

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Three-Dimensional Ultrasound Imaging Utilizing Hardware Accelerator Based on FPGA

Keiichi Satoh¹, Jubee Tada², Gensuke Goto², Toshio Koga²,
Kazuhiro Kondo² and Yasutaka Tamura²

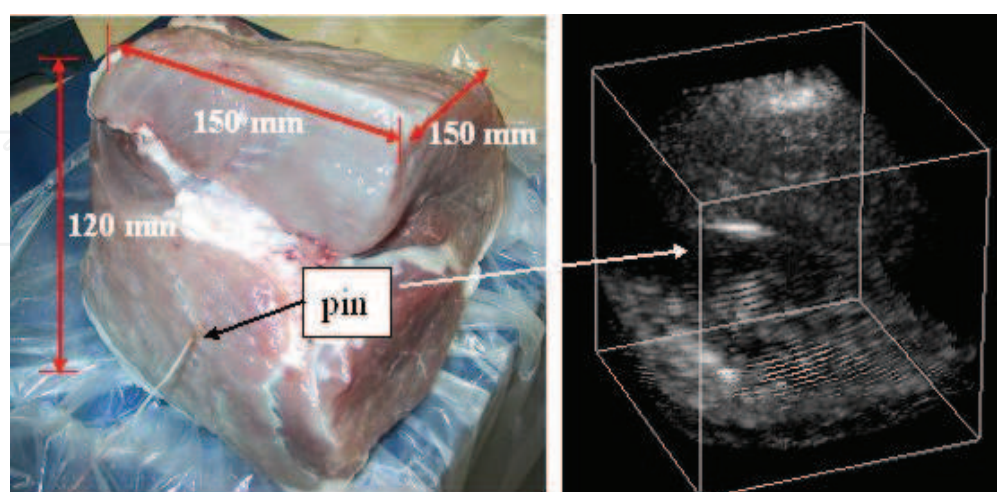
¹*Yamagata university of Graduate School (current affiliation is Fujitsu. Ltd.),*

²*Yamagata University of Graduate School
Japan*

1. Introduction

We have developed a 3D ultrasound imaging system involving computations for use in medical diagnostic applications [1,2]. This system enables us to observe 3D images of moving objects for each transmission. Therefore, we can acquire a 3D image sequence at a high frame rate. Fig. 1 shows a 3D image reconstructed by our 3D ultrasound imaging system. In the pork block, a needle is inserted as a marker.

In the present system, a 3D image reconstruction algorithm is implemented by using the C language software (SW). When processing is performed by a personal computer (PC) with Pentium 4 2.53 [GHz] CPU and DDR512 [MB] memory, the latency for generating a 3D image is approximately 40 s. As apparent from the above latency value, it would be difficult to realize high-speed image reconstruction by employing a SW implementation approach. Reduction in the processing time is one of the most important issues in practical applications. As a solution to this issue, we have investigated the HW implementation of the algorithm.



(a) Pork block

(b) Reconstructed 3D image

(Image size: a cube of dimensions 70 × 70 × 70 mm)

Fig. 1. Three-dimensional image reconstructed by our system.

Presently, FPGA is mainly utilized to implement algorithms in various applications such as radio telescopes [3], streaming computations [4], and neurocomputing [5]. The reason for using FPGA is that it enables us to design flexible HW at a low cost by using a reconfiguration function. Thus, we decided to employ the HW implementation approach and used FPGA as the target device.

In this paper, we first present the 3D image reconstruction algorithm and then describe the processing system and HW architecture. Next, we search critical path delay in the HW and the modify the path to accelerate raising the maximum frequency. Finally, we evaluate the performance (latency required for a 3D image output) and the scale of the synthesized HW.

2. Principle of ultrasound transmission and reception

Ultrasound waves are simultaneously transmitted from N_T (must be in powers of 2) transmitters and the reflected echo waveforms are detected by N_R receivers from two-dimensional (2D) transducer arrays. We use sinusoidal waves of frequency f_0 modulated by a system of Walsh functions synchronized by a clock signal.

The period of the clock signal Δt is equal to an integral multiple of the sine wave period $1/f_0$. The transmitting and receiving processes are repeated at a constant period. The transmission codes corresponding to the transmitters are changed in every transmission and reception cycle. At the p -th transmission and reception cycle, the transmission signal corresponding to the i -th ($i = 0, 1, 2, \dots, N_T-1$) transmitter at x_{Tj} and the waveform detected by the j -th ($j = 0, 1, 2, \dots, N_R-1$) receiver at x_{Ri} are denoted by $u_i^{(p)}(t)$ and $r_i^{(p)}(t)$, respectively. In order to simplify the discussion, the origins ($t = 0$) of these functions are fixed at the starting positions of each transmitting pulse.

The pulse transmitted from the i -th transmitter in the p -th cycle is given as

$$u_i^{(p)}(t) = \begin{cases} \sum_{k=0}^{N_T-1} w_{(i \oplus p)k} \cdot f(t - k \cdot \Delta t) & 0 \leq t \leq N_T \Delta t \\ 0 & \dots t < 0, t > N_T \Delta t \end{cases} \quad (1)$$

where

$$f(t) = \begin{cases} \sin(2\pi f_0 t) & 0 \leq t \leq \Delta t \\ 0 & t < 0, t > \Delta t \end{cases} \quad (2)$$

is a single sinusoidal pulse of frequency f_0 , w_{nm} denotes the (n, m) component of the $N_T \times N_T$ Hadamard matrix (to simplify the equation, the column and row numbers are indexed from 0 to $N-1$), and \oplus denotes the dyadic sum, i.e., the modulo 2 addition for every corresponding bit of binary numbers. Fig. 2 shows a schematic diagram of ultrasound transmission and reception in which the coded *wavefront* generated by Walsh functions is employed.

3. Principle of image reconstruction

The equations given below represent a mathematical model that can be used for image reconstruction. A sequence of complex pixel values, $s^{(p)}(\mathbf{x})$, corresponding to the position vector \mathbf{x} is given by

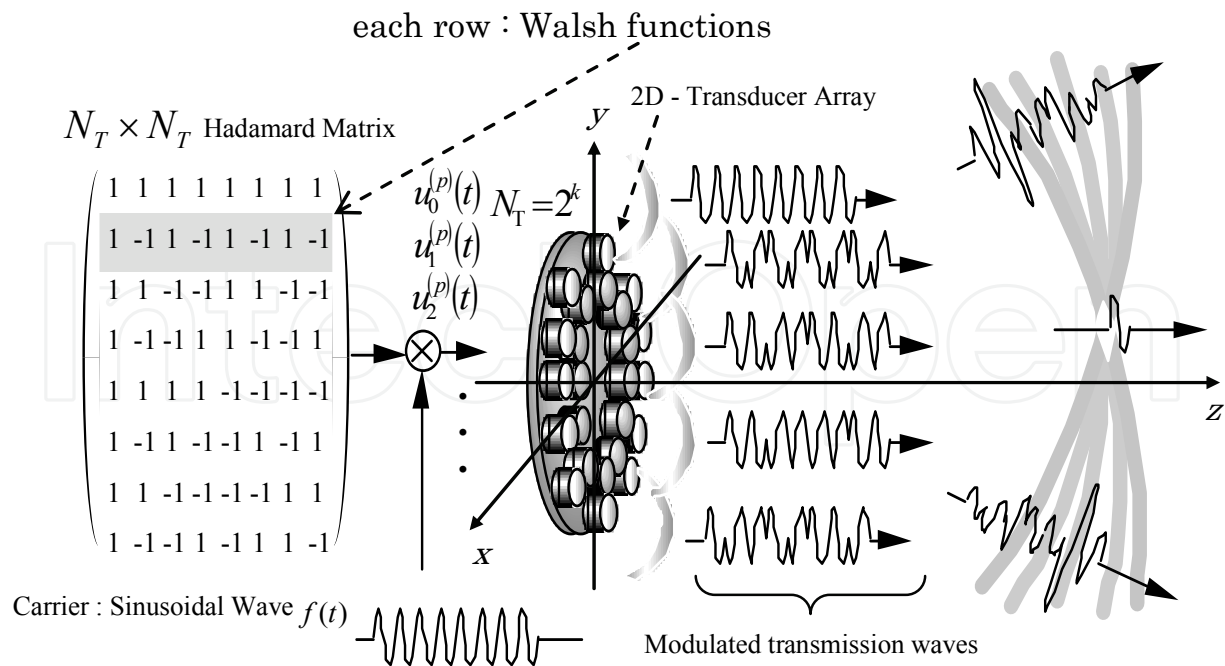


Fig. 2. Schematic drawing of the coded *wavefront* generated by Walsh functions.

$$\begin{aligned} s^{(p)}(\mathbf{x}) &= \int \sum_{j=0}^{N_R-1} r_j^{(p)}(t + \tau_j) \sum_{i=0}^{N_T-1} u_i^{(p)}(t - \tau_i)^* dt \\ &= \int \sum_{j=0}^{N_R-1} r_j^{(p)}\left(t + \frac{|\mathbf{x} - \mathbf{x}_{Rj}|}{c}\right) \sum_{i=0}^{N_T-1} u_i^{(p)}\left(t - \frac{|\mathbf{x} - \mathbf{x}_{Ti}|}{c}\right)^* dt, \end{aligned} \tag{3}$$

where τ_i and τ_j denote the sound propagation delays between the position \mathbf{x} and the transducers, the symbol $*$ indicates complex conjugate, \mathbf{x}_i and \mathbf{x}_j represent the position vectors of the i -th transmitter and j -th receiver, respectively, and c is the speed of sound. Fig. 3 shows the concept of image reconstruction.

The delay between positions \mathbf{x} and \mathbf{x}_i is given by

$$\tau_i = \frac{|\mathbf{x} - \mathbf{x}_i|}{c}, \tag{4}$$

and is computed according to the following approximation:

$$\tau = \frac{\sqrt{(x - x_i)^2 + (y - y_i)^2 + z^2}}{c} \tag{5}$$

$$\approx \frac{R - r_i \cos(\varphi - \varphi_i) \sin \theta + r_i^2 / (2R)}{c}, \tag{6}$$

$$\text{where } R = |\mathbf{x}| = \sqrt{x^2 + y^2 + z^2}, \tag{7}$$

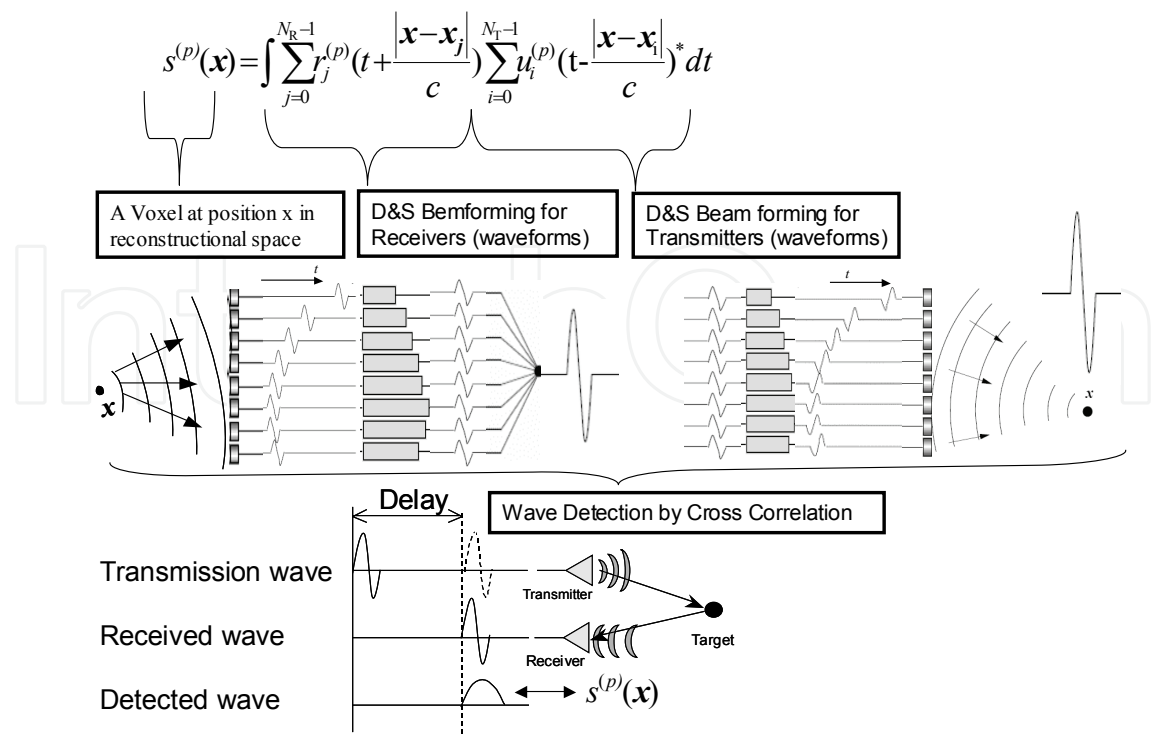


Fig. 3. Concept of image reconstruction.

$$\theta = \sin^{-1}(\sqrt{x^2 + y^2} / R),$$

(8)

and

$$\varphi = \tan^{-1}(y / x)$$

(9)

represent the polar coordinate of a position in 3D space, $r_i = \sqrt{x_i^2 + y_i^2}$ and φ_i gives the polar coordinate of the i -th element in the array plane. Fig. 4 shows the geometry of the transducer array and the image space.

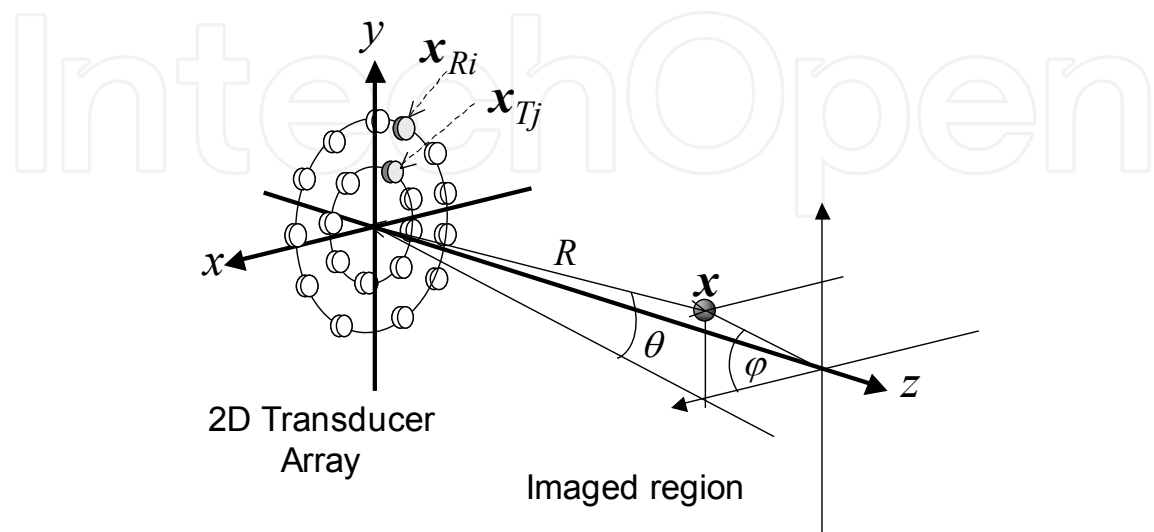


Fig. 4. Geometry of the array and imaged region.

When it is known that the object is stationary, a single image can be obtained as an accumulation of the complex image sequence:

$$I(\mathbf{x}) = \left| \sum_p s^{(p)}(\mathbf{x}) \right|^2. \quad (10)$$

4. Image reconstruction algorithm

The image reconstruction algorithm [6] involves the use of matched filter banks; it is generally composed of the following operations: a beamforming operation by delay and sum (D&S) and a matching operation by cross-correlation. The operations are performed in the frequency domain. Cross-correlation is performed by FFT, complex multiplications, and IFFT [7]. Frequency domain beamforming is computationally efficient [8,9]. Additionally, it is advantageous for HW implementation [9]. First, beamforming and cross-correlation in the frequency domain are performed by simple multiplications. Consequently, these computations can be performed with lesser computational complexity than that in the time domain. Second, the implementation of the frequency domain beamformer requires lesser hardware resource than that required in the time domain. The frequency domain beamformer requires few complex multipliers, while the time domain beamformer requires FIR filters by MACs (Multiply and ACcumulation). The time domain beamformer requires more HW resources because the FIR filter must contain more taps to achieve high throughput. Further, for HW implementation on an FPGA it is difficult to obtain a large number of taps with the limited FPGA resources. On the other hand, the frequency domain beamformer requires few complex multipliers for high throughput. Thus, the resource utilization is lesser than that in the time domain. Consequently, the HW architecture can provide high cost performance. Fortunately, the FPGA selected as the target device provides many high-performance dedicated multipliers; it is suitable for the HW implementation on the FPGA to utilize the multipliers for frequency domain image reconstruction.

From the above discussion, it is efficient for the HW to perform the operation in the frequency domain.

A schematic diagram of the operation is shown in Fig. 5. The filters compute the cross-correlations between the outputs of the received beamformer and the reference waveforms. Each signal is obtained by performing the D&S operation on the reference and received beamformer waveforms.

We have introduced a paraxial approximation [6] to compute the delays for the D&S operation in order to reduce the computational complexity. The procedure of the algorithm is as follows:

- i. The reference and received waveforms are transformed by using FFT into those in the frequency domain. The components within a given frequency band are processed. Consequently, the computational complexity is reduced. Let $H_i^{(p)}(f)$ ($i = 0, \dots, N_T - 1$) and $R_j^{(p)}(f)$ ($j = 0, \dots, N_R - 1$) denote the frequency-domain description of the transmitted waveform related to the i -th transmitter and the received waveform related to the j -th receiver for a frequency f in the p -th transmission, respectively.
- ii. The reference waveform $B_H^{(p)}(\theta, \varphi, f)$ corresponding to the direction (θ, φ) is computed by the D&S operation using $H_i^{(p)}(f)$ and phase rotations $(e^{-j2\pi f \tau_{\theta, \varphi, i}})$ that

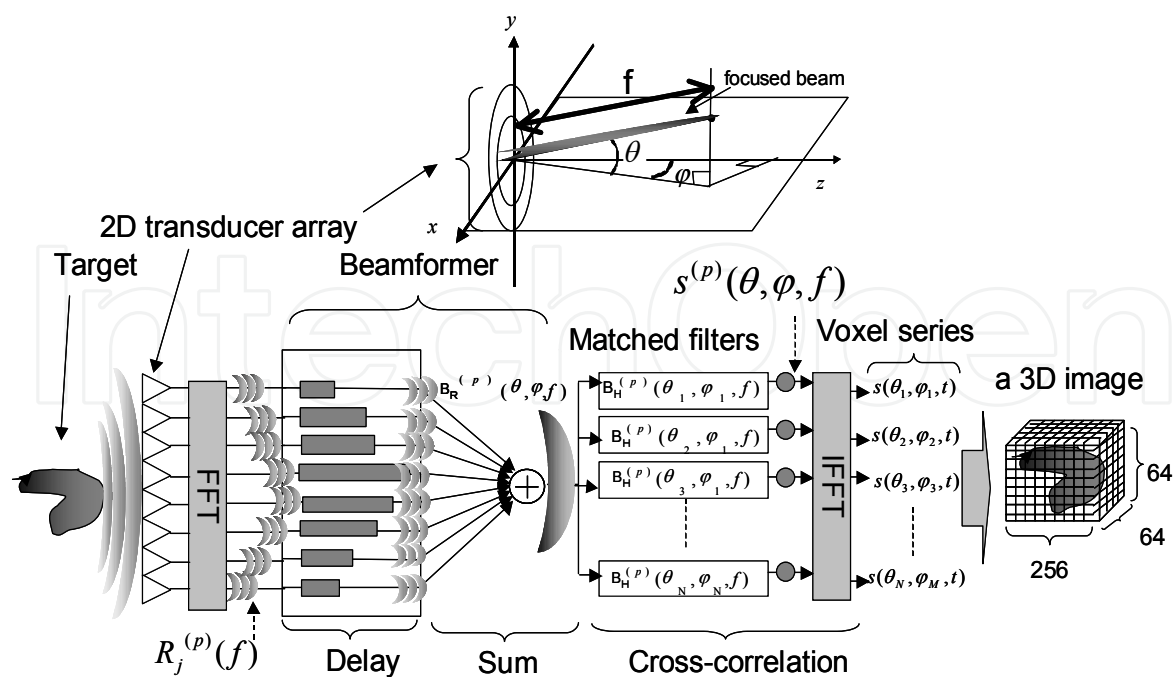


Fig. 5. Schematic representation of image reconstruction algorithm.

are obtained from the delay for the transmitters, which are stored as filter coefficients. Next, the waveforms received by the receivers are transformed by the FFT to the frequency domain, and $R_j^{(p)}(f)$ for each receiver is input to the beamformer containing phase rotations ($e^{j2\pi f\tau_{\theta,\varphi,j}}$) of the receivers; subsequently, the received beamform $B_R^{(p)}(\theta,\varphi,f)$ focused in the direction (θ,φ) in the frequency domain is computed. Each beamform is represented as follows.

$$B_H^{(p)}(\theta,\varphi,f)=\sum_{i=0}^{N_T-1}H_i^{(p)}(f)e^{-j2\pi f\tau_{\theta,\varphi,i}} \tag{11}$$

$$B_R^{(p)}(\theta,\varphi,f)=\sum_{j=0}^{N_R-1}R_j^{(p)}(f)e^{j2\pi f\tau_{\theta,\varphi,j}} \tag{12}$$

- iii. Cross-correlation is performed using the matched filters, and IFFT is carried out for $B_R^{(p)}(\theta,\varphi,f)$ and $B_H^{(p)}(\theta,\varphi,f)$. First, the matched filters perform the operation, which is expressed as follows.

$$s^{(p)}(\theta,\varphi,f)=B_R^{(p)}(\theta,\varphi,f)\cdot B_H^{(p)}(\theta,\varphi,f)^* \tag{13}$$

$$=\sum_jR_j^{(p)}(f)e^{j2\pi f\tau_{\theta,\varphi,j}}\cdot\sum_iH_i^{(p)}(f)^*e^{j2\pi f\tau_{\theta,\varphi,i}} \tag{14}$$

This operation is referred to as complex multiplication; $s^{(p)}(\theta,\varphi,f)$ denotes the output result from the matched filter to form a beamform in the direction (θ,φ) for the p -th ultrasound transmission and reception cycle.

- iv. The IFFT of $s^{(p)}(\theta,\varphi,f)$ is performed and the cross-correlation is completed. Consequently, complex voxel series is reconstructed in the direction (θ,φ) . The data samples output from the HW is 2048 [samples/line] \times 2 (real/imaginary). The series is in the time domain, and it is scaled to the spatial domain.

$$s^{(p)}(\theta,\varphi,t)=\mathfrak{I}^{-1}\left\{s^{(p)}(\theta,\varphi,f)\right\}$$

(15)

- v. The abovementioned sequence of operations is repeated for all the directions ($N_{\theta}\times N_{\varphi}=64\times 64$) in the reconstruction space.
- vi. Finally, the output voxels form a 3D image. The abovementioned operation is repeated for every shot of ultrasound.

5. Computational complexity for the algorithm

We estimate the computational complexity of the algorithm on the basis of the parameters shown in Table 1. The obtained result is shown in Fig. 6. Here, “op” is a unit of computational complexity that denotes the number of multiplications or additions. As shown in Fig. 6, the D&S operation accounts for approximately 80% of the overall operation. This is because this operation involves complex multiplication, as shown in equations (11) and (12); the operation must be performed for all the directions in the imaging space. Therefore, the parallel implementation of the D&S operation should be effective for high-speed operation.

Parameters			Values
Number of receivers	[ch]		32
Number of transmitters	[ch]		32
Observed data length for a channel (time domain)		[samples/ch]	2048
Observed data length for a channel (frequency domain)		[samples/ch]	512
Number of focus directions in the image reconstruction space	[line]		64 \times 64

Table 1. Parameters for estimating the number of operations.

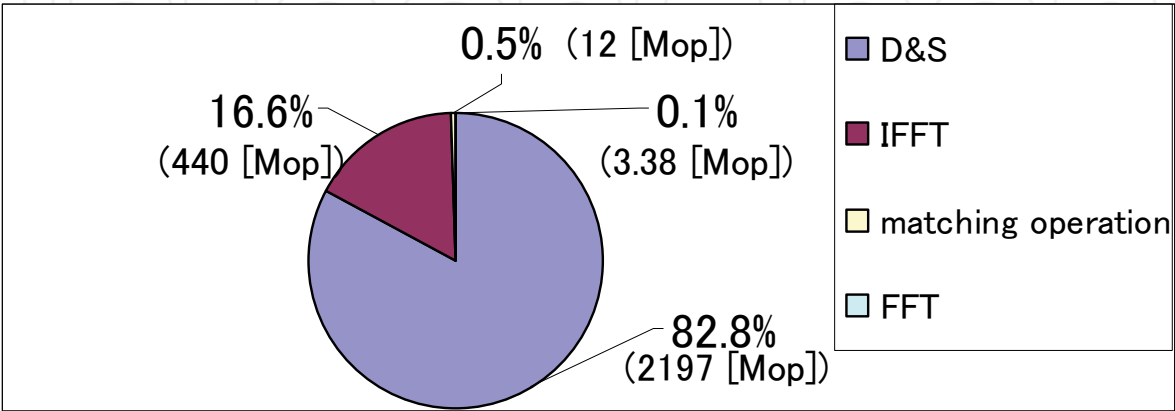


Fig. 6. Classification of operations.

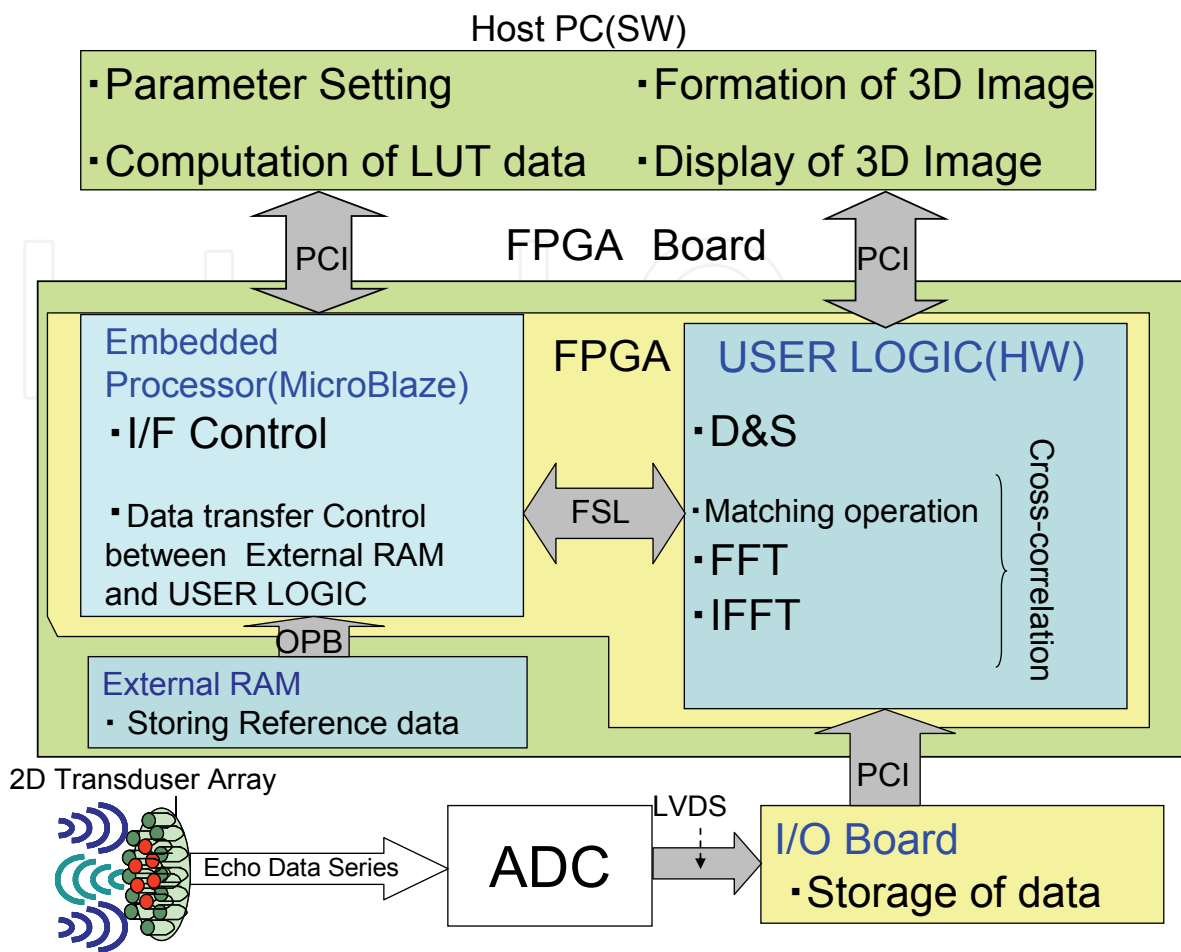


Fig. 7. Operational system.

6. Processing system

An outline of the processing system is described in this section. A schematic diagram of the system is shown in Fig. 7. The system consists of a host PC, the FPGA, an external RAM, ADCs, I/O boards, and a 2D transducer array. The function of each unit is as follows.

Host PC

Preprocessing and postprocessing operations are performed. In the former, the parameters required for the operation—reference waveforms and delay data—are determined. Subsequently, the obtained data are transferred to the FPGA board. In postprocessing, the host PC accepts the output of the FPGA and forms a 3D image, which is then displayed. The processing is repeated for every output of the FPGA.

FPGA

The use of FPGAs enables us to reconfigure the HW architecture when the specifications of the imaging system change. The FPGA consists of an embedded processor and user-defined logic (USER LOGIC). MicroBlaze (soft-core processor) is used as the embedded processor [10], and it performs I/F control of the PCI, fast simplex link (FSL) [10], and on-chip peripheral bus (OPB) [10]. In addition, it controls the data transfer between the host PC and USER LOGIC and between the external RAM and USER LOGIC. The delay data are stored

in the internal RAM of the FPGA since the data size is a few hundred kilobytes. USER LOGIC is a dedicated operational unit that performs the D&S operation and cross-correlation. After USER LOGIC accepts the reference and delay data through the embedded processor, it performs the D&S operation and cross-correlation by using the echo data obtained from the I/O board. Further, the operated results are output to the host PC through the PCI.

External RAM

The reference waveforms for ultrasound transmissions are stored in the external RAM because the total size of the data for the operation is a few megabytes.

ADC (A/D Converter)

The echo data observed by the 2D transducer array are converted to 16 [bit/sample] digital data.

I/O Board

It temporarily stores the echo data obtained from the ADC. When USER LOGIC requests echo data for starting an operation, the I/O board transfers the required data.

2D Transducer Array

The 2D transducer array comprises 32 transmitters and 32 receivers.

7. HW design

The functional block diagram of USER LOGIC shown in Fig. 8 is developed in a manner similar to that shown in Fig. 9. H-data, T-data, and R-data in Fig. 9 denote the reference waveform data in the frequency domain, echo data in the frequency domain, and delay corresponding to each transmitter and receiver needed to perform D&S operation, respectively. The functions of the components included in USER LOGIC are as follows:

FFT

The echo data are transformed into the frequency domain.

H-data and T-data Look-up Table (LUT)

The H-data LUT contains H-data for one-shot image reconstruction [11], while the T-data LUT contains the T-data corresponding to the distance between the focuses in the image reconstruction space and the devices in the 2D transducer array.

D&S Beamformer

The D&S operation is performed with the H- and R-data for each channel. The T-data are read from the T-data LUT and the phase rotations are computed. Subsequently, the product of the phase rotation element and the waveform data is computed, and the products are then summed. Finally, the outputs of the reference and the received beamforms are generated.

Matching Operator

$s^{(p)}(\theta, \phi, f)$ is obtained as the product of the outputs of the reference and the received beamform; this product is considered as a matching operation.

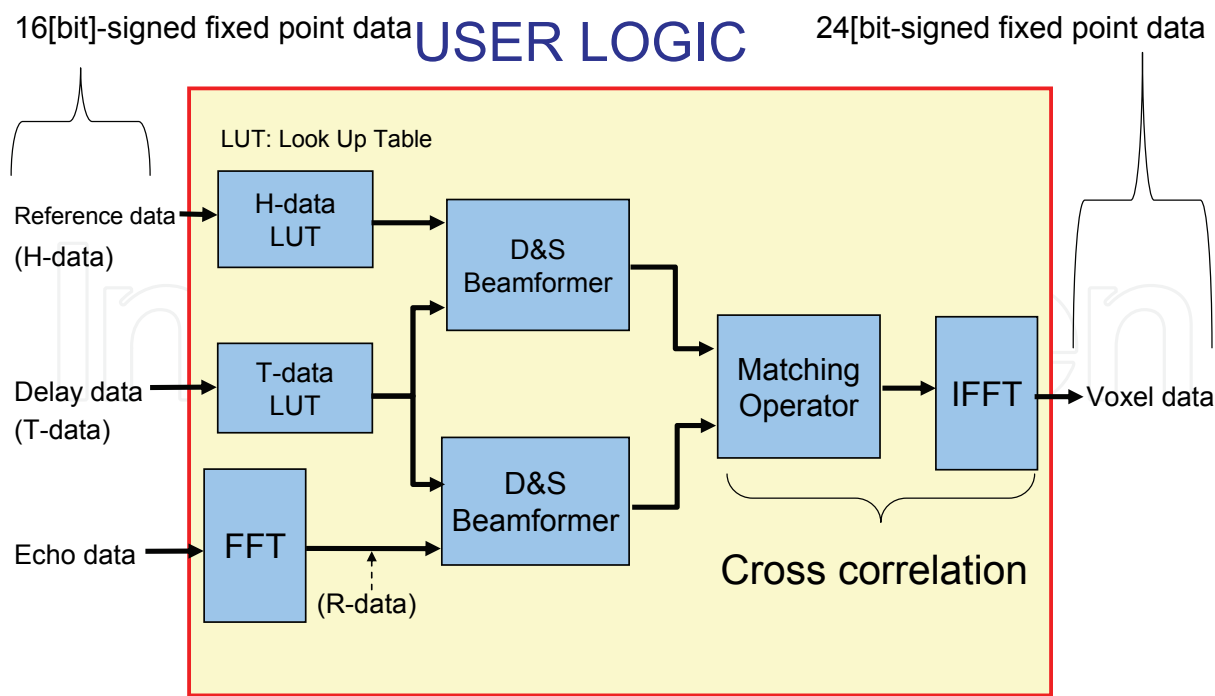


Fig. 8. Functional block diagram for HW design.

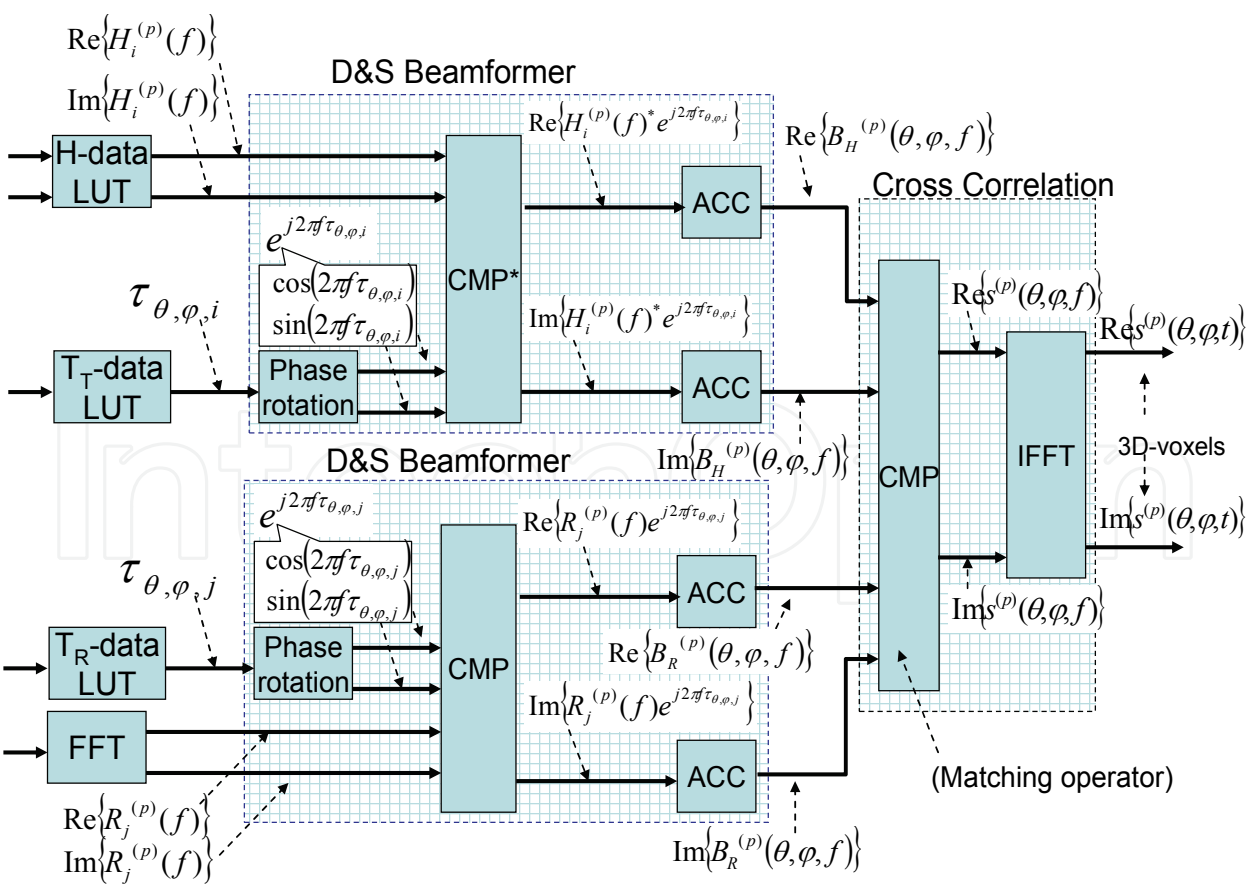


Fig. 9. Architecture mapping based on functional blocks.

IFFT

IFFT is performed for $s^{(p)}(\theta, \varphi, f)$ to complete the cross-correlation, and the complex voxel data $s^{(p)}(\theta, \varphi, t)$ are output.

Architecture mapping is presented in Fig. 9 for the HW design; the mapping is based on functional block diagram shown in Fig. 8. T_T - and T_R -data denote the delay corresponding to each transmitter and receiver, respectively. In addition, the figure includes the above equations to associate with the operational flow in the HW. We describe the function of each HW unit in detail.

The FFT and IFFT operators are realized by using Xilinx IP (Architecture type is Radix-2, pipelined, streaming I/O); the operations are performed for every one-channel data series and R-data (received data series in the frequency domain) are output as complex numbers.

The phase rotation unit (Fig. 10) [11] performs phase rotation using the T-data. The frequency f and delay τ are treated as discrete data. When the D&S operation begins, T-data from the RAM are input into the unit and an enable signal is asserted. This signal drives a counter that counts the number of frequencies. Sampled f_k and τ_l are input to the multiplier, and phase data $f_k \cdot \tau_l$ are output. The effective phase rotation angles are determined by the fractions of the phase data $f_k \cdot \tau_l$. Thus, fractional bits are used as effective data. The bits are then input to the sin/cos table ROM (Xilinx IP) to generate phase rotation. The table ROM outputs the sin and cos data corresponding to the ROM address; finally, the unit generates phase rotations.

CMP (Fig. 11) [11] and CAD (Fig. 12) [11] are the 18-bit multiplier and adder for the complex data, respectively. The units are composed of four multipliers and two adders; embedded multipliers are utilized to generate the multipliers. The CMPs in the D&S beamformer obtain the product of the waveform data and the phase rotation element. In fact, the CMP performs phase adjustment of the waveform data and the CMP used in the matching operator is utilized for the matching operation between the H-beam and the R-beam data in the frequency domain cross-correlation.

ACC (Fig. 13) [11] is the accumulator used to operate the beam data; it comprises an adder, two registers, a DMUX, and dual port RAM (Xilinx IP). The unit simultaneously performs addition and data write/read from the RAM for every clock cycle with a sample data output from the CMP in the D&S beamformer. The ACC repeats the operation for all the channel waveform data output from the CMP for the D&S operation and accumulates the results. Further, each beam data series is output by the switching of the DMUX when accumulation for all the channels is completed.

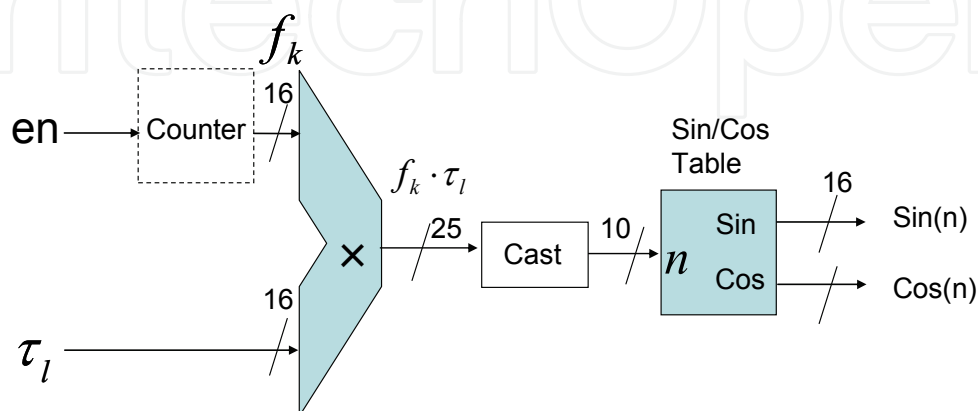


Fig. 10. Phase rotation unit.

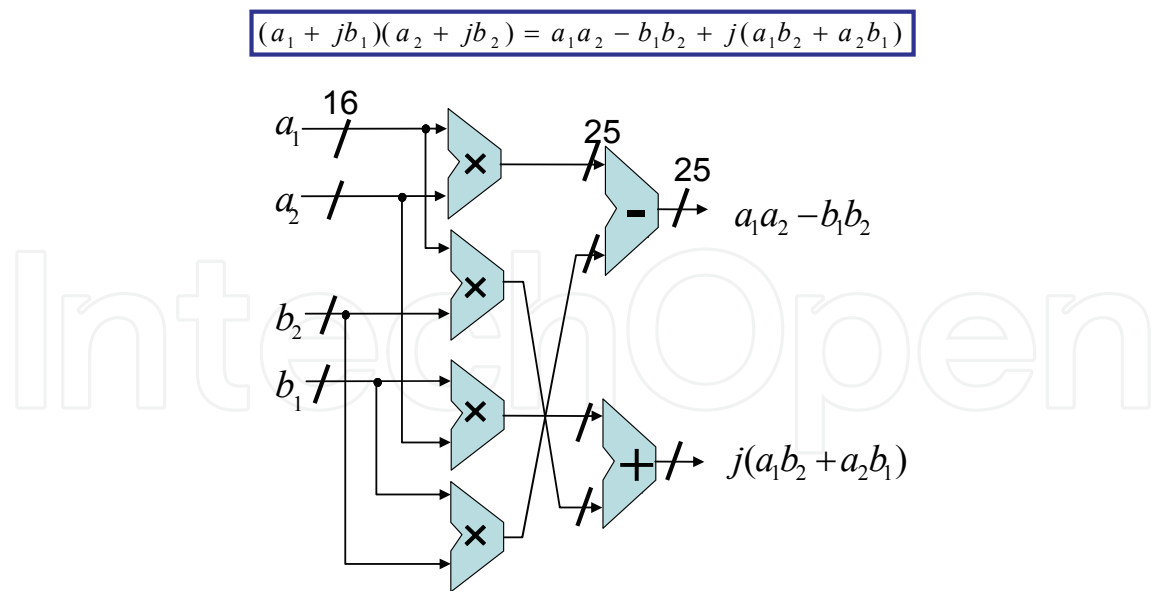


Fig. 11. Complex multiplier (CMP)

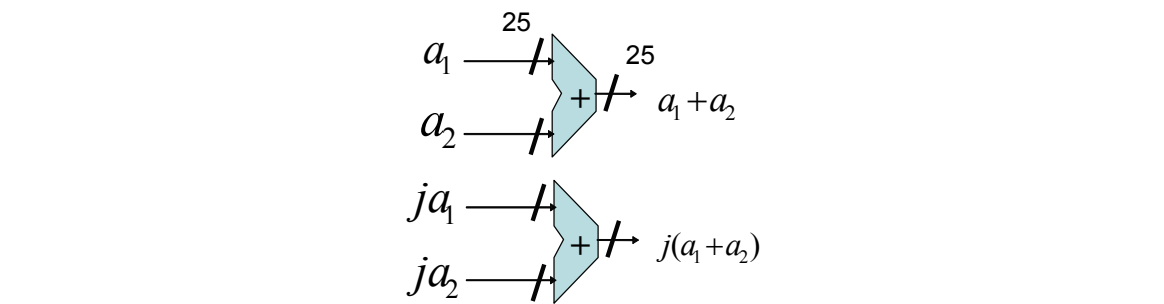


Fig. 12. Complex adder (CAD).

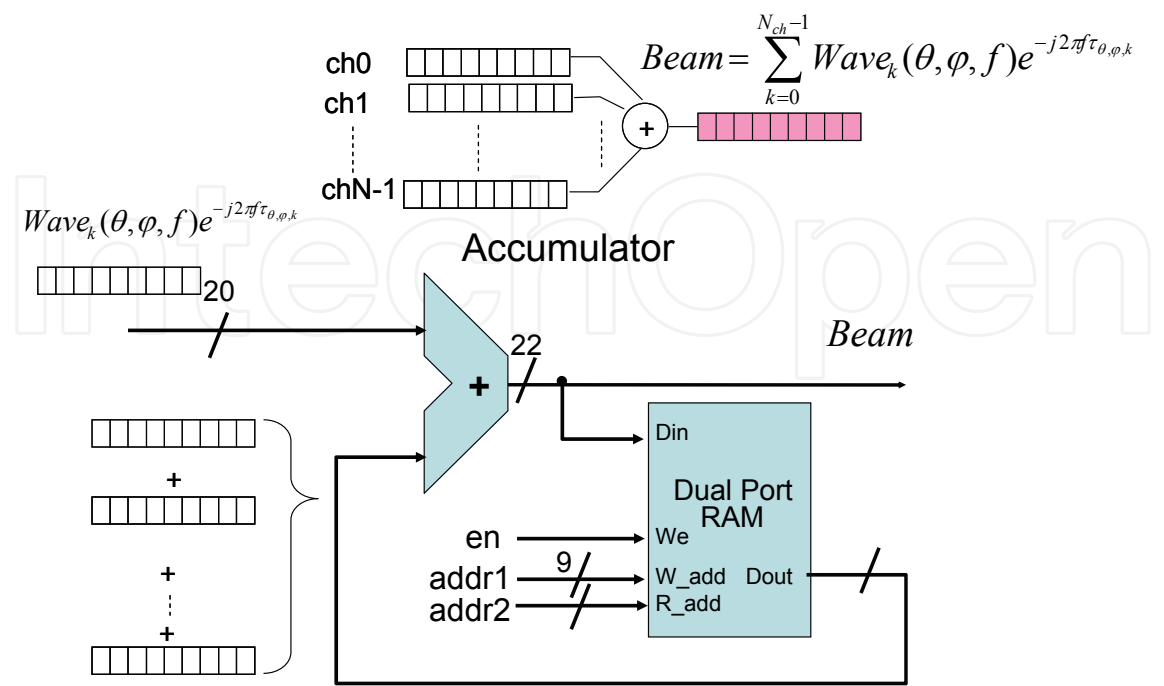


Fig. 13. Accumulator (ACC).

The architecture based on Fig. 6 corresponds to the use of the minimum number of functional units. In fact, the optimization placing multioperational units is not applied for the architecture to realize an efficient parallel operation. Here, we consider where the parallel operation is possible. Consequently, the parallel operation of the D&S operation is possible for optimization because the complex multiplication and accumulation can be performed for the waveform data of every channel. Fig. 14 shows refined architecture of the system shown in Fig. 9. The MicroBlaze MPU transfers the one-shot H-data series to the internal RAM whenever the HW completes a one-shot operation. The HW architecture is designed and simulated by using the design tools mentioned below. Further, VHDL is utilized for the HW design.

- Xilinx-ISE9.1i
- MathWorks – MATLAB/Simulink 2006a
- Xilinx – System Generator for DSP 9.1
- Mentor Graphics – ModelSimXELII 6.0a

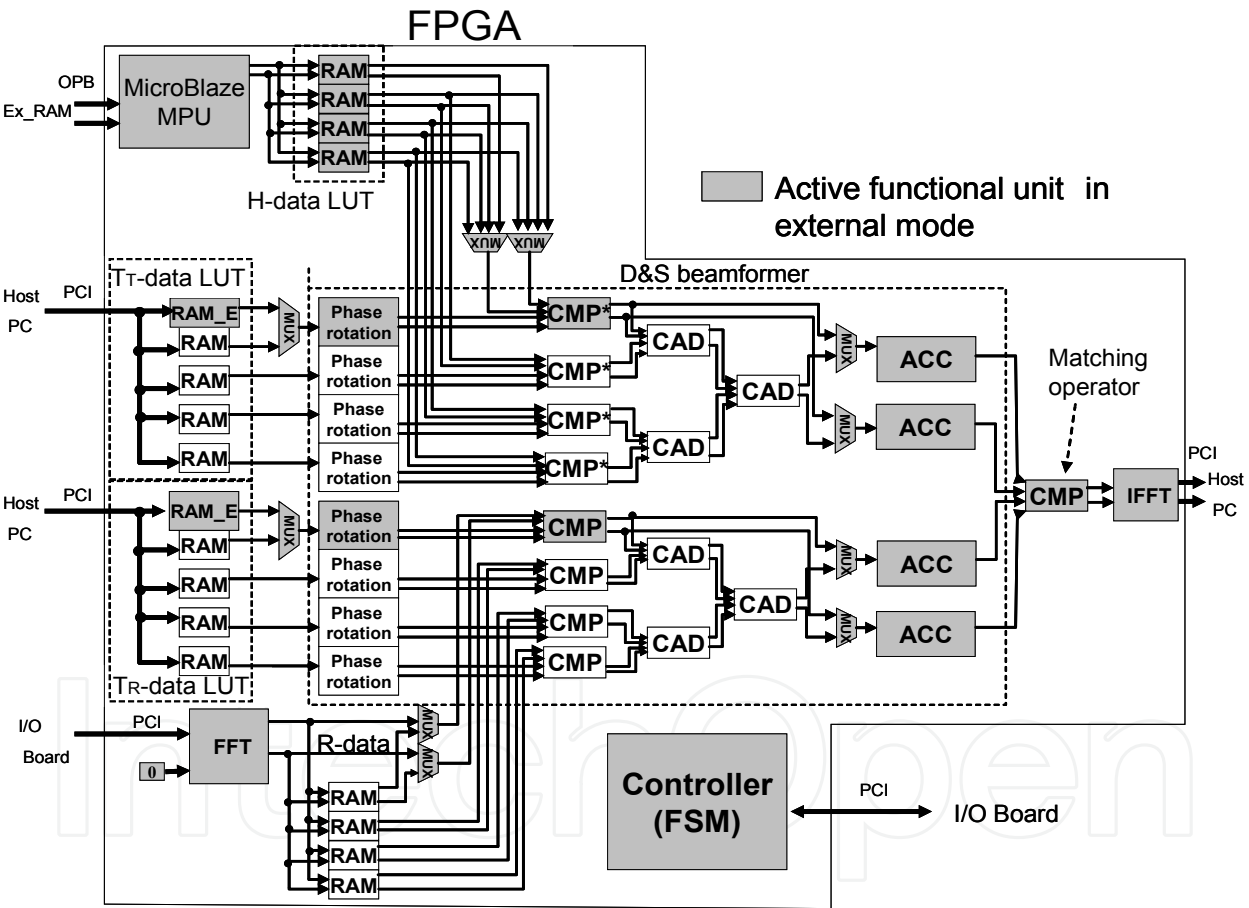


Fig. 14. Schematic diagram of HW.

8. Two operational modes

The HW operates in the following two modes.

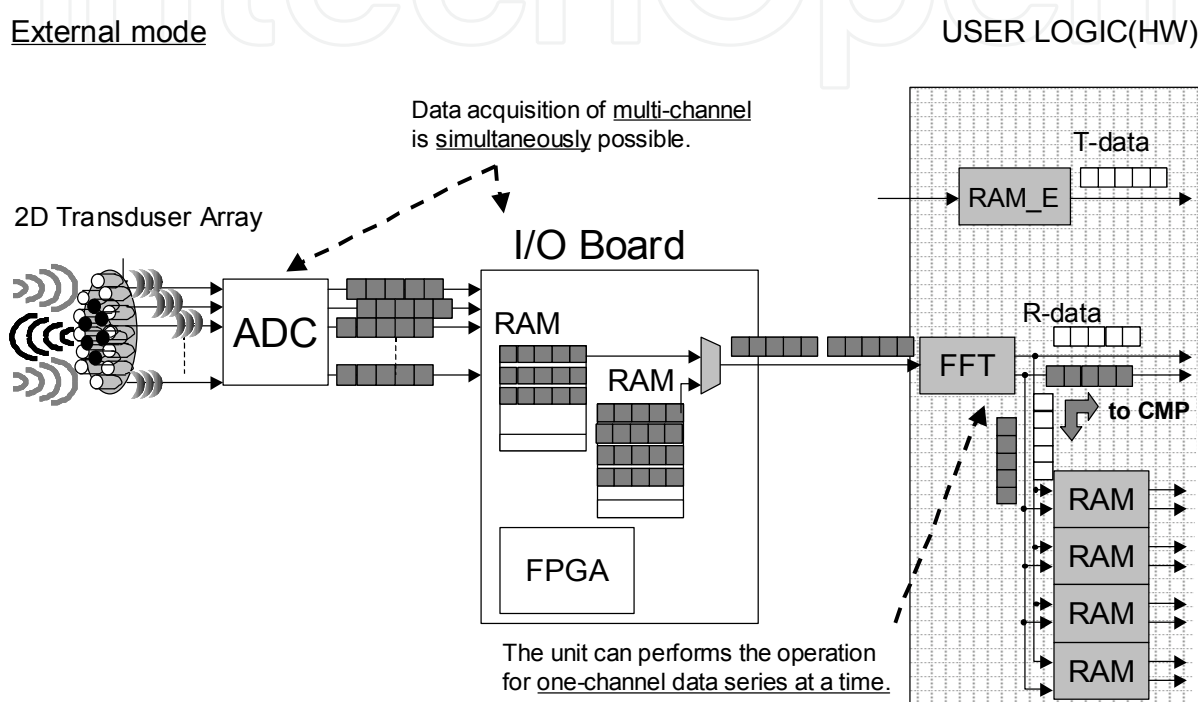
External Mode

The HW begins the operation in this mode. A schematic diagram of this mode is shown in Fig. 15. The operation is performed in this mode when the echo data that are transferred

from the I/O board are accepted by the FFT operator. This operator processes every 2048-point data (one-channel data length in the time domain), and the output is 512-point data in the frequency domain. RAM_Es are utilized when the HW operates in this mode only; the RAMs keep T-data (data size is 1 [line] \times 32 [ch] words) to perform image reconstruction in a direction. The H-data is read from the RAM following the operational output of the FFT operator; the D&S operation and cross-correlation are performed.

The abovementioned operations are performed for every single channel's waveform data. If the operation in a particular direction is completed, then the operational mode is switched to the internal mode.

External mode

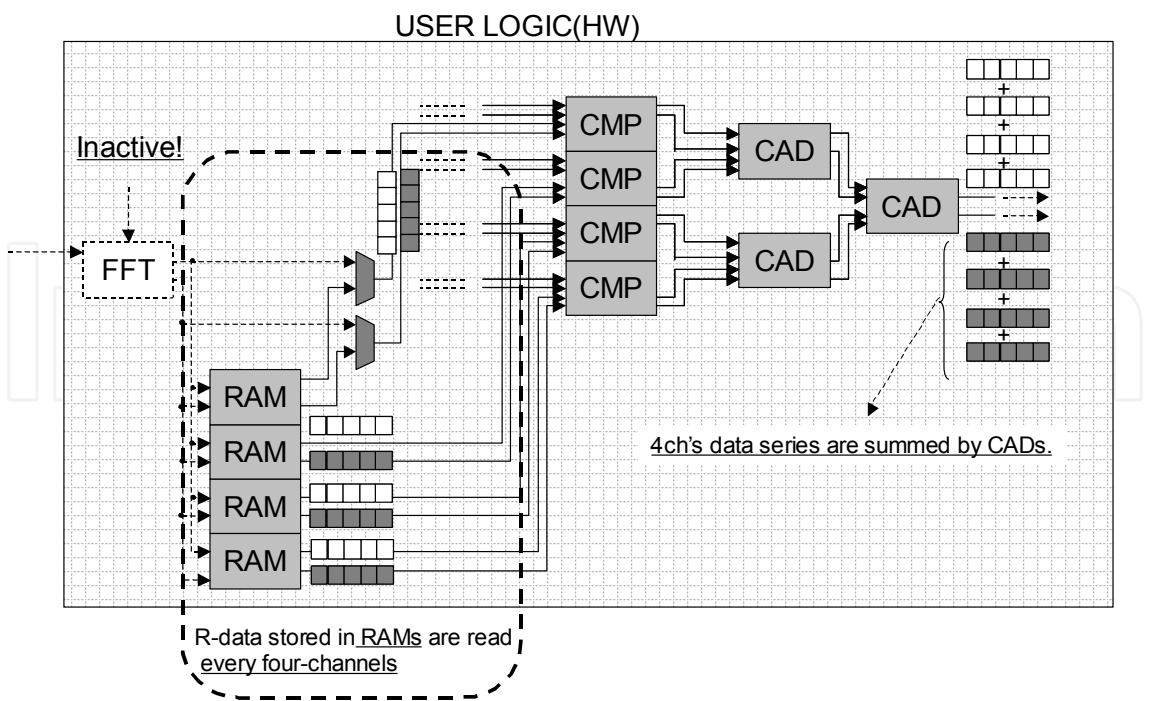


- Data acquisition and the operation are concurrently performed.
- Data are sequentially computed every one-channel.

Fig. 15. Schematic diagram for the HW operation in external Mode.

Internal Mode

In the internal mode (Fig. 16), four-channel R- and H-data series are simultaneously read from each RAM; these data are performed in parallel. To perform the operation in parallel, all the functional units, except the front FFT operator and RAM_Es, become active and drive. A schematic diagram of this mode is shown in Fig. 12. The CADs are utilized to sum the waveform data output from the eight CMPs in the D&S beamformer; the summed data series is sent to the ACC as a partial beamform generated by delayed wave data from four channels. The operation of this mode is repeated for all the remaining directions ((64 \times 64) - 1 [line]) in the image reconstruction space. When the HW is in this mode, the front FFT operator is in the inactive mode because the data acquisition is completed. However, if the operation in this mode is completed, MicroBlaze transfers the H-data from the external RAM to the internal RAM for the next shot operation. After that, the HW restarts the operation in the external mode.



- Only the operation is performed.
- Data are parallelly computed every four-channels.

Fig. 16. Schematic diagram for the HW operation in internal mode

The state diagram for the operational flow in the HW is shown in Fig. 17. The HW performs the operation following this state transition and as the state transition is repeated.

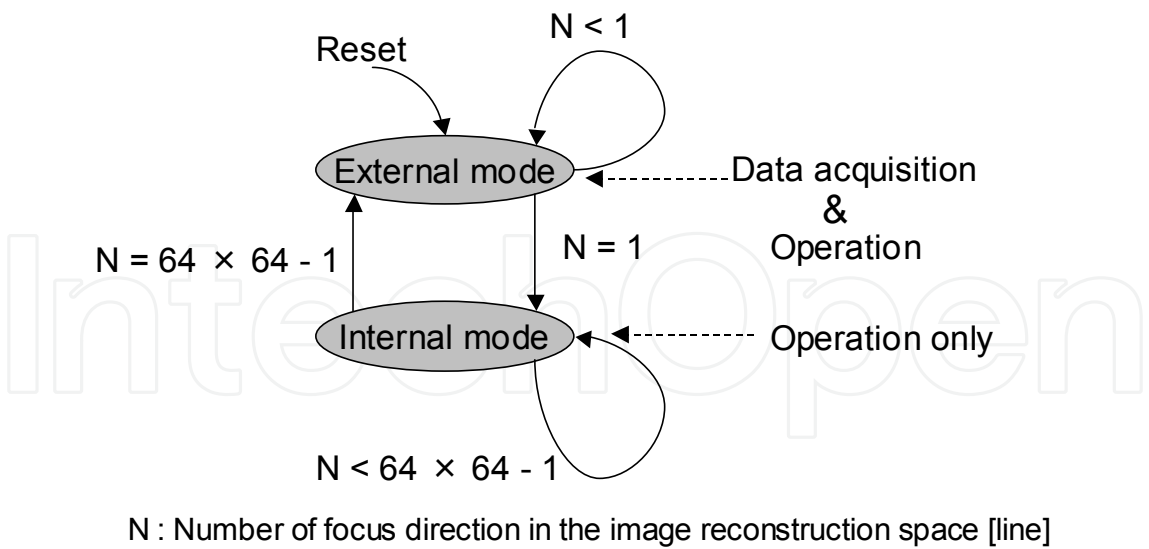


Fig. 17. State diagram for operational flow in the HW.

9. Synthesis result and performance evaluation

We synthesize the HW operation to implement it on an FPGA, and the scale and performance of the HW are evaluated. The target device is XCVFX100 of the Virtex-4 family.

Table 2 lists the device utilization summary for the resources required to implement the HW on the FPGA. Memory resources (FIFO16/RAMB16) are utilized more in comparison to other resources because a number of data with 64×64 [line] $\times 2 \times 32$ [ch] = 2^{18} words are required as the T-data for the D&S operation. Further, a number of data with 512 [word/ch] $\times 32$ [ch] $\times 2$ (real/imaginary part) = 2^{15} words are required to store the H- and R-data. DSP48s are utilized to construct FFT operators and CMPs; 60 DSP48s are utilized for FFT and IFFT operators; and 36 DSPs are utilized for CMPs.

Resource name	Used/Available	Utilization[%]
Slices	18541/42176	43.9
Slice Flip Flops	20928/84352	24.8
Four input LUTs	27098/84352	32.1
bonded IOBs	122/860	14.1
FIFO16/RAMB16s	350/376	93.0
GCLKs	1/32	3.1
DSP48s	96/160	60.0

Table 2. Device utilization summary for the HW on the FPGA.

With regard to the performance of the HW, the maximum frequency is approximately 137[MHz] (cycle time = 7.3 [ns]).

Next, we evaluate the latency to reconstruct a 3D image. For this, we obtain the number of the operational clock cycles required for a 3D image; the latency is obtained from the number of clock cycles and the maximum frequency. Consequently, the latency for 3D images with a resolution of $64 \times 64 \times 256$ voxels is approximately 170 [ms/frame], and the throughput is approximately 5.9 [frame/s]. When this result is compared with the latency of SW processing (approximately 40 [sec/frame]), the processing speed of HW is approximately 236 times faster than that of SW.

10. Critical path analysis and improvement

Next, we search critical (longest) path and then improve the path by modifying the architecture to the designed HW to raise clock frequency and processing performance. First, the HW is divided into some rough functional units. Then we search each path delay by synthesizing every the divided units. Consequently, there is the critical path on D&S beamformers from the synthesis report.

We search the path in detail again, we specify that the critical path exists on sin/cos table in phase rotation unit (Fig. 19). There is the path delay between input port for multiplier and output for sin/cos table.

So, we modify the implementation type of the table. As the table was implemented utilizing FPGA logic blocks in previous architecture, we change the implementation type utilizing Block RAM which is memory resources in FPGA (Fig. 20). Because FPGA logic block is treated as a combinational block, output signals from the table must be generated passing through plural stage's the logic block passes. Thus, the pass delay becomes longer in proportion to the stages. On the other hand, a table utilizing Block RAM generates by memory access only. Consequently, the pass delay becomes shorter than case of FPGA logic type, the critical path of the HW can be improved.

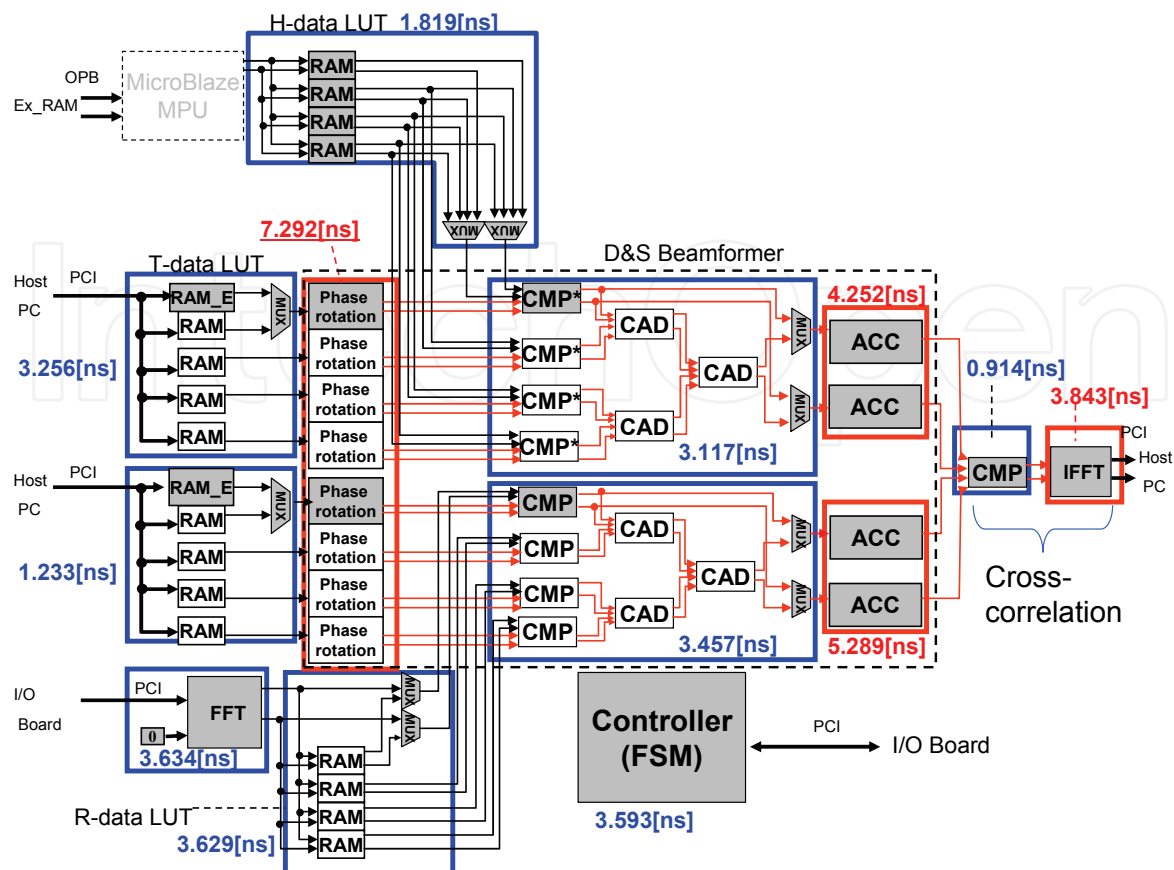


Fig. 18. Path delay for each units.

Phase rotation

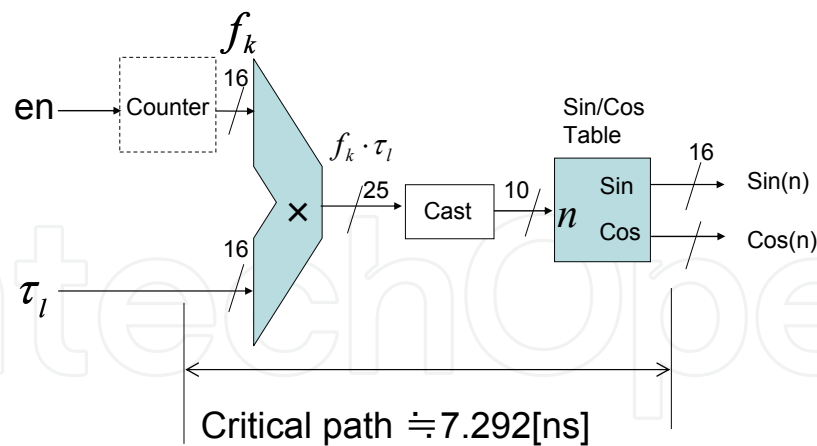


Fig. 19. Critical path in Phase rotation unit.

Subsequently, the next critical path exists on Dual port RAM in ACC of left hand in Fig. 21. In the same way, we modify the implementation type. The Dual port RAM was implanted by Block RAM. As alternative architecture, we implement the ACC utilizing shift register containing enable signal because ACC performs the operation by directly utilizing input data while abovementioned phase rotation unit’s operation indirectly is performed by table access. Subsequently, some registers are inserted between CMP in cross-correlation and IFFT core, critical path on the data path is improved.

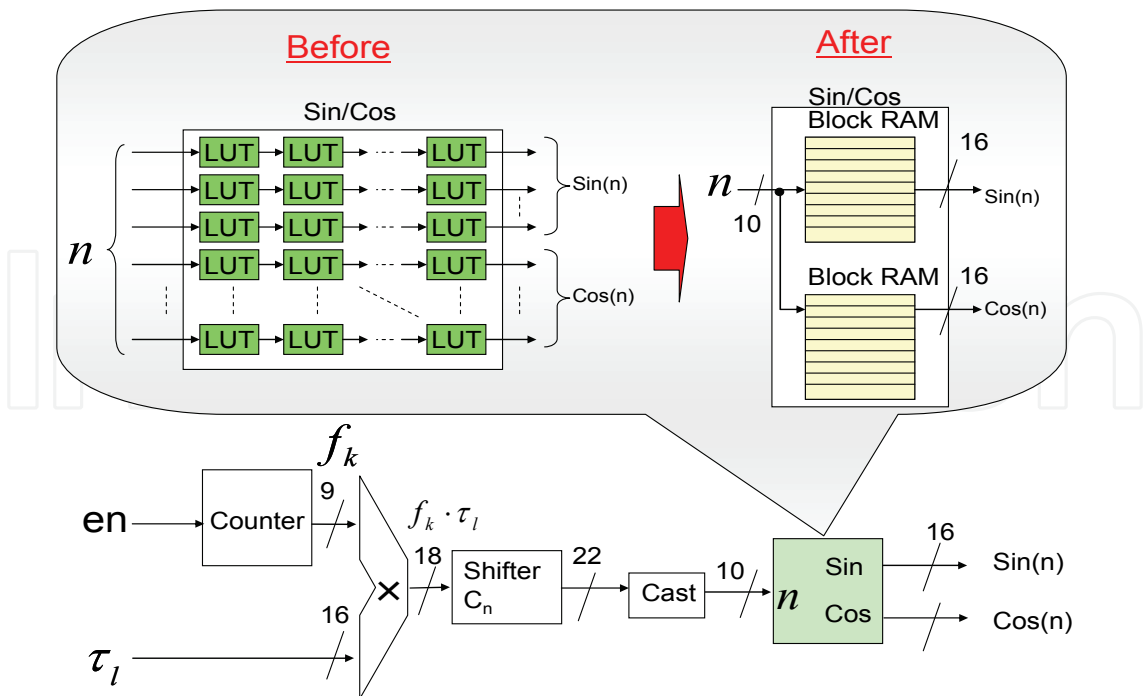


Fig. 20. Modifying sin/cos table based on FPGA structure.

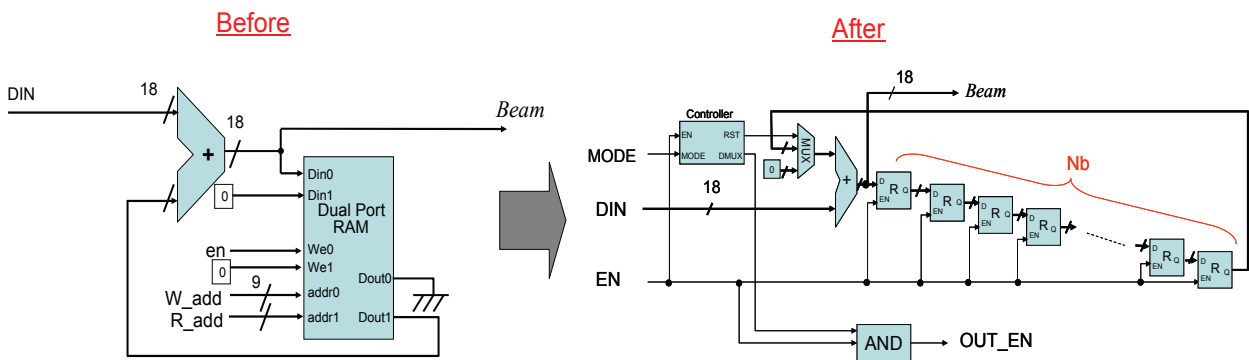


Fig. 21. Modification of architecture based on FPGA resources for ACC.

11. Synthesis result and performance evaluation after architecture improvement

Next, we synthesize the modified HW. Synthesis results are shown into TABLE 3. Consequently, FPGA logic blocks (Slices and Slice Flip Flops) are utilized in great quantities, HW scale became larger. Because architecture of ACC is improved from Dual port RAM to FPGA logic blocks, a large amount of the logic blocks are consumed.

Next, we evaluate performance after architecture improvement. Performance comparison is shown in Fig. 22.

Consequently, the maximum frequency is approximately 200 [MHz] (path delay \approx 5[ns]), the path delay is approximately 32% shorter than previous version. In performance (processing speed) evaluation, the latency is approximately 116 [ms/frame], and the throughput is approximately 8.6 [frame/s].

The modified HW processing is approximately 1.5 times faster than the previous HW, and the processing speed is approximately 350 times faster than above the SW processing.

Resource name	Used/Available	Utilization[%]
Slices	34245 /42176	81.1
Slice Flip Flops	60280 /84352	71.4
Four input LUTs	18035 /84352	21.3
bonded IOBs	122/860	14.1
FIFO16/RAMB16s	376/376	100.0
GCLKs	1/32	3.1
DSP48s	96/160	60.0

Table 3. Device utilization summary for the HW after architecture improvement

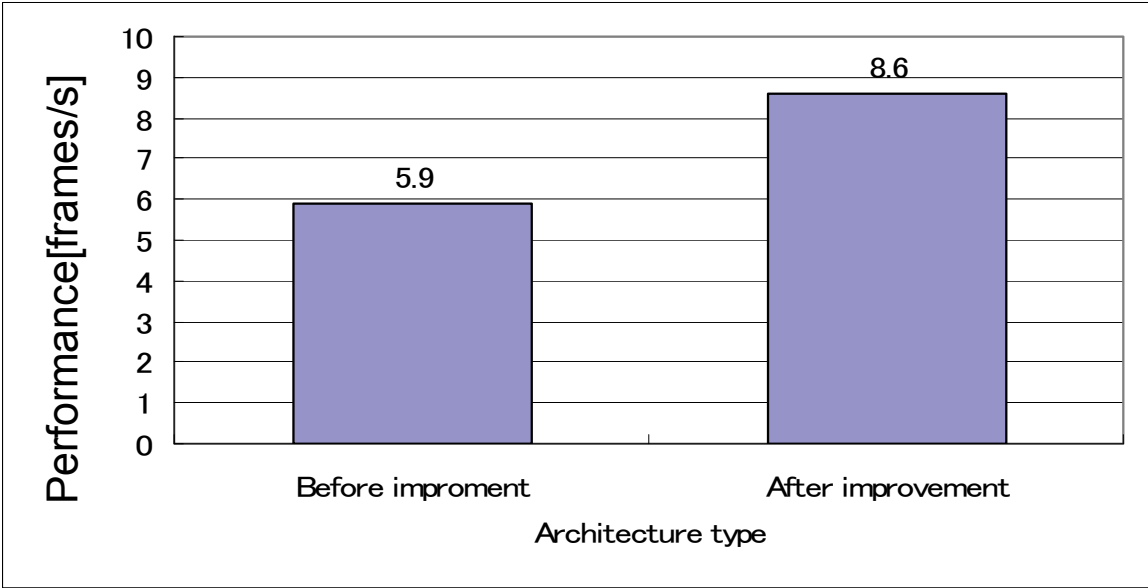


Fig. 22. Performance comparison.

12. Conclusion

In this study, we examined HW implementation on FPGA to realize efficient high-speed computation and lower cost with 3D ultrasound imaging. Image reconstruction is performed in frequency domain to reduce computational complexity of cross-correlation, designed HW contains tens of complex multipliers and FFT/IFFT units. Also we implemented some operational pipelines to realize parallel D&S which includes the most computational complexity in the operation. By performing the operation in frequency domain, the HW can be obtain high parallelism for HW implementation on FPGA including limited resources. Moreover the HW drives switching two operational modes according to action of the imaging system to allow efficient operation. Consequently, the HW's processing speed is approximately 240 times faster than the SW processing. Also, we tried critical path analysis and improvement of the HW, the

architecture modification based on FPGA resources was tried. Consequently. The modified HW's processing speed was approximately 1.5 times faster than previous version and 350 times faster than SW processing, respectively.

From the results in this study, we showed an effectiveness to design higher performance HW for FPGA utilizing cleverly its structure.

13. References

- [1] Y. Tamura and T. Akatsuka: "A Multiple Shots 3alized Modulating Signals", *Acoustical Imaging*, 20, pD Holographic Sonar Using a Set of Orthogon, pp737-743, 1993.
- [2] Y. Tamura, C. Ishihara, N. Okada, N. Ishii, M. Sato, T. Aoki, T. Hisamoto, and H. Yanagida : "High-Speed 3D Imaging System Using Coded Wavefront Generated by Walsh Function Modulated Signals", *Proc. of IEEE UFFC*, pp1666-1669 , 2002.
- [3] C. Patterson, B. Martin, S. Ellingson, J. Simonetti, and S. Cutchin : "FPGA Cluster Computing in the ETA Radio Telescope", pp25-32, *Proceedings of the 2007 International Conference on Field Programmable Technology, ICFPT 2007*, Kokurakita, Kitakyushu, Japan, December 2007.
- [4] Kentaro Sano, Oliver Pell, Wayne Luk, and Satoru Yamamoto, "FPGA-based Streaming Computation for Lattice Boltzmann Method", pp233-236, *Proceedings of the 2007 International Conference on Field Programmable Technology, ICFPT 2007*, Kokurakita, Kitakyushu, Japan, December 2007.
- [5] Yoshihisa Tanaka, Susumu Kuroyanagi, and Akira Iwata, "A Technique for Hardware Implementation of Neural Networks using FPGA", pp175-182, *IEICE technical report. Neurocomputing Vol.100, No.688*, March 2001.
- [6] K. Satoh, M. Miura, Y. Tamura, H. Yanagida, T. Yamasaki, C. Ishihara, and N. Okada, "Algorithms of 3-dimensional beamforming for synthetic aperture imaging system using pulses coded with Walsh functions", *Proc. of IEEE-UFFC Ultrasound Symposium*, pp1405-1408, Montreal, Canada, August 2004.
- [7] Steven W. Smith, "Digital Signal Processing: A Practical Guide for Engineers and Scientists - chapter9 Applications of the DFT", pp180-184, *Newnes*, September 2002.
- [8] F. Bertora, P. Pellegritti, and A. Questa, "An Alternative Frequency Domain Beam forming", *IEEE-UFFC Ultrasound Symposium*, pp1749-1752, Montreal Canada, August 2004.
- [9] Paul Graham and Brent Nelson, "FPGAs and DSPs for Sonar Processing Inner Loop Computations", *Technical Report CCL-1998-GN-1*, *Configurable Computing Laboratory*, Electrical and Computer Engineering Department, Brigham Young University, 1998.
- [10] Xilinx Co., "Embedded Magazine - The MicroBlaze v5.0 Soft-Processor Core: Optimized for Performance", pp18-21, *Xcell Publications*.
- [11] K. Satoh, J. Tada, and Y. Tamura: "Image Reconstruction Operational Hardware for 3D Ultrasound Imaging", pp43-49, *Journal of Society of Signal Processing Applications and Technology of Japan*, December 2006.



Vision Sensors and Edge Detection

Edited by Francisco Gallegos-Funes

ISBN 978-953-307-098-8

Hard cover, 196 pages

Publisher Sciyo

Published online 12, August, 2010

Published in print edition August, 2010

Vision Sensors and Edge Detection book reflects a selection of recent developments within the area of vision sensors and edge detection. There are two sections in this book. The first section presents vision sensors with applications to panoramic vision sensors, wireless vision sensors, and automated vision sensor inspection, and the second one shows image processing techniques, such as, image measurements, image transformations, filtering, and parallel computing.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Keiichi Satoh, Jubee Tada, Yasutaka Tamura, Gensuke Goto, Toshio Koga and Kazuhiro Kondo (2010). Three-Dimensional Ultrasound Imaging Using Hardware Acceralator Based on FPGA, Vision Sensors and Edge Detection, Francisco Gallegos-Funes (Ed.), ISBN: 978-953-307-098-8, InTech, Available from: <http://www.intechopen.com/books/vision-sensors-and-edge-detection/three-dimensional-ultrasound-imaging-using-hardware-acceralator-based-on-fpga>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen