A2.8

IT'S GNOT SIGNAL PROCESSING

Mark Kahrs

Dept. of Electrical & Computer Engineering
Rutgers University
Piscataway, New Jersey 08855-0909
(908)932-4280 kahrs@winlab.rutgers.edu

Tom Killian

AT&T Bell Laboratories
Murray Hill, New Jersey 07974
(908)582-6215 tom@research.att.com

ABSTRACT

We have built an experimental machine to address three specific problems: narrow bandwidth of the control channel; limited access to the sound production mechanism; mixing and effects processing. Our hardware uses an experimental diskless workstation to control a variety of cards: an array of DSPs; an array of FM oscillators; the "final mix" with AES output; and MIDI/SMPTE I/O. Each card except the last has a core DSP (an AT&T DSP-16) to mix the output from upstream cards and perform effects.

1. Introduction

Existing electronic instruments have a number of difficulties. We have identified three problems that we believe can be solved.

(1) While versatile, the MIDI protocol is too slow to permit fast changing of instrument parameters. This is a direct result of MIDI's design; it was really designed as a gestural language and not a language to transmit large amounts of data such as envelope parameters. It is too slow to control many instruments simultaneously without running into synchronization problems.

(2) Algorithmic (as opposed to sampling) synthesis is useful because it provides a parameter space that may be explored for creative effects. FM algorithms in particular yield sounds with rich spectra for relatively few arithmetic operations. Currently, the largest FM synthesis capability using Yamaha FM chips is the TX816. Access to parameters is slow (via 8 separate MIDI streams), and only a fixed set of operator connection topologies is available. As a result, "Orchestral synthesis" isn't truly possible; it's difficult to layer enough timbres without running into bandwidth difficulties due to MIDI limitations (this is particularly true if the 8 MIDI streams are demultiplexed from a single source). The TX-816 has another limitation: it provides only analog outputs; these must in turn be mixed together with an analog mixer thereby providing extra opportunities for noise and programming complexity.

(3) Finally, most signal processing is done by separate "effects boxes" after the sound is generated. This eliminates possible feedback between the signal processing and the audio output, except through the slow MIDI channel. It also requires extraneous D/A and A/D conversions that reduce the sound quality and add to the expense of the system.

In our system, instruments are controlled over a high-speed bus by a fast processor, sound samples may be produced by fully programmable DSP's, and mixing and effects processing are integrated. The output is an AES/EBU [1] digital audio stream that can be fed directly to a CD mastering recorder.

The "gnusic" (**gn**ot **music**) project was created to overcome these difficulties. It is based on an experimental diskless workstation (the "gnot") featuring a Motorola 68020 processor, a bitmap gray level display, and an interface to a 1.5 Mb network. The Gnot is an experimental 68020 based workstation developed as part of a grand experiment in distributed computing. [2]. We have been developing an experimental machine for sound synthesis based on using the gnot as a real time controller. Our principal purpose was to explore the notion of "orchestral" synthesis; that is, the *real time* synthesis of large numbers of instruments comparable to that found in a modern orchestra.

The processor (housed in the display base) has direct access to the computational resources of instrument cards over an 8 bit bus. The bus is distributed over a backplane to the various cards; these cards can perform any number of functions depending on the card. The architecture of the basic machine is shown on the next page in figure 1.

An earlier version of this paper appeared as AES preprint 2868.

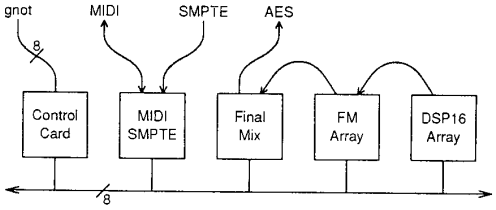

Figure 1.
Basic machine architecture

To begin with, the 8 bit data bus and associated control signals are delivered to a control card at one end of the backplane. These signals arrive from the gnot via a flat ribbon cable. The control card performs simple decoding tasks and buffers the signals onto the backplane. It also contains a SCSI interface for a disk when on the road. There are several different kinds of i/o cards. There is a MIDI/SMPTE card for interfacing to keyboards and video tape recorders respectively. There are three different kinds of sound generation ("instrument") cards: a DSP-16 array, an FM array and the "final mix" card. Each instrument card has a digital signal processor (an AT&T DSP-16 [3]) for mixing the digital outputs. This processor must (a) mix the output from the card "upstream" with the sample generated locally, and (b) perform any effects desired with the leftover time. Such effects might include filtering or feedback control of the oscillators. The basic core includes a pseudo-dual-ported static program memory for the DSP-16. Because the DSP-16 is so fast (80 ns cycle time), memory access must be shared between the 8-bit processor bus and DSP; when the processor sets the DSP to "run", it also prevents itself from accessing the memory.

The "final mix" card is last in the chain; it does the sample rate conversion from the Yamaha sample rate to EIA rate (44.1K). The card also contains an AES/EBU converter attached to the serial port of the DSP-16.

The serial output section of the DSP-16 contains both input and output sides. Each side has a shift register attached to the internal data bus of the DSP-16. The Input Buffer Full (IBF) and Output Shift register Empty (OSE) bits are available to external interfaces. The chip also provides load clocks for the shift registers and frame sync. The DSP-16 can either be a master (called "active" in DSP-16 terminology) or a slave ("passive"). We put all the DSP-16s *except* the final mix DSP in passive mode. They are fed clocks from the final mix DSP. This guarantees that all DSPs are on the same output clock. The final mix DSP also provides the left right clock so that the channels are synchronized as well. The serial data output of the DSP-16 is fed to the serial data input of the next DSP-16 in line. All of the serial I/O is done via flat ribbon cables on the end of the cards (we use a 26 conductor cable with alternating grounds). The cable consists of the following signals:

DI/DO	data in/data out
ICK/OCK	input/output clock
ILD-/OLD-	input/output load clock
IBF	input buffer full
OSE	output shift register empty
SADD	serial address (for TDMA mode)
LR	left right clock
BCK	bit clock (32 * 44.1 Khz)
YCK	Yamaha clock
SYNC-	frame sync

1.1. The final mix card

The final mix card has the basic ("core") DSP-16 circuitry found on *all* instrument cards. A block diagram is found below in figure 2.

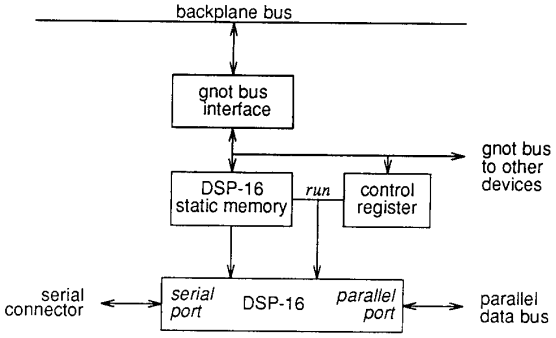

Figure 2.
Core DSP architecture

The gnot can write into the memory of the DSP-16; but only when the DSP-16 is stopped. The DSP-16 is too fast to allow true dual port access. This is perfectly acceptable since the core program typically doesn't change when the synthesizer is running. The gnot can also set a control register which contains the run flag and other useful bits.

1.2. FM array card

Our FM instrument card uses 16 Yamaha YM-2151 oscillator integrated circuits. This is approximately the same chip as in the Yamaha commercial product FB-01. The card is organized as shown on the next page in figure 3.

† The DSP-16A is even faster: 50 ns cycle time.

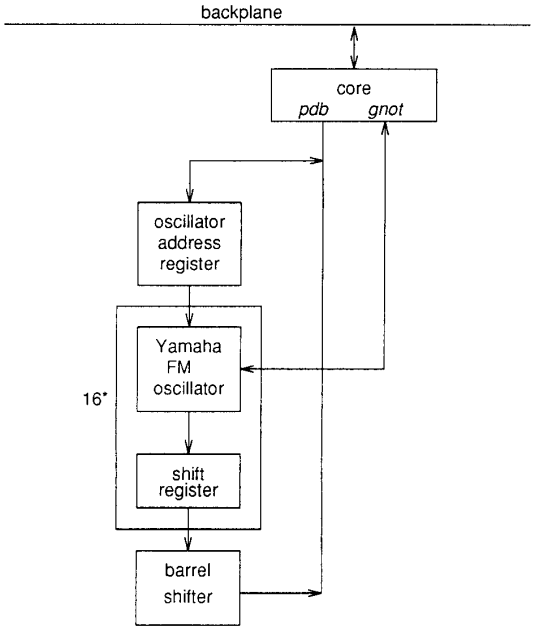

Figure 3.
FM array block diagram

Note how the core DSP circuitry integrates into the card; the core DSP provides a mechanism to collect samples from all the FM chips as well as scaling and possible signal processing hacks inside the DSP. Since the output of the Yamaha chips is in 13-bit pseudo "floating point" format, it must be converted to linear 16 bit for use by the DSP-16. A barrel shifter does the trick here although it is overkill for the application. The output of the barrel shifter is put on the parallel input bus of the DSP-16. An auto-incremented address register selects which FM chip is driving the shifter. Meanwhile the serial output of the previous card is fed to the serial input of the DSP-16, for mixing with the barrel shifter output. The final result is put on the serial output pin which goes to the next card in the chain.

The gnot programs the Yamaha chips one at a time by setting a chip-select register and then writing data. The YM-2151 is fairly slow; a bus cycle takes about 500 nanoseconds, and incurs a 12 microsecond dead time. It takes 10 bytes to reset a channel (i.e., gracefully bring its output to zero), 38 bytes to configure a new instrument (voice), and an additional 22 bytes to sound a note, so the worst-case reload time is about 850 microseconds. Data intended for different chips may be interleaved (at the expense of more writes to the chip-select register), so most of the dead time can be recovered when things are busy. The equivalent FB-01 system exclusive MIDI message is 139 bytes long, requiring 44 milliseconds for transmission.

Unfortunately the voices for the YM-2151 just don't sound as good as the voices for the DX-7, the well known Yamaha 6 oscillator product, nor is there available anything comparable to the enormous library of DX-7 instruments. There are fewer oscillators per voice on the FB-01 (4 oscillators per "algorithm" versus 6), and their envelopes are more restricted. Of the 32 DX-7 algorithms, 25 can be simulated (modulo the envelope problem) using a set of YM-2151 channels. It is interesting that the remaining seven algorithms (for afficionados, 4, 6, 12, 13, 16, 17, 18) are commonly used in our favorite DX-7 voices. In spite of all this, the YM-2151 provides us a reasonable way to get a fair number of instruments in a short period of time.

Note that each Yamaha chip has 32 oscillators (16 FM oscillators); therefore each board provides 256 FM oscillators. For comparison, the 4B [4] at IRCAM provides a total of 64 FM oscillators per card using totally MSI scale logic.

1.3. DSP-16 array card

The DSP-16 array card has 4 DSP-16s and a core DSP. A block diagram of the card is shown below in figure 4.

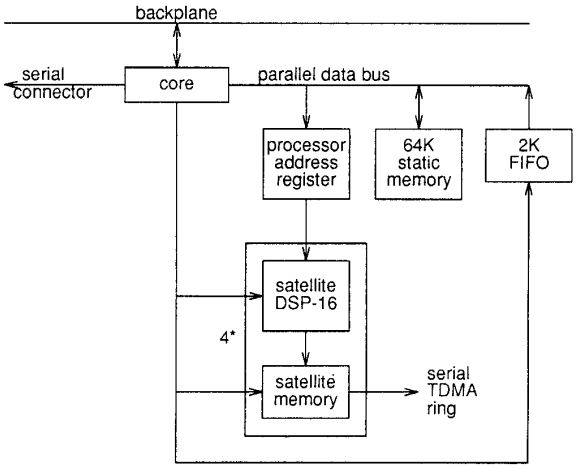

Figure 4
DSP-16 array block diagram

The core DSP addresses the 4 satellite DSPs via the 16 bit wide parallel I/O bus. The core can address any of the satellites as well as address a 64K word external memory This is specifically designed for reverberation (64K is about 1.5 seconds at 44.1 K sampling rate). The serial I/O of the satellites are connected together in a TDMA ring using the on chip logic of the DSP-16 shown on the next page in figure 5:

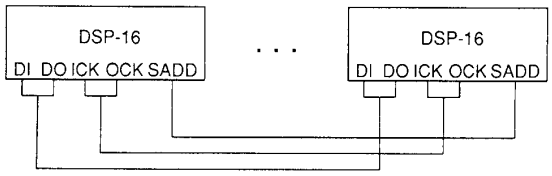

Figure 5
DSP-16 TDMA logic

The DSP-16 multiprocessor interface permits up to 8 processors to be connected on a bus. Each processor can be programmed via an internal register to transmit data (and an address) in a specific time slot. Data is transmitted (or received) on one pin and the address is transmitted (or received) on another pin (SADD).

The data, clock, serial address and sync pins are all bused together. Slots must be reserved (i.e., statically allocated) before hand as there is no contention mechanism. Furthermore, each processor must have a unique address.

The gnot has the same interface to the memory of the satellites as it does to the core. It also has a 2K by 8 fifo attached to the parallel I/O bus of the core for use in parameter passing from the host. Status bits from the fifo can be used to interrupt the core DSP should the fifo become too full and risk data overrun.

The slave DSP's are for general algorithmic synthesis and DSP algorithms (including pitch detection and various filtering algorithms). They are fast enough to compete with specialized VLSI in additive and FM methods.

1.4. SMPTE/MIDI card

The SMPTE/MIDI card does not have a core DSP section; it doesn't need one. It provides a simple interface to the SMPTE time code so that the synthesizer can be slaved to video tape machines. The MIDI channel is formed around the Yamaha YM-3802 MIDI Controller. While providing almost too much of everything, it does provide a way to accept keyboard input from a keyboard controller like a KX-88. Or even a DX-7.

1.5. AES converter

The final mix card cohabitates with an AES/EBU receiver and transmitter. This uses the (in)famous Sony CX-23035 and CX-23033 chips. The Yamaha sample rate (58.8 KHz) is converted to EIA rate (44.1 KHz) by interpolation and decimation by a ratio of 3:4. The output of the core is connected (via the standard serial flat ribbon cable) to the AES transmitter section. Likewise, the AES receiver section can be connected the serial input of the core DSP of the final mix card.

2. Software details

The DSP-16 is programmed in assembler and cross compiled on a VAX-8550†. The peculiar nature of the DSP-16 register set makes it very difficult to write a code generator for C. The resulting object files can be downloaded from the gnot via a remote file system. The file system also contains the instrument definitions for the Yamaha oscillator chips.

Our first cut at a 'music' interface is a MIDI file interpreter which builds on an existing score compiler [5]. Notes are assigned to one of the 128 channels on an oscillator card in a strict least-recently-used fashion. Thus each note-on event potentially causes a full voice load, but we guarantee full sonority to a timbre with long decay.

3. Software development

Future software to be developed includes a score editor and printer as well as graphical interfaces to instrument and signal processing definitions.

4. Conclusions

Gnusic was created to try out various ideas in orchestral synthesis and signal processing that are difficult to try out in MIDI systems. The combination of a powerful workstation and a flexible combination of versatile instrument I/O cards makes such experimentation possible.

5. Bibliography

1. Serial Transmission Format for Linearly Represented Digital Audio Data, *J. of the AES*, , Dec. 1985, 976-984.
2. R. Pike, D. Presotto, K. Thompson and H. Trickey, Plan 9 from Bell Labs, in *UKUUG Proceedings of the Summer 1990 Conference*, London, England, July, 1990, 1-10.
3. A. Microelectronics, WE DSP-16 Digital Signal Processor information manual, , 1987.
4. H. G. Alles and G. D. Giugno, The 4B: A One-Card 64-Channel Digital Synthesizer, in *Foundations of Computer Music*, C. Roads and J. Strawn (ed.), MIT Press, 1985, 250-256.
5. T. J. Killian, Computer Music under Unix Eighth Edition, *proc. European Unix Systems User Group (EUUG) Spring Conference*, Florence, Italy, April 21-24, 1986.

†VAX is a trademark of Digital Equipment Corporation