NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
«IGOR SIKORSKY POLYTECHNICAL INSTITUTE»

Informatic and Computer Engineering Faculty
Computer Engineering Department

«On the rights of the manuscript»
УДК __004.942_____

«Defence is allowed»
Head of the Computer Science Dep-t

_____ S.G. Stirenko
(sign) (name)

“___”_____2018p.

Master's dissertation

In speciality _____123 Computer Engineering_____

Specialization: _____123. Computer systems and networks _____

theme: __Method of hardware simulation of the propagation
_____ of ultrasonic waves in a solid _____

Fulfilled: student of VI course, group _ IO 64м _____
(group sign)

_____ Mustafa Rekar Quasim _____ _____
(Full Name) (signature)

Науковий керівник Ass.Prof., Dr.Sci, S.Sci. Sergiyenko A.M. _____ _____
(position, scientific degree, academic rank, surname and initials) (signature)

Reviewer Ass.Prof., Dr.Sci, Docent Romankevich V.O. _____ _____
(position, scientific degree, academic rank, surname and initials) (signature)

I certify that in this master's thesis there are no borrowings from the works of other authors without the corresponding references.
Student _____
(signature)

Kyiv – 2018

РЕФЕРАТ

Метод апаратного моделювання поширення ультразвукових хвиль у твердому тілі

Актуальність теми. Чисельне моделювання фізичного явища поширення хвиль є основою багатьох досліджень та впроваджень у галузях акустики, ультразвукової техніки, радіоелектроніки. Апаратна реалізація такого моделювання дає змогу його суттєво прискорити і покращити ефективність приладів, які використовують хвилеві процеси. Сучасні засоби такого моделювання, як правило, використовують системи на основі графічних акселераторів, які виконують паралельні обчислення з плаваючою комою. Їх недоліками є висока вартість, велике енергоспоживання, невисока надійність. Застосування програмованих логічних інтегральних схем (ПЛІС) для моделювання хвилевих процесів дає змогу збільшити швидкодію, зменшити енергоспоживання та підвищити надійність засобів, що використовують таке моделювання. Причому моделювання ультразвукових процесів у реальному часі дозволяє створювати засоби ультразвукової діагностики з підвищеною якістю вимірюваних результатів.

Об’єктом дослідження є високопродуктивні процесори для моделювання фізичних явищ.

Предметом дослідження є алгоритми та структури конвеєрних спецпроцесорів для моделювання поширення хвиль у твердому тілі.

Мета роботи: створення методу проектування високопродуктивних процесорів моделювання акустичних явищ, призначених для конфігурування в ПЛІС.

Наукова новизна полягає в наступному:

1. Запропоновано удосконалений хвилевий алгоритм моделювання поширення ультразвуку, який полягає у представленні середовища у вигляді системи хвилевих фільтрів та відрізняється тим, що завдяки реалізації

багатоканальних фільтрів з програмованою затримкою, зменшується похибка моделювання дисперсійного поширення звуку.

2. Розроблено метод апаратного моделювання поширення ультразвукових хвиль у твердому тілі, який оснований на реалізації системи хвилевих фільтрів у ПЛІС і відрізняється високою швидкодією за рахунок того, що фільтри мають конвеєрну структуру і використовують особливості архітектури ПЛІС.

Практична цінність отриманих в роботі результатів полягає в тому, що система моделювання, яка побудована за запропонованим методом, може бути основою приладів ультразвукової діагностики нового покоління, які здатні знаходити і локалізувати неоднорідності більш точно.

Матеріали роботи використані у науково-дослідній роботі «Удосконалені методи та засоби проектування конфігурованих комп'ютерів на основі відображення просторового графу синхронних потоків даних у структури на базі програмованих логічних інтегральних схем», № ДР.047U005087, шифр ФІОТ-30Т/2017, яка проводиться у НТУУ "КПІ ім. Ігоря Сікорського.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на 20-тій Міжнародній конференції «Системний аналіз та інформаційні технології» SAIT-2018 21 – 24 травня 2018 року, Київ та міжнародній конференції "Безпека, Відмовостійкість, Інтелект" 10 – 12 травня 2018 року, Київ.

Структура та обсяг роботи. Магістерська дисертація складається зі вступу, трьох розділів та висновків.

У вступі подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову новизну

отриманих результатів і практичну цінність роботи, наведено відомості про апробацію результатів і їхнє впровадження.

У першому розділі розглянуто алгоритми моделювання хвилевих процесів та їх відомі реалізації в паралельних обчислювальних системах і ПЛІС.

У другому розділі розроблено хвилевий алгоритм моделювання поширення ультразвуку та метод апаратного моделювання поширення ультразвукових хвиль у твердому тілі

У третьому розділі досліджено ефективність використання запропонованих алгоритму моделювання поширення ультразвуку та метод апаратного моделювання на прикладі моделювання поширення ультразукових хвиль у циліндричному тілі.

У висновках представлені результати проведеної роботи.

Робота представлена на 80 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: ПЛІС, дисперсія, ультразвук, граф синхронних потоків даних, хвилевий фільтр, конвеєр.

ABSTRACT

Method of hardware simulation of the propagation of ultrasonic waves in a solid

Relevance of the topic. The numerical modeling of the physical phenomenon of the wave propagation is the basis of many studies and implementations at the fields of acoustics, ultrasound technology, and radio electronics. The hardware implementation of such a simulation allows us to significantly accelerate and improve the efficiency of devices that use the wave propagation processing. The modern wave propagation simulators use typically the graphic processing units that perform parallel floating-point calculations. Their drawbacks are high cost, high power consumption, and low reliability. The use of the field programmable gate arrays (FPGAs) to simulate the wave processes allows us to increase speed, reduce power consumption of tools that use this simulation. Moreover, the real time simulation of the ultrasonic processes allows us to create the means of ultrasound diagnostics with increased quality of measured results.

The object of the research is the high-performance processors for modeling the physical phenomena.

The subject of the research is the algorithms and structures of application-specific pipelined processors for modeling the propagation of waves in a solid.

The objective is the creation of a method for designing the high-performance processors for the simulation of the acoustic phenomena, which is intended for configuring in FPGA.

The scientific novelty is as follows:

1. The ultrasound propagation wave modeling algorithm is improved, which consists in representing the medium in the form of a system of the wave digital filters and differs in that due to the implementation of the multichannel filters with the programmable delays, resulting the decrease of the simulation error of the sound propagation dispersion.

2. A method for the hardware simulation of the ultrasonic wave propagation in a solid is developed, which is based on the implementation of the system of the wave digital filters in the FPGA and is characterized by high speed due to the fact that the filters have a pipelined structure and use the features of the FPGA architecture.

The practical value of the obtained results is that the simulation system, built on the proposed method, can be the basis of a new generation of the ultrasonic diagnostic devices that are able to localize the heterogeneities more precisely.

The materials of the thesis were used in the research work "Advanced methods and tools of designing the configurable computers on the basis of mapping the spatial synchronous data flow graphs into the structure for FPGA", № ДР.047U005087, ФІОТ-30Т / 2017, which is held at NTUU “Igor Sikorsky’s KPI”.

Approbation of the work. Substantive provisions and results of the work were presented and discussed at the 20-th International conference “System Analysis and Informational Technologies”, SAIT-2018, May, 21 – 24, 2018, Kyiv, and at the International Conference on Security, Fault Tolerance, Intelligence (ICSFTI2018), May 10 – 12, 2018, Kyiv.

The structure and scope of the work. Master's thesis consists of an introduction, four sections and conclusions.

The introduction provides a general description of the work carried out to assess the current state of the problem, justified the relevance of research areas, formulate goals and objectives of research shows the scientific novelty of the obtained results and the practical value of the work, it shows the information about the testing results and their implementation.

The first section deals with the basic methods of designing the pipelined application-specific processors, algorithms for the numerical modeling of the physical phenomenon of the wave propagation in the rigid body.

The second section describes a general method for the hardware simulation of the ultrasonic wave propagation in a solid.

The third section describes the experimental results of the wave propagation simulation in the designed processor.

The conclusions describe the results of the work.

The work submitted 80 pages, contains a list of references to the used literature.

Key words: FPGA, dispersion, ultrasound, synchronous data flow graph, wave digital filter, pipeline.

РЕФЕРАТ

Метод аппаратного моделирования распространения ультразвуковых волн в твердом теле

Актуальность темы. Численное моделирование физического явления распространения волн лежит в основе многих исследований и внедрений в области акустики, ультразвуковой техники, радиоэлектроники. Аппаратная реализация такого моделирования позволяет существенно ускорить его и улучшить эффективность приборов, использующих процессы распространения волн. Современные средства такого моделирования, как правило, используют системы на основе графических акселераторов, которые выполняют параллельные вычисления с плавающей запятой. Их недостатками являются высокая стоимость, большое энергопотребление, невысокая надежность. Применение программируемых логических интегральных схем (ПЛИС) для моделирования волновых процессов позволяет увеличить быстродействие, уменьшить энергопотребление и повысить надежность средств, использующих такое моделирование. Причем моделирование ультразвуковых процессов в реальном времени позволяет создавать средства ультразвуковой диагностики с повышенным качеством измеряемых результатов.

Объектом исследования являются высокопроизводительные процессоры для моделирования физических явлений.

Предметом исследования являються алгоритмы и структуры конвейерных спецпроцессоров для моделирования распространения волн в твердом теле.

Цель работы: создание метода проектирования высокопроизводительных процессоров моделирования акустических явлений, предназначенных для конфигурирования в ПЛИС.

Научная новизна заключается в следующем:

1. Предложен усовершенствованный волновой алгоритм моделирования распространения ультразвука, который заключается в представлении среды в виде системы волновых фильтров и отличается тем, что благодаря реализации многоканальных фильтров с программируемой задержкой, уменьшается погрешность моделирования дисперсионной распространения звука.

2. Разработан метод аппаратного моделирования распространения ультразвуковых волн в твердом теле, основанный на реализации системы волновых фильтров в ПЛИС и отличается высоким быстродействием за счет того, что фильтры имеют конвейерную структуру и используют особенности архитектуры ПЛИС.

Практическая ценность полученных в работе результатов заключается в том, что система моделирования, которая построена по предложенному методу, может быть основой приборов ультразвуковой диагностики нового поколения, которые способны находить и локализовать неоднородности более точно.

Материалы работы использованы в научно-исследовательской работе «Усовершенствованные методы и средства проектирования конфигурируемых компьютеров на основе отображения пространственного графа синхронных потоков данных в структуры на базе программируемых логических интегральных схем», № ДР.047U005087, шифр ФІОТ-30Т / 2017, которая проводится в НТУУ "КПИ им. Игоря Сикорского".

Апробация работы. Основные положения и результаты работы были представлены и обсуждались на 20-ой Международной конференции «Системный анализ и информационные технологии» SAIT-2018, 21 — 24 мая 2018, Киев и международной конференции "Безопасность, Отказоустойчивость, Интеллект" 10 — 12 мая 2018, Киев.

Структура и объем работы. Магистерская диссертация состоит из введения, трех глав и выводов.

Во введении представлена общая характеристика работы, произведена оценка современного состояния проблемы, обоснована актуальность направления исследований, сформулированы цели и задачи исследований, показано научную новизну полученных результатов и практическую ценность работы, приведены сведения об апробации результатов и их внедрение.

В первом разделе рассмотрены алгоритмы моделирования волновых процессов и их известные реализации в параллельных вычислительных системах и ПЛИС.

Во втором разделе разработаны волновой алгоритм моделирования распространения ультразвука и метод аппаратного моделирования распространения ультразвуковых волн в твердом теле

В третьем разделе исследована эффективность использования предложенных алгоритма моделирования распространения ультразвука и метод аппаратного моделирования на примере моделирования распространения ультразуковое волн в цилиндрическом теле.

В выводах представлены результаты проведенной работы.

Работа представлена на 80 листах, содержит ссылки на список использованных литературных источников.

Ключевые слова: ПЛИС, дисперсия, ультразвук, граф синхронных потоков данных, волновой фильтр, конвейер.

CONTENT

ABBREVIATIONS

DSP	Digital Signal Processing
FPGA	field programmable gate array
GPU	Graphic Processing Uunit
RTL	Register Transfer Logic
VHSIC	Very High Speed Integrated Circuits
VHDL	VHSIC Hardware Description Language
CPU	Central Processing Unit
VLSI	Very Large Scale Integration
IC	Integrated Circuits
ASIC	Application Specific Integrated Circuit

INTRODUCTION

The numerical modeling of the physical phenomenon of the wave propagation is the basis of many studies and implementations at the fields of acoustics, ultrasound technology, and radio electronics. The hardware implementation of such a simulation allows us to significantly accelerate and improve the efficiency of devices that use the wave propagation processing. The modern wave propagation simulators use typically the graphic processing units that perform parallel floating-point calculations. Their drawbacks are high cost, high power consumption, and low reliability.

So far three main approaches have been widely used: digital waveguide modeling, modal modeling and finite element methods. Digital waveguide modeling is efficient and realistic, and captures the complete dynamics of the underlying physics but is restricted to the bodies that are well-described by the one-dimensional wave propagation equation. Modal synthesis is efficient yet abandons complete dynamical description and hence cannot used for certain types of objects. Finite element methods are realistic and capture the behavior of the constituent physical equations but on current commodity hardware does not perform in real-time.

The digital waveguides offer efficient simulations for many cases for which the wave scattering and dispersion is not taken into account. The key realization is that the dynamic behavior of traveling waves, which is being used in the waveguide synthesis, can be applied to individual modes and that the efficient computational structure can be utilized to achieve an approximate dynamical description in the neighborhood of modes. Such approach provides both the realistic sound propagation modeling and hardware implementation, which utilizes the configurable resources of middle volume field programmable gate array (FPGA).

This thesis proposes the digital waveguide synthesis, and its implementation in FPGA as an approach to the real-time ultrasound process modeling, which is based on the underlying physics.

The use of FPGAs to simulate the wave processes allows us to increase speed, reduce power consumption of tools that use this simulation. Moreover, the real time simulation of the ultrasonic processes allows us to create the means of ultrasound diagnostics with increased quality of measured results.

The object of the research is the high-performance processors for modeling the physical phenomena.

The subject of the research is the structure of application-specific pipelined processors for modeling the propagation of waves in a solid.

The objective is the creation of a method for designing the high-performance processors for the simulation of the acoustic phenomena, which is intended for configuring in FPGA.

To achieve the objective, the following tasks are solved in the dissertation:

1. The methods of the mathematical modeling of the wave propagation in solids, and their comuter implementation are analysed.

2. The method of the waveguide modeling is analysed and its application to the modeling the solids is investigated.

3. The method of hardware simulation of the propagation of ultrasonic waves in a solid based on the waveguide models is developed.

4. The method of hardware simulation the propagation of ultrasonic waves is adapted for its implementation in modern FPGAs.

5. The proposed method effectiveness is proven by modeling of the wave propagations in the solid rod.

The research methods used in the work are based on the theory of graphs, algorithm theory, modeling theory, theory of the wave propagation in solids, combinatorial optimization methods, as well as theorems, assertions and implications that are proved in the dissertation. The main provisions and theoretical evaluations are confirmed by the results of simulation on a computer, as well as by tests of a number of experimental and experimental samples of specialized calculators.

Experimental verification of scientific positions, proposals and results was carried out by designing computing tools by the developed method using their description in standard VHDL language with their further simulation in the simulator, compiling in the circuit and configuring the Xilinx FPGA.

The scientific novelty is as follows:

1. The ultrasound propagation wave modeling algorithm is improved, which consists in representing the medium in the form of a system of the wave digital filters and differs in that due to the implementation of the multichannel filters with the programmable delays, resulting the decrease of the simulation error of the sound propagation dispersion.

2. A method for the hardware simulation of the ultrasonic wave propagation in a solid is developed, which is based on the implementation of the system of the wave digital filters in the FPGA and is characterized by high speed due to the fact that the filters have a pipelined structure and use the features of the FPGA architecture.

The practical value of the obtained results is that the simulation system, built on the proposed method, can be the basis of a new generation of the ultrasonic diagnostic devices that are able to localize the heterogeneities more precisely.

The materials of the thesis were used in the research work "Advanced methods and tools of designing the configurable computers on the basis of mapping the spatial synchronous data flow graphs into the structure for FPGA", № ДР.047U005087, ФІОТ-30Т / 2017, which is held at NTUU “Igor Sikorsky’s KPI”.

Approbation of the work.

Substantive provisions and results of the work were presented and discussed at the 20-th International conference “System Analysis and Informational Technologies”, SAIT-2018, May, 21 – 24, 2018, Kyiv, and at the International Conference on Security, Fault Tolerance, Intelligence (ICSFTI2018), May 10 – 12, 2018, Kyiv.

Publications of the work.

The main features of these investigations are published in two works. In the work [1] the author has proposed an approach, which provides the modeling the sound dispersion. In the work [2] the author has proposed the way to connect the digital waveguides in a system, and provides the experimental results.

The structure and scope of the work.

Master's thesis consists of an introduction, four sections and conclusions.

The introduction provides a general description of the work carried out to assess the current state of the problem, justified the relevance of research areas, formulate goals and objectives of research shows the scientific novelty of the obtained results and the practical value of the work, it shows the information about the testing results and their implementation.

The first section deals with the basic methods of designing the pipelined application-specific processors, algorithms for the numerical modeling of the physical phenomenon of the wave propagation in the rigid body.

The second section describes a general method for the hardware simulation of the ultrasonic wave propagation in a solid.

The third section describes the experimental results of the wave propagation simulation in the designed processor.

The conclusions describe the results of the work.

The work submitted 93 sheets, contains a list of references to the used literature.

1 METHODS AND TOOLS FOR THE DIGITAL WAVE PROPAGATION MODELING

1.1 Basics of the mathematical modeling

It is impossible to imagine the modern science without a broad application of the mathematical modeling. The essence of this methodology consists in replacing the original object with its "image", which is the mathematical model, and in further study of the model with the help of the algorithms implemented on the modern computers. This method of cognition and design combines many of the virtues of both the theory and the experiment. The investigator works not with the object itself (phenomenon, process), but with its model. And this makes it possible to study its properties and behavior painlessly, relatively quickly and without significant expenditure in any conceivable situations.

At the same time, the computer simulation experiments with the object models allow us to study the objects in detail and deeply in sufficient completeness that is inaccessible to purely theoretical approaches. This experiment is based on the extremely high power of the modern computing methods and technical tools of informatics. It is not surprising that the methodology of mathematical modeling is rapidly developing, covering all new areas — from the development of technical systems and their management to the analysis of the most complicated economic and social processes [3].

Technical, ecological, economic and other systems, which are studied by the modern science are no longer subject to research (in the required completeness and accuracy) by conventional theoretical methods. A direct full-scale experiment on them is long, expensive, often either dangerous, or simply impossible, since many of these systems exist in a "single copy." The price of mistakes and miscalculations in handling them is unacceptably high. Therefore, the mathematical and information modeling is an inevitable component of scientific and technological progress.

The modeling method has a set of definitions.

Analogy is the establishing some similarities in the properties of the unidentical objects. The conclusion by the analogy is based on the detected similarities. The analogy gives not the reliable but the probabilistic knowledge. In the conclusion by the analogy, the knowledge, which is derived from consideration of a particular object ("model"), is transferred to other, less explored and less available object under research.

Modeling is a method of research the objects using their models.

Model is an analog of a particular fragment of reality.

Due to the nature of the model the following modelings are distinguished:

— material (substantive) modeling;

— perfect modeling, which is performed using formulas, algorithms, etc, i.e. the mathematical modeling.

System approach is a combination of general methodological principles (requirements), which are based on consideration of the object as a system. Among these requirements are:

a) identifying the dependencies of each element on its place and function in the system, taking into account that a property can not be reduced to the sum of the properties of these elements;

b) analysis of how the behavior of the system is due to features of its individual elements and properties of its structure;

c) study of the mechanism of interaction of the system and the environment;

d) study of the hierarchy inherent in the system;

e) comprehensive and multidimensional description of the system;

g) consideration of the system as a dynamic integrity which is under developing [5].

The formulation of the mathematical modeling of an object generates a clear plan of action. It can be conditionally divided into three stages: the model, the algorithm, and the program.

At the first stage, the "equivalent" of the object is chosen, or constructed, reflecting in its mathematical form the most important properties of the laws to which it obeys, the connections inherent in its constituent parts, etc. The mathematical model is studied by theoretical methods, which allows the investigator to get the important preliminary knowledge about the object.

The second stage is the choice or development of the algorithm for implementing the model on the computer. The model is presented in a form, which is convenient for the application of the numerical methods. Next, a sequence of computational and logical operations is determined, which must be done in order to find the desired quantities with a given accuracy. The computational algorithms should not distort the basic properties of the model and, consequently, of the original object, be economically effective and adapted to the specific features of the tasks being solved and the computers used.

At the third stage, the programs are created that "translate" the model and algorithm into a computer-accessible language. They are also subject to the requirements of economy and adaptability. They can be called the "electronic" equivalent of the studied object, already suitable for direct testing on an "experimental setup" — a computer. At this stage, the application-specific computer can substantially improve the economic and/or time limitations of the mathematical experiment.

Having created the triad "model-algorithm-program", the researcher gets a universal, flexible and inexpensive tool. At first, this tool is debugged and tested in the testing computing experiments. Then, the adequacy, or sufficient correspondence of the triad to the source object is certified. And finally, various and detailed experiments are implemented with the model, giving all the required qualitative and quantitative properties and characteristics of the object. The modeling process is accompanied by the improvement and refinement, as necessary, of all links of the triad "model-algorithm-program".

The mathematical modeling is based on a set of the following principles.

1) *Fundamental laws of the nature*

The most common method of constructing a model is to apply the fundamental laws of the nature to a specific situation. These laws are generally recognized, repeatedly confirmed by the experience, they are the basis of many scientific and technical achievements. Therefore, their validity is beyond doubt, which, among other things, provides the researcher with the powerful support. On the foreground are questions related to which laws should be applied in this case and how to do it. The most common fundamental laws are the following:

— conservation of energy,

— conservation of matter,

— persistence of momentum.

A particular case of the conservation of matter is the Kirchhoff's law, according to which the sum of the flows of matter entering a certain node is equal to the sum of the outflows.

2) *Variational principle*

Another approach to constructing the models is to apply the so-called variational principle. It represents very general statements about the object under consideration (the system, the phenomenon) and says that of all the possible variants of its behavior (movement, evolution), only those are selected, that satisfy a certain condition. Usually, according to this condition, some value associated with the object reaches an extreme value when it changes from one state to another.

3) *Application of analogies*

In many situations of a model design, either it is impossible to directly indicate the fundamental laws or variational principles, or, there is no certainty about the existence of such laws permitting mathematical formulation. When trying to construct a model of an object in these conditions, one of the fruitful approaches to these kinds of objects is the use of analogies with already studied phenomena.

4) *Hierarchical approach*

Only in rare cases it is convenient and justified to construct the mathematical models of even relatively simple objects at once in their entirety, taking into account all the factors essential for its behavior. Therefore, a natural approach is usually used, that realizes the principle of "from simple to complex", when the next step is done after a sufficiently detailed study of a not very complex model. In this case, a chain, of hierarchy of more and more complete models appears, each of which generalizes the previous ones, including them as a special case.

5) *Nonlinearity in the mathematical models*

The simplicity of most models is largely related to their linearity. Mathematically, this important concept means that the superposition principle holds, that is, any linear combination of solutions (for example, their sum) is also a solution of the problem. Using the principle of superposition, it is not difficult to find a solution in any particular case, to construct a solution in a more general situation. Therefore, the qualitative properties of the general case can be judged from the properties of the particular case, i.e. the difference between the two solutions is only quantitative. In the case of linear models, the response of an object to a change in some conditions is proportional to the magnitude of this change.

For the nonlinear phenomena, whose mathematical models do not obey the superposition principle, the knowledge about the behavior of a part of the object does not yet guarantee knowledge of the behavior of the entire object, and its response to changing conditions can qualitatively depend on the magnitude of this change.

Most of the real processes and the corresponding mathematical models are nonlinear. Linear models also correspond to very special cases and, as a rule, serve only as a first approximation to reality.

The process of constructing models can be conditionally divided into the following stages.

1) The construction of the model begins with a verbal and semantic description of the object or phenomenon. In addition to general information about the

nature of the object and the purposes of its investigation, this stage may also contain certain assumptions (a weightless rod, a thick layer of matter, rectilinear propagation of sound rays, etc.). This stage can be called the formulation of the pre-model.

2) The completion of the idealization of the object. All factors and effects that appear to be not the most significant for his behavior are discarded. If possible, idealizing assumptions are written in mathematical form so that their fairness can be quantified.

3) The choice or formulation of a law (variational principle, analogy, etc.), to which the object obeys, and its recording in mathematical form. If necessary, additional information is used to the object, also written mathematically (for example, the constancy of the value of the phase frequency for all trajectories of the sound rays, which follows from the geometry of the problem). It should be borne in mind that even for simple objects the choice of the corresponding law is by no means a trivial task.

4. Arrangement of the boundary conditions of the model. For example, the investigator needs to specify the information about the initial state of the object or other characteristics of the object without knowing which it is impossible to determine the behavior of the object. Finally, the goal is to study the model (to find a law, to determine the requirements for the modeled design etc.).

5. The constructed model is studied by all methods available to the researcher, including with mutual verification of various approaches. Unlike the simplest cases, most models do not lend themselves to a purely theoretical analysis, and therefore it is necessary to make extensive use of computational methods.

As a result of the study of the model, not only the goal is achieved, but it must be established in all possible ways (by comparison with practice, comparison with other approaches) its adequacy. The ***adequacy*** is the conformity to the object and the formulated assumptions. An inadequate model can produce a result that is arbitrarily different from the true and the result and must either be discarded or appropriately modified.

1.2 Finite difference schema and the finite element method

There is a very wide class of models, those that reduce the differential equations to the finite difference equations. For them, the process of creating the computational algorithm consists of two main stages. On the first stage, the discrete analogue of the original model is constructed and its properties are studied. On the second stage, the discrete equations are solved numerically.

The transition from a continuous to a discrete model is divided into two stages. On the first stage, the modeled domain is quantized. In this process, a finite number of points are selected. In the one dimensional model this number is N. In the two dimensional model this number is $N_1 \cdot N_2$, and so on. The simplest method is uniform division. The set of these points is a uniform difference grid with step h, the points $\boldsymbol{X}_i$ are its nodes. All functions appearing in the continuous model are now considered as the grid functions of a non-continuous argument, and a discrete argument.

In the second stage, discrete analogues of the differential equation and input data are constructed. The most natural discretization of a differential operator is the replacement of the derivatives by the corresponding finite differences. For the wave propagation modeling, the step h is usually selected in the range of $(0.01 — 0.2)\lambda$, depending on the model precision and stability, where λ is the considered minimum wave length of the sound, electromagnetic or other field [4].

Thus, at least two requirements are required for discrete models: approximation of the initial model and its stability. Then for a sufficiently accurate numerical solution of the difference equations (as a rule, they are systems of linear and nonlinear algebraic equations of order N, where N is the number of grid nodes) and sufficiently small steps yields a fairly accurate approximate solution [6].

Consider the 2-dimensional model of a bar, which dimensions are $100\lambda \cdot 10\lambda \cdot 10\lambda$. Then, considering the minimum step is $h = 0.01\lambda$, the grid number is equal to $N = 10^{10}$ nodes. Consider the node step is calculated for 100 clock cycles of the processor taking into account the data interchanges between the nodes. Then, a

single modeling step is calculated on a single modern processor unit approximately for 1000 seconds, and a single modeling experiment will last ca. 6000 hours. This means that in this situation when the ultrasound wave propagation is investigated, the modeling is implemented in the time scale 1 to $5 \cdot 10^{13}$. This is much far from the real time modeling. And this ratio could not be decreased dramatically even by the utilization of the GPU units or supercomputers.

Moreover, in the case of the nonlinear equations the number of the modeling steps has to be increased in many times, because these equations are solved by the respective methods of successive approximations or iterative procedures [3,6].

As a result, it has be emphasized that the number of nodes cannot be too large (h is not too small), since a numerical solution must be found for an acceptable number of operations, i.e. using the real grids.

And even the node number and the step h are selected as an optimum value, then it is obviously that the modeling of the wave propagation in the solid body could not be implemented in real time on the base of the Finite difference schema.

1.3 Method of the digital waveguide modeling

The developed in eighteens digital waveguide method follows a different path to the physical model: the wave equation is first solved in a general way to obtain the traveling waves in the medium interior. The traveling waves are explicitly simulated in the waveguide model, in contrast to computing a physical variable. The traveling waves must be summed together to produce a physical output. In the lossless case, a traveling wave between two points in the medium can be simulated using nothing but a digital delay line. In the general linear case, in which there are frequency-dependent losses and dispersion, the commutativity of linear time-invariant systems allows the losses and dispersion to be lumped at discrete points such that most of the simulation still consists only simple delay lines. This is essentially why computational costs are so low in the waveguide synthesis algorithms [7].

The Morse wave equation for the ideal lossless, linear, flexible body, depicted in Fig. 1.1, is given by

$$K\frac{d^2}{dt^2}y(t,x) = \varepsilon \frac{d^2}{dx^2} y;$$

where K is string tension,

ε is linear mass density,

y string displacement.

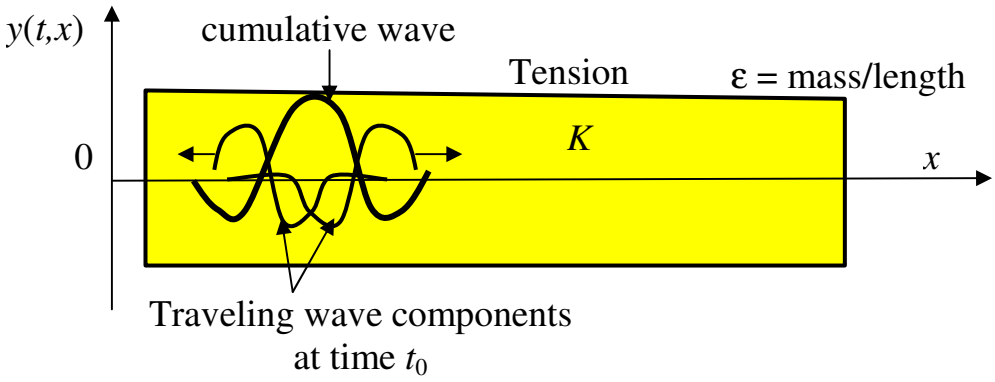

Fig.1.1. The ideal vibrating bar

The same wave equation applies to any perfectly elastic medium, which is displaced along one dimension. For example, the air column of a clarinet or organ pipe can be modeled using the one-dimensional wave equation, substituting air pressure deviation for string displacement, and longitudinal volume velocity of air in the bore for transverse velocity on the string. We refer to the general class of such media as one-dimensional waveguides. Extensions to two and three dimensions and more are also possible.

It can be readily checked that the wave equation is solved by any wave shape which travels to the left or right with speed $c = \sqrt{K/\varepsilon}$. If we denote right-going traveling waves by $y_r(x - ct)$, and left-going traveling waves by $y_l(x + ct)$, where y_r

and y_l are arbitrary twice-differentiable functions, then the general class of solutions to the lossless , one-dimensional, second-order wave equation can be expressed as

$$y(t,x) = y_r(x - ct) + y_l(x + ct).$$

To carry the traveling-wave solution into the digital domain, it is necessary to sample the traveling-wave amplitudes at intervals of T seconds, corresponding to a sampling rate $f_s = 1/T$ samples per second. The natural choice of spatial sampling interval X is the distance sound propagates in one temporal sampling interval T, or $X = cT$ meters. In a traveling-wave simulation, the whole wave moves left or right one spatial sample each time sample; hence, simulation only requires digital delay lines.

Formally, sampling is carried out by the change of variables

$$x \rightarrow x_m = mX.$$

$$t \rightarrow t_n = nT.$$

Substituting into the traveling-wave solution of the wave equation gives

$$y(t_n,x_m) = y_r\,(t_n - x_m/c) + y_l\,(t_n - x_m/c) =$$

$$= y_r\,[(n - m)T] + y_l\,[(n + m)T]. \qquad (1.1)$$

Since T multiplies all arguments, let 's suppress it by defining

$y^+(n) = y_r\,(nT)$ $\qquad$ $y^-(n) = y_l\,(nT)$

This notation also introduces a "+" superscript to denote a traveling-wave component propagating to the right , and a "-" superscript to denote propagation to the left.

The term $y_r\,[(n - m)T] = y^+(n - m)$ can be thought of as the output of an m-sample delay line whose input is $y^+(n)$. In general, subtracting a positive number m from a time argument n corresponds to delaying the waveform by m samples. Since y^+ is the right-going component, its delay line is placed with input $y^+(n)$ on the left and its output $y^+(n - m)$ on the right. This can be seen as the upper rail in Fig. 1.2.

Similarly, the term $y_r\,[(n + m)T] = y^-(n + m)$ can be thought of as the input to an m-sample delay line whose output is $y^-(n)$. Adding m to the time argument n produces an m-sample waveform advance. Since y, is the left-going component , it

makes sense to draw the delay line with its input $y^-(n + m)$ on the right and its output $y^-(n)$ on the left . This can be seen as the lower rail in Fig. 1.2. Note that the position along the bar, or string, $x_m = mX = mcT$ meters, is laid out from left to right in the diagram, giving a physical interpretation to the horizontal direction in the diagram. Finally, the left- and right-going traveling waves must be summed to produce a physical output according to the formula

$$y(t_n, x_m) = y^+(n — m) + y^-(n + m). \quad (1.2)$$

We may compute the physical string displacement at any spatial sampling point xm by simply adding the upper and lower rails together at position *m* along the delay-line pair. In Fig. 2, the transverse displacement outputs have been arbitrarily placed at $x = 0$ and $x = 3X$.

A more compact simulation diagram, which stands for either sampled or continuous simulation is shown in Fig. 3. The figure emphasizes, that the ideal, lossless waveguide is simulated by a bidirectional delay line.

Any ideal one-dimensional waveguide can be simulated in this way. It is important to note that the simulation is exact at the sampling instants, to within the numerical precision of the samples themselves provided that the wave shapes traveling along the string are initially band limited to less than half the sampling frequency. In other words, the highest frequencies present in the signals $y_r(t)$ and $y_l(t)$ may not exceed half the temporal sampling frequency $f_s = 1/T$; equivalently, the highest spatial frequencies in the shapes $y_r(x/c)$ and $y_l(x/c)$ may not exceed half the spatial sampling frequency $v_S = 1/X$.

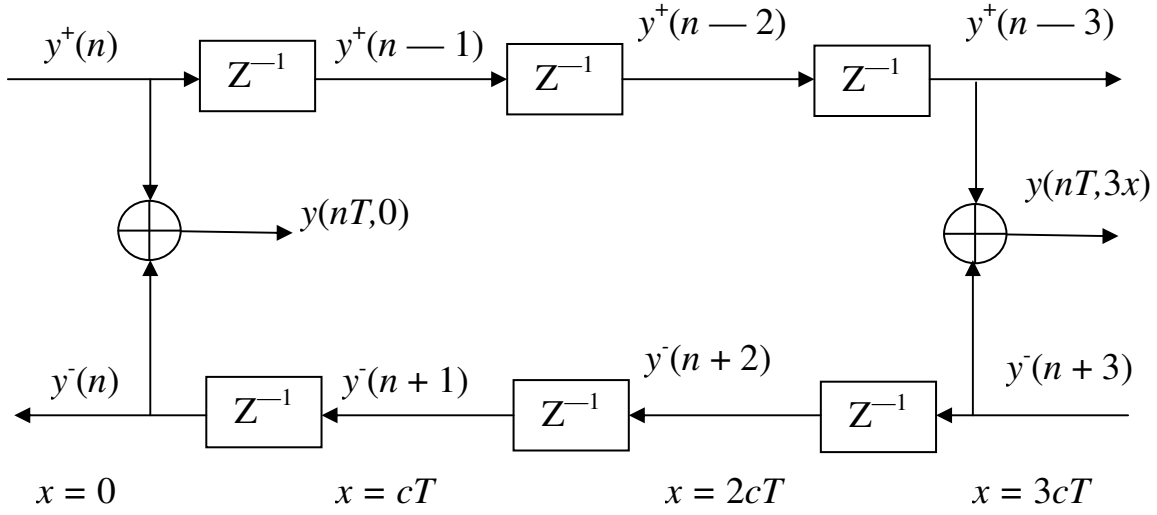

Fig.1.2. Digital simulation of the ideal, lossless waveguide

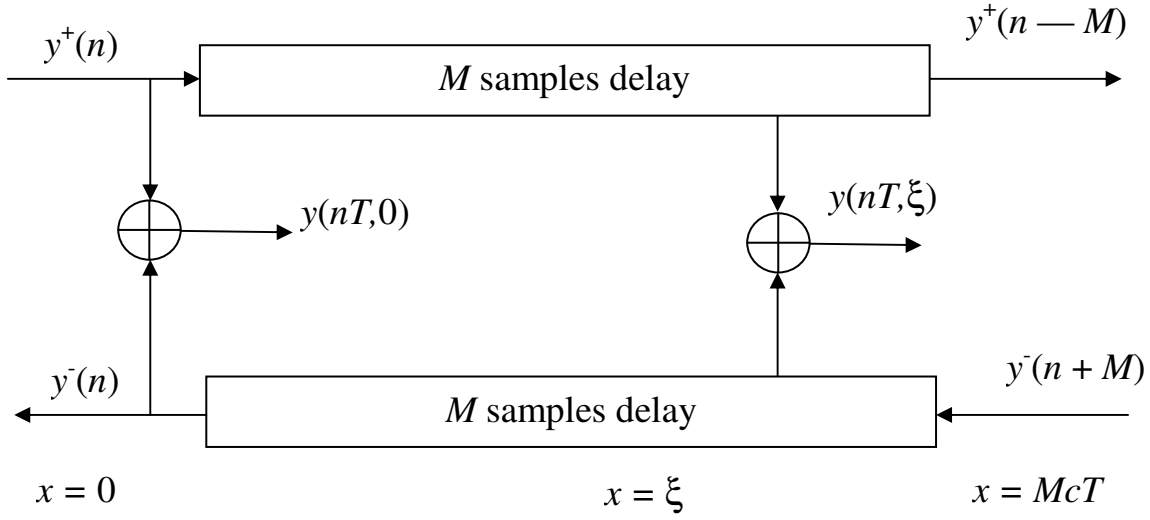

Fig. 1.3. Simplified picture of ideal waveguide simulation.

The bandlimited spatial interpolation may be used to construct a displacement output for an arbitrary x not a multiple of cT, as suggested by the output drawn in Fig. 3. Similarly, the bandlimited interpolation across time serves to evaluate the waveform at an arbitrary time not an integer multiple of T.

Ideally, bandlimited interpolation is carried out by convolving a continuous sinc function $\text{sinc}(x) = \sin(\pi x)/\pi x$ with the signal samples. Specifically, convolving a sampled signal $x(t_n)$ with $\text{sinc}[(t_n - t_0)/T]$ evaluates the signal at an arbitrary continuous time t_0. The sinc function is the impulse response of the ideal lowpass filter, which cuts oo at half the sampling rate.

In practice, the interpolating sinc function must be windowed to a finite duration. This means the associated lowpass filter must be granted a transition band in which its frequency response is allowed to roll off to zero at half the sampling rate. The interpolation quality in the pass band can always be made perfect to within the resolution of human hearing by choosing a sufficiently large product of window-length times transition-bandwidth. Given audibly perfect quality in the pass band, increasing the transition bandwidth reduces the computational expense of the interpolation.

This is one reason why oversampling at rates higher than twice the highest audio frequency is helpful. This topic is described further in [8].

We have thus far considered discrete-time simulation of traveling displacement waves $y^{\pm}$ in the ideal bar. It is equally valid to choose traveling velocity $v^{\pm} = \frac{dy^{\pm}}{dt}$, acceleration $a^{\pm} = \frac{d^2y^{\pm}}{dt^2}$, or slope waves $\frac{d^2y^{\pm}}{dx^2}$, or perhaps some other derivative or integral of displacement with respect to time and /or position.

The conversion between various time derivatives can be carried out by means integrators or diierentiators. Since integration and differentiation are linear operators, and since the traveling wave arguments are in units of time, the same conversion formulas hold for the traveling wave components $y^{\pm}$; $v^{\pm}$; $a^{\pm}$.

In discrete time, integration and differentiation can be accomplished using digital filters [9]. Commonly used approximations are shown in Fig. 1.4.

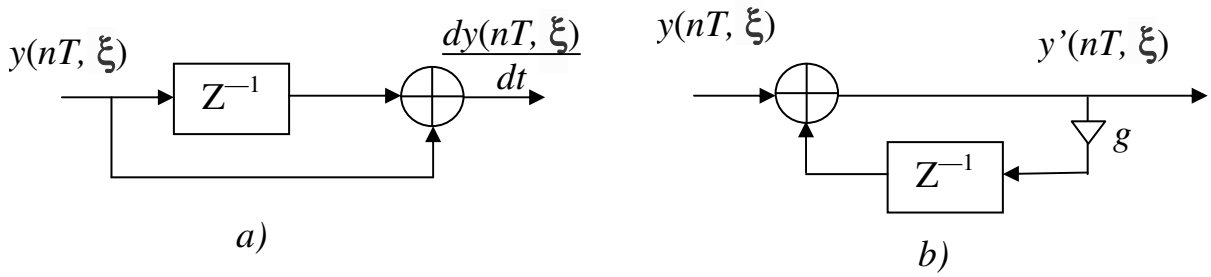

Fig.1.4. Simple approximate conversions between time derivatives in the discrete-time case: a) the first-order difference, b)the first-order leaky integrator with loss factor $g < 1$.

The first-order approximations in Fig.1.4 are accurate though scaled by T at low frequencies relative to half the sampling rate, but they are not best approximations in any sense other than being most like the definitions of integration and differentiation in continuous time. Much better approximations can be obtained by approaching the problem from a digital filter design viewpoint [10]. Arbitrarily better approximations are possible using higher order digital filters. In principle, a digital differentiator is a filter whose frequency response $H(\omega)$ optimally approximates $j\omega$ for $-\pi T < \omega < \pi T$.

Similarly, a digital integrator must match $1/j\omega$ along the unit circle in the z plane. The reason an exact match is not possible is that the ideal frequency responses $j\omega$ and $1/j\omega$, when wrapped along the unit circle in the z plane, which is the frequency axis for discrete time systems, are not simple functions anymore. As a result, there is no filter with a rational transfer function □i. e., finite order that can match the desired frequency response exactly.

In addition to time derivatives, we may apply any number of spatial derivatives to obtain yet more wave variables to choose from. The first spatial derivative of the bar displacement yields slope waves:

$$y'(t,x) = \frac{d}{dx}(t,x) = y'_r(t—x/c) + y'_l(t + x/c)$$

or, in discrete time,

$$y'(t_n,x_m) = y'(nT,mX) = y'^{+}(n - m) + y'^{-}(n + m) =$$

$$= \frac{1}{c}[v^{-}(n + m) - v^{+}(n - m)].$$

From this we conclude that $v^{-} = cy'^{-}$, and $v^{+} = cy'^{+}$. That is, the traveling slope waves can be computed from traveling velocity waves by dividing by c and negating in the right-going case. Physical bar slope can thus be computed from a velocity-wave simulation by subtracting the upper rail from the lower rail and dividing by c.

By the wave equation, the curvature waves are $y'' = \frac{d^2y^{\pm}}{dx^2}/c^2$, are essentially identical to the acceleration waves.

In the field of acoustics, the state of a vibrating bar at any instant of time t_0 is normally specified by the displacement $y(t_0,x)$ and velocity $v(t_0,x)$ for all x. Since the displacement is the sum of the traveling displacement waves and velocity is proportional to the difference of the traveling displacement waves, one state description can be readily obtained from the other.

In summary, all traveling-wave variables can be computed from any one, as long as both the left- and right-going component waves are available. Alternatively, any two linearly independent physical variables, such as displacement and velocity, can be used to compute all other wave variables. The wave variable conversions requiring the differentiation or integration are relatively expensive since a large-order digital filter is necessary to do it right. The slope waves can be computed from velocity waves by a simple scaling, and vice versa, and curvature waves are the same as acceleration waves to within a scale factor.

In the absence of other factors dictating a choice, the velocity waves are a good overall choice because, firstly, it is numerically easier to perform the digital integration to get the displacement than it is to differentiate the displacement to get

the velocity, secondly, the slope waves are immediately computable from the velocity waves.

At an arbitrary point x along the rigid bar, the vertical force applied at time t to the portion of a bar to the left of position x by the portion of a bar to the right of position x is given by

$$f_l(t,x) = K\sin(\theta) \approx K\tan(\theta) = Ky'(t,x),$$

assuming that $|y'(t,x)| \ll 1$, as is assumed in the derivation of the wave equation. Similarly, the force applied by the p ortion to the left of p osition x to the p ortion to the right is given by

$$f_r(t,x) = -K\sin(\theta) \approx -Ky'(t,x).$$

These forces must cancel since a nonzero net force on a massless point would produce infinite acceleration.

The vertical force waves propagate along the bar like any other transverse wave variable, since they are just slope waves multiplied by tension K. Either f_l or f_r as the force wave variable, one being the negative of the other. It turns out that to obtain a unifcation of vibrating bars, strings and air columns, f_r has to be picked, the one that acts to the right. This makes sense intuitively when one considers longitudinal pressure waves in an acoustic tube: a compression wave traveling to the right in the tube pushes the air in front of it and thus acts to the right. Thus, the force wave variable is defined to be

$$f(t,x) = f_r(t,c) = -Kv(t,x).$$

Substituting from above, the following is got

$$f(t,x) = \frac{K}{c}[v_r(t - x/c) - v_l(t + x/c)].$$

Note that $K/c = K/\sqrt{K/\varepsilon} = \sqrt{K\varepsilon}$. This is a fundamental quantity known as the *wave impedance* of the body, which is also called the *characteristic impedance*, denoted as $R = = \sqrt{K\varepsilon}$.

The wave impedance can be seen as the geometric mean of the two resistances to displacement: tension, or spring force, and mass, or inertial force. Note that $R = K/c = \varepsilon c$ since $c = \sqrt{K/\varepsilon}$.

The digitized, traveling, force-wave components become

$$f^+(n) = Rv^+(n); \tag{1.3}$$
$$f^-(n) = -Rv^-(n).$$

which gives us that the right-going force wave equals the wave impedance times the right-going velocity wave, and the left-going force wave equals minus the wave impedance times the left-going velocity wave. Thus, in a traveling wave, force is always in phase with velocity, considering the minus sign in the left-going case to be associated with the direction of travel rather than a 180° phase shift between force and velocity. Note that if the left-going force wave were defined as the string force acting to the left, the minus sign would disappear.

In the rigid cylindre context, the wave impedance is given by

$$R = \frac{\rho c}{A},$$

where ρ is the mass per unit volume of the material, c is sound speed, and A is the cross-sectional area of the cylindre. Note that if the particle velocity is chosen rather than the volume velocity, the wave impedance would have been $R = \rho c$ instead, the wave impedance in open air.

In any real vibrating bar, there are energy losses due to the yielding terminations, drag by the surrounding air, and internal friction within the bar. While the losses in solids generally vary in a complicated way with frequency, they can usually be well approximated by a small number of odd-order terms added to the wave equation. In the simplest case, the force is directly proportional to the transverse velocity, independent of frequency. If this proportionality constant is μ, the modified wave equation is obtained:

$$K\frac{d^2}{dt^2}y(t,x) = \varepsilon \frac{d^2}{dx^2}y + \mu \frac{d}{dx}y;$$

Thus, the wave equation has been extended by a first-order term, i.e., a term which is proportional to the first derivative of y with respect to time. More realistic loss approximations would append the terms, which are proportional to the third, and fifth derivatives of *y,* giving the frequency-dependent losses.

It can be checked that for small displacements, the following modified traveling wave solution satisfies the lossy wave equation:

$$y(t,x) = e^{-(\mu/2\varepsilon)x/c} y_r(t - x/c) + e^{(\mu/2\varepsilon)x/c} y_l(t + x/c).$$

The left-going and right-going traveling-wave components decay exponent-tially in their respective directions of travel.

Sampling these exponentially decaying traveling waves at intervals of T seconds or $X = cT$ meters gives

$$Y(t_n,x_m) = g^{-m} y^+(n - m) + g^{-m} y^+(n - m),$$

where $g = e^{-\mu T/2\varepsilon}$.

The digital simulation diagram for the lossy waveguide is shown in Fig. 1.5. Note, that the simulation of the decaying traveling-wave solution is exact at the sampling positions and instants, even though losses are admitted in the wave equation. Note also, that the losses, which are distributed in the continuous solution have been consolidated, or lumped, at discrete intervals of cT meters in the simulation. The lumping of distributed losses does not introduce an approximation error at the sampling points. Furthermore, the bandlimited interpolation can yield the arbitrarily accurate reconstruction between samples. The only restriction is again that all initial conditions and excitations be bandlimited to half the sampling rate.

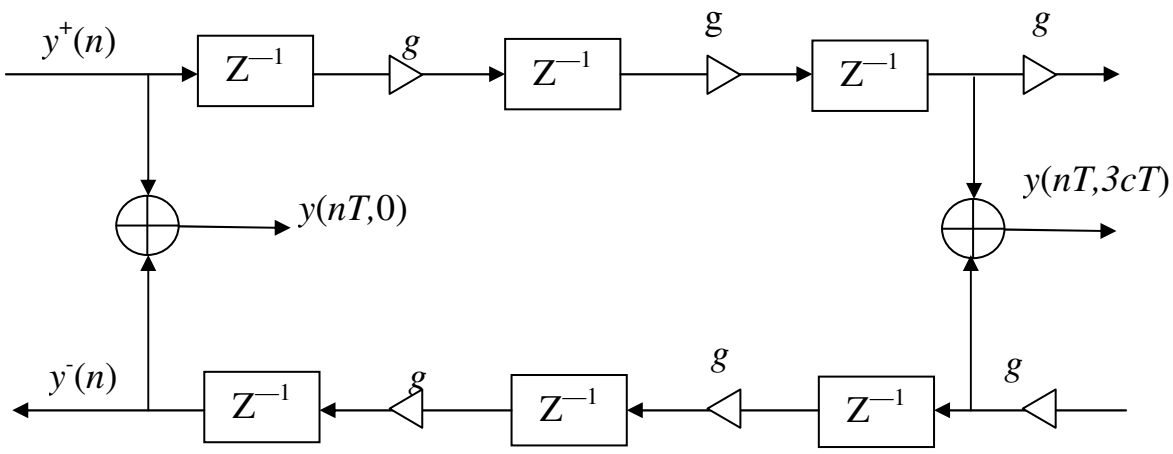

Fig.1.5. Discrete simulation of the ideal, lossy waveguide with the loss factor *g*

In many applications, it is possible to realize vast computational savings in waveguide simulation by commuting losses out of unobserved and undriven sections of the medium and consolidating them at a minimum number of points. Because the digital simulation is linear and the time invariant given constant medium parameters *K*, ε, μ, and because the linear, time-invariant elements commute, the diagram in Fig. 1.6 is exactly equivalent to within numerical precision to the previous diagram in Fig.1.8.

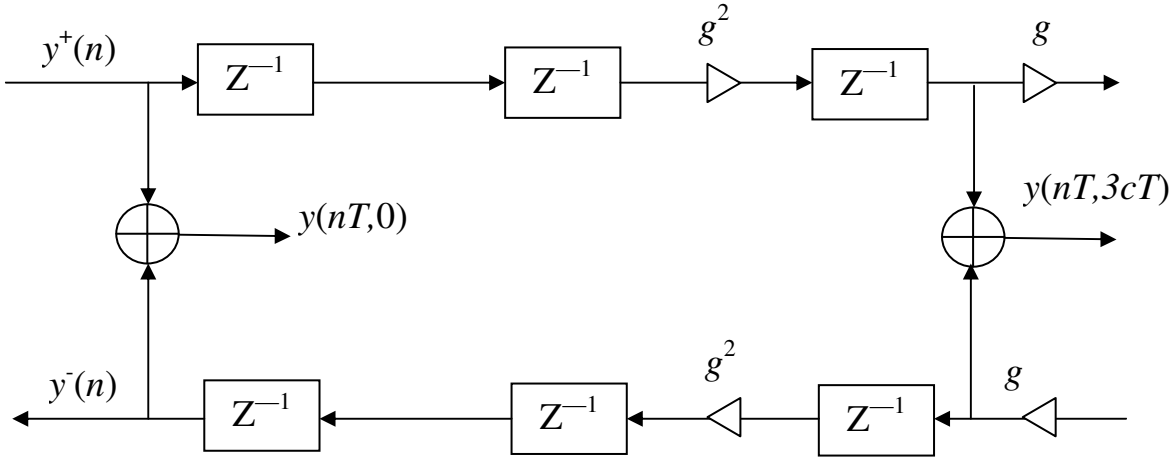

Fig.1.6. Discrete simulation of the ideal, lossy waveguide with the loss factor *g*, when the number of the factor multipliers is minimized

In the nearly all natural wave phenomena, the losses increase with frequency. The largest losses in the real bars occur at the frequencies, which couple to body resonances. There are also losses due to air drag and internal bulk losses, which increase monotonically with frequency. Similarly, air absorption increases with frequency, providing an increase in propagation loss for sound waves in air [11].

The solution to the lossy wave equation can be generalized to the frequency-dependent case. Instead of the factors g distributed throughout the diagram, the factors are now $G(\omega)$. These loss factors, implemented as digital filters, may also be consolidated at a minimum number of points in the waveguide without introducing an approximation error.

In an efficient digital simulation, each lumped loss factor $G(\omega)$ is to be approximated by a rational frequency response $G(e^{jT\omega})$. In general, the coefficients of the optimal rational loss filter are obtained by minimizing $|G(\omega) - G(e^{jT\omega})|$ with respect to the filter coefficients or the poles and zeros of the filter. To avoid the introducing the frequency-dependent delay, the loss filter should be a zero-phase, fnite-impulse-response (FIR) filter [9]. Restriction to zero phase requires the impulse response $g(n)$ to be finite in length i.e., an FIR filter and it must be symmetric about time zero, i. e., $g(-n) = g(n)$. For the real-time implementations, the zero-phase FIR filter can be converted into a causal linear phase filter by "stealing" delay from the loop delay lines in an amount equal to half the impulse response duration.

The stiffness in a long vibrating bar introduces a restoring force proportional to the fourth derivative of the displacement [11,12]:

$$\varepsilon \frac{d^2}{dx^2} y = K \frac{d^2}{dt^2} y(t,x) - k \frac{d^4}{dt^4} y(t,x) \; ;$$

where for a cylindre of radius a and Young's modulus Q, the moment constant k is equal to $k = Q\pi a^4/4$.

At very low frequencies, or for very small k, we return to the non-stiff case. At very high frequencies, or for very large k, we approach the ideal bar, in which the stiffness is the only restoring force. At intermediate frequencies, between the ideal string and bar, the stiffness contribution can be treated as a correction term [13]. This is the region of most practical interest because it is the principal operating region for a bar. The frst-order effect of the stiffness is to increase the wave propagation speed with frequency:

$$c(\omega) = c_0\left(1 + \frac{k\omega^2}{2Kc_0^2}\right), \tag{1.4}$$

where c_0 is the wave travel speed in the absence of the stiffness. Since the sound speed depends on the frequency, the traveling waveshapes will disperse as they propagate along the bar. That is, a traveling wave is no longer a static shape moving with speed c and is expressible as a function of $t \pm x/c$. In a stiff bar, the high frequencies propagate faster than the low-frequency components. As a result, a traveling velocity step, such as would be caused be a hammer strike, unravels into a smoother velocity step with high-frequency ripples running out ahead.

In a digital simulation, a frequency-dependent speed of the propagation can be implemented using the allpass filters, which have a non-uniform delay versus frequency. Note that every linear, time-invariant filter can be expressed as a zero-phase filter in cascade with an allpass filter. The zero-phase part implements thefrequency-dependent gain damping in a digital waveguide, and the allpass part gives the frequency-dependent delay, which in a digital waveguide yields the dispersion. Every linear wave equation with constant coefficients, regardless of its order, corresponds to a waveguide, which can be modeled as a pure delay and a linear, time-invariant filter, which simulate propagation over a given distance.

A *rigid termination* is the simplest case of a long bar termination. It imposes the constraint that the bar, or string cannot move at all at the termination. If we terminate a length L ideal string at $x = 0$ and $x = L$, we then have the boundary conditions:

$$y(t,0) = 0, \;\; y(t,L) = 0.$$

Since $y(t,0) = y_r(t) + y_l(t) = y^+(t/T) + y^-(t/T)$ and $y(t,L) = y_r(t-L/c) + y_l(t+L/c)$, the constraints on the sampled traveling waves become

$$y^+(n) = -\,y^-(n);$$
$$y^-(n+N/2) = -\,y^+(n-N/2).$$

where $N = 2L/X$ is the time in samples to propagate from one end of the string to the other and back, or the total string loop delay. The loop delay is also equal to twice the number of spatial samples along the string. A digital simulation diagram for the terminated ideal string is shown in Fig. 1.7. A pick-up is shown at the arbitrary location $x = \xi$.

The damping should increase at higher frequencies for better realism of the wave propagation modeling. This means the loss factors g are replaced by the digital filters having gains which decrease with the frequency and never exceeding 1. These filters commute with delay elements because they are time invariant. Thus, following the reasoning above, they can be lumped at a single point in the digital waveguide. Let $G(z)$ denote the resulting string loop filter. We have the constraint $|G(e^{j\omega T})| < 1$, and making it zero or linear phase will restrict consideration to symmetric FIR filters only.

In the simplest case of a first-order low pass loss filter, $G(z) = b_0 + b_1 z^{-1}$, the linear-phase requirement imposes $b_0 = b_1$. Assuming the damping approaches zero at the frequency zero implies $b_0 + b_1 = 1$. Thus, two equations in two unknowns uniquely determine the coefficients to be $b_0 = b_1 = ½$, which gives a string loop frequency response equal to $G(e^{j\omega T}) = \cos(\omega T/2)$, $|\omega| < \pi f_s$.

The simulation diagram for the ideal string with the simplest frequency-dependent loss filter is shown in Fig. 1.8. This is the structure of the Karplus-Strong algorithm [14, 15].

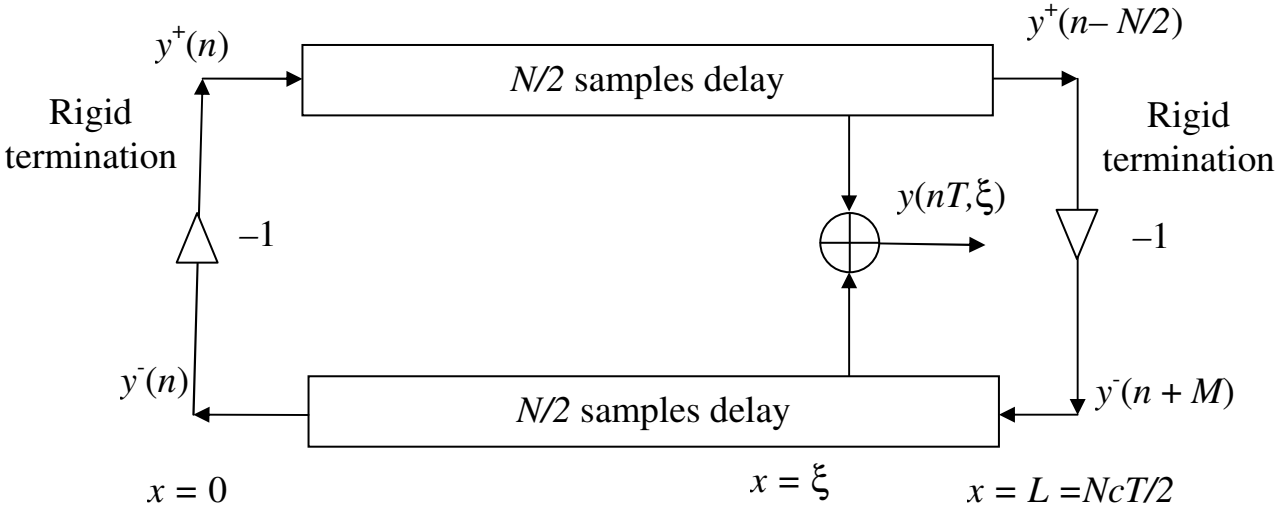

Fig.1.7. The rigidly terminated ideal string, with a position output indicated at $x = \xi$

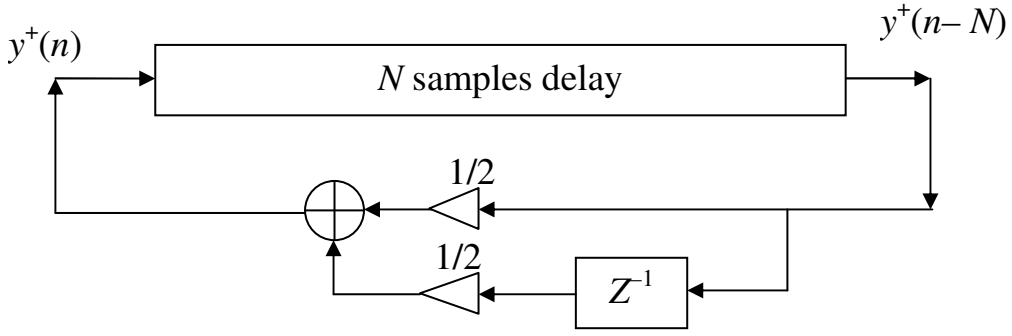

Fig.1.8. Rigidly terminated string with the simplest frequency-dependent loss filter.

Starting with the traveling-wave solution to the wave equation and sampling across time and space, a modeling framework known as the digital waveguide approach is obtained. Its main feature is computational economy in the context of a true physical model. Successful computational models have been obtained for several musical instruments of the string and wind families, and more are on the way.

1.4 Improvements of the digital waveguide modeling method

In [15], the digital waveguide modeling method is considered from the opposite side. The authors propose the model for the acoustic wave investigations, which is based on the electric models of the wave processes. Then, different electrical network elements represent the elements of the mechanical model. The resistor represents the damping element in the lossy waveguide, inductance represents the differentiating element for the displacement, capacitance does the differentiating element for the velocity and so on. The multiport parallel adaptor is added to the theory for connecting together different waveguides. The resulting model is sampled form a set of blocks, which are connected to each other through their ports in the similar manner. This theory is based on the prominent fundamental work [17], which described in [18] as well.

Block-based physical modeling allows us to build the digital sound synthesis algorithms with a great variety of models, including linear and nonlinear lumped and distributed parameter systems. An interconnection strategy based on the wave digital principle guarantees the computability of the resulting algorithms. They can be generated by an automated synthesis strategy based on the binary connection tree.

Local waveguides and their connections can form a mesh, which models some rigid body. In the work [19], a formulation of a specific finite difference time domain model structure is proposed, which is based on this principle. This formulation allows for high flexibility in building 1D or higher dimensional physical models from interconnected blocks.

In this model, the local properties of the waveguide are modeled using the special wave digital filters, which selection was shown in [20]. Analogous approach is proposed in [21].This approach is expanded and widely described in a book [22].

The real rigid body with the limited dimensions has the nonlinear acoustic properties. These properties can be taken into account by the special digital filters, proposed in [23]. In [24] the allpass filters are proposed to realize the nonlinear delays in the objects. In [25,26] this approach is expanded to a set of nonlinearities.

As usual, the modeling of the wave propagation is performed on the usual computers and servers. In [27], the application specific parallel system of the microprocessors is proposed, which performs this task in real time.

In [28], the modeling structure is proposed which is implemented in FPGA. It provides the real time sound synthesis up to the several megahertz range. The similar method is used in the model of the piano, which is performed in real time in FPGA [29].

So, the modeling of the wave propagation in the solid bar can be implemented using the methods of the digital waveguides, which can be performed in FPGA.

1.5 Preliminary conclusions

In this section, the basics of the mathematical modeling, finite difference schema and the finite element method, and method of the digital waveguide modeling were considered.

It was defined that the method of the digital waveguide modeling is effective one. It provides simple but precise modeling of the sound wave propagation in the solid bodies taking into account the nonlinearities, space dimensions of the object.

The waveguide modeling method has a set of improvements directed to the increase of the model adequacy application it to the different body configurations.

The simplicity of the waveguide model helps to perform it in real time utilizing the computer system with the medium performance. There are many efforts to implement the method in the hardware of the modern FPGA. These efforts show that the modeling can be implemented in real time for the high frequency sounds.

We can make a conclusion that the digital waveguide method is perspective one and it must be improved. The improvements must deal with the taking into account the dispersion of the sound propagation, adding the ability to correct the model dynamically.

A new method of the hardware modeling of the ultrasound wave propagation in the solid bodies has to be developed. This method is based on the digital waveguide method, and designing the application specific processor implemented in FPGA.

In the next section, the theoretical basics of the new methods are developed, which satisfy the mentioned above features.

2 METHOD OF THE HARDWARE MODELING OF THE ULTRASOUND WAVE PROPAGATION IN THE SOLID BODIES

2.1 Features of ultrasound propagation in a rod

In the previous section we discussed the acoustic wave propagation in the ideal area, not to consider the nonlinear effects. And it is shown that this physical process is effectively modeled using the waveguide model. But when the ultrasound propagation is investigated in the real stiff rods, then the model must be improved.

Firstly, we take some definitions and relations.

The ***phase velocity*** v_p of a harmonic wave is the velocity of displacement in the space of a point at which the phase of the wave remains constant, and is equal to

$$v_p = \frac{dx}{dt} = \frac{\omega}{k},$$

where $k = 2\pi/\Lambda = \omega/ v_ф$ is the wave number;

Λ is the wave length.

Consider the ultrasound signal ω_0=6,2 $\cdot 10^6$ rad/s, v_p = 3000 m/s, then $k = 6{,}2 \cdot 10^6/3000 = 2100$ rad/m = 0,3 periods per millimeter. If the ultrasound signal has the frequency f_0 = 3 MHz, then Λ= 1 mm.

The dependence of the phase velocity on the wavelength (wave number or frequency) is called the ***dispersion of the wave***. If there is a dispersion in the system, then the frequency ω is a nonlinear function of the wave number k and vice versa.

The limiting value of the sound envelope velocity characterizes the motion of a wave group (wave packet) and is called the ***group velocity***. The group and phase velocities are related to the following relation (Rayleigh formula):

$$v_g = v_p + k\frac{dv_p}{dk} = v_p - \Lambda\frac{dv_p}{d\Lambda}$$

The waves that characterize the change in the shape of an element of the medium with an unchanged volume. are called the ***distortion waves*** or ***shear waves***. They propagate at a constant rate $c_\tau = \sqrt{\frac{\mu}{\rho}}$ and do not have dispersion.

The ratio of the propagation velocities of the elastic shear waves and the waves of the expansion-compression depends only on the Poisson's ratio ν:

$$\beta = \frac{c_\tau}{c_l} = \sqrt{\frac{1-2\nu}{2-2\nu}} .$$

It is known that the Poisson's ratio for various materials varies within $0 < \nu < 0.5$. For the steel material $\nu = 0{,}29$.

The stretch-compression wave is called the ***longitudinal wave***, and c_l is the propagation velocity of the longitudinal wave.

The shear wave is called the ***transverse wave***, and $c_\tau < c_l$ is the propagation speed of the transverse wave.

In an infinite medium, the stretching-compression waves (p-waves) and the shear waves (s-waves) propagate independently of each other. The waves of different types can transform into each other, interacting with the medium surface.

When the waves are mutually transformed, a combination of p and s components can arise, such that the perturbation will be a wave traveling along the boundary and abruptly damped as it moves deeper into the half-space. These waves are called ***Rayleigh surface waves***. The Rayleigh wavest does not have the dispersion, its c_R speed is a constant value for each material.

The ***normal waves*** in a layer that have particle displacements both in the direction of wave propagation parallel to the plane of the layer, and perpendicular to the plane of the layer, are called ***Lamb waves***.

The Lamb waves are divided into two groups: ***symmetrical*** (s) and ***antisymmetric*** (a) with respect to the median plane.

The Lamb waves have the dispersion, their velocities are functions of frequency.

A definite finite number of ***symmetric*** and ***antisymmetric normal waves*** can propagate simultaneously at a frequency ω in a layer. These waves will differ among themselves by phase and group velocities and the distribution of displacements (and stresses) along the thickness of the layer. The number of normal waves increases with increasing ratio $\omega h/c_\tau$, where c_τ is the velocity of propagation of the shear waves in an unbounded medium.

Consider a cylindrical rod. Due to the theory, the primary longitudinal wave with the phase velocity c_p, and the secondary transverse wave with the velocity c_s, $c_s < c_p$ are propagated in the solid bars [30]. The waves of different types can be transformed into each other interacting with the media boundaries. Besides, the longitudinal waves have the dispersion, i.e. its velocity c_p depends on the wave length Λ. The velocity c'_p in a cylinder is approximated as follows [30]:

$$c'_p = c_o\left(1 - \frac{\nu^2\pi^2a^2}{\Lambda^2}\right), \qquad (2.1)$$

where a is the cylinder radius.

But this approximation is correct only for the long waves. Fig. 2.1 illustrates the ratio c'_p/c_o , and more realistic ratio c_p/c_o depending on the ratio a/Λ for the steel rod. It shows that when the wavelength becomes shorter than the rod radius then the velocity approaches to $0.576c_o$ [30].

The antisymmetric normal waves also have dispersion but in the low-frequency range. The phase velocity of the wave is proportional to the square root of the frequency: $c_p = k\sqrt{\omega}$. But these waves are prevalent in the too long rods, for example, in strings.

So, developing a new wave propagation model we have to consider the primary longitudinal waves with the phase velocity c_p, and the secondary transverse wave with the velocity c_s. Besides, the longitudinal waves have a dispersion, i.e., its velocity depends on the wavelength Λ. And the model has to take into account the dependency shown in Fig. 2.1.

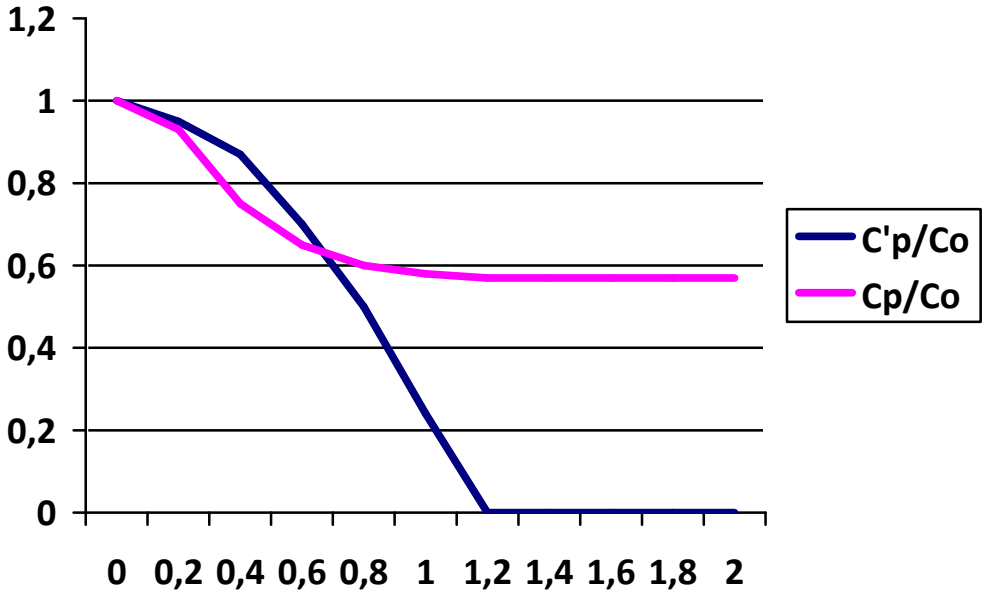

Fig.2.1. Dependence of the longitudinal wave velocity on the wavelength ratio a/Λ

2.2 Wave digital filters in the waveguide models

2.2.1 Background of the wave digital filter use

In the previous section, it was shown, that the waveguide model has the properties to represent correctly the wave propagation phenomena, and that this model can be implemented using the registered signal delays, adders and multiply units. Below we concrete the waveguide model design on the base of the digital wave propagation theory and wave digital filters [17].

The wave digital filters are based on a direct analogy with the wave propagation processes in the waveguides, for example, microwave waveguides or sound waveguides, which was shown in the previous section. In this case, the distribution of the direct f_i and the reverse (reflected) b_i waves in the i-th point of the waveguide are different. In the case of an electric waveguide

$$f_i = v + i\,R; \qquad b_i = v - i\,R;$$

where v is the voltage at the i-th point of the waveguide (between its poles), i is the current at this point (flowing into 1-st pole and flowing out of the 2-nd pole) and R is the wave resistance in it.

In the case of an acoustic waveguide

$f_i = Rv_f$ is the pressure of the direct wave,

$b_i = -Rv_b$ is the back pressure,

where, R is the wave impedance, v_f, v_b are the particle velocity in the forward and backward waves. For a sound pipe of section A, air density ρ and sound velocity with impedance $R = \rho c/A$. For an elastic body, if f_i and b_i are assumed to be forces, then $R = \rho c$, where ρ is the density of the material, and c is the velocity of the longitudinal waves.

The true value of the physical quantity at the i-th point is $u = f_i + b_i$.

In the wave model, the signals are sampled with a sampling frequency Fd at least 2 times greater than the maximum considered frequency (actually in 4 — 20 times larger). Then the i-th section of the waveguide of the length L looks like 2 delay lines at $n = L / (cFd)$ cycles, which is illustrated by Fig.2.2.

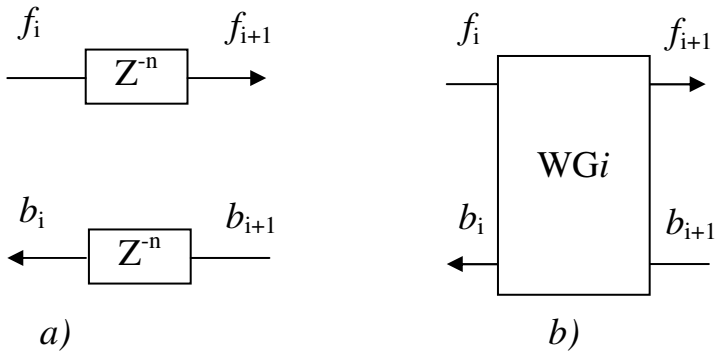

Fig. 2.2. The i-th segment of the waveguide (a) and its symbol (b)

The homogeneous parts of waveguides are connected together by the means of the adapters, also called scattering nodes, or energy distributions. The adapter function is designed to fulfill the Kirchhoff law. According to it, in the absence of losses, with parallel connection of N two-port networks (waveguide inputs), the total voltage at the poles for all two-ports (speed of elastic body particles or air pressure) should be the same, and the sum of currents i (sum of pressures in all sides of the

elastic body or air velocity) should be zero. Hence the total velocity of the particles at the site is expressed in terms of the particle velocities of the incoming waves v_{fi} and the impedances:

and the velocities of the waves which leave the node:

$$v = 2\sum_{i=1}^{N} R_i v_{fi} \,/\, \sum_{i=1}^{N} R_i$$

$$v_{bi} = v - v_{fi}.$$

In the case of connection of 2 waveguides:

$$v_{b1} = r v_{f1} + (1\text{-}r) v_{f2}; \qquad (2.2)$$

$$v_{b2} = (1+r) v_{f1} - r v_{f2};$$

where $r = (R_2 - R_1)/(R_2 + R_1)$ is the coefficient of the reflectance from the waveguide border. Such a network is illustrated by Fig. 2.3.

In the extreme case, when a complete reflection is obtained, for example, at the end of a waveguide in contact with an open space, the model looks like one in Fig.2.4.

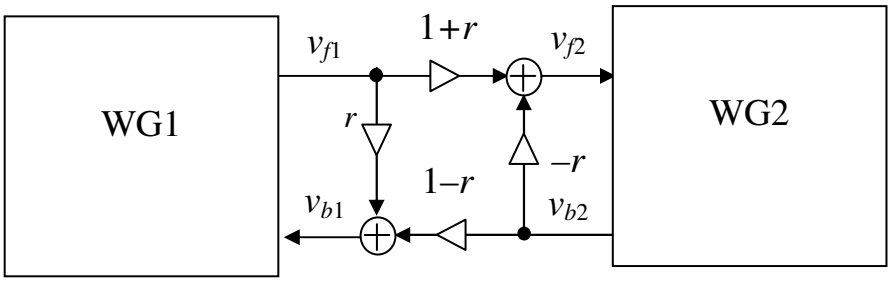

Fig. 2.3. Connection of two waveguides through an adaptor network

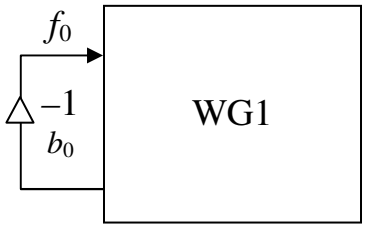

Fig. 2.4. Waveguide, which is ended by the open space

When two waveguides with the same intrinsic impedance are connected, then the adapter is minimized to a trivial one, which is shown in Fig. 2.5. In the Fig. 2.5, an adder calculating the resultant signal y at the point of coupling of the waveguides and the adders supplying the excitation signal x to the waveguides are also shown.

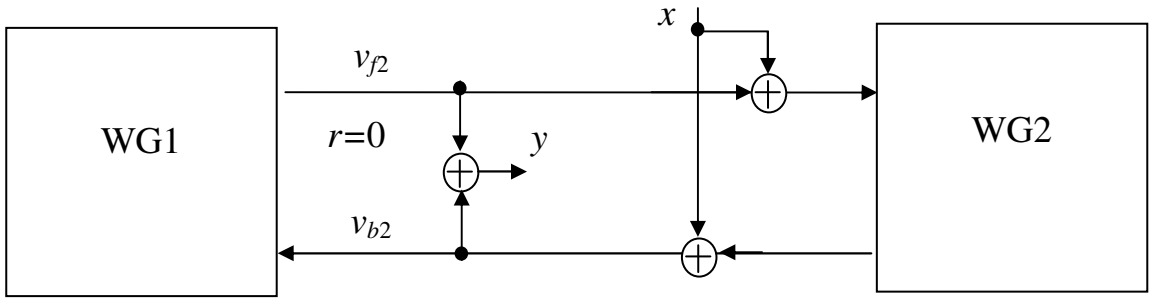

Fig. 2.3. Connection of two waveguides with equal intrinsic impedances

Similarly, when 3 waveguides are connected, the adapter calculates the following formulas:

$$\begin{aligned} Ao &= (2g_1 - 1)A + 2g_2B + 2g_3C; \\ Bo &= 2g_1A + (2g_2 - 1)B + 2g_3C; \\ Co &= 2g_1A + 2g_2B + (2g_3 - 1)C; \end{aligned} \qquad (2.3)$$

where A, B, C are the waves entering the node, Ao, Bo, Co are the waves leaving the node, g_1, g_2, g_3 are the specific impedances of the waveguides, and $g_1 + g_2 + g_3 = 1$, and

$$g_i = \frac{R_i}{\sum_{i=1}^{N} R_i}, \; i = 1 \ldots N.$$

The connection of three waveguides through an adaptor is shown in Fig. 2.4.

It should be noted that, usually in adapter networks, the number of multiplications is minimized by increasing the addition number using the equivalent transformations of formulas, considering, for example, that $g_1 + g_2 + g_3 = 1$.

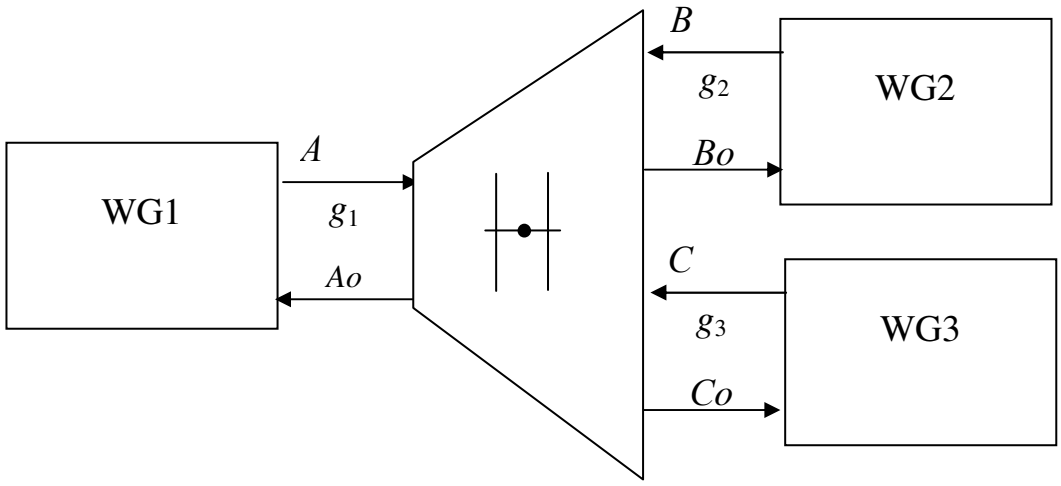

Fig. 2.4. Connection of three waveguides with different intrinsic impedances

Thus, a two-input adapter usually has 2 additions and 2 multiplications or 3 additions and 1 multiplication, and the three-input adapter has 6 additions and 2 multiplications.

All waveguide models and wave filters are represented as structures composed of waveguides of various lengths and adapters (parallel and serial), as well as multipliers by the attenuation coefficient (if necessary). In this case, the waveguides and adapters should alternate. In the opposite case, when connecting two adapters, or if the adapter is connected to a line without a delay, they will have a cycle of zero length (without delays), which could not implemented both physically and technically.

When modeling a two-dimensional or three-dimensional medium, the structure of the waveguide model is a two-dimensional or three-dimensional lattice, at the nodes of which are performed as the adapters with a corresponding number of inputs and outputs. The nodes of such a network are connected to each other by the segments of waveguides with an appropriate delay (usually 1 clock cycle).

In many cases, the waveguide delays are minimized, for example, the delay registers are placed either in the forward branch, or in the reverse branch. However, it should be taken into account that the simulation and hardware implementation of

such a scheme is much slower because of the emerging long-distance routes without intermediate (pipeline) registers.

2.2.2 All-pas wave digital filters

The second order all-pass filter has a transfer characteristic:

$$H_{(z)} = \frac{Z^{-2} + b(1+a)Z^{-1} + a}{1 + b(1+a)Z^{-1} + aZ^{-2}}. \qquad (2.4)$$

It has a frequency characteristic which magnitude is equal to exactly to 1 at all frequencies if the coefficients in the numerator of the formula are strictly equal to the coefficients in the denominator. But it has the nonlinear phase characteristic, which is illustrated by Fig. 2.5.

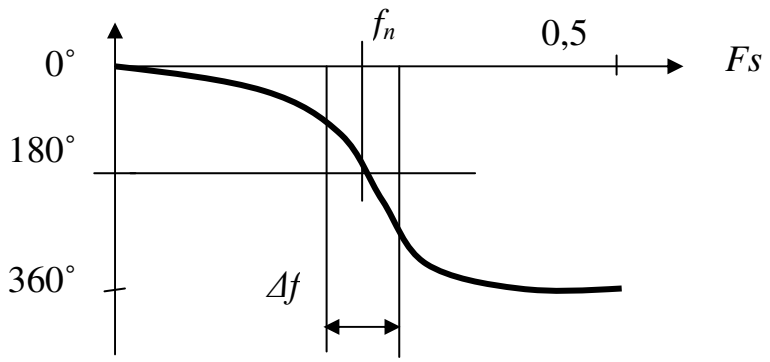

Fig. 2.5. Phase characteristic of the second order all-pass filter

In (2.4) $b = \cos(2\pi f_n)$, where f_n is the frequency of the shift by 180 degrees in fractions of the sampling frequency Fs. Thus, for $f_n = 0.25\ Fs$, the coefficient b = 0, for $f_n = 0.17\ Fs$, the coefficient $b = -0.5$, for $f_n = (0.17 + 0.25)\ Fs$ the coefficient $b = 0.5$.

The band $\Delta f = 2\text{arctg}((1 - a)/(1 + a))$ is the frequency band of the steep region in fractions of the sampling frequency, where the phase characteristic varies from 90 to 270 deg.

The peculiarity of the filter is that the adjustment of Δf and f_n are independent of each other. When a waveguide filter is implemented, its structure looks like that represented in Fig 2.6.

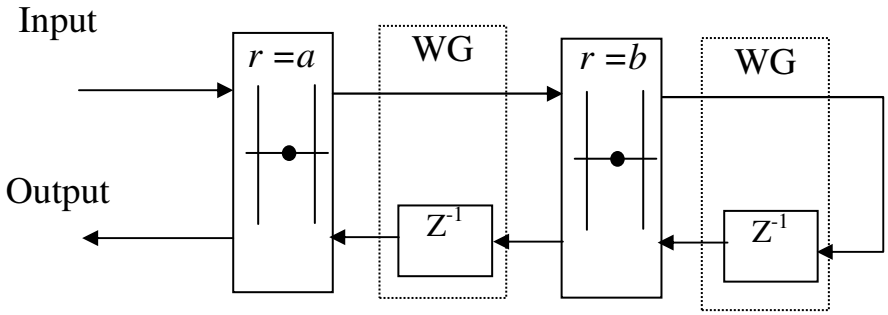

Fig. 2.6. Structure of the allpass waveguide filter with the characteristic (2.3)

In Fig. 2.6, the wave guides are dashed. With an increase in the number of delay registers in waveguides by N times, the resultant phase characteristic consists of the original phase characteristic, which is duplicated N times, with N times reduced scale along the frequency axis.

2.2.3. Band filters of the waveguide models

The band pass filters with the dynamically regulated characteristics are needed for the design of the waveguide used for the real time modeling. Such filter design is investigated in the work [31]. Below, the improved filter structures are proposed, which satisfy the conditions of the waveguide modeling. Partially they based on the theory, described in [17]. The frequency masking approach [32] is utilized as well.

The effective band pass filters are based on the allpass filters described above. The behavior of such filter is based on the fact that the allpass filters rotates the signal phase to 180° at the frequency f_n , and to approximately 0, and 360° at other frequencies (see Fig. 2.5). When the rotated signal is subtracted from the input signal, then it is magnified in 2 times at the frequency f_n , and is suppressed in other frequencies. The transfer characteristic of such filter is based on (2.4):

$$H_{BP}(z) = 0.5\left(1 - \frac{Z^{-2} + b(1+a)Z^{-1} + a}{1 + b(1+a)Z^{-1} + aZ^{-2}}\right). \quad (2.5)$$

The filter coefficients are calculated due to the following formulas:

$$\begin{aligned} t &= \mathrm{tg}(2\pi\Delta f); \\ a &= (1 - t)/(1 + t); \\ b &= -\cos(2\pi f_n). \end{aligned} \quad (2.6)$$

Here, a determines the bandwidth of the filter,

b determines the centrum frequency of the filter.

f_n is the centrum frequency of the filter,

Δf is the bandwidth of the frequency band.

This filter is well suited for the frequencies from 0.1 to 0.5 of the sampling frequency. To implement the filters for lower frequencies, it is proposed to make the two staged filter. Its transfer characteristic is:

$$H(z) = H_{LP}(z) \cdot H_{BP}(z), \quad (2.7)$$

where $H_{BP}(z)$ is similar to the filter (2.5), but it has the two fold delays:

$$H_{BP}(z) = 0.5\left(1 - \frac{Z^{-4} + b(1+a)Z^{-2} + a}{1 + b(1+a)Z^{-2} + aZ^{-4}}\right), \quad (2.8)$$

and $H_{LP}(z)$·represents the low pass filter characteristic, based on the allpass filter as well:

$$H_{LP}(z) = 0.5\left(Z^{-1} + \frac{Z^{-2} + b(1+a)Z^{-1} + a}{1 + b(1+a)Z^{-1} + aZ^{-2}}\right)$$

The low pass filter characteristic by $a = 0.28125$, and $b = -0.25$ has the form as in Fig. 2.7. The characteristic of the band pass filter (2.8) for the frequency $f_n = 0.1\ Fs$, and for $\Delta f = 0.01$ is shown in Fig. 2.8. The resulting characteristics of the band pass filter (2.7) for the frequency $f_n = 0.1\ Fs$, and for values of Δf from 0.01 to 0.03 are shown in Fig. 2.9.

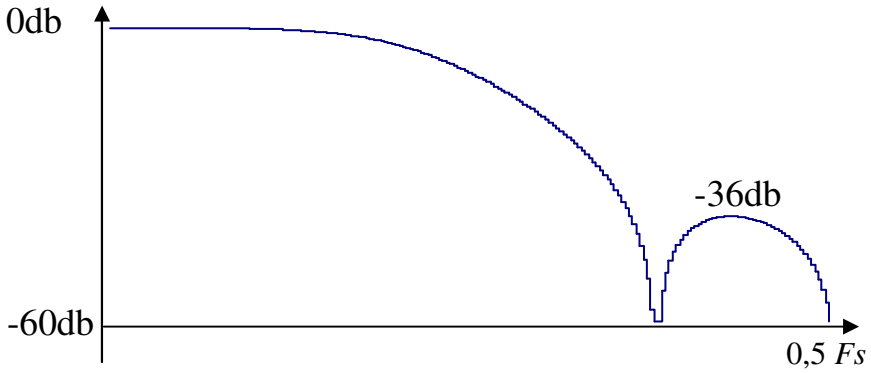

Fig. 2.7. Low pass filter frequency characteristic

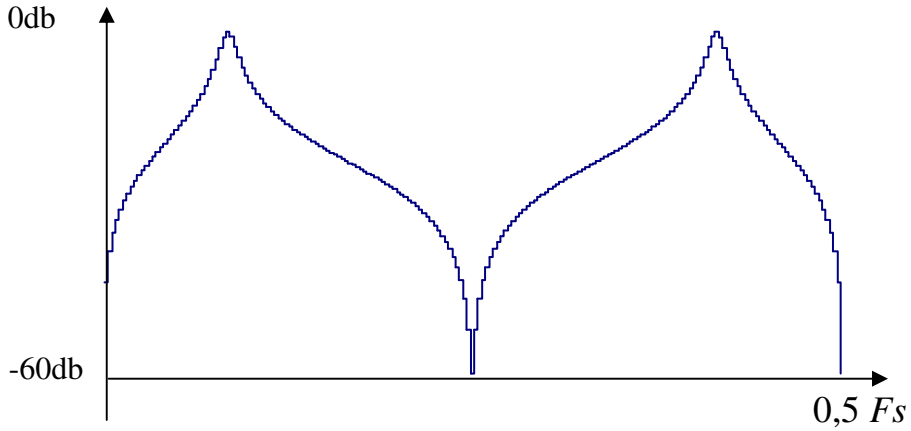

Fig. 2.8.Band pass filter frequency characteristic

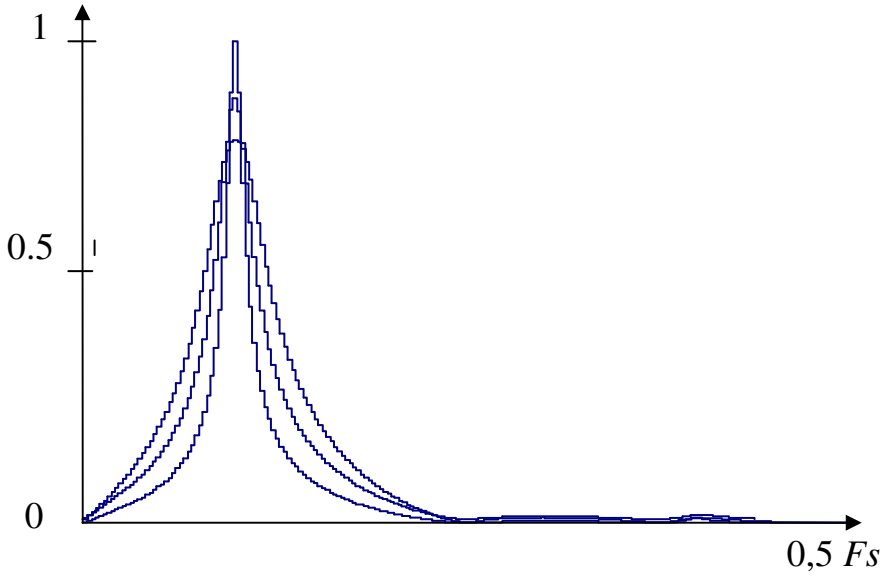

Fig. 2.9. Overall band pass filter frequency characteristics

So, the structure of one channel of the band pass filter, which is used in forming the waveguide model, is shown in Fig. 2.10. Here, the parameters f_n, Δf, and D regulate the central frequency, bandwidth, and damping.

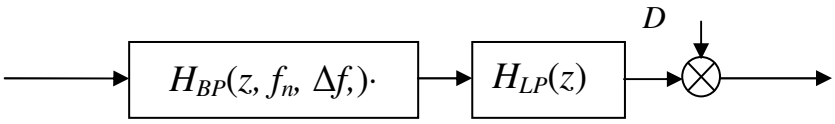

Fig. 2.10. Regulated band pass filter structure

This filter is regulated dynamically due to the fact that its coefficients are calculated according to the formulas (2.6), which provide independent exchange of both the pass band frequency and the bandwidth using simple calculations. The regulation based on the table functions fit this filter as well.

2.3 Design of the rod model

2.3.1 Model of the waveguide for the shear waves.

A feature of the propagation of the shear waves is that the speed of their rectilinear propagation does not depend on the shape of the elastic body and frequency, and is defined as

$$c_\tau = \sqrt{\frac{\mu}{\rho}} \quad ,$$

where μ is the Young's modulus of elasticity along the wave propagation axis, and ρ is the specific density. In a real body, these waves decay with distance and frequency increase.

The waveguide model for the shear waves consists of 2 identical parts for the transmission of forward and backward waves, respectively. Each part has a structure as shown in Fig. 2.11. A one-clock delay, and an adder form a simple low-pass filter in it, which frequency response is shown in Fig.2.12.

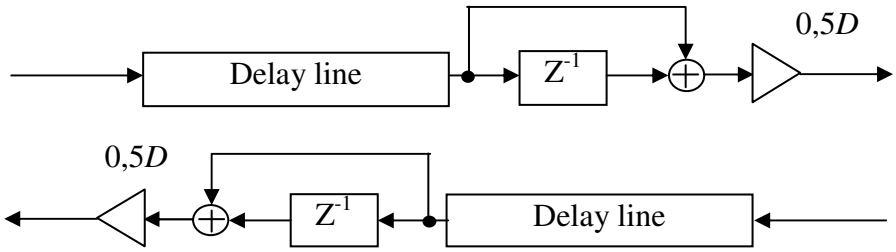

Fig.2.11. Structure of the waveguide for the shear waves

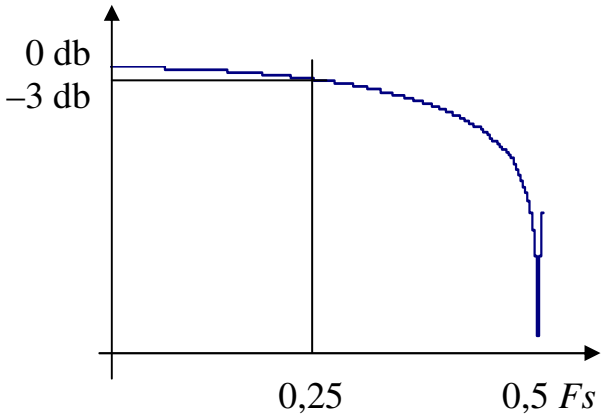

Fig.2.12. Amplitude frequency response of the waveguide for the shear waves

It is possible to include to the waveguide a sharpness characteristic, the parameters of which are regulated as it was shown in the previous subchapter. This waveguide is described in VHDL as a module named Prod_del. The generic constants of this module are:

TL is the delay of the shear waves, nanoseconds;

TD is the sampling period, nanoseconds;

DECRL is the damping coefficient (see *D* in Fig. 2.11);

DeltF is the bandwidth of the high-frequency hold in the fraction of *Fs*, if a steep filter is installed.

2.3.2 Model of the waveguide for the longitudinal waves

A feature of the propagation of longitudinal waves is that the speed of their rectilinear propagation depends both on the shape of the elastic body and on the wave frequency. This dependence in the high frequencies is estimated as (2.1), and is

described by the diagram in Fig.2.1. In the low frequencies, as well as for very long rods this dependence is given as (1.4).

In a real body, these waves also fade with the distance and increasing frequency.

The waveguide model for the longitudinal waves consists of 2 identical parts for the transmission of forward and backward waves, respectively. Each part has a structure as one in Fig. 2.13.

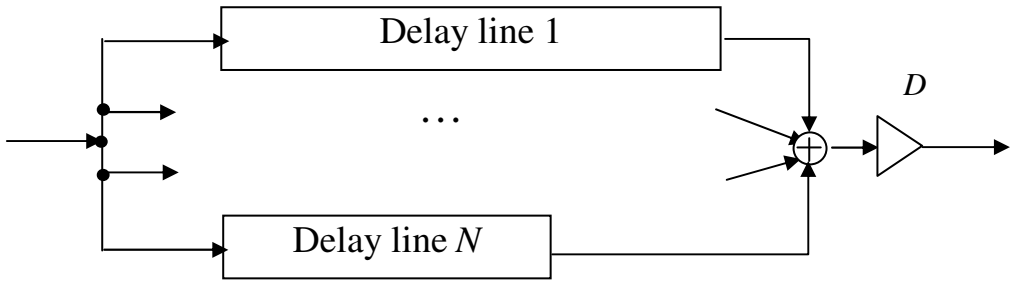

Fig. 2.13. Structure of the waveguide model for the longitudinal waves

Each delay channel is designed to delay the signal in a certain frequency band, the delay being inversely proportional to the center frequency of this band. The center frequencies of the delay channels are given by the constants, which are the fraction of the sampling frequency.

The waveguide model is described in VHDL in the module Poper_del.vhd. Each channel of this waveguide has regulated delay N_i, central cannel frequency f_n, frequency band Δf, and damping factor D_i.

The waveguide model under investigations contains 8 channels. The amount amplitude-frequency characteristic of this waveguide is shown in Fig. 2.14. Here, the frequency f is measured in parts of the sampling frequency Fs, the characteristics of two neighboring channels are shown as well.

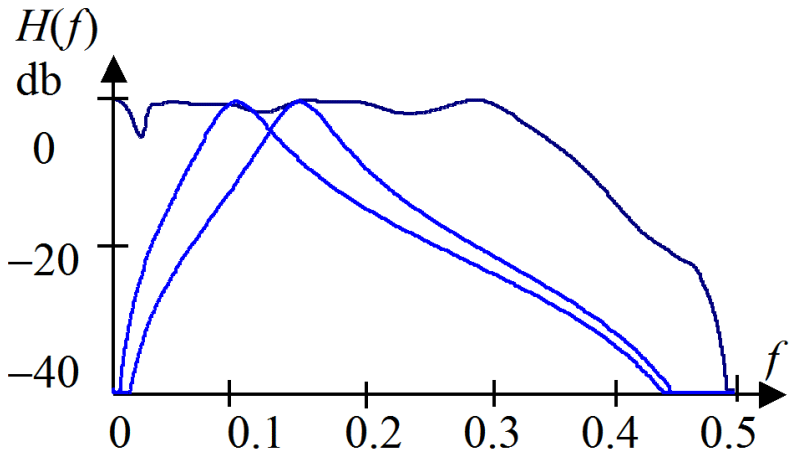

Fig.2.14. Resulting amplitude-frequency characteristic of the waveguide

Since the resulting signal is equal to the sum of the signals from the delay channels, which overlap somewhat in frequency, the resultant characteristic turns out to be uneven (with suppressions at certain frequencies).

2.3.4 Model of the rod

The modified model of the rod is illustrated by Fig.2.15. This model consists of the left LA and the right RA adapters, the waveguide P of longitudinal waves, and the waveguide S of transverse waves. Each three-port adaptors compute the equations (2.3). Its port, which means the rod end, is connected to the network, which implements the wave reflection with the suppression factor s_L or s_R due to (2.2).

The excitation signal v_{in} is fed to the model through an adder. However, it can be added to the model as it is shown in Fig. 2.3.

The waveguide S performs the delay to the given number of clock cycles and some wave attenuation. The waveguide P does the same but it has a set of channels. Each of them has the separate frequency band, delay, and attenuation depending on the respective wavelength Λ according to (2.1), or to the diagram in Fig. 2.1. The structure of this waveguide is illustrated by Fig. 2.13. Each channel of it consists of a FIFO delay and the band pass filter with the structure in Fig. 2.10.

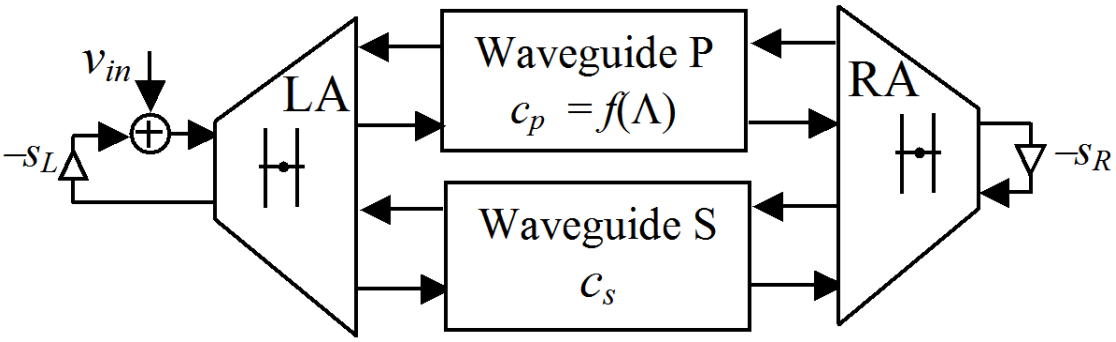

Fig.2.15. Structure of the rod model

More precisely, this model is described in [1].

2.4 Method of hardware simulation of ultrasonic processes in solids

2.4.1 Formulation of the method

The method of hardware simulation of ultrasonic processes in solids can be formed on the basis of the theoretical information developed above.

The method is formulated as follows.

The method of hardware simulation of ultrasonic processes in solids which consists in representing a solid body in the form of the network of waveguides and adapters, which model the routes of the sound rays propagation and their reflections from the heterogeneities. The method is distinguished in differentiating the waves as longitudinal waves, and transverse waves, and modeling their propagation in different waveguides. The waveguides for the longitudinal waves take into account the wave dispersion by using the multichannel network, each channel in which performs the different delay in the separate frequency band. Due to implementation on the base of the allpass wave digital filter, the waveguides are performed in hardware and can have the dynamically tuned characteristics.

2.4.2 Steps of the method

The method consists in performing a set of steps.

In the *first step*, the region to be modeled is represented as a grid structure of the waveguides and adaptors.

The graph of the structure is designed. Its edges are waveguides, and nodes are the adaptors. If the modeled body has two axes of the symmetry, then the graph has the linear form. The example is the structure in Fig. 2.15. If the body has one axe of the symmetry but with the cross-section, which does not infer the modeling process, then this situation of the graph takes place as well. For example the bar with the square cross-section, which side is comparable to the wavelength Λ.

If the body has one axe of the symmetry, then the plane graph of the grid is designed. The grid step is selected which is comparable to the wavelength Λ.

If the body has none axe of the symmetry, then the designed graph has the form od the three-dimensional grid. The grid step is selected, which is comparable to the wavelength Λ.

In the *second step*, the models of the waveguides, representing the graph edges, are installed.

Firstly the sampling frequency *Fs* is selected. It has to be at least in 2 times higher than the maximum considered signal frequency f_{max}. Really, to get the adequate results, $Fs = (3 — 10) f_{max}$. This frequency is a compromise, because when *Fs* increases, then both model complexity, and its precision are increased. Therefore, the parameter *Fs* is usually adjusted in the modeling process.

Each graph edge has some spatial position, which is parallel to the respective axis and directions of the wave propagation. If the transverse waves are propagated along the given direction, then the S-type waveguide is put in this edge. If the longitudinal waves are propagated along the given direction, then the P-type waveguide is put in this edge.

Each installed waveguide is adjusted. The adjusted parameters are the waveguide delay, its dumping factor *D*.

When the P-type waveguide is considered, then the number of channels is selected according the ratio $R_{Fs} = Fs/f_{max}$. Approximately, this number is equal to $]20/R_{Fs}[$, where "][" means the integer part of a fraction. Then, each channel parameters, like band range, dumping factor, waveguide delay, are adjusted according to (2.1) or Fig.2.1. Finally, for each waveguide, the intrinsic impedance R is calculated.

The grid configuration defines the adaptor structures. The number of the adaptor ports is equal to the degree of the respective graph node.

For each adaptor the impedances R_i of its ports are assigned. These impedances are equal to the intrinsic impedance of the adjacent waveguides. When the port represents the edge of the body, then the impedance is assigned which represent the boundary conditions in the point under consideration. For example, if it is air condition, then $R \rightarrow \propto$, and if it is connected to the ideal rigid body, then $R = 0$. In the first case, the structure looks like one in Fig. 2.4., in the second case, the structure is the same but the dumping coefficient is a 1.

In the *third step*, the excitation sources and points of measurement are introduced. This is done according to the example in Fig.2.3. Then, the generator of the excitation signal is designed, which generates a considered excitation signal with the sampling frequency Fs. It may be sine wave generator with the given frequency to investigate, for example, the resonance properties of the object. It may be the short impulse to investigate the impulse response of the object under investigations.

In the *fourth step* the model is modeled. Here, the parameters of the model can be adjusted, the excitation signal can be exchanged, the measured signals are sampled and stored for the further analysis.

2.5 Preliminary conclusions

In this section, the method of the hardware simulation of ultrasonic processes in solids is designed. The method consists in representing a solid body in the form of the network of waveguides and adapters, which model the routes of the sound rays propagation and their reflections from the heterogeneities.

Comparing to known methods, the proposed method is distinguished in differentiating the waves as longitudinal waves, and transverse waves, and modeling their propagation in different waveguides. Besides, the waveguides for the longitudinal waves take into account the wave dispersion by using the multichannel network, each channel in which performs the different delay in the separate frequency band.

In the next section, the implementation of the waveguides on the base of the allpass wave digital filter is considered with the goal to implement the waveguides in the FPGA hardware, providing the dynamically tuned characteristics.

3 IMPLEMENTATION OF THE WAVE PROPAGATION MODEL IN HARDWARE

3.1 Modeling the wave propagation in VHDL simulator

The usual computational modeling is performed with the floating point. The model which is built using the method described above, can be performed with the floating point as well. For this purposes, the VHDL language can be used very well. This fact is proven in the works [33] and [34].

Modeling with the floating-point representation of data is the first step of the model investigation. In this step, the appropriate parameter set is selected, and the model is approved due to its adequacy to the real object.

After the preliminary modeling, the decision is made about the model correctness. If the model is simple, then the performance of the usual VHDL simulator is enough for the investigations fulfillment.

The significant acceleration of execution of VHDL-programs (in thousands and more times) is achieved by their translation by the compiler-synthesizer in the FPGA of the configurable computer [35]. In this case, the floating-point calculations must be replaced by integer computations. But integer computations with VHDL can be implemented with arbitrarily large numbers.

For this purpose, all the modeling variables must be scaled and rounded to the integers. Then, the modeling is performed with the integer values, and the modeling results are multiplied to the respective scale factors.

In the process of redesigning the model to the integer arithmetic, it is important to select the bit width of the numbers. It is preferable to select the bit width of the intermediate results less or equal to 18, because in most of FPGAs the 18·18 hardware multipliers are used. Such a bit width is enough for the most situations of modeling.

When the higher precision is needed, then the bit width is selected as 36. But in this situation, the hardware volume is increased up to four times. The modeling speed decreases respectively.

3.2 Implementation of the waveguide model in FPGA

The model, which is built using the method described above, refers to the systems of the digital signal processing. Therefore, potentially it can be implemented in FPGA. But by transferring this model to a one, which is fit FPGA it is not enough to rewrite the VHDL description in the synthesis style. Such a description must to obey the affordances to the pipelined systems.

For example, if the model is described as a set of digital waveguide filters, which is argued in [17], then the resulting model will be to slow or even could not be synthesized. This is explained by the fact, that the classic waveguide filters have very long critical paths and small number of the intermediate result registers, which serve as the pipelining registers.

So, the waveguide model implementation in FPGA must satisfy the conditions of the pipelined datapath design. Such conditions are presented in the works [35], [36].

These methods are based on the structural synthesis of the datapath, describing it at the register transfer level and further conversion to the gate level. The basis of many methods is a representation of the algorithm as a synchronous dataflow graph (SDF) and its transformation [37].

Such SDF optimization techniques as retiming, folding, unfolding and pipelining, are widely used in the design of digital signal processing (DSP) devices [36].

By the high-level synthesis of the computer hardware, the cyclically repeated algorithm is represented by SDF. The actor nodes in it correspond to the algorithm operators, and the edges correspond to variables, which are transferred between actors with a delay to the predetermined number of the algorithm cycles. Thus, each

actor generates and consumes a number of variables (tokens), which is constant from cycle to cycle.

SDF is isomorphic to the graph of the computer structure, which performs a predetermined algorithm. The nodes of such a graph correspond to the computing resources like adders, multipliers, processing units (PUs). The edges correspond to the communication lines, and the labels on them are mapped to the registers. Consequently, SDF is a directed graph $G = (V, E)$, representing the computer structure, where $v \in V$ represent some logic network with delay of d time units. The edge $e \in E$ corresponds to a link and is loaded by $w[e]$ labels, which is equal to the depth of the FIFO buffer.

The minimum duration of the clock cycle T_C is equal to the maximum delay of the signal from one register output to the input of another register, i.e., to the critical path through the adjacent nodes with delays d, for which $w[e] = 0$. It should be noted, that with such a one-to-one mapping of SDF, the duration of the algorithm cycle T_A coincides with the duration of a clock period, i.e., $T_A = T_C$, that in the other algorithm mapping is not respected.

The retiming is such a change of the labels in SDF edges, which does not affect the algorithm results. Usually it is realized as a sequence of elementary retimings, each of them consists of a transferring a group of labels (i.e., registers) from the input edges of some node v to its outputs.

In most cases, it is allowed to increase the latent delay of the algorithm and to insert the additional registers on the inputs or outputs of SDF. After retiming such modified SDF, the pipelined network with low value of T_C is achieved. This technique is called as SDF pipelining.

Consider the design of the filter, shown in Fig.2.10. Its usual representation as the signal flow graph is shown in Fig.3.1. Here, the coefficients $c = (a+1)b$.

The respective SDF is shown in Fig.3.2. This SDF can be mapped directly into the filter structure implemented in FPGA. But it has too long critical path T_C, which is equal to the sum of delays of 2 multipliers and 6 adders.

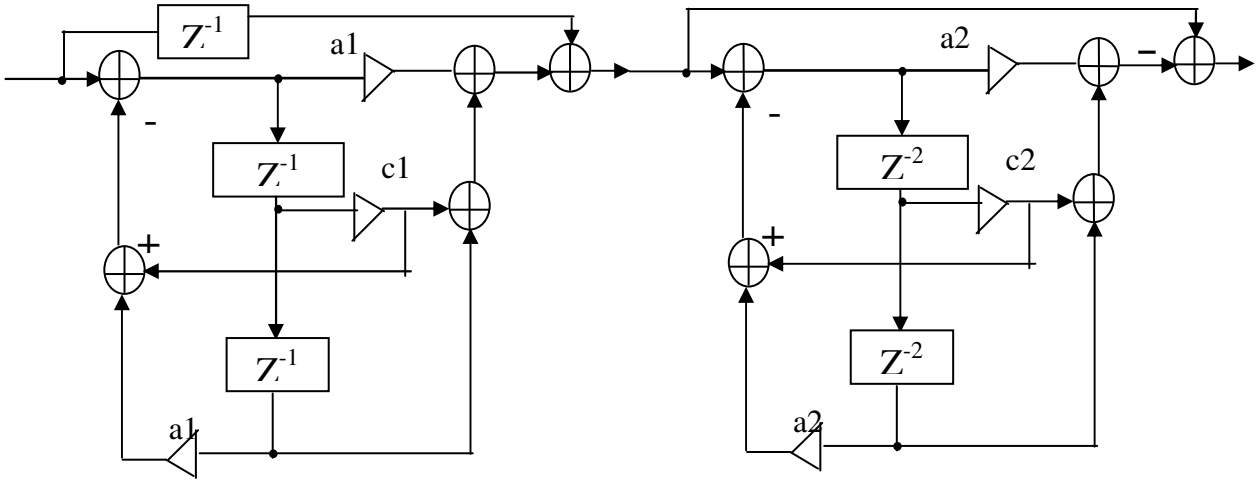

Fig.3.1. Signal flow graph of the pass band filter

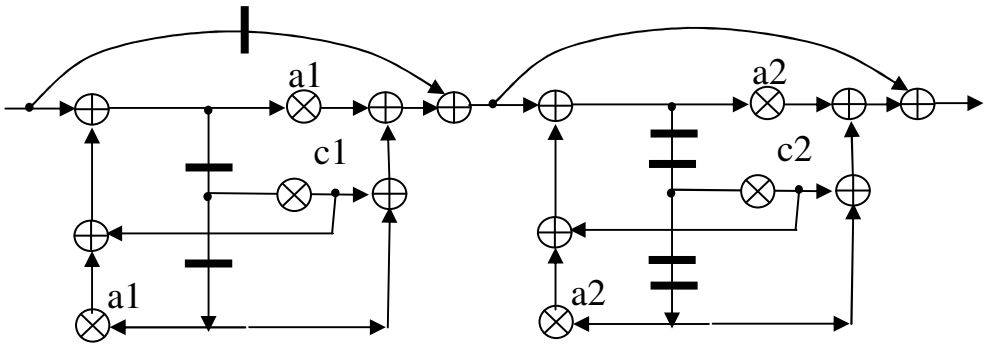

Fig.3.2. Synchronous dataflow graph of the pass band filter

The performance of the filter is increased by the pipelining of the SDF and by its folding. The resulting folded and pipelined SDF is shown in Fig. 3.3. Here, the black bar represents a delay to a single clock cycle, and the white bar represents a multiplexor, which is switched to the opposite direction in each clock cycle. This schema works in two times slower, i.e., $T_A = 2T_C$. But the critical path is decreased to the delay of a single multiplier and two adders, and the hardware volume is in two times less.

The critical path is decreased up to a single multiplier delay if the folded SDF is additionally retimed, as it is shown in Fig.3.4.

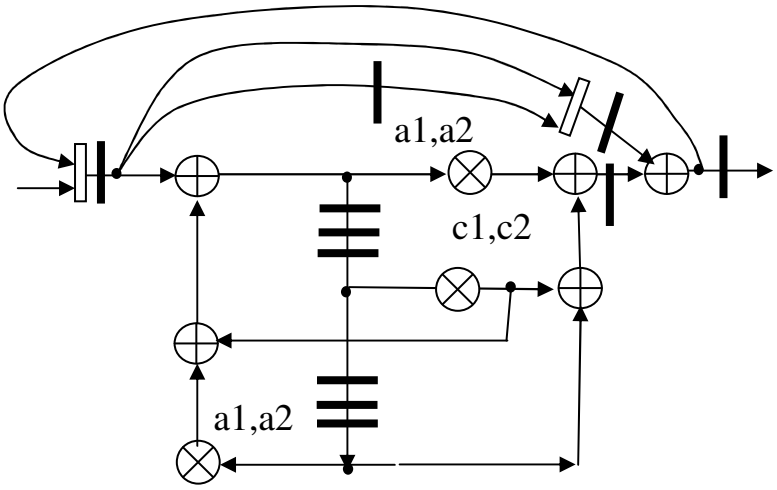

Fig.3.3 Pipelined and folded SDF of the pass band filter

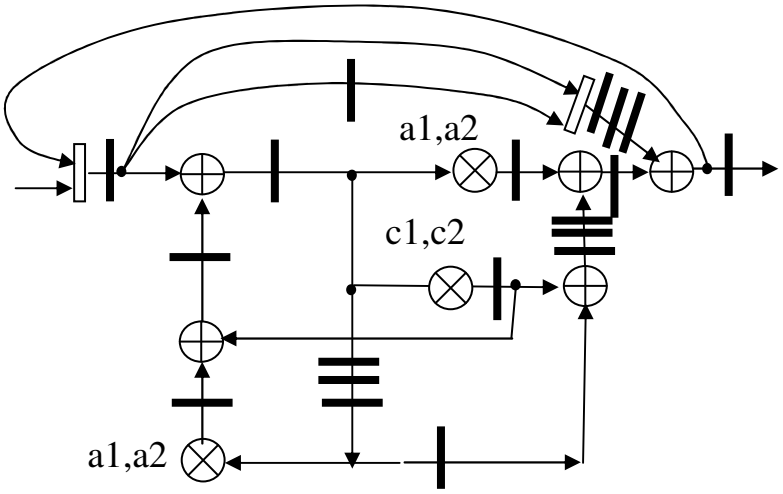

Fig.3.4 Folded SDF after the retiming

Consider the adder, and multiplier delays are 2 ns and 4 ns, respectively. These figures are true for the Xilinx Spartan-6 FPGAs. Then the filter in Fig. 3.2 has the minimum clock period of 20 ns. And the minimum clock period of the filter in Fig. 3.4 is equal only to 4 ns. This means that the optimized network has the speed-

up to 5/2 = 2.5 times and decreased hardware volume in two times. This is an example of the pipelining optimization effectiveness.

Besides, considering the filter is the critical component of the model, then the maximum clock frequency of the modeling is estimated as $Fs = 1/(4 \cdot 2)$ ns = 125 MHz. This means that such hardware model can simulate in real time the rigid body for the frequencies up to 50 MHz.

The respective VHDL files describing this filter are given in addendum.

3.3 Modeling the rod

Below, the designed model of a stiff rod is investigated. The steel rod has the diameter 1 mm and different length. It is damped from each side. It is excited as it is shown in Fig. 2.15.

3.3.1 Testbench for modeling

The rod model is modeled in the special VHDL testbench. Its entity description is shown below

```
entity Cylindre_r_tb is
    generic( TL:time:=500 ns;  --delay of S-waves to TL/20 cycles
             TCR:time:=250 ns; --delay of P-waves to TL/20, cycles
             TD: time:=20 ns;   -- Sampling period
             G1L:real:=0.48;  -- reflection factor left edge
             G1R:real:=0.55; --reflection factor right edge
             GCRL:real:=0.6;-- fraction of P-waves
             GCRR:real:=0.6;-- fraction of S-waves
             DECRL:real:=0.5; -- damping of S-waves
             DECRCR:real:=0.5; --damping of P-waves
             fsampl : INTEGER := 1000;
             fstrt : INTEGER := 0;
             deltaf : INTEGER := 1;
             maxdelay : INTEGER := 200;
             slowdown : INTEGER := 1;
```

```
                magnitud : INTEGER := 100000 );
    end Cylindre_r_tb;
```

As it can be seen, the model has the adjustable parameters like main delay of S- and P-type waves, sampling period, reflection factors, and fraction values different waves.

The testbench provides the deriving the frequency responses. For this purpose, the testbench is used [38]. The frequency response deriving consists in computing a set of reactions to the analytical signal with different frequencies. The magnitude-frequency characteristic and phase-frequency characteristic are distinguished by this approach using the FilterTB component. For this purposes, two copies of the same tested devices must be attached to the FilterTB component.

Then to probe the rod behavior, the excitation impulse is feeded as it is shown in Fig. 2.15. The response to this impulse is measured in the left (DFL) and right (DFR) sides of the rod.

3.3.1 Measurements of the frequency characteristics

It is important to test the amplitude-frequency characteristic, and phase-frequency characteristic of the P-type waveguide for different delays. The magnitude-frequency characteristic and phase-frequency characteristic for the main delays of 125, 250, and 500 nanoseconds are shown in Fig. 3.5, 3.6, and 3.7, respectively. Here, the frequency is exchanged in the range 0 … 0.5Fs, magnitude is in range –40 to 0 db, phase is in range –180°… 180°.

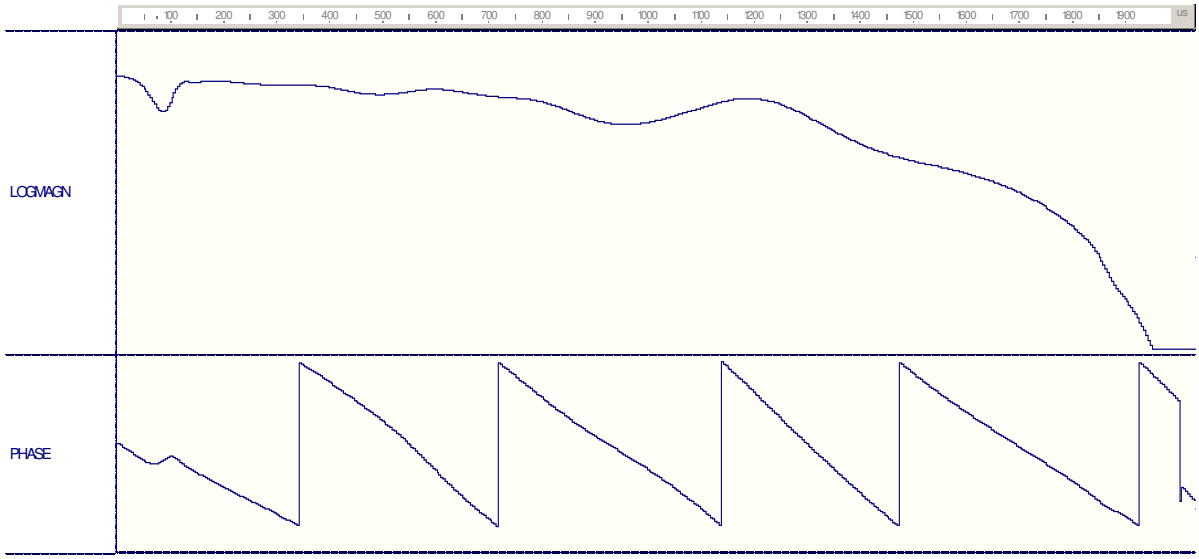

Fig. 3.5. The magnitude- and phase-frequency characteristics of the P-type waveguide with the delay of 125 ns

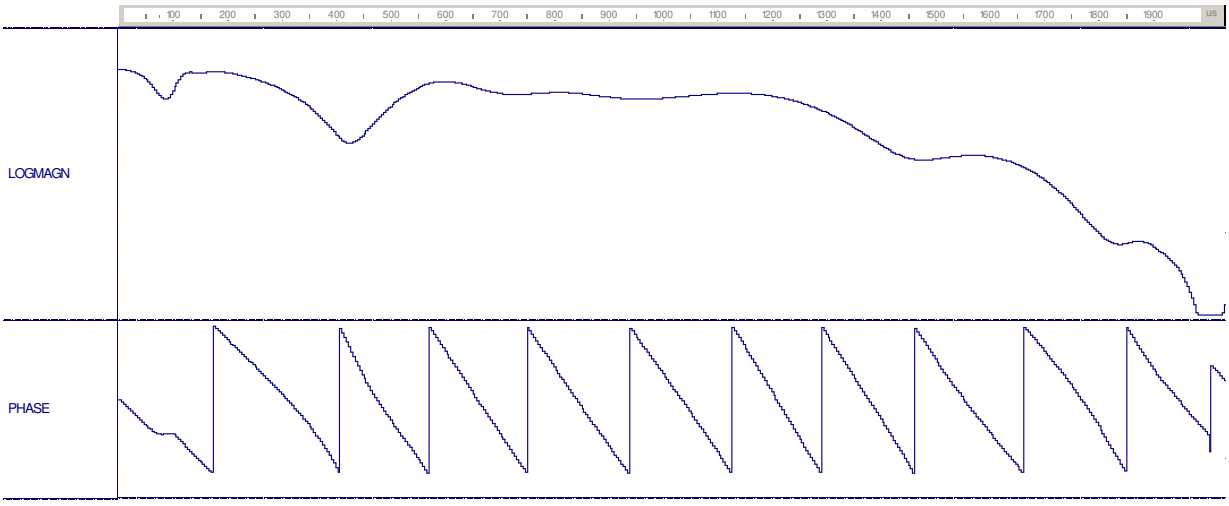

Fig. 3.6. The magnitude- and phase-frequency characteristics of the P-type waveguide with the delay of 250 ns

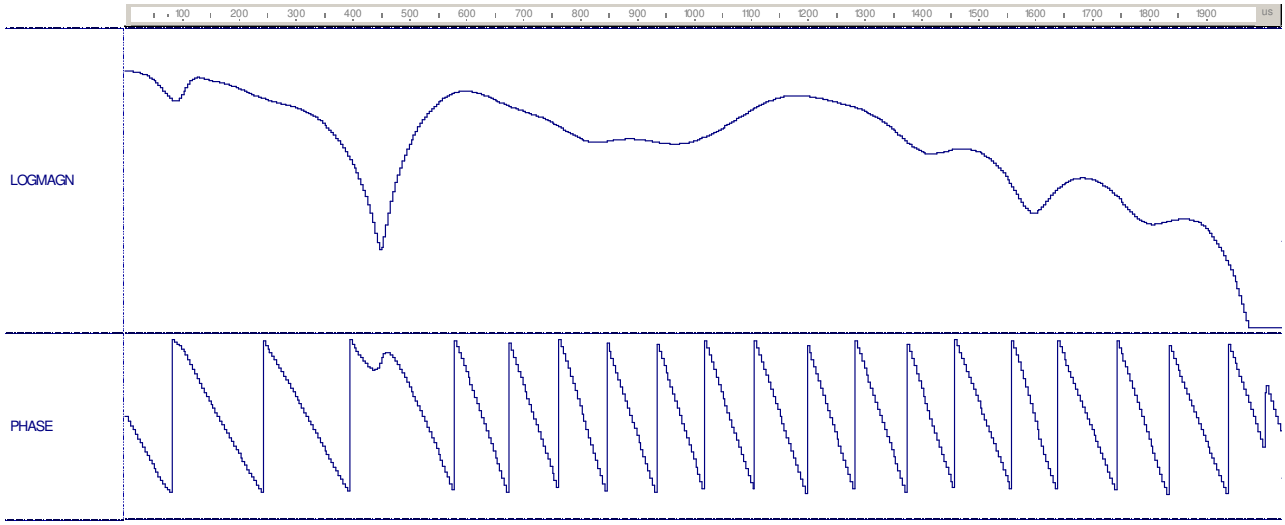

Fig. 3.7. The magnitude- and phase-frequency characteristics of the P-type waveguide with the delay of 500 ns

The analysis of the magnitude- and phase-frequency characteristics shows the following. The magnitude-frequency characteristic has the irregularity ca. ±6 db in the working frequency range. The laydowns occur in this characteristic when the delay increases. It is explained by the wave interference in the edges of the neighboring channels. The phase-frequency characteristics are linear ones except the points where the laydowns occur in the magnitude-frequency characteristics.

3.3.2 Investigation of the model response without the P-channel

The P-channel models the wave dispersion. The results of the modeling the rod using the model without the P-channel show the effectiveness of this cannel use.

In Fig. 3.8, the signals are shown, which are measured in the left and the right sides of a rod model, in which the P-channel is absent. The propagation delay is set as 500 ns.

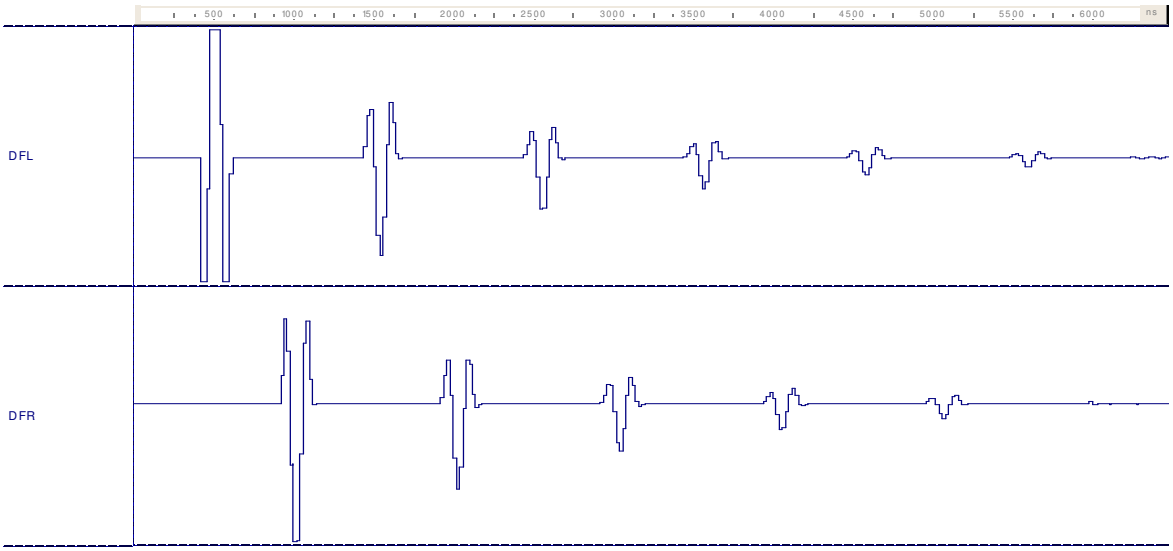

Fig. 3.8. Responses of the rod model without the P-channel for the delay of 500 ns

3.3.3 Investigation of the model response using the full rod model

The results of modeling the rod using the full model is shown in Fig. 3.9, 3.10, and 3.11 for the delays of 500, 1000, and 2000 nanoseconds, respectively.

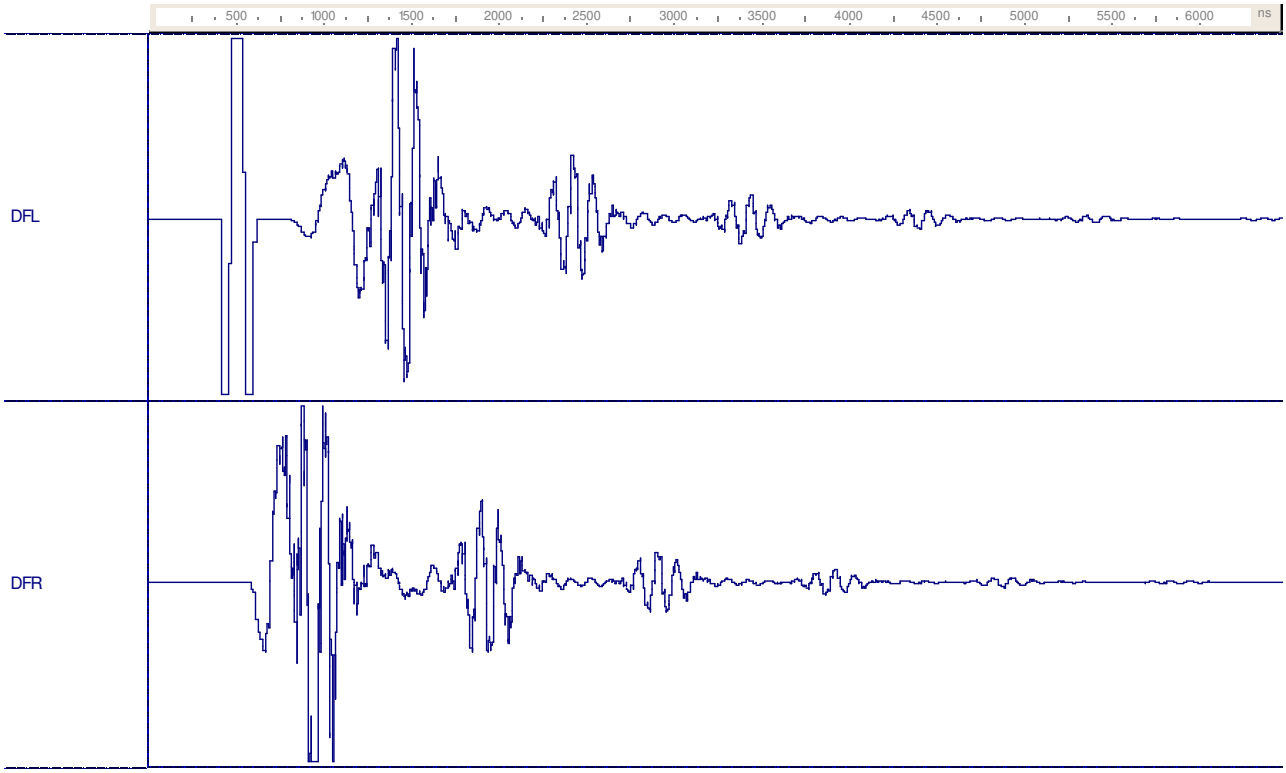

Fig. 3.9. Responses of the rod model without the P-channel for the delay of 500 ns

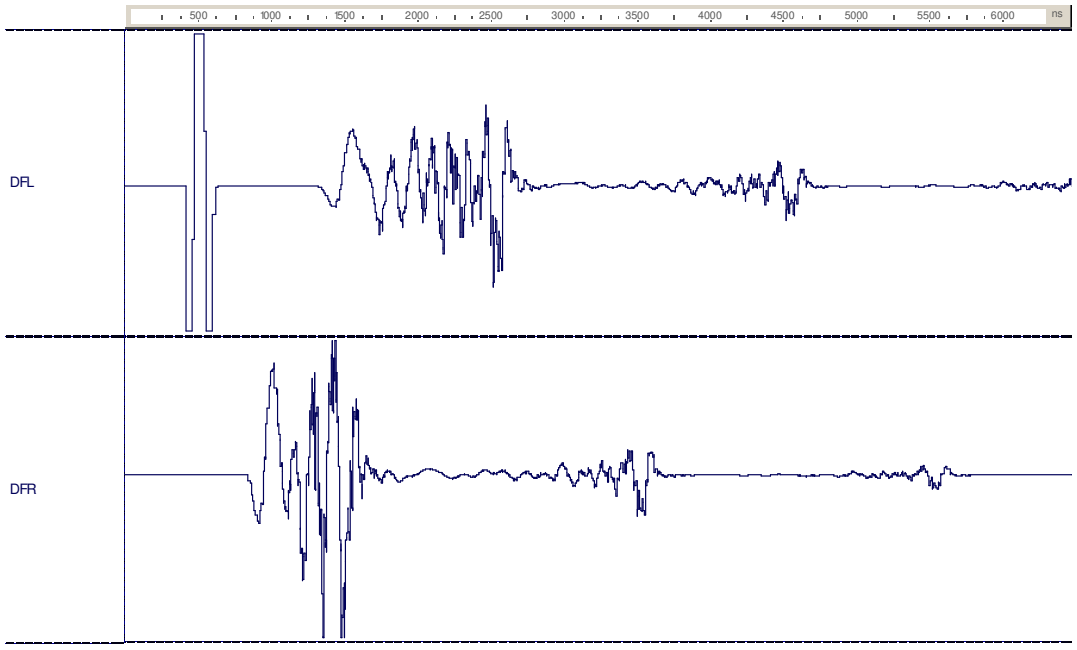

Fig. 3.10. Responses of the rod model without the P-channel for the delay of 1000 ns

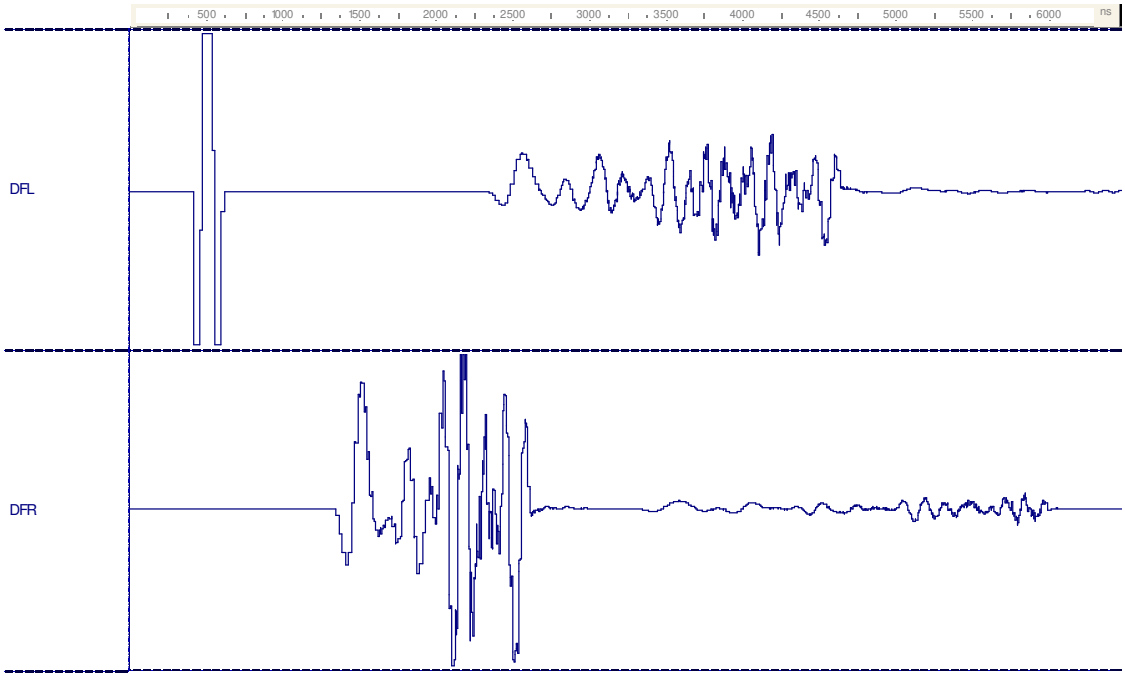

Fig. 3.11. Responses of the rod model without the P-channel for the delay of 2000 ns

The model was actuated by a narrow ultrasound impulse. The diagrams in Fig.3.8 — 3.11 show that the waves are really dispersed after the propagation through the bar model. A set of reflections from the rod edges is shown, which are decayed in time. And the width of the reflected impulse envelope is expanded proportionally to the rod length.

As a conclusion, the model, which is designed according to the proposed method of the hardware simulation of the propagation of ultrasonic waves in a solid, is adequate to the real solid rod.

CONCLUSIONS

This thesis has presented a detailed description and analysis of the digital modeling of the wave propagation in solids, and the development of the method of such modeling implemented in hardware. On the base of the thesis materials the following conclusions are made.

1) The basics of the mathematical modeling, finite difference schema and the finite element method, and method of the digital waveguide modeling were considered. It was defined that the method of the digital waveguide modeling is effective one for simple but precise modeling of the sound wave propagation in the solids, taking into account the nonlinearities, space dimensions of the object. Therefore, the digital waveguide method is taken into account as the perspective method, which can be improved.

2) The method of the hardware simulation of ultrasonic processes in solids is designed. The method consists in representing a solid body in the form of the network of waveguides and adapters, which model the routes of the sound ray propagation and their reflections from the heterogeneities.

Comparing to known methods, the proposed method is distinguished in differentiating the waves as longitudinal waves, and transverse waves, and modeling their propagation in different waveguides. Besides, the waveguides for the longitudinal waves take into account the wave dispersion by using the multichannel network, each channel in which performs the different delay in the separate frequency band.

3) The implementation of the waveguides on the base of the allpass wave digital filter is developed with the goal to implement the waveguides in the FPGA hardware, providing the dynamically tuned characteristics.

4) The developed method was used for design of the model of solid rod, which is described in VHDL, and can be implemented in FPGA. A set of experiments was made with this model, which results show that it is adequate to the

real solid rod, can be tuned in large ranges of its parameters, and can perform the real time modeling when it is implemented in FPGA.

5) The future works at this theme can be directed to the improvement of the model elements, to minimizing its hardware volume, to designing the framework for the model forming and description automatisation, and to design of the interface to the modern configurable computers.

REFERENCES

1. Qasim M. R. Modeling of the wave propagation in the solid bar / A. M. Sergiyenko // Proc. 20-th Int. Conf. System Analysis and Information Technology, SAIT2018, May 21 – 24, 2018, Kyiv. 2018. — P.203–204.
2. Касім М. Р. Апаратне моделювання поширення ультразвукових хвиль у твердому тілі / А. Сергієнко // Праці міжнародної конференції "Безпека, Відмовостійкість, Інтелект", 10-11 травня 2018. — С. 66—69.
3. Самарский А.А. Математическое моделирование: Идеи. Методы. Примеры / А.П. Михайлов / М: Физматлит, 2-е изд. 2005. — 320 с.
4. D'Acuinto B. Computational Methods for PDE in Mechanics. World Sci. Pub. 2004. — 280 p.
5. Ковальчук В. В. Основи наукових досліджень [Текст]: Навчальний посібник / В. В. Ковальчук, Л. М. Моїсеєв. — 3-е вид., перероб. і допов. — К.: ВД «Професіонал», 2005. — 240 с.
6. Самарский А. А. Компьютеры, модели, вычислительный эксперимент. Введение в информатику с позиций матема тического моделирования /Авт. пред..— М.: Наука, 1988.— 176 с.
7. Smith J. O. Physical modeling using digital waveguides // Computer Music Journal, vol. 16, Winter 1992, P. 74–91.
8. Smith J. O. Principles of digital waveguide models of musical instruments // in Applications of Digital Signal Processing to Audio and Acoustics. ξSpringer, 2002. P. 417-466.
9. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. М.: Мир. 1978. — 848 с.
10. Хемминг Р. В. Цифровые фильтры. М.: Сов. Радио. 1980. — 224 с.
11. Morse P. M. Theoretical Acoustics / K. U. Ingard, / McGraw-Hill, New York, 1968.

12. Bergman D. R. Computational Acoustics: Theory and Implementation. J.Wiley and Sons, Ltd. 2018. 296 p.
13. Cremer L. The Physics of the Violin. MIT Press, Cambridge MA, 1984.
14. Karplus K. Digital Synthesis of Plucked String and Drum Timbres / A. Strong // Computer Music J. Vol. 7, No. 2. 1983. P. 43–55.
15. Sullivan C. R. Extending the Karplus-Strong Algorithm to Synthesize Electric Guitar Timbres with Distortion and Feedback // Computer Music J., Vol. 14, 1990, No. 3 , P. 26–37.
16. Rabenstein R. Block-Based Physical Modeling for Digital Sound Synthesis / S. Petrausch, Augusto Sarti, Giovanni De Sanctis, Cumhur Erkut, Matti Karjalainen // IEEE Signal Processing Magazine, Vol. 24, 2007. No 2, P. 42-54.
17. Fettweis A. Wave digital filters: Theory and practice // Proceedings of the IEEE, vol. 74 1986.no. 2, P. 270–327.
18. Bilbao S. Wave and Scattering Methods for Numerical Simulation. John Wiley & Sons, Ltd, Chichester, England, 2004.
19. Karjalainen M. Time-domain physical modeling and real-time synthesis using mixed modeling paradigms // in Proc. Stockholm Music Acoustics Conference, Stockholm, Sweden, Vol. 1, August 2003. – P. 393–396.
20. Karjalainen M. Special Digital Filters for Audio Reproduction / T. Paatero, J. Pakarinen, V.VaËlimËaki // AES 32nd Int. Conf, Hillerııd, Denmark, 2007 Sept. 21–23., 2007, – P. 91–108.
21. Bilbao S. Numerical Sound Synthesis. Finite Difference Schemes and Simulation in Musical Acoustics. John Wiley & Sons, 2009, 430 p.
22. Smith J. O. Generalized Digital Waveguide Networks /D. Rocchesso // IEEE Trans. Speech and Audio Processing, Vol. 11, No. 3, 2003. – P. 242–254.
23. Pierce J. R. A passive nonlinear digital filter design which facilitates physics-based sound synthesis of highly nonlinear musical instruments / S. A. Van Duyne // J. Acoust. Soc. Am. Vol. 101, 1997, No. 2, – P.1120–1126.

24. Kleimola J. Dispersion Modulation Using Allpass Filters // Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx-08), Espoo, Finland, September 1-4, 2008. – P. DAFX-1 – DAFX-5.
25. Petrausch S. Wave digital filters with multiple nonlinearities / R. Rabenstein // in Proc. European Signal Process. Conf. (EUSIPCO), Vol. 12, Vienna, Austria, Sep. 2004. – P. 33-37.
26. Schwerdtfeger T. A multidimensional approach to wave digital filters with multiple nonlinearities / A. Kummert // in Proc. European Signal Process. Conf. (EUSIPCO), Lisbon, Portugal, Sep. 2014. – P. 2405–2409.
27. Kang M. Design space exploration of many-core processors for a high-performance guitar synthesizer / Myeong-Jae Yi, Jong-Myon Kim // 8th International Forum on Strategic Technology (IFOST) 2013, 2013, Vol. 2, – P. 298-301.
28. Aisyah S. FPGA-based sound synthesis by digital waveguide // 6th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), 2015, 2015. – P. 1-6.
29. Pfeifle F. Real-Time Physical Model of a Piano-Hammer String Interaction Coupled to a Soundboard /Int. Symp. On Musical Acoustics, ISMA'2014, Le Mance, France. 2014. – P.391-397.
30. Kolsky H. Stress Waves in Solids. Dover Publications Inc. 2012.
31. Сергиенко А.М. Динамически перестраиваемые цифровые фильтры на ПЛИС / Лесик Т.М. // Электронное моделирование. –2010. –Т.32, №6. –С. 47-56.
32. Lim Y.C. Frequency-response masking approach for digital filter design: Complexity reduction via masking filter factorization / Lian Y. // IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing. –1994. –41. –N4. –P. 518–525.

33. Сергиенко А.М. Особенности VHDL как языка параллельного программирования / Сергиенко А.М.// Электрон. моделирование. –Т. 25.–2003.-№ 3.-с.115-123.
34. Сергиенко А.М. VHDL против Matlab'a / Сергиенко А.М. // Argc&Argv. Программирование / информационные технологии / стандарты. 2005, № 1(58), стр. 40-46.
35. Mangione-Smith W.H. Seeking Solutions in Configurable Computing. / Hutchings B., Andrews D., De Hon A., et al. // Computer. -1977. -№ 12. -P. 38-43.
36. Сергиенко А.М. Отображение периодических алгоритмов в программируемые логические интегральные схемы / Симоненко В.П.// Электрон. моделирование.–2007, Т. 29, № 2.-С. 49-61.
37. Khan S. A. Digital Design of Signal Processing Systems. John Wiley & Sons. 2011. Edwards S. Design of Embedded Systems: Formal Models, Validation, and Synthesis / , L. Lavagno, E.A. Lee, A. Sangiovanny-Vincentelli // Proc. IEEE, vol.85, pp.366–390, March 1997.
38. Sergiyenko A. M. Testbench for the filter testing. 2017. – Available at: http://kanyevsky.kpi.ua/en/useful-ip-cores/testbench-for-the-filter-testing/

APPENDICES

APPENDIX 1

VHDL programs of the solid modeling

```
-------------------------------------------------------------------------------------------------
-- Title       : Test Bench for filtertb_r
-- Design      : fanera
-- Author      : Quasimi
-- Company     : КПИ
-------------------------------------------------------------------------------------------------
-- File        : $DSN\src\TestBench\filtertb_r_TB.vhd
-- From        : $DSN\src\FilterTB_R.vhd
-------------------------------------------------------------------------------------------------
-- Description : Automatically generated Test Bench for Cylindre_r_tb
-------------------------------------------------------------------------------------------------

library ieee;
use ieee.math_real.all;
use ieee.std_logic_1164.all;
entity Cylindre_r_tb is
        generic(
           TL:time:=1000 ns;  --delay of S-waves to TL/20 тактов
                TCR:time:=500 ns; --delay of P-waves to TL/20, cycles
                TD: time:=20 ns;   -- Sampling period
                G1L:real:=0.48;  -- reflection factor left edge
                G1R:real:=0.55; --reflection factor right edge
                GCRL:real:=0.6;-- fraction of P-waves
                GCRR:real:=0.6;-- fraction of S-waves
                DECRL:real:=0.6; -- damping of S-waves
                DECRCR:real:=0.5; --damping of P-waves
                fsampl : INTEGER := 1000;
                fstrt : INTEGER := 0;
                deltaf : INTEGER := 1;
                maxdelay : INTEGER := 200;
                slowdown : INTEGER := 1;
                magnitud : INTEGER := 100000 );
end Cylindre_r_tb;

architecture TB_ARCHITECTURE of Cylindre_r_tb is
        -- Component declaration of the tested unit
        component filtertb_r
                generic(
                fsampl : INTEGER := 1000;
                fstrt : INTEGER := 0;
                deltaf : INTEGER := 4;
                maxdelay : INTEGER := 11;
                slowdown : INTEGER := 1;
                magnitud : INTEGER := 1000 );
        port(
                CLK : in std_logic;
                RST : in std_logic;
                START : in std_logic;
                RERSP : in REAL;
                IMRSP : in REAL;
                REO : out REAL;
```

```
                IMO : out REAL;
                FREQ : out INTEGER;
                MAGN : out INTEGER;
                LOGMAGN : out REAL;
                PHASE : out REAL;
                ENA : inout std_logic );
        end component;

component Poper_Del is
        generic(TCR:time:=1000 ns; --задержка поперечных волн на Fd/5, наносекунд
                TD: time:=20 ns);
        port(
                CLK : in STD_LOGIC;
                RST : in STD_LOGIC;
                DF : in real;
                DB : out real
                );
end    component;
component  Cylindre is
        generic(TL:time:=1000 ns; --задержка продольных волн, тактов
                TCR:time:=2000 ns; --задержка поперечных волн на Fd/10, тактов
                TD: time:=20 ns;   -- период дискретизации
                G1L,G1R:real:=0.8;  --к-ты отражения от торцов
                GCRL,GCRR:real:=0.2; --к-ты доли поперечных волн
                DECRL:real:=0.9; --затухание продольных
                DECRCR:real:=0.9 --затухание поперечных
                );
        port(
                CLK : in STD_LOGIC;
                RST : in STD_LOGIC;
                DFL : in real;
                DBR : in real;
                DBL : out real:=0.0;
                DFR : out real:=0.0
                );
end component ;

       constant Fb:real:=0.026;
        constant DF:real:=0.01;
        Constant          DT:time:=00 ns;
        Constant G:real:=0.5;
        signal CLK : std_logic:='1';
        signal RST : std_logic:='0';
        signal START : std_logic;
        signal RERSP : REAL;
        signal IMRSP : REAL;
        signal ENA : std_logic;
        signal REO : REAL;
        signal IMO : REAL;
        signal FREQ : INTEGER;
        signal MAGN : INTEGER;
        signal LOGMAGN : REAL;
        signal PHASE,imp,impresp,imps : REAL:=0.0;
        signal otr,dotr:real:=0.0;

begin
        CLK<=not clk after 10 ns;
        RST<= '1', '0' after 205 ns;
```

```
  imp<=0.0, -1.0 after 400 ns, -2.0 after 420 ns,
  0.0 after 440 ns, 2.0 after 460 ns,
  3.0 after 480 ns, 2.0 after 500 ns,
  0.0 after 520 ns, -2.0 after 540 ns,
   -1.0 after 560 ns, 0.0 after 580 ns;

        UUT : filtertb_r
                generic map (
                        fsampl => fsampl,
                        fstrt => fstrt,
                        deltaf => deltaf,
                        maxdelay => maxdelay,
                        slowdown => slowdown,
                        magnitud => magnitud
                )

                port map (
                        CLK => CLK,
                        RST => RST,
                        START => START,
                        RERSP => RERSP,
                        IMRSP => IMRSP,
                        REO => REO,
                        IMO => IMO,
                        FREQ => FREQ,
                        MAGN => MAGN,
                        LOGMAGN => LOGMAGN,
                        PHASE => PHASE,
                        ENA => ENA
                );

re: Poper_Del
        generic map(TCR=>1000 ns, --задержка
                TD=>20 ns)
        port map(CLK,RST,
        DF =>REO,
        DB=>RERSP
                );

im: Poper_Del
        generic map(TCR=>1000 ns, --задержка                TD=>20 ns
                )
        port map(CLK,RST,
                DF =>ImO,
                DB=>IMRSP
                );
cyl: Cylindre
        generic map(TL=>tl, --задержка
                TCR=>tcr, --задержка
                TD=>td,   -- период дискретизации
                G1L=>g1l,
                G1R=>g1r,  --к-ты отражения от торцов
                GCRL=>gcrl,
                GCRR=>gcrr, --к-ты доли поперечных волн
                DECRL=>decrl, --затухание продольных
                DECRCR=>decrcr --затухание поперечных
                )
        port map(CLK,RST,
```

```
                DFL =>imps,
                DBR =>dotr,
                DBL =>impresp,
                DFR=> otr
                );
 dotr<=transport -otr  after 20 ns;
 imps<= transport imp - impresp after 20 ns;
end TB_ARCHITECTURE;

-------------------------------------------------------------------------------------------
-- Title       : Cylindre
-- Design      : fanera
-- Author      : 0
-- Company     : 0
-------------------------------------------------------------------------------------------
-- File        : Cylindre.vhd
-------------------------------------------------------------------------------------------
-- Description :Модель цилиндра с пл.зпт.
-------------------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity Cylindre is
        generic(TL:time:=200 ns; --задержка продольных волн
                TCR:time:=400 ns; --задержка поперечных волн
                TD: time:=20 ns;   -- период дискретизации
                G1L,G1R:real:=0.3;  --к-ты отражения от торцов
                GCRL,GCRR:real:=0.1; --к-ты доли поперечных волн
                DECRL:real:=0.9; --затухание продольных
                DECRCR:real:=0.9 --затухание поперечных
                );
        port(
                CLK : in STD_LOGIC;
                RST : in STD_LOGIC;
                DFL : in real;
                DBR : in real;
                DFR : out real;
                DBL : out real
                );
end Cylindre;

architecture Cylindre of Cylindre is

        component       Poper_Del is
                generic(TCR:time:=1000 ns; --задержка поперечных волн
                TD: time:=20 ns;            -- период дискретизации
                DECRCR:real:=1.0);
                port(
                        CLK : in STD_LOGIC;
                        RST : in STD_LOGIC;
                        DF : in real:=0.0;
                        DB : out real:=0.0
                        );
        end component ;

component     Conn3_r is
        generic(g1:real:=0.5; --доля энергии, затекающая с 1 входа A
        g2: real:=0.3
        );
```

```
        port(
        AI : in real:=0.0;
        BI : in   real:=0.0;
        CI : in   real:=0.0;
        AO : out  real:=0.0;
        BO : out   real:=0.0;
        CO : out   real:=0.0
        );
        end component;


        signal DCRFL,DCRBL,DLFL,DLBL:real:=0.0;
        signal DCRFR,DCRBR,DLFR,DLBR:real:=0.0;
        Signal DR1,DL1:real:=0.0;
begin
        LEFT:Conn3_r
        generic map(g1=>G1L, --доля энергии, затекающая с 1 входа A
                g2=>GCRL
                )
        port map(
        AI => DFL,
        BI => DCRBL,
        CI => DLBL,
        AO => DBL,
        BO => DCRFL,
        CO => DLFL
        );

        Forward_CR:Poper_Del
        generic map(TCR=>TCR, --задержка поперечных волн на Fd/5, наносекунд
        TD=>TD,
                DECRCR=>DECRCR
        )
        port map(CLK,RST,
                DF =>DCRFL,
                DB=>DCRFR
                );

        Backward_CR:Poper_Del
        generic map(TCR=>TCR, --задержка поперечных волн на Fd/5, наносекунд
        TD=>TD,
        DECRCR=>DECRCR
        )
        port map(CLK,RST,
                DF =>DCRBR,
                DB=>DCRBL
                );

        --Упрощенная задержка продольных волн
        DR1<= transport DLFL*DECRL after  TD;
        DLFR<= transport (DLFL*DECRL+DR1)/2.0 after  TL;
        DL1<= transport DLBR*DECRL after  TD;
        DLBL<= transport (DLBR*DECRL+DL1)/2.0 after  TL;

        RIGHT:Conn3_r
        generic map(g1=>G1R, --доля энергии, затекающая с 1 входа A
                g2=>GCRR
                )
```

```
        port map(
        AI => DBR,
        BI => DCRFR,
        CI => DLFR,
        AO => DFR,
        BO => DCRBR,
        CO => DLBR
        );
end Cylindre;

------------------------------------------------------------------------------------------------
-- Title       : Cylindre
-- Design      : fanera
-- Author      : 0
-- Company     : 0
--
------------------------------------------------------------------------------------------------
-- File        : Poper_del.vhd
------------------------------------------------------------------------------------------------
-- Description :Модель задержки поперечных волн
--          период дискретизации=20 нс
------------------------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity Poper_Del is
        generic(TCR:time:=1000 ns; --задержка поперечных волн
        TD: time:=20 ns; -- период дискретизации
        DECRCR:real:=1.0);
        port(
                CLK : in STD_LOGIC;
                RST : in STD_LOGIC;
                DF : in real;
                DB : out real:=0.0
                );
                begin
                assert (TCR>2*TD) report "Слишком короткий цилиндр"
                severity failure;
end Poper_Del;

architecture Realn of Poper_Del is

        constant G1:real:=3.0; --к-т затухания на Fd*0.025
        constant G2:real:=2.8;--0.99; --к-т затухания на Fd*0.035                5
        constant G3:real:=0.156;--0.95; --к-т затухания на Fd*0.05
        constant G4:real:=0.078;--0.9; --к-т затухания на Fd*0.07
        constant G5:real:=0.65;--0.75; --к-т затухания на Fd*0.1
        constant G6:real:=1.560;--0.5 --к-т затухания на Fd*0.14
        constant G7:real:=0.78;--0.5 --к-т затухания на Fd*0.2
        constant G8:real:=1.8;--0.5 --к-т затухания на Fd*0.35

        constant FB1:real:=0.020;  --центральные частоты
        constant FB2:real:=0.03;
        constant FB3:real:=0.05;
        constant FB4:real:=0.07;
        constant FB5:real:=0.1;
        constant FB6:real:=0.15;
        constant FB7:real:=0.2;
        constant FB8:real:=0.3;
```

```
        constant DF1:real:=0.04;          -- ширина полосы частот
        constant DF2:real:=0.04;      --чтоб -3 дб в др. канале
        constant DF3:real:=0.04;
        constant DF4:real:=0.04;
        constant DF5:real:=0.05;
        constant DF6:real:=0.05;
        constant DF7:real:=0.06;
        constant DF8:real:=0.07;

        constant DT1:time:=TCR*1.010 -6.0*TD;--0 ns;  --40 taktow
        constant DT2:time:=TCR*1.030 -6.0*TD;--0 ns;  --30 taktow
        constant DT3:time:=TCR*1.081 -2.0*TD;--0 ns;  --10 taktow
        constant DT4:time:=TCR*1.176 -2.0*TD;--0 ns;  --5 taktow
        constant DT5:time:=TCR*1.316- 2.0*TD;  --4 taktow
        constant DT6:time:=TCR*1.564 -2.0*TD;--0 ns;  --3 taktow
        constant DT7:time:=TCR*1.667 -2.0*TD;--0 ns;  --2 taktow
        constant DT8:time:=TCR*1.735 -2.0*TD;--0 ns;  --1 taktow

        component  BandDel_r is
                generic(Fb:real;
                        DF:real;
                        DT:time;
                        G:real
                        );
                port(CLK,RST: in STD_LOGIC;
                        DI: in real:=0.0;
                        DO: out real:=0.0) ;
        end component;
        signal DCR,DO1,DO2,DO3,DO4,DO5,DO6,DO7,DO8:real:=0.0;

begin
        DCR<=DF;

        f1:     entity BandDel_r(allpassL) generic map (Fb=>Fb1,
                DF=>DF1,
                DT=>DT1,
                G=>G1
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO1) ;

        f2:entity BandDel_r(allpass6)     generic map (Fb=>Fb2,
                DF=>DF2,
                DT=>DT2,
                G=>G2
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO2) ;
        f3:entity          BandDel_r(allpass2) generic map (Fb=>Fb3,
                DF=>DF3,
                DT=>DT3,
                G=>G3
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO3) ;
```

```
        f4:entity BandDel_r(allpass2)      generic map (Fb=>Fb4,
                DF=>DF4,
                DT=>DT4,
                G=>G4
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO4) ;
        f5:entity          BandDel_r(allpass2) generic map (Fb=>Fb5,
                DF=>DF5,
                DT=>DT5,
                G=>G5
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO5) ;
        f6:entity BandDel_r(allpass2) generic map (Fb=>Fb6,
                DF=>DF6,
                DT=>DT6,
                G=>G6
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO6) ;
        f7:entity BandDel_r(allpass2) generic map (Fb=>Fb7,
                DF=>DF7,
                DT=>DT7,
                G=>G7
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO7) ;
        f8:entity BandDel_r(allpass2) generic map (Fb=>Fb8,
                DF=>DF8,
                DT=>DT8,
                G=>G8
                )
        port map(CLK,RST,
                DI=>DCR,
                DO=>DO8) ;

        DB<=(DO1+DO2+DO3+DO4+DO5+DO6+DO7+DO8)*DECRCR/3.0;--

end realn;
```

```
----------------------------------------------------------------------------------------------------
-- Title       : Cylindre
-- Design      : fanera
-- Author      : 0
-- Company     : 0
----------------------------------------------------------------------------------------------------
-- File        : Prod_del.vhd
----------------------------------------------------------------------------------------------------
-- Description :Модель задержки продольных волн
--          период дискретизации=20 нс
----------------------------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.math_real.all;

entity Prod_Del is
        generic(TL:time:=1000 ns; --задержка продольных волн,наносекунд
        TD: time:=20 ns; -- период дискретизации
        DECRL:real:=1.0;  --затухание
        DeltF:real:=0.1); -- крутизна полосы задержания верхних частот,
        port(
                CLK : in STD_LOGIC;
                RST : in STD_LOGIC;
                DF : in real;
                DB : out real:=0.0
                );
                begin
                assert (TL>=1.0*TD) report "2.5 Слишком короткая задержка"
                severity failure;
                        assert (DeltF>0.06) report "Слишком крутой фильтр"
                severity failure;
end Prod_Del;

architecture Realn of Prod_Del is
   constant FB:real:=0.2;--0.55-DeltF; --полоса среза ФНЧ
        type arr is array( integer range <>) of real;
        constant tg:real:=TAN(Math_2_PI*DeltF/2.0);
        constant a:real:=(1.0-tg)/(1.0+tg);
        constant b:real:= -COS(Math_2_PI*Fb);
        signal RGO,DD,DD2 : real:=0.0;
        signal Z3,Z1,Z2: real:=0.0;

begin
        FNCH:process(CLK,RST)
                variable del1,del2,c,t,t1,t2,tt1,tt2: real;
        begin
                if  RST='1' then
                        Z1<=0.0;
                        Z2<=0.0;
                        Z3<=0.0;
                        RGO<=0.0;
                elsif CLK='1' and CLK'event then

                        t2:=(DF-Z2)*a;
                        tt2:=DF+t2;
                        t1:=(tt2-Z1)*b;
                        tt1:=tt2+t1;
                        Z1<=tt1;
```

```
                    Z2<= t1+Z1;
                    Z3<=DF;
                    RGO<=(Z2+t2)+Z3; -- output register

            end if;
        end process;

        DB<=transport DECRL*RGO/2.0 after (TL-TD*2.5);

end realn;
```

```
----------------------------------------------------------------------------------------------------
-- Title       : allpass2
-- Design      : allpass
-- Author      : 0
-- Company     : 0
----------------------------------------------------------------------------------------------------
-- File        : allpass2.vhd
----------------------------------------------------------------------------------------------------
-- Description : канал задержки в полосе частот
--      с центральной частотой Fb ,    (  1 = Fдискретизации)
--      полосой частот DF,
--      задержкой  DT наносекунд +1 такт,
--                  затуханием G   =0.9 .
----------------------------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.math_real.all;

entity BandDel_r is
        generic(Fb:real;
                DF:real;
                DT:time;
                G:real
                );
        port(CLK,RST: in STD_LOGIC;
                DI: in real;
                DO: out real:=0.0) ;
end BandDel_r;

architecture allpass2 of BandDel_r is
        type arr is array( integer range <>) of real;
        constant tg:real:=TAN(Math_2_PI*DF/2.0);
        constant a:real:=(1.0-tg)/(1.0+tg);         --0.75;--
        constant b:real:= -COS(Math_2_PI*Fb); -- -0.6875;--
        signal RGO,r1,r2,r3 : real:=0.0;
        signal Z11,Z12,Z1,Z2: real:=0.0;
        constant K1:real:=-256.0;---128; --k=-0.25

begin
        DP:process(CLK,RST)
                variable del1,del2,c,t,t1,t2,tt1,tt2: real;
        begin
                if  RST='1' then
                        Z1<=0.0;
                        Z2<=0.0;
                        RGO<=0.0;
                elsif CLK='1' and CLK'event then

                        t2:=(DI-Z2)*a;
                        tt2:=DI+t2;
                        t1:=(tt2-Z1)*b;
                        tt1:=tt2+t1;
                        Z1<=tt1;
                        Z2<= t1+Z1;
                        --z1<=z11;
                        --z2<=z12;
                        RGO<=-(Z2+t2)+DI; -- output register
                        r1<=rgo;
```

```
                r2<=r1;
                r3<=r2;
        end if;
    end process;


    DO<=transport G*(RGO)/2.0 after DT ;

end allpass2;

architecture allpass6 of BandDel_r is
    constant tg:real:=TAN(Math_2_PI*DF/2.0);
    constant a:real:=(1.0-tg)/(1.0+tg);
    constant b:real:= -COS(Math_2_PI*Fb);
    constant b1:real:= -COS(Math_2_PI*Fb*2.0);
    signal RGO : real:=0.0;
    signal Z1,Z2: real:=0.0;
    signal z11,z12,z21,z22,RGtmp:real;
begin
    DP:process(CLK,RST)
        variable del1,del2,c,t,t1,t2,tt1,tt2: real;
    begin
        if  RST='1' then
            Z1<=0.0;
            Z2<=0.0;
            RGtmp<=0.0;
        elsif CLK='1' and CLK'event then

            t2:=(DI-Z2)*a;
            tt2:=DI+t2;
            t1:=(tt2-Z1)*b;
            tt1:=tt2+t1;
            Z1<=tt1;
            Z2<= t1+Z1;
            RGtmp<=-(Z2+t2)+DI; -- output register

        end if;
    end process;

    DP2:process(CLK,RST)
        variable del1,del2,c,t,t1,t2,tt1,tt2: real;
    begin
        if  RST='1' then
            Z11<=0.0;
            Z12<=0.0;
            Z21<=0.0;
            Z22<=0.0;
            RGO<=0.0;
        elsif CLK='1' and CLK'event then

            t2:=(    RGtmp-Z22)*a;
            tt2:=    RGtmp+t2;
            t1:=(tt2-Z12)*b1;
            tt1:=tt2+t1;
            Z11<=tt1;
            z12<=z11;
            Z21<= t1+Z12;
            z22<=z21;
```

```
                RGO<=-(Z22+t2)+       RGtmp; -- output register

        end if;
    end process;
    DO<=transport G*RGO*0.25 after DT ;

end allpass6;

architecture allpassL of BandDel_r is
    constant tg:real:=TAN(Math_2_PI*DF/2.0);
    constant a:real:=(1.0-tg)/(1.0+tg);
    constant b:real:= -COS(Math_2_PI*Fb*6.0);
    constant b1:real:= -COS(Math_2_PI*Fb*3.4);
    signal RGO : real:=0.0;
    signal Z1,Z2: real:=0.0;
    signal z11,z12,z21,z22,RGtmp,R11,R12,R2:real;

begin
    DP:process(CLK,RST)
        variable del1,del2,c,t,t1,t2,tt1,tt2: real;
    begin
        if  RST='1' then
            Z1<=0.0;
            Z2<=0.0;
            RGtmp<=0.0;
            R2<=0.0;
        elsif CLK='1' and CLK'event then

            t2:=(DI-Z2)*0.156;--a*1.5;
            tt2:=DI+t2;
            t1:=(tt2-Z1)*(-0.8);--b;
            tt1:=tt2+t1;
            Z1<=tt1;
            Z2<= t1+Z1;
            R2<=DI;
            RGtmp<=(Z2+t2)+R2; -- output register

        end if;
    end process;

    DP2:process(CLK,RST)
        variable del1,del2,c,t,t1,t2,tt1,tt2: real;
    begin
        if  RST='1' then
            Z11<=0.0;
            Z12<=0.0;
            Z21<=0.0;
            Z22<=0.0;
            RGO<=0.0;
            R11<=0.0;
            R12<=0.0;
        elsif CLK='1' and CLK'event then

            t2:=(    RGtmp-Z22)*a;
            tt2:=    RGtmp+t2;
            t1:=(tt2-Z12)*b1;
            tt1:=tt2+t1;
            Z11<=tt1;
```

```
                    z12<=z11;
                    Z21<= t1+Z12;
                    z22<=z21;
                    R11<=RGtmp;
                    R12<=R11;
                    RGO<=(Z22+t2)+       R12; -- output register

            end if;
        end process;

        DO<=transport G*RGO*0.25 after DT ;
end allpassL;
```

APPENDIX 2

Copies of publications

***Sergiyenko A. M.*[1], *Qasim M. R.*[1]**
[1]*Computer Engineering Department of NTUU "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*

Modeling of the wave propagation in the solid bar

Introduction. The acoustic processes in solids are usually modeled in the computers using the finite difference method, which affords the huge computational resources. On the contrary, the digital waveguide (DWG) method provides the high-speed computations utilizing much less computation volume [1]. A lot of successful examples of the DWG implementation are shown in modeling and sound synthesis of the string and wind musical instruments [2,3]. But this model does not take into account the dispersion in the sound propagation.

In this work, a DWG method modification is proposed which provides the modeling of the sound dispersion in the solid bar.

DWG model basics. DWG is a digital model of the wave propagation in the ideal waveguide. It is based on the principles, described in the work [4], but adapted to the sound wave modeling. The forward $f_i = Rv_f$ and backward $b_i = -Rv_b$ waves are considered in the i-th point of the waveguide, where f_i and b_i are the pressure of the forward and backward (reflected) waves, respectively, R is the wave impedance, v_f, v_b are the particle velocities. For the solid cylinder with the cross-section A the impedance is equal to $R = \varrho c/A$, where ϱ is the matter density, c is the velocity of the longitudinal waves. The real pressure value is equal to $u = f_i + b_i$.

In the DWG model, the signals are sampled at a sampling frequency of F_s which is at least two times greater than the maximum considered frequency. Then the i-th section of the waveguide of the length L looks like two delay lines at $n = L/(c_o F_s)$ cycles, where c_o is the wave velocity.

The homogeneous parts of the waveguides are connected together by the adapters, also called as the scattering nodes. The adapter function is designed to satisfy the Kirchhoff's law. For the case of joining two waveguides:

$$v_b 1 = rvf_1 + (1-r)v_f 2; v_b 2 = (1+r)v_f 1 - rv_f 2; \tag{1}$$

where $r = (R_2 - R_1)/(R_2 + R_1)$ is the reflection coefficient. Similarly, when three waveguides are connected, then the adapter node calculates the following formulas:

$$\begin{aligned} A_o &= (2g_1 - 1)A + 2g_2 B + 2g_3 C; \\ B_o &= 2g_1 A + (2g_2 - 1)B + 2g_3 C; \\ C_o &= 2g_1 A + 2g_2 B + (2g_3 - 1)C; \end{aligned} \tag{2}$$

where A, B, C are the waves entering the node, A_o, B_o, C_o are the waves leaving the node, g_1, g_2, g_3 are the specific impedances of the waveguides, and $g_1 + g_2 + g_3 = 1, g_i = R_i/(R_1 + R_2 + R_3)$.

DWG model improvement. Due to the theory, the primary longitudinal wave with the phase velocity c_p, and the secondary transverse wave with the velocity $c_s, c_s < c_p$ are propagated in the solid bars [5]. The waves of different types can be transformed into each other interacting with the media boundaries. Besides, the longitudinal waves have a dispersion, i.e. its velocity c_p depends on the wave length λ. The velocity c_p in a cylinder is approximated as follows [5]:

$$c_p = c_o(1 - \nu^2 p^2 a^2/\lambda^2) = f(\lambda), \tag{3}$$

where ν is the Poisson ratio, which is equal to 0.29 for the steel, a is the cylinder radius. Therefore, the DWG model has to be corrected according to this formula.

The modified DWG model of a bar is illustrated by Fig.1. This model consists of the left LA and the right RA adapters, the waveguide P of longitudinal waves, and the waveguide S

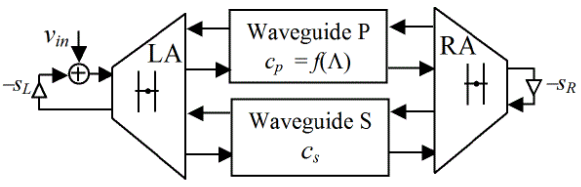

Figure 1. Structure of the solid bar model

of transverse waves. Each three port adapter
compute the equations (2). Its port, which means the bar end, is connected to the network, which implements the wave reflection with the suppression factor s_L or s_R due to (1). The actuating signal v_{in} is fed to the model through an adder.

The waveguide S performs the delay to the given number of clock cycles and some wave attenuation. The waveguide P does the same but it has a set of channels. Each of them has the separate frequency band, delay, and attenuation depending on the respective wave length λ according to (3).

Experimental results. Up to eight channels of the waveguide P are implemented on the basis of the allpass filters described in [6]. Each channel has the dynamically tunable band pass filter, which is implemented as is shown in [7]. The resulting amplitude-frequency characteristic of the waveguide S is shown in Fig.2. Here, the frequency f is measured in parts of the sampling frequency F_s, the characteristics of two neighboring channels are shown as well.

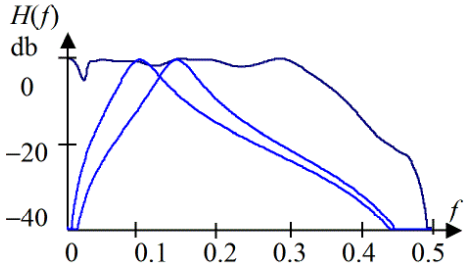

Figure 2. Amplitude-frequency characteristic of the waveguide P

The model was actuated by a narrow ultrasound impulse, and the velocity v was measured in the output of the adder node (see Fig.1). The time diagram of it which represents the wave signal is shown in Fig.3. This diagram shows that the waves are really dispersed after the propagation through the bar model.

The model is described by the VHDL language and therefore, it can be implemented both in the VHDL simulator and in FPGA. The model described in the style for the synthesis and configured in Xilinx Spartan-6 FPGA contains 2680 logic cells, 60 DSP48 blocks, 16 BlockRAMs, and provides the sampling frequency $F_s <$ 100 MHz. This provides the modeling in real time.

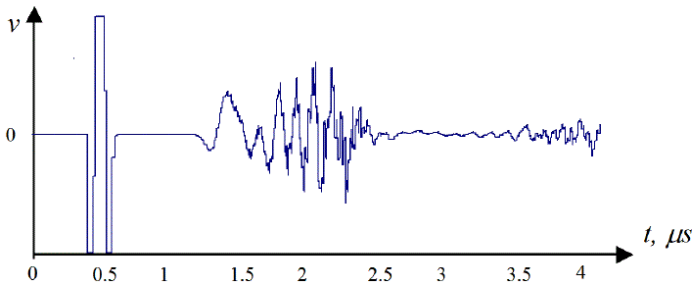

Figure 3. Time diagram of the wave signal in the solid bar model

Conclusion. A modified DWG method for modeling the solids is proposed which takes into account the dispersed wave propagation. The method is based on the digital wave guides which delay is depending on the wave length. An example of the solid bar model described in VHDL shows the effectiveness of this model.

References. **1.** D. R. Bergman "Computational Acoustics: Theory and Implementation". J.Wiley and Sons, Ltd. 2018. 296 p. **2.** J. O. Smith III "Physical Modeling Using Digital Waveguides". Computer Music Journal V.16. 1993. №4.pp.74-91. **3.** M. Karjalainen, and C. Erkut "Digital Waveguides versus Finite Difference Structures: Equivalence and Mixed Modeling". EURASIP Journal on Applied Signal Processing 2004. №7. pp. 978–989. **4.** A. Fettweis "Wave digital filters: Theory and practice". Proc. IEEE, V74, 1986. №2. pp. 270 – 327. **5.** H. Kolsky "Stress Waves in Solids". Dover Publications Inc. 2012. **6.** P. A. Regalia, S. K. Mitra, and P. P. Vaidyanathan "The Digital All-Pass Filter: A Versatile Signal Processing Building Block". Proc. IEEE. V.76. 1988. — 76, №1. pp. 19—37. **7.** А.М Сергиенко, Т.М Лесик "Перестраиваемые цифровые фильтры на ПЛИС". Электронное моделирование, Т.32. 2010, №6. С. 47-56.(A. M. Sergiyenko, and T. M. Lesyk "Tunable digital filters in FPGA"Electronic Modeling. V.32. 2010.№6. pp. 47-56).

УДК 004.942

Анатолій Сергієнко, Мустафа Рекар Касім

АПАРАТНЕ МОДЕЛЮВАННЯ ПОШИРЕННЯ УЛЬТРАЗВУКОВИХ ХВИЛЬ У ТВЕРДОМУ ТІЛІ

Anatoliy Sergiyenko, Mustafa Rekar Quasim

MODELING THE ULTRASOUND WAVE PROPAGATION IN THE SOLID BODY BY THE HARDWARE

Розглянуто удосконалений хвилевий алгоритм моделювання поширення ультразвуку, який полягає у представленні середовища у вигляді системи хвилевих фільтрів та відрізняється тим, що завдяки реалізації багатоканальних фільтрів з програмованою затримкою, зменшується похибка моделювання дисперсійного поширення звуку. Алгоритм при реалізації у ПЛІС дає змогу виконувати моделювання у реальному часі.

Ключові слова: ПЛІС, дисперсія звуку, хвилевий фільтр.
Рис.: 3. Бібл.: 8.

The improved ultrasound propagation wave modeling algorithm is proposed, which consists in representing the medium in the form of a system of the wave digital filters. The algorithm differs in that due to the implementation of the multichannel filters with the programmable delays, the simulation error of the sound propagation dispersion is decreased. The algorithm implementing in FPGA provides the modeling in real time.

Key words: FPGA, sound dispersion, wave digital filter.
Fig.: 3. Bibl.: 8.

Вступ. Акустичні процеси в твердих тілах зазвичай моделюються в комп'ютерах за допомогою методу різницевих рівнянь, який вимагає великих обчислювальних ресурсів [1]. Для мінімізації цих ресурсів часто використовують метод цифрових хвилеводів, або метод DWG (digital waveguide), який забезпечує високошвидкісні обчислення [2]. Багато успішних прикладів впровадження методу DWG показано в моделюванні та звуковому синтезі струнних, духових та віртуальних музичних інструментів, а також в вокодерах [2, 3, 4]. Але ця модель не враховує розсіювання (дисперсію) хвиль у поширенні звуку.

У даній роботі пропонується модифікація методу DWG, яка забезпечує моделювання звукової дисперсії в твердій смужці.

Основи моделі DWG. DWG – це цифрова модель розповсюдження хвиль в ідеальному хвилеводі. Вона базується на принципах, описаних у роботі [5], але адаптованих до моделювання звукових хвиль [2,4]. При цьому у i-й точці хвилеводу розглядаються прямі $f_i = Rv_f$ і зворотні хвилі $b_i = -Rv_b$, де f_i та b_i – це тиск прямих і зворотних (віддзеркалених) хвиль, відповідно R – хвилевий імпеданс, v_f, v_b – швидкості частинок речовини. Для жорсткого циліндра з площею перетину A цей імпеданс дорівнює $R = \rho c_o A$, де ρ – густина матералу, c_o – швидкість поздовжних хвиль. Реальне значення тиску дорівнює сумі тисків прямої та зворотної хвиль: $u = f_i + b_i$.

У цифровій моделі DWG сигнали квантуються з частотою дискретизації F_s, яка, щонайменше, в два рази перевищує максимальну частоту хвиль, які досліджуються. Тоді i-й сегмент хвилеводу довжиною L виглядає як дві лінії затримки на $n = L/(c_o F_s)$ тактів сигналу дискретизації.

Кілька однорідних хвилеводів з'єднануються між собою в одній точці за допомогою вузла адаптера, який також називають вузлом розсіювання. Функціонування адаптера повинне підлягати закону Кірхгофа. Для випадку з'єднання двох хвилеводів це означає виконання рівнянь:

$$v_{b1} = rv_{f1} + (1 - r)\, v_{f2}; \qquad (1)$$
$$v_{b2} = (1 + r)\, v_{f1} - rv_{f2};$$

д $r = (R_2 - R_1)/(R_2 + R_1)$ – коефіцієнт віддзеркалювання. Так само, коли підключено три хвилеводи, то у вузлі адаптера розраховуються наступні формули:

$$v_{b1} = (2g_1 - 1)\, v_{f1} + 2g_2\, v_{f2} + 2g_3\, v_{f3};$$
$$v_{b2} = 2g_1\, v_{f1} + (2g_2 - 1)\, v_{f2} + 2g_3\, v_{f3}; \qquad (2)$$
$$v_{b3} = 2g_1\, v_{f1} + 2g_2\, v_{f2} + (2g_3 - 1)\, v_{f3};$$

де v_{f1}, v_{f2}, v_{f3} – хвилі, що входять у вузол, v_{b1}, v_{b2}, v_{b3}, – хвилі, що виходять з вузла, g_1, g_2, g_3 – питомі імпеданси хвилеводів, такі що $g_1 + g_2 + g_3 = 1$, та $g_i = R_i/(R_1 + R_2 + R_3)$, $i = 1, 2, 3$.

Удосконалена модель DWG. Згідно з теорією поширення акустичних хвиль у твердих тілах, які мають обмежені розміри, наприклад, у стрижнях, розглядають основні – повздовжні хвилі з фазовою швидкістю c_p та вторинні – поперечні хвилі зі швидкістю c_s, причому $c_s < c_p$ [6]. При взаємодії цих хвиль вони можуть перетворюватись одна в іншу. Це можна промоделювати при з'єднанні хвилеводів з такими хвилями у вузлі адаптера.

Крім того, повздовжні хвилі мають дисперсію, тобто, їх швидкість c_p залежить від довжини хвилі λ. Наприклад, швидкість повздовжних хвиль у циліндрі оцінюється формулою [6]:

$$c_p = c_o(1 - \nu^2 p^2 a^2 / \lambda^2) = f(\lambda), \qquad (3)$$

де ν – коефіцієнт Пуассона, що дорівнює 0,29 для сталі, a – радіус циліндра. Тому модель DWG повинна бути зкоректована відповідно до цієї формули.

Модифіковану модель DWG деякого стрижня або циліндра можна представити структурою, яка показана на рис.1. Ця модель складається з лівого (LA) та правого (RA) адаптерів, хвилепроводу P продольних хвиль та хвилеводу S поперечних хвиль. У кожному з трьохпортових адаптерів обчислюються вирази (2). Їхні порти, які означають торці стрижня, підключені до мереж, які реалізують віддзеркалення хвиль з коефіцієнтом придушення s_L та s_R згідно з рівняннями (1). Сигнал збудження v_{in} подається у модель через суматор.

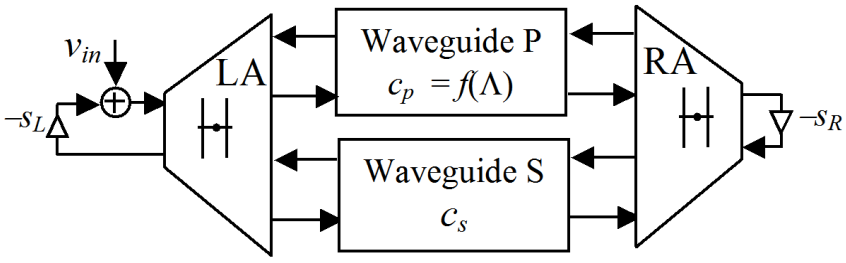

Рис. 1. Структура моделі стрижня

Хвилевід S виконує затримку на задану кількість тактів частоти дискретизації та певне ослаблення хвилевого сигналу. Хвилевід P складається з кількох каналів, кожен з яких має окремі діапазон частот пропускання, затримку і згасання в залежності від відповідної довжини хвилі λ згідно з (3).

Отже, модифікована модель DWG здатна приймати до уваги дисперсність поширення звуку у твердому тілі, тобто, залежність фазової швидкості звуку від його довжини хвилі або частоти.

Експериментальні результати. Була розроблена програмна модель стрижня зі структурою як на рис. 1. В ній до восьми каналів хвилеводу P реалізуються на основі смугових фільтрів на базі фазового хвилевого фільтра, який описано у [7]. Кожен канал

хвилеводу має динамічно настроюваний смуговий фільтр, який реалізовано так, як показано в [8]. Сумарна амплітудно-частотна характеристика хвилеводу S показана на рис.2. Тут частота *f* вимірюється в частках частоти дискретизації *Fs*. На рис.2 також показані частотні характеристики двох сусідніх каналів хвилеводу.

Модель збуджувалась вузьким ультразвуковим імпульсом, а інтенсивність звуку *v* вимірювалась на виході вузла суматора (див. рис.1). Часова діаграма, яка представляє хвильовий сигнал, показана на рис.3. Ця діаграма показує, що хвилі дійсно зазанають дисперсії під час розповсюдження через модель стрижня.

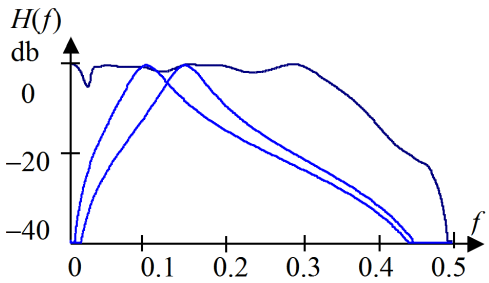

Рис. 2. Сумарна амплітудо-частотна характеристика хвилеводу P

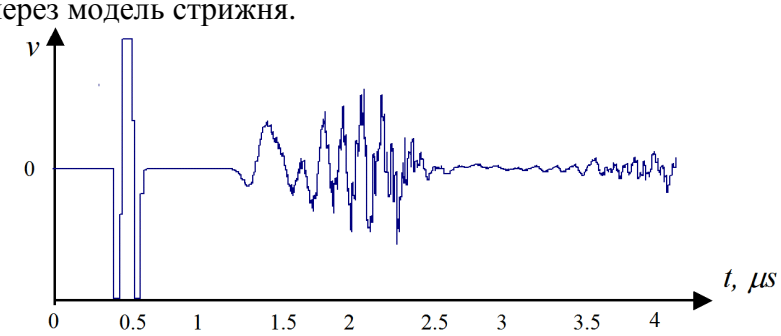

Рис. 3. Часова діаграма поширення ультразвукового імпульсу у стрижні

о VHDL, і тому вона може бути реалізована як у симуляторі VHDL, так і в ПЛІС. Модель, описана стилем для синтезу і при конфігуруванні у ПЛІС Xilinx Spartan-6, містить 2680 логічних елементів, 60 блоків множення DSP48, та 16 блоків пам'яті BlockRAM. При цьому забезпечується частоту дискретизації сигналу F_s < 100 МГц. Така частота забезпечує моделювання в режимі реального часу.

Висновки. Запропоновано модифікований метод DWG для моделювання твердого тіла, який враховує дисперсність поширення хвиль. Метод заснований на моделі цифрового хвилеводу, затримка якого залежить від довжини хвилі. Приклад моделі стрижня, описаної на VHDL, показує ефективність цієї моделі.

Посилання

1. Самарский А.А. Математическое моделирование. Идеи. Методы. Примеры / Михайлов А.П. / М.: Физматлит, 2001. -320 с.
2. Smith III J. O. Physical Modeling Using Digital Waveguides // Computer Music Journal, V.16. 1993. №4. P.74-91.
3. Karjalainen M. Digital Waveguides versus Finite Difference Structures: Equivalence and Mixed Modeling / C. Erkut // EURASIP Journal on Applied Signal Processing 2004. №7. P. 978–989.
4. Маркел Дж. Д. Линейное предсказание речи. – М.: Связь. 1980. 308 с.
5. Fettweis A. Wave digital filters: Theory and practice // Proc. IEEE, V74, 1986. №2. P. 270 – 327.
6. Kolsky H. Stress Waves in Solids. Dover Publications Inc. 2012.
7. Regalia P. A., The Digital All-Pass Filter: A Versatile Signal Processing Building Block / S. K. Mitra, P. P. Vaidyanathan //Proc. IEEE. V.76. 1988. №1. P. 19-37.
8. Сергиенко А. М, Перестраиваемые цифровые фильтры на ПЛИС / Т. М. Лесик // Электронное моделирование, Т.32. 2010, №6. С. 47-56.