



A Compact Reconfigurable Filter Bank Implemented on a Low-Power Programmable Gate Array

Kebede Tremblay^{1*}, Abebe Kowalski², Pawel Ortega³

¹Department of Electrical Electronics, Sharavati University Centre for Electrical Electronics, Argentina

²Department of Electrical Electronics, Carpathian Institute of Electrical Electronics, Ireland

³Department of Electrical Electronics, Transylvanian Research Institute of Electrical Electronics, Greece

Article Info

Volume 1, Issue 1, 2026

[doi: 10.62715/ijvlsidsp.2026.v1.i1.pp21-25](https://doi.org/10.62715/ijvlsidsp.2026.v1.i1.pp21-25)

Received: 18 September 2025

Accepted: 02 December 2025

Published: 26 January 2026

Abstract

Filter banks that can change the number, width and position of their sub-bands while running are wanted in software-defined radio front ends, hearing instruments and spectrum monitors. The smallest field-programmable gate arrays, flash-based or low-leakage devices that draw little standby power, are attractive hosts for such banks in battery-operated equipment, but they offer a limited number of logic cells, very few hardened multipliers and modest block memory. This technical note examines how a reconfigurable filter bank can be made to fit. It compares four published routes to reconfigurability, namely modulated polyphase banks, coefficient decimation, frequency-response masking and cascaded integrator-comb stages, with respect to the resources each demands of a small device. It then describes an implementation pattern that suits the constraint: a single prototype filter, polyphase decomposition, time-multiplexed arithmetic, coefficients held in block memory and reconfiguration by writing parameters instead of loading a new bitstream. The note reports no measurements. It states the design reasoning, points out where the pattern is likely to fail, and lists what a full report of such an implementation should disclose so that results from different devices can be compared.

Keywords: Reconfigurable filter bank, Field-programmable gate array, Low-power design, Polyphase filters, Coefficient decimation, Software-defined radio

© 2026 Kebede Tremblay et al. This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

1. Introduction

A filter bank splits a signal into sub-bands so that each can be processed, measured or discarded separately. When the bands are fixed, the design problem is well understood and the theory is laid out in standard texts (Vaidyanathan, 1993). The harder case is a bank whose bands must change in service. A radio front end built on the software radio principle is expected to serve standards with different channel spacings (Mitola, 1995), and its channeliser has to follow. A hearing instrument fits its bands to the user's audiogram. In both, the equipment runs from a battery.

Field-programmable gate arrays are a natural platform for prototyping such banks, and the smallest

* Corresponding author: Kebede Tremblay, Department of Electrical Electronics, Sharavati University Centre for Electrical Electronics, Argentina. E-mail: kebede.tremblay@sharavati.edu

© 2026 Kebede Tremblay et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

families, built on flash or low-leakage memory cells, have static power low enough to remain in a portable product. They are also small. Programmable logic pays a substantial penalty in area and power relative to cell-based design (Kuon and Rose, 2007), and a low-power device offers few logic cells and very few hardened multipliers. This note considers how a reconfigurable filter bank can be arranged to fit within those limits. It is a commentary on design choices drawn from the published literature and presents no experimental results.

2. Routes to Reconfigurability

Four approaches recur in the literature, and they differ in what is shared between bands and in what has to change when the band plan changes. Table 1 sets them side by side, together with the plain alternative of storing a separate coefficient set for every band plan.

Table 1: Four routes to a reconfigurable filter bank and their demands on a small gate array			
Approach	What reconfigures	Main resource demand	Limitation
Modulated polyphase bank	Number of bands; prototype coefficients	One prototype filter and a transform	Bands uniform unless outputs are merged
Coefficient decimation	Decimation factor applied to one coefficient set	Multiplexers on the tap selection	Stopband attenuation falls as the factor rises
Frequency-response masking	Interpolation factor and masking filters	Long delay lines in memory	High group delay
Cascaded integrator-comb	Rate change factor	Adders and registers only	Passband droop; needs compensation
Stored coefficient sets	Whole coefficient set swapped per band plan	Memory for every stored set	Memory grows with the number of plans

In a modulated bank, a single low-pass prototype is shifted in frequency to form every band. With polyphase decomposition the prototype is computed once at the low output rate and a discrete Fourier or cosine transform performs the modulation (Bellanger et al., 1976), an arrangement that has become the standard channeliser for communication receivers (harris et al., 2003). The cost per band is low, but the bands are uniform. Unequal bands are obtained by merging adjacent outputs or by changing the prototype.

Coefficient decimation obtains several responses from one stored filter. Keeping every M -th coefficient and replacing the others with zeros produces a multi-band response, and discarding the others altogether widens the passband, so that the integer M becomes the control (Mahesh and Vinod, 2008). The same authors built low-complexity uniform and non-uniform channelisers on the idea (Mahesh and Vinod, 2011). Frequency-response masking, due to Lim (1986), reaches sharp transitions with sparse coefficients by interpolating a model filter and removing the unwanted images with masking filters. Cascaded integrator-comb filters use no multipliers at all (Hogenauer, 1981) and suit the first stage of rate reduction more than band selection.

3. An Implementation Pattern for Small Devices

The reasoning that follows leads to the arrangement in Figure 1, in which each stage of the signal path is paired with the measure that keeps its cost down. The starting point is that hardened multipliers are the scarcest resource and that the sample rates of audio and narrowband radio are far below the clock rate a gate array can sustain. Arithmetic should therefore be shared in time. A polyphase prototype with N taps needs N multiplications per output frame, and at a low output rate one multiply-accumulate unit, cycling through the taps, can perform them all. Coefficients and the sample delay line are held in block memory, which small devices provide in more generous measure than multipliers.

Where even one multiplier is unavailable, two older techniques apply. Distributed arithmetic replaces

the multiplier with a look-up table addressed by one bit of every input sample and an accumulator, which maps directly onto the look-up tables of programmable logic (White, 1989). Its table depends on the coefficients, so for a reconfigurable filter the table has to live in writable memory. The alternative is to restrict coefficients to sums of a few signed powers of two and build the products from shifts and adders with shared sub-expressions (Dempster and Macleod, 1995). This gives the smallest fixed filter but hard-wires the coefficients, and it is best kept for the parts of the bank that never change, such as a compensation filter after an integrator-comb stage. Practical guidance on both is collected by Meyer-Baese (2014).

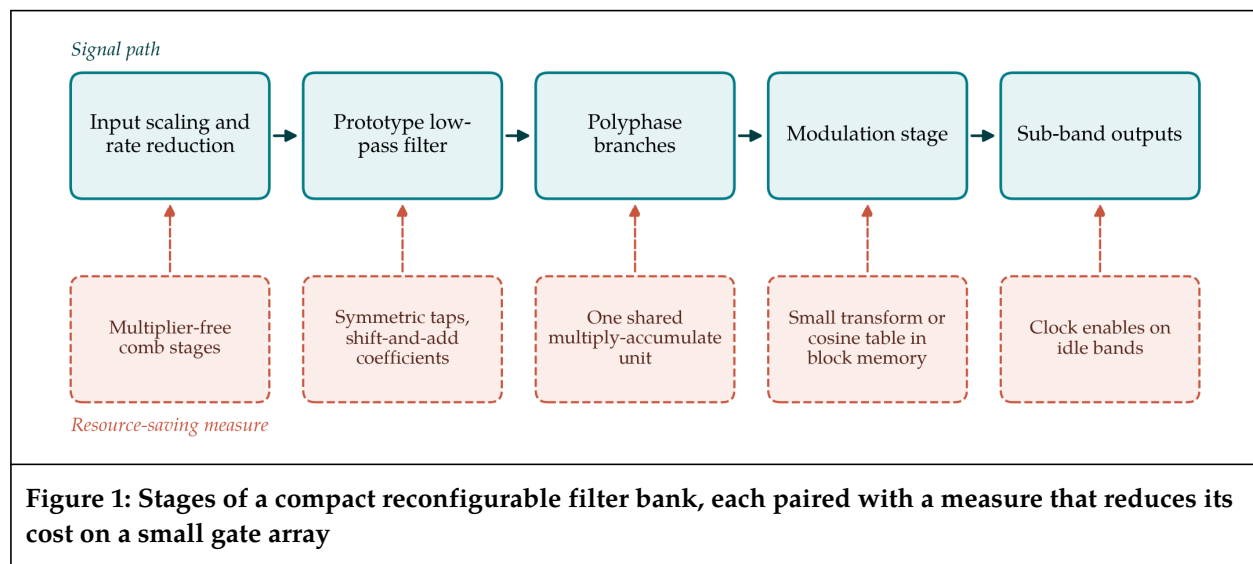


Figure 1: Stages of a compact reconfigurable filter bank, each paired with a measure that reduces its cost on a small gate array

Reconfiguration should be a matter of writing parameters. Loading a new bitstream interrupts the signal and, on flash-based devices, is slow. If the band plan is expressed as a coefficient set in memory, a tap count, a decimation factor M and a table saying which transform outputs are merged, a small controller can change it between frames. Coefficient decimation fits this scheme well, since a new response needs only a different address stride through the same memory. The polyphase structure fits it too, provided the transform size is fixed and unused outputs are simply not computed.

Dynamic power in programmable logic is dominated by the clock network and by switching in the routing. Three habits follow. Sections that are idle in the current band plan should have their clock enables held low, so that their registers do not toggle. Word lengths should be set stage by stage from the required dynamic range and not left at a uniform width, because every surplus bit is carried through every adder and memory word. The fast clock that drives the shared arithmetic should be no faster than the tap count requires.

Memory organisation needs some care as well. The delay line is most economically built as a circular buffer in a block memory with a moving write pointer, so that no data are shifted, and the coefficient memory is read in step with it. Where the prototype has linear phase its coefficients are symmetric, and half of them need be stored; a pre-adder that sums the two samples sharing a coefficient then halves the number of multiplications per frame. Both measures are standard, and both reduce the clock rate required of the shared arithmetic unit, which is what the power budget ultimately depends on.

4. Where the Pattern Is Likely to Fail

Three limits should be expected. Time-multiplexing breaks down as the sample rate rises: once the product of tap count and output rate approaches the achievable clock frequency, a second arithmetic unit is needed and the area advantage erodes. Coefficient decimation degrades the stopband as the factor increases, because the retained coefficients sample the prototype response more coarsely, so the prototype must be over-designed for the largest factor in use. Fixed-point effects are sharper in a shared datapath as well. A single accumulator width has to serve every band plan, and the guard bits that suffice for a

narrow band may overflow for a wide one.

5. What a Full Report Should Disclose

Published accounts of filter banks on gate arrays are hard to compare, because resource counts are given without the conditions that produced them. A report of an implementation along these lines would be more useful if it stated the device and tool version; the look-up tables, registers, memory blocks and hardened multipliers used; the sample rate and clock rate; the word lengths at each stage; and the prototype specification, including passband ripple and stopband attenuation for every supported configuration. Power should be measured on the board, separated into static and dynamic parts, and accompanied by the input signal used, since tool estimates depend strongly on assumed switching activity. The time taken to change configuration, and whether the output is interrupted, should be given as well.

6. Conclusion

A reconfigurable filter bank can be made compact enough for a low-power gate array by sharing one prototype among the bands, computing it with time-multiplexed arithmetic, keeping coefficients in block memory and changing the band plan through parameters. The approach trades throughput for area, and it suits audio and narrowband applications better than wideband ones. This note has given the reasoning and the expected limits. Measured implementations, reported with the detail listed above, are needed to show how far the pattern holds on particular devices.

Conflict of interest

To be declared by the author(s) before publication.

Funding

To be declared by the author(s) before publication.

References

- Bellanger, M., Bonnerot, G. and Coudreuse, M. (1976). [Digital filtering by polyphase network: Application to sample-rate alteration and filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24\(2\): 109-114. <https://doi.org/10.1109/TASSP.1976.1162788>](https://doi.org/10.1109/TASSP.1976.1162788)
- Dempster, A.G. and Macleod, M.D. (1995). [Use of minimum-adder multiplier blocks in FIR digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42\(9\): 569-577. <https://doi.org/10.1109/82.466647>](https://doi.org/10.1109/82.466647)
- harris, f.j., Dick, C. and Rice, M. (2003). [Digital receivers and transmitters using polyphase filter banks for wireless communications. *IEEE Transactions on Microwave Theory and Techniques*, 51\(4\): 1395-1412. <https://doi.org/10.1109/TMTT.2003.809176>](https://doi.org/10.1109/TMTT.2003.809176)
- Hogenauer, E.B. (1981). [An economical class of digital filters for decimation and interpolation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29\(2\): 155-162. <https://doi.org/10.1109/TASSP.1981.1163535>](https://doi.org/10.1109/TASSP.1981.1163535)
- Kuon, I. and Rose, J. (2007). [Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26\(2\): 203-215. <https://doi.org/10.1109/TCAD.2006.884574>](https://doi.org/10.1109/TCAD.2006.884574)
- Lim, Y.C. (1986). [Frequency-response masking approach for the synthesis of sharp linear phase digital filters. *IEEE Transactions on Circuits and Systems*, 33\(4\): 357-364. <https://doi.org/10.1109/TCS.1986.1085930>](https://doi.org/10.1109/TCS.1986.1085930)
- Mahesh, R. and Vinod, A.P. (2008). [Coefficient decimation approach for realizing reconfigurable finite impulse response filters. *Proceedings of the IEEE International Symposium on Circuits and Systems*.](#)
- Mahesh, R. and Vinod, A.P. (2011). [Low complexity flexible filter banks for uniform and non-uniform](#)

channelisation in software radios using coefficient decimation. *IET Circuits, Devices & Systems*, 5(3): 232-242.

Meyer-Baese, U. (2014). *Digital Signal Processing with Field Programmable Gate Arrays*, 4th edition. Springer, Berlin.

Mitola, J. (1995). *The software radio architecture*. *IEEE Communications Magazine*, 33(5): 26-38.
<https://doi.org/10.1109/35.393001>

Vaidyanathan, P.P. (1993). *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ.

White, S.A. (1989). *Applications of distributed arithmetic to digital signal processing: A tutorial review*. *IEEE ASSP Magazine*, 6(3): 4-19. <https://doi.org/10.1109/53.29648>