signal. The input and output to the WDM couplers were preceded by polarisation insensitive isolators. The input and output SMF-28 fibre was connected by ST/PC connectors. All power and gain values quoted are system values, those measured at the input and output connectors.

System performance: An output power against system gain plot is shown in Fig. 3 for a 1535 nm input signal. A maximum power of +24·6 dBm (290 mW) has been obtained for a +4 dBm input signal at 1535 nm. A maximum small signal of 51 dB has been obtained with a −30 dBm 1535 nm input signal. It can be seen that at an output power of +20 dBm, the gain is only compressed to 47 dB.

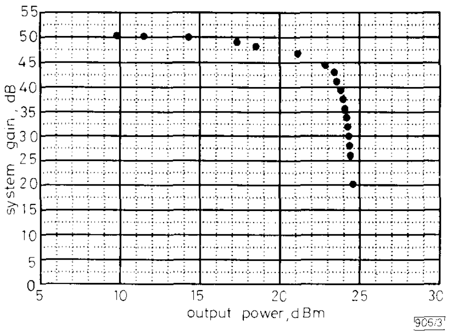

Fig. 3 *Output power against gain for single DPL-pumped optical amplifier*

To test the power scaling of this codoped amplifier, we pumped with a flashlamp-pumped CW Nd : YAG laser in the counter-propagating direction while pumping in the copropagating direction with an Nd : YLF DPL. We have observed up to +27 dBm of output power for just under 1500 mW of coupled pump power, as shown in Fig. 4. The input signal was +2 dBm at 1542 nm. From the single DPL result of Fig. 2, it can be seen that by the use of two Nd : YLF DPLs in a bidirectional pumping configuration, +27 dBm of output power could be readily obtained in a diode-based system. The system noise figure of the Er/Yb amplifier is shown in Fig. 5, at each of 40 CATV channel frequencies. The average value of the noise figure was found to be 3·5 dB, indicating a high degree of inversion. When corrected for input coupling losses, a quantum-limited noise figure of 3 dB is obtained.

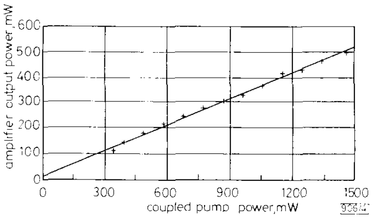

Fig. 4 *Optical amplifier output power against pump power for bidirectionally pumped unit*

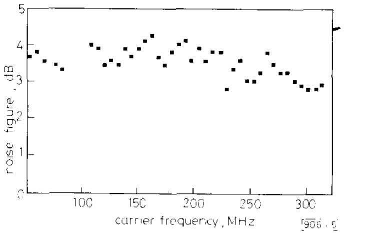

Fig. 5 *Amplifier system noise figure against carrier frequency*

Conclusion: A high output power Er/Yb codoped optical amplifier has been demonstrated using a diode-pumped Nd : YLF laser as the pump source. Small signal gains of up to 51 dB and output saturation powers of +24·6 dBm have been demonstrated when pumped with a single DPL. The diode-pumped Nd^{3+} : YLF laser pump source has allowed the use of high power, reliable AlGaAs diode laser arrays. As demonstrated, this approach is readily scalable in power with pump array size. Such ultrahigh output power optical amplifiers, although pumped by an argon-ion pumped Ti : sapphire laser, have already been used in systems experiments such as the demonstration of how a single laser transmitter could service 39·5 million subscribers [5].

Acknowledgments: We wish to thank J. E. Townsend and W. L. Barnes of the Optoelectronics Research Centre, University of Southampton for the fabrication and initial characterisation of the Er/Yb codoped fibre.

5th March 1992

S. G. Grubb, W. H. Humer, R. S. Cannon and S. W. Vendetta (*Amoco Technology Company, Naperville, IL 60566, USA*)

K. L. Sweeney, P. A. Leilabady, M. R. Keur, J. G. Kwasegroch, T. C. Munks and D. W. Anthon (*Amoco Laser Company, Naperville, IL 60566, USA*)

References

1 TAKENAKA, H., OKUNO, H., FUJITA, M., ODAGIRI, Y., SUNOHARA, Y., and MITO, I.: 'Compact size and high output power Er-doped fiber amplifier modules pumped with $1{\cdot}48\,\mu m$ MQW LD's'. Proc. Second Topical Meeting on Optical Amplifiers and Their Applications, Paper FD2, Snowmass, Colorado, 1991

2 TOWNSEND, J. E., JEDREZEWSKI, K. P., BARNES, W. L., and GRUBB, S. G.: 'Yb^{3+} sensitized Er^{3+} doped silica fibre with ultrahigh transfer efficiency and gain', *Electron. Lett.*, 1991, **27**, pp. 1958–1959

3 GAPONTSEV, V. P., MATITSIN, S. M., ISINEEV, A. A., and KRAVCHENKO, V. B.: 'Erbium glass lasers and their applications', *Opt. and Laser Technol.*, August 1982, pp. 189–196

4 TOWNSEND, J. E., POOLE, S. B., and PAYNE, D. N.: 'Solution doping technique for the fabrication of rare-earth doped optical fibres', *Electron. Lett.*, 1987, **23**, pp. 329–331

5 HILL, A. M., WYATT, R., MASSICOTT, J. F., BLYTH, K. J., FORRESTER, D. S., LOBBETT, R. A., SMITH, P. J., and PAYNE, D. B.: '39·5 million-way WDM broadcast network employing two stages of erbium doped fibre amplifiers', *Electron. Lett.*, 1990, **26**, pp. 1882–1884

PIPELINED VOLTERRA FILTER

M. M. Banat

Indexing terms: Adaptive filters, Pipeline processing

A pipelined realisation for a kth order Volterra filter is introduced. Although it needs input data broadcasting, the realisation uses a minimum number of simple identical processing elements (PEs), produces a fixed output delay equal to the order of the filter, and is suitable for implementing adaptive Volterra filters.

Introduction: Many signal processing applications involve nonlinearities having memory. In most of these cases, the Volterra series can provide a suitable means of modelling this kind of nonlinearity. Typical applications of nonlinear filtering in general, and Volterra filters in particular, include among many others, optimal prediction [1], channel equalisation [2], noise cancellation [3, 4], echo cancellation [5], and voiceband transmission with intersymbol interference [6].

Memory-oriented realisations of Volterra filters have been studied in References 7–9. A semisystolic realisation of a second order Volterra filter (SVF) was proposed by Kocur [10]. However, in that realisation, three types of processing element (PE) are needed. In addition, some of these PEs have to add three quantities, one of which is a product of two other quantities. The output is delayed by an amount equal to the length of the linear part of the filter. These problems are solved in the realisation suggested here by adding more input broadcasting. The proposed architecture may represent a Volterra filter of any order.

New realisation: The kth order Volterra filter output has the form

$$\begin{aligned} y_k(n) = y_0 &+ \sum_{i_1=0}^{N} h_1(i_1)x(n-i_1) \\ &+ \sum_{i_1=0}^{N}\sum_{i_2=0}^{N} h_2(i_1,i_2)x(n-i_1)x(n-i_2) + \ldots \\ &+ \sum_{i_1=0}^{N}\sum_{i_2=0}^{N}\cdots\sum_{i_k=0}^{N} h_k(i_1,i_2,\ldots,i_k) \\ &\times x(n-i_1)x(n-i_2)\ldots x(n-i_k) \end{aligned} \qquad (1)$$

In most situations of practical interest, it is convenient to assume symmetry in the filter coefficients. This implies that the arrangement of the arguments of h_k is not important. If the number of different arrangements of $i_1, i_2, \ldots, i_k$ is $C(i_1$ $i_2, \ldots, i_k)$ then

$$\begin{aligned} y_k(n) = y_0 &+ \sum_{i_1=0}^{N} H_1(i_1)x(n-i_1) \\ &+ \sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N} H_2(i_1,i_2)x(n-i_1)x(n-i_2) + \ldots \\ &+ \sum_{i_1=0}^{N}\sum_{i_2=i_1}^{N}\cdots\sum_{i_k=i_{k-1}}^{N} H_k(i_1,i_2,\ldots,i_k) \\ &\times x(n-i_1)x(n-i_2)\ldots x(n-i_k) \end{aligned} \qquad (2)$$

where

$$H_k(i_1,i_2,\ldots,i_k) = C(i_1,i_2,\ldots,i_k)h_k(i_1,i_2,\ldots,i_k) \qquad (3)$$

and $C(i_1, i_2, \ldots, i_k)$ can be derived on a combinatorial basis. The exact value of C is of no real importance if the filter is adaptive.

It is interesting to note that eqn. 2 may be rewritten in the alternative form

$$y_k(n) = y_{0)} + \sum_{i_1=0}^{N} x(n-i_1)g_{i_1}(n) \qquad (4)$$

where

$$g_{i_1}(n) = H_1(i_1) + \sum_{i_2=i_1}^{N} x(n-i_2)g_{i_1 i_2}(n) \qquad (5)$$

This can be repeated all the way up to

$$\begin{aligned} g_{i_1 i_2 \ldots i_{k-1}}(n) &= H_{k-1}(i_1,i_2,\ldots,i_{k-1}) \\ &+ \sum_{i_k=i_{k-1}}^{N} x(n-i_k)g_{i_1 i_2 \ldots i_k}(n) \end{aligned} \qquad (6)$$

where

$$g_{i_1 i_2 \ldots i_k}(n) = H_k(i_1,i_2,\ldots,i_k) \qquad (7)$$

The usefulness of these expressions becomes evident by noting that eqn. 4 can be represented by the pipeline in Fig. 1. Because all gs have the same form as $y_k(n)$, they can be realised the same way. A third order example with memory extent $N = 2$ is shown in Fig. 2. Note that if $N = 3$ then a second order Volterra filter realised in the proposed architecture requires 14 PEs compared with 18 in Kocur [10]. Hence, the new architecture requires the minimum number of PEs needed because it is equal to the total number of terms in the truncated series. From this Figure it can be easily seen that the total input-output delay is equal to k, the filter order, and is independent of N, the system memory size.

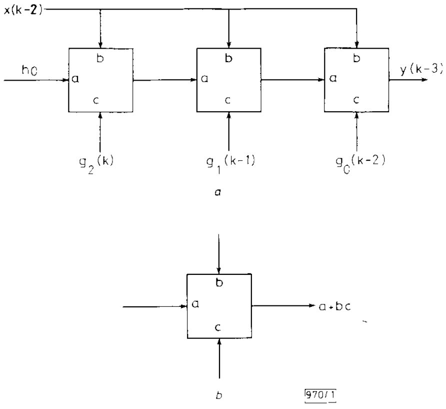

Fig. 1 *General pipeline to compute $y_k(n)$*

The multipliers in the last stages $g_{i_1 i_2 \ldots i_k}$ are constants and are equal to $H_k(i_1 i_2 \ldots i_k)$. These multipliers can be stored in the corresponding PEs. Alternatively, they can be treated as external multipliers. The second option looks better because of three important observations. First, it will make all the PEs in the realisation the same type, which is highly desirable in VLSI implementations. Secondly, no additional PE memory will be required. Thirdly, in most cases of practical interest the filter is adaptive; this means that its coefficients change with time according to an adaptive algorithm. This makes it much more convenient to recursively generate the H_ks in a separate adaptive unit, then feed them into the corresponding PEs.

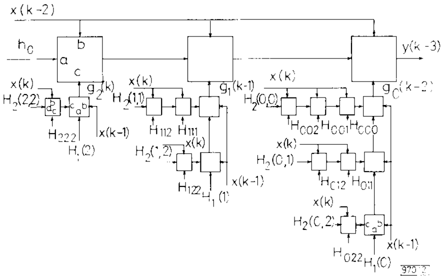

Fig. 2 *Third order Volterra filter with memory $N = 2$*

Conclusions: A pipelined realisation has been proposed to implement a Volterra filter of arbitrary order k. This realisation has some inviting features when compared to available structures which are limited to the 2nd order case. First, the number of processing elements (PEs) is minimised. Secondly, the complexity of PEs is reduced. Thirdly, and most importantly, the delay between the input and the output is fixed and does not grow with the length of the linear section. The delay depends totally on the filter order which is equal to the number of terms in the truncated Volterra series that represents the filter. The proposed architecture is suitable for adaptive Volterra filters because the adjustable weights are treated as external multipliers to the PEs. This also reduces the memory requirements of the PEs.

11th May 1992

M. M. Banat (*Electrical Engineering Department, University of Ottawa, 161 Louis Pasteur, Ottawa, Ontario K1N 6N5, Canada*)

References

1 MCCANNON, T. E., GALLAGHER, N. C., MINOO-HAMEDANI, D., and WISE, G. L.: 'On the design of nonlinear discrete-time predictors', *IEEE Trans.*, 1982, **IT-28,** pp. 366–371

2 FALCONER, D. D.: 'Adaptive equalization of channel nonlinearities in QAM data transmission systems', *BSTJ*, 1978, **57,** pp. 2589–2611

3 COKER, M. J., and SIMKINS, D. N.: 'A nonlinear adaptive noise canceller'. Proc. ICASSP, 1980, pp. 470–473

4 SLAPETON, J. C., and BASS, S. C.: 'Adaptive noise cancellation for a class of nonlinear dynamic reference channels'. Proc. ISCAS, 1984, pp. 268–271

5 AGGAZI, O., MESSERSCHMITT, D. G., and HODGES, D. A.: 'Nonlinear echo cancellation of data signals', *IEEE Trans.*, 1982, **COM-30,** pp. 2421–2433

6 BIGLIERI, E., GERSHO, A., GITLIN, R. D., and LIM, T. L.: 'Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission', *IEEE J. Sel. Areas Commun.*, 1984, **SAC-2,** pp. 765–777

7 SICURANZA, G. L.: 'Nonlinear digital filter realization by distributed arithmetic', *IEEE Trans.*, 1985, **ASSP-33,** pp. 939–945

8 SICURANZA, G. L., and RAMPONI, G.: 'Adaptive nonlinear digital filters using distributed arithmetic', *IEEE Trans.*, 1986, **ASSP-34,** pp. 518–526

9 SICURANZA, G. L., and RAMPONI, G.: 'A variable step adaptation algorithm for memory-oriented Volterra filters', *IEEE Trans.*, 1987, **ASSP-35,** pp. 1492–1494

10 KOCUR, D.: 'Semisystolic implementation of second-order Volterra digital filters based on recurrence algorithm of Volterra filtering', *Electron. Lett.*, 1989, **25,** pp. 25–26

NEW ARCHITECTURE FOR PARALLEL MULTIPLIERS

Zhongde Wang, G. A. Jullien and W. C. Miller

Indexing terms: Binary arithmetic, Multipliers, Computer architecture, Integrated circuits

An architecture for parallel multipliers, based on 2 bit full adders, is proposed. Multipliers built in this way possess the same structure as previously published five-counter multipliers, but with fewer cells and a reduction of one stage. The architecture is particularly suited to nFET complex block pipelined dynamic logic, and the 2 bit adder cell appears to be more efficient than the five-counter cell.

Introduction: Multipliers are of vital importance for modern computers and for a variety of signal processors. Although column compression techniques [1, 2], which reduce the critical path delay to $O(\log_{1.5} n)$ cell delays, have been proved to provide time optimal techniques for building multipliers, the resulting irregular interconnections between cells produce awkward structures for implementation in silicon. Parallel multipliers are often adopted for silicon implementation because of the regular structure and simple interconnection between cells [3, 4].

Conventionally, parallel multipliers are built with a binary full adder as the basic cell [3, 4]. An $n \times n$ bit parallel multiplier built in this way requires $2(n - 1)$ cell delays to complete the computation [3]. Recently, Nakamura proposed several algorithms for parallel multipliers which require only $n + 1$ cell delays to perform the multiplication operation [5]; the architecture used a five-counter multiplier as the basic cell. Under the assumption that the cell delay for a binary full adder and a five-counter cell are the same, multipliers with architectures proposed by Nakamura will yield about half the latency of conventional parallel multipliers. Maden and Guy [7] proposed another architecture for parallel multipliers which uses (2, 2, 3) counters as basic cells. Among these different multiplier architectures, the five-counter multiplier appears to be the most useful and a VLSI implementation of this type of multiplier has recently been reported [6].

We propose a multiplier architecture which uses a 2 bit adder as the basic building cell. The multiplier possesses almost the same structure as the five-counter multiplier, but with fewer cells and a slightly lower critical path delay.

Five-counter multiplier: Because our architecture represents an improvement on the five-counter multiplier [5], we will briefly discuss this comparison architecture. Fig. 1 shows the architecture of a 6 × 6 multiplier built with five-counter cells. There is a redundancy associated with the fact that although the three outputs of the basic cell can accommodate a count of seven equally weighted inputs, the maximum count is only five. This redundancy reflects an inefficiency in the overall architecture. There is also considerable redundancy associated with the five-counter cells on the first stage, i.e. cells on the left-most column in Fig. 1, because these have only two active inputs.

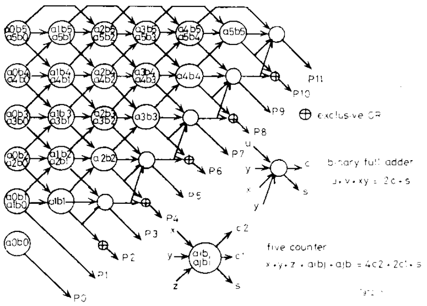

Fig. 1 *Five-counter based 6 × 6 multiplier*

2-bit adder multiplier: For this new architecture, we replace the five-counter cell with a 2 bit full adder cell. The use of the 2 bit adder (the output of which completely spans the 3 bit outputs) removes the first redundancy, and also helps eliminate the second.

Fig. 2 shows the architecture of a 6 × 6 2 bit full adder multiplier. The 2 bit full adder block is used to add five independent 1 bit numbers. Three of them are generated by cells of the previous stage, and they possess the same weight. The other two numbers, each with a weight twice the weight of the previous stage inputs, are elements of the partial products of the two n bit input numbers:

$$b_i(a_{n-1}a_{n-2}\dots a_0);\ i = 0, 1, \dots, n - 1 \tag{1}$$

The structure is similar to that of the inner part of the five-counter multiplier (starting one column in); the internal product elements are identical. Each cell has symmetric partial product elements for the most significant inputs: $a_i b_j$ and $a_j b_i$

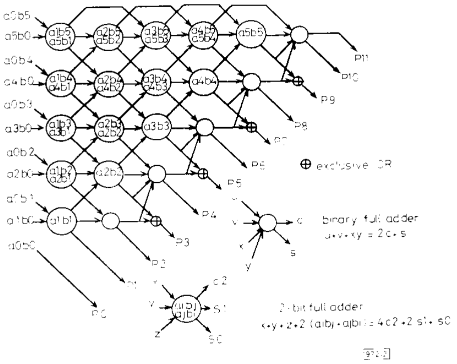

Fig. 2 *6 × 6 2 bit adder multiplier*