

Real-time Synthesis on a Multi-processor Network

Takebumi ITAGAKI^{†#}
Takebumi.Itagaki@dur.ac.uk

Alan PURVIS[†]
Alan.Purvis@dur.ac.uk

Peter D. MANNING[#]
P.D.Manning@dur.ac.uk

Durham Music Technology*

[†]School of Engineering
University of Durham
South Road
DURHAM DH1 3LE
U.K.

[#]Department of Music
University of Durham
Palace Green
DURHAM DH1 3RL
U.K.

Abstract

At the 1990 ICMC, this research group presented a paper on a multi-transputer based system [Bailey *et al.*, 1990] and demonstrated a prototype module consisting of 16 transputers on a single card. During the intervening period, ten of these cards have been developed into an operational audio processing network, using 160 x T800 floating-point transputers, clocked at 17.5 MHz. This paper describes the network and the first implementations of real-time direct synthesis on it, using a MIDI-based control system. The key research issue concerns the optimal usage of processor inter-communications and scheduling for audio processors.

1. Introduction

1.1. Transputer

The Transputer™ family of devices, designed by INMOS Ltd., offer versatile building blocks for the construction of multi-processor computing engines that are capable of establishing a high degree of parallelism. A T800 transputer consists of a 32-bit CPU, a 64-bit Floating Point Unit, four standard transputer communication links, 4k-byte of on-chip RAM, a memory interface and peripheral interfacing on a single chip, using a 1.5 micron CMOS process. The general purpose DSP chip achieves a performance of 8.77 MIPS at a processing speed of 17.5 MHz. Its FPU delivers a sustained floating point performance in excess of 1.32 MFLOPS in 32-bit, and 0.965 MFLOPS in 64-bit. There are four link interfaces that can transfer data at a sustained uni-directional rate of 1.74 Mbytes/sec or 2.35 Mbytes/sec in bi-directional mode.

As a single processor, the transputer's performance is not exceptional for audio processing when compared with state of the art DSP chips. For example, the Texas Instruments C40 executes 25 MFLOPS in 32-bit floating point mode at a processing speed of 50 MHz and Motorola's DSP56000 achieves 10.25 MIPS at a clock speed of 20 MHz. However, once a number of transputers are connected together

the resultant configuration of processors gives audio engineers a powerful and flexible resource. To achieve optimum performance, these devices have to be linked together in configurations that take full advantage of their distributed communication capability.

1.2. The 160 transputer Network

Since 1988, this research group has reported on issues concerning multi-transputer audio processors; for example [Purvis *et al.*, 1988] and [Bowler *et al.*, 1989]. A prototype architecture for a transputer network was described and demonstrated at the ICMC 1990, Glasgow [Bailey *et al.*, 1990]. This has subsequently been developed into a fully operational audio processor, using 160 x T800 floating-point transputers inter-connected as a ternary tree as shown in Figure 1.

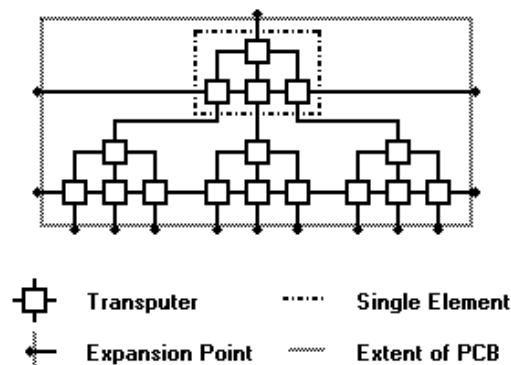


Figure 1: Basic Topology of the transputer Tree

* Durham Music Technology is a Collaboration between the School of Engineering and the Department of Music at the University of Durham.

A T800 transputer has four communication links that permit the construction of a ternary tree, which provide short path lengths between arbitrary nodes [see Bailey, 1991]. The fundamental single element employed by the machine is a modified version of a basic ternary tree in that the siblings are directly connected.

Four of these single elements, a total of 16 transputers, can be fitted onto a standard 3U printed circuit board. This represents a processing power of 140 MIPS per board. The transputers are hard-wired to each other permanently, but the software configuration of the network is flexible and re-programmable. An unusual feature of the design is the absence of any external memory on the board; only the 4k-byte of fast on-chip memory is available per transputer. This results in a total of 640k bytes across the network for our ten-board system, and a maximum processing power of 1400 MIPS. The rationale behind this design is that a real-time system should not require a large amount of memory for storage. The absence of external memory necessitates compact algorithms for execution at audio sampling rates.

2. Synthesis Method

For our initial evaluation of the performance characteristics of this synthesis engine, we have chosen to implement a recursive sine synthesis algorithm. This makes minimal demands on memory for multiple oscillators implemented in parallel. The recursive method generates high resolution sine waves by computing the projection of a rotating vector on the x- and y-axes.

Considering the second-order linear difference equation and applying the Z-transform:

$$Y_{(n)} = \alpha Y_{(n-1)} + \beta Y_{(n-2)} + X_{(n)}$$

$$H_{(z)} = Y_{(z)} X_{(z)}^{-1} = (1 - \alpha z^{-1} - \beta z^{-2})^{-1}$$

Solving for the roots of the denominator leads to the case $\alpha^2 + 4\beta \leq 0$, where the poles of $H_{(z)}$ are complex conjugates. They appear in the z-plane at $z = R e^{j\theta}$ and $z = R e^{-j\theta}$.

Here, $\theta = 2\pi \times f_q / f_s$ where f_s is sample frequency and f_q is frequency of a tone. R is the radial distance of the poles from the origin in the z-plane and θ is the angle made with the real axis. The equation for $H_{(z)}$ can be rewritten as:

$$H_{(z)} = \{(1 - R e^{j\theta} z^{-1}) (1 - R e^{-j\theta} z^{-1})\}^{-1}$$

$$H_{(z)} = (1 - 2R \cos\theta z^{-1} + R^2 z^{-2})^{-1}$$

For $R=1$, this leads to the following difference equation:

$$Y_{(n)} = 2 \cos\theta Y_{(n-1)} - Y_{(n-2)}$$

To generate an oscillator of amplitude A , the difference equation is started from $Y_{(n-1)} = 0$, and the amplitude is set by seeding the correct $Y_{(n-2)}$ value to $A \sin\theta$.

At a 32 kHz sampling frequency each transputer is able to provide 8 recursive oscillators that can be controlled independently in both amplitude and frequency. If the sampling frequency is increased to 44.1 kHz (CD quality) the capacity reduces to 5 oscillators per transputer.

Calculations for the oscillators are executed in 32-bit floating point format. When a sound sample exits from an oscillator unit it is converted into an integer number which is then accumulated with other synchronous samples throughout the network. Although the Digital-to-Analogue converter (DAC) has only a 16-bit bandwidth, 32-bit integer format is used internally since this achieves optimal performance from the transputer software and also ensures that changes in amplitude levels will not result in a loss of quantisation accuracy.

3. Applications

3.1. "88-note organ"

A custom-designed MIDI-to-transputer interface board provides the communication link between the transputer network and a suitable MIDI performance device. For the purposes of initial software development a standard MIDI keyboard has been used, generating; "key number", "note on", "note off" and "note velocity" control commands in polyphonic mode. The interface board transfers the information to the host transputer, which processes the information for distribution to the transputers in the network.

Since the processes of synthesis to be employed are entirely additive, working from basic sinusoids, the generation of interesting timbres becomes a function of how many individual sine wave oscillators are assigned to each note, and how they are regulated in terms of both frequency and amplitude. An acoustic piano has 88 keys covering a fundamental frequency range from 27.50 Hz to 4186 Hz. Since there

are only 81 transputers at the bottom of the tree structure some optimisation is necessary. As an initial step a prototype configuration for 81 notes was implemented and tested. In this configuration, the network is capable of accommodating 81 oscillator groups (one group per transputer) which provide 567 recursive sine oscillators in total, at a 32 kHz sampling rate.

Considering the harmonic content of notes to be synthesised and the limiting bandwidth of the DAC, it is not always necessary to assign all the oscillators on a transputer for the generation of a particular timbre throughout the note range. Some components indeed will lie above the Nyquist frequency and introduce fold over errors, if unintentionally generated. These factors introduce some measure of economy. For example, some of the transputers can be shared by two notes. Further gains can be made by utilising transputers at intermediate levels of the tree in combining their primary function as connectors with additional synthesis tasks.

As a result of this study of the network's performance we concluded that a transputer otherwise working as just a connector could additionally perform half the tasks of an oscillator group. This optimisation increased the capacity of the network to 108 oscillator groups which provide 752 recursive sine oscillators in total at a 32 kHz sampling rate. These oscillator groups can be controlled independently by MIDI signals, providing up to 88-note polyphony with the added enhancement of touch sensitivity control. Within the current limitations of sixteen oscillators per note, a wide variety of timbres can be pre-programmed and performed. The system performs reliably providing the event rate is less than about 150 keystrokes per second. If a higher rate of MIDI information is fed into the network, the system will temporarily halt until the control information has been delivered to each oscillator unit. The need to include a sound sample buffer between the DAC driver and the root transputer introduces a constant 32 msec of delay between the start of performance event and its realisation. In most performance situations, however, this could not be detected.

3.2. Dynamic Allocation of Notes

3.2.1. Implementation

In our current configuration of the network as an 88-note organ, the situation arises in normal performance situations where a static allocation

of oscillators to specific notes involves a high degree of redundancy. Since 88-note polyphony is not normally required, given that even two performers playing a duet on a single keyboard have only twenty fingers at their disposal, there are significant advantages to be gained from an allocation system that allows the maximum deployment of oscillators per note, thus increasing the range of timbres that can be generated.

Although the implementation of optimal clustering algorithms, whereby closely spaced harmonics are approximated by one oscillator, has yet to be fully investigated, considerable benefits have already been gained from the implementation of a simple dynamic scheduling system. In this configuration the system has been programmed to accommodate 27 simultaneous notes which can be synthesised with an enhanced harmonic content using 24 oscillators per note and improved dynamic control. In this case a set of five transputers can be fully deployed in a typical arrangement where three transputers are programmed as oscillators, a fourth provides up to 5 envelope functions for assignment to the above and the fifth acts as a signal mixer.

Given the complexities of such a large parallel network the scheduling of events presents a number of difficulties. In devising a suitable dynamic scheduling system a number of problems specific to a parallel processing environment have to be overcome, in particular, the risk of creating a 'deadlock' where unforeseen sequences of events can result in two or more processes becoming interdependent and thus unable to complete. The resolution of these conflicts appears to increase the latency of the system.

3.2.2. Control Strategy

In adopting a tree-structured approach to configuring the network, it would seem logical to use an approach where the distribution of control information flows down through the branches of the tree and the results of the synthesis process are accumulated in the reverse direction for output at the top level. In practice, when both strategies are implemented simultaneously, the density of traffic especially on the input side can result in a serious bottleneck at the point of entry. Rather than modify the basic tree structure, which in turn would fundamentally alter the operational

characteristics of the network, advantage has been taken of the special architecture of the transputer which allows external communications through any unattached link. By diverting some control information to side links near the top of the tree, this bottleneck can be contained without significant loss of efficiency as regards internal communications. Despite the above modification it still remains the case that the distribution between communication and synthesis tasks changes progressively from one almost entirely devoted to the former at the top of the tree to the reverse situation at the bottom.

4. Summary

We have confirmed the viability of the 160 transputer network as a real-time audio processor, in particular as an additive synthesis engine. The potential processing power of the network can only be realised with optimal software configuration. At present we have succeeded in making available 752 real-time oscillators at 32 kHz with individual control of amplitude and frequency. Faster control rates can be provided at the expense of the number of active oscillators. A dynamic allocation results 27 note polyphony with up to 24 oscillators per note. The system provides a programmable test-bed to investigate real-time control issues for audio synthesis.

5. Acknowledgements

Alan Purvis and Peter D. Manning gratefully acknowledge the generous donation of processors from INMOS Ltd., UK. Takebumi Itagaki would like to acknowledge travel funds from the University of Durham.

References

- [Bailey *et al.*, 1990] Bailey, N.J., Bowler, I., Purvis, A. and Manning, P.D. "An Highly Parallel Architecture for Real-time Music Synthesis and Digital Signal Processing Application", Proceedings of ICMC Glasgow, UK
- [Bailey 1991] Bailey, N.J. "On the Synthesis & Processing of High Quality Audio Signals by Parallel Computers", Ph. D Thesis, University of Durham

[Bowler *et al.*, 1989] Bowler, I., Manning, P.D., Purvis, A. and Bailey, N.J. "Additive Sound Synthesis on a Multi-transputer Network", Proceedings of ICMC Ohio, USA

[Jansen, 1992] Jansen, C. "Sine Circuitu", Proceedings of ICMC San Jose, USA

[Parash *et al.*, 1991] Parash, A. and Shimony, U. "An Expandable Real-time transputer Sound Generator", Proceedings of ICMC Montreal, CANADA

[Purvis *et al.*, 1988] Purvis, A., Berry, R. and Manning, P.D. "A Multi-transputer Based Audio Computer with MIDI and Analogue Interface", presented in Euromicro 1988, Zürich, SWITZERLAND. published in *Microprocessing and Micro Computing* **25** (1989): pp. 271-276