OPEN ACCESS

University of BRISTOL

Link to published version (if available):
10.1109/ICASSP.1993.319141

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research
General rights

A COMPOUND PRIMITIVE OPERATOR APPROACH TO THE REALISATION OF VIDEO SUB-BAND FILTER BANKS

David R. Bull*, Graham Wacey*, John J. Stone+ and Jon M. Soloff+

*Dept. Electrical and Electronic Eng., University of Bristol, Bristol BS8 1TR, UK
+Sony Broadcast and Communications, Priestley Road, Basingstoke, Hants. RG24 9JP, UK

ABSTRACT

This paper presents a new video sub-band filtering architecture appropriate for VLSI implementation. The system employs a reduced complexity multiply-accumulate structure realised using an extension to the primitive operator graph synthesis technique. This, in conjunction with a data multiplexing regime which implements a two channel QMF on a single FIR structure, has facilitated the fabrication of a sixty four sub-band coder / decoder on a single gate array. The circuitry is reconfigurable, allowing vertical and horizontal filtering in analysis or synthesis mode. The paper describes the conceptual development of the new approach and presents novel architectural features associated with its implementation. Also included are complexity comparisons with conventional approaches.

1. INTRODUCTION

Image decorrelation using sub-band filtering methods has been the subject of increased interest in recent years [1,2,3,4]. Since the image artifacts produced can be controlled by the choice filters and architecture, a performance superior to that obtainable with the DCT is possible.
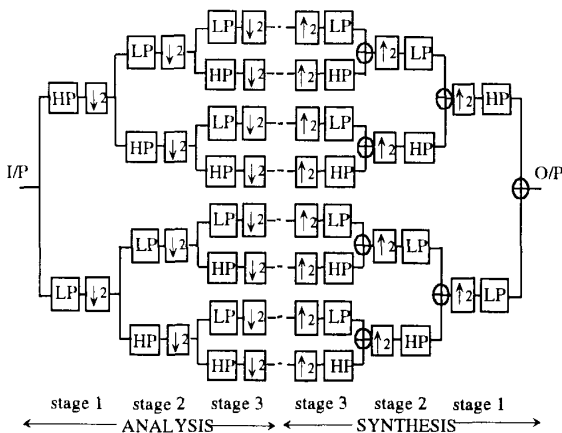

stage 1   stage 2   stage 3   stage 3   stage 2   stage 1
←————— ANALYSIS —————→ ←————— SYNTHESIS —————→

Figure 1: One dimensional sub-band structure for n=3

Sub-band filtering relies on the Quadrature Mirror Filter (QMF) structure [1]. In its simplest form, the two channel QMF can be used to separate an input signal into low and high-pass sub-bands such that the overall data rate remains constant after decimation. Furthermore, in the absence of quantisation, it is possible to perfectly reconstruct the original input signal from the low and high pass sub-bands [1,3]. The two channel QMF can be applied in subsequent stages to yield multiple sub-bands. For example, a flat decomposition yields 2^n sub-bands, where n is the number of stages, and this is shown in figure 1 for n=3. For images, the QMF can be applied to the rows and then to the columns of the image yielding multiple sub-bands in the two dimensional spatial frequency plane.

The filter architecture presented here derives its efficiency from two independent techniques. The first utilises a data multiplexed QMF (DMQMF) architecture to efficiently implement the two-channel QMF as a single finite impulse response (FIR) structure which allows constant data rate processing throughout all filtering stages. The second technique focuses on the multiplier array as a major influence on both implementation complexity and power consumption. Available techniques for complexity reduction range from the elimination of redundant multiplier rows through canonical coefficient recoding [5], to distributed arithmetic [6] and residue number system based methods. An alternative approach, which eliminates the requirement for explicit multiplication operations by realising the coefficient vector - data vector inner product in the form of an optimised directed graph, is the primitive operator filter (POF) technique [7]. POF has, in the past, only been applied to situations requiring a single invariant filter response. However, an extension to the method is developed here which facilitates its use in reconfigurable filter bank designs.

2. THE DATA MULTIPLEXED QUADRATURE MIRROR FILTER

The data multiplexed quadrature mirror filter (DMQMF) architecture is based on the standard FIR structure as shown in figure 2. The output of an odd-symmetrical N-tap filter ,y[n], is given by equation 1, where $M=(N-1)/2$ and D the amount of inter-tap delay in multiples of the original sampling period, $1/f_s$.
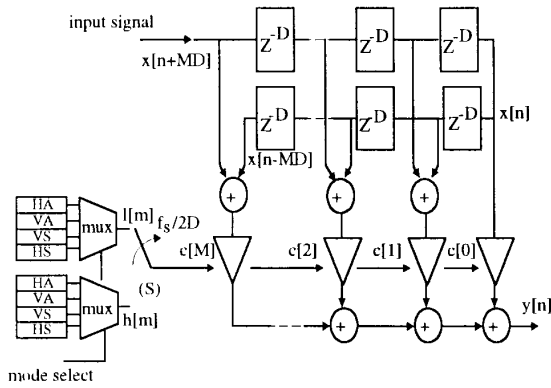
Figure 2: Data multiplexed QMF structure

$$y[n] = x[n].c[0] + \sum_{m=1}^{M}\left(x[n+mD]+x[n-mD]\right).c[m] \quad (1)$$

Since each stage of the analysis QMF filtering process is followed by a factor of two decimation, half of the samples in the output sequence are redundant. If the decimation operations are performed out of phase then both analysis half-rate filters can be combined to form a single full-rate filter with low-pass / high-pass coefficient multiplexing.. Similarly, the synthesis half-rate filters can also be combined to form a single full-rate filter, this time with the synthesis low-pass and high-pass coefficients interleaved. A more efficient hardware realisation of the filter bank can thus be devised which generates only non-redundant output samples.

For the first stage of analysis in the DMQMF, D=1. A selector (S) is used to switch the coefficients c[m] between low-pass and high-pass vectors in such a way that the filtered samples, y[n], alternate between low and high-pass values. Thus, when generating output sample n, the filter coefficients c[m] are given by:

$$c[m] = l[m] \quad :0 \le m \le (N_l - 1)/2 \quad \text{for n even}$$

$$c[m] = h[m] \quad :0 \le m \le (N_h - 1)/2 \quad \text{for n odd}$$

where l[m] and h[m] are the low-pass and high-pass filter coefficients, respectively of length N_l and N_h. The output sequence y[n] thus consists of a data multiplexed sequence of low and high pass samples represented by (L,H,L,H,L,H,L,H).

In the second stage of DMQMF, D=2. Each delay block thus represents two delay units and the coefficient selector S switches at fs/4. The low-pass filter is therefore applied to the first two samples and the high-pass filter is then applied to the next two samples such that y[n] now takes the form: (LL,HL,LH,HH,LL,HL,LH,HH). In a similar way D=4 for the third stage resulting in an output sequence of (LLL,HLL,LHL,HHL,LLH,HLH,LHH,HHH).

The function of the synthesis process is to add together the interpolated low-pass and high-pass filtered signals into a single data stream. This is done using the same FIR structure as shown in figure 2 with the coefficient selector S selecting between two sets of filter coefficients for alternate output samples. However, in this case, one of the sets comprises odd low-pass coefficients and even high-pass coefficients and the other set comprises odd high-pass coefficients and even low-pass coefficients. By arranging the coefficients in this manner, the addition of the low-pass and high-pass interpolated signals can be implemented within the filter itself. The coefficients are switched in such a way that low-pass input samples are always multiplied by low-pass coefficients and high-pass input samples are always multiplied by high-pass coefficients.

For the sub-band filtering of some images, it is preferable to use different filters for the vertical and horizontal dimensions. Therefore, four filtering modes can be defined: horizontal analysis (HA), vertical analysis (VA), vertical synthesis (VS) and horizontal synthesis (HS), these being selected using the multiplexers in figure 2. Each mode comprises two filters (HP and LP) giving a total of eight coefficient sets. Table 1 shows a typical example of eight coefficient sets represented to 15 bit precision and derived using a method described in [8].

Implementation of successive banks of DMQMF can be achieved through multiple concatenation of the circuit shown in figure 2, using an appropriate value of D at each stage.

3. A COMPOUND PRIMITIVE OPERATOR MULTIPLIER-ACCUMULATOR REALISATION

3.1 Compound Primitive Operator Graph Synthesis

The POF approach [7] exploits redundancy present in the coefficient vector-data vector inner product computation to yield an optimised multiplier-free replacement for the conventional multiplier-accumulator array. In its basic form, the technique could be employed to generate eight independent primitive operator graphs, one for each sub-band filter type required. This however yields little or no saving over general purpose multipliers with switchable coefficients.

The POF technique can be adapted to synthesise a single graph embodying coefficient vertices for all eight filters. Just as the conventional primitive operator graph exploits vertex reuse within a single filter, so the adapted method extends this to allow reuse across multiple filters. In the case of the DMQMF, depending on the QMF coefficients selected, a high degree

	c[7]	c[6]	c[5]	c[4]	c[3]	c[2]	c[1]	c[0]
LP.AH	0	0	0	256	-512	-1024	4608	9728
HP.AH	-10	19	62	-218	-659	664	4703	-9122
LP.AV	0	0	0	0	0	-1024	4096	10240
HP.AV	0	0	0	0	-171	683	4267	-9558
LP.SV	0	0	0	0	0	-683	4096	9558
HP.SV	0	0	0	0	-171	1024	4267	-10240
LP.SH	0	-19	0	218	-512	-664	4608	9122
HP.SH	-10	0	62	-256	-659	1024	4703	-9728

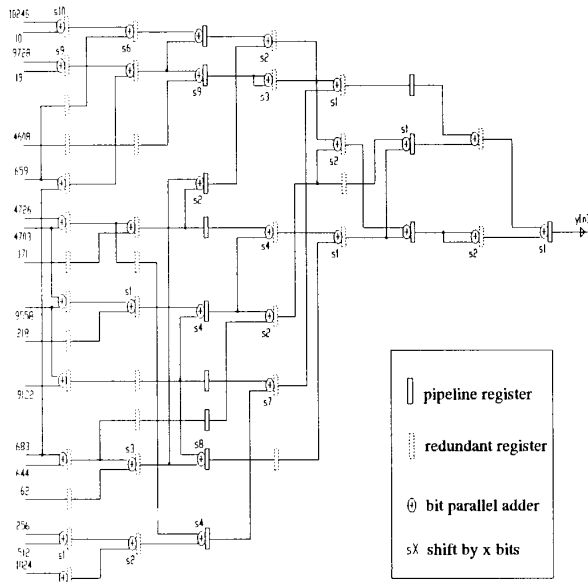**Table 1 - Filter Coefficient Vectors**

Figure 3: Pipelined primitive operator graph

of commonality can exist between vectors (see [8] and table 1). This characteristic is exploited by the new approach.

3.2 Implementation details

The set of unique coefficients from table 1 were employed to synthesise a compound primitive operator graph using the *POFGEN* design package [9]. The resulting graph when fully pipelined at the word level is shown in figure 3. The complexity of this structure can however be further reduced when wordlength and timing rules are considered. These facilitate reductions in internal data path widths and the number of pipeline stages respectively.

With knowledge of the input data wordlength together with individual coefficient values, the maximum internal signal wordlength can be determined. An upper bound on the output wordlength, B_{out}, can be computed using equation 2, where B_{in} is the input wordlength and ceil(.) returns the least integer greater or equal to its argument.

$$B_{out} = ceil\left(log_2\left(\left| c[0] \right| + \sum_{i=0}^{M} \left| c[i] \right| \right) \right) + B_{in} \qquad (2)$$

With an initial wordlength of 13-bits (after the folding addition operation), an upper limit of 28-bits, dominated by the HP.HS filter, results. The internal data path width of the graph has thus been allowed to increase to this value prior to MSB truncation.

The processing delay caused by each processing element is a function of the wordlength at the associated graph vertex, the delay characteristics of the cell components used and the capacitative loading due to tracking. Ignoring the effects of the

latter, the delay, d_a, for a single adder comprising k+1 4-bit look-ahead-carry adder blocks is given by equation 3,

$$d_a = t_1 + t_2 + (k - 1)t_3 \qquad (3)$$

where:
t_1:= worst case delay for an input to output transition.
t_2:= worst case delay for a carry in to output transition.
t_3:= worst case delay for a carry in to carry out transition.
t_4:= worst case delay for an input to carry out transition.

The total delay, d_t, for a path through A consecutive adders is given by equation 4. Assuming $t_1 > t_2 > t_4 > t_3$ and $t_1 + t_3 < t_2 + t_4$ then,

$$d_t = At_2 + At_4 + (K - A)t_3 \qquad :K \geq A$$
$$\qquad (A - K)t_1 + Kt_2 + Kt_4 \qquad :K \leq A \qquad (4)$$

where K is the largest value of k associated with any of the A adders in the chain. Using the above equations and incorporating additional delays due to capacitative loading, each path through the graph can be optimally load balanced and any redundant pipeline registers removed. The result of this exercise has been to reduce the POF structure from eight to only three pipeline stages as indicated in figure 3.

The POF structure requires a control unit to configure data paths according to filtering task. Table 2 gives example connections required to route folding adder output signals $x_m[n]$ (equation 5), for each filter, to the correct POF weighted path.

$$x_m[n] = x[n + mD] + x[m - mD] \qquad (5)$$

It can be observed that each graph input vertex need only be switched between two possible sources: $x_m[n]$ (for fixed m) or signal ground. Each switch can thus be realised with minimal overhead and controlled by signals derived from a three bit filter identifier code.

FUNCTION	Example	Weighted	Paths
	10240	10	4608
LP.AH	Gnd	Gnd	$x_1[n]$
HP.AH	Gnd	$x_7[n]$	Gnd
LP.AV	$x_0[n]$	Gnd	Gnd
HP.AV	Gnd	Gnd	Gnd
LP.SV	Gnd	Gnd	Gnd
HP.SV	$x_0[n]$	Gnd	Gnd
LP.SH	Gnd	Gnd	$x_1[n]$
HP.SH	Gnd	$x_7[n]$	Gnd

TABLE 2: Signal connections to selected graph input vertices for all filters.

The overall sub-band filtering ASIC architecture is indicated in figure 4.

4. IMPLEMENTATION EFFICIENCY

Area comparisons between a compiled cell multiplier and the POF approach are given below, based on the following assumptions (BC= basic cell, ie one p-n transistor pair.):
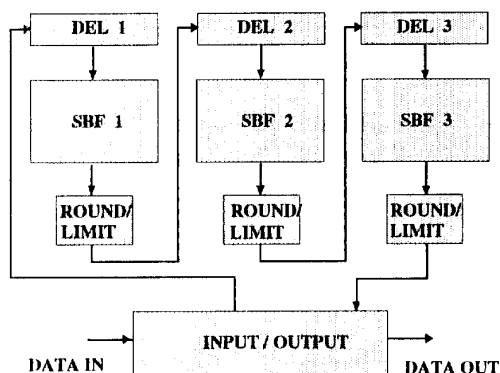
I-407

Figure 4: Three stage filter architecture

(i)   48   BCs per 4-bits of addition.
        20   BCs per 4-bits of register.
        2   BCs per bit of data switch.
        1   BCs per invertor.
        4   BCs per exclusive-OR gate.
        $(2m+8)*(4n+11)$ BCs for a compiled cell multiplier
        with m bit multiplicand and n bit multiplier.
(ii)   Only multiples of 4-bit adder blocks are used.
(iii)   An input wordlength of 12-bits is used.

It can be seen from table 3 that the compiled cell multiplier results in an additional 24861 basic cells when compared with the compound POF solution, representing an overall increase in basic cell count of approximately 68%. In practice however, silicon area will be utilised less efficiently by the POF approach due to routing complexity. Typical area utilisation adjustment factors are 0.8 for compiled cell layouts and 0.7 for normally routed (POF) layouts. Taking these factors into account, comparative complexity values can be derived which result in the compiled cell approach having a complexity 47% higher than that for the POF.

5. CONCLUSIONS

This paper has presented a new approach to the implementation of video sub-band filter banks. The system provides a reduced complexity solution through the use of a data multiplexing regime in conjunction with a primitive operator realisation of the filter multiplier accumulator These techniques together have facilitated the fabrication of a system,

on a single gate array, capable of operation in both analysis and synthesis modes for image decomposition into 64 sub-bands. The system described has been combined with quantisation, entropy encoding, field and line stores and rate control hardware to provide a complete, 2 bits per pixel 'perfect reconstruction' compression / decompression system for NTSC composite video signals [4].

ACKNOWLEDGEMENTS

REFERENCES

[1]   Vaidyanathan, P.P., "Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques", IEEE ASSP Magazine, Vol. 4, pp 4-22, 1987.
[2]   Woods, J. W. and O'Neil, S. D., "Subband Coding of Images", IEEE Trans. ASSP, Vol. ASSP-34, No. 5, pp 1278-1288, 1986.
[3]   Le Gall, D., and Tabatabai, A., "Sub-band Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques", Proc. ICASSP-88, Glasgow, UK, pp761-764, 1988
[4]   Hurley, T.R. and Stone, J.J., "Sub-band Coding of Composite Video For Data Compression In A Solid State Recorder", Proc IEE 4th International Conference on Image Processing and its Applications, Maastricht, Netherlands, pp 465-469, 1992.
[5]   Peled, A.,"On the Hardware Implementation of Digital Signal Processors", IEEE Trans. ASSP, Vol. ASSP-24, No.1, pp76-86, 1976.
[6]   Peled, A. and Liu, B.,"A New Hardware Realisation of Digital Filters", IEEE Trans. ASSP, Vol. ASSP-22, No.6, pp456-462, 1974.
[7]   Bull, D. R. and Horrocks, D. H.,"Primitive Operator Digital Filters", IEE Proc. Part G, Vol.138, No. 3, pp401-412, 1991.
[8]   Wilkinson, J.W., "Wavelet Transform in a Digital Video Tape Recorder", IEE Colloquium on Applications Of Wavelet Transforms in Image Processing, London, UK, 1993.
[9]   Wacey, G. and Bull, D. R.,"Architectural Synthesis of Digital Filters for ASIC Implementation", Proc IEE Saraga Colloquium on Digital and Analogue Filters', London, UK, pp11/1-11/6, 1991.

COMPOUND POF	BC COUNT	COMPILED CELL	BC COUNT
Folding addition	1530	Folding addition	1530
Input data inversion	182	Coefficient storage	1552
Data multiplexing	494	and multiplexing	
Multiplier/adder tree	9924	Multipliers	14266
Control logic	32	Adder tree	3100
Total for one stage	12161	Total for one stage	20448
Total for 3 stages	**36483**	**Total for 3 stages**	**61344**

TABLE 3 Basic Cell Count Comparisons for POF and Compiled Cell Approaches.

I-408