



## Implementation of ultrasound time-domain cross-correlation blood velocity estimators

Jensen, Jørgen Arendt

*Published in:*  
I E E E Transactions on Biomedical Engineering

*Link to article, DOI:*  
[10.1109/10.243415](https://doi.org/10.1109/10.243415)

*Publication date:*  
1993

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

[Link back to DTU Orbit](#)

*Citation (APA):*  
Jensen, J. A. (1993). Implementation of ultrasound time-domain cross-correlation blood velocity estimators. *I E E E Transactions on Biomedical Engineering*, 40(5), 468-474. <https://doi.org/10.1109/10.243415>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Implementation of Ultrasound Time-Domain Cross-Correlation Blood Velocity Estimators

Jørgen Arendt Jensen

**Abstract**—The implementation of real-time blood velocity estimators using time-domain cross-correlation is investigated. The basic algorithm is presented for doing stationary echo canceling, cross-correlation estimation and subsequent velocity estimation. Sampled data acquired at rates of approximately 20 MHz are used in the algorithm imposing a heavy burden on the signal processing hardware. The algorithm is analyzed with regard to the high sampling frequency, and a method for performing real-time high-speed data movement and cross-correlation is suggested. Implementation schemes based on using the sign of the data as well as the full precision are proposed. From an analysis of the process it is concluded that the sign data implementation can attain real-time processing. This can also be obtained for the full precision data, however, at the expense of using a number of dedicated signal processing chips. Both implementations suggested can handle the estimation of velocities for A-lines acquired from multiple directions.

## I. INTRODUCTION

THE study of the flow dynamics in the body has been greatly eased by the introduction of real-time Doppler ultrasound scanners. Using these scanners a map of the blood velocity in the body can be displayed at a rate of two to ten images a second thereby visualizing the dynamics of the flow. This makes it possible to diagnose heart valve deficiencies and locate possible stenosis of veins.

Most current scanners are based on an autocorrelation algorithm, which estimates the phase shift between successively acquired lines of ultrasound data [1]. A drawback of this is that aliasing can occur at high flow rates due to the use of a phase shift estimation [2]. A new approach using high frequency sampled data from the ultrasound transducer and time domain cross-correlation has, therefore, been suggested [3]–[5]. Here no algorithmic limitations are enforced on the maximum detectable velocity.

The use of high frequency data imposes a heavy burden on the electronics performing the signal processing, as sampling frequencies up to 20 and 30 MHz are used. This paper investigates how the algorithm can be implemented using readily available chips, and still attain real-time processing.

The basic algorithm is presented in the next section. It is explained how the data are divided into segments, how the cross-correlation function can be smoothed, and how to evaluate the resulting estimate. Section III then proceeds with an examination of how to structure the data conveniently for

the calculation. The section also elucidates why it often is advantageous to interleave processing of lines from different directions in order to scale the velocity range that can be estimated.

The simplest approach, when implementing the technique, is to use only the sign of the data. That this is indeed possible with only a minor reduction in performance was shown in [6]. The electronics for making one bit correlation is presented in Section IV. It is shown that real-time processing can be done using suitable data buffers and a single integrated circuit for performing the cross-correlation. The electronics can also handle multiple lines.

The use of full precision data is slightly more complicated as full precision multiplications must be performed very fast. It is shown in Section V that the one bit correlation chip can be substituted by a 32-stage multiplication chip from Immos, and that real-time performance then can be obtained, when a number of these chips are used.

## II. THE ALGORITHM

The estimation of the blood velocity by the time-domain cross-correlation approach is done by emitting pulses into the tissue and then correlating successively received signals from the ultrasound transducer. A series of pressure pulses is emitted into the tissue at a frequency of  $f_{prf}$ . The response received is high frequency sampled, and successive lines are cross-correlated. The position of the largest peak in the cross-correlation function indicates the time-shift between successive lines. The shift  $t_s$  is related to the velocity of the blood by [3]:

$$v_z = \frac{c}{2} \frac{t_s}{T_{prf}} \quad (1)$$

where  $v_z$  is velocity in the direction away from the transducer.  $c$  is the propagation velocity of the ultrasound, and  $T_{prf} = 1/f_{prf}$  is the time period between pulse emissions.

The signal received from the area under investigation is often dominated by echoes from stationary objects. These echoes must be removed in order to make the signal useful for calculating the cross-correlation. The removal is done by subtracting successive lines, so that only the difference from line to line is present. The consequence of doing this was studied in [7] and is mainly manifested as a reduction in signal-to-noise ratio.

The purpose of the estimation is to show a real-time image of the blood flow in the region under investigation. So the velocity as a function of depth in tissue should be displayed.

Manuscript received August 31, 1991; revised December 11, 1992.

The author was with the Electronics Institute, Technical University of Denmark, Lyngby, Denmark. He is now with the Bioacoustics Research Laboratory, University of Illinois, Urbana, IL 61801.

IEEE Log Number 9207665.

0018-9294/93\$03.00 © 1993 IEEE

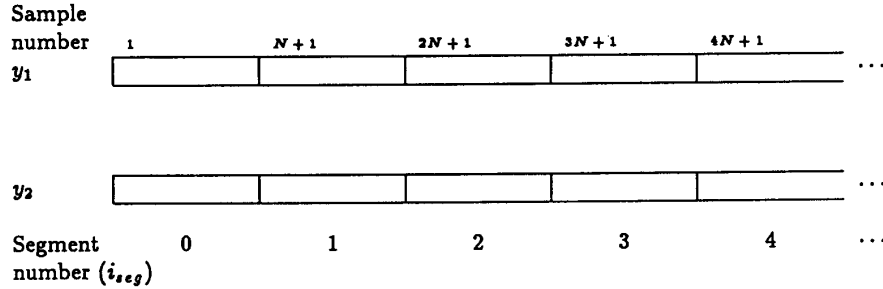


Fig. 1. Segmentation of A-lines.

This leads to a segmentation as shown in Fig. 1 of the RF lines into range gates, in which the velocity is determined.  $y$  denotes one measured line of high frequency data. The size of the range gates (number of samples) is a compromise between axial resolution and the possibility of determining the velocity. If a very small segment is used, erroneous results can arise as shown in [6]. Contrary, a large segment will make it difficult to study the spatial velocity variation in the tissue.

The discrete cross-correlation function of the segmented data is calculated by

$$\hat{R}'_{12d}(n, i_{\text{seg}}) = \frac{1}{N(N_{\text{ave}} - 2)} \cdot \sum_{i=1}^{N_{\text{ave}}-2} \sum_{k=0}^{N-1} [y_{i+1}(k + i_{\text{seg}}N) - y_i(k + i_{\text{seg}}N)] \cdot [y_{i+2}(k + n + i_{\text{seg}}N) - y_{i+1}(k + n + i_{\text{seg}}N)] \quad (2)$$

where  $k$  denotes sample number in a segment and  $i$  is line number.  $N$  is the number of samples in a segment,  $N_{\text{ave}}$  is the number of lines, and  $i_{\text{seg}}$  is the segment number in the line.  $n$  is the lag in the cross-correlation function. An equivalent implementation is

$$\hat{R}'_{12d}(n, i_{\text{seg}}) = \frac{1}{N(N_{\text{ave}} - 2)} \cdot \sum_{i=1}^{N_{\text{ave}}-2} \sum_{k=0}^{N-1} [y_{i+1}(k + n + i_{\text{seg}}N) - y_i(k + n + i_{\text{seg}}N)] [y_{i+2}(k + i_{\text{seg}}N) - y_{i+1}(k + i_{\text{seg}}N)] \quad (3)$$

which gives the same velocity estimates. In these expressions a number of lines is used. This is appropriate as long as the data acquisition time is small, so the velocity can be assumed constant. Then the cross-correlation function calculated from successive lines will have roughly the same shape, and can, thus, be added to obtain a smoother estimate. This essentially amounts to increasing the integration time used in the calculation and thereby reducing the influence from noise.

The span of lags in  $\hat{R}'_{12d}$  determines the maximum velocity that can be detected. A reasonable choice is to use the segments just before and after the current range gate, so  $n$  has a range

from  $-N$  to  $N$ . The largest detectable velocity then is:

$$v_{\text{max}} = \frac{c}{2} N \frac{f_{\text{prf}}}{f_s} \quad (4)$$

and the minimum velocity due to time quantization is

$$v_{\text{min}} = \frac{c}{2} \frac{f_{\text{prf}}}{f_s} \quad (5)$$

If all available time for data acquisition is used,  $f_s$  samples will be produced per second, where  $f_s$  is the sampling frequency. Using lines holding  $N_s$  segments then:

$$N_{\text{load}} = \frac{f_s}{N_s N N_{\text{ave}}} (N_{\text{ave}} - 2) N (2N + 1) (N_s - 1) = f_s \cdot \frac{N_{\text{ave}} - 2}{N_{\text{ave}}} \cdot \frac{N_s - 1}{N_2} \cdot (2N + 1) \quad (6)$$

multiplications and additions must be performed per second to calculate the estimates of the cross-correlation function. Using  $N = 32$ ,  $N_s = 100$ ,  $N_{\text{ave}} = 8$ , and  $f_2 = 20$  MHz results in

$$N_{\text{load}} = 20 \cdot 10^6 \cdot \frac{6}{8} \cdot \frac{99}{100} \cdot (2 \cdot 32 + 1) = 965.3 \cdot 10^6 \quad (7)$$

multiplications and additions per second. This clearly demonstrates the difficult task of making a real-time implementation.

The most expensive operation to perform is the multiplication of data values. To make this simple, an implementation using only the sign of the data has been suggested [4]. The signals involved are of the form:

$$y_s(k) = \text{sgn}[y(k)] = \begin{cases} 1 & \text{for } y(k) \geq 0 \\ -1 & \text{for } y(k) < 0. \end{cases} \quad (8)$$

The feasibility of this method was demonstrated in [6] and [7], where only a slight reduction in performance was found.

The cross-correlation velocity estimator is a nonlinear algorithm due to the detection of the location of the maximum. A small perturbation of the cross-correlation function can, therefore, lead to a detection of a wrong peak and, thus, a wrong velocity. The probability of correct detection depends on the measurement system used (transducer bandwidth), on the averaging employed (integration time and number of lines) and on the actual measurement location (velocity and signal-to-noise ratio) as shown in [6] and [7]. When using stationary echo canceling the probability gets very low at low velocities, as the signal-to-noise ratio goes to zero for the velocity going to zero [7]. As velocity and signal-to-noise ratio cannot be known ahead of time, it is of importance to get an indication

of how reliable the result at hand is. An indication of this can be found by [7]:

$$I_{rs} = \max\{\hat{R}_{12s}(n)\} \quad (9)$$

for the cross-correlation function estimate for the sign data calculated by

$$\hat{R}_{12s}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \text{sgn}[y_1(k)] \text{sgn}[y_2(k+n)] \quad (10)$$

$\max\{\cdot\}$  is the maximum value of the function. The index  $I_{rs}$  has a value between 0 and 1, and its mean value is correlated to the probability of correct detection. A mean value near one indicates a very reliable estimate. Smoothing of  $I_{rs}$  is a necessity as it fluctuates around its mean value. A useful recursive formula for calculating the mean of the index is

$$\begin{aligned} \hat{I}_{rm}(i) &= \lambda \hat{I}_{rm}(i-1) + (1-\lambda) \hat{I}_{rs}(i) \\ &= \hat{I}_{rs}(i) + \lambda(\hat{I}_{rm}(i-1) - \hat{I}_{rs}(i)). \end{aligned} \quad (11)$$

This will after 10 to 20 estimates of  $\hat{I}_{rs}$  yield a useful value for  $\lambda = 0.9$  [7].  $\hat{I}_{rm}$  can then be used in rejecting unreliable velocity estimates and used in displaying the reliability of the remaining velocity estimates by using color intensity modulation.

### III. IMPLEMENTATION ASPECTS

One of the biggest problems encountered, when implementing algorithms with very high sampling frequencies, is often that associated with the movement of data. Thus, it is of importance to find a calculation structure in which the flow of data is convenient. This section analyzes the algorithm with this in mind.

The central part of the algorithm is the calculation of the cross-correlation function for two successive lines. This is after stationary echo canceling calculated by

$$\hat{R}_{12d}(n, i_{\text{seg}}) = \frac{1}{N} \sum_{k=0}^{N-1} y_1(k+i_{\text{seg}}N) y_2(k+i_{\text{seg}}N+n). \quad (12)$$

The equation can be written as a vector product between the two column vectors  $Y_1(i_{\text{seg}}N)$  and  $Y_2(i_{\text{seg}}N)$  as

$$\hat{R}_{12d}(n, i_{\text{seg}}) = Y_1^t(i_{\text{seg}}N) \otimes Y_2(i_{\text{seg}}N+n). \quad (13)$$

$Y_x(i_{\text{seg}}N)$  denotes a column vector holding  $N$  elements starting with the sample having time index  $i_{\text{seg}}N$ .  $\otimes$  denotes vector multiplication and  $^t$  transpose. The division of the result by  $N$  is discarded, as it is a mere scaling of the result, which has no influence on the final velocity estimate. The cross-correlation function is then calculated for successive values of  $n$  in which a time shifted version of  $Y_2$  is used.  $q^1$  is the forward time shift operator, and by this the next  $Y_2$  can be expressed as

$$Y_2(n+1+i_{\text{seg}}N) = q^1 Y_2(n+i_{\text{seg}}N). \quad (14)$$

So

$$\hat{R}_{12d}(n, i_{\text{seg}}) = Y_1^t(i_{\text{seg}}N) \otimes q^n Y_2(i_{\text{seg}}N). \quad (15)$$

By symmetry we have from the previous section:

$$\hat{R}'_{12d}(n, i_{\text{seg}}) = q^n Y_1^t(i_{\text{seg}}N) \otimes Y_2(i_{\text{seg}}N) \quad (16)$$

which is equivalent to  $\hat{R}_{12d}(n, i_{\text{seg}})$ . When  $\hat{R}_{12d}(n, i_{\text{seg}})$  is calculated by running through the lags for  $n = -N$  to  $N$  the initial vectors are  $Y_1(i_{\text{seg}}N)$  and  $Y_2((i_{\text{seg}}-1)N)$  and the final vectors entering the calculation are  $Y_1(i_{\text{seg}}N)$  and  $Y_2((i_{\text{seg}}+1)N)$ . Calculating the cross-correlation by (16) for the next segment,  $Y_1(i_{\text{seg}}N)$  and  $Y_2((i_{\text{seg}}+1)N)$  must be used initially and the final vectors are  $Y_1((i_{\text{seg}}+2)N)$  and  $Y_2((i_{\text{seg}}+1)N)$ , which can be used directly in (15). So the estimates of  $R_{12}$  for the two lines can be calculated by alternating between (15) and (16).

The signal  $y_2$  is measured  $T_{prf}$  seconds after  $y_1 \cdot N \cdot N_s$  samples are acquired, and the delay between the two signals is, therefore, perceived as  $N \cdot N_s$  sample periods after acquisition. So just delaying the received, sampled signal from the transducer  $N \cdot N_s$  sample periods will yield  $y_1$ .

The resolution of the estimated velocity is given by (5), and is 0.12 m/s at  $c = 1540$  m/s,  $f_{prf} = 3.2$  kHz, and  $f_s = 20$  MHz. This is often too coarse a resolution for low velocities and  $f_{prf}$  should, therefore, be lowered, which also has the advantage that the echo canceling does not cause such a large reduction in signal-to-noise ratio (see [7]). The lowering of  $f_{prf}$  does, however, also waste valuable time, which could be used for data acquisition. Therefore,  $f_{prf}$  should be kept at a high value, and lines in different directions should be acquired. Measuring data interleaved from  $N_d$  directions would give a presumed pulse repetition frequency of  $f_{prf}/N_d$  and a velocity resolution of  $(c/2)(f_{prf}/N_d f_s)$ , but still use the time efficiently. The only change to the signal processing used is that the delay between  $y_1$  and  $y_2$  would now be  $N_d N N_s$  sample

The cross-correlation function estimate is averaged by using a number of lines. For each pair of lines  $(N_s - 1)$  cross-correlation functions are calculated yielding  $(2N + 1)(N_s - 1)$  correlation values. Delaying the output from the vector multiplier by this number of sample periods will give the previous correlation coefficient, which can be added to the newly calculated coefficient.

The reliability index given by (11) is smoothed on an image by image basis. Each image holds  $(N_s - 1)L$  velocity estimates, when  $L$  is the number of lines in an image. As before the output from the reliability index calculation must be delayed  $(N_s - 1)L$  calculation periods in order to be used in smoothing the reliability index.

### IV. IMPLEMENTATION USING ONE-BIT DATA

The central part of the algorithm is the multiplication of two vectors, which must be made at a rate of twice the sampling frequency. A multiplication of the single elements is facilitated by using only the sign of the data as mentioned in Section II. If 0 denotes negative numbers and 1 positive numbers, then an inverted Boolean exclusive OR operation will yield the correct multiplication result. The result of the vector multiplication is then found by summing the multiplied values, subtracting  $N/2$  and multiplying by 2. This is the operation performed by

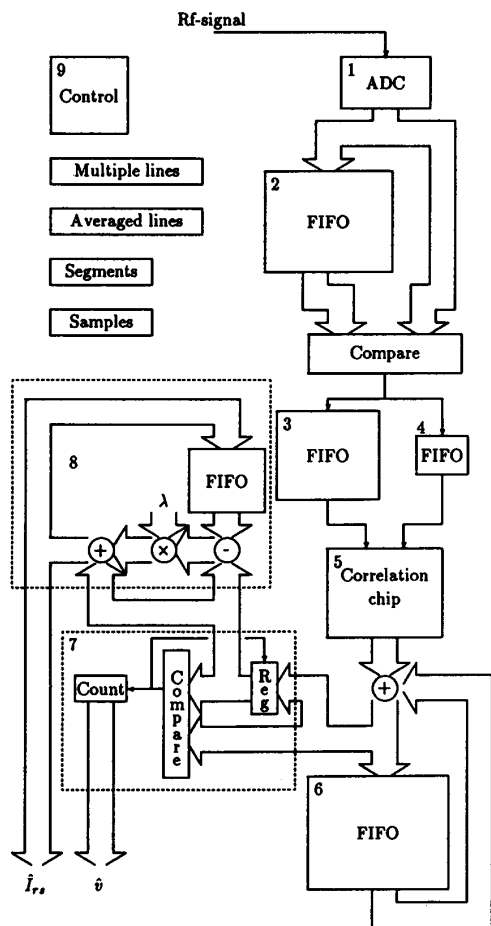


Fig. 2. Implementation of the time-domain cross-correlation estimator using sign data.

the TMC1023 integrated circuit from TRW [8]. It can perform multiplications at a rate of 30 MHz on vectors having up to 64 elements of one bit, so real-time operation can be made. The circuit holds two  $64 \times 1$  bit shift registers, where data can be shifted at a rate of 30 MHz. A vector product is produced for each clock period for the part of the registers that are enabled.

The delay operation can conveniently be performed by FIFO (first in first out) buffers, in which data can be read and written independently. A full implementation of the algorithm using FIFO's and the correlation chip are shown in Fig. 2.

A more detailed description of the electronic components shown in Fig. 2 is given below.

- 1) Analog to digital converter (ADC) converting the analog RF transducer signal to 12 bits samples at a rate of  $f_s$ . 12 bits are used due to the high dynamic range of clinical ultrasound signals [9].
- 2) FIFO buffer holding samples for the last line measured. This introduces a delay of  $N \cdot N_s$  sample periods, so echo canceling is performed by subtracting the current sample from the output sample of the FIFO. As only the sign is used, a compare operation need only be made.

The FIFO must hold at least  $N \cdot N_s \times 12$  bits, and the access time for read and write operations must be less than  $1/f_s$ . The size of the FIFO should be increased to  $N \cdot N_s \cdot N_d \times 12$  bits if multiple lines are used.

- 3) FIFO buffer holding  $N \cdot N_s \cdot N_d \times 1$  bit samples, so the signal is delayed before being cross-correlated by the signal for the next line.
- 4) Small FIFO buffer holding  $2 \cdot N \times 1$  bit to compensate for the asynchronous processing. During one of the two calculations by (15) and (16), no data is shifted out from this FIFO into the correlation chip. Data must, therefore, be accumulated for subsequent processing.
- 5) TMC2023 chip from TRW for calculating the vector product. A cross-correlation product is produced between a selected part of the shift registers at a rate of 30 MHz. As  $2N + 1$  correlation products must be made for each  $N$  samples, the maximum sampling frequency is currently limited to 15 MHz. However, if the span of lags in  $\hat{R}_{12s}$  is limited to  $-N/2 : N/2$ , a sampling rate of roughly 30 MHz can be maintained. The result from the correlation consists of  $D_x = 1 + \log_2 N$  bits.
- 6) FIFO buffer for the cross-correlation estimates. The cross-correlation process for one pair of lines yields  $(2N + 1)N_s$  correlation values, which are stored in this FIFO. It delays the values, so addition to the next correlation value for the next pair of lines can be done. The wordlength of the values in the FIFO is  $1 + \log_2(N(N_{ave} - 2))$  bits. The size of the FIFO should be increased to  $(2N + 1) \cdot N_s \cdot N_d$  if multiple lines are used.
- 7) Electronics for finding the result. The velocity estimated is found by running through all the values of the resulting smoothed cross-correlation function and finding the index of the maximum value. This can be done when the last pair of lines are correlated. A register (**Reg**) holds the maximum value, and **count** keeps track of its position in the cross-correlation function.
- 8) FIFO buffer and electronics for calculating the reliability index. The maximum correlation value is held in a register and used for calculating the reliability index given by (11). The FIFO holds  $(N_s - 1)L$  values, and the calculation is carried out at a rate of  $f_s/(N_s N_{ave})$ . This amounts to 78.1 kHz for  $N = 32$ ,  $f_s = 20$  MHz and  $N_{ave} = 8$ . At this rate a signal processing chip could do the whole signal processing, and give flexible choices for using the resulting index.
- 9) Control logic for managing the system. The basis of this is registers that count the number of samples, segments and lines processed. The control can conveniently be divided into subcontrol structures keeping track of samples, segments, and lines.

It should be noted that this implementation can also handle the acquisition of interleaved lines. Only an addition of a register that counts the interleaved lines and a slight change of the control logic is needed.

The implementation suggested here relies heavily on the use of FIFO buffers. This not only makes the solution simple, but also offers the advantage that the different parts of the

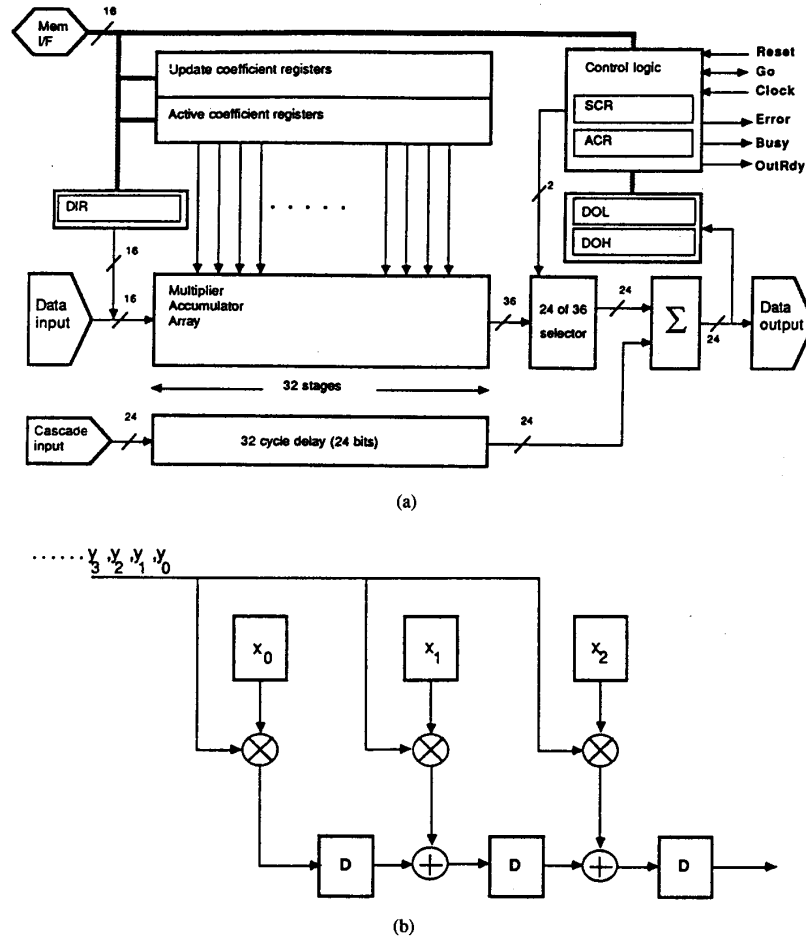


Fig. 3. Basic components of the Inmos A100 vector multiplier. (Drawings from Inmos technical data).

implementation are independent. Three processing stages are present: Data acquisition, data buffering after echo canceling, and cross-correlation. If the FIFO buffers are not saturated with data, a simple handshake operation need only take place after each set of  $N_{ave} \cdot N_d$  lines. Then all available time can be used for calculation. The chip is capable of performing  $32 \cdot 30 \cdot 10^6 = 960 \cdot 10^6$  multiplications and additions per second. Using this to its fullest, a real-time implementation up to  $f_s = 20$  MHz can be made with a single TMC2023 chip [see (7)].

#### V. IMPLEMENTATION USING FULL PRECISION DATA

A slight increase in the probability of correct detection can be obtained, when the cross-correlation function is calculated using the full precision of the data [6], [7]. 12 bits multiplication must then be carried out. A chip capable of full vector multiplication must then be used. A possible choice is to use the Inmos IMS A100 chip [10], shown in Fig. 3.

This chip contains 32 multipliers capable of either 4, 8, 12, or 16 bits multiplication. The contents of a coefficient register is multiplied with a data input value. The output from the multipliers is summed successively to form an output after the

32 multiplications, which is the sum of the coefficient register multiplied by the last 32 data values presented to the chip. The contents of the coefficient register can be swapped by a second coefficient register. The register not in use during multiplication can be loaded with new coefficients.

With these facilities the continuous cross-correlation can be made by using the following method. First, a segment of the first line is loaded into the coefficient register, and data is then presented to the chip one at a time. After 32 data values, correct results will appear.  $2N$  values are then presented and the correlation coefficients are calculated. The last  $N$  data values are also stored in the second coefficient register. After the last correlation coefficient the system holds intermediate results from the correlation of segment 1 and C as seen in Fig. 4, and the second coefficient register holds the samples of segment C. Swapping the coefficient registers and presenting consecutive sample values for segment 2 will then start the calculation for the next range gate.  $2N$  correlation coefficients need only be calculated as the first correlation coefficient is already calculated from the previous range gate. After the calculation, data values for segment 3 are swapped into the coefficient register. A continuous correlation can be performed

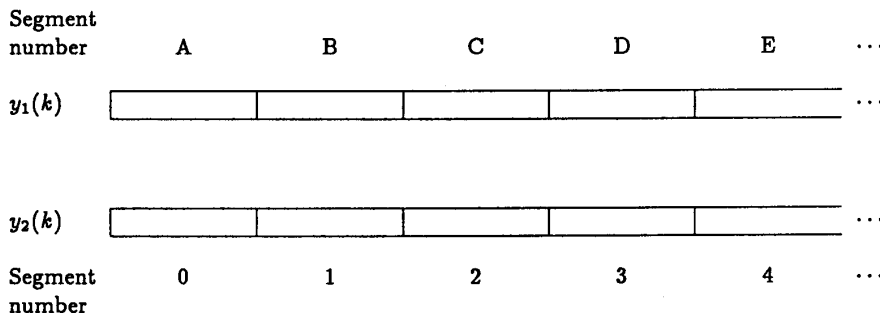


Fig. 4. Use of the segments in the calculation.

by this swapping scheme, and the IMS A100 can replace the TMC2023 correlation chip in Fig. 2 directly.

12 bits multiplication should be used, and this currently limits the data rate of the A100 to 3.3 MHz. Multiple chips can, however, be used by inserting small FIFO buffers before and after the chip. A real-time implementation can then be reached if the span of lags is restricted to  $-N/2 : N/2$  and four chips used in parallel, when all available calculation time is used.

Another method for increasing signal processing speed is to ensure proper scaling of the acquired data after echo canceling. 8 bits data would then be sufficient to digitize the signal, which has a signal-to-noise ratio of 0-30 dB. The use of only 8 bits data would increase the data rate to 5 MHz, so four chips can attain real-time processing for the full span of lags.

The calculation of the reliability index is done by [7]:

$$I_{rd} = \frac{\max\{\hat{R}_{12}(n)\}}{\sqrt{\hat{P}_1 \hat{P}_2}}$$

$$\hat{P}_x = \frac{1}{N} \sum_{k=0}^{N-1} y_x^2(k), \quad x = 1, 2 \quad (17)$$

where  $\hat{P}_x$  is the power in the current range gate. As  $y_1$  and  $y_2$  are time shifted versions of each other and have the same power,  $\sqrt{\hat{P}_1 \hat{P}_2}$  can be replaced by  $\hat{P}_1$ . Thus, it is only necessary to calculate the power in one of the segments. This can be done by a simple multiplier-accumulator chip, which receives the same data presented to the cross-correlation chip. Using  $2N$  values, the index would be

$$I_{rd} = \frac{\max\{\hat{R}_{12}(n)\}}{\frac{1}{2N} \sum_{k=0}^{2N-1} y_1^2(k)} \quad (18)$$

The division could be made by a look-up table, and the smoothing operation done by the same electronics as for the sign reliability index.

### VI. CONCLUSION

The use of ultrasound scanners for the study of blood flow in the body has steadily increased during these last years. An important part of this evolution stems from the improvement of the equipment by providing better algorithms

for velocity estimation. A useful step in this direction appears to be the introduction of the time-domain cross-correlation algorithm, which seems to solve the velocity-depth ambiguity. The algorithm uses high frequency sampled data making the implementation a challenge. It was shown how the task can be solved by either using the sign of the data or by using multiple, dedicated vector multiplier chips. Both implementations use FIFO buffers for data storage, making it possible to use the full potential of the multiplication chips. The use of FIFO buffers also makes it possible to acquire data and estimate velocities for lines from different directions, which can be advantageous for reducing the degradation in signal-to-noise ratio introduced by the stationary echo canceling.

### ACKNOWLEDGMENT

The author would like to express his thanks to Dr. J. E. Wilhjelm, and Associate Professor Dr. S. D. Hansen for valuable discussions and help during the preparation of the paper.

### REFERENCES

- [1] C. Kasai, K. Namekawa, A. Koyano, and R. Omoto, "Real-time two-dimensional blood flow imaging using an autocorrelation technique," *IEEE Trans. Sonics. Ultrason.*, vol. SU-32, no. 3, pp. 458-463, May 1985.
- [2] L. Hatle and B. Angelsen, *Doppler Ultrasound in Cardiology, Physical Principles and Clinical Applications*, 2nd ed. Philadelphia, PA: Lea & Febiger, 1985.
- [3] O. Bonnefous and P. Pesqué, "Time domain formulation of pulse-doppler ultrasound and blood velocity estimation by cross correlation," *Ultrason. Imag.*, vol. 8, pp. 73-85, 1986.
- [4] O. Bonnefous, P. Pesqué, and X. Bernard, "A new velocity estimator for color flow mapping," in *IEEE Ultrason. Symp.*, pp. 855-860, 1986.
- [5] S. G. Foster, P. M. Embree, and W. D. O'Brien, "Flow velocity profile via time-domain correlation: Error analysis and computer simulation," *IEEE Trans. Ultrason., Ferroelect. Freq. Contr.*, vol. 37, no. 2, pp. 164-175, May 1990.
- [6] J. A. Jensen, "Artifacts in velocity estimation using ultrasound and cross-correlation," submitted for publication.
- [7] ———, "Stationary echo canceling in velocity estimation by time-domain cross-correlation," to be published in *IEEE Trans. Med. Imaging*.
- [8] TRW Databook, Data Converters and DSP Products, TRW LSI Products Inc., 1991.
- [9] J. A. Jensen and J. Mathorne, "A sampling system for clinical ultrasound images," in *Medical Imaging V: Image Capture, Formatting, and Display*, Y. Kim, Ed., *Proc. SPIE*, Volume 1444, pp. 221-231, 1991.
- [10] Inmos: IMS A100 Cascadable signal processor, Inmos Technical Data, Apr. 1987.



**Jørgen Arendt Jensen** was born in Roskilde, Denmark, in 1960. He received the M.Sc. degree in electrical engineering in 1985 and the Ph.D. degree in 1989 from the Technical University of Denmark. The subject of his Ph.D. was digital signal processing of medical ultrasound images.

He was a postdoctoral fellow at the Electronics Institute at the Technical University of Denmark from 1989 to 1992, and has been a visiting scientist at King's College, UK, Duke University, NC, and Stanford University, CA. He currently holds a postdoctoral fellowship at the Bioacoustics Research Laboratory at the University of Illinois at Urbana-Champaign, and has published a number of reports and papers on digital signal processing, ultrasound acoustics, and medical ultrasound imaging.