A Multi-Processor DSP System

for Real-Time Interactive Sound Processing and Synthesis

Dr. John A. Bate
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2
email: Bake@ca.UManitoba.CA

Abstract UniSon is a software/hardware system which allows sound synthesis and processing algorithms to be designed and entered by connecting graphical representations of modules on a computer screen. The results are synthesized in real time, and the system is fully interactive. The UniSon software will run on any Macintosh computer equipped with a Sound Accelerator or AudioMedia DSP card. However, in order to provide greater computational power for larger applications, a multi-processor system is required. This paper will describe the design of a NuBus card containing 4 56001 DSP chips which UniSon can use to provide additional resources in a manner which is completely transparent to the user. The inter-processor connection scheme allows any number of digital signals to be shared between any pair of processors with no overhead. An algorithm for distributing the tasks automatically among the processors will also be described.

I. Introduction to UniSon

The UniSon system combines a graphical user interface, a compiler, a linkage editor, and a DSP interface to create an environment in which real-time sound synthesis and signal processing algorithms may be created, used, and modified interactively. It runs on Macintosh computers and uses a Digidesign Sound Accelerator (or AudioMedia) DSP board to perform the actual synthesis or processing operations. One of the principal UniSon windows is shown in Figure 1.

Figure 1. A circuit window from UniSon
Each of the interconnected blocks in this window is a *device* which performs a particular function such as generating a sinusoid or an envelope. Each device has a number of input/output pins and may also contain *controls* which allow the user to directly affect its operation. The devices in Figure 1 contain scroll bars, envelope shapers, and a simple on/off switch. Circuits are formed by connecting a set of devices using lines which represent digital audio signals. New devices may be created by encapsulating circuits in which case the device may inherit any of the controls present in the circuit, forming a type of control panel. Devices may also be created by directly supplying 56001 DSP code together with the required *linkage* information. The synthesis is done in real time, and interactive control of all parameters is possible. Whenever a significant change is made to a circuit, it is recompiled into a block of DSP code, re-linked, and re-loaded into the Sound Accelerator. For practical circuit sizes, this happens in a fraction of a second. Whenever the user manipulates one of the controls, updated parameter information is sent to the DSP using a fast interrupt mechanism which makes its effect almost instantaneous. The UniSon system is similar in concept to Opcode’s MAX system, but all processing is done on individual digital sound samples at 44.1 kHz. It is therefore useful for applications such as sound synthesis and digital filtering rather than note-level or event-level tasks. For a more detailed description, see [1] or contact the author directly.

II. Single DSP Operation

UniSon currently utilizes a Sound Accelerator board which contains a single 56001 DSP together with a small amount of fast memory. Each device represents a small block of DSP machine code with single entry and exit points which is executed once per sample. Each interconnecting line in a circuit represents a single location in the DSP data memory. This location is written once per sample by the device which has an output connected to the corresponding line, and it may be read any number of times by devices whose inputs are connected to the same line.

To compile a circuit, UniSon simply concatenates the code blocks for all devices present in the circuit and assigns unique memory locations to all of the lines (and to any internal temporary variables needed by the device code blocks). A simple linkage editor then patches the code block for each device so that it refers to the memory locations assigned to its inputs and outputs. A linkage map is also created which shows the DSP memory locations which must be updated whenever any of the devices’ controls are manipulated by the user. The complete block of code is then loaded into the program memory of the DSP where it becomes an interrupt handler which is executed whenever a new output sample is required. When the user manipulates a control, UniSon uses the linkage map to determine the DSP memory location(s) which must be updated. The changes are made using interrupts which are handled during the idle time between the calculation of consecutive samples. The interrupt handlers are very short, consisting of only one or two DSP instructions, and therefore require a very minimal overhead.

The principal drawback of UniSon is that at 44.1 kHz even a 20MHZ 56001 DSP can only execute a little more than 200 instructions per sample. Since all of the input and output values are kept in memory locations whose addresses are unpredictable, and since each device must perform its task once per sample “from scratch” without assuming anything about the current state of the DSP’s registers, many of the pipelining features of the DSP go to waste and a lot of MOVE instructions are required. A high-quality linearly-interpolated oscillator will take almost 20 instructions which means that only about 10 such oscillators may be used before the capacity of the 56001 is exceeded. To be really useful, more processing power is required.
III. A Multi-processor Architecture for UniSon

In order to handle a greater number of devices in a UniSon circuit, it is only necessary to provide more computation cycles per sample. I propose to use a NuBus card which contains four additional 56001 DSP chips with a small amount of support hardware, as shown in Figure 2. The processors have very limited memory requirements since only a very small amount of memory can be referenced in the 200 or so instructions that it is possible to execute during a single sample time. The 56001 chip contains 512 words of program memory and 512 words of data memory which is adequate for storage of all internal connecting lines, local variables, and DSP code. Additional memory is needed only for storing tabular data such as wavenumbers, envelopes, and the like. The 8k RAMs provide for 32 wave tables of 256 words each which is more than adequate. Unfortunately, UniSon has no facility for working directly with sampled sounds since the architecture of the 56000 prevents it from addressing the required megabytes of memory.

The only other requirement is that connected devices which are running on different DSP's must have a fast and completely transparent mechanism for passing digital signals between them. In order to do this, a small (1Kx256), fast, four-port RAM is used as a "transposable memory array" or TMA as shown in Figure 2. This central memory may be accessed simultaneously by all four DSP chips with no wait states at all, provided that they do not reference the same address concurrently. Each processor generates a 10-bit address in the form $a_1 a_0 b_1 b_0 c_5 c_4 c_3 c_2 c_1$ where the bits $a_1 a_0$ represent its own processor number (in the range 0..3), the bits $b_1 b_0$ represent the processor number of the processor which it needs to communicate with, and the remaining bits specify one of 64 possible addresses which are devoted to signals shared between that particular pair of processors. If the TMA is viewed as an array with 4 rows, 4 columns, and 64-word entries, then each processor will always generate addresses for a different "row", thereby guaranteeing that identical addresses may never occur simultaneously. On alternate samples, the order of the "a" and "b" address bits is reversed using 16 2-to-1 multiplexors. This is transparent to the DSP's and results in the entire array being effectively "transposed" each cycle with no overhead at all.

A line in a UniSon circuit which connects two devices running on different processors will function as shown in Figure 3. The device whose output is connected to that line is running on processor 0 and the device using that line as an input is running on processor 1. The linkage editor will assign an address to this signal which lies in the 64-word block reserved for interconnections between these two processors, which will be (binary) 0001xxxxxx from the point of view of processor 0 and 0100xxxxxx from the point of view of processor 1. During the first cycle, Figure 3 shows the value 25 being produced as an output. During the next cycle, the array will be transposed and so the 25 which was written by processor 0 on the previous cycle will now be read in by processor 1. If the compiler simply assigns an appropriate address to the connecting line, the data transfer will take place automatically.

![Figure 2. Multiprocessor Card Basic Organization](image-url)
IV. Software Support for the Multiprocessor

Each output of each device in the UniSon system is either "delayed" or "immediate". A value produced on an immediate output is meant to be used by the devices receiving it on the same cycle, whereas a delayed output's value is not intended to be used until the following cycle. The "z_1^{10}\) unit delay device found in digital filter flow diagrams is an example of a device with a delayed output whereas arithmetic operations should have immediate outputs. In more complex devices such as oscillators or envelope generators, or in user-defined devices, the type of output is not usually significant. UniSon uses these output types to determine the proper execution order for devices in a circuit, and to check for improper feedback loops. It also assumes that every output is a delayed output unless otherwise specified by the user. Note that connected devices executing on different processors will automatically have the "delayed" property regardless of their actual order of execution. This makes the compilation process simple. Devices connected by an "immediate" output must be assigned to the same processor, but this is the only requirement. In almost all cases, UniSon is free to assign any device to any processor. The only possible exception occurs when delay-sensitive circuits such as digital filters are constructed from the basic arithmetic and unit-delay devices.

The multiprocessor card contains no D/A or A/D converters or other I/O devices. UniSon still relies on the Sound Accelerator to provide all I/O functions. Communication between the two cards is done using a synchronous communications interface. One of the four processors on the multiprocessor card is therefore designated as the master processor and its synchronous serial port is used to send and receive data from the Sound Accelerator. Special devices which represent this connection are provided to the user, and the compiler must recognize these and assign them to the master processor. This does not represent a significant amount of overhead since the devices in question are only a few DSP instructions in length. The only other difficult situation arises when one processor must send a signal simultaneously to more than one other processor. In this case, and in the case of controls which affect more than one device simultaneously, the compiler must insert a few invisible "copy" devices which simply copy their input to their output. Again, the overhead is minimal for these devices.

Space does not permit a more complete and accurate description of this system. If you would like additional information or a copy of the UniSon software system, please contact the author.

Reference