Abstract

The architecture for a real time dedicated digital sound synthesis processor is described. The system consists of a master computer that provides parameters to the dedicated processor (slave computer); a set of up to 4 DACs with sampling clock and a low pass filter bank is connected to the slave. Specifically, the master computer transmits to the slave both data and instructions that make up the sound generation program that pilots the slave and calculates, according to the data, the value of each sample to be sent to the converters. A new data-instruction block is transferred to the slave approximately every 10 μsec.

The processor makes use of two independent memories, one for data, the other for the instructions. Each memory is partitioned into two separate data areas, such that while one is being loaded from the master, the other is sending its previously memorized contents to the processor for execution. During the next load cycle the data areas exchange function.

A particular kind of phase register is another structure designed specifically for use with sound generating processor.

The major advantage offered by such a structure is that an elementary oscillator with amplitude and linearly interpolated frequency control may be realized with only 4 instructions (2 additions and 2 multiplications). Thus the processor is capable of generating, for example, around 60 oscillators, or harmonics, at a sampling rate of 30 kHz.
The paper describes the characteristics and performance of a last processor dedicated to the analysis and synthesis of sound in real time. This unit has been constructed in the joint laboratories of the Institute of Electronics and Electrotechnics and the Center for Computational Sonology—both at the University of Padua—in 1981/82.

A typical use of this computer is as a digital signal processor, and as such the machine realizes digital filters, implements the FFT algorithm, analyzes voice parameters through linear prediction, etc. Above all, however, this processor is designed to synthesize sounds (or voices) efficiently and in real time making use of the techniques currently being employed in the field of computer music, such as additive synthesis, AM, FM, non-linear distortion, and others.

The complete sound analysis-synthesis system consists of a master computer that sends and receives parameters from a slave computer, the dedicated processor. The master computer may have connected to it an ordinary alphanumeric keyboard as well as other equipment such as piano keyboards, consoles equipped with push-bottoms and pots or A/D converters. The slave computer has attached to it up to 4 D/A converters, a sampling clock generator, and a filter bank. This sort of structure is common to many digital sound synthesis systems.

The system currently running at the University of Padua (Fig. 1) consists of an IBM 51370/158 connected via channel attachment to an IBM S/370 minicomputer which acts as a buffer and in turn feeds the dedicated processor through the appropriate I/O ports. Thus in this case the master computer function is handled by the two IBM machines, together, for convenience; naturally, it is possible to use less powerful computers, even micro or personal machines.

During operation, the master computer transmits data and instructions to the slave. If sound is being generated, for example, the master sends the synthesis program to be executed as well as the data which is the basis for the program's computations that determine the value for each sample sent to the DAC. Whenever an up-date of data or program is necessary the slave sends an interrupt to the master causing a transfer of a new block of data or instructions.

It was decided to keep the instruction and data memories separate (Fig. 2) so as to be able to perform simultaneously the fetch and execute phases relative to two successive instructions, a feature common to many real time processors.

Each of these data and instruction memories consists of two distinct banks such that while one is being loaded by the master, the other causes execution of the program previously memorized. During the next load operation the two banks exchange functions.

The processor has an instruction set consisting of 16 logical operations and 16 24-bit arithmetic operations; there is also 16 x 16 bit multiplication hard-
ware, and both unconditional and conditional branching. For conditional bran-
ing, four flags may be tested in exiting from the ALU. There is also a wait
instruction necessary for halting the processor at the end of a sound synthe-
sis program execution generating a sample. In this case the computed sample
is stored in one of the 4 output registers and then sent to the DAC's accu-
diner to the timing of the sampling clock. At the same time the program coun-
ter is reset and begins execution of the program generating the next sample.
Clearly, the sampling clock is external, but can have any frequency compatibi-
ble with the complexity of the program. Normally, the sampling rate is 32 KHz.

The 34-bit instructions contain, besides the operation code, the addresses
for two operands and the result, if an arithmetic-logie type. In order to in-
crease execution speed, the operands are extracted simultaneously from the da-
ta memories. The result, on the other hand, can be put back in the memory for
further elaboration, or can be sent to an output where there is a summing acu-
mulator whose function is to facilitate additive synthesis by summing values
relative to various harmonics. These details are shown in Fig. 3.

During execution of an instruction, the next instruction is extracted from
the program memory and loaded in an instruction (pipeline) register. A branch
instruction contains, besides the opcode, the absolute branch address (relati-
ve or indirect addressing is not provided).

The two program memories can each hold up to 256 instructions. Altogether,
in one machine cycle, up to 6 different actions may be performed taking advan-
tage of pipelining techniques and parallel executions; there are: 4 memory o-
perations (3 dealing with operands and result, 1 dealing with the instruction
itself), one logical-arithmetic operation, and if desired an accumulation of
the sample in output.

The operands for logical-arithmetic computations can come from 5 different
sources: two externally accessible memories (data banks 1 and 2 each containing
128 locations); one scratchpad memory for intermediate computations (data bank
0, also 128 locations); one tables memory (16 512 location tables); one register
called the phase register. The tables memory contain the computed values of com-
monly used functions such as sine, exponents, etc. The logic organization of
this particular memory provides for a subdivision in 2 pages. Page 0 contains
the actual table values while page 1 contains difference tables. The latter are
obtained simply by subtracting the value of the function (page 0) at point N
from the value of the function at point N+1. In this way a subtraction operation
between adjacent table entries is saved whenever linear interpolation is requi-
red, since the differences are already available from the table in page 1. Ano-
the feature that adds considerably to the efficiency of oscillator synthesis by
linear interpolation is the automatic 8 bit shift that makes immediately availa-
ble the fractional part of the current phase (24 bits) to an operand bus through
the phase register. The fraction is multiplied directly by the stored table dif-
ference, thus permitting the synthesis of a simple oscillator, with amplitude and
frequency control, and with linear interpolation precision, with only 4 instruc-
tions: 2 adds and 2 multiplications. Without linear interpolation, an oscillator
may be realized with only 2 instructions: 1 add and 1 multiplication.

227
Since all instructions require one machine cycle per execution, except for multiplication which needs two, and that minimum cycle time for the processor is set at 100 ns, a simple non-interpolating oscillator runs at a minimum 300 ns per sample, while the interpolating oscillator requires 600 ns/sample. The maximum capacity, therefore, of the processor is 120 simple non-interpolating oscillators, or 60 interpolating oscillators at a sampling rate of around 30 kHz.

The output samples are 16 bit which assures a dynamic range of approximately 90 dB; the sampling increment is 24 bits long, 9 bits for the sign and integer part, 15 for the fraction. The frequency resolution is thus around 2 mHz (at 30 kHz sampling rate) which permits realization of any amplitude envelope: the lowest possible frequency that may be generated coincides with the frequency resolution (2 mHz), while the maximum frequency is the Nyquist frequency.