

# Computer Music (So Far)

## Part I. Mainframes

Computers have made noises since Eniac was rolled out in 1947. It seems one of the problems with programming the early systems was knowing that everything was operating properly. That's why you see computers in 50s science fiction movies covered in lights. A common trick was to connect a loudspeaker to a key component in the main processor -- this would give tones of various pitches as data moved in and out of the accumulator. As long as a program was running, the tones would change in a random sort of way. A steady pitch meant the program was caught in a loop and needed to be shut down. That's how beedle beep became the signifier of computer operation.

The first real music program was written by Max Mathews in the early 1960s. He was working at Bell labs, a research center run by AT&T when they were the phone company. Max was primarily developing more practical things for the phone company (he invented the little square plug, for instance). But he worked on music software in his spare time. He called his software MUSIC, with the different versions indicated by Roman numerals. MUSIC made its first sound in 1957, playing single line tunes. MUSIC II, a year later, had four part polyphony. These ran on the most powerful computer of the day, and took something like an hour of computing time to generate a minute of music. The sound was similar to the tunes played by some wristwatches today.

In 1960, MUSIC III introduced the concept of a "unit generator", a subroutine that would create a specific kind of sound and only needed a few numbers from the composer. This simplified the process a great deal, and opened everything up to more composers. MUSIC IV and V added refinements and improved efficiency. Composers such as James Tenny, F. B. Moore, Jean Claude Risset , and Charles Dodge came to the labs and began creating serious works. Some were hired as assistants, some just seemed to be around a lot.

These people, who were mostly recent university graduates, moved on to permanent academic jobs and took the program with them. By 1970, computer music was happening at about a dozen schools, and by the end of the decade, there were probably a hundred universities and research centers exploring computer composition and related work.

A typical music research center of the day was built around a large mainframe computer. These were shared systems, with up to a dozen composers and researchers logged on at a time. (Small operations would share a computer with other departments, and the composers probably had to work in the middle of the night to get adequate machine time.) Creation of a piece was a lot of work. After days of text entry, the composer would ask the machine to compile the sounds onto computer tape or a hard drive. This might take hours. Then the file would be read off the computer tape through a digital to analog converter to recording tape. Only then would the composer hear the fruits of his labor.

To tell the truth, research papers were a more common product than music during this period, as the people involved were really just figuring things out. Most of the papers were published in *The Journal of the Audio Engineering Society* or the *Computer Music Journal*. Work was also shared at International Computer Music Conferences, which began in 1974. These conferences are put on by an organization known as the International Computer Music Association.

Research at these centers was aimed at producing hardware as well as software. The goal was to develop a system that could do serious synthesis in real time. This produced many unique instruments, some of which eventually became commercial products. The Centers also created a lot of new software. They all started with Mathew's MUSIC, but developed it in their own way to suit their needs. Thus you saw MUSIC 360, MUSIC 4BF, MUSIC 11, (variants for different computers) as well as different approaches such as Sawdust, and Chant.

The major centers of the 70's included M.I.T., The University of Illinois at Champaign-Urbana, The University of California at San Diego, The Center for Computer Research in Music and Acoustics (CCRMA, usually pronounced "karma") at Stanford University, and the Institut de Recherche et Coordination Acoustic/Music in Paris (IRCAM). People tended to move around from one to the other for a few years before permanently joining staff or founding new programs somewhere.

Some names turn up a lot in the literature from that period:

Lejaren Hillier was one of the first to use a computer in music, and is generally credited with the earliest published work, *Illiac suite* from 1958. This was composed by computer, but is performed by musicians.

Jean Claude Risset worked at Bell Labs from 1964 to 1969 under sponsorship of the French government and became a founder of IRCAM, later moving to Marseilles. He is best known for his analysis of traditional musical timbres, which became the basis for many research papers that followed his methods and many compositions.

John Chowning was the founding director of CCRMA. He is most famous for the development of frequency modulation as a synthesis technique, which led to the Yamaha DX synthesizers (and which generated significant patent income for CCRMA).

Curtis Roads, teacher at MIT and UC Santa Barbara, is best known for writing the book on computer music. Literally. It's called the Computer Music Tutorial and was circulated informally for years before he completed the published edition. He was the editor of the Computer Music Journal for over a decade.

F. Richard Moore worked with Mathews at Bell labs, then developed major hardware for CCRMA before founding the computer music program at UC San Diego.

John Snell was the founding editor of the Computer Music Journal. In the days before the internet, a good journal was essential to the health of the discipline.

John Strawn is a well known developer of synthesis and DSP software. He was associated with CMJ in the early years and edited many of the foundation texts of the discipline.

William Buxton designed musician-computer interfaces at the University of Toronto before moving into more general computer interface work.

## **Part II. Hybrid Systems**

Most composers of the 60s and 70s were not involved in computer work. There were various reasons for this, the limited access to computers (particularly for students), the difficulty of programming compared to composition, and lack of interest. To be honest, the audible product of the research labs was not very attractive compared to what was happening in other parts of electronic music. The analog synthesizer was much faster in giving gratification.

In 1968 Max Mathews designed a system that allowed a computer to control an analog synthesizer in real time. The GROOVE program would record a musician's actions as he played a synthesizer with a variety of controls. After the recording was made, the

action list could be edited and played on the synthesizer again. Emmanuel Ghent, F. R. Moore, Laurie Spiegel and others used the machine and expanded its capabilities. Other mainframe/analog hybrids were built, especially when so called minicomputers such as the PDP-11 came on the market. One example is PLAY, which Joel Chadabe had running at NYU Albany in 1977.

A big area of development in analog synthesizers of the early 70's was in building larger and larger sequencers. These were arrays of knobs that would be activated one at a time, producing a cyclical pattern of voltages. As it happens, the underlying circuitry of a sequencer is essentially digital. Pursuit of state of the art sequencers in the 70s led designers to investigate the hot new integrated circuits known as microprocessors. One of the very early ones was available with a prototyping and development board known as the KIM. This board was designed to fit into a ring binder (it had holes and everything) it would execute a 256 step program that you burned into a programmable read only memory chip (PROM) one location at a time. The most attractive feature of the KIM was its price. You could get one for about \$200 when the least expensive computer cost over \$8000.

Many composers connected KIMs to analog synthesizers. Robert Ashley's system was one of the most successful, as he worked out a system for detecting the pitch of acoustic instruments and using that to control his "Melody driven electronics".

Of course the microprocessor soon led to the first home computers, such as the apple II. It was even easier to connect one of these to a synthesizer (a schematic was published in CMJ in 1978) and programming in BASIC was much easier than burning PROMs. This brought a form of computer music to many musicians that were not in the network of research institutions. This included private individuals, artists co-ops and small schools which were focused primarily on producing compositions and performances. There must have been over a thousand hybrid systems built between 1978 and 1984. Most of these were home made, although Donald Buchla sold a complete system in the model 400 music box.

At the same time synthesizers were becoming less and less analog. Digital sequencers and digital keyboards were the norm, and digital oscillators offered better stability than the analog type. A microprocessor of the day could be programmed to produce 16 voices of somewhat limited tone colors, and this could be built on a circuit board that would fit into a slot on an Apple II. These boards were mostly built by hobbyists, but commercial versions were sold by companies like Mountain Hardware. The Alpha

Centauri was based on such a board and also included a piano style keyboard and sequencing software.

When the Commodore 64 was introduced in 1982, it included a four voice synthesizer on a single chip within the computer. This was apparently intended for games, but software soon appeared that used it for music composition.

In 1981 a consortium of musical instrument manufacturers began talks that led to the MIDI standard in 1983. This made it possible to connect practically any computer to practically any synthesizer. Since then the music stores have become stuffed with MIDI devices of all sorts, and the hybrid system is the norm.

Probably the most advanced hybrid at this time is the Kyma system by Symbolic Sound, which is based on a very powerful DSP unit (the Capybara) attached to a host computer running Kyma software. The program is basically an instrument designer that allows the composer to configure the Capybara for any kind of synthesis or processing.

### **Part III. Synthesis on the Desktop**

The coming of MIDI had little effect on the computer music research institutions. It was primarily used where quick and simple connections were needed. MIDI gave a tremendous boost to commercial sales of synthesizers, and some of the manufacturers began to fund research programs at places like CCRMA. This created a pipeline from the lab to the musician on the street.

The bulk of research was still focused on computer applications. (Most MIDI instruments are simplified computers running a single program.) The mainframes began to disappear as engineering workstations and finally ordinary personal computers exceeded the capabilities of the old machines, but the work was (and is) much the same, development of better sounding, more efficient and easier to control methods of synthesis and signal processing.

In 1985, Barry Vercoe of MIT made a version of MUSIC that could be compiled for any computer that supported the C programming language. His program, which went quite a bit beyond MUSIC in ability (incorporating ten years of additional research) would run nicely on advanced desktop machines of the day. It wasn't very long before Csound was available on Macintosh and PC type computers and is now the standard synthesis language nearly everywhere. You no longer need a lab for computer music, just a computer good enough to run the hot new games.

Csound still smacks of the computer music lab in application and design (unit generators and such) but there is now a class of programs that are nearly as powerful and wrapped in an interface that makes them accessible to composers with any level of computer finesse. These range from simple applications that emulate popular hardware of the past (complete with a screen display that is a picture of the original) through Reaktor, which allows the design of virtual instruments with a modular flavor, to Max/MSP which realizes graphical flowcharts of complex logical and DSP systems. Such things are available as stand alone applications or as "plug-ins" to comprehensive sequencing and recording programs.

Source: [http://www.co-bw.com/Audio\\_history.htm](http://www.co-bw.com/Audio_history.htm)