Florian Pfeiffle and Rolf Bader

Real-Time Physical Modelling of a complete Banjo geometry using FPGA hardware technology

Physical Modeling of Musical Instruments

In Musical Acoustics, realizing real-time solutions for Physical Models is a hot topic. The sound quality and variability of musical instrument sounds calculated by whole-body formulations will be pushing this field strongly towards working solutions in the near future. Many attempts are discussed here, yet none of them is working with satisfaction in real-time (Smith 2008). Reducing the number of Degrees of Freedom (DOF) in the models does affect the higher partials of sounds considerably. Changing the mode shapes of musical instruments in terms of spatial distortion is too time consuming to work in real-time. Reducing the complexity of instruments in terms of geometrical simplifications clearly leads to simplified solutions. Also, these models can scarcely be varied in terms of the geometrical fine structure of the instruments and therefore in modelling the parts crucial for musical expression and articulation.

Physical Modelling of Musical Instruments has become one of the major fields in research concerned with properties important for musical performance or instrument building (Bader 2008a). It has been applied to many musical instruments. It could be found that with guitars the coupling of bending and in-plane waves in the guitar body plays a crucial part in the sound (Bader 2005a, Bader 2005b). Also the need to build the guitar back plate under tension to enhance the brightness of the sound could be shown. When systematic changes are applied to the thickness of the top plate, back plates, rims, fan bracing and ribs, the change in brightness of the sound radiated from the different parts do sometimes show a linear but mostly a nonlinear behaviour, caused by the complex coupling of the parts. So e.g. if the fan bracing is added to the top plate and so reduces the flexibility of this plate (resulting in reduced radiation energy of the higher harmonics) this stiffer top plate is able to transport the higher modes to the neck much stronger. That again leads to an enhanced radiation of the high frequencies with increasing fan bracing thickness (Bader 2008c). Other investigations show the possibility to model the guitar in terms of modal synthesis according to the top plate only (Bécache et al. 2005). An example concerning the precision of such a modelling is shown in (Elejabarrieta 2002), where a Finite-Element analysis of a guitar top plate is compared to a real top plate through different stages of construction.

Other instruments have been investigated in a similar way. The pianos' sound board has been modelled in (Giordano 2006), where the fan bracing and the stiffness of the soundboard play crucial roles in the overall sound behaviour. The kantele, a finish dulcimer is investigated in (Erkut et al. 2002). Also a labium is studied in such a way by taking flow-dynamics and turbulence into consideration. It could be shown that the turbulent damping of the flute is crucial for its impedance. Only because of the damping caused by air vortices in the flute at the embochure hole the energy of the flow into the tube is only about 3 % of the whole blowing energy (Bader 2005f). Also fine structures of reeds used with single reed instruments could be investigated. With saxophones, the reed thickness distribution along the area of the reed could be associated quantitatively with the sound expected from different reeds of a commercial reed producer (Bader 2008d). Other investigations concerning the reed show a very similar air vortex distribution in the mouth piece (de Silva 2007).

Also different percussion instruments have been studied using Physical Modelling techniques. Bells show a difficult behaviour within their transient phase as well as in terms of damping or radiation directivity (Schoofs 2000, Lau et al. 2004, 2009). Optimization algorithms have been added to a Finite-Element calculation for bells to find perfect shapes for the bell (Özakča 2004). Other percussion instruments have been investigated, too. Essl and Cook (2000) discussed travelling waves along the rim of a round drum to be important for the transient sound behaviour. In a model/measurement comparison of the complete Bass-Drum, coupling the drums membranes to the wooden box and the inclosed air, it could be shown, that the higher frequencies are radiated from the wooden shell, showing its importance in sound production (Bader 2006c). An investigation of a Balinese percussion instrument (the gender dasa played in a gamelan *wayang* used for puppet theatre) showed, that the trapezoid shape of the plates is producing additional modes through scattering of waves during the initial transient phase of the sound (Bader 2009, 2004). This effect is so strong that it must be considered as the main mechanism producing the overall sound quality of the instrument.

Violins have been studied using whole body Finite-Difference Methods, also implementing the string-bow interactions with changing bowing pressure and velocity (Bader 2005d). Here, also the changes in thickness of different body parts show a complex interaction structure, meaning that linear changes in the violin body normally mean nonlinear changes in the radiated sound. This is also confirmed by investigations of instrument families, maybe most prominent with the violin octet (Bissinger 2003).

Summarizing these investigations, musical instruments do show properties in their sound design which can only be treated by examining the fine structure of the instrument geometry. This fine structure can be the precise shape of the instrument body, the coupling between different body parts, the thickness distribution of plates like violin top plates, the shape of fan bracing appearing with guitars, the distribution of turbulence during blowing, or the thickness distribution along reeds with saxophones or clarinets. All these parameters do influence the sound in a way that they are responsible for the slight changes in timbre which tell a master piece apart from another. Also, these fine structures are responsible for musical expression and articulation, i.e. they are the parts with musical instruments where “the music is happening”.

The investigation of these fine structures is often only possible when a geometrical model of the instrument is built, which can show up nearly all the details of the instrument (Bader 2006d, Elejabarrieta 2004, Bader 2009, Bader 2007b, Bader 2007c). These details are responsible for the fine tuning of the sound under different playing conditions. I.e. the quality of a violin may only be judged after using it under many different conditions, in a chamber orchestra, in large concert halls, with folk music, as a solo instrument etc. Under all these conditions, the violin has to fit the needs of presence, loudness, or timbre flexibility. These changes are very hard to detect with methods only concentrating on basic characteristics of the instruments.

The methods used in the field of modelling of whole instrument geometries to date are basically Finite-Element Methods (FEM) (Barthe 2002, Knothe & Wessels 1999), Finite-Difference Methods (FDM) (Bader 2005a), methods of flow dynamics like Lattice-Boltzmann method (LBM), or simplified methods like Waveguides, Delay Lines or Wave Front Methods. All these methods use the basic differential equations for bending, in-plane movement or flow dynamics with appropriate boundary conditions well known from the literature (Leissa 1969a, 1969b, Wagner 1947, Hutchings 1981, Fletcher & Rossing 2000, Flügge 1962). The advantages of FEM or FDM are that geometries of any complexity can be perfectly resolved and any kind of differential equation found to govern the system can be used to underlay the spatial discretization. The disadvantage is the use of many nodal points to resolve the geometry, resulting in high demands on both memory and computation time. So a method which allows the same spatial resolution but on the other side is able to speed up calculation time a lot would be of interest.

Advantages of real-time implementations of Physical Models

For many reasons real-time implementations are of great advantage:

1. Musical instrument shapes are so complex that many alternations are possible. Some sound improvements can only be found by trial and error. Until now, there is no theoretical framework enabling one at first to design a desired sound and subsequently use an algorithm which calculates the adequate instrument shape for that sound. So as the possibilities of instrument shapes are innumerable, a fast real-time tool is needed to try out different possibilities while judging the differences aurally at once.

2. Instrument builders are very seldom familiar with Physical Modelling and computer implementations. They need a soft- or hardware tool to try out different instrument shapes and listen to the results immediately. Although instrument builders might accept several minutes of waiting time for a result, a calculation time of several hours for one sound is too much for a feasible use in musical instrument building environments. Here a fast, if not real-time implementation is needed, which would mean a substantial progress in instrument building practice.

3. Modern music production is based on real-time sound producing algorithms, but no sound designer or musician would wait for hours for a sound to be generated. What they need is a real-time sound synthesis tool. With Physical Modelling one is able to change sounds within musically reasonable regions and therefore the creative possibilities are enlarged to a great extent. Many attempts have been made by commercial soft- and hardware musical instrument building companies (Roland, Yamaha, Apple, Steinberg, etc.) to incorporate Physical Modelling into

their instruments, and as a result hundreds of 'Virtual Musical Instruments' are on the market. Most of them are just samples of real instruments which are filtered and manipulated using modern musical signal processing techniques. Those who try to get as close to a Physical Modelling as possible use too many simplifications which results in sounds that are not convincing, i.e. they do not sound real (e.g. Roland VG 88 modelling guitars where e.g. the body height can be changed).

So the aim is to produce a real-time Physical Modelling hardware implementation using FPGA technology that keeps the full complexity of instruments while performing the sound synthesis in real-time. In the next section the method is explained, and the development status at the Institute of Musicology in Hamburg is presented.

FPGA-Hardware

A Field Programmable Gate Array (FPGA) is a special form of a Very Large Scale Integrated Circuit (VLSI) consisting of matrix-like ordered Logic Blocks, Input/Output-Blocks, and Routing Channels (interconnection network), and therefore similar to an Application Specific Integrated Circuit (ASIC). The Logic Blocks are build on different kinds of logical units, mainly Look-Up-Tables (LUT) with AND- and OR-Gates and other vendor specific logic such as FLIP-FLOPs or MUXs. So FPGAs are similar to single Algorithmic Logical Unit (ALU) microprocessors as logical operations and calculations can be performed, but additionally, they are like memory chips in terms of the amount of logical units performing calculations in parallel. Input and output are realized using the IO-Blocks of the FPGA, equipped with FLIP-FLOPs, which are attached to the physical boundary of the device.

Unlike an ASIC, all elements of the FPGA are freely programmable by the user and can be connected with the interconnection network to realise any logical problem, ranging from a simple adder to a Finite State Machine (FSM) or a complex arithmetic unit. This freely routable matrix-like structure yields the main advantage of an FPGA over other forms of VLSIs.

These capabilities of free programmability and of parallel rather than sequential processing are the great advantages of FPGAs compared to CPUs, DSPs or other Micro Controllers of VLSI architecture. It is therefore possible to compute massive numbers of instructions in parallel within one clock cycle on an FPGA. To benefit from this advantage it is crucial to implement an algorithm that processes as many parallel instructions as possible. It has been shown that an optimized parallel FPGA algorithm is always superior in means of calculation time and maximum clock rates compared to a similar sequential algorithm (Brassail et al. 2007, Subasri et al 2006, Lis et al. 2008, Zouet al. 2006, Inoguchi 2004). This massive parallel computation is implemented on a single small board, and so it is much smaller and much cheaper than a multi-core server system, where MPI can be used to parallel the algorithm (Karniadakis and Kirby 2003). These systems are very expensive even if only about 64 knots are used. Furthermore, more knots do not result in a linear speed-up. The FPGAs on the other hand are capable of processing e.g. 10 000 calculations in parallel on one single small board. An FPGA board can also be implemented easily in a commercial solution.

Another great advantage of an FPGA chip is that it can be programmed from a Personal Computer via a Vendor Specific Flash Tool. The model architecture which is to be implemented on the FPGA device is written in a special Hardware Description Language such as VHDL or Verilog. These Programming Languages contain many High Level constructs like *if-, for-,* or *while*-loops and some Low Level aspects like bitwise declaration of signals, variables and constants. The main difference in comparison to other programming languages is the concurrent processing of instructions. This means that unlike e.g. with the C programming language, where sequential code is needed, the code is processed in parallel and needs to be written accordingly. Although the FPGA structure is parallel by nature, sequential statements can also be realised using a Finite State Machine (FSM) as mentioned above and discussed in more detail below.

To avoid time consuming error corrections in codes and to reduce the flashing processing time, a simulation tool is used. Here, similar to a debugging tool, the FPGA code can be simulated using different simulation tools. These simulations can be performed with all kinds of different constraints, like clock speed and changing temperature specifications in the model. Only if the simulated model behaves as expected, one needs to compile and build the source code into a bit-file which then can be flashed onto the hardware-device.

The Build-Process includes different steps like the translation to hardware language, mapping of the IO-Ports and finally Placing&Routing the design so that all preliminaries for the hardware are met. With those advanced programming techniques the engineer is capable of modelling and simulating complex digital hardware systems completely in software before transforming them on a real hardware device. This allows a very fast and stable development of hardware designs without the need of a physical Integrated Circuit (IC) development.

FPGA - Implementations

Because of their high speed processing capability and flexibility FPGA-devices are used in many fields of digital signal processing.

In the area of sound analysis and sound source localization, implementations have been realized for e.g. Real-Time Noise Source Identification (Veggeberg et. al 2008), High Speed Direction of Arrival Algorithms (Hao et. al 2000), Delay-Sum Beamforming (Chen et al. 2008) and countless other applications. All of these works either show a real-time realization of the problem for the first time or a tremendous speed-up using an FPGA with its parallel processing capabilities.

Typical signal processing applications for FPGAs are implementations of IIR- or FIR-filter designs (Meyer-Baese 2004). In Madanayake et al. (2004) 2D/3D Plane-Wave filters are realised using IIR/FIR-filters. Shuang et al. (2008) focus on converting analog to digital controllers using a filter-design with FPGA. Maslenikow et al. (2006) and

Brich et al. (2006) discuss a method how DSP filter-designs (IIR/FIR) can easily be implemented on an FPGA chip.

In the field of sound production, several publications have been published in which an FPGA is used as a function generator for simple signals like a sine wave or a rectangle signal (Meyer-Baese 2004, Reichardt et al. 2006, Kilts 2007). Most of these works focus on high frequency signals like an AM-demodulation chain for a digital radio receiver (Meyer-Baese 2004).

Solving differential equations with FPGAs using finite differences have been reported before, too. Suzuki et al. (2005) discuss the use of an FPGA to simulate an electric field solving Maxwell's equations using a FDTD (Finite Differences in the Time Domain) algorithm. This algorithm is widely used for the analysis of electromagnetic problems (Shlager et al. 1995). Other approaches to solve differential equations on an FPGA implicitly are e.g. the implementation of the conjugate gradient method (Strzodka et al. 2006) or the Euler Method (Jayalakshmi et al. 2006). For parallel algorithms solving linear equation systems see also Karniadakis and Kirby (2003).

Papers focusing mostly on the real-time aspect of FPGA implementations are e.g. Particle Track Recognition (Liu et al. 2008), Direction of Arrival Algorithms (Hao et al. 2002), High Speed Cross Correlation (Von Herzen 1998), Noise Source Identification (Veggeberg et al. 2009), or Digital Beamforming (Wang et al. 2005).

Only one paper recently discussed the use of FPGA to synthesize industrial sounds (Martins et al. 2008). This paper focuses on a low-cost method to process sound on an FPGA without the need for additional electronics and omits musical aspects.

Physical Model of the Banjo

When building a Physical Model of an instrument it is crucial to know a great deal about the instruments' structure and its vibrational behaviour. The banjo's most important parts in the generation of its sound and timbre are the five metal strings and the plastic membrane, made of mylar (see Fig. 1). It has been shown that the wooden/metal body and the peg play a minor roll in the sound production of a banjo, as they are very stiff and therefore mainly fixed. They act as boundary conditions for the membrane and the strings (Dickey 2003). It can be assumed that the bridge serves as a low impedance transfer function element between the strings and the membrane. Only its position on the membrane has an influence on the sound (Stephey et al. 2008). The air beneath the membrane dampens this membrane and lowers the deepest modes of it, similar to the behaviour of a wooden snare drum (Fletcher and Rossing 2000). Although a great deal is known about the single parts of the banjo and their dynamic behaviour, very little research has been done on the coupled system, which is the membrane/bridge/string interaction (Stephey et al. 2008).

Formulating a more detailed model of the banjo as proposed here will lead to many insights, not only into the banjos sound producing behaviour, but also into those of other coupled systems and their dynamic behaviour similar to that of the banjo.

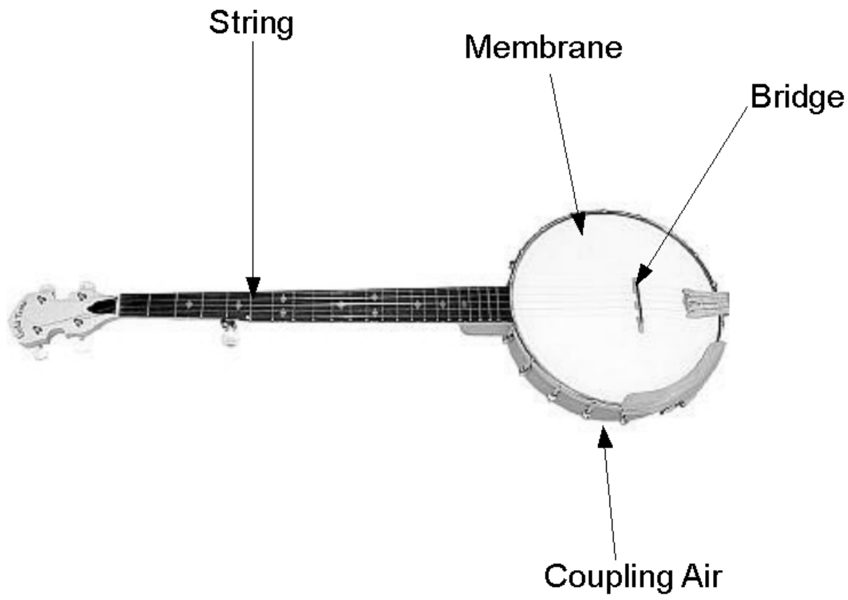


Fig. 1: The Banjo used in the model.

The dynamic system of the modelled Banjo is presented in Fig. 2. The model consists of string, membrane, bridge, and the coupling air beneath the membrane. As mentioned above this model is reduced to the core elements of the banjos sound producing parts. The bridge is the most sensitive part of it. Stephey et al. (2008) have pointed out that the position and the small weight of the bridge are of critical importance for the sound of the banjo whereas the shape is negligible. Because we found that the shape of the bridge is of importance for its transfer characteristics, we have implemented the bridge to be able to vary its shape and mass and to investigate the effects on the resulting sound. There may be a similar case as for the violin, where the two feet of the bridge are crucial to the sound but any carving of it is not, so it could be replaced by a so-called plate bridge (Jansson 2003). An earlier model of the banjo has shown that the bridge can be approximated by a linear factor without a big difference in the resulting sound compared to the sound produced by the final model. Still, in the literature there are many works in instrument acoustics focusing on the bridge, for instance with the violin (Wienreich 1977, Rossing et al.). The impedance function of the bridge may follow the violin, too, where the bridge-hill is crucial for the sound (Woodhouse 2000). Because there are many details that have not been fully understood in the role of the bridges as a transducer we decided to take the bridge into account.

First, each of the banjo parts is modelled separately in MATLAB and C Programming Language with the Finite-Difference Method (FD). The Physical Model of the string is a 100 Node 1D-FD. The membrane has 900 nodes using a 2-dimensional FD and the bridge is a 3-dimensional bending wave PDE using 750 nodes. With the air, 400 nodes are used. All parts take damping into account.

After successful single-system stability tests, the four systems are coupled linearly, yielding the dynamic system of a realistic full-scale banjo. To simulate the behaviour of the complex system, the model can now be excited, where appropriate initial values can be set on any node. To simulate a realistic banjo it is obvious that one has to deflect the string with a triangular shape at time point t = 0. Still with this time-dependent model all kinds of deflections can be realised at any node of the model and so one could also knock on the membrane, the bridge or the string, simulate additional damping anywhere or model unisotropic stiffness of all parts.

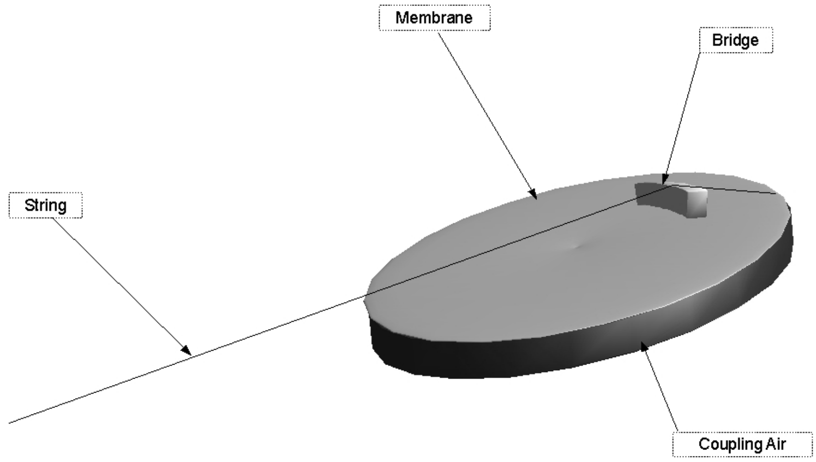


Fig. 2: Geometrical representation of the banjo model

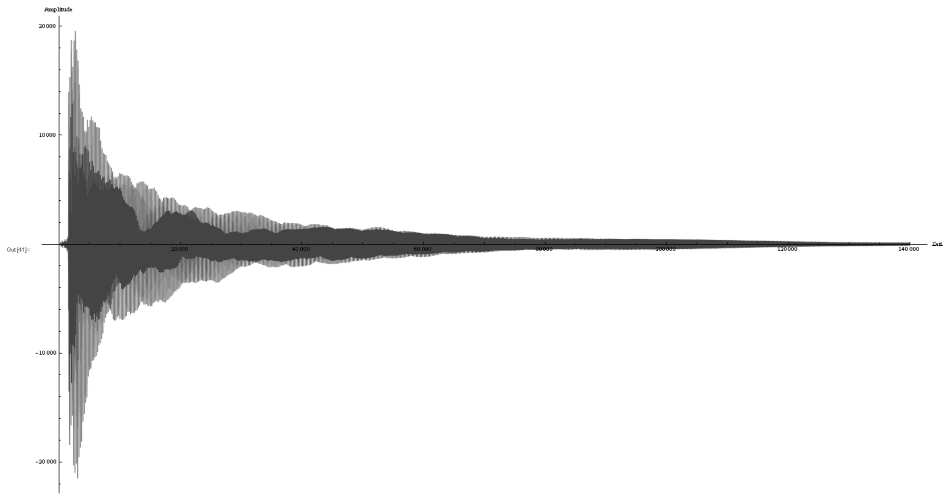
The resulting time series of a triangular deflected string is shown in Fig. 3 in comparison to the recorded and the simulated sound. The sound produced by the model is almost indistinguishable compared to the sound from a real banjo, even for a musical trained ear.

The drawback of this model is its tremendous computational costs. E.g., even with a speed-optimized C-Code the shown time series takes almost 30 seconds to compute on a 2,4 GHz Intel Quad Core Processor. With the normal processing capacities of a standard computer an implementation of the presented banjo model in real-time is not possible. A full-scale model with five strings instead of one string like the presented model can not be realized with the common computer architectures of a personal computer in real time.

FPGA Implementation of the Banjo

To implement the proposed model in real-time with high accuracy the C source-code was transferred into VHDL-programming language. All the *k* nodes of the model are build in parallel forming a 1-dimensional-array of interconnected parallel pipelined computations. In Fig. 4 a systematic view of the implemented string algorithm is shown. As one can see, the individual nodes are depending on their adjacent nodes, the clock and the FSM signal. Because of the interconnected node structure it is possible to compute one calculation step for all the *k* nodes in parallel within one clock cycle. Therefore the algorithm is extremely fast and effective.

a)



b)

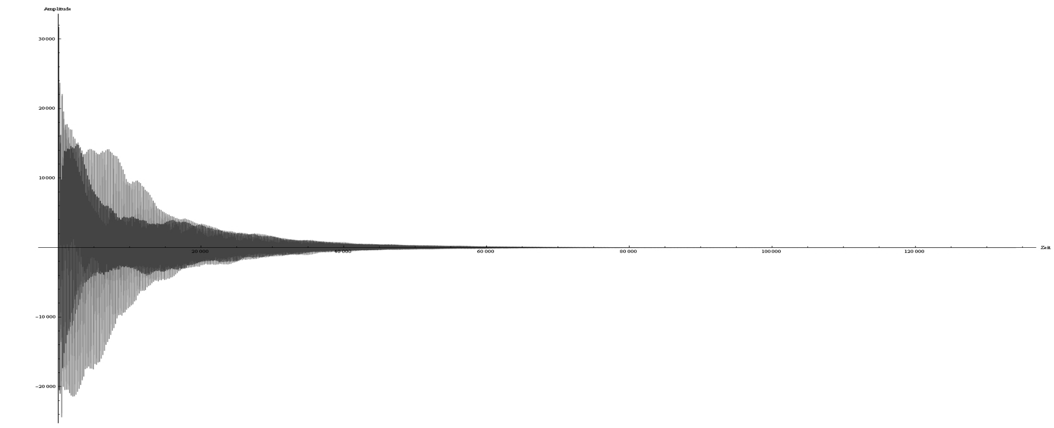


Fig. 3: A physical model of the banjo a) implemented in C++ compared to b) a real recorded sound.

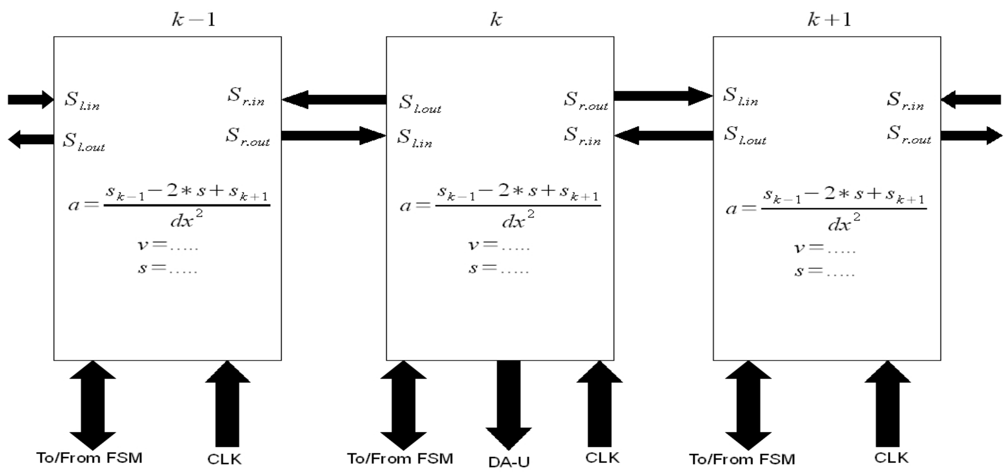


Fig. 4: Finite-Difference Model of the Banjo used for the sound in Fig. 3a.

However, the inherent dependencies of the FD algorithm had to be implemented sequentially. As mentioned above, sequential statements can be realized in a Finite State Machine (FSM) on an FPGA. In Fig. 5 a block diagram of the State Machine and its single states is depicted. As the name suggests the state machine consists of several (in our case six) finite states which are active when the precedent case has finished its task. Every state of the FSM forms one step of the FD algorithm. To use all the parallel processing capabilities of the FPGA, the pipelined FSM-FD algorithm is divided into different subroutines.

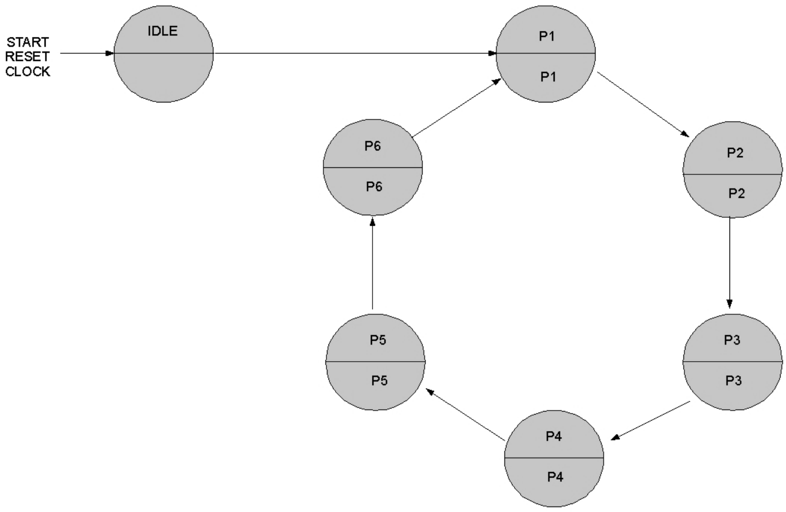


Fig. 5: Finite-state machine of the model

The possible number of nodes which can be computed within one clock cycle depends upon the used hardware device. The hardware described above is able to compute about 50 nodes in 10 clock cycles. The maximum system clock speed is 100 MHz. This means a maximum number of 10,400 nodes for a sampling frequency of 48 kHz.

The banjo model consists of 100 string nodes, 900 membrane nodes, 750 bridge nodes and 400 air nodes. Thus a full-scale model of the banjo with five strings can be modelled with 2,650 nodes, i.e. it can be realized in real time without any restrictions.

The hardware used is a Xinlinx XUP Virtex-2 Pro Development System sponsored by Xilinx company (see Fig. 6 and Fig. 7). The core of this system is a Virtex-2 FPGA Chip with 30,000 logic blocks, 2,448 kB internal Block RAM and 136 18x18 bit multipliers. Besides other features the board is equipped with an AC97 audio codec providing 2 inputs and 2 outputs, a video XSGA output and a 100 MHz system clock. The software used is the latest version of Xilinx ISE Design Suite 10.2 which includes a VHDL/VERILOG editor, a build-, place- and route-tool and a flash-tool. The software used to simulate the model is MentorGraphics Modelsim 6.3f.

A fully functional hardware device has been built that can be attached to the FPGA as a user friendly input-output device. It consists of four voltage controlled (VC) faders, two continuous potentiometers, an alphanumeric LCD display HS286 and a maxim Dallas

8051 Micro Controller. In the current state of development the hardware device communicates with other hardware via the serial RS232-Port using a MIDI protocol.

The aim is to be able to connect a variety of controllers directly to the FPGA Board to influence crucial parameters in real time. Those parameters could be the stiffness of the membrane, the thickness of the banjo string or even the geometry of the bridge, which could then be changed during playing.

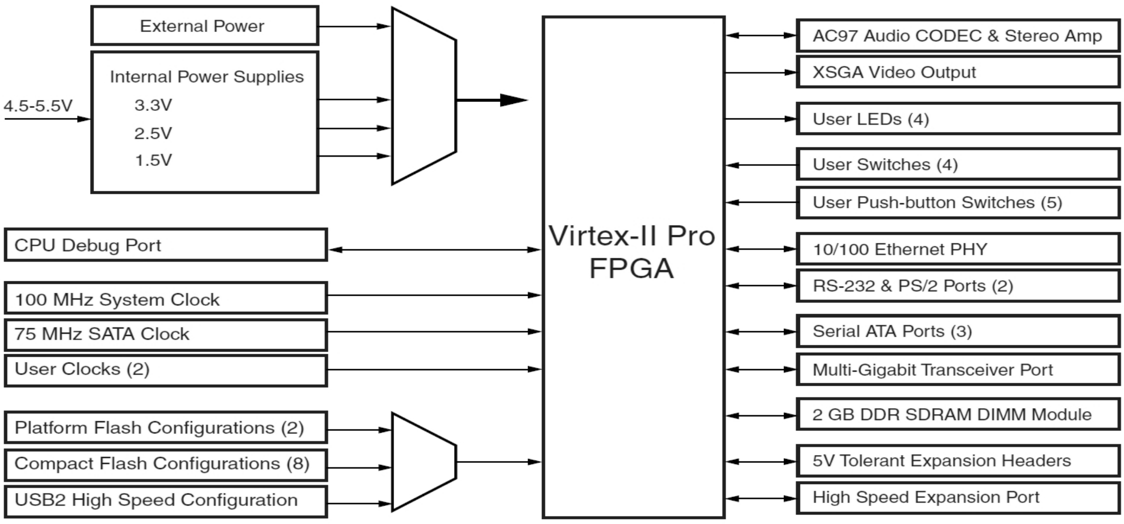


Fig. 6: FPGA board structure

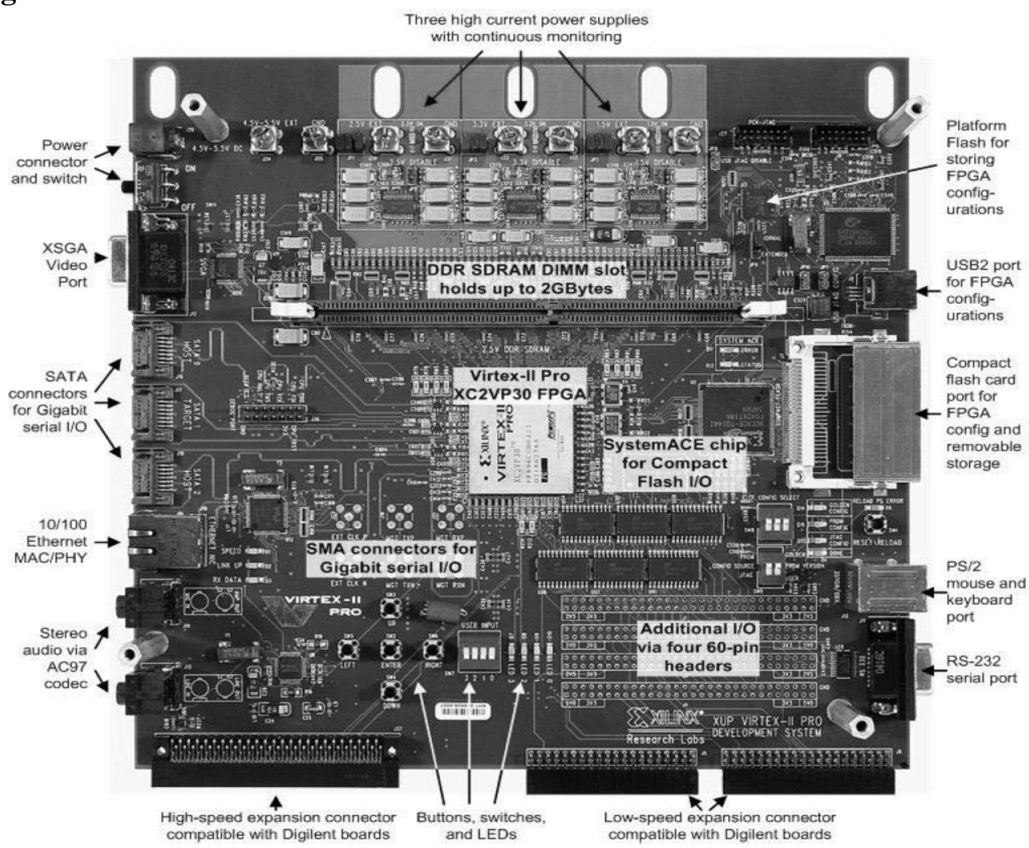


Fig. 7: Xilinx Virtex 2 Pro FPGA Board

Moreover one may even be able to vary the coupling strength between the different parts of the banjo while the banjo-sound is playing. This could deliver many insights into instrument specific parameters and their role in the overall sound. To control the parameters one may use faders, a MIDI keyboard, and any kind of sensors, transducers etc. By using a string-to-MIDI controller, it furthermore would be possible to actually play a real banjo with damped strings which triggers the FPGA model, and so being able to play while changing the instrument geometry in real time.

Conclusion

FPGA technology makes a real-time solution of a complete body model of a musical instrument possible. As via I/O controllers parameters like e.g. geometry, string tension, playing style and articulation can be changed, this synthesis tool is combining the advantages of a sound device of maximum flexibility with the physical properties of a musical instrument. It is therefore the perfect solution to understand the constructional reasons for sound features relevant for music and musicians. Additionally, it might be a helpful tool for instrument builders, because they would be able to listen to different instrumental designs before building them. Concerning the commercial musical production processes this new technology will provide musicians, producers and sound designers with a new gear for sound production. Further studies will include collaborations with musicians and instrument builders to learn about the possibilities and limits of the technique.

References

Bader, R. (2009), Additional modes in a Balinese gender dasa plate due to its trapezoid shape, In: *Concepts, Experiments, and Fieldwork: Studies in Systematic Musicology and Ethnomusicology*, R. Bader, Ch. Neuhaus, U. Morgenstern (Eds.), Hamburg: Peter Lang Verlag, 93-110,

Bader, R., U. Hansen (2008a), Acoustical Analysis and Modeling of Musical Instruments Using Modern Signal Processing Methods, In*: Handbook of Signal Processing in Acoustics*, D. Havelock, M. Vorländer, S. Kuwano (ed.), Berlin, New York: Springer, 219-247.

Bader, R. (2008b), Efficient Auditory Coding of a Xylophone Bar, In: *Hamburger Jahrbuch für Musikwissenschaft*, Hamburg: Peter Lang Verlag, 197-212.

Bader, R. (2008c), Fine tuning of guitar sounds with changed top plate, back plate and rim geometry using a whole body 3D Finite-Difference model, In: *Forum Acusticum joined with American Acoustical Society Paris 08*, 5039-5044.

Bader, R. (2008d), Individual reed characteristics due to changed damping using coupled flow-structure and time-dependent geometry changing Finite-Element calculation, In*: Forum Acusticum joined with American Acoustical Society Paris 08*, 3405-3410.

Bader, R. (2007a), Physical Modeling of the self-sustained oscillation of a Labium using Molecular Dynamic Simulation Techniques, *Proceedings ISMA (International Symposium of Musical Acoustics), Barcelona.*.

Bader, R. (2007b), Spatial cognitive timbre dimensions of physical modelling sounds using Multi-Dimensional Scaling Techniques (MDS), *Proceesings ISMA (International Symposium of Musical Acoustics),* Barcelona 2007.

Bader, R. (2007c), Spatial cognitive timbre dimensions of physical modelling sounds using Multi-Dimensional Scaling Techniques (MDS), *J. Acoust. Soc. Am*. 121, 3184.

Bader, R. (2006a), Finite-element transient calculation of a bell struck by a clapper, with Björn-Andre Lau, Peter Wriggers, Albrecht Schneider, *J. Acoust. Soc. Am*. 119, 3290.

Bader, R. (2006b), Amplitude fluctuations within musical instrument sounds because of impulse reflections, *J. Acoust. Soc. Am*. 119, 3382.

Bader. R. (2006c), Finite-element calculation of a bass drum., *J. Acoust. Soc. Am*. 119, 3290.

Bader, R. (2006d), Characterization of Guitars through Fractal Correclation Dimensions of Initial Transients, *Journal of New Music Research*, Vol. 35, 4, 323-332.

Bader, R (2005a), *Computational Mechanics of the Classical Guitar*, Berlin, New York: Springer.

Bader, R. (2005b), Complete Geometric Computer Simulation of a Classical Guitar, *Lay-Language paper of the American Acoustical Society 05*, http://www.aip.org/149th/bader_Guitar.htm.

Bader, R. (2005c), Identifying Musical Instruments through their Microrhythms, *Lay-Language paper of the American Acoustical Society 05*, http://www.aip.org/149th/bader_microrhythm.htm.

Bader, R. (2005d), Whole geometry Finite-Difference modelling of the violin, *Proceedings of the Forum Acusticum 2005*, 629-634.

Bader, R. (2005e), Nonlinearities in the sound production of the classical guitar, *Proceedings of the Forum Acusticum 2005*, 685-689.

Bader, R. (2005f), Turbulent k-e model of flute-like musical instrument sound production, In: *Fractals in Engineering. New trends in theory and applications*, E. Lutton and J. Lévy-Véhel (Eds.), Berlin, New York: Springer, 109-122.

Bader, R. (2004), Additional modes in transients of a Balinese Gender dasa plate, *J. Acoust. Soc. Am*. 116, 2621.

Bader, R. (2003), Physical model of a complete classical guitar body, In: *Proceedings of the Stockholm Music Acoustics Conference 2003*, R. Bresin (Ed.), Vol. 1, 21-124.

Bader, R. (2002), *Fraktale Dimensionen, Infomationsstrukturen und Mikrorhythmik der Einschwingvorgänge von Musikinstrumenten*, Doktorarbeit der Universität Hamburg, (http://www.sub.uni-hamburg.de/volltexte/2002/598).

Bissinger, J. (2003), Modal Analysis of a Violin Octet, J. *Acoust. Soc. Am*. 113 (4), 2105-2113.

Bécache, E., A. Chaigne, G. Derveaux, P. Joly (2005), Numerical simulation of a guitar, *Computers and Structures*, 83 (2-3), 107-126.

Barthe, K.J. (2002), *Finite-Element Methoden*, Berlin, New York: Springer.

FPGA Parallel Implementation of CMAC Type Neural Network with on Chip Learning*, IEEE 2007*, 111 – 115.

Brich, T., K. Novacek, A. Khateb (2006), The Digital Signal Processing Using FPGA*, IEEE 2006*, 322 – 324.

Chen, P., X. Tian, Y. Chen, X. Yang (2008), Delay-sum Beamforming on FPGA, *ICSP2008 Proceedings*, 2542-2545.

Da Silva, A, G. Scavone, M. van Waltijen (2007), Numerical simulations of fluid-structure interactions in single-reed mouthpieces, *J. Acoust. Soc. Am*. 122 (3), 1798-1809.

Elejabarrieta, M.J., A. Ezcurra, C. Santamaria (2004), Vibrational behaviour of the guitar soundboard analysed by means of finite element analysis, *Acustica united with Acta Acustica*, 87, 128-136.

Elejabarrieta, M.J., A. Ezcurra, C. Santamaria (2002), Coupled modes of the resonance box of the guitar, J. *Acoust. Soc. Am*., 111, 2283-2292.

Essl, G., P. Cook (2000), Measurements and efficient simulation of bowed bars, *J. Acoust. Soc. Am*., 108 (1), 379-388.

Erkut, C., M. Karjalainen, P. Huang, V. Välimäki (2002), Acoustical Analysis and model-based sound synthesis of the kantele, *J. Acoust. Soc. Am*. 112 (4), 1681-91.

Fletcher, N., Th. Rossing (2000), *Physics of Musical Instruments,* Berlin, New York: Springer.

Flügge, Wilhelm (1962), *Statik und Dynamik der Schalen*, Berlin, New York: Springer.

Giordano, N. (2006), Finite-difference modeling of the piano, *J. Acoust. Soc. Am.* 119, 3291.

Hai, C; W. Ping (2002), The High Speed Implementation of Direction-of-Arrival Estimation Algorithm, *IEEE 2002*, 922-925.

Hutchings, C.M. (1981), Klang und Akustik der Geige. *Spektrum der Wissenschaft*, Dezember 1981, 112-122 (original: Scientific American, October 1981).

Jansson, E.V. (2003), The BH-Hill and tonal quality of the violin, *Proc. Stockholm Music Acoust. Conf. (SMAC 03)*, Vol. 1, 71-74.

Karniadakis, George Em, Robert M. Kirby II (2003), *Parallel Scientific Computing in C++ and MPI*, Cambridge: Cambridge University Press.

Knothe, K., H. Wessels (1999), *Finite Elemente. Eine Einführung für Ingenieure*, 3. Auflage, Berlin, New York: Springer.

Lau, A., R. Bader, A. Schneider, P. Wriggers (2009), Finite-Element transient calculation of a bell struck by its clapper. In: *Concepts, Experiments, and Fieldwork: Studies in Systematic Musicology and Ethnomusicology*, R. Bader, Ch. Neuhaus, U. Morgenstern (Eds.), Hamburg: Peter Lang Verlag, 135-154.

Lau, A., P. Spiess, P. Wriggers, A. Schneider, R. Bader (2004), Analysis of bell vibrations. In: *2nd Conference on Advances and Application of GiD*. Barcelona, Feb. 2004, Preprints. Intern. Center for numerical methods in engineering.

Leissa, A.W. (1969a), *Vibration of Plates*, Publication of the American Acoustical Society (1993).

Leissa, A.W. (1969b), *Vibration of Shells*, Publication of the American Acoustical Society (1993).

Lis, J. D., C.T. Kowalski (2008), Parallel Fixed Point FPGA Implementation of Sensorless Induction Motor Torque Control*, IEEE 2008*, 1359 – 1364.

Liu, M., W. Kuehn, Z. Lu, A. Jantsch (2008), System-on-an-FPGA Design for Real-Time Particle Track Recognition in Physics Experiments, 11th EUROMICRO CONFERENCE on DIGITAL SYSTEM DESIGN Architectures, Methods and Tools, *IEEE 2008*, 599 – 605.

Madanayake, A., L. Bruton, F. Comis, C. Comis (2004), FPGA Architectures for Real-Time 2D/3D FIR/IIR Plane Wave Filters*, IEEE 2004*, III 613 – III 616.

Martins, G., M. Barata, L. Gomes (2008), Low Cost Method to Reproduce Sound with FPGA*, IEEE 2008*, 1932 – 1936.

Maslennikow, O., A. Sergiyenko (2006), Mapping DSP Algorithms into FPGA, *IEEE 2006.*

Meyer-Baese, U. (2004): *Digital Signal Processing with Field Programmibalbe Gate Arrays*. Springer.

Özakča, M.T.G. (2004), Structural Analysis and optimization of bells using finite elements, *Journal of New Music Research*, 33, 61-69.

Subasri, V., K. Lavanya, B. Umamaheswari (2006), Implementation of Digital PID controller in Field Programmable Gate Array (FPGA), *Proceedings of India International Conference on Power Electronics 2006*, 172 – 176.

Schneider, A., R. Bader (2003), Akustische Grundlagen Musikalischer Klänge, *Mitteilungen der mathematischen Gesellschaft Hamburg*, Bd XXII, 27-44.

Schoofs, A. (2002), Computer-aided bell design and optimization in the quality of bells, Eurocarillon 2002, Brugge, Sept. 6th, 2002 (*=Proc. of the 16th Meeting of the FWO Research Soc. on Foundations of Music Research*), Ghent: IPEM, Ghent University 2002.

Shlager, K. L., B. Schneider (1995), A Selective Survey of the Finite-Difference Time-Domain Literature, IEEE Antennas and Propagation Magazine, Vol. 37, *IEEE 1995*, 39-56.

Kai, S., X. Yankai, J. Shan, Z. Hongwu (2008), Converting Analog Controllers to Digital Controllers with FPGA, *ICSP2008 Proceedings*, 486-489.

Smith, J. III, Digital Waveguide Modelling and Signal Processing, http://ccrma.stanford.edu/~jos/wg.html.

Smith, J. III (2008), Musical Acoustics: Virtual Instruments I & II', Session at the Joined Meeting of the American Acoustical Society with the European Acoustical Society, Paris 2008, *J. Acoust. Soc. Am*., 123 (5, 2), 3521-23, 3664-66.

Strzdoka, R., D. Göddeke (2006), Pipelined Mixed Precision Algorithms on FPGAs for Fast and Accurate PDE Solvers from Low Precision Components, *IEEE 2006*.

Veggeberg, K., A. Zheng (2009), Real-time noise source identification using programmable gate array (FPGA) technology, *Proceedings of Meetings on Acoustics, Vol 5., Acoustical Society of America*, 1 – 10.

Von Herzen, B. (1998), Signal Processing at 250 MHz Using High-Performance FPGA's*, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, VOL. 6, NO. 2, 238 – 246.

Wagner, K.W. (1947), *Einführung in die Lehre von den Schwingungen und Wellen*, Wiesbaden.

Wang, Z., R. Jin, J. Geng, Y. Fan (2005), FPGA implementation of Downlink DBF Calibration, *IEEE 2005*.

Woodhouse, J. (2005), *O*n the „Bridge Hill“ of the Violin, *Acta Acustica united with Acustica,* 91, 155-65.

Zou, Z., W. Hongyuan, W., Y. Guowen (2006), An Improved MUSIC Algorithm Implemented with High -speed Parallel Optimization for FPGA, *IEEE 2006*.