2005 WSEAS Int. Conf. on DYNAMICAL SYSTEMS and CONTROL, Venice, Italy, November 2-4, 2005 (pp463-466)

A Compact FPGA Beamformer Architecture

IOAN LIE, MIHAIL EUGEN TANASE
Applied Electronic Department
„Politehnica“ University of Timisoara
Bd. Vasile Parvan, no. 2, Timisoara, 300223
ROMANIA

Abstract: - This paper presents the structure and the implementation of a delay and sum beamformer with non-uniform sampling. For phased array systems with large channel counts, simplifying the front-end electronics is of fundamental importance because it comprises a critical part of the system in terms of cost, size and power consumption. Digital beam forming based on non-uniform sampling accomplishes sampling and focusing simultaneously and hence can possess efficient front-end architectures. A compact architecture can be implemented in programmable circuits (FPGA) which contains the sampling clock generators, the memory buffers and the digital adder. For nonuniform sampling clock generation iterative concepts for on-chip calculation of delay information was investigated. The digital beamformer was tested using Altera Quartus environment and implemented in Stratix devices.

Key-Words: - beamformer**,** nonuniform sampling, ultrasound, delay

1 Introduction

Beamforming is a method of observing signals received by a transducer array from a desired direction while attenuating the response of the array from other directions. The fundamental idea which beam forming is based on is achieving the condition of coherent summation of the signals received by the elements of an array from a point in space. For the near field, where the front of wave is spherical, the beam formation supposes the steering after a direction and focusing in a given point on that direction. In the far field where wave front can be considered plane the formation of the beams is reduced to the steering after a direction.The implementation of the coherent summation condition is made through the algorithm known as “delay and summ”. The time-domain beam formation techniques are practically, versions of the “delay and summ” algorithm whose goal is to achieve as accurately as possible the coherent summation condition.

The evolution of ultrasound beam formation techniques is a process that rolls up collaterally with the development and improvement of the processing resources. The final goal of the formation process was always the same, the most accurate summation of delayed signals, but the implementation was subordinated to the stage reached on designing the electronic circuits and processing algorithms.

In most conventional digital focusing systems, pulse echoes received at array elements are sampled synchronously by the same uniform sampling clock. For each imaging point, one of the samples is selected at each element, taking into account the focusing delay, and is summed to those at other elements. Hence, the time delay between consecutive samples is the most important parameter to determine focusing delay error which is related to the frequency range of an employed pulse echo. In addition, the digital beamforming hardware is complex, being dependent on the sampling rate and the number of array elements. As the sampling rate increases in a digital beamformer, larger and faster digital memories are required, and accordingly, the addressing circuit on these memories becomes more complex. Hence, for dynamic focusing using a two-dimensional transducer for three-dimensional scanning, the whole beamforming system can be extremely complex and expensive.

The beamforming technique by using non-uniform sampling marks the focal interest in designing ultrasonic imaging equipment over the complexity of the system. In this beam forming technique, the non-uniform sampling of the RF signal is used in order to ensure the desired delay accuracy, still keeping the sampling frequency at a relatively low value. Only the necessary samples are taken in order to focus the ultrasonic beam in each image point. With this in mind different clock signals are generated for each transducer element, and these signals comply with the Nyquist criteria but are delayed one in relation to the other to ensure focusing in the desired point. This leads to a dramatic reduction in hardware requirements

compared to conventional methods that use uniform sampling techniques.

2 Beamformer Architecture

Figure 1 show an efficient digital beamforming architecture, called the Pipelined Sampled-Delay Focusing (PSDF) scheme which has been proposed previously by one of the authors [1] in order to reduce the high sampling rate. Each channel contains a Time Gain Control Amplifier (TGC) an 8 bit converter (ADC) and a FIFO memory. In this scheme, only the samples needed to focus the ultrasonic beam at each imaging point are obtained. This requires nonuniform sampling using different sampling clocks at each element. In other words, the sampling and the delay processes required for focusing are done simultaneously by nonuniform sampling of signals rather than being done separately as in conventional digital beamforming. Consequently, the sampling rate in the PSDF scheme is determined by the distance between imaging points, or more generally by the minimum sampling rate for envelope detection of the focused signal. For an ultrasonic transducer with a bandwidth equal to its central frequency, the minimum sampling rate in the PSDF scheme, when f_o = 3 MHz, can be 9 MHz (Nyquist rate) or 3 MHz (twice the baseband bandwidth) using bandwidth sampling techniques such as the second-order sampling and quadrature sampling. This leads to a dramatic reduction in hardware requirements compared with the conventional uniform sampling methods. Furthermore, the usage of FIFO memories eliminates the addressing operation for sampled signals, further reducing the structural complexity of the beamformer.

The FIFO writing operation is synchronized with the nonuniform sampling clock and the reading operation is synchronized with a uniform clock used to drive the digital adder. The adding process starts after the data acquisition process is complete and therefore any conflict between reading and writing in the memory is avoided.

In this paper, we propose the FPGA implementation of an efficient dynamic focusing system based on the PSDF scheme. The PSDF system is efficient in the implementation of dynamic focusing from the signal processing and hardware points of view. The amount of digital memory in the PSDF scheme cannot be reduced in any other digital multi-bit beamforming architectures.

The size of the FIFO memory is dependent on the maximum investigated range and the sampling rate. For a length of R=30cm of the scanning range, and a sampling rate ten times higher than the central frequency of a transducer, 30 Mhz, the FIFO size is: 12000. Considering the usage of the Altera Stratix devices which have the ram memory organized in blocks of 4 Kbits a FIFO memory size of 16 Kbits for each channel was chosen. The key element of this system is the non-uniform sampling clock generator. The sampling clock generator (SCG) can be implemented with a simple circuit using an iterative "in the circuit" calculating for the delay information

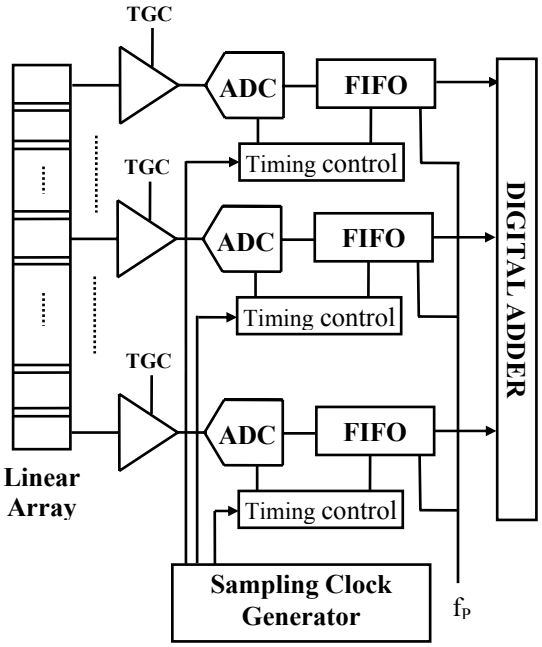

Fig. 1 Beamformer Architecture

3 The Sampling Clock Generator

Fig. 2 presents the geometry used for determining the focusing delays for a phased array. The central point of the array is located in the origin O of the reference system. P is the focal point, r is the distance between the focal point and the central point, θ is the steering angle for the beam and x represents the position of one element in relation to the array center. In a homogeneous undamped medium, the focusing delay for an array element is represented as the time, l/c, where c is the speed of sound and l is the travel difference expressed by the following equation.

$$l = (r^2 + x^2 + 2r \cdot x \cdot \sin\theta)^{1/2} - r, \qquad (1)$$

In (1) r is varied according to a scan line OP, θ is invariant with respect to the scan line and x is constant with respect to a given array element. The direct calculation of the focusing delay is impractical because it requires time-consuming operations of

multiplying and square root extraction. Therefore it is necessary to apply a numerical method to simplify the evaluation of (1). One such method is the "midpoint algorithm" described in the [2]. In short, the use of this algorithm for the delay time calculation is explained in reference to the 1-r curve in figure 3.

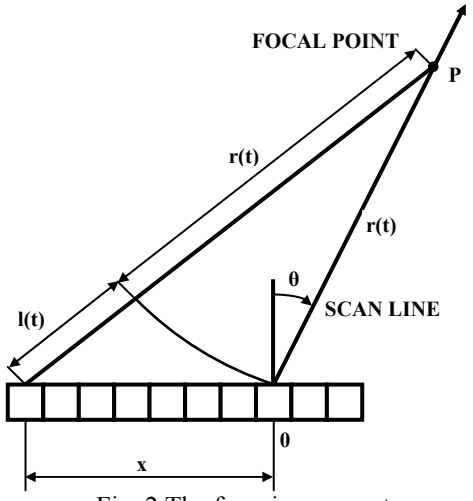

Fig. 2 The focusing geometry

The variable *i* is the estimated quantified value of the relative delay and the indexed variable *r* describes the domain sampling with the period $t_s = r/2c$. Applying this algorithm assumes that the equation (1) can be expressed as $f(r,l)=0$ and it satisfies the following conditions:

The coefficients of $f(r,l) = 0$ equation are integers

$$f(r,l) = l^2 + 2lr - \alpha r - \beta,\ \alpha = 2x\sin\theta,\ \beta = x^2 \quad (2)$$

r and l satisfies the equation: $-1 \le dl/dr \le 0$ (3)

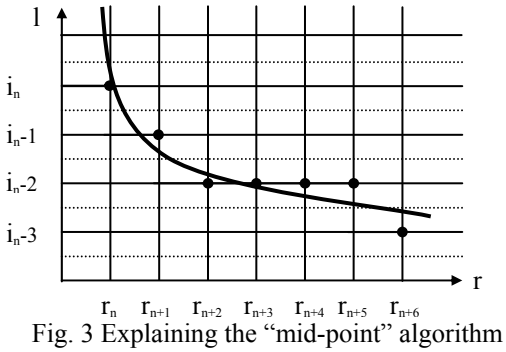

Fig. 3 Explaining the "mid-point" algorithm

The "mid-point" algorithm estimates the value of *l* depending on *r* from the delay equation using addition operations of integers. Heuristic the mid-point algorithm chooses, from the grid, the quantified value *i*, the closest to the real *l-r* curve. Because the quantified values are uniformly spaced, the closest *i* value is situated at less than half the quantifying level from the curve.

The algorithm: it is assumed that i_n is the integer value closest to the real value of the delay l_n corresponding to the focal point *n* situated at the r_n distance If r_n and i_n are set as initial conditions and further the $r_{n+1} = r_n+1$ condition is fulfilled, then it is possible to obtain an equation $i_{n+1} = i_n$ or $i_{n+1} = i_n-1$ by (2). So the integer value of the delay for the n+1 focal point is determined a being i_n or i_n-1 depending on the decision variable d_n defined by following equation:

$$d_n = 4f(r_n+1, i_n-0.5) =$$
$$= -4i_n^2 + 4i_n + 8i_nr_n - 4r_n(1+\alpha) - 4\alpha - 4\beta - 3 \quad (4)$$

The decision variable results by multiplying by four the equation (2) and represents its value for a mid-point situated between the coordinates (r_{n+1}, i_n) and (r_{n+1}, i_n-1). As can be seen in figure 3 when the decision variable is positive, because the i_{n+1} delay is closer to the i_n-1 value than to the i_n value, the i_{n+1} delay will be updated to the i_n-1 value. Else, i_{n+1} will be updated to the i_n value. In this manner, the i_n delay can be calculated based on the r_n focal domain that is progressively incremented.

The algorithmic calculation for one integer value of the focusing delay using the relation (4) is obviously more efficient than the initial one, but it still has multiplication operations that require time and resources. In order to avoid the multiplications we can take into account an incremental expression for the decision variable

If the distance unit is the distance between two focal points, then the decision variable must be refreshing at every incrementing of the *r* distance and for high sampling frequencies the amount of calculations increase significantly. If the value of *i* isn't calculated for each *r=n*, but only for *r = Kn* then the amount of calculations are reduced by *1/K*. In this case (2) and (3) becomes:

$$f(r,l) = l^2 + 2\cdot l\cdot K\cdot r - \alpha\cdot K\cdot r - \beta \quad (5)$$

$$-1/K \le dl/dr \le 0 \quad (6)$$

Substituting *r* with *Kr*, the algorithm is described by:

$$d_n = 4i_n^2 + 4i_n(2K-1) + 8Ki_nr_n - 4Kr_n(1+\alpha) - 4K(1+\alpha) - 4\beta+1 \quad (7)$$

for $d_n \le 0$: $d_{n+1} = d_n + 8Ki_n - 4K(1+\alpha)$ (8)

for $d_n > 0$: $d_{n+1} = d_n - 8(1-K)i_n - 8Kr_n - 4K\alpha - 20K + 8$ (9)

Implementing the algorithm only with addition operations that have two terms to be added is possible if the decision function is expressed by [3]:

$d_{n+1} = d_n + A_n$ for $d_n \le 0$ and $d_{n+1} = d_n + B_n$ for $d_n > 0$

$A_n = 8Ki_n - 4K(1+\alpha)$;

$B_n = -8(1-K)i_n - 8Kr_n - 4K\alpha - 20K + 8$

With these modifications, A_{n+1} and B_{n+1} can be incrementally expressed as:

$A_{n+1} = A_n$, $B_{n+1} = B_n - 8K$, for $d_n \leq 0$

$A_{n+1} = A_n - 8K$, $B_{n+1} = B_n - 16K + 8$, for $d_n > 0$

Therefore if the initial A_0 and B_0 are known then A_n, B_n, d_n, r_n and i_n can be sequentially obtained for n = 0, 1,... The block schematic presented in Fig. 4 contain only registers, multiplexers and two-entry adding blocks. The values for the C1 and C2 constants stored in REG_{C1} and REG_{C2} are available at MUX1 multiplexer's exits. The REG1 register stores the refreshed value of the A term, and the REG2 stores the refreshed value of the B term. The sign of the decision variable (MSB) commands the refreshing, and it is held in the REG3 register. MSB(D) also selects witch one of the MUX2 multiplexer's exits is added to the content of the REG3 register, in order to get the updated value of the decision variable *d*. The sampling clock generator decrements the delay value every time *d* is positive, MSB is Low, and generates an additional sampling pulse Sclk.

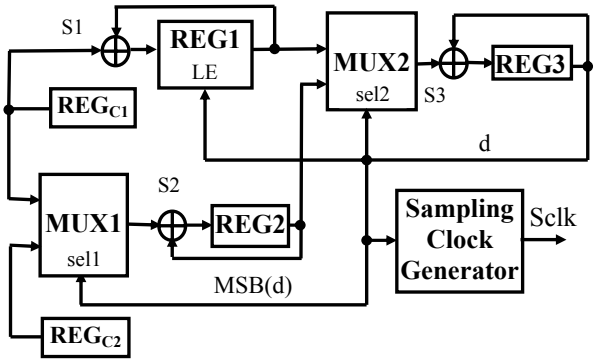

Fig. 4 Block diagram for "mid point" algorithm

4 Results

The sampling clock generator has been simulated in Quartus environment and the structure for one channel, except the FIFO memory, has been implemented in the largest Flex10K device. Based on the partial implementations in a Flex device the digital section for a 32 channel beamformer has been successfully implemented in an Altera Stratix device. In order to optimize the memory structure a device which has the ram memory organized in blocks of 4 Kbits was choose. The Stratix II family offers a TriMatrix memory consisting of three RAM block sizes to implement true dual-port memory and first-in first-out buffers. The largest Stratix II device EP2S180 contains 930 M512 RAM blocks with 512 bits, 768 M4K RAM blocks 4K bits and 9 M-RAM blocks with 512K bits. The 16K x 8 bits FIFO memory required for each channel can be structured from an integer number of elementary blocks. Because different sampling clock drive each channel each M-RAM block can host the FIFO memory for a single channel. The device utilization summarized in table II suggest that four Stratix device contains enough resources to implement the hardware for a digital beamformer with 128 channels.

Table 1 Device utilization for EP2S180F1508C4

Flow Status	Successful - Wed Sep 14 15:00:55 2005
Quartus II Version	4.1 Build 181 06/29/2004 SJ Full Version
Entity Name	Stratix128c
Family	Stratix II
Device	EP2S180F1508C4
Timing Models	Preliminary
Total ALUTs	7,996 / 143,520 (5 %)
Total registers	2912
Total pins	352 / 1,171 (30 %)
Total memory bits	4,063,232 / 9,383,040 (43 %)
Total PLLs	0 / 12 (0 %)
Total DLLs	0 / 2 (0 %

4 Conclusion

A hardware implementation is presented for a receive beamformer based on nonuniform sampling. Using this method results an important reduction in hardware requirements compared with the conventional uniform sampling methods. Furthermore, the usage of FIFO memories eliminates the addressing operation for sampled signals, further reducing the structural complexity of the beamformer. In order to generate the non-uniform sampling clock with a minimum logic and memory resources, the possibility of iterative "in circuit" calculating for the delays was analyzed. A 32 channel beamformer using 8-bit ADC been successfully implemented in a Stratix device. The results show the possibility to build a compact multi-bit beamformer using programmable devices with embedded memory blocks.

References:

[1] T. K. Song and J. F. Greenleaf, Ultrasonic Dynamic Focusing Using an Analog FIFO and Asynchronous Sampling, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, Vol. 41, No. 3, May 1994, pp. 326-332

[2] J. R. Van Aken, An Efficient Ellipse-Drawing Algorithm, *IEEE Journal of Computer Graphics and Application*, Vol. 4. no. 9, 1984, pp. 24-35.

[3] S.B. Park, M. H. Bae and K. Jeon, Real Time Digital Reception Focusing Method and Apparatus Adopting the same, *U. S. Patent #5,669,384*, September 23, 1997.