SHAPES - a Scalable Parallel HW/SW Architecture Applied to Wave Field Synthesis

Thomas Sporer[1], Michael Beckinger[1], Andreas Franck[1], Iuliana Bacivarov[2], Wolfgang Haid[2], Kai Huang[2], Lothar Thiele[2], Pier S. Paolucci[34], Piergiovanni Bazzana[3], Piero Vicini[4], Jianjiang Ceng[5], Stefan Kraemer[5], Rainer Leupers[5]

[1] *Fraunhofer Institut fuer Digitale Medientechnologie, Ilmenau, 98693, Germany*

[2] *Swiss Federal Institute of Technology Zurich, Computer Engineering and Networks Laboratory, Zurich, CH-8092, Switzerland*

[3] *ATMEL Roma, Roma, Italy*

[4] *INFN, Dip. Fisica Univ. Roma "La Sapienza", Roma, 00185, Italy*

[5] *Institute for Software for Systems on Silicon, RWTH Aachen University, Aachen, 52056, Germany*

Correspondence should be addressed to Thomas Sporer (`spo@idmt.fraunhofer.de`)

ABSTRACT

The usage of advanced audio processing algorithms in products has always been limited by the available processing power. For powerful concepts like the wave field synthesis (WFS) the performance is limited by the execution speed. In the past it was possible to increase the performance of digital signal processors by increasing the clock rate. The next generation will be highly parallel heterogeneous multi-processor systems. This paper presents a new parallel processor architecture and the first steps towards an adequate optimization of WFS. A software development environment which assists in creating scalable programs for highly parallel hardware will be further explained. An enhanced WFS convolution structure is presented which use position dependent filtering and improve the interpolation necessary for moving sound sources.

1. INTRODUCTION

Modern digital audio applications like wave field synthesis (WFS) [1] require an increasing amount of calculation power. WFS algorithms with enhanced audio quality and low latency between input channels and loudspeaker output channels exceed calculation power capabilities of recent digital signal processors. In the past processing power of DSPs were raised up by increasing the clock speed. In modern DSP, on-chip wiring does not allow anymore to increase clock speed. Software development environments have to deal with the growth in complexity of DSP and multiprocessor architectures. The modern audio algorithms must be massivly executed in parallel on these DSP architectures. For instance, the WFS processing power scales linear with the number of loudspeaker channels. For this reason, a scalable DSP platform is desirable.

The SHAPES project[1] (Scalable Software Hardware computing Architecture for Embedded Systems) targets three main objectives: investigate the tiled HW paradigm (see section 3.1 and figure 4), experiment a real-time, communication aware system SW, and validate the HW and system SW platform through a set of benchmarking applications, including the wave field synthesis. For an introduction to the SHAPES System SW and HW architecture, see [2][3].

2. SPATIAL AUDIO REPRODUCTION

2.1. Historical remarks

The history of spatial sound reproduction began with the concept of the acoustic curtain (many microphone wired 1:1 with many loudspeakers) at the Bell Laboratories [4]. Later research resulted in the reduction of the number of channels to basically three channels [5],

[1] SHAPES is a European Project (FET-FP6-2004-IST-4.2.3.4(viii) - Advanced Computer Architectures). See www.shapes-p.org for a complete documentation.

but due to practical limitations for long time only two channel stereo was applicable. Looking at further evolution from two-channel stereophony over quadrophony to 5.1, there are limitations which have not been overcome since the early days. Channel based spatial sound reproduction is based on the concept of using phantom sources. Phantom sources inherit the problems of no precise source localization and position dependet sound coloration. Far from the sweet spot the spatial impression usually collapses completely. Some efforts to solve these problems have been investigated, but all these investigations have not achieved economic impact. One of the reasons was the lack of a efficient data format. Since the late 90th the 3D audio profile of the MPEG-4 standard solves this problem. Since the beginning of 2001 universities, research institutes and companies joined their efforts in the development of 3D audio. The EU-project called CARROUSO [6] has developed key technologies for recording, encoding, transmitting, decoding and rendering a sound field in an efficient way at highest perceived quality. Important key components for these technologies were the Wave Field Synthesis (WFS) as a new way of reproducing sound and MPEG-4. WFS was invented at the TU Delft in Holland and has been demonstrated in academic environments successfully in the past [7][8]. Due to its high computational complexity it has not found broad application until today. The progress in microelectronics with decreasing costs of computing power enabled the first application in the professional market. WFS now is used in cinemas, open air cites, themed entertainment and VR installations.

2.2. Basic concept of WFS

WFS is based on the wave theory concept of Huygens: All points on a wave front serve as individual point sources of spherical secondary wave fronts. This principle is applied in acoustics by using a large number of small and closely spaced loudspeakers (loudspeaker arrays, see Figure 1). Each loudspeaker in the array is fed with corresponding driving signal calculated by means of algorithms based on the Kirchhoff-Helmholtz integrals and Rayleighs representation theorems [9].

$$P_A = \frac{1}{4\pi} \oint_S \left[\left(P \frac{1 + jk\Delta r}{\Delta r} cos\phi \frac{exp(-jk\Delta r)}{\Delta r} \right) + + \left(j\omega\rho_0 v_n \frac{exp(-jk\Delta r)}{\Delta r} \right) \right] dS \qquad (1)$$

The superposition of the sound fields generated by each loudspeaker composes the wave field. This technique enables an accurate representation of the original wave field with its natural temporal and spatial properties in the entire listening space.

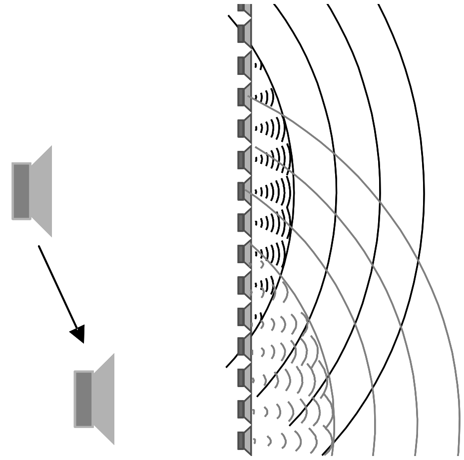


Fig. 1: Wave Field Synthesis based on the wave theory. Virtual sources can be placed anywhere.

By means of WFS virtual sound sources can be placed anywhere in the room, both behind the loudspeaker arrays as well as inside the room (focused sound sources). WFS is also capable of reproducing plane waves. Anywhere in the reproduction room the direction of a plain wave is the same and the sound pressure level is approximately constant [2]. Natural sound fields in general are composed of the sound fields of each individual sound source and the room impulse response. The acoustical properties of a reproduced sound scene can either be those of the recording room, those of a prerecorded different venue or obtained from an artificial room model (Figure 2). It has been shown that it is sufficient to merge the room response to a small number of plane waves [10].

2.3. WFS Realisation

Using WFS it is possible to treat signals coming from sound objects separately from signals coming from the room. If recorded and stored properly the musical recordings can be replayed mimicking the orchester playing in a different concert hall.

The best sound experience using WFS can be achieved when using specially prepared material. Such material

[2] WFS with just a linear array of loudspeakers creates cylindrical waves resulting in a decrease by 3dB by doubling distance.

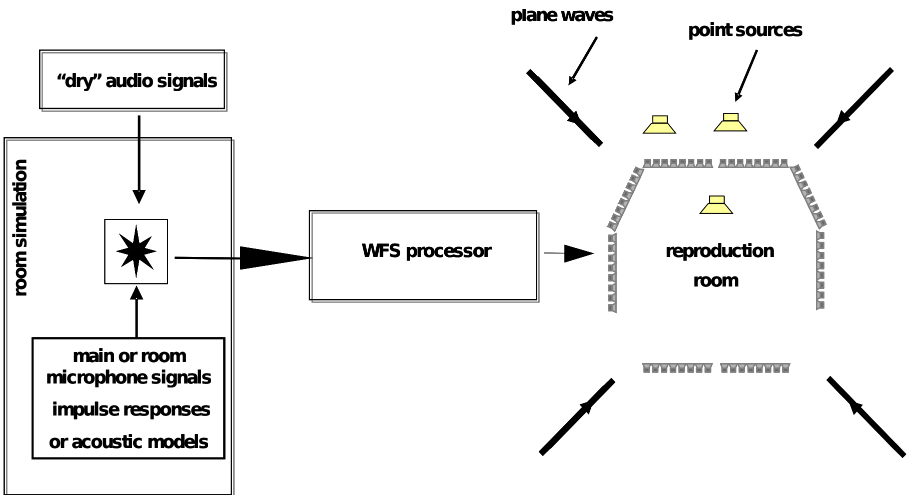


Fig. 2: Workflow from creating or recording signals for WFS reproduction.

consists of dry recordings of separate sound sources, their position in the room and information about the desired room acoustics. Using microphone array techniques recording of sound sources requires subsequent signal processing. By means of signal processing, sound source signals can be separated and unwanted signals can be suppressed. In addition, information about the position of possibly moving signal sources is extracted [11]. Besides the microphone array technique conventional 5.1 recording techniques including spot microphones can be applied, too.

For reproduction the acoustical scene consisting of audio objects, room acoustical parameters and scene description is rendered to loudspeaker signals. The number of transmitted audio tracks (either point sources or plane waves) is related to the scene and independent from the number of loudspeakers at the reproduction site.

2.4. Applications

Over the long run, WFS and mathematically related sound rendering methods like higher-order ambisonics reproduction will find its way to all sound reproduction systems where ever it is possible to use more than just one or two loudspeakers.

2.4.1. Application areas

Concert halls The WFS algorithms exhibit intrinsic delay times short enough for life performances. In contrast to the systems used today WFS can provide spatial angular and distance resolution of the acoustic scenes on stage.

Open air events Key requirements for open air concerts are equal distribution of the sound pressure level across the whole listening area and spatial coherence of sound and visual scene on stage. While line arrays of loudspeakers can only satisfy the first requirement WFS provides much better spatial coherence. While line-arrays control the sound pressure level in regions giving problems at the cross-sections of neighbouring regions such problems can not occur with WFS because it is based on continuous sound fields.

Cinema In addition to an accurate representation of the original wave field in the listening room, WFS gives the possibility to render sound sources in their true spatial depth and therefore shows enormous potential to be used for creation of audio in combination with motion pictures. On February 19th, 2003 the first cinema equipped with WFS system started daily service in Ilmenau, Germany (Figure 3). A larger setup is in operation in Studio City, LA, USA. Both installations can also serve as mixing sites. They reproduce WFS content, but all legacy format films benefit from the increased sweet spot, too. Such films are reproduced by the WFS system via virtual loudspeakers placed outside the cinema hall. Plane waves improve the spatial sound quality for the surround channels.

Fig. 3: WFS system in the cinema (Lindenlichtspiele, Ilmenau, Germany)

Home theatre systems Today WFS is rather expensive, but this will change over time. Other obstacles for WFS in the home are the placement of the loudspeaker arrays and the acoustics of the room. For the latter, a combination of acoustic treatment (e.g. curtains) and the application of room equalization techniques (e.g. compensation of a few early reflections) is probably the best solution. DML panels [12] might be part of the WFS equipped home theatre of the future.

2.5. Current research topics

While WFS systems are ready for widespread applications, a number of research topics still remains. While there is still some work left on basic theory, a lot of the current research topics are application driven. The following list only addresses some of the major issues.

2.5.1. Acoustic echo cancellation for WFS

If WFS is used in a communications setup (e.g. high quality video conferencing), AEC (Acoustic Echo Cancellation) is a necessary part of the system.

2.5.2. Array equalization

According to the theoretical WFS driving function for the loudspeakers a correction filter must be implemented to get a flat frequency response of the system. Most theoretical papers only focus on virtual sound sources far behind the loudspeaker array. For this position a 3dB per octave suppression of low frequencies has to be applied. Taking into account, that for source positions exactly on a speaker no frequency correction is necessary practical implementations require an adaptation of the filter to the current source position. For simple configurations, like rectangular arrays, closed solutions are possible. For real world applications (non-rectangular arrays, irregular gaps between loudspeakers) the problem is far more complicated. For cost-efficient realisation it is essential that such equalisation methods do not have to be perfect in acoustic sense but only have to be perfect in psychoacoustic sense. The performance of algorithms will therefore be evaluated with the help of listening tests.

2.5.3. Listening room compensation

If the acoustics of the listening room overlay the virtual acoustics of the simulated listening space, the sensation of immersion is greatly reduced. One way to get around this problem is of course to use a dry, non-reverberant listening space. Often this is not possible or economically feasible and electronic room compensation methods need to be applied. The search for the best compromise to do this is still on. From a psychoacoustic perspective it might be sufficient to cancel the first few reflections of the reproduction room which occur before the reflection of the room to be reproduced. Solutions to reduce reflections coming from the walls, which are equipped with loudspeakers, have been proposed. The cancelation of reflections from other room boundries is still an unaddressed problem.

2.5.4. Complex scenes

WFS is ideal for the creation of sound for motion picture or virtual reality applications. In both cases the creation of highly immersive atmospheres is important to give the auditorium the illusion of being a part of the auditory scene. Especially demanding are atmosheres with many objects like rain and applause.

3. SHAPES HARDWARE

Current WFS implementations are either based on standard PCs or DSPs. These systems usually are only able to render small number of sound objects in realtime. If there are moving sound sources the number of sound sources is limited even more. This section describes a new DSP architecture which is highly scalable in computational power and communication capacity which promises to overcome these limitations.

3.1. SHAPES hardware architecture

A serious challenge is to identify a scalable HW/SW design style for future CMOS technologies enabling high gate counts [13][14][15]. The main HW problem is wiring [16][17], which threatens Moore's law. A second HW problem is the management of the design complexity of high gate count designs. Tiled architectures [18]-[19] suggest a possible HW path: ”small” processing tiles connected by ”short wires”.

Each tile of SHAPES includes a few million gates, for optimal balance among parallelism, local memory, and IP reuse on future technologies. The SHAPES inter-tile routing fabric connects on-chip and off-chip tiles, weaving a distributed packet switching network. 3D next neighbours engineering methodologies are studied for off-chip networking and maximum system density and scalability, leveraging on the know-how accumulated by INFN during the design and development of several generations of massive parallel processors [20]-[21] dedicated to numerical computations.

Each tile of SHAPES always contains one Distributed Network Processor (DNP[3]) for inter-tile communications, plus one mAgicV VLIW floating-point DSP[4], for numerical computations, and/or a RISC processor for control intensive codes. Intra-tile communications are sustained by a Multi-layer Bus Matrix, while inter-tile communications are supported by the Network-on-a-Chip (NoC based on Spidergon Network-on-Chip[5]) and by the 3DT (off-chip 3 Dim. Toroidal next neighbours interconnection network). The DNP acts as a generalized DMA controller, off-loading the RISC and DSP processors from the task of managing the packets flowing through the inter-tile network. SHAPES includes a Distributed Memory Architecture: each Tile is equipped with distributed on-chip memories and can be associated with an external distributed memory (DXM). Each tile may also contain a POT (a set of Peripherals On Tile). In its first implementation, the SHAPES tile will be developed as the combination of a new generation of the DIOPSIS (RISC + DSP) MPSOC [23][24], with a DNP. mAgicV VLIW DSP is a fully C programmable, high performance digital signal processor delivering 10 floating-point operations per cycle and 16 ops per cycle. It is new member of the mAgic [25] processor family used in the Atmel Diopsis product line (multiprocessor systems on chip combining a RISC and a DSP).

3.2. SHAPES Audio interfaces

Output data (corresponding to loudspeakers) and input data (moving sound sources) are forecasted to be carried through a set of Multi Channel Audio Digital Interfaces (MADI) [26]. An exercise with 32 input channels and 128 output channel (48 Khz sampling rate, 24 bit) is a good ”low-end” starting point for our analysis. As discussed later, the computational power of a SHAPES system with 64 tiles should be adequate for this basic exercise, while an advanced system would require 512 tiles. A SHAPES system is organized using a 3 Dimensional next-neighbours topology (3DT), where 2 or 3 dimensions can be closed in a toroidal manner. A board with 64 tiles can be designed according to different topologies, e.g. 4*4*4 or 8*8*1. Each tile is equipped with 6 bidirectional links in the gigabit/s range, supporting the 3DT network, driven by the DNP (Distributed Network Processor) which is integrated in each tile. Each tile can be equipped also with 4 bidirectional SSC (Serial Synchronous Channels) (50 Mbit/s) hosted by the POT (Peripherals on Tile), which can be also configured to act as I2S stereo channels at audio bit rates. Each MADI interface can support up to 64 channels @ 48 KHz @ 24 bit. Therefore we need to interface a SHAPES system with 1 MADI for input and 2 MADI for output to support this working exercise. In this case, the interface between a SHAPES board hosting 64 tiles and the set of 3 MADI interfaces can be realized in a FPGA based system which acts as a protocol converter toward: either a set of 8 SSC links for output and 4 SSC links for input, used at an effective rate of 25 Mbit/s, or a pair of 3DT links (one for input, one for output). A SHAPES board can also directly offer a parallel set of I2S channels at audio-bit rate offering a direct interface toward a set of DAC/ADCs.

4. SOFTWARE DEVELOPMENT FRAMEWORK

4.1. Introduction

The SHAPES SW structure can be seen as a composition of several layers, mainly the application, the Operating System (OS) and the Hardware dependent Software (HdS). The SW layers have a distributed execution on the underlying parallel HW. In order to distribute its execution onto different parallel resources, the application is parallelized by the application designers. The application functionality is described using the standard C language and using a fixed set of API primitives.

The underlying layers include the OS and HdS. They offer an abstract access to the HW architecture resources. In these layers, most of the computation sharing and communication refinement mechanisms are implemented, taking into consideration the application map-

[3] designed by INFN
[4] designed by ATMEL Roma
[5] designed by ST Microlectronics [22]

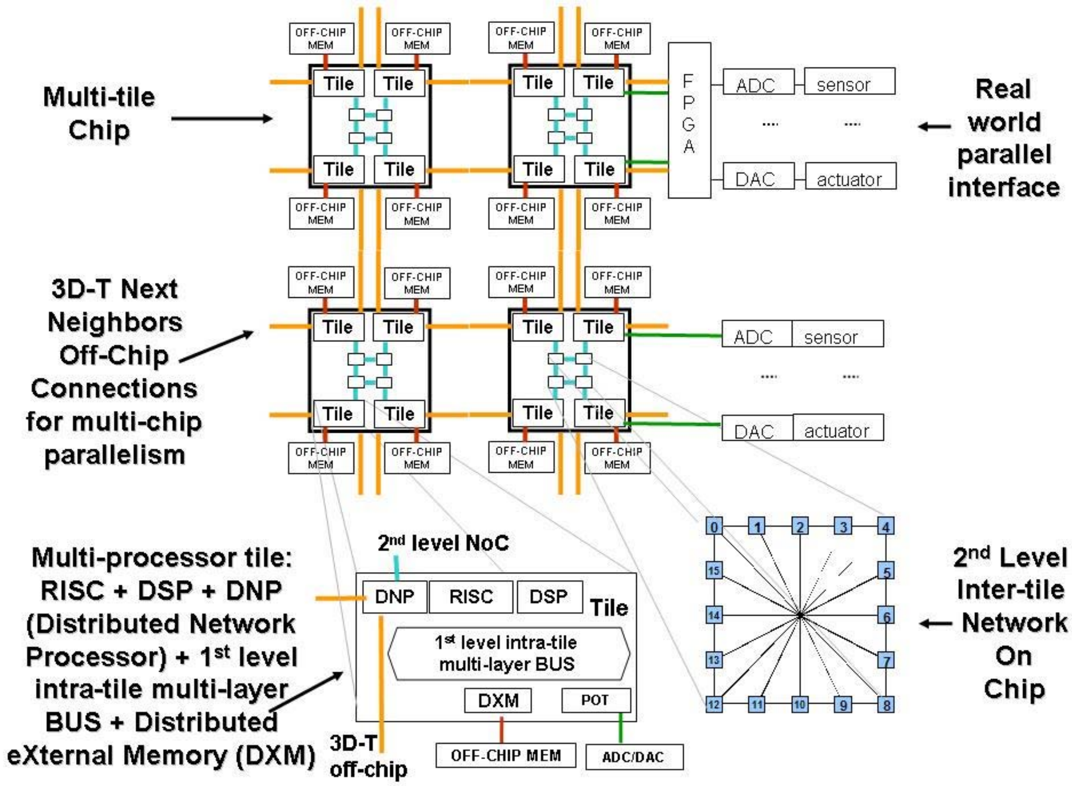


Fig. 4: The tiled HW Architecture of SHAPES

ping information. The OS and HdS run locally on each processor, their code being optimized for the resource where they execute.

The Distributed Operation Layer (DOL) is another component of the SW framework. Its main role is to automatically map the application onto the underlying architecture, trying to optimize the overall performance. The DOL framework enables the SW flow automation, minimizing the effort associated with the application designer. It facilitates the automatic refinement of the OS and HdS, by offering besides mapping information, a standard API and programming model for the application description.

The whole architecture is simulated for validation and performance evaluation. The first functionality check is done in DOL, as well as the functional profiling of the application (Section 4.2.5). For a more accurate simulation and performance data collection, the virtual SHAPES platform is used (Section 4.3).

4.2. The Distributed Operation Layer

4.2.1. The DOL Framework

The central role of the DOL in the SW development framework is summarized in Figure 5, representing the DOL structure and interactions with other tools and elements in the SW flow. However, Figure 5 does not show the complete SW flow, omitting the intermediate steps of OS and HdS generation, for which the DOL mapping information is an input. Typically, the simulated mapping instance is the complete SW architecture.

The DOL requires as inputs the description models of the application, the architecture and the mapping constraints concerning the application and the architecture. The main output of the DOL is the mapping specification of application processes onto different architectural resources, e.g. RISC or DSP processors, and of the interprocess communication onto the available communication paths between resources. Moreover, the DOL can offer performance evaluation results for a given mapping, based on its internal analytic model.

The core of the DOL tools is dedicated to the mapping optimization. The DOL generates automatically the

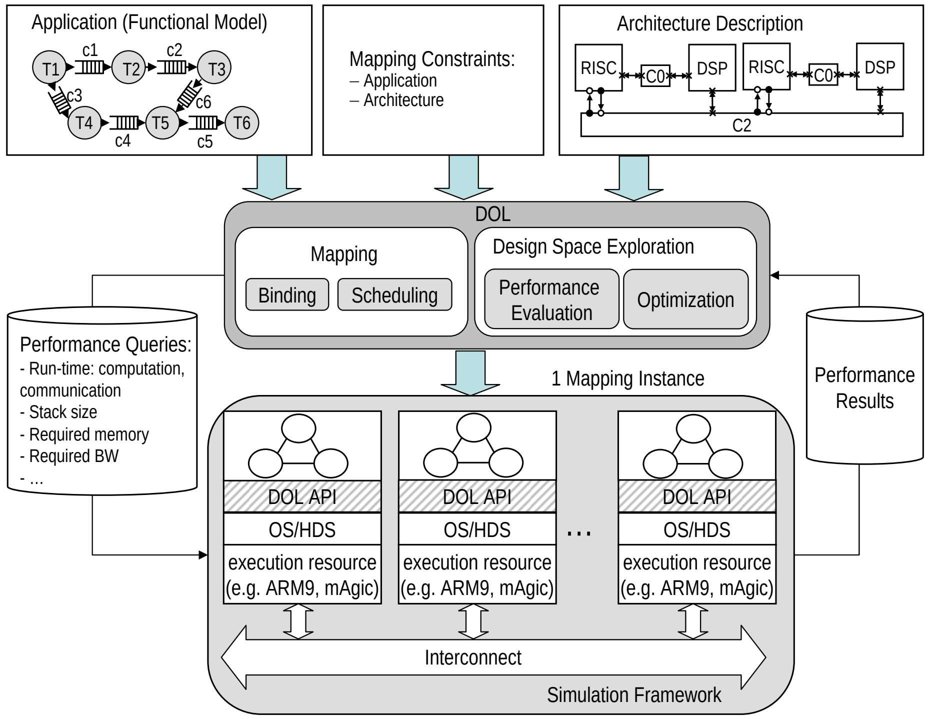


Fig. 5: The DOL framework and I/Os

mapping, evaluates its performances and optimizes the mapping solution based on the simulated or analytically computed performance figures.

4.2.2. The DOL programming model

The DOL programming model is the process network. This model of computation allows separating the application functionality (i.e., the application C code) from the application structure. It also allows for the separation between computation and communication in the application specification. The DOL proposes a high-level API[6] which provides uniform primitives for all the SW layers. The DOL API is necessary for the description of processes, for inter-process communication and, later on, for the communication refinement in OS and HdS.

It is often the case that the applications contain some degree of regularity, especially stream and signal processing applications. The DOL offers the possibility to describe repetitive structures in an efficient way, by using the so-called ”iterators”. The usage of iterators for the WFS application is illustrated in section 5.4.

[6]The DOL API includes the primitives: DOL_read(), DOL_write(), DOL_rtest(), DOL_wtest() and DOL_detach()

4.2.3. The DOL architecture description

The architecture description is one of the DOL main inputs, in order to derive the optimal mapping of the application. The DOL relies on an abstract HW architecture description, including elements like: execution resources, storage resources, communication resources, their interconnection and their performance parameters.

4.2.4. The DOL mapping

The mapping designates the relation between the application and the underlying HW architecture. The aim is to find the best distribution of the application processes and their communication on different execution resources and communication paths, respectively. For shared resources, scheduling strategies and parameters are decided. Examples of sharing parameters are the static ordering of process execution or priorities in case of fixed-priority scheduling.

Moreover, the mapping needs to comply with given constraints, externally specified by the application designer. The constraints are limitations for the mapping of processes onto processors or the mapping of their communication onto potential HW communication paths.

4.2.5. The DOL application evaluation and mapping optimization

The first functionality check of the parallel application is realized using the DOL functional simulation tool. The DOL simulator is implemented in SystemC [27] and automatically generated based on the application code and the process network structural description.

In the DOL, the functional simulator is coupled with an automatic profiling tool. Application profiling data can be collected at run-time, at a functional level. Examples of profiling data are the buffer usage, the number of communications per communication channel and the number of process invocations. For a more accurate design space exploration, interactions with a more precise simulator, i.e. the virtual SHAPES platform are necessary. Mainly, the simulator is used to provide algorithm-specific or dynamic data, like for instance the run-times of processes.

The DOL mapping optimization is an automatic iterative process, which makes use of the performance evaluation results, obtained from the DOL internal analytic model and/or by interacting with the simulation framework and the DOL profiler.

4.3. SHAPES Simulation Environment

4.3.1. Overview

Virtual SHAPES Platform (VSP) is the simulation environment for the SHAPES architecture. Within the SHAPES project, the role of VSP is to support the software and hardware development. With the capability of fast prototyping the target system, VSP enables the SHAPES system architects to evaluate different design options during the high-level architecture exploration. For the SHAPES software developers, VSP is a simulator of the SHAPES hardware platform. With VSP, they can test and debug their applications before the actual hardware prototype is available. In this way, Hardware dependent Software (HdS) and operating systems can be developed concurrently with the hardware.

For the implementation of VSP, the tradeoff between the simulation speed and accuracy has been taken into account. In order to fulfill the requirements of the SHAPES project, VSP currently employs instruction-accurate instruction-set simulators, TLM [27] interconnection models and functional peripheral models.

4.3.2. VSP/DOL Interaction

For the development of the Wave Field Synthesis (WFS) application on SHAPES, VSP may be used in two ways.

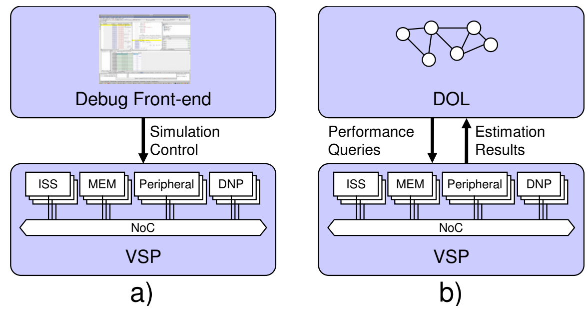


Fig. 6: VSP User Interfaces

Firstly, a lightweight GUI debugger front-end is available, which allows the user to manage the simulation conveniently (Figure 6 a). During the simulation, a full featured debugger can be connected. This mode is mainly used for software debugging. The GUI front-end provides an easy to use interface, but it has to be manually controlled by the user. For the mapping exploration done in DOL, more automation is required. Therefore, VSP supports another interface (Figure 6 b), through which DOL is able to interact with VSP without user intervention. In this mode, the simulation is controlled by a script file, and the results will be automatically returned to DOL for further analysis.

5. WFS AND SHAPES

5.1. WFS process network

Figure 7 shows an overview of processes running in parallel on the SHAPES platform. On the left side, m input channels are connected to appropriate clusters of *WFS processes*. Each output of an *WFS process* cluster is connected only to one summing node which creates the loudspeaker signal. Each *WFS process* is directly controlled by the *control unit* with a point-to-point connection. The control unit directly interacts with the mixing console or virtual scene player in the WFS setup. Control data packets must be simultaneously received at the *WFS processes*. Each *WFS process* equalizes directivity dependant sound colourations of the loudspeakers. Additionally, the virtual source position dependant 3dB/Oct. low pass filter effect of WFS and room acoustics of the listening room can be compensated. For this reason an impulse response *IR data base* is used to provide the *WFS processes* with appropriate impulse response correction filters. Measurement methods for the creation of multichannel equalization filters can be found in [28],[29],[30]

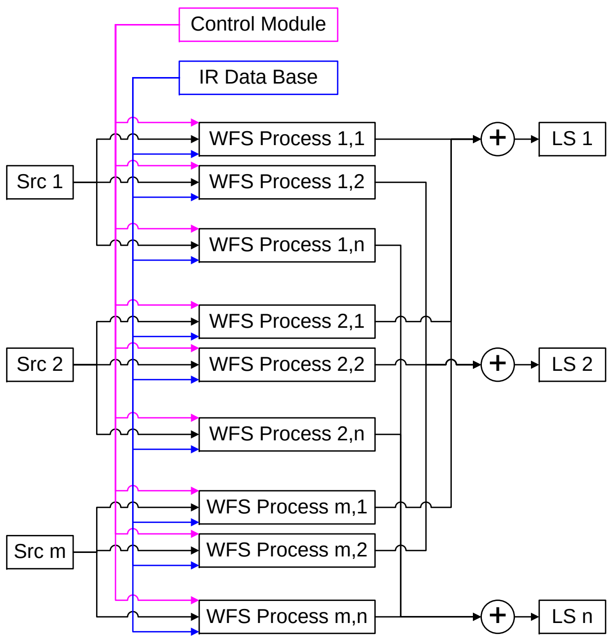


Fig. 7: Overview of processes running in parallel on the SHAPES platform

and [31]. The WFS process network is similar to the solution proposed in [32].

5.2. WFS process - overview

An overview of the internals of one *WFS process* shows figure 8. Audio input is fed into the *moving sound source* module further described in section 5.3. The output of the *moving sound source* module is connected to two filtering modules. Only one module at a given time is active. If the filter coeffcients change because the position of the virtual sound source changes significantly, the opposite filter module is loaded with the new filter coefficients and the output signal is stereo blended from the old filter coefficients to the new one with the *blending module*. The filter coefficients are loaded from a huge global *impulse response database* accessible from all *WFS processes*. The *control* module controls the movement of the virtual sound source and the choice of filter coefficients to the appropriate spatial sound source position. Convolution is done with a fast FFT partial block convolution in frequency domain according to [33] and [34]. If a low latency application with latencies of 3ms is required, the impulse response can be devided into blocks of 256 samples in size. FFT is optimized by interleaving real and complex values of the real input into a complex FFT.

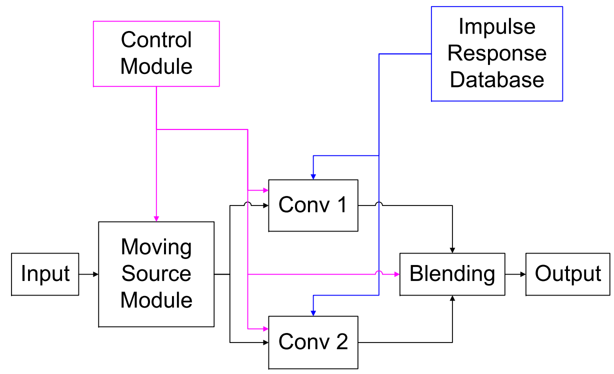


Fig. 8: Internal structure of each WFS process

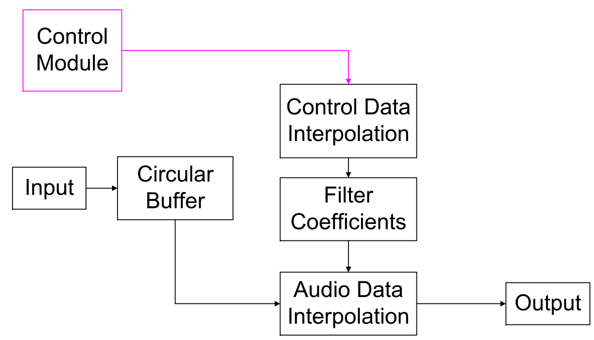


Fig. 9: Moving Source Module

5.3. WFS process - moving source module

Figure 9 shows the structure of the *moving sound source* module. Audio is fed into a circular buffer, is read with a certain delay from this buffer and is interpolated in an audio data interpolation unit to realize a sub-sample accurate delay line. To decrease the control data bandwidth from this module a control data interpolation module interpolates the incoming control data.

Application of the WFS synthesis operators requires arbitrary delayed input signals. In case of moving sound sources these delays are changing continuously. Because the audio signals are represented by discrete-time data, it is necessary to interpolate between values between sample points of the input signal. The concept of audio signal interpolation is depicted in fig. 10.

The interpolation of signal values between sample points of an discrete-time sample points is termed *delay interpolation* or *fractional delay*. A survey of this area of research is given in [35].

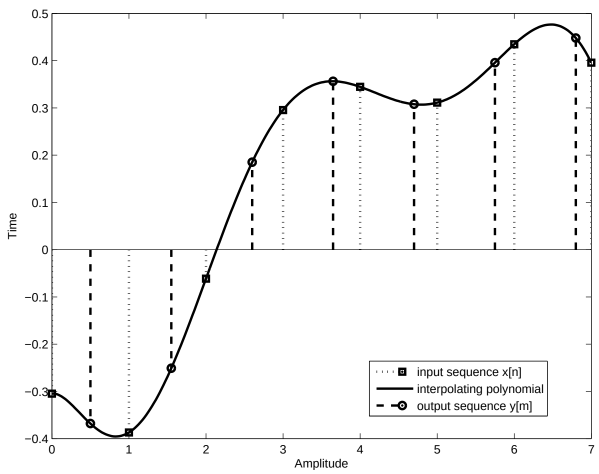


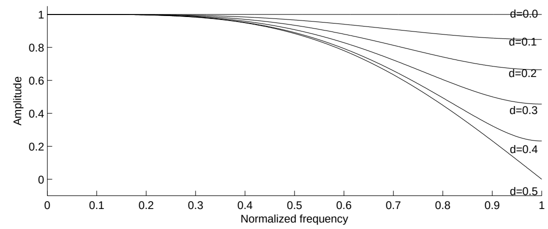
Fig. 10: Fractional Delay interpolation by means of Lagrange interpolation

Lagrange interpolation is one of the most widely used fractional delay intepolation methods. It is based on polynomial interpolation, i.e. a signal is interpolated by a polynomial of order N that is defined by N successive points of the input signal, N being the order of the interpolation. Lagrange interpolation has the property of maximal flatness [36], that is, the frequency response error and its first N derivatives equals 0 at a predefined frequency ω. This leads to very small interpolation errors especially at low frequencies. Moreover, the interpolation coefficients can be calculated using explicit formulas. These properties make lagrange interpolation to one of the most widely used delay interpolation algorithms in audio signal processing.

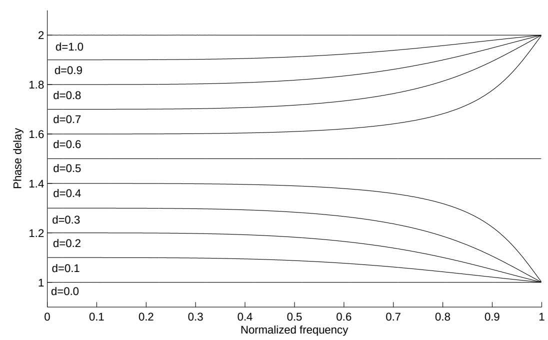
Fractional delay interpolation consists of two tasks: calculation of the filter coefficients and the actual filtering. Filtering requires $N+1$ additions and N multiplications per output sample, N being the order of the FIR filter. For lagrange interpolation, an explicit formula exists for the filter coefficients $h(n)$. d denotes the fractional delay.

$$h(n) = \prod_{k=0, k\neq n}^{N} \frac{d-k}{n-k}, \quad 0 \leq n \leq N \tag{2}$$

Application of this formula requires $O(N^2)$ multiplications and additions to evaluate the $N+1$ filter coefficients. By exploiting common subexpressions of eq. (2), (see e.g. [37]), the operation count can be reduced to



(a) Magnitude



(b) Phase delay

Fig. 11: Frequency response for a Lagrange interpolator (order $N=3$)

$N+1$ subtractions and $4N-2$ multiplications. Thus the calculation of the coefficients can be realized efficiently for higher interpolation orders, too.

Since fractional delay interpolation is an approximation of an ideal fractional delay, it introduces interpolation errors, which generally depend on the fractional delay value d. The magnitude response and phase delays for an 3rd order lagrange interpolator are shown in fig. 11. These interpolation errors may lead to audible artifacts, especially in case of moving sound sources. The most severe artifacts are:

- Delay-dependent amplitude response errors lead to amplitude modulations.
- Delay-dependent phase delay errors cause frequency modulations.
- Continuously changing delay values, as caused e.g. by moving sound sources, lead to aliasing or imaging artifacts equivalent to the effects known from sample rate conversion [38].

In a WFS reproduction system, a fractional delay operation is performed for every combination of a virtual source and a loudspeaker. Therefore, both the use of highly parallel architectures and the development of efficient algorithms that offer good audio quality are of utmost importance for the overall performance of future WFS reproduction systems.

5.4. Implementing the Wave Field Synthesis algorithm in DOL

A reduced version of the WFS algorithm described above was implemented in the applications programmer's interface DOL. A process network according to figure 7 was created, except the connection to the *IR data base*. The process instances were programmed in plain C language. Each process is connected with a point-to-point FIFO. Control data packets must be simultaneously received at the processes. More than 500 processes were generated in the WFS example application by DOL iterators. The application contains two sound sources SRC, one control module, 256 WFS processes, 128 summing modules and 128 loudspeaker output channels (LS). Two SRC processes generate two sine waves with a sampling rate of 48 kHz which are then rendered by the WFS process network. Loudspeaker channel signals of 1 s length were generated and evaluated correctly using a functional SystemC simulation. The application was automatically profiled using the DOL design space exploration engine. The profiling results include the buffer usage, the number of communications per communication channel, and in future versions the estimated runtimes of processes on the underlying architecture. In this example, buffers are used only 50 % (with a data filling level of 128bytes). Data throughput from each SRC to WFS process is 192 kByte/s.

6. CONCLUSION

The DSP platform and software development environment developed in the European project SHAPES offers new possibilities to parallel processing. Several algorithmic blocks of Wave Field Synthesis, which are essential for WFS in real world applications, have been analysed in respect to parallelisation and implementation on that hardware. This includes a new approach for delay interpolation for moving sound sources in WFS. The new scheme provides an artefact reduced, highly parallelized, algorithm for moving sound sources. A first implementation of some blocks of these algorithms was simulated in the VSP simulator.

7. REFERENCES

[1] M. M. Boone, E. N. G. Verheijen, "Multichannel Sound Reproduction Based on Wavefield Synthesis“, presented at the AES 95th convention, New York, USA, 1993 September

[2] L.P. Carloni, A.L. Sangiovanni-Vincentelli, ”Coping with latency in SOC Design”, IEEE Micro 22-5 (2002) 24-35.

[3] P.S. Paolucci, P. Vicini et al. ”Introduction to the Tiled HW Architecture of SHAPES” presented at DATE 2007 Conference, France, Nice, 2007 April

[4] Steinberg JC, Snow WB: ”Auditory Perspectives - Physical Factors”, ”Electrical Engineering”, volume 53, pages 12–17, 1934

[5] Snow WB: ”Basic Principle of Stereophonic Sound”, Journal of SMPTE, volume 61, pages 567–589, 1953

[6] Brix S., Sporer T., Plogsties J.: CARROUSO ” An European Approach to 3D Audio”, AES 110th convention, 2001 May, preprint 5314.

[7] Berkhout, A.J.; de Vries, D.; ”Acoustic Holography for Sound Control”, AES 86th convention, 1989 March, preprint 2801.

[8] De Vries, D.; Vogel, P., ”Experience with a Sound Enhancement System Based on Wavefront Synthesis”, AES 50th convention, 1993 October, preprint 3748.

[9] De Vries, D.: ”Sound Reinforcement by Wave Field Synthesis: Adaptation of the Synthesis Operator to the Loudspeaker Directivity Characteristics”, J. Audio Eng. Soc., Vol.44, No. 12, 1996.

[10] Theile G., Wittek H., Reisinger M., ”Potential Wavefield Synthesis Applications in the multichannel stereophonic world”, AES 24th conference, Banff Canada, 2003 June 26–28.

[11] N. Strobel, S. Spors, R. Rabenstein: ”Joint Audio-Video Signal Processing for Object Localization and Tracking”, In M. Brandstein, D. Ward (Eds.) ’Microphone Arrays: Techniques and Applications’, Springer, Berlin, 2001, pp. 197–219.

[12] Ulrich Horbach, Diemer de Vries, Etienne Corteel: ”Spatial Audio Reproduction Using Distributed Mode Loudspeaker Arrays”, AES 21st conference, St. Peterburg, 2002 June.

[13] D. Sylvester and K. Keutzer, ”Impact of Small Process Geometries on Microarchitectures in Systems on a Chip”, Proc. IEEE, 89-4(2001)467-489.

[14] W.J. Dally and S. Lacy, ”VLSI Architectures: Past, Present and Future”, Proc. Advanced Research in VLSI Conf., IEEE Press (1999)232-241.

[15] A. Allan et al., ”2001 Technology Roadmap for Semiconductors”, IEEE Computer 35-1(2002)42-53.

[16] R. Ho, K. Mai and M. Horowitz, ”The Future of Wires”, Proc. IEEE, 89-4 (2001)490-504.

[17] J. Rabaey, A. Chandrakasan, B. Nikolic, ”Digital Integrated Circuits”, 2-nd Edition, Prentice-Hall (2003) Chapter 4 and 9.

[18] M.B. Taylor et al., ”The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs”, IEEE Micro 22-2(2002)25-35.

[19] Paolucci, P. S., ”The Diopsis Multiprocessor Tile of SHAPES”, 6th International Forum on Application-Specific Multi-Processor SoC MP-SOC’06 (Colorado, August 2006)

[20] A. Bartoloni, P.S. Paolucci, P. Vicini et al., ”A Hardware Implementation of the APE100 Architecture”, Int. Journ. Mod. Phys. C 4(1993)969.

[21] Belletti, F, Vicini, P. et al. ”Computing for LQCD: apeNEXT”, Computing in Science and Engineering, 8-1, pp. 18-29, Jan/Feb, 2006

[22] R. Locatelli, G. Maruccia, L. Pieralisi, A. Scandurra, M. Coppola, ”Spidergon: a novel on-chip communication network”, International Symposium on System-on-Chip, 2004, pp. 15-26

[23] P.S. Paolucci et al. ”Janus: A gigaflop VLIW+RISC Soc Tile”, Hot Chips 15 IEEE Stanford Conference (2003). http://www.hotchips.org (note: Janus was the development name of the first generation of DIOPSIS).

[24] ATMEL Roma, ”DIOPSIS: Dual Inter Operable Processor in A Single Silicon”, www.atmelroma.it

[25] P.S. Paolucci, P. Kajfasz et al., ”mAgic-FPU and MADE: A customizable VLIW core and the modular VLIW processor architecture description environment”, Computer Physics Communication 139(2001)132-143.

[26] AES 10-2003, “AES Recommended Practice for Digital Audio Engineering - Serial Multichannel Audio Digital Interface (MADI)“, Audio Engineering Society, 2003

[27] SystemC Standard Language Reference Manual, http://www.systemc.org

[28] E. Corteel, U. Horbach, R. Pellegrini, “Multichannel Inverse Filtering of Multiexciter Distributed Mode Loudspeakers for Wave Field Synthesis“, presented at the 112th AES Convention, Germany, Munich, 2002, April

[29] D. de Vries, “Sound Enhancement by Wave Field Synthesis: Adaptation of the Synthesis Operator to the Loudspeaker Directivity Characteristics“, presented at the 98th AES Convention, 1995, January

[30] A. Apel, T. Roeder, S. Brix, “Equalization of Wave Field Synthesis Systems“, presented at the 116th AES Convention, Germany, Berlin, 2004, May

[31] E. Corteel, R. Nicol, “Listening Room Compensation for Wave Field Synthesis: What Can Be Done? “ presented at the 23rd International Conference, Denmark, Helsingor, 2003, April

[32] Horbach, Ulrich; Karamustafaoglu, Attila; Boone, Marinus M., “Practical Implementation of a Data-Based Wave Field Reproduction System“ presented at the 108th AES Convention, France, Paris, 2000, February

[33] W. H. Press, S. A. Teukolsky, W.T. Vetterling, B.P. Flannery, “Numerical Recipes in C“, Cambridge University Press, 1988-1992

[34] S. W. Smith, “Digital Signal Processing Guide“, Elsevier Science, 2003

[35] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine, “Splitting the unit delay: Tools for fractional delay filter design“, IEEE Signal Processing Magazine, 1330–1360, January 1996.

[36] Saed Samadi, M. Omair Ahmad, and M.N.S. Swamy, “Results on maximally flat fractional-delay systems“, IEEE Transactions on Circuits and Systems-I: Regular papers, 51(11):2271–2285, November 2004.

[37] Holger Strauss. “Simulation instationärer Schallfelder für auditive virtuelle Umgebungen“, PhD thesis, Ruhr-Universität Bochum, Bochum, 2000.

[38] Ronald E. Crochiere and Lawrence R. Rabiner, “Multirate Digital Signal Processing“, Prentice-Hall Signal Processing Series. Prentice Hall, Inc., 1983.