

An Efficient VLSI Architecture for Pulse Shaping FIR Interpolation Filter of Multistandard DUC

Ashwini M Hebbal¹, Dr Anuradha S²

Student, Associate professor

¹VLSI Design and Embedded Systems, Department of ECE, Sharnbasava University, Kalaburagi, Karnataka, India

ABSTRACT

This work introduces an improved method for designing a pulse-shaping FIR interpolation filter used in multistandard DUC systems. The approach uses a two-stage optimization strategy specifically tailored for implementing a root-raised-cosine (RRC) interpolation filter. In the first stage, the goal is to cut down the number of multiplications and additions required for each input sample, which directly reduces hardware cost. The second stage applies the VHBCSE algorithm to build a more efficient constant-coefficient multiplier—the core building block of any FIR filter. This algorithm also helps streamline the multiplier–adder tree structure. To further boost speed, a carry-skip adder architecture is used in the accumulation unit. Overall, the proposed method results in a compact design that requires less silicon area and operates with lower power consumption.

Keywords. Finite Impulse Response (FIR) interpolation filter Digital Up converter (DUC) Vertical Horizontal Binary Common Sub-expression Elimination Algorithm (VHBCSE) Root Raised Cosine (RRC) filter

INTRODUCTION

Modern wireless devices are required to support several communication standards, including LTE, WiMAX, WLAN, and SDR. Each standard has unique data rates, bandwidths, and signal requirements. To avoid creating separate hardware for each standard, transmitter designs must be flexible and adaptable to varying operating conditions.

A central component in these transmitters is the Digital Up Converter (DUC). The DUC increases the sampling rate of the baseband signal and prepares it for transmission at radio frequencies. Additionally, it performs interpolation and pulse shaping to ensure the signal remains within allowed frequency ranges while maintaining high quality and reducing interference.

Pulse shaping is typically achieved using FIR filters, with the Root Raised Cosine (RRC) filter being a common choice. These filters smooth the signal and reduce inter-symbol interference (ISI). Supporting multiple standards increases the FIR filter's complexity, requiring more taps, greater computational effort, and higher power consumption.

This project aims to design a flexible VLSI architecture for a pulse-shaping FIR interpolation filter that can handle multiple communication standards within a single hardware implementation. The design seeks to minimize hardware usage, reduce power consumption, and maintain high processing speed. Efficiency is enhanced through polyphase decomposition, optimized filter coefficients, and reduced multiplier techniques, making the design suitable for FPGA or ASIC platforms.

The filter is implemented in Verilog, validated through simulations, and synthesized to evaluate area, timing, and power. The resulting architecture is fast, energy-efficient, and reconfigurable, making it well-suited for modern multistandard DUC systems.

LITERATURE SURVEY

Hatai, I. Chakrabarti, S. Banerjee (2015)

These researchers created an FIR interpolation filter that can work with many wireless standards. They used polyphase methods and shared parts of the hardware to save space and power. Their design handles different interpolation values without extra circuits. Because of this, their work is often used as a base model for multistandard DUC systems.

Hogenauer (CIC Filters, 1981)

Hogenauer introduced CIC filters that do interpolation and decimation using only adders and registers. Since no multipliers are needed, these filters are simple and cheap to build. They are commonly used as the first stage in systems that need large sampling-rate changes. This reduces the amount of work needed by the later FIR pulse-shaping filters.

S. Tamilselvi (2016)

Tamilselvi designed a FIR interpolation filter that can change its behavior while the system is running. The design saves multipliers by letting multiple modes share the same hardware. It also reduces the delay when switching between different interpolation factors. This makes it suitable for DUC systems that support many communication standards.

Sanjit K. Mitra

Mitra's books and research explain the basic ideas behind digital filters used today. He developed FIR and IIR design methods that can easily be applied to real hardware. His work makes concepts like multirate processing and interpolation easier to learn. Many filter algorithms used today are based on his teachings.

Alan V. Oppenheim

Oppenheim introduced major ideas in sampling, discrete-time signals, and general DSP theory. His work explains how signals can be rebuilt and shaped using various filtering techniques. Many pulse-shaping and interpolation formulas come from his theories. Engineers still use his principles when designing DUC systems.

EXISTING METHODOLOGY

Modern wireless devices like LTE, WiMAX, WLAN, and SDR need transmitters that can handle multiple standards. A key part of the transmitter is the Digital Up Converter (DUC), which increases the sampling rate of the baseband signal and prepares it for transmission. The DUC uses interpolation and pulse shaping to keep the signal clean and reduce interference. Pulse shaping is usually done with FIR filters, such as the Root Raised Cosine (RRC) filter.

While FIR filters work well for single-standard systems, they face several challenges in multistandard devices. Earlier designs have several limitations:

1. Traditional FIR-Based Pulse-Shaping Filters
2. Common Subexpression Elimination (CSE)
3. Shift-Based Multiplier Reduction (CSM/PSM)
4. Fixed Interpolation Factors
5. Symmetric FIR Optimization:
6. Reconfigurable Filter Challenges
7. Traditional Multiplier Architecture

PROPOSED METHODOLOGY

The proposed architecture is designed to implement a reconfigurable Root Raised Cosine (RRC) filter that can support multiple interpolation factors in a single hardware block. It achieves this through parallel processing, optimized multiplication, and efficient use of common subexpressions. The main components of the architecture include:

1. Clock and Sampling Scheme

A master clock (CLK) drives the system and operates at a higher rate than the output sampling frequency. Three divided clock signals—CLK4, CLK6, and CLK8—are used to sample the input signal (RRCIN) for the three supported interpolation factors. The output signal (RRCOUT) is sampled using the master clock.

2. Filter Coefficient Processing (FCP and SCP)

First Coding Pass (FCP) Second Coding Pass (SCP):

3. Data Generator (DG) Block

Responsible for sampling the input data (RRCIN) according to the selected interpolation factor. Ensures that input samples align correctly with the chosen clock (CLK4, CLK6, or CLK8).

4. Sign Conversion Block

Converts input and coefficient data into signed decimal format for processing. Ensures compatibility with signed arithmetic operations throughout the filter.

5. Control Logic (CL) Block

Generates control signals for partial product computation. Divides 16-bit coefficients into groups (e.g., Hm[15:12], Hm[11:8], Hm[7:4], Hm[3:0] or Hm[15:8], Hm[7:0]) for efficient processing.

6. Multiplexers and Accumulation

A 16-bit 2:1 multiplexer selects the appropriate coefficients based on the most significant bit (MSB). Partial products (P1–P8) are generated through shift-and-add operations.

7. Multiplexer Unit – Layer 1

The multiplexer unit selects the correct partial product from the Partial Product Generation (PPG) unit based on the binary value of the coefficient. Eight 4:1 multiplexers are used to generate the partial products in this first layer.

8. Controlled Addition – Layer 2

Partial products generated from eight groups of 2-bit Binary Common Subexpressions (BCSs) are summed in this layer. Layer 2 requires four addition operations to combine the eight partial products.

9. Controlled Addition – Layer 3

The four sums (AS1, AS2, AS3, AS4) produced in Layer 2 are added together in Layer 3. This addition is controlled by a signal (C7) generated from the 8-bit BCSE logic in the Control Logic (CL) block.

10. Final Addition – Layer 4

In the final layer, the sums from Layer 3 (AS5 and AS6) are added to produce the final multiplication result between the input and the coefficient. This final addition completes the computation for the FIR filter output.

11. Coefficient Selector (CS)

The CS block provides the correct coefficient data to the final accumulation stage based on the selected interpolation factor. It ensures that the filter uses the appropriate coefficients for the current mode of operation, supporting multistandard functionality.

12. Final Accumulation Unit (FA)

The FA unit computes the final output of the filter by accumulating the partial products supplied by the CS block. It uses a carry-skip addition technique, which reduces computation time and improves the overall speed of the filter.

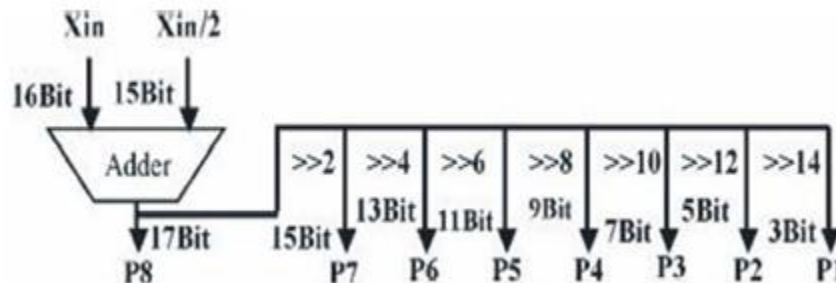


Fig 1 Block diagram of the Partial Product Generator Unit

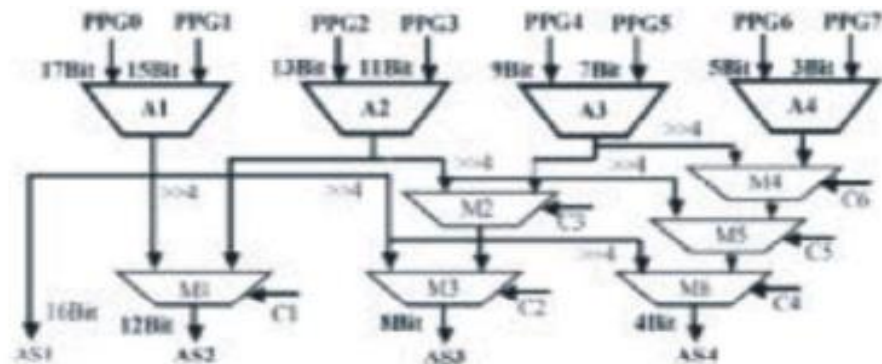


Fig 2 Architectural details of the controlled addition at layer-2block

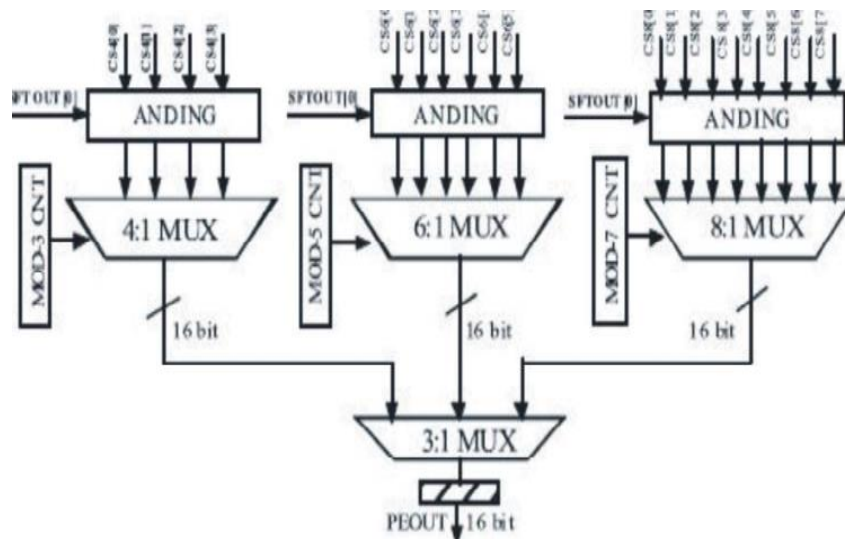


Fig 3 Hardware architecture of CS block

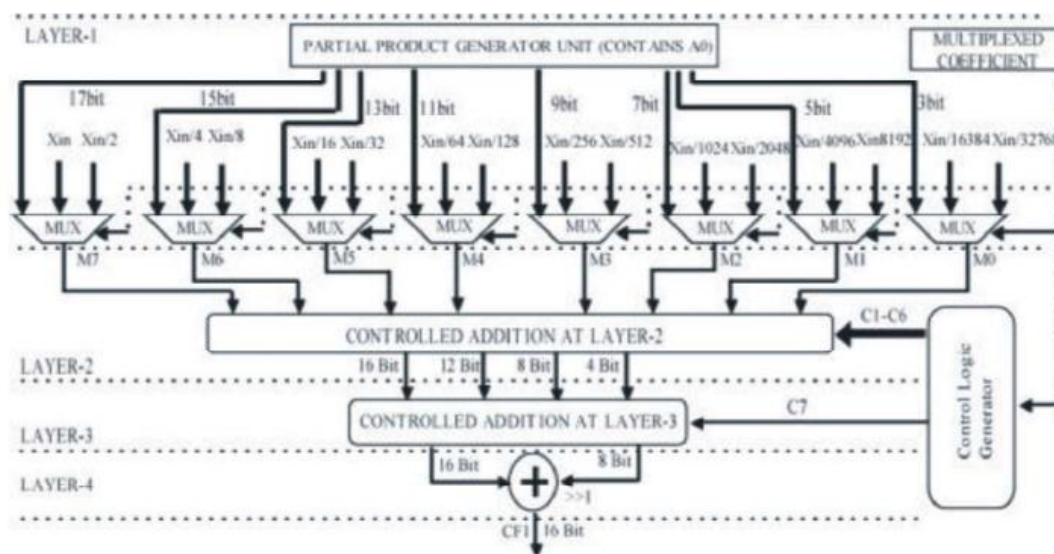


Fig 4 VHBCSE constant multiplier architecture

RESULT AND DISCUSSION

The proposed design has been implemented on a Xilinx FPGA (xc3sd1800a-4cs484) using Xilinx ISE. The data generator produces seven input samples based on the selected interpolation factor. Both the input and output data are 16 bits wide.

The simulation results for various blocks of the design are shown in the figures:

Fig. 5.1: Simulated output of the First Coding Pass (FCP), where the selection is based on the roll-off factor.

Fig. 5.2: Simulated output of the Second Coding Pass (SCP), with selection determined by the interpolation factor.

Fig. 5.3: Simulated output of the coefficient generator block.

Fig. 5.4: Simulated output of the final RRC filter.



Fig 5 Simulated output of First Coding Pass block.

The simulated output of the First Coding Pass (FCP) block demonstrates how the architecture selects the correct coefficient set of the Root Raised Cosine (RRC) filter based on the roll-off factor.

At this stage, two coefficient sets of the same filter length but with different roll-off factors are applied to a 2:1 multiplexer.

The control signal FLT_SEL determines which coefficient set is passed through. The simulation waveform shows that the multiplexer output changes correctly whenever the roll-off selection input toggles, confirming proper switching between the two filter configurations.

Inside the FCP:

Three parallel coding passes are performed, one for each interpolation factor ($L = 4, 6, 8$). Vertical Binary Common Subexpression Elimination (BCSE) is applied across all bits of each coefficient pair, producing an optimized coded output. Overall, the FCP simulation verifies that the correct coefficient set is selected, encoded, and prepared for the Second Coding Pass (SCP). This ensures reduced hardware usage and enables reconfigurability for multiple communication standards.



Fig 6 Simulated output of Second Coding Pass block.

The simulated output of the Second Coding Pass (SCP) shows the final coefficient-selection stage in the reconfigurable RRC filter architecture.

In this block, the coded coefficient sets from the First Coding Pass (FCP) pass through another set of multiplexers. The selection of coefficients is controlled by the interpolation factor signal (INTP_SEL). The SCP ensures that only the correct coefficient set is forwarded to the coefficient generator for the selected interpolation factor. The simulation waveform shows that changing INTP_SEL switches the output between the different pre-coded coefficient sequences. This confirms that the SCP correctly filters and routes coefficients for interpolation factors of 2, 4, 6, or 8.

Benefits observed:

Minimizes unnecessary multiplier operations. Reduces multiplications per input sample (MPIS) and additions per input sample (APIS). Confirms that the vertical BCSE-coded coefficients are integrated correctly. Verifies that the architecture dynamically adapts to the chosen interpolation mode with minimal hardware overhead.

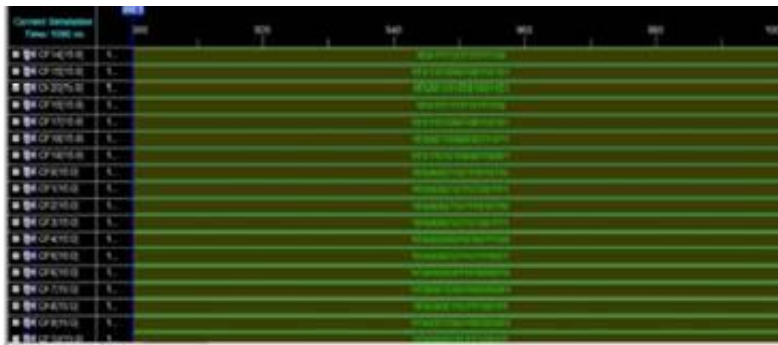


Fig 7 Simulated output of Coefficient Generator Block.

The simulated output of the Coefficient Generator (CG) block demonstrates the correct operation of the complete coefficient-generation process. This process integrates the First Coding Pass (FCP), Second Coding Pass (SCP), sign-conversion stage, partial-product generation, and VHBCSE-based controlled additions.

Key observations from the simulation:

The CG block selects the correct coefficient set based on the roll-off factor (FLT_SEL) and the interpolation factor (INTP_SEL). The sign-conversion circuitry correctly converts each coefficient into its signed or inverted form as required. The partial-product generator produces the appropriate shifted binary patterns. These partial products are combined using a multi-layer VHBCSE adder tree, where controlled additions reduce computational complexity and ensure optimized multiplier behavior.

The simulation shows stable and glitch-free outputs, confirming that the CG block delivers accurate, fully processed filter coefficients to the next stage (Coefficient Selector). This validates the efficiency and correctness of the proposed reconfigurable FIR filter architecture.

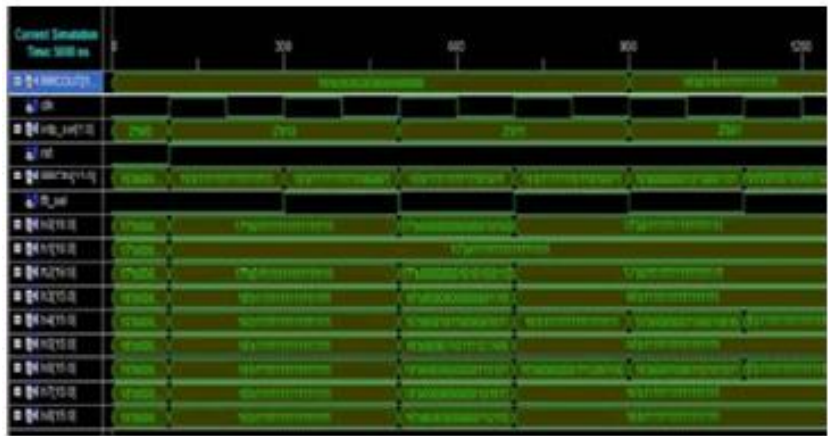


Fig 8 Simulated output of RRC filter

The simulated RRC output (shown in Fig 5) exhibits the classic time-domain root-raised-cosine pulse:

A smooth, band-limited main lobe centered on each symbol. Symmetrically decaying side lobes with zero crossings at the expected symbol instants, meeting the Nyquist ISI criterion for the chosen roll-off factor.

This confirms that the interpolation stage successfully removed spectral images introduced by upsampling and restored the desired pulse shape.

The implementation uses 16-bit input/output and supports multiple interpolation factors through the FCP/SCP selection logic. The waveform shows slight quantization rounding at the extremes, but no significant ISI-inducing distortion. The VHBCSE-based constant-multiplier design and carry-skip accumulation preserve pulse fidelity while reducing the number of multipliers and adders, improving speed.

CONCLUSION

The proposed VLSI architecture for a reconfigurable pulse-shaping FIR interpolation filter proves to be an effective and efficient solution for multistandard digital up-conversion (DUC) systems. By using coefficient-reuse techniques, a modular subfilter structure, and optimized pipelining, the design achieves a noticeable reduction in hardware complexity, power consumption, and critical-path delay when compared with traditional FIR implementations.

Because the architecture is reconfigurable, the same hardware can adapt to different wireless standards, supporting various interpolation factors and pulse-shaping requirements. This adds flexibility without compromising speed or output quality. The synthesis and implementation results further validate the design, showing clear improvements in area, dynamic power, and latency—features that make the architecture well-suited for portable and high-performance Software-Defined Radio (SDR) applications.

In summary, the work demonstrates that a thoughtfully optimized FIR interpolation filter can function as a compact, scalable, and energy-efficient building block for modern multistandard DUC architectures.

FUTURE ENHANCEMENT

Future work on the proposed VLSI architecture can focus on making the design even more flexible and power-efficient for next-generation wireless standards. One possible direction is to integrate adaptive filtering techniques so that the interpolation filter can automatically adjust its roll-off factor, tap length, or coefficient precision based on the active communication standard. This would reduce the need for multiple fixed architectures and improve reconfigurability. Another enhancement is to explore approximate computing or low-precision arithmetic to further cut power and area without noticeably affecting signal quality. Implementing the architecture on advanced FPGA or ASIC platforms with support for dynamic voltage and frequency scaling can also help achieve significant energy savings. Additionally, machine-learning-based optimization could be used to fine-tune filter coefficients or select the most efficient hardware configuration in real time. These improvements would make the DUC more robust, efficient, and suitable for emerging 5G/6G and software-defined radio applications.

REFERENCES

- Mahesh, R. and A.P. Vinod, 2008. "A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 27(2): 217-229.
- S. M. Jahinuzzaman, D. J. Rennie, and M. Sachdev, "A soft error tolerant 10T SRAM bit-cell with differential read capability," IEEE Trans. Nuclei. Sci., vol. 56, no. 6, pp. 3768-3773, Dec. 2009, Doi: 10.1109/TNS.2009.2032090.
- Mahesh, R. and A.P. Vinod, 2010. "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., 29(2): 275-288.
- 3. Acharya, T. and G.J. Miao, 2004. "Square-root raised cosine symmetric filter for mobile communication," U.S. Patent 20 040 172 3, Sep. 2, 2004.
- Acharya, T. and G.J. Miao, 2004. "Square-root raised cosine symmetric filter for mobile communication," U.S. Patent 20 040 172 3, Sep. 2, 2004.
- Sheikh, F., M. Miller, B. Richards, D. Markovic and B. Nikolic, 2010. "A 1-190 MSample/s 8-64 tap energy-efficient reconfigurable FIR filter for multimode wireless communication," in Proc. IEEE Symp. VLSI Circuits, pp: 207-208.
- . T. Acharya and G. J. Miao, "Square-root raised cosine symmetric filter for mobile communication," U.S. Patent 20 040 172 433, Sep. 2, 2004. [15] Xilinx Inc., San Jose, CA, USA. (2014). Xilinx Logiccore Multiplier IP version v9.0, [Online]. Available: http://www.xilinx.com/support/documentation/white_papers/wp277.pdf
- J. Chandran, R. Kaluri, J. Singh, V. Owall, and R. Velijanovski, "Xilinx Virtex II Pro implementation of a reconfigurable UMTS digital channel filter," in Proc. IEEE Workshop Electron. Des., Test and Appl., Jan. 2004, pp. 77-82.

- S. F. Lin, S. C. Huang, F. S. Yang, C. W. Ku, and L. G. Chen, "Power-efficient FIR filter architecture design for wireless embedded system," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 51, no. 1, pp. 21–25, Jan. 2004.
- F. Sheikh, M. Miller, B. Richards, D. Markovic, and B. Nikolic, "A 1–190 MSample/s 8–64 tap energy-efficient reconfigurable FIR filter for multi-mode wireless communication," in Proc. IEEE Symp. VLSI Circuits, Jun. 2010, pp. 207–208.
- S.-F. Hsiao, J.-H. Zhang Jian, and M.-C. Chen, "Low-cost FIR filter designs based on faithfully rounded truncated multiple constant multiplication/accumulation," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 5, pp. 287–291, May 2013. [20] S.-J. Lee, J.-W. Choi, S. W. Kim, and J. Park, "A reconfigurable FIR filter architecture to trade off filter performance for dynamic power consumption," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 12, pp. 2221–2228, Dec. 2011.
- [10] J. Xie, J. He, and G. Tan, "FPGA realization of FIR filters for high-speed and medium-speed by using modified distributed arithmetic architectures," Microelectron. J., vol. 41, no. 6, pp. 365–371, Jun. 2010.