Experimental ultrasound system for real-time synthetic imaging

Jensen, Jørgen Arendt; Holm, Ole; Jensen, Lars Joost; Bendsen, Henrik; Pedersen, Henrik Møller; Salomonsen, Kent; Hansen, Johnny; Nikolov, Svetoslav

Published in:
1999 IEEE Ultrasonics Symposium Proceedings

Link to article, DOI:
10.1109/ULTSYM.1999.849300

Publication date:
1999

Document Version
Publisher's PDF, also known as Version of record

Experimental ultrasound system for real-time synthetic imaging

Jørgen Arendt Jensen (1), Ole Holm (2), Lars Joost Jensen (2), Henrik Bendsen (2),
Henrik Møller Pedersen (1,2), Kent Salomonsen (2), Johnny Hansen (2) and Svetoslav Nikolov (1)

(1) Center for Fast Ultrasound Imaging, Department of Information Technology, Build. 344,
Technical University of Denmark, DK-2800 Lyngby, Denmark
(2) I/O Consulting A/S, Lautrupvang 1B, Postboks 199, DK-2750 Ballerup, Denmark

Abstract

Digital signal processing is being employed more and more in modern ultrasound scanners. This has made it possible to do dynamic receive focusing for each sample and implement other advanced imaging methods. The processing, however, has to be very fast and cost-effective at the same time. Dedicated chips are used in order to do real time processing. This often makes it difficult to implement radically different imaging strategies on one platform and makes the scanners less accessible for research purposes. Here flexibility is the prime concern, and the storage of data from all transducer elements over 5 to 10 seconds is needed to perform clinical evaluation of synthetic and 3D imaging. This paper describes a real-time system specifically designed for research purposes.

The purpose of the system is to make it possible to acquire multi-channel data in real-time from clinical multi-element ultrasound transducers, and to enable real-time or near real-time processing of the acquired data. The system will be capable of performing the processing for the currently available imaging methods, and will make it possible to perform initial trials in a clinical environment with new imaging modalities for synthetic aperture imaging, 2D and 3D B-mode and velocity imaging.

The system can be used with 128 element transducers and can excite 128 channels and receive and sample data from 64 channels simultaneously at 40 MHz with 12 bits precision. Data can be processed in real time using the system's 80 signal processing units or it can be stored directly in RAM. The system has 24 GBytes RAM and can thus store 8 seconds of multi-channel data. It is fully software programmable and its signal processing units can also be reconfigured under software control. The control of the system is done over an Ethernet using C and Matlab. Programs for doing *e.g.* B-mode imaging can directly be written in Matlab and executed on the system over the net from any workstation running Matlab. The overall system concept is presented and an example of a 20 lines script for doing phased array B-mode imaging is presented.

1 Introduction

New imaging techniques based on synthetic imaging are currently being suggested and investigated [1, 2]. The methods can potentially increase both resolution and frame rate, since the images are reconstructed from RF data from the individual transducer elements. Hereby a perfectly focused image in both transmit and receive can be made. Research in real time 3D imaging is also underway [3, 4]. The purpose is to make systems that in real time can display a pyramidal volume of the heart, where different slices hereafter can be visualized. These images have a poor signal-to-noise ratio, and several groups are working on employing coded signals to enhance the signal-to-noise ratio [5].

All of the above techniques require digital signal processing on the signals from the individual transducer elements, and in some instances it is also necessary to send out coded signals on the individual elements. For research purposes this can be difficult to attain with commercial scanners, since they are often highly integrated and it is difficult to access individual signals. Programming commercial scanners for new imaging techniques is often also either cumbersome or impossible. It is, thus, beneficial to develop a dedicated research system, that can acquire, store, process, and display ultrasound images from multi-element transducers.

2 System specification

The purpose of the system is to make possible the acquisition of multi-channel data in real-time from clinical multi-element ultrasound transducers, and to enable real-time or near real-time processing of the acquired data. The system will be capable of performing the processing for all currently available imaging methods, and will make it possible to carry out initial trials with new imaging modalities for synthetic aperture imaging, 3D imaging, and 2D and 3D velocity estimation. It is capable of working in a clinical environment to evaluate the performance of various algorithms. The system is specifically intended for research purposes, and is not intended for

0-7803-5722-1/99/$10.00

commercial use.

The function of the system is defined by the different imaging methods for which it can be used. Each of the imaging types will be described and the consequence for the system then given.

Linear array imaging: A linear array image is generated by a multi-element transducer with 128 to 256 elements. The beam is moved by selecting *e.g.* 64 adjacent elements and emitting a focused beam from these. The focusing in receive is also done by a number of elements, and multiple foci are used. Apodization in both transmit and receive are often applied. The focusing delay in both transmit and receive are both less than 40 μs. The number of active elements is usually 32 to 64. The transducer frequency is from 2 MHz to 10 MHz. Imaging is done down to a depth of 30 cm.

The demands on the system is, thus, for 64 channels simultaneous sampling at 40 MHz. The maximum delay in both transmit and receive is 40 μs. The maximum time to sample one line is $2 \times 0.3/1540 + 40 \cdot 10^{-6} = 430\,\mu$s corresponding to 17,200 samples at 40 MHz.

Phased array imaging: The beam is here electronically swept over the imaging area by using a 128 to 256 element array. All the elements are used at the same time, and focusing time delays used are less than 50 μs. The transducer frequency is from 2 MHz to 10 MHz. Investigations are done to a depth of 20 cm.

The demands on the system is, thus, for 128 channels sampling at 40 MHz. The demands on delay, sampling time and storage are the same as for linear array imaging.

Flow estimation, spectrum: Beamforming is done in one direction with either a linear or phased array. The flow signal from blood has 40 dB less power than that from stationary tissue. The dynamic range of the flow signal is 30 dB. The effective number of bits must be 12 or more, when the signals from all channels have been combined. The pulse emitted can have from 4 to 16 periods of the center frequency of the transducer or a coded signal can be employed.

Flow imaging: Imaging is done by pulsing repeatedly in one direction and then change the direction to generate an image. An image can therefore be assembled from up to a 1000 pulse emissions.

Three-dimensional imaging: A matrix element transducer is used with up to 40 $\times$ 40 elements. Only some of the elements are used for transmit and receive. The area of the elements is small and pulsing should be done with 100 to 300 volts. Coded signals should be used. Coded pulses with up to 64 cycle periods must be possible with a high amplitude accuracy. This corresponds to emission over a period of 32 μs with a sampling frequency of 40 MHz and an accuracy of 12 bits.

Phasing is done with a delay up to 50 μs, and parallel lines are generated by using parallel beam formers and reusing data from one pulse emission. The system must be capable of reading the data sampled from one elements a number of times, and use different phasing schemes for each cycle through the data.

Synthetic aperture imaging: A standard linear or phased array multi-element transducer is used. Pulsing is done on a few elements and the received response is acquired for all elements. The image is then reconstructed from only a small number of pulse emissions by using the data from all the elements.

This type of imaging needs large amounts of storage and the ability to reuse the data for the different imaging directions. This should be solved by having a multiprocessor system connected to the sampling system for storage and image reconstruction.

It must be possible to acquire several seconds of data. Assuming sampling in 80 % of the time at 40 MHz, and 8 seconds of sampling gives a storage need of 256 Mbytes per channel.

3 System realization

The multi-channel sampling and processing system consists of four distinct modules: The transmit unit, the receive/transmit (Rx/Tx) amplifiers, receiver sampling unit, and the sync/master unit. The main blocks are depicted on the drawing in Fig. 1. The connection to the transducer is through a 128 wire coaxial cable through the Rx/Tx amplifiers. The transmitter sends the signals through the transmit amplifier, and the receiver unit samples the amplified and buffered signals from the Rx/Tx amplifiers. The sync/master unit holds a crystal oscillator and controls the central timing of the scanning process. The overall operation of the system is controlled through a number of single board PCs in the individual units interconnected through a standard 100 Mbit Ethernet. The waveforms and phasing data are transmitted from the controlling PC to the transmitters and receiver boards. The data from the sampling is processed by FPGAs (field programmable gate arrays), that can be configured for specific signal processing tasks over the net. One Focus FPGA is used for each element and a Sum FPGA is placed for each eight elements. The processed and summed signal can then be routed from Sum FPGA to Sum FPGA. The resulting signal is read by one or more signal processors, that can be connected through serial interfaces capable of transmitting 40 Mbytes per second. Each processor has 6 such links. Data and programs are transferred through these links. The beamformed

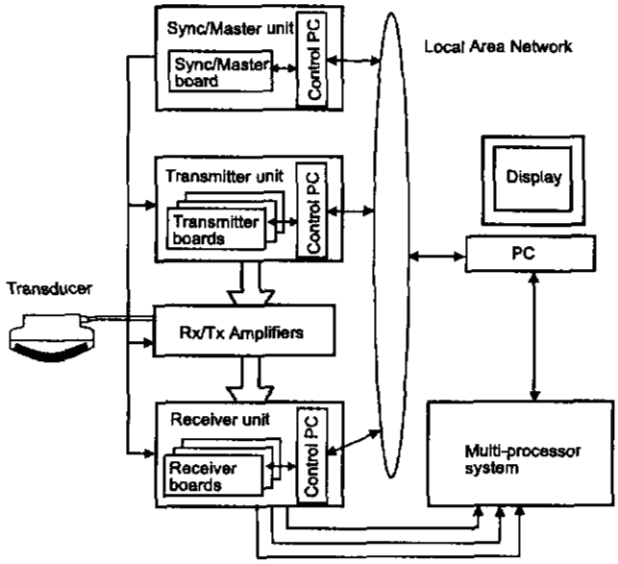

Figure 1: Overall diagram of system.

and envelope detected signal is send via the link channels to the PC for display.

The following paragraphs detail the overall design of the individual boards.

3.1 Transmitter

The transmitter is capable of generating an arbitrary transmitted pulse with a bandwidth below 20 MHz. The transmitter consists of 8 transmitter boards each equipped with 16 channels. In total the transmitter controls 128 channels.

A transmitter board consists of two control FPGA's, 16 pulse RAM's, two delay RAM's and sixteen 12 bit digital to analog converters (DAC).

Each of the 16 channels has a pulse RAM memory implemented as a 128 k $\times$ 12 bit SRAM, which allows the user to store for instance 32 different pulse emissions of 100 μs duration.

The delay RAM holds the start address of the pulse emission in the pulse RAM and the corresponding delay for each line. The delay RAM is implemented as 32 k $\times$ 32 bit SRAM. At the start of each line the pulse emission is delayed according to the delay value for each channel.

3.2 Receiver

The Receiver board is illustrated in Fig. 2. The board samples and processes 8 analog signals selected from 16 inputs.

The receiver, transmitter and sync/master boards are accessible from a general purpose LINUX based compact PCI single board PC, which controls the different boards.

A 12 bit analog to digital converter (ADC) samples one input channel and the data is temporarily stored in a SRAM buffer. The data in the SRAM buffers are processed by the Focus FPGA using the Focus RAM. The Sum FPGA processes the focused data from the 8 Focus FPGA's and transfers it to the Analog Devices signal processor (ADSP) or stores it in the storage RAM. The functionality of the individual blocks of the receiver board is explained in further detail below.

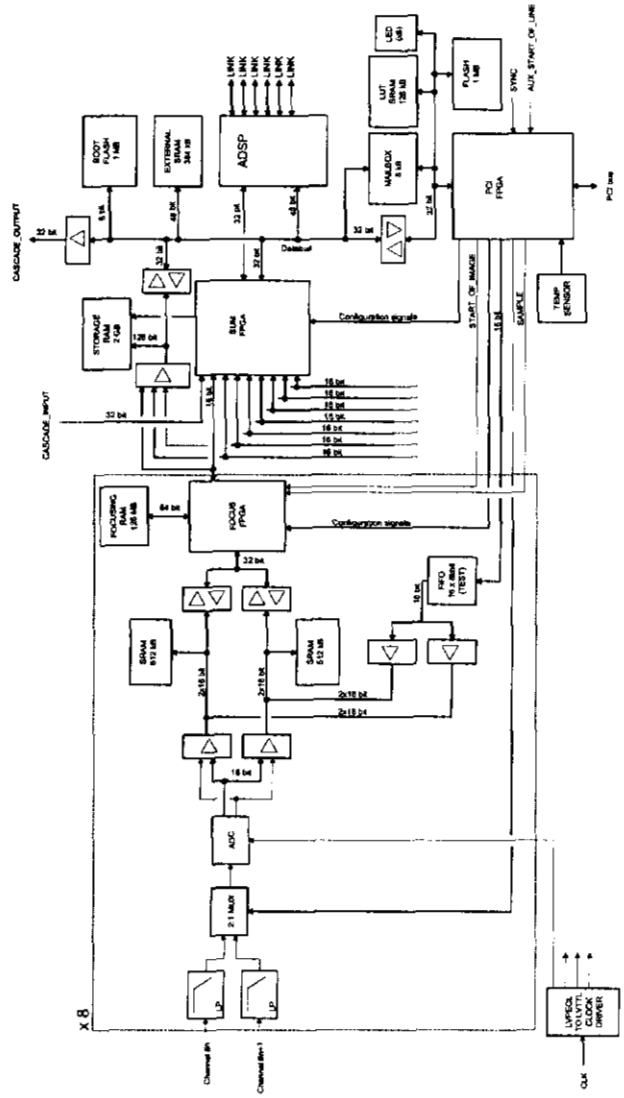

Figure 2: Main diagram of Receiver board.

3.2.1 Focus FPGA

The Focus FPGA controls the initial storing and processing of the sampled data. The Focus FPGA fetches the sampled data from the SRAM and the corresponding focusing parameters from the Focusing RAM and processes the data before transferring the result to the Sum FPGA.

Two independent memory burst SRAM banks are used to bank switch between the sampled data and processed data. While the sampled data is being written to one of the two banks, the other bank can be read by the Focus FPGA. Each SRAM is implemented as 256 kbytes, which is equivalent to a line length of 3.3 ms sampled at 40 MHz.

The basic focusing algorithm uses a combination of coarse and fine delays. The coarse delay is in steps of the 25 ns sampling interval, and it is implemented as a general table look up address generator. For each sample a 16 bit address index is read from the SDRAM. In this way a random sorting algorithm can be implemented. The fine delay is implemented as a linear interpolation with two signed 10 bit apodization coefficients, which are read from the focusing RAM for each sample.

The Focus FPGA is implemented using a XILINX device from the Virtex family: XCV300 in a 352 pin BGA package speed grade -4. The simple B-mode beamformer described above uses less than 10% of the logical resources of the chip. This makes it possible to investigate hardware implementations of more advanced beamformers including pulse compression, synthetic aperture, and parallel beamforming.

3.2.2 Sum FPGA

The Sum FPGA is used to perform digital signal processing on the 8 channels. The most basic operation is to sum the 8 focused channels. Further, it is used as the gateway between the eight independent sampling channels and the ADSP. The Sum FPGA controls the 2 Gbyte storage SDRAM.

When the focusing is done in the Focus FPGA, the 8 channels are added to the accumulated sum that is passed to the next Receiver board using a high speed cascade bus connecting the Sum FPGA's directly with each other. The last Sum FPGA in the chain uses the ADSP link ports to transfer the final result to the multiprocessor cluster.

The Sum FPGA is implemented using a XILINX device from the Virtex family: XCV1000 in a 560 pin BGA package speed grade -4. The simple design described above uses less than 5% of the logical resources of the chip.

3.3 Sync/master unit

The Sync/master unit controls the timing of the system. A highly stable oven controlled crystal oscillator generates the 40 MHz clock frequency. The clock jitter is below 5 ps. The clock is distributed using coax cables and emitter coupled logic (ECL) in order to minimize jitter. The timing source also transmits a synchronization signal. The receiver and transmitter uses the SYNC signal to start and stop the sampling cycles. An image consists of a number of lines, each with a transmission of a pulse and reception of the echoes. The transmitter and receiver generates an internal LINE signal from the SYNC signal to control the sampling process for the received signal.

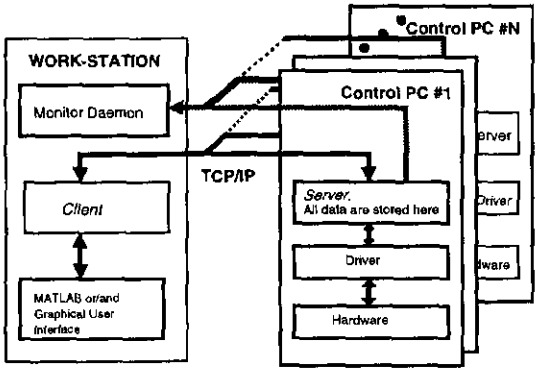

Figure 3: Client-server model of software.

4 Programming of the system

From the software point of view, the system consists of several computers that are directly connected to a number of transmitter and receiver boards. The computers are linked by a LAN, and uses Linux as operating system and TCP/IP as the underlying communication protocol.

The software was designed to meet the following requirements:

- Flexibility and ease of use. It is of prime importance that new imaging methods can be quickly implemented with a minimal amount of programming also for new users.
- Data encapsulation. All data for the imaging algorithm are stored in the boards of the scanner.
- Distributed computations. The computers work independently one from another, and each calculates only those imaging parameters concerning the boards plugged in it.
- Portability. The software is written in ANSI C and is platform independent.

The client/server communication model was adopted for the software. The computers controlling the boards run a *server* program. The server waits for requests coming from the LAN and processes them. The requests can be sent by any *client* program running on a computer connected to the LAN using the TCP/IP communication protocol.

Figure 3 shows the client-server model of the software. At start-up the server detects which boards are in the PCI enclosures. The computers can handle any combination of transmitter and receiver boards, plugged into the same PCI backplane. The server is in idle mode until an event occurs. In

the case of a hardware malfunction, the server sends an emergency message to a program, called *monitor daemon*. Another event is a request by the *client*. The request can be for transferring parameters to the boards, for performing some calculations, or for retrieving data from the boards to the computer with the user.

The interface to the client program is implemented as a MATLAB tool-box. The function calls are implemented to be as close to the functions in the simulation program Field II [6] as possible. Algorithms created using Field II can thereby easily be tested on the scanner with only minor changes in the Matlab program. An example for setting the system to perform phased array B-mode imaging is shown below:

```
% Auto-detect and initialize the system
sys_init('auto');
% Set the pulse repetition frequency
sys_set_fprf(fprf);
% Set the sampling range gate in receive
sys_set_sampling_interval(start_depth, end_depth);
% Set the number of scan-lines per frame
sys_set_no_lines(no_lines);
% Define the transducer. Necessary for the delay calculations
tr_linear_array(no_elements, width, height, kerf);
% Do for all lines in the image:
for line_no = 1:no_lines
  % Set the pulse and the apodization for the current line
  xmt_excitation(waveform(line_no));
  xmt_apodization(line_no,xmt_apo(line_no,:));
  rcv_apodization(line_no,times, rcv_apo(line_no,:,:));
  % Set the focus, defined in 3D coordinates
  xmt_focus(line_no,focus(line_no));
  rcv_dynamic_focus(line_no, theta(line_no), fi(line_no));
end

% Set the time-gain compensation curve
tmg_tgc(tgc_vector);
% Start the continuous imaging process
tmg_start
```

In order to make the system perform linear array imaging only one line needs to be added, which changes the origin of the individual scan-lines.

5 Conclusion

A system for doing research into new imaging methods in medical ultrasound has been described. The system is capable of emitting complex, arbitrary waveforms, and can sample and store data for all transducer channels in real time. Sufficient storage is included in the system for 8 seconds of data at a sampling frequency of 40 MHz for a 64 element transducer. The system is easily programmable through Matlab and a network interface. It can perform real time processing and image display for all commercially available ultrasound systems and can function in a clinical environment.

Acknowledgment

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation, by B-K Medical A/S, and by grant EF-782 from the Danish Academy of Technical Sciences.

References

[1] J. T. Ylitalo and H. Ermert. Ultrasound synthetic aperture imaging: monostatic approach. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 41:333–339, 1994.

[2] C. R. Hazard and G. R. Lockwood. Theoretical assessment of a synthetic aperture beamformer for real-time 3-D imaging. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 46:972–980, 1999.

[3] S. W. Smith, H. G. Pavy, and O. T. von Ramm. High-speed ultrasound volumetric imaging system – Part I: Transducer design and beam steering. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 38:100–108, 1991.

[4] G. R. Lockwood, J. R. Talman, and S. S. Brunke. Real-time 3-D ultrasound imaging using sparse synthetic aperture beamforming. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 45:980–987, 1998.

[5] P. Li, E. Ebbini, and M. O'Donnell. A new filter design technique for coded excitation systems. *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 39:693–699, 1992.

[6] J. A. Jensen. Field: A program for simulating ultrasound systems. *Med. Biol. Eng. Comp.*, 10th Nordic-Baltic Conference on Biomedical Imaging, Vol. 4, Supplement 1, Part 1:351–353, 1996b.