' as it hums, in bathroom style, the lively ditty, 'Lucy Long.'

CSIRAC's song is the result of several days' mathematical and musical gymnastics by Professor T. M. Cherry. In his spare time Professor Cherry conceived a complicated punched-paper programme for the computer, enabling it to hum sweet melodies through its speaker. 'CSIRAC was really no more made for music than a merry-go-round was

made for mathematics,' Professor Cherry says. 'But the brain makes a better job of music than a merry-go-round makes of maths.'

Speaking Hopes

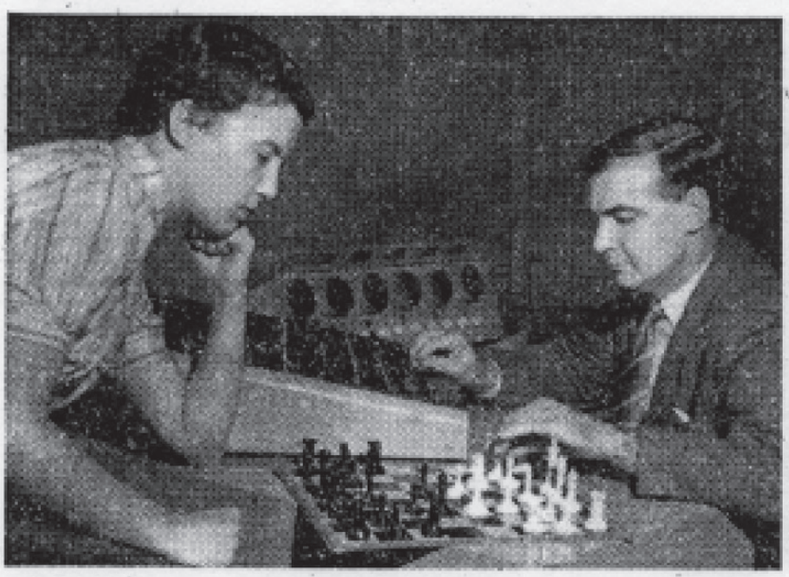
A bigger computer, Professor Cherry says, could be programmed in sound-pulse patterns to speak with a human voice. 'But it would be a laborious process, serving little apparent purpose,' he adds.

Plays Games

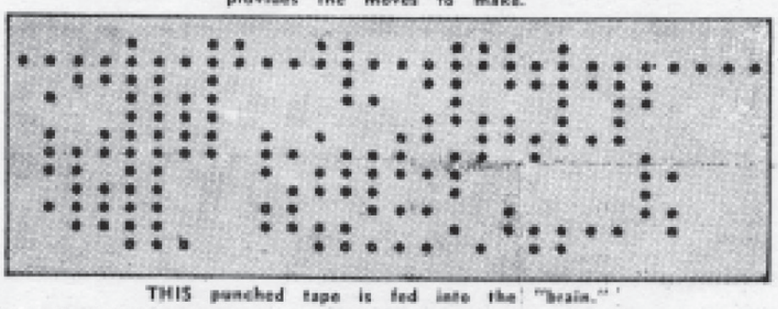
CSIRAC (when not engaged in serious computation) plays games with the cunning of a two-ton child. Mr G. W. Hill, who helps tend the brain in the university's physics department, invited me to match wits with the computer in a match game. Ten matches were placed on a table. CSIRAC reproduced them as points of eerie blue light on one of its tiny TV screens. The winner would be he/it who picked up the last match. My match moves were relayed to the brain via a signal switch. In reply, the brain blotted up with darkness several blue light points on its screen. We played many match games – but the brain, with devilish scientific cunning – won every time. Then Mr. Hill showed me how to beat the brain. He threw a switch – and a second TV screen lit up. The screen showed, in pin-pricks of light, the advance moves CSIRAC was planning. By peeping, unfairly, into the brain's brain, we won every game. Despite its songs and games, CSIRAC has been taught to reject silly trivialities …

(Opposite page) From 2/3 down, … When CSIRAC began sporting its musical gifts, we jumped on his first intellectual flaw. When he played "Gaudeamus Igitur," the university anthem, it sounded like a refrigerator defrosting in tune. But then, as Professor Cherry said yesterday, "This machine plays better music than a Wurlitzer can calculate a mathematical problem…"

Phew! This brain's a snob!



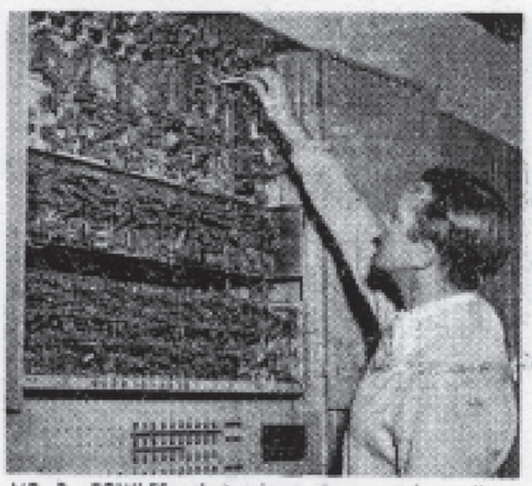
REPORTER v. "BRAIN." . . . Kerry Pearce, Herald staff reporter, puts her wits against the electronic brain in a game of chess at the University today. Dr. Frank Hirst puts her moves into the machine which is punched on a special tape, and the machine provides the moves to make.



THIS punched tape is fed into the "brain."

W.	B.
P-Q4	N-KB3
P-QB4	P-KN3
N-QB3	B-N2
P-K4	P-Q3
P-KW3	0-0
B-N2	

THE "BRAIN" gives Dr. Hirst the moves to make.



MR R. BOWLES, electronic engineer, makes adjustments to the machine during the one-hour "warming-up" period.

By KERRY PEARCE, Herald Staff Reporter

THIS electronic brain is a real intellectual snob. It won't talk in the vulgar tongue. And you're beaten before you start when it comes to chess.

You'd think Csirac, as they call him, would be glad of a friendly game to take his mind off mathematical formulae. Not him.

He has fitted in his memory every possible move for chessmen in every position. **AND HE MOVES THE BEST ONE.**

I paid court to Csirac at Melbourne University today, bearing such questions as *What chance have I got of winning Tatts?* or *Will the price of potatoes go up?*

It didn't work. If you can't talk about tangents of angles and polynomials Csirac is just not interested.

HIS ANGLE

For instance the first question University officials asked Csirac (so named by the Commonwealth Scientific and Industrial Research Organisation who made him) was: *Find out the tangents of angles in steps of .01 of a right angle.*

It took the staff the day to frame this question in a language acceptable to Csirac. He answered it nonchalantly in 1¼ minutes. You could almost swear he yawned.

Csirac has two inbuilt memories. A fast one, with 1024 words of information, and a slow one, with 4096 words.

He measures 19 ft. x 7 ft. x 5 ft.

When Csirac began sporting his musical gifts we jumped on his first intellectual flaw. When he played "Gaudeamus Igitur," the university anthem, it sounded like a refrigerator defrosting in tune.

But then, as Professor Cherry said yesterday: "This machine plays better music than a Wurlitzer can calculate a mathematical problem."

If you do have a really big intelligent industrial problem, Csirac will help you.

Dr. Frank Hirst, officer in charge of running the electronic brain, said Csirac's talent was open to anyone who wanted a big mathematical programme worked out.

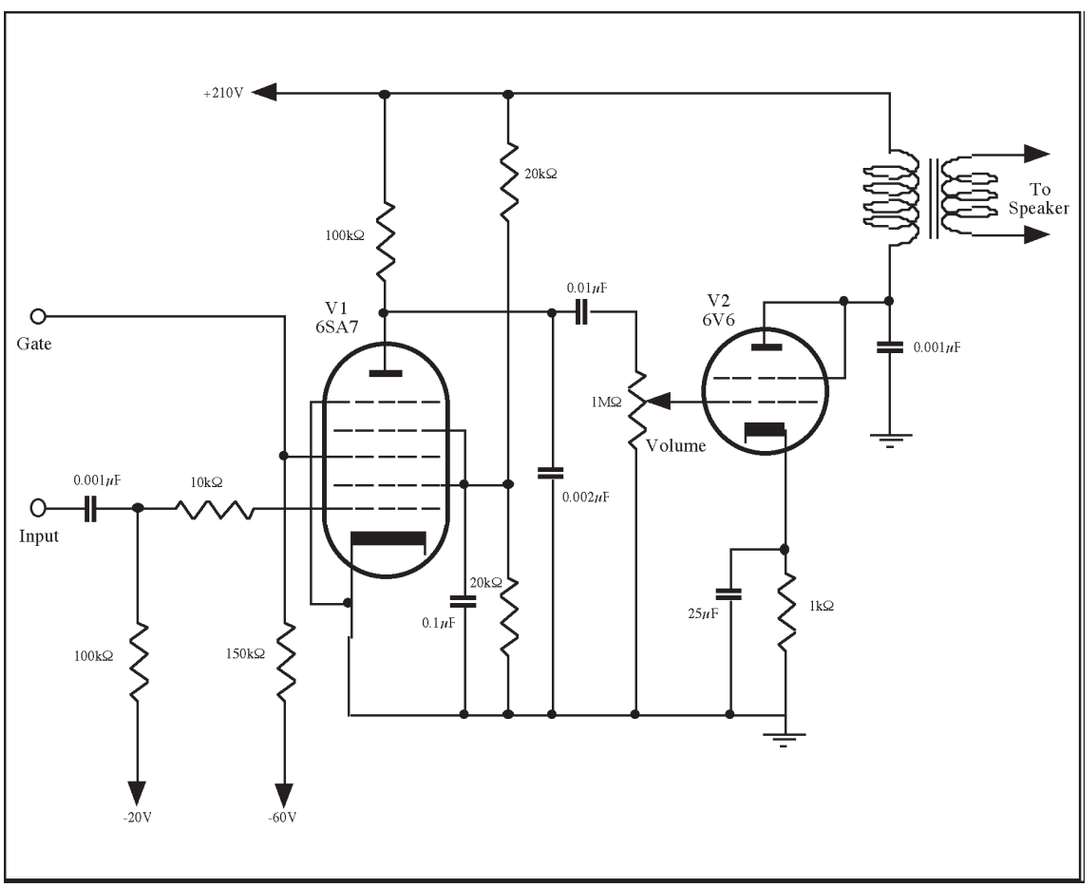
FOOTNOTE: A tangent of an angle: In a right angle triangle if you want the tangent of the angle you have to take the ratio of the opposite side over the adjacent side.

A polynomial is a many-termed expression, e.g. A plus Bx plus Cx squared plus Dx cubed.

From The Herald, *Friday 15 June 1956. The chess game was staged for the newspaper reporter.*

Appendix B: Pulse curves and circuits

The following tables and graphs show details of the pulses generated by the reconstructed CSIRAC hardware. The values are the 16-bit signed data values for each sample. The output was sampled directly to DAT running at 44,100Hz, taken from across the speaker terminals of the valve output amplifier. There is a previous oscilloscope screen shot of the pulses for comparison. The schematic diagram of the output amplifier is below. One point that may require explanation is the 'gate' input. Each hardware source and destination had a gate line to enable it to receive or send data to and from the bus. This was permanently enabled in the reconstructed amplifier.



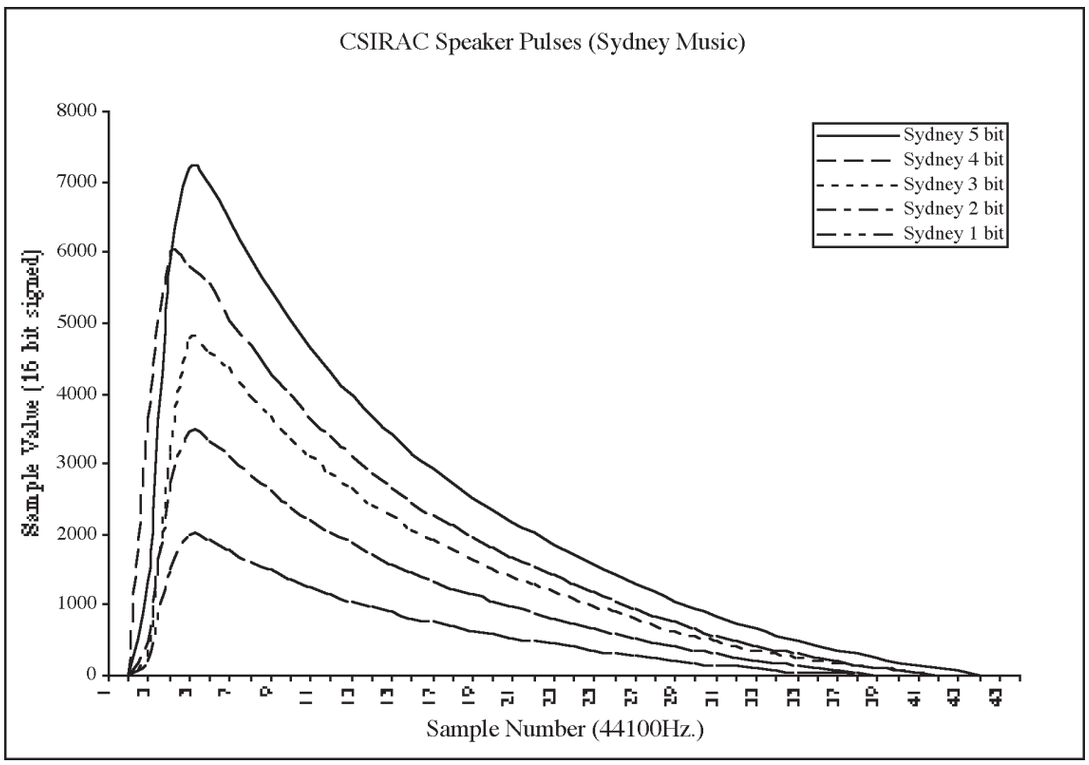
Schematic diagram of the CSIRAC *'hooter' circuit.*

Sydney 5-bit	Sydney 4-bit	Sydney 3-bit	Sydney 2-bit	Sydney 1-bit
0	0	0	0	0
1323.919	3641.778	288.982	460.972	208.987
5864.642	5993.634	3192.805	2729.833	1516.907
7212.560	5800.646	4776.708	3457.789	1980.879
6936.577	5569.660	4589.720	3319.797	1905.884
6468.605	5020.693	4374.733	3109.810	1785.891
5907.640	4684.714	3954.759	2825.828	1623.901
5465.667	4262.740	3676.776	2612.841	1495.909
5020.693	3965.758	3348.796	2389.854	1371.916
4645.716	3633.778	3107.810	2205.865	1260.923
4294.738	3379.794	2848.826	2022.876	1153.930
3977.757	3106.810	2641.839	1869.886	1061.935
3680.775	2881.824	2425.852	1718.895	971.941
3411.792	2657.838	2253.862	1583.903	896.945
3159.807	2462.850	2069.874	1455.911	819.950
2930.821	2274.861	1914.883	1343.918	756.954
2713.834	2113.871	1767.892	1237.924	693.958
2525.846	1953.881	1636.900	1141.930	636.961
2347.857	1812.889	1513.908	1052.936	584.964
2177.867	1669.898	1397.915	968.941	535.967
2019.877	1539.906	1286.921	886.946	489.970
1863.886	1414.914	1182.928	811.950	445.973
1716.895	1291.921	1084.934	734.955	401.975
1575.904	1173.928	989.940	663.959	364.978
1435.912	1062.935	895.945	597.964	321.98
1297.921	951.942	806.951	533.967	283.983
1173.928	851.948	715.956	468.971	246.985
1054.936	752.954	633.961	403.975	209.987
936.943	658.960	561.966	352.978	180.989
834.949	575.965	484.970	302.982	149.991
737.955	497.970	422.974	254.984	123.992
646.961	430.974	362.978	210.987	97.994
565.965	363.978	303.981	169.990	78.995
493.970	302.982	255.984	128.992	50.997
424.974	249.985	209.987	97.994	32.998
354.978	198.988	167.990	67.996	17.999
299.982	154.991	123.992	39.998	4.000
247.985	111.993	91.994	15.999	2.144
191.988	70.996	59.996	8.999	0
144.991	27.998	23.999	3.783	0
100.994	11.999	5.932	1.347	0
57.996	3.062	2.847	0	0
12.999	1.031	1.025	0	0
2.041	0	0	0	0
0	0	0	0	0

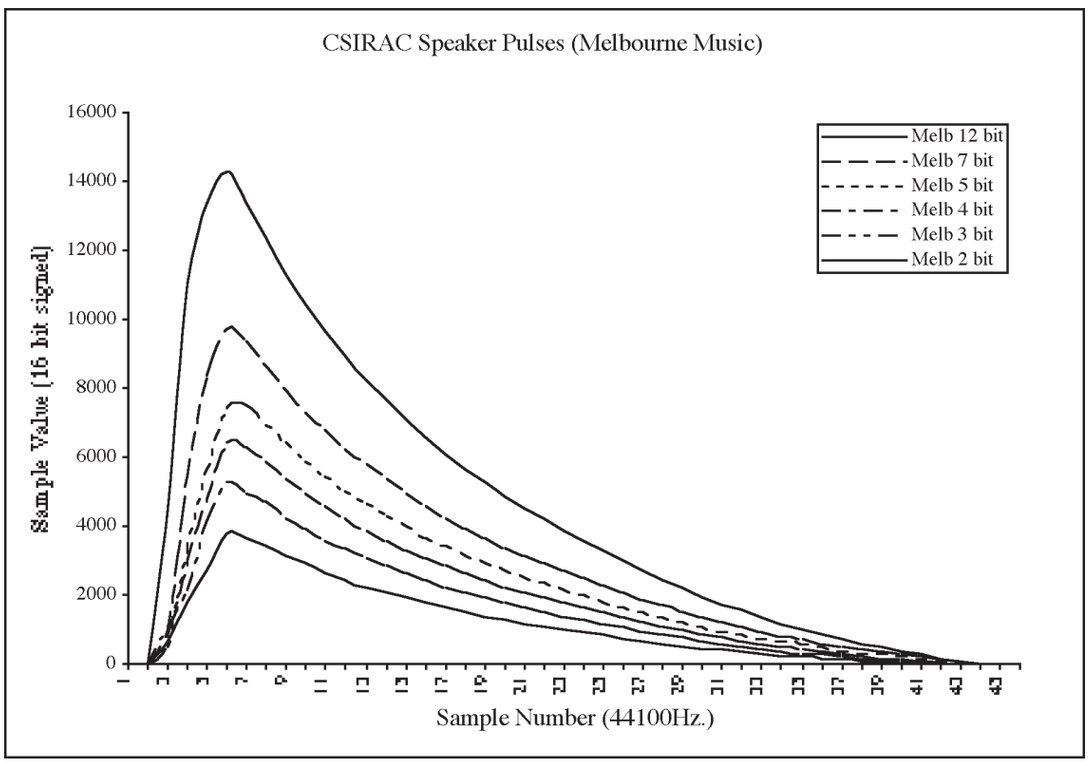
Sample values for the pulses reconstructed for the Sydney music.

Melb 12-bit	Melb 7-bit	Melb 5-bit	Melb 4-bit	Melb 3-bit	Melb 2-bit
0	0	0	0	0	0
4516.724	1008.938	515.969	850.948	1041.936	613.963
11021.327	5434.668	3356.795	2965.819	2172.867	1797.890
13372.184	8382.488	5649.655	4779.708	4118.749	2722.834
14314.126	9721.406	7398.548	6423.608	5269.678	3751.771
13382.184	9360.429	7516.541	6255.618	4945.698	3629.778
12380.244	8653.472	6948.576	5874.642	4696.713	3438.790
11321.309	7963.514	6448.606	5376.672	4234.742	3130.809
10448.362	7313.554	5889.641	4959.697	3956.759	2897.823
9623.412	6755.588	5457.667	4552.722	3601.780	2647.838
8912.456	6227.620	5016.694	4211.743	3354.795	2454.850
8234.497	5764.648	4654.716	3885.763	3071.812	2245.863
7634.534	5329.675	4287.738	3590.781	2859.825	2078.873
7073.568	4931.699	3974.757	3314.798	2620.840	1903.884
6557.600	4569.721	3673.776	3063.813	2439.851	1761.892
6082.629	4231.742	3408.792	2832.827	2240.863	1622.901
5654.655	3919.761	3152.808	2623.840	2077.873	1502.908
5252.679	3641.778	2927.821	2429.852	1916.883	1375.916
4884.702	3377.794	2711.834	2244.863	1778.891	1277.922
4528.724	3132.809	2515.846	2077.873	1642.900	1170.928
4196.744	2903.823	2322.858	1924.883	1519.907	1079.934
3879.763	2682.836	2151.869	1764.892	1389.915	992.939
3576.782	2470.849	1976.879	1622.901	1280.922	904.945
3279.800	2268.862	1810.889	1481.910	1169.929	825.950
2992.817	2071.874	1660.899	1349.918	1062.935	741.955
2718.834	1878.885	1508.908	1222.925	954.942	663.959
2455.850	1693.897	1362.917	1095.933	856.948	589.964
2201.866	1522.907	1221.925	979.940	760.954	520.968
1962.880	1354.917	1088.934	860.947	674.959	453.972
1741.894	1198.927	962.941	760.954	590.964	396.976
1537.906	1059.935	843.948	663.959	512.969	333.980
1342.918	927.943	736.955	572.965	439.973	282.983
1173.928	809.951	646.961	493.970	378.977	234.986
1013.938	694.958	549.966	409.975	313.981	186.989
864.947	592.964	467.971	345.979	259.984	147.991
730.955	498.970	386.976	276.983	208.987	110.993
605.963	410.975	319.980	219.987	161.990	80.995
491.970	331.980	254.984	165.990	117.993	46.997
382.977	259.984	196.988	113.993	80.995	16.999
273.983	186.989	139.991	67.996	40.997	2.999
172.989	116.993	83.995	19.999	14.624	1.050
74.995	53.997	31.998	10.999	4.074	0
16.999	17.021	15.999	3.650	1.515	0
4.732	4.010	3.970	1.530	0	0
0	0	0	0	0	0

Sample values for the pulses reconstructed for the Melbourne music.



Graph of the pulse sample values as used for the Sydney music.



Graph of the pulse sample values as used for the Melbourne music.

Appendix C: CD Contents

The CD contains two parts, an audio part and a data part, titled CD Extras. The audio CD can be played in a normal CD player and the track listing is below. However, if it is placed into a computer, there in a movie and a slide show of images that can be viewed.

The audio part contains all of the musical material found on the music tapes, as described in the text. The tracks are:

Track	Title	Duration
01	Sydney Scale	00:07
02	Colonel Bogey	00:17
03	Bonnie Banks	00:18
04	Girl with Flaxen Hair	00:41
05	Auld Lang Syne	00:18
06	Chopin March	00:30
07	Thanks fro the Memories	00:16
08	Saul (from Handel)	00:40
09	Saul with Blurts (Geoff Hill's humour)	00:26
10	So Early in the Morning	00:22
11	Scale 1	00:23
12	Scale 2	00:41
13	In Cellar Cool	00:39
14	Lucy Long	05:04
15	Lucy Long alternate var 3	00:27
16	In Cellar Cool with CSIRAC environment	01:00

Track 16 is an example of the music with an accurate simulation of the environmental noise that CSIRAC made or had around it, based on recollections of users and technical details of the moving parts, noises of such equipment and the air conditioning. This is of course much like it was heard in the time CSIRAC was operating, and implies a level of masking of the sounds from the speaker.

The CD Extras part contains two components: a slide show of CSIRAC images named 'csirac-slides.mp4' and a video clip of an interview with Trevor Pearcey titled 'csirac-interview.mp4'. Both of these files are playable by most computers. The slide show runs for approximately six minutes and the interview runs for five minutes. The slides are many high resolution photographs of CSIRAC and the personnel involved in the project. The interview is a video clip of a much larger interview with Trevor Pearcey–the selection is when he discusses the music.

Sources

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 a Tuesday with John Spencer, Jurij Semkiw, Ron Bowles, Peter Thorne, Doug McCann and others.

- *Interview with Trevor Pearcey, recorded on 1st October 1996, interview conducted by Steven Pass, Doug McCann and Peter Thorne.*
- *Interview with Reginald Ryan, conducted by the author, 25 January 2000.*
- *Interview with Terry Holden, recorded 7 June 1997 by Steven Pass.*
- *Interview with Kay Thorne, conducted by the author, 21 February 2000.*
- *Interview with Peter Thorne, conducted by the author, 16 February 2000.*
- *Interview with Ron Bowles, conducted by the author, 25 February 2000.*
- *Interview with Eileen Hill, conducted by the author, 2 March 2000.*
- *Interview with Douglas McCann, conducted by the author, 15th May 2000.*
- *Interview with Dick McGee, conducted by the author, 20 May 2000.*
- *Private correspondence with Chris P. Burton, Computer Conservation Society.*

Bird, J. 1999. “Percy Grainger” Sydney, Australia: Currency Press.
Cage, J. 1967. “Silence” Cambridge, Massachusetts, USA: The MIT Press
Chadabe, J. 1997. “Electric Sound the Past and Promise of Electronic Music” New Jersey, USA: Prentice-Hall, Inc.
CSIRAC Archive, The University of Melbourne Computer Science Department.
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, USA: Schirmer Books, Macmillan Publishing Co., Inc.
Hill, G. W. 1954. “Programming for High-Speed Computers” Sydney, Australia: MSc thesis, The University of Sydney.
Hiller, L. A. and Isaacson, L. M. 1959. “Experimental Music” New York, USA: McGraw-Hill Book Company, Inc.
Lehr, A. 1991. “Beiaard Kunst in de Lange Landen” Belgium: Lannoo n.v., Tielt.
Manning, P. 1993. “Electronic and Computer Music” Oxford, England: Oxford University Press.
McCann, D. and Thorne, P. 2000. “The Last of the First: CSIRAC: Australia’s First Computer” Melbourne, Victoria, Australia: Melbourne University Computer Science.
Pierce, J. 1995. “Recollections by John Pierce” Part of “The Historical CD of Digital Sound Synthesis. Computer Music Currents 13” Mainz, Germany: Schott Wergo Music Media GmbH.
Roads, C. 1980. “Interview with Max Matthews” Computer Music Journal 4(4). Reprinted in C. Roads, ed. 1989. “The Music Machine” Cambridge, Massachusetts, USA: The MIT Press
Roads, C. 1996. “The Computer Music Tutorial” Cambridge, Massachusetts, USA: The MIT Press
Williams, M. R. 1997. “A History of Computing Technology” California, USA: IEEE Computer Society Press.

There are many internet resources for computer music, electronic music, CSIRAC and historical computing, here are a few:

http://www.computermusic.org

http://www.emf.org

http://www.emf.org/guidetotheworld/internetdirectory.html

http://darkwing.uoregon.edu/~dlo/list.html

http://www.cs.mu.oz.au/csirac/

http://www.cs.mu.oz.au/csirac/music

http://www.asap.unimelb.edu.au/pubs/articles/asa97/csirac.htm

http://archive.comlab.ox.ac.uk/other/museums/computing.html

http://www.iinet.net.au/~daveb/history.html

http://columbia.digiweb.com/~hansp/ccc/cclist1.htm

http://www.cs.man.ac.uk/ccs

http://www.computer.org/history

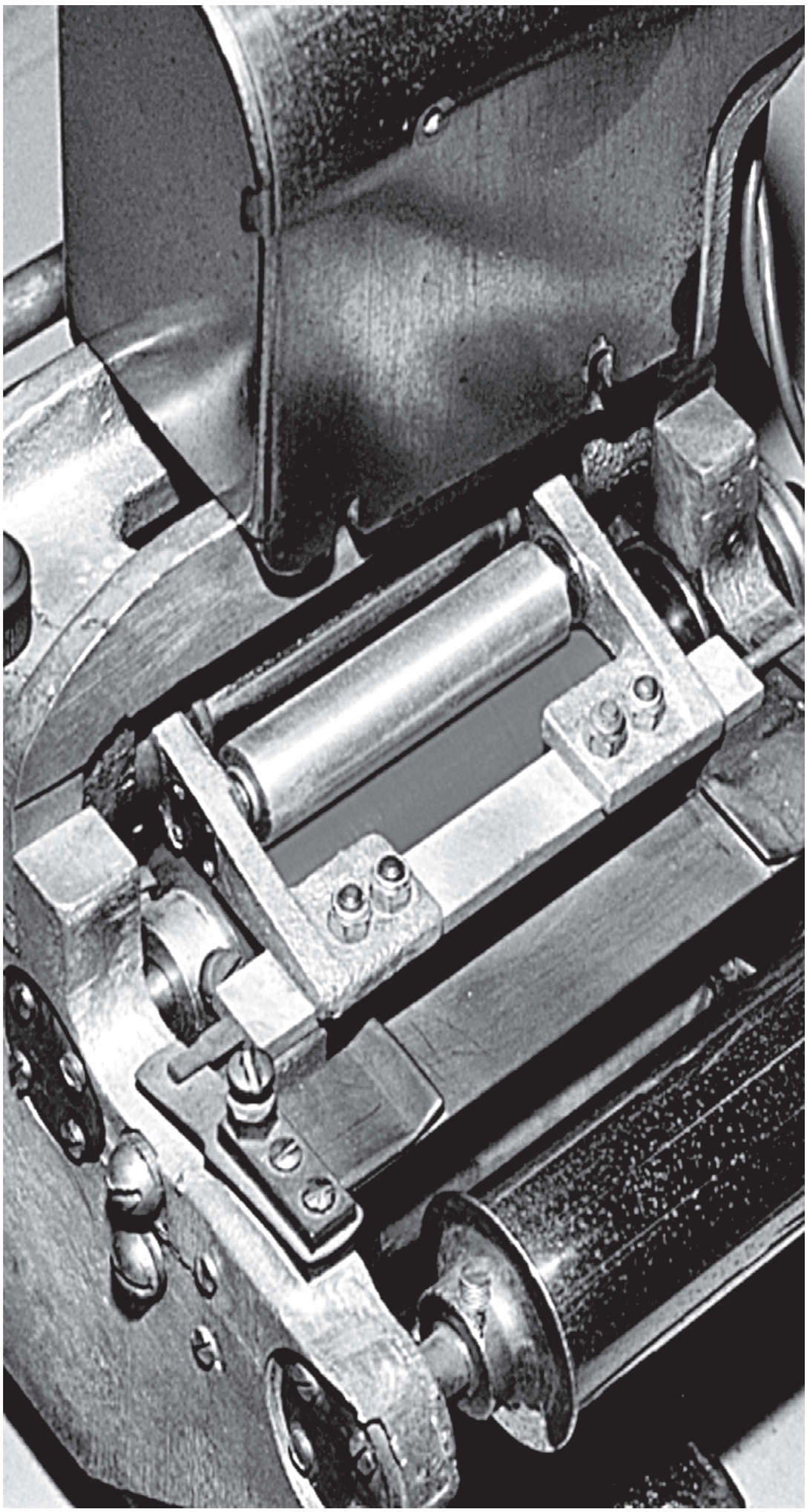
http://archive.comlab.ox.ac.uk/other/museums/computing.html

http://www.davros.org/misc/chronology.html

http://www.cs.virginia.edu/brochure/museum.html

Abbreviations

ACE	Automatic Computing Engine
ASCC	Automatic Sequence Controlled Calculator
CRT	Cathode Ray Tube
CSIR	Council For Scientific and Industrial Research
CSIRAC	Council For Scientific and Industrial Research Automatic Computer
CSIRO	Commonwealth Scientific and Industrial Research Organisation
EDSAC	Electronic Delay Storage Automatic Computer
EDVAC	Electronic Discrete Variable Automatic Computer
ENIAC	Electronic Numerical Integrator and Calculator
MADM	Manchester Automatic Digital Machine, also Manchester Mark I, also the production version – the Ferranti Mark I
SEAC	Standard Eastern Automatic Computer
SSEC	Selective Sequence Electronic Calculator
SSEM	Small Scale Experimental Machine
TREAC	Telecommunications Research Establishment Automatic Computer



Index

The Music of CSIRAC charts a new history of computer music starting in Australia in the early 1950s. It brings to life the great spirit of invention and imagination shown by the engineers and programmers who made it possible for a machine to make music. Complete with a CD containing a rigorous reconstruction of the first music played by a computer, this book provides an accessible and cogent overview of the origins of a now pervasive social and technical transformation in the way music is mediated.

Paul Doornbusch is an expert in the field of computer music and has worked as a composer and sonologist throughout Europe and Australasia.

COMMON GROUND

museum VICTORIA

Pearcey FOUNDATION

ISBN 186335569-3

9 781863 355698