



Bull, DR., & Wacey, G. (1994). Gate level optimisation of primitive operator digital filters using a carry save decomposition. In *Unknown* (Vol. 2, pp. 93 - 96). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/ISCAS.1994.408913>

Peer reviewed version

Link to published version (if available):
[10.1109/ISCAS.1994.408913](https://doi.org/10.1109/ISCAS.1994.408913)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Gate Level Optimisation of Primitive Operator Digital Filters Using a Carry Save Decomposition.

David R. Bull and Graham Wacey
 Dept Electrical and Electronic Engineering, University of Bristol,
 Queens Building, University Walk, Bristol BS8 1TR, UK.
 (+44)272-288613
 Dave.Bull@uk.ac.bristol

ABSTRACT

This paper introduces a method for optimising digital filter realisations at the gate level. The method is based on a derivative of the primitive operator approach of Bull and Horrocks which is extended using a carry-save decomposition of the primitive operator graph. This facilitates the generation of a set of boolean expressions for the multiply-accumulate section of the filter which can be minimised using standard sum of products or Reed Muller techniques. The technique is fully described and results are presented for a representative range of FIR filters. Savings of up to 83% are obtained for sum-of-products minimisation when compared to a CSD coded hard-wired multiplier solution. Initial results suggest further improvements in excess of 20% for the Reed Muller case

INTRODUCTION

The efficient (multiplier-free) realisation of fixed transfer function digital filters has been the focus of much research in recent years, particularly for applications such as video signal processing where high throughput and low silicon cost are dominant design constraints. The *primitive operator filter* (POF) methodology [1] embodies one technique for reducing implementation complexity by exploiting the redundancy inherent in a multiplier-based realisation. It does this by replacing all inner product multiplication operations by a single directed signal flow graph, employing only primitive operations (additions, subtractions and power of two gains). The graph is formed in a way which preserves the specified filter transfer function, with no loss of coefficient accuracy. The design process has been automated in the form of a package, *POFGEN* [2], and has been shown to offer significant savings in real applications (eg[3]).

This paper discusses a derivative of the above technique which facilitates further decomposition of the POF graph, using a *carry-save* approach, into a form amenable to boolean minimisation. This in turn, allows the system to be realised efficiently in either a *sum of products* or *Reed Muller*

form [4]. This methodology was introduced in basic form in reference [4], but it is only recently that design tools have become available [2] which facilitate the full evaluation of its potential. The approach yields benefits in the following ways:

- inexpensive and flexible technologies such as PLAs and FPGAs can be used for implementation,
- the need for pipeline registers and hence the overall latency of the filter can be significantly reduced,
- gate counts are reduced especially if PLA based methods are employed,
- the irregular communication structure of the POF graph is easily mapped onto a regular array structures amenable to VLSI realisation,
- bit-parallel and bit-serial arithmetic methodologies are supported, and
- control overheads are reduced for bit-serial designs.

The paper begins with an introduction to the decomposition technique and includes an illustrative example. It continues by presenting results for both sum-of-products and Reed Muller synthesis methods, indicating the savings possible for a range of FIR filters.

THE CARRY-SAVE DECOMPOSITION

In the simple addition-only POF graph [1], vertices represent two-input adders and edge gains are constrained to values of zero or unity. A sub-graph is illustrated in figure 1, where the output signal from any vertex, k , is represented by $W_k[n]$ and where the carry signal, $C_k[n]$, propagates within the vertex. This propagation occurs in time for a bit-serial system and space for a bit-parallel system.

If the carry signal is unfolded and propagated to an adjacent vertex in a replicated version of the original sub-graph, figure 2, then, provided that all sub-graph outputs are accumulated correctly, the graph transfer function is preserved. This form of decomposition can be applied iteratively to all vertices in the original and replica graphs allowing each vertex input output relationship to be described in terms of two simple

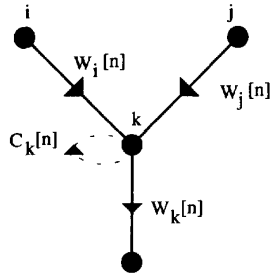


Figure 1 Primitive operator sub graph

boolean expressions. Decomposing all vertices in this manner facilitates minimisation of the entire graph using standard boolean techniques.

In general, if the sum output at a vertex, k , in precedence level, p , of the graph is given by $W_{p,k}[n]$ and the associated carry by $C_{p,k}[n]$, then the general boolean expressions for a full adder k at level p are given by equations (1) and (2). Clearly if $p=0$, then the carry in, $C_{-1,k}[n] = 0$, and hence all vertices at this level represent half adders.

$$W_{p,k}[n] = W_{p,i}[n] \oplus W_{p,j}[n] \oplus C_{p-1,k}[n] \quad (1)$$

$$C_{p,k}[n] = (W_{p,i}[n] \vee W_{p,j}[n]) \wedge (W_{p,i}[n] \vee C_{p-1,k}[n]) \wedge (W_{p,j}[n] \vee C_{p-1,k}[n]) \quad (2)$$

All vertices in the graph can thus be represented in the form of (1) and (2). At each successive level of decomposition, all signals associated with the top precedence level in the preceding sub-graph are eliminated. Hence the process terminates after all precedence levels in the graph have been fully decomposed. This results in a structure comprising D sub-graph sections with decreasing complexity. An upper bound on D is given by:

$$D_{\max} = \Delta_{\max} \quad (3)$$

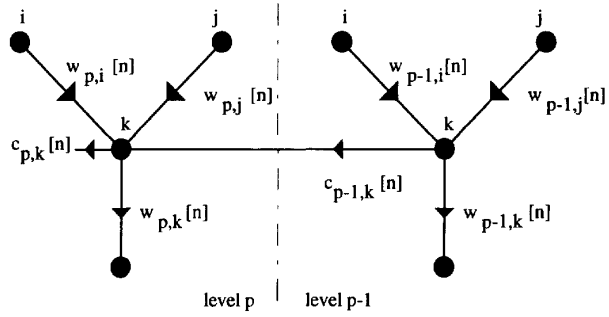


Figure 2 Single vertex decomposition

Where Δ_{\max} is the maximum number of precedence levels in the original POF graph.

If edge gains in the original POF graph are allowed to assume any value in $\{2^i\}$, Δ_{\max} can be significantly reduced. Shifts can be accommodated in the decomposition process by allowing sum paths to propagate across sub-graph boundaries in the same manner as the carry paths. The number of sub-graphs, D , is now a function of the graph edge gains as well as the number of precedence levels and the upper bound is now given by:

$$D_{\max} = \max_{\forall_{ij}} [s_{ij} + (\Delta_{\max} - \Delta_i)] \quad (4)$$

An added advantage in this case is that many vertices within the carry-save graph have a simpler form and the resulting boolean expressions are more amenable to reduction. This technique is illustrated by the example in figure 3 for the case of the a simple FIR filter coefficient set $\{1, 2, 3, 6\}$.

The boolean equations for this example are given below:

$$\begin{aligned} W_{00} &= X_{00} \\ W_{01} &= X_{00} \\ W_{02} &= W_{00} \\ W_{03} &= W_{01} \oplus W_{02} \\ C_{03} &= W_{01} \vee W_{02} \end{aligned}$$

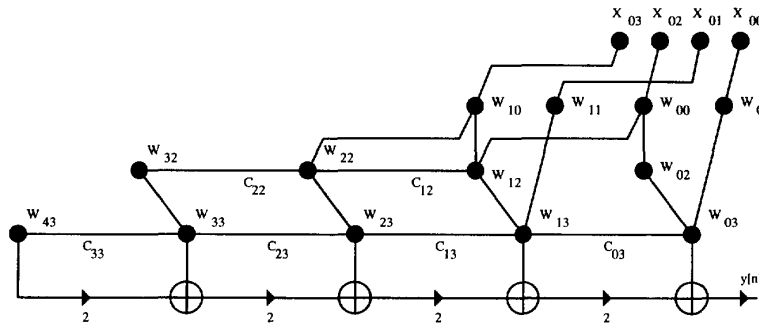


Figure 3 Carry save graph for the set $\{1, 2, 3, 6\}$. Signals X_{ij} represent the z^{-1} delay line taps

$$\begin{aligned}
W_{10} &= X_{03} \\
W_{11} &= X_{01} \\
W_{12} &= W_{10} \oplus W_{00} \\
C_{12} &= W_{10} \vee W_{00} \\
W_{13} &= W_{01} \oplus W_{02} \oplus C_{03} \\
C_{13} &= (W_{01} \vee W_{02}) \wedge (W_{01} \vee C_{03}) \wedge (W_{02} \vee C_{03}) \\
\\
W_{22} &= W_{10} \oplus C_{12} \\
C_{22} &= W_{10} \vee C_{12} \\
W_{23} &= W_{22} \oplus C_{13} \\
C_{23} &= W_{22} \vee C_{13} \\
\\
W_{32} &= C_{22} \\
W_{33} &= W_{32} \oplus C_{23} \\
C_{03} &= W_{32} \vee C_{23} \\
W_{34} &= C_{33}
\end{aligned} \tag{5}$$

The carry-accumulate chain at the bottom of figure 3 is required to combine the outputs from each sub-graph, appropriately shifted to ensure correct bit alignment. The filter output is thus given by:

$$y[n] = \sum_{i=0}^D W_{i,Q}[n] \cdot 2^i \tag{6}$$

where Q is the index of the output vertex in each sub-graph.

For the majority of practical filters negative as well as positive coefficient values are required. Subtraction operations (or equivalently, signed edge gains) must be accommodated in the decomposition procedure. A solution to this problem has been previously outlined [4] which combines the use of one's complement subtraction with a carry correction stage at the filter output. The vertex equations must be modified for subtraction as follows:

$$W_{p,k}[n] = W_{p,i}[n] \oplus \overline{W}_{p,j}[n] \oplus C_{p-1,k}[n] \tag{7}$$

$$\begin{aligned}
C_{p,k}[n] &= (W_{p,i}[n] \vee \overline{W}_{p,j}[n]) \wedge (W_{p,i}[n] \vee C_{p-1,k}[n]) \wedge \\
&(\overline{W}_{p,j}[n] \vee C_{p-1,k}[n])
\end{aligned} \tag{8}$$

SUM OF PRODUCTS SYNTHESIS

Initial minimisation of the resulting carry-save graph expression are performed in *POFGEN* [2]. Further reductions have been obtained using a modified version of the standard sum-of-products synthesis tool, *MIS II* [6]. Gate count results from this minimisation are presented in figures 4 and 5 for the case of linear phase FIR filters with 12 bit word lengths and orders ranging from 8 to 100. Figure 4 plots values for the case of a decomposition derived from an addition only (type 1) primitive operator graph whereas, for figure 5, results are based on a shift-add-subtract graph (type 2). Both show comparisons between a hardwired CSD multiplier solution [7], the original primitive operator graph,

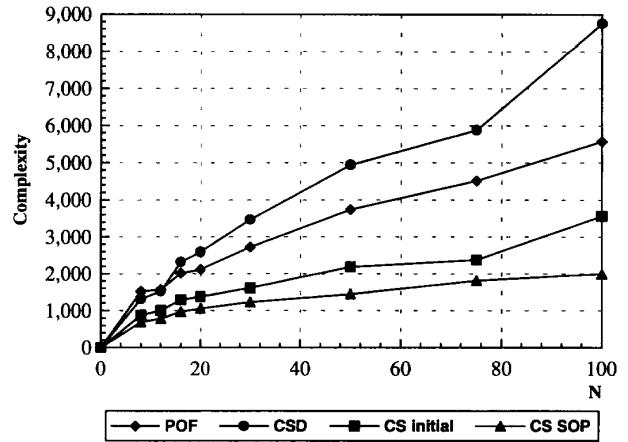


Figure 4: Addition only gate count comparisons

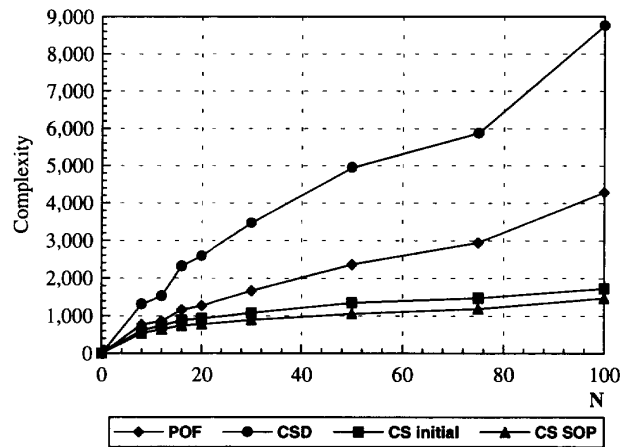


Figure 5: Add/Subtract/shift gate count comparisons

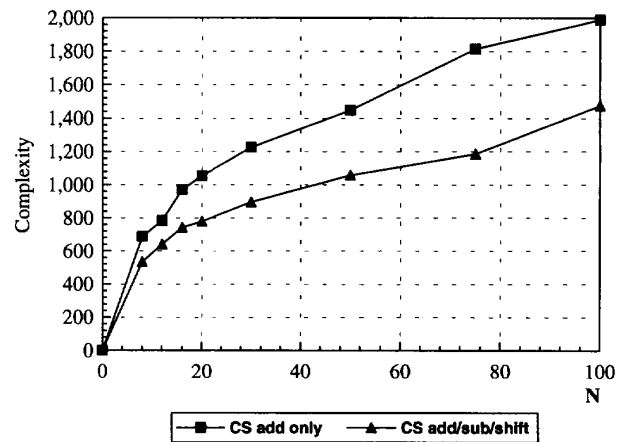


Figure 6: Comparisons for add only and add/sub/shift.

and the carry-save solutions. For the filters investigated, these results indicate savings for the type 1 case of 77% when compared with CSD and 64% when compared with the original POF graph. For the type 2 case these figures increase to 83% and 65% respectively. A direct comparison between the type 1 and type 2 algorithms is shown in figure 6 which indicates savings for the latter of up to 35%.

All results assume a bit serial realisation with full pipelining for the CSD and standard POF variants. Gate counts are computed for the multiply accumulate section of the filters only, with complexity calculations performed on the following basis:

Full adder: 14 gates
Register: 8 gates

Similar results have been computed for the case of a 4 gate register with results showing savings for the type 2 algorithm of 75% compared to CSD and 50% compared to the original POF solution.

These figures will of course vary according to the implementation technology. Further benefits will arise from reduced latency (and hence pipeline register requirements), savings in control overhead or the potential of realisation using PLA type techniques.

REED MULLER SYNTHESIS

It is well known that conventional sum-of products minimisation does not perform well for arithmetic functions such as adders or coding functions such as parity functions. Recent research has focused on the use of a Reed Muller representation - an exclusive-OR sum of AND product terms - as a way of describing and manipulating functions [4]. Work is ongoing at Bristol University in this respect and design tools currently under development have been used to evaluate the potential of this approach in the current problem area. Preliminary results have been produced for a 16th order FIR filter with 12 bit coefficients show a saving of 29% when compared with its sum-of products counterpart. We believe that this demonstrates the potential of this approach and further work is underway.

CONCLUSIONS

A general method has been presented which facilitates the optimisation of digital filter multiplier-accumulator block at the gate level. When compared to a similar gate count for the conventional primitive operator graph, savings up to 65% have been demonstrated. Such techniques also facilitate the adoption of PLA technology offering the potential of layout regularity and the development of standard parts. Current work is continuing to investigate the benefits of Reed Muller logic synthesis techniques in this area.

ACKNOWLEDGEMENTS

The authors would like to express their thanks to Nigel Lester of the Department of Electrical and Electronic Engineering at the University of Bristol for his assistance in producing the logic minimisation results presented here.

REFERENCES

- [1] Bull, D.R. and Horrocks, D.H., 'Primitive Operator Digital Filters', IEE Proceedings-G.,138 (3), June 1991, pp 401-412.
- [2] Wacey, G. and Bull, D.R., 'POFGEN: A Design Automation System for VLSI Digital Filters with Invariant Transfer Function', IEEE ISCAS '93, Chicago USA, April 1993, pp. 631-634.
- [3] Bull, D.R., Wacey, G., Stone, J.J. and Soloff, J.M., 'A Compound Primitive Operator Approach to the Realisation of Video Sub-Band Filter Banks', IEEE ICASSP '93, Minneapolis USA, May 1993, pp. 405-408.
- [4] Saul, J., 'Logic Synthesis for Arithmetic Circuits using the Reed Muller Representation', Proc. European Conf. on Design Automation, Belgium, March 1992, pp109-112.
- [5] Bull, D. R. and Horrocks, D. H., 'A Carry Save Architecture for Primitive Operator Digital Filters', Proc. European Conf. on Circuit Theory and Design, Brighton, UK, September 1989, pp180-184.
- [6] Brayton, R.K. and Rudell, R., Sangiovanni, A. and Wang, A., 'MIS: A Multiple Level Logic Optimisation System', IEEE Trans., CAD-6 (6), Nov 1987, pp1062-1081.
- [7] Peled, A., 'On the Hardware Realisation of Digital Signal Processors', IEEE Trans. ASSP-24, 1976, pp76-86.