PARAMETRIC SPECTRAL ESTIMATION ON A SINGLE FPGA

S. Bellis, W. Marnane

National Microelectronics Research Centre
University College Cork
Cork, Ireland
sbellis@nmrc.ucc.ie, liam@rennes.ucc.ie

P. Fish

School of Electronic Engineering
& Computer Systems, University of Wales
Bangor, Wales, UK
fish@sees.bangor.ac.uk

ABSTRACT

Parametric, model based, spectral estimation techniques can offer increased frequency resolution over conventional short-term fast Fourier transform methods, overcoming limitations caused by the windowing of sampled, time domain, input data. However, parametric techniques are significantly more computationally demanding than the Fourier based methods and require a wider range of arithmetic functionality; for example, operations such as division and square-root are often necessary. These arithmetic processes exhibit communication bottleneck and their hardware implementation can be inefficient when used in conjunction with multipliers. A programmable, bit-serial, multiplier/divider, which overcomes the bottleneck problems by using a data interleaving scheme, is introduced in this paper. This interleaved processor is used to show how the parametric Modified Covariance spectral estimator can be efficiently routed on a field programmable gate array for real-time applications.

1. INTRODUCTION

Due to its ease of hardware and software implementation the short-term fast Fourier transform (STFFT) is widely used for spectral estimation and is known as the conventional method. However, the technique has drawbacks in terms of spectral resolution and accuracy caused by the finite length of the input data sequence used. Windowing of input data causes spectral broadening and Gibb's phenomenon of spectral leakage can mask the weaker frequency components of the true power spectral density (PSD) [1]. These unwanted effects can be reduced by using longer data sequence lengths, so that the transformed signal becomes a better representation of the infinite data sequence, but in real life this usually is not feasible as the characteristics of the input data may change with time. Over short periods of time the data signals can often be assumed to exhibit wide-sense stationarity, where the signal characteristics are assumed approximately constant but the spectral resolution is therefore limited. In attempts to improve the PSD estimation, windowing functions, Bartlett or Hanning for example, can be used to reduce side-lobe levels but these lower spectral resolution by broadening the main lobe of the PSD [2].

Model based, parametric spectral estimation techniques can alternatively be used, where the unrealistic assumption that data is zero outside the window of interest is dropped [1]. Either knowledge of the underlying process or reasonable assumptions about the nature of the unobserved data are used to improve frequency resolution over the conventional approaches. The computational burden of such processors is however much higher than the STFFT and arithmetic functions such as division and square-root often become necessary. In the division and square-root non-restoring algorithms there is an inherent dependency that the result bits must be computed in a most significant bit (MSB) first manner, with the computation of a bit directly dependent upon the result of the previous one [3]. This interdependency makes it difficult to efficiently realize such arithmetic functions in hardware, and implementations are usually much slower than other basic functions such as multiplication, addition and subtraction. Communication bottlenecks can therefore easily occur in systolic arrays where different types of processors are interconnected.

The difficulties with hardware implementation of parametric spectral estimators have led to a preference of software implementation on homogeneous DSP networks [4]. However, high levels of processing capacity have not been fully reflected in system throughput since the increased communication incurred as a result of parallelism is constrained by communication bus performance. This restricts the range of problems that can be computed in real-time and the software approach may sometimes be inadequate for real-time spectral estimation.

In this paper, hardware implementation of a parametric spectral estimator is addressed. A bit-serial processor capable of division and inner product step computation is developed by combining separate processors for these functions. The design uses a high level of pipelining so that division can be computed at a high rate and multiplication is performed on a MSB first data stream, eliminating the bottleneck problem. The high level of pipelining allows many independent computations to be performed simultaneously or interleaved. The use of the interleaving scheme is demonstrated by implementing the design of a Modified Covariance type of parametric spectral estimator, to produce a field programmable gate array (FPGA) based system for the spectral analysis of Doppler signals from ultrasonic blood flow detectors.

2. MODIFIED COVARIANCE SPECTRAL ESTIMATION

The model order $p = 4$ Modified Covariance (MC) spectral estimator, proven to be optimally cost efficient for the blood flow application where mean velocity and flow disturbance are of interest [5], involves solving the following linear system of covariance matrix equations:

$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \end{bmatrix} = - \begin{bmatrix} c_{1,0} \\ c_{2,0} \\ c_{2,0} \\ c_{4,0} \end{bmatrix} \quad (1)$$

where each element $c_{i,j}$ is obtained from:

$$c_{i,j} = \frac{1}{2(N-p)} \left(\sum_{n=p}^{N-1} x_{n-i}.x_{n-j} + \sum_{n=0}^{N-1-p} x_{n+i}.x_{n+j} \right) \quad (2)$$

for a window of length N data samples. The $\hat{a}_k$ filter parameter estimates are obtained by solution of the linear system (1), using the Cholesky, forward elimination and back substitution algorithms. The signal white noise variance estimate, $\hat{\sigma}^2$, is calculated as:

$$\hat{\sigma}^2 \; = \; c_{0,0} \; + \; \sum_{k=1}^{p} \hat{a}_k \; . \; c_{0,k} \quad (3)$$

and the power spectral density (PSD), $\hat{P}_{MC}(f_n)$, is obtained from:

$$\hat{P}_{MC}(f_n) = \frac{\hat{\sigma}^2}{|A(f_n)|^2} = \frac{\hat{\sigma}^2}{\left| 1 + \sum_{k=1}^{p} a_k z^{-k} \big|_{z=e^{j2\pi f_n}} \right|^2} \quad (4)$$

Hence, the MC spectral estimator may be partitioned onto four different programming modules:

- CMR - calculation of the elements of the covariance matrix and right-hand side vector, $5N$ multiply accumulates taking into account matrix symmetry,
- Cholesky - solution of the linear system of equations, 6 divisions and 10 inner step products for non-square-root Cholesky, 4 divisions and 12 inner step products for solving triangular systems,
- WNV - calculation of the white noise variance, 4 multiply accumulates,
- PSD - computation of the power spectral density, $4N$ inner step products for a zero padded DFT, N multiplications to find absolute value of DFT and $N/2$ divisions for the PSD.

The number of samples N, over the fixed time duration window of 10ms, is required to be either 64, 128, 256 or 512 depending on Doppler signal conditions. Implementation of the algorithm in Matlab software proved to be in excess of a factor of 10^3 times too slow for real-time operation and that a performance of up to 13.5 MFLOPS/s is required [4]. Execution times of MC algorithm implementation using various topologies of Texas Instruments TMS320C40 DSPs with T8 transputers as routers have also fallen short of the real-time requirements, where processing time is over 150ms too long in the worst case [4][6]. Use of a single DSP in a PC hosted system has been shown sufficient for the smaller N but the specification of $N = 512$ could not be achieved [4], thus prompting consideration of the hardware approach.

3. BIT-SERIAL INTERLEAVED PROCESSOR

Study of word-parallel systolic implementations of the MC method has shown the method to provide more than adequate throughput for the specified real-time blood flow application but the cost of such a system is very high in terms of arithmetic units, communication burden and control complexity [7]. For example, a systolic array processor for non-square root Cholesky decomposition [8] requires 13 processing elements (PEs), each PE having 2 to 6 ports of either m (single precision) or $2m$ (double precision) lines, and control is necessary to reverse data streams before back substitution. An alternative way to approach the hardware design involves consideration of bit-serial processing techniques.

The nature of multiplication algorithms normally involve the computation of least significant bits (LSBs) first and bit-serial multipliers reflect this in their output ordering. Conversely, division algorithms such as non-restoring are MSB first in nature [9]. Computation of each quotient bit can be performed from m controlled add subtract (CAS) operations, the decision on whether to add or subtract being taken given the result of the previous bit computed (except on the first operation where the signs of the input operands are used to decide). Allowing carries to ripple through therefore leads to a propagation delay greater than m CAS cells. In a bit-serial multiplier, the delay between successive bits being output is likely to be around a single full adder (FA) delay, leading to a maximum clock frequency approximately m times higher and a communication bottleneck with the divider. The clock rate of the divider can be increased to a similar maximum rate as the multiplier by pipelining the carries in each individual CAS stage. However, this means that each output bit is then available only once in every m clock cycles. There is also the problem that data streams must be reversed between multipliers and dividers. One possibility is to use registers and extra control logic to reorder the bit stream from the divider but the operation time is still limited.

The efficiency of the divider with the pipelined carry can be greatly improved by using the redundant slots between the output of successive bits to perform other separate divisions. The bit-serial/word-parallel divider shown in [3] allows $m+1$ individual divisions to be performed simultaneously or interleaved. This decreases the mean division operation time to achieve similar performance to a bit-serial multiplier but there is still the problem of data stream matching when interfacing such devices. One way to tackle this problem is to redesign the multiplier so that it works on a MSB first data stream, rather than storing and reordering the divider outputs which increases latency and control requirement [10]. MSB first multiplication, first demonstrated by McCanny *et al.* [11], shows it is possible to perform multiplication on positive numbers by summing partial products (PPs) in reverse order to the norm. This also requires inclusion of an MSB first addition unit to ensure that output carries from the PPs are added into the final product. Larsson-Edefors and Marnane [12], extend the concept of MSB first multiplication to the two's complement number system and show bit-serial architectures for this application. In order to match the divider bit-streams exactly to the multiplier bit-streams it is then just a matter of inserting extra delays along the FA sum pipeline so that the addition of PPs from a number of different multiplications can be performed simultaneously as shown by Bellis *et al.* [13].

Study of the bit-serial interleaved divider and multiplier reveals that both architectures show a large degree of similarity. Both work in load/operational phases; the loading networks for the divisor and multiplier both consist of $m+1$ delay feedback SISO registers and the FA sum/carry pipelines are alike. Both designs also require MSB first, half adder (HA) cell, addition stages; the divider requires m PEs, for 1's complement error correction which occurs for negative dividends, and the multiplier requires $m-1$ HA PEs to add the output carries from the PPs. Therefore, it is possible to combine the two designs to make a programmable bit-serial device which allows $m+1$ computations to be simultaneously interleaved, as shown in figure 1.

The processor has two mode selection inputs $DIVi$ and $SUBi$,

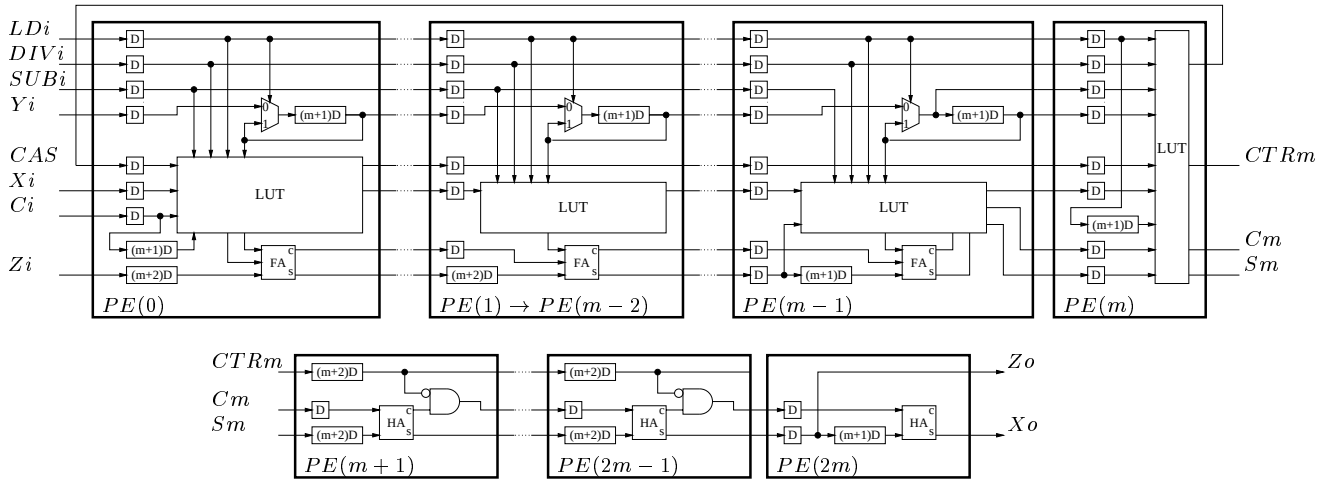

Figure 1: Bit-serial interleaved divider/inner product step architecture

which control four modes of operation $Xo = \pm Zi/Yi$ or $Zo = Zi \pm Xi.Yi$ where Zi and Zo are both double precision. LDi is the load/operational mode select signal for the storage of Yi and Zi over the first $m(m+1)$ clock cycles. LDi switches into operational mode over the next $m(m+1)$ clock cycles where the remaining data is input and the bulk of the computation is performed in the FA array. All control signals are fully pipelined similarly to the data, allowing the shortest possible block pipeline period of $2m(m+1)$ clock cycles and continuous input/output of data (i.e. while one block set of $m+1$ computations are being output, the next block set may be loaded in). The pipeline also allows independent functionality between each of the separate interleaves and on the same interleave a division may immediately follow an inner step product computation and vice-versa.

4. INTERLEAVED PROCESSOR BASED MODIFIED COVARIANCE SYSTEM

Cost-benefit analysis on systolic array implementation of the CMR and Cholesky sections of the MC spectral estimator shows that a 12 bit fixed point word-length is sufficient for these computations [7]. Using the bit-serial processor with a 12 bit word-length results in the capacity for interleaving 13 computations.

On interleaves 0 to 4 the CMR multiplications are performed over N consecutive block sets, such that the products $x_n.x_{n+i}$ are produced on interleave i ($0 \leq i \leq 4$) and block set n ($0 \leq n \leq N-1$). A bit-serial systolic array provides the correct input data sequencing from consecutive Doppler signal samples and a separate MSB first double precision accumulator, whose architecture is similar to that of HA section in figure 1, computes the covariance matrix elements, which are then stored in RAM. The system for computing the CMR calculation is shown in figure 2.

The entire Cholesky, forward elimination, back substitution and WNV computations are performed on interleave 5 on the system shown in figure 3. Here division and inner product step computation are necessary. Once the covariance matrix elements are stored in the dual port RAM after block set N the Cholesky decomposition can commence on interleave 5 while in parallel the CMR computation on the next set of data can be processed on interleaves 0 to 4. A ROM block controls the addressing of the dual port RAM for retrieval of stored data to go onto the processor inputs and storage of the processor results. To achieve good dynamic resolution for the low word-length used, a systolic array

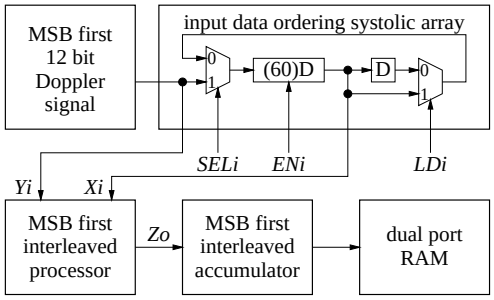

Figure 2: CMR computation on interleaves 0 to 4.

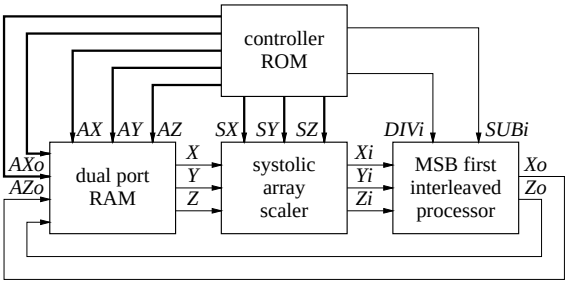

Figure 3: Cholesky and WNV computation on interleave 5.

scaling module is included between the RAM and the processor, whose scaling factors are also produced by the ROM controller along with the mode control. Overall timing in the system is controlled by three counters, qi (range 0 to 12), qb (range 0 to 23) and qw (range 0 to N) corresponding to the interleaves, bit-position and input word.

A zero padded N point DFT is computed on interleaves 6, 7, 8 and 9. This is basically a matrix vector multiplication and is computed by using the processor in inner product step mode. The system for this section consists of a ROM to provide storage of the twiddle factor matrix W_N, another ROM to control the addressing of the twiddle factors for a particular qw and 4 registers which continuously recirculate the filter parameter results ($\hat{a}_n$) from the Cholesky decomposition stage. On interleave 6 the real and imaginary parts ($N/2$ of each) of the first set of products $W_N^i.\hat{a}_1$ are alternately formed. Using a single flip-flop delay the results of these computations are then fed back into the Zi input of the interleaved processor to be added to the products $W_N^i.\hat{a}_2$ and the DFT is built up in this way. The dynamic range of the PSD computation is quite high compared to the rest of the system, therefore, at this stage a floating point representation of the DFT results is taken using a systolic based conversion circuit. PIPO registers are used to store the 6 bit exponents of the real and imaginary parts of the DFT, whose squares are computed on interleave 10. On interleave 11 the absolute value of the DFT is computed. The maximum of each pair of real and imaginary results from interleave 10 is fed to the Z_i input while the other value is piped into the Y_i to be appropriately scaled by the difference in the two squared exponents appearing on the X_i input. The PSD is then computed on interleave 12, involving $N/2$ divisions of the WNV formed on interleave 5 with the absolute values from interleave 11. The exponents of the PSD are then easily derived from the exponents of the DFT results.

5. CONCLUSION

This paper has proposed a bit-serial interleaved processor which can be programmed for use in division or inner product step computations. The interleaving idea was introduced in order to perform bit-serial division at the same high clock rate as multiplication without resorting to carry look-ahead schemes to remove the communication bottleneck. The result is a high throughput processor which is cost efficient in terms of VLSI implementation, since communication between PEs in the linear array is localised and control is very simple. An application in parametric spectral estimation, namely implementation of the Modified Covariance spectral estimator, which makes full use of the interleaving scheme, was described. This system has been programmed and simulated using VHDL. Synthesis was targeted to exploit the resources of a Xilinx XC4036EX-2 FPGA. This type of FPGA has dual port RAM capability, where a 16x1 bit dual port RAM can be implemented in a single configurable logic block (CLB). A dual port RAM cell is an area efficient method to implement a 13 or 14 bit SISO register, as used in the interleaving process. Such registers would otherwise have to be implemented using the pairs of flip-flops in each CLB, i.e. 7 CLBs. CLBs can also be configured as ROM blocks which are useful for generating the address signals in the Cholesky and PSD modules, and for storage of the DFT twiddle factors. The processor design exhibits mostly localised communication to make use of the fast routing resources between nearest neighbours in the FPGA's CLB matrix and enable high speed operation. Timing analysis of the FPGA layout shows that the maximum processor clock frequency of 35MHz allows real-time spectral estimation to be performed for the specified constraints. The re-programmable aspect of the FPGA is also useful; rather than designing control logic to switch between the different values of N, which uses resources and is likely to slow clock speed, a different bit-stream can be downloaded for each N. This idea can also be extended for changing to higher model order estimations where otherwise it would be difficult to parameterise p in such a system.

6. REFERENCES

[1] S. M. Kay, *Modern Spectral Estimation - Theory & Application.* Prentice Hall, 1988.

[2] M. Kassam, K. W. Johnston, and R. S. C. Cobbold, "Quantitative estimation of spectral broadening for the diagnosis of carotid arterial disease: Method and in vitro results," *Ultrasound in Medicine and Biology*, vol. 11, pp. 425–433, 1985.

[3] W. P. Marnane, S. J. Bellis, and P. Larsson-Edefors, "Bit-serial interleaved high-speed division," *Electronics Letters*, vol. 33, pp. 1124–1125, June 1997.

[4] M. M. Madeira, S. J. Bellis, M. G. Ruano, and W. P. Marnane, "Configurable processing for real-time spectral estimation," in *Preprints of AARTC98*, pp. 209–214, 1998.

[5] M. G. Ruano and P. J. Fish, "Cost/benefit criterion for selection of pulsed Doppler ultrasound spectral mean frequency and bandwidth estimation," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 12, pp. 1338–1341, 1993.

[6] M. G. Ruano, D. F. G. Nocetti, P. J. Fish, and P. J. Fleming, "Alternative parallel implementations of an AR-modified covariance spectral estimator for diagnostic ultrasound blood-flow studies," *Parallel Computing*, vol. 19, no. 4, pp. 463–476, 1993.

[7] S. J. Bellis, P. J. Fish, and W. P. Marnane, "Optimal systolic arrays for real-time implementation of the Modified Covariance spectral estimator," *Parallel Algorithms and Applications*, vol. 11, no. 1-2, pp. 71–96, 1997.

[8] S. J. Bellis, W. P. Marnane, and P. J. Fish, "Alternative systolic array for non-square-root Cholesky decomposition," *IEE Proceedings: Computers and Digital Techniques*, vol. 144, pp. 57–64, Mar. 1997.

[9] K. Hwang, *Computer Arithmetic: Principles Architecture and Design.* John Wiley & Sons, 1979.

[10] A. E. Bashagha and M. K. Ibrahim, "Radix digit-serial pipelined divider/square-root architecture," *IEE Proceedings on Computers and Digital Techniques*, vol. 141, no. 6, pp. 375–380, 1994.

[11] J. V. McCanny, J. G. McWhirter, and S.-Y. Kung, "The use of data dependence graphs in the design of bit-level systolic arrays," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 38, pp. 787–793, May 1990.

[12] P. Larsson-Edefors and W. P. Marnane, "A most-significant-bit-first serial/parallel multiplier," *IEE Proceedings: Circuits, Devices and Systems*, vol. 145, no. 4, p. 278 284, 1998.

[13] S. J. Bellis, W. P. Marnane, and P. Larsson-Edefors, "Bit-serial, MSB first processing units," *International Journal of Electronics - In press*, 1998.