2581-4575
International Journal For Recent Developments in Science & Technology
Crossref
A Peer Reviewed Research Journal

Implementation of Finite Impulse Response Filter Using Multi Level Arithmetic

J. Harilal[1],Moyyala Sravani[2],Avusula Uday Chary[2],Banoth Tharun[2],Prabaharan Vibu Nandhan[2]

[1]Assistant Professor,[2]UG Student,[1,2]Department of Electronics & Communication Engineering
[1,2]Malla Reddy Engineering College and Management Sciences,Kistapur,Medchal,501401, Telangana

ABSTRACT

The demand for efficient signal processing has led to continuous advancements in the design of digital filters. This paper focuses on the implementation of a Finite Impulse Response (FIR) filter using Multi-Level Arithmetic, incorporating adders, multipliers, and D-type flip-flops (DFFs). The conventional system, which relies on basic binary arithmetic, often faces challenges in achieving high-speed processing and reduced power consumption. This paper introduces a novel approach leveraging multi-level arithmetic to enhance the performance of FIR filters. The proposed system aims to address the drawbacks of the conventional system by exploiting the advantages of multi-level arithmetic operations. The incorporation of adders, multipliers, and DFFs at multiple levels facilitates more efficient computation, enabling higher processing speeds and improved energy efficiency. The proposed system's design and implementation are discussed in detail, showcasing its potential to outperform traditional FIR filter implementations. Experimental results and comparative analyses demonstrate the efficacy of the proposed approach in terms of speed, power consumption, and overall filter performance. This research contributes to the ongoing efforts in advancing digital signal processing techniques and offers a promising avenue for optimizing FIR filter implementations in various applications.

Keywords: Finite impulse, multi level arithmetic,D-type flip-flops.

1. INTRODUCTION

On a VLSI device, the complexity of the signal processing systems that are implemented is increasing all the time because the scale of integration is growing all the time. Not only do these signal processing applications need a large calculation capability, but they also use a significant amount of power. In today's VLSI system design, performance and area are still the two most important design considerations; nevertheless, power consumption has emerged as an issue of paramount importance []. There are two primary drivers that contribute to the need for low-power VLSI systems. First, as the operating frequency and processing capacity per chip continue to steadily increase, big currents need to be provided, and the heat caused by high power consumption has to be removed using appropriate cooling methods. Both of these challenges must be met. Second, the amount of time that portable electronic gadgets may operate on a single charge is restricted. A design that consumes less power results in these portable gadgets having an operational life that is much longer. Addition is a fundamental mathematical operation that, in most cases, has a significant influence on the overall performance of digital systems. DA are the most common kind of calculator used in electronic applications. Multipliers and digital signal processors (DSPs) utilise these components to carry out a variety of algorithms, such as FFT, FIR, and IIR,

respectively. DA are introduced into the discussion whenever the notion of multiplication is discussed. As is well knowledge, microprocessors are capable of carrying out millions of instructions in a single second. When it comes to the design of multipliers, the most crucial factor to take into account is the maximum possible processing speed. Miniaturization of the gadget should be prioritized, and power consumption should be kept to a minimum, so that it may be easily transported. Mobile phones, laptops, and other electronic devices demand a larger battery backup.Therefore, a VLSI designer has to make sure that these three design parameters are optimized. Due to the extreme difficulty of satisfying these limitations, it may be necessary, depending on the nature of the need or the intended use, to strike a balance between them. The RCAs have the smallest footprint of any of the three designs, but they are also the most cumbersome. while the carry glance ahead is the most efficient way to go, but it takes up more space. HSCG function as a kind of middle ground between the two other types of DA. By giving a hybrid carry look-ahead/HSCG design in 2002, Wang et al. introduced a novel idea of hybrid DA with the intention of accelerating the process of adding numbers. The year 2008 saw the presentation of low power multipliers that are based on innovative hybrid full DA. One of the most significant areas of study in VLSI system design is the creation of area- and power-efficient high-speed data route logic systems. This is one of the design goals of the field. The speed of addition that may be achieved with digital DA is restricted by the amount of time that is necessary to propagate a carry through the adder. A basic adder generates the total for each bit location in a sequential fashion only after the sum for the preceding bit position has been computed and a carry has been passed on to the next place in the adder. By separately producing numerous carries and then selecting a carry to produce the total, the CSLA is utilized in many computing systems to solve the issue of carry propagation delay. This is accomplished by selecting a carrier to generate the sum.

2. Literature survey

Akhter et al. [11] made the suggestion that a FIR filter that was based on distributed arithmetic should be designed and analysed. At first, the process of multiplication, which requires the input data to be multiplied with the preset coefficients, required a considerable amount of space for the temporary data it was storing. An effective LUT-less Modified Distributed Arithmetic architecture was suggested by Narendrian et al. [12] in their work. In the beginning, this design does away with the need for precomputing weighted sums that are necessary for the LUT in a DA that uses multiplexers and adders. A filtering application was built by Radhakrishnan et al. [113] utilising a hybrid adder, a modified Booth multiplier, and a multiple-and-accumulate (MAC) unit. The hybrid adder initially integrates two or three of the most prevalent adder schemes, which ultimately results in a considerable gain in speed of operation and a reduction in power consumption. In a similar manner, for the purpose of multiplication, we chose to implement the modified Booth multiplier, which makes use of a radix-4 encoding technique. In order to construct a MAC unit that is capable of doing addition and multiplication simultaneously, the developed adder and multiplier are put to use. A digital FIR filter with 8-MAC units is created by making use of the established MAC unit as a building block. Cadence Virtuoso with 180 nm, 90 nm, and 45 nm TSMC technology libraries are used to conduct the final performance metrics measurements. These measurements include delay and power consumption. An arithmetic optimization of reconfigurable digital FIR filters was presented by Mahalakshmi et al. [14] in order to minimise the amount of power that was used.

According to the results of the mathematical study, it is possible to realise considerable cost reductions in terms of hardware by using the design that was provided. When compared to the usual technique, the ASIC implementation that was done using TSMC 65 nm technology demonstrates that the suggested methodology takes up less space and uses less power than the conventional way. A modified Montgomery multiplier design was developed by Thanmai et al. [15] and its implementation was suggested for the fifth order FIR filter. The number of transistors in the traditional design of the multiplier has been reduced by a considerable amount thanks to the implementation of the suggested compressor, which has also enabled the design to achieve its full potential in terms of its space use. CMOS and PTL logic implemented in 45 nm technology are used throughout the whole of the design simulation process. In compared to the traditional design, the multiplier has a power reduction of around 68% and an area efficiency of 65%, both of which are significant improvements. Additionally, there is an increase in the results that are produced.

The FIR filter was first suggested by Chinnapparaj and colleagues [16]. Due to the fact that FIR has this capability, it is possible for it to be used in audio applications. The finite input and output of a FIR filter is a stable quality that is also particularly beneficial in digital signal processing (DSP). datta, et al. [17] suggested a pipeline-based FIR architecture by making use of the DA approach. A parallel process is used to gather the bits of the input, with one word representing each interval. Because of the new and better design, the FIR filter's overall performance has been significantly accelerated. In conclusion, the findings of the implementation reveal that the modified DA-based FIR filter with 8-bit input and 16-coefficients is more efficient than other modern designs in terms of both the amount of space it requires and the amount of processing power it provides.

The multiplier-less multiplication approach was first suggested by Ramamoorthy and colleagues [18]. Instead of using multiplication, the researchers use a procedure called shift and add. The proposed structure makes use of both the direct and the transposed structures of the FIR filter design. This is done in order to avoid the potential conflict that would result from having to pick whether to utilise the direct and transposed structure or one of the other structures.

The Parallel FIR Digital Filter that Narendran et al. [19] presented is based on the Even Symmetric Fast FIR Algorithm and utilises a variety of adders. In the beginning, the objective of the suggested structure is to provide area-optimized adders. These adders are intended to have symmetric coefficients, which lowers the overall hardware cost of the design. In addition, the filter tap must be of even order. The RCA needs reduced space consumption compared to the other three adders, which is taken into consideration in the performance analysis report for a three parallel 96-tap filter that is based on the adder described above.

Kumar et al. [20] suggested a novel architecture for a 2-D block FIR filter that makes use of the DA method. This technique is well-known for its ability to efficiently build a multiply and accumulate block. In the beginning, a hardware-based design is presented for the DA lookup table, also known as the DA-LUT. This architecture makes the architecture of the 2-D FIR filter adjustable. In addition, since the processing is done in blocks, sharing occurs across the DA-LUTs at several levels. Therefore, it is possible to create a common DA-LUT that can be used for block inputs, which will decrease the complexity of the hardware required for DA-LUT. In addition, the suggested design makes advantage of memory overlap in order to decrease the number of systolic structures required in comparison to previous systems.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

3. Proposed Method

3.1 Introduction

Image processing, video processing, audio processing, and even telecommunications are some of the many fields that have found widespread usage for digital signal processing systems. [1] In point of fact, the DSP engages in many distinct kinds of activities. Filtering is one of the processes that is used the most often [2]. The process of filtering removes components from a signal that are not vital to its operation [3].

In the research that has been done, reconfigurable FIR filters as a means of reducing power usage have been offered. The implementation of FIR filters often involves the use of multipliers, adders, and delay components [4]. Because of the complexity of these procedures, it is almost always desired to cut down on the number of calculations by using specialised approaches and algorithms [5]. By employing algorithmic strength reduction, it is feasible to reduce the computational complexity of filters and other devices that rely heavily on arithmetic [6]. If the altered implementation techniques make advantage of this strength decrease, they may reduce the computational complexity while simultaneously enhancing performance [7]. One example of one of these changed implementation approaches is quick linear convolution [8]. Iteratively dividing a huge convolution into a number of smaller convolutions is the method behind the rapid convolution methods [9]. In addition to this, the idea of Lagrange interpolation is used in order to reduce the number of times that the variable is multiplied [10].

The Modified Winograd Algorithm has been successfully used in order to lower the overall number of computations as well as the amount of space that is needed for parallel FIR filters. Therefore, the memory-based multiplication approach replaces multipliers in order to lessen the system's need for space and its response time. One of the memory-based techniques is known by its acronym DA, which stands for distributed arithmetic. DA based approach replaces multipliers in FIR filters. In addition to that, the architecture is made to accommodate a wider variety of input values. In addition to this, a 16-Tap FIR filter is developed, synthesised using Xilinx ISE, and implemented for an XC4VSX35-FF668-10 based FPGA so that the performance of this architecture may be evaluated. In conclusion, the results of the implementation demonstrate that the design has less of an impact on the resources available and is able to accomplish quicker filtering than earlier implementations of the filter.

3.2 Pipelining and Retiming Topologies

The critical path may be shortened with the pipelined technique since it incorporates additional latches between the multipliers. When compared to the comparable route in the original sequential circuit. This is because M is the number of levels in the pipeline. In spite of the fact that it shortens the critical path, the result is an increase in latency. The critical path, which is the path that takes the largest amount of time, may be shortened in the DSP design by correctly arranging the pipelining latches.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

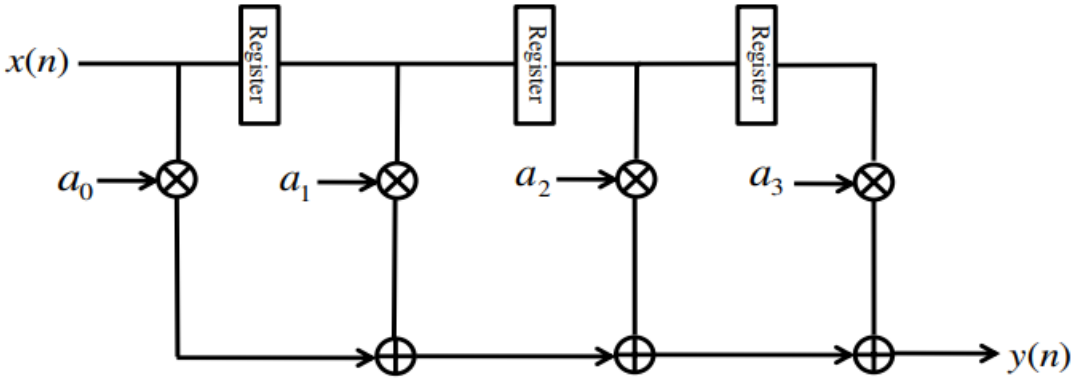


Figure 1. Pipelining design for RNS-based Design.

Retiming is a technique of transformation that may change the placements of delay elements in a circuit without altering the characteristics of the circuit's inputs and outputs. By reducing the amount of time needed to calculate the critical path, retiming makes it possible to increase the clock rate of a circuit. Retiming is an additional method that may be used to cut down on the number of registers in a circuit. In synchronous circuit design, retiming may serve a variety of purposes. This technique has a variety of applications, some of which include reducing the power consumption, clock period, and clock period of a circuit, as fine as performing logic synthesis. By using the retiming algorithmic approach, it is possible to reduce the number of crucial accesses, the register count, as well as the power consumption of the filter design. Because adding additional levels of pipelining would result in an increase in latency, it can only have a maximum of two layers. The two-level pipelining will result in the whole filter being cut in half, creating two distinct halves. It is important to emphasise that the rescheduling will not result in any delay. The pipelining delay dp divides the addition and multiplication processes, which in turn limits the critical path delay to T_{max}.

$$T_{CP} = T_{max} = max\{T_m, T_a\}$$

Where Tm represents the worst-case combinational route delay of the part that deals with multiplication and Ta represents the worst-case combinational path delay of the section that deals with addition. When using shorter filter lengths

$$T_{CP} = T_m$$

As the length of the filter grows, the entire addition period Ta will eventually surpass the time required for multiplication, Tm. That is the

$$T_{CP} = T_a$$

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

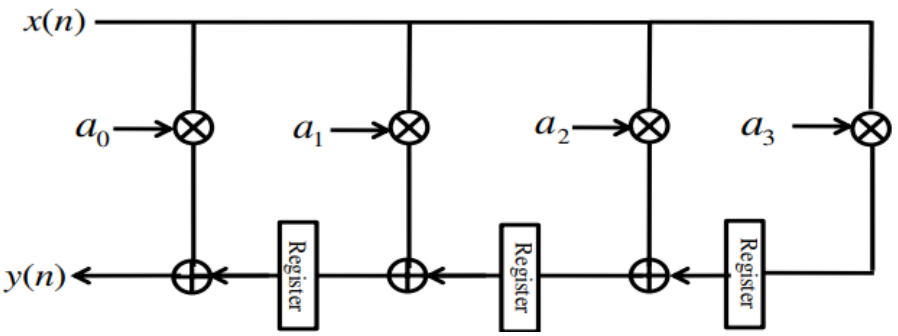
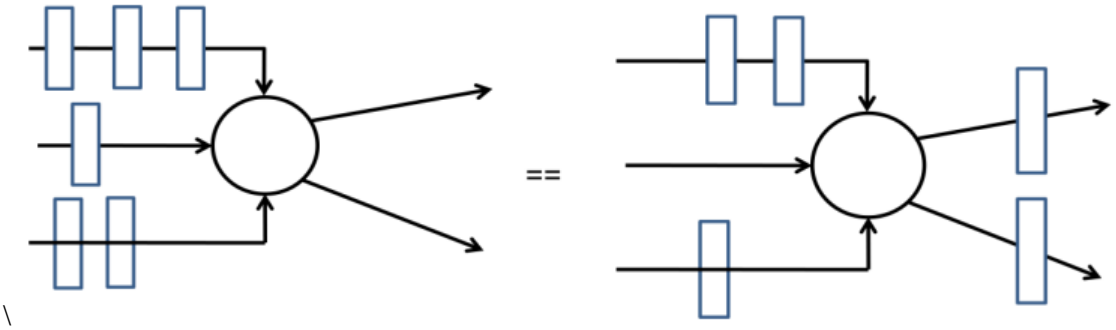


Figure 2. The FIR Filter's retiming circuit.

HIn this scenario, the addition procedure has had its timing adjusted in order to reduce the worst-case addition latency. The FIR filters shown here are the result of developing them using this approach. There are two separate forms of retiming, and those are cutset and pipelining. These retiming strategies will result in a reduction in both the clock dated and the amount of records that are necessary for the construction of the circuit.



\

Figure 3. The RNS-based FIR Filter retiming sample.

The retiming of digital circuits is a crucial and very successful method for increasing their overall performance. This may be done by moving the locations of the delay components. It helps to cut down on the clock period of the circuit, as well as the amount of power used, besides reason synthesis. It is put to use to increase the clock rate of the circuit. In addition to this, it is used to lessen the amount of switching, which may lead to an increase in the amount of power that is wasted in static CMOS circuits. These systems make use of a variety of processing techniques, such as pipelining, parallel processing, retiming, unfolding, and many more.

3.3 Proposed FIR filter's architectural design

In this section, this architecture was designed so that the FIR filters may be configured in a variety of different ways. Figure 4 depicts the FIR digital channelizer as seen from the standpoint of the user. The length of each channel filter is equal to N, and this channelizer is constructed up of R-channel filter coefficients. An array of LUTs is used to hold all of the potential filter coefficient combinations that may be applied. In all, there are N different LUTs.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

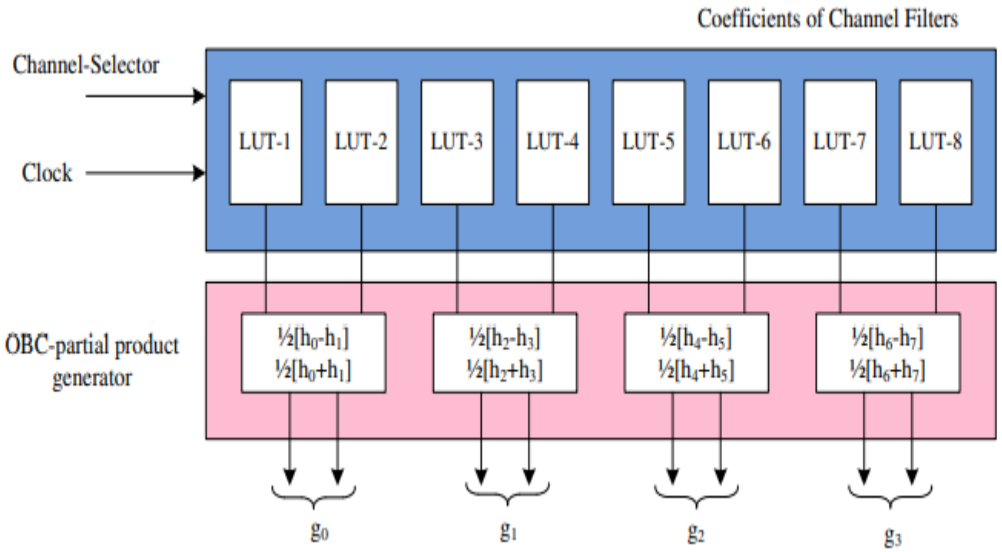


Figure 4: Proposed MCM-DA based FIR Filter

Following this, the MCM-DA is used throughout the whole process of translating the filter coefficients into partial products as a point of reference. In light of this, it is absolutely necessary to produce the MCM-DA-based partial products before putting them to use in the filter processing blocks. In this scenario, the deconstruction of an MCM-DA-based DA-LUT is taken into consideration with the goals of reducing the memory capacity required to preserve the MCM-DA contents and simplifying the procedure. These aims are taken into account in order to: The component that determines the level of complexity is the square root of N, which is equal to 8/2, which is equal to 4.

The emergence of two MCM-DA partial product terms is shown to be the consequence of translating these four filter coefficients, as seen in Figure 5. The MCM-DA partial products generating block is the one that is responsible for producing MCM-DA-based partial inner products at the beginning of each clock cycle. This task is performed by the MCM-DA partial products generating block. At the beginning of each cycle of the clock, the filter coefficients hi are sent to the MCM-DA partial product generator in order to be generated. As a consequence of this, N/2 is produced, which is equivalent to four distinct sets of partial inner products that are important to MCM-DA.

These products may be identified by the notation gj, and the range of values that are considered acceptable for j is from 0 to N/2 minus 1. Each group of partial products has two nonzero MCM-DA product terms of filter coefficients that make up the group. Each group is constructed using these filter coefficients. Expressions such as 12[h(2j) h(2j + 1)] and 12[h(2j) + h(2j + 1)] are examples of products that are only partially formed.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

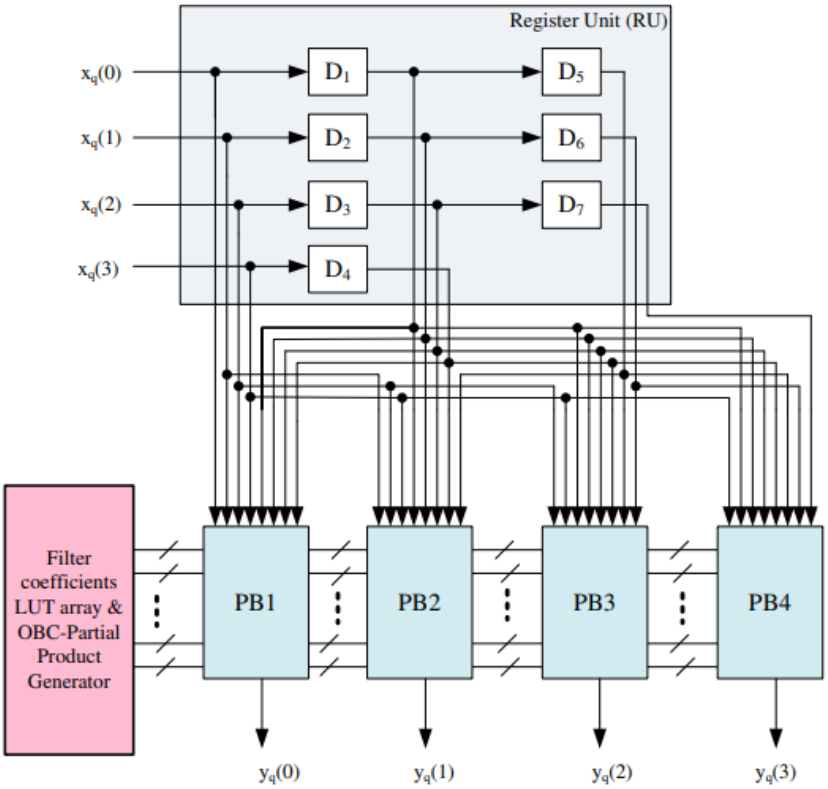


Figure 5: The projected MCM-DA architecture.

The block-based direct form systolic FIR filter that was constructed previously is shown in its fundamental design in figure 6, which may be seen below. During this process, a vast number of samples that are identical to one another will be generated. In order to reduce the amount of memory registers that are required for the filter processing, these samples are recycled and utilised again. It is the duty of the RU to produce LN times the whole number of samples, which are subsequently used using L times the amount of PBs. In addition, the partial products g0 through g3 that were generated by the partial product generator are applied in parallel to the PBs in order to achieve the outcomes that have been envisioned. Each component of a partial product set consists of a pair of vectors that contain the coefficients. This is the case with each and every partial product collection. After the L number of operations have been carried out on the input samples and the filter coefficients, the L number of PBs will be accountable for the production of the L number of outputs.

International Journal For Recent Developments in Science & Technology
Crossref
A Peer Reviewed Research Journal
2581-4575

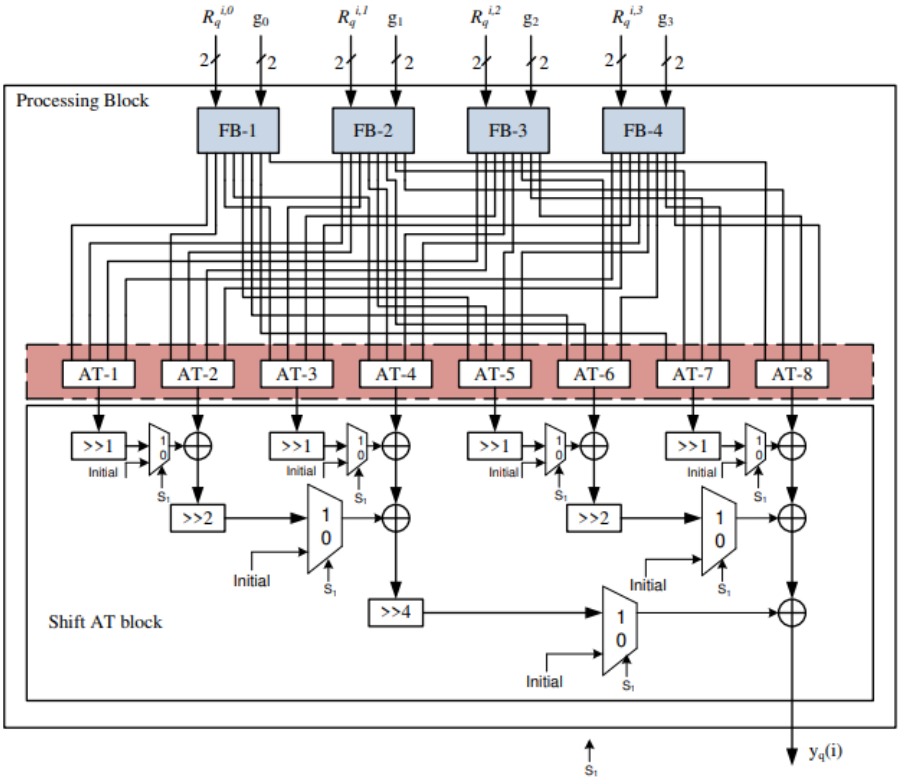


Figure 6: Internal organisation of the FIR processing block.

An L-sized array of Functional Blocks (FB) as shown in Figure 7, an adder tree-shaped array of Accumulator Shift Blocks (AT), and shift-accumulators come together to form the Processing Block (PB). The input samples, which have been decomposed, as well as the mean of two samples and two partial products based on MCM-DA are applied to each FB. Additionally, the mean of two partial products is also applied.

Each PB is responsible for the same set of responsibilities; for instance, if L is equal to four, then each of the four PBs that contribute to this set of outputs is composed of four FBs. Processing the input samples and filter coefficients in each FB is accomplished with the help of multiplexer logic. Input sample bits are used to determine which MCM-DA partial products to produce. Another MUX, the state of which is determined by the sample bit that is now being input, inverts the signal that is produced by the first MUX. In this case, a single control signal known as S0 is evaluated to decide whether the original MUX output or the inverted output is to be selected. In a similar manner, B-1 combination logics are carried out, which results in the production of B amount of output lines. These output lines are then provided to B amount of AT blocks. In each of the PBs and FBs that make up the filter architecture, the exact identical processing is carried out.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

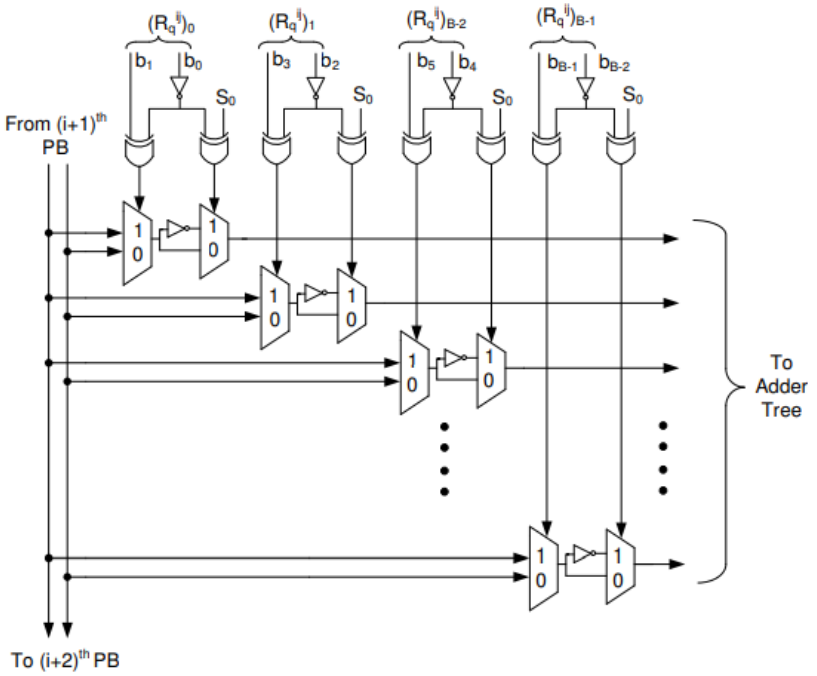


Figure 7: Function Block in PB Organization.

4. SIMULATION RESULT

This section gives a detailed analysis of various simulation results, which are implemented using XILINX-ISE software and Verilog programming. Figure 10 shows the simulation waveforms where MCM-DA-FIR functionality is justified. The design description (RTL schematic) of the suggested technique may be seen in Figure 11. The timing breakdown of the suggested technique is shown in Figure 12. In this situation, the suggested technique incurred a total time delay of 6.7690ns, of which 5.759ns was considered to be a logical delay and 1.01ns was considered to be a route delay. The report of the suggested MCM-DA-FIR's power consumption may be seen in Figure 13. In this case, the suggested MCM-DA-FIR had a power consumption of 0.065. Figure 8 shows the design summary of the proposed method. Table 1 compares the performance of various approaches. Here, the proposed MCM-DA-FIR filter resulted in reduced LUTs, power consumption, Flip-Flops, and delay, as compared to conventional Conv-FIR filter [12], Fixed FIR filter [17], DFG FIR filter [13].

Fig. 8. Simulation Results.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

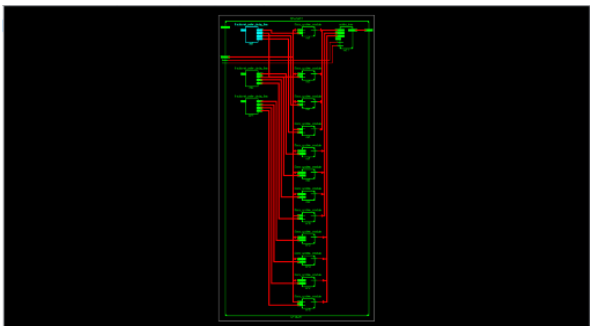


Fig. 9. RTL Schematic

```
================================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk_out'
  Total number of paths / destination ports: 9 / 7
--------------------------------------------------------------------------------
Offset:              6.769ns (Levels of Logic = 2)
  Source:            C1/out_7 (FF)
  Destination:       out<8> (PAD)
  Source Clock:      clk_out rising

  Data Path: C1/out_7 to out<8>
                                Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q               2   0.591    0.590  C1/out_7 (C1/out_7)
     LUT2:I0->O            1   0.648    0.420  P1/Mxor_b<8>_Result1 (out_8_OBUF)
     OBUF:I->O                 4.520           out_8_OBUF (out<8>)
    ----------------------------------------
    Total                      6.769ns (5.759ns logic, 1.010ns route)
                                       (85.1% logic, 14.9% route)

================================================================================
```

Fig. 10. Time summary.

Device	
Family	Zynq-7000
Part	xc7z010
Package	clg400
Temp Grade	Commercial
Process	Typical
Speed Grade	-2

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	8 to 11
Custom TJB (C/W)	NA

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	34	17600	0
Signals	0.000	75	---	---
IOs	0.000	50	230	22
Leakage	0.065			
Total	0.065			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	4.0	84.7	25.3

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.005	0.000	0.005
Vccaux	1.800	0.006	0.000	0.006
Vcco18	1.800	0.001	0.000	0.001
Vccbram	1.000	0.000	0.000	0.000
Vccpint	1.000	0.020	0.000	0.020
Vccpaux	1.800	0.013	0.000	0.013
Vcco_ddr	1.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.065	0.000	0.065

Fig. 11. Power summary

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	12	126800	0%
Number of Slice LUTs	18	63400	0%
Number of fully used LUT-FF pairs	9	21	42%
Number of bonded IOBs	18	210	8%
Number of BUFG/BUFGCTRL/BUFHCEs	1	128	0%

Fig. 12. Device utilization summary.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

Table 1. comparison table

Parameter	Conv FIR filter [2]	Fixed FIR filter [7]	DFG FIR filter [3]	Proposed FIR filter
Time delay (ns)	20.118	13.35	10.34	0.761
Power utilized (uw)	1.293	2.356	1.46	0.065
Lookup tables	271	562	72	42
Flip flops	237	395	103	59

5. **CONCLUSION**

6. One crucial component of a FIR filter is the multiply and accumulate unit. The effectiveness of a FIR filter is determined by its Multiply and accumulate unit. The way the MAC unit is built, from the programming stage to the final implementation, determines a great many parameters, including area, power consumption, speed, and other crucial elements. The calculations presented in this article use the MCM-DA technique. In order to make the most of MCM-DA schemes and the associated pipelining for the area-delay efficient implementation of high order FIR filters for both fixed and reconfigurable applications, this study investigates the feasibility of realising a block FIR filter in a transpose form configuration. The register complexity of a transpose-form FIR filter is analysed numerically, and a flow diagram for such a filter is derived. This transpose-form FIR filter is formulated in a block form. Creating a block filter design in transpose form for flexible use. The MCM-DA scheme provides a simple design technique for implementing FIR filters in blocks. As can be shown in the simulation results, the suggested MCM-DA based FIR filter has lower area, latency, and power consumption than competing FIR filters.

References

[1] Jiang, Honglan, et al. "A high-performance and energy-efficient FIR adaptive filter using approximate distributed arithmetic circuits." IEEE Transactions on Circuits and Systems I: Regular Papers 66.1 (2018): 313-326.

[2] Praveen Sundar, P. V., et al. "Low power area efficient adaptive FIR filter for hearing aids using distributed arithmetic architecture." International Journal of Speech Technology 23.2 (2020): 287-296.

[3] Venkatachalam, Suganthi, et al. "Design and analysis of area and power efficient approximate booth multipliers." IEEE Transactions on Computers 68.11 (2019): 1697-1703.

[4] Vahdat, Shaghayegh, et al. "TOSAM: An energy-efficient truncation-and rounding-based scalable approximate multiplier." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27.5 (2019): 1161-1173.

[5] Cai, Fuxi, et al. "Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks." Nature Electronics 3.7 (2020): 409-418.
[6] Ying, Zhoufeng, et al. "Electronic-photonic arithmetic logic unit for high-speed computing." Nature communications 11.1 (2020): 1-9.
[7] Leon, Vasileios, et al. "Walking through the energy-error Pareto frontier of approximate multipliers." IEEE Micro 38.4 (2018): 40-49.
[8] Soares, Leonardo Bandeira, et al. "Design methodology to explore hybrid approximate adders for energy-efficient image and video processing accelerators." IEEE Transactions on Circuits and Systems I: Regular Papers 66.6 (2019): 2137-2150.
[9] Ankit, Aayush, et al. "Panther: A programmable architecture for neural network training harnessing energy-efficient reram." IEEE Transactions on Computers 69.8 (2020): 1128-1142.
[10] Bashar, Manijeh, et al. "Energy efficiency of the cell-free massive MIMO uplink with optimal uniform quantization." IEEE Transactions on Green Communications and Networking 3.4 (2019): 971-987.
[11] Akhter, Shamim, Satyendra Kumar, and Divya Bareja. "Design and analysis of distributed arithmetic based FIR filter." 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2018.
[12] Narendiran, S., and E. P. Jayakumar. "An Efficient Modified Distributed Arithmetic Architecture Suitable for FIR Filter." 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, 2021.
[13] Radhakrishnan, S., et al. "Design of Low Power and High Speed MAC based FIR Filter using Hybrid Adder and Modified Booth Multiplier." 2020 5th IEEE International Conference on Emerging Electronics (ICEE). IEEE, 2020.
[14] Mahalakshmi, R., and T. Sasilatha. "An Improved Digital FIR Filter Design Using Fast FIR Algorithm and Modified Carry Save Addition." National Academy Science Letters 41.3 (2018): 147-150..