International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

FIR Filter Design Based on CSD and MSD Concepts for Fixed Applications

Raseena .K .A[1], Vincy Mathew[2]

[1]Malabar College of Engineering and Technology, Department of Electronics and Communication, Thrissur, India
[2]Assistant professor, [1]Malabar College of Engineering and Technology, Department of Electronics and Communication, Thrissur, India

Abstract: ***In some applications the coefficients of FIR filters are remain fixed. This paper describes the design of FIR filter for fixed applications. The FIR filter design using transpose form structure is usually a pipelined architecture and it holds MCM (multiple constant multiplications) technique. This MCM technique results in large computation saving. In this paper, the speed of FIR filter is increased by combining CSD (canonic signed digit) and MSD (minimal signed digit) representation of numbers. The MSD representation is suitable for common sub expression elimination method. It significantly reduces the number of adders needed for in the filter design process. The proposed structure significantly reduces the area delay product (ADP) and energy per sample (EPS) than the existing FIR structure.***

Keywords: FIR, MCM, CSD, MSD, ADP, EPS

1. Introduction

Many applications require FIR filters of large order to meet the stringent frequency specifications [2]–[4]. Very often these filters need to support high sampling rate for high-speed digital communication [5]. The number of multiplications and additions required for each filter output, however, increases linearly with the filter order. Chen and Chiueh have proposed a canonic sign digit (CSD)-based RFIR filter, where the nonzero CSD values are modified to reduce the precision of filter coefficients without significant impact on filter behavior. But, the reconfiguration overhead is significantly large and does not provide an area-delay efficient structure. The architectures are more appropriate for lower order filters and not suitable for channel filters due to their large area complexity. We explore the possibility of realization of block FIR filter in transpose form configuration in order to take advantage of the MCM schemes and the inherent pipelining for area-delay efficient realization of large order. The main contributions of this paper are as follows.

1) A low-complexity design method using MCM scheme for the block implementation of fixed FIR filters.
2) CSD and MSD representations of numbers increase the speed of FIR filter.
3) The proposed structure significantly reduces the area delay product (ADP) and energy per sample (EPS) than the existing FIR structure.

2. Filter Design

The input for the ADSL circuit requires 12 bits at 4 MHz. The ADC can be realized with a multistage multi bit sigma-delta modulator as suggested by Azadet, and followed by a suitable decimation filter to reduce the sampling rate and achieve the required resolution. We use a decimation factor of 16 to decimate from 64 MHz to4 MHz. The pass band of the filter is 2 MHz with a pass band ripple of 0.001, which corresponds to the flat response of the filter. The stop band attenuation is specified according to the output word resolution. Considering 6 dB per bit, a minimum of -72 dB of attenuation is required to support the accuracy of our 12-bit output.

After the decimation filter is decided, we use the VHDL to implement the hardware. As the input signals are fed from the sigma-delta modulator to the multiplexer, which is used to choose which one input signal will be sent to the comb filter, the decimation filter will process the input signal with down rate of 128. Note that the comb filter works with larger value of down rate (=16), whereas that of FIR is 2. The main function of FIR is to strain the high-frequency noise. Furthermore, we need to emphasize that this research is focused on the integration of comb filter and low pass (LP) finite impulse response (FIR) to implement an application-specific integrated circuit (ASIC).

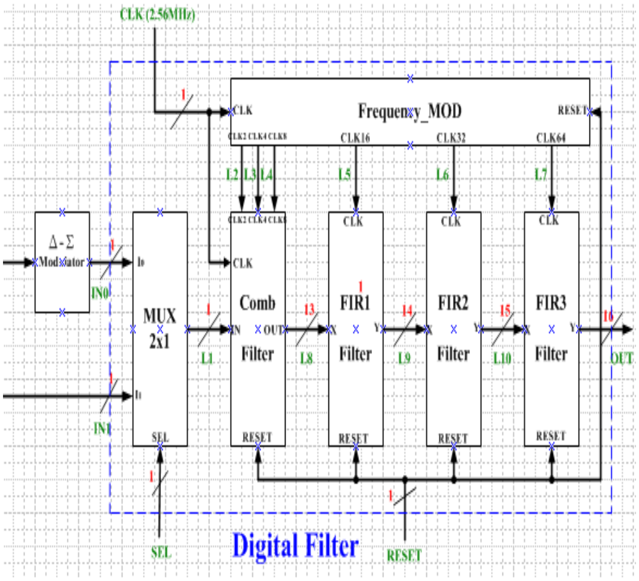


Figure 1: Digital filter design

In order to verify the frequency-divided function of decimation filter, we firstly use the MATLAB software to complete this verification. Fig. 1 shows the Simulation block of decimation with MATLAB software. Note that the input signal is operating with 2.5 kHz. After oversampling, the sampling frequency is up to 2.56 MHz and feeds to the input of the first stage of FIR filter. The input 2.56 MHz signal will be divided by 128 and send to the output of the decimation filter. That is, the output frequency is 20 kHz. This means that the high-frequency noise which is higher than 20 kHz will be removed from the decimation filter. In practice, if we feed the decimation filter with frequency of 2.5 kHz, the main frequency, 2.5 kHz, is filtered out obviously at the output port of the decimation filter.

In addition to the architectural-level technique, circuit-level techniques are also presented and used in the FIR filter implementation. In the CSHM structure, adders are critical for performance. A new carry-select adder, which is based on the dual transition skewed logic (DTSL), is presented and efficiently used in our filter implementation. The proposed carry-select adder based on DTSL is superior to the Domino-based carry-select adder in terms of power and performance. Flip-flops are also crucial elements from both a delay and power standpoint. Conditional capture flip-flop (CCFF) is explained and used in our filter design. CCFF is a dynamic style flip-flop that has a negative setup time and small clock-to-output delay. Moreover, depending on data switching activity, CCFF also reduces the power consumption.

Using the CSHM presented in the figure 2, a 10–tap FIR filter with programmable coefficients has been implemented for fabrication. FIR filter can be implemented in direct form (DF) [1] or transposed direct form (TDF) architecture.

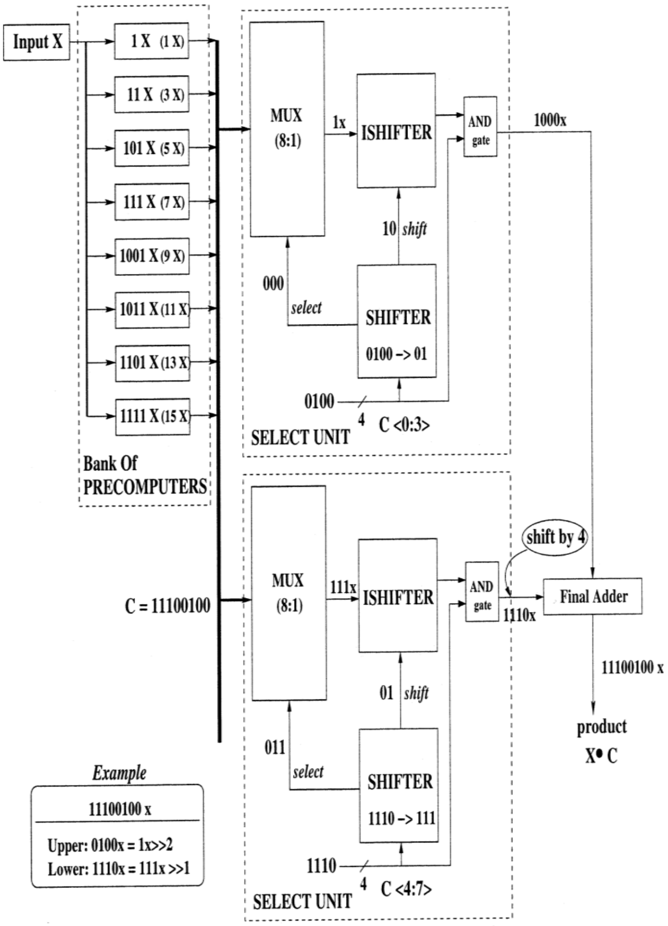


Figure 2: Computation sharing multiplier (CSHM) architecture

In the DF FIR filter, a large adder in the final stage lies on the critical path and it slows down the FIR filter. For high-performance filter structure, TDF is used in our implementation. Floor planning was done to minimize the total interconnect lengths, especially for global signals. The pre computer is placed in a rectangular area on the top of the floor plan so that its outputs can be distributed to all the taps through shortest possible paths. The power supply of the core is separated from the power supply of PADs to be able to separately measure the power of the filter core and the power of the PADs and interfacing to the testing instrument.

3. General Background

The MCM-based structure for FIR filters for block size L = 4 is shown in Fig. 3 for the purpose of illustration. The MCM-based structure (shown in Fig. 3) involves six MCM blocks corresponding to six input samples. Each MCM block produces the necessary product terms. The sub expressions of the MCM blocks are shift added in the adder network to produce the inner-product values (rl, m), for 0 ≤l ≤L −1 and 0 ≤m ≤(N/L) −1. The inner-product values are finally added in the pipelined adder unit (PAU) to obtain a block of filter output.

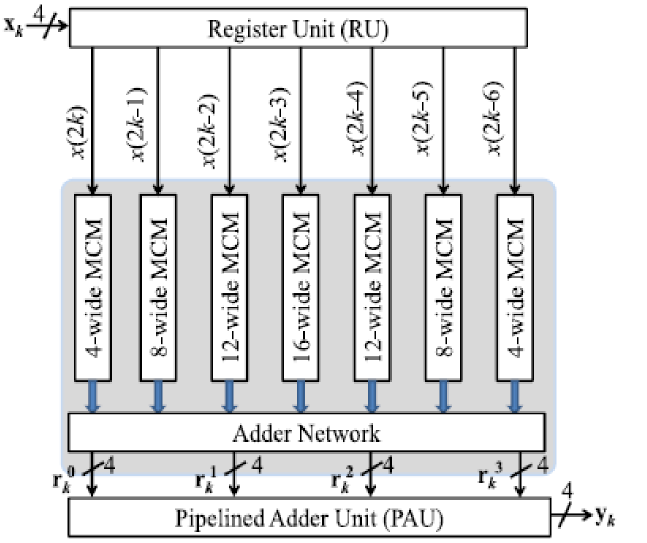


Figure 3: MCM-based structure for fixed FIR filter of block size L = 4 and filter length N = 16.

Typical DSP algorithms involve large number of multiplications, and multipliers consume significant amount of area and computation time. It is therefore important to reduce the area and time complexity of implementations of multipliers. In some applications such as linear transformations and transposed form finite impulse response (FIR) filters, the same variable is multiplied by a set of constant coefficients, which are known a priori. Such structures are referred to as multiple constant multiplications (MCM). [2]

Efficient implementation of MCM is important for high-speed, low-complexity and low-power DSP systems. Typically the multiplications in an MCM block are realized by a network of adders (subtractors) and hardwired shifts with sharing of partial products across all the multiplications. The MCM problem is extensively studied and many different algorithms have been proposed by researchers to optimize the area consumption and computation time of the MCM block. All these approaches can be put into two broad categories: i) the common sub expression elimination (CSE) technique and ii) the graph-dependence (GD) algorithms.[2]

The CSE technique primarily searches for the most frequently occurred common sub expressions which could be maximally shared across the multipliers in the MCM block.

Potkonjak et al. and Hartley are pioneers to explore the redundancy in MCM blocks using CSE technique. Potkonjak represented coefficients in signed digit (SD) form and used a recursive bipartite matching algorithm to identify the maximally-shared common sub expressions. [2]

Hartley expressed the coefficients in canonical signed digit (CSD) form and arranged them in a two-dimensional array to search for identical bit patterns in horizontal (intra-coefficient), vertical and oblique (inter-coefficient) directions. Thereafter, more CSE algorithms with different common sub expression identification strategies are proposed to reduce the logical operators (LOs) and logic depth (LD). However, the disadvantage of CSE algorithms is that the performance of these algorithms depends on the number representation. [2]

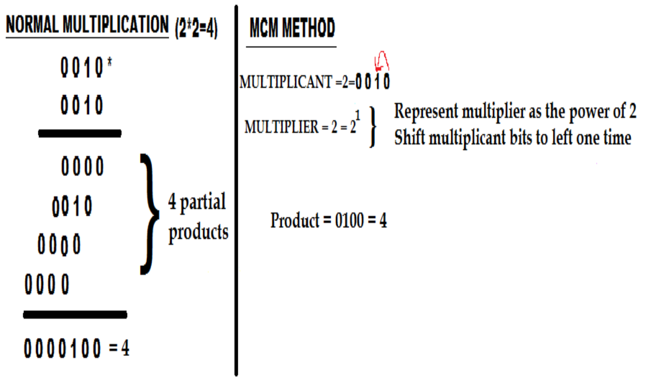


Figure 4: Normal multiplication Vs MCM method

GD algorithms, on the other hand, make no assumptions on the number representation, such that they offer more degrees of freedom to the optimization of the MCM problem. The idea of representing multiplier blocks with directed acyclic graphs (DAGs) was introduced. [2]

4. Proposed Method

The realization of novel high speed filter architecture is implemented using the minimal signed digit (MSD) representation. The MSD representation is suitable for common sub expression elimination, and it significantly reduces the number of adders required for the filter synthesis.

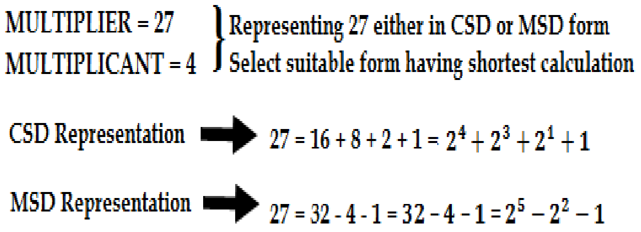


Figure 5: CSD Vs MSD representation of numbers.

The canonic signed digit (CSD) of a given number is unique, and the filters are implemented using hardware efficient CSD multipliers. The hardware complexity is further reduced by searching for a common sub expression among multiple constants in applications requiring multiple constant multiplications (MCMs).

Figure 5 shows the representation of numbers using CSD and MSD methods. After the representation in appropriate form MCM algorithm is applied. MSD representation minimizes the number of adders required.

Many of the approaches have tried to select the common sub expressions after representing the constants in the CSD representation. The minimal signed digit (MSD) representation has the same number of non-zero digits as the CSD representation but provides multiple representations for a constant. The MSD representation is suitable for common sub expression elimination, and it significantly reduces the number of adders required for the filter synthesis.

The CSD representation is a radix-2 signed digit system with the digit set {1, 0, $\underline{1}$}, where $\underline{1}$ denotes -1. The CSD representation for a given number is unique and has two properties; the first is that the number of non-zero digits is minimal and the second is that the product of adjacent two digits is zero. The CSD number system is an efficient way of representing the coefficients, as it reduces 33% of non-zero digits compared with the binary representation. The CSD representation is widely used in implementing MCMs because it guarantees the least number of additions for a given constant multiplication. However, this results in limited sub expressions for multiple constants. If the second property is relaxed in the CSD representation, it is called minimal signed digit (MSD) representation.

The MSD representation is more appropriate in finding common sub expressions for multiple constants if a proper MSD form is selected for each constant to be synthesized. The only transformations needed to convert the CSD representation to MSD representations are 10 $\underline{1}$ →011 and $\underline{1}$ 01→0 $\underline{1}$ $\underline{1}$. The CSD representation is registered as the first MSD representation. A pattern of either 10 $\underline{1}$ or $\underline{1}$ 01 is searched next, starting from the most significant digit and transformed into 011 or 0 $\underline{1}$ $\underline{1}$ respectively. A new MSD representation is generated for each transformation. The transformation is applied repeatedly to the new MSD representations found in the previous transformations until there is no such pattern. The pattern is searched in an MSD representation from the next position of the digit where a transformation is applied to generate the MSD representation.

5. Result and Comparison

In this section, results of the exact algorithm on randomly generated and real-sized FIR filter instances under CSD and MSD representations are presented and compared.

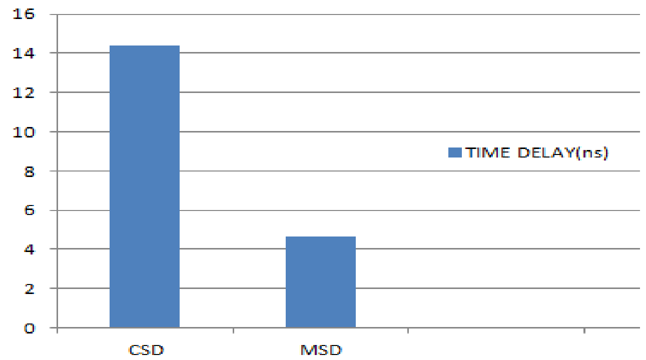


Figure 6: Graph showing time delay (ns) of two systems

The proposed FIR filter using CSD and MSD concept is simulated using model sim software and the output parameters are analysed using Xilinx software.

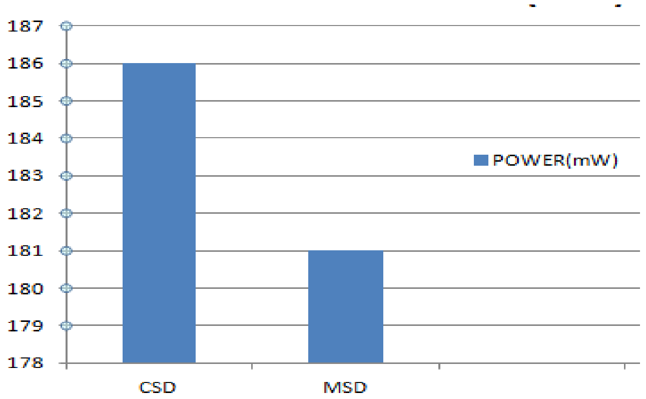


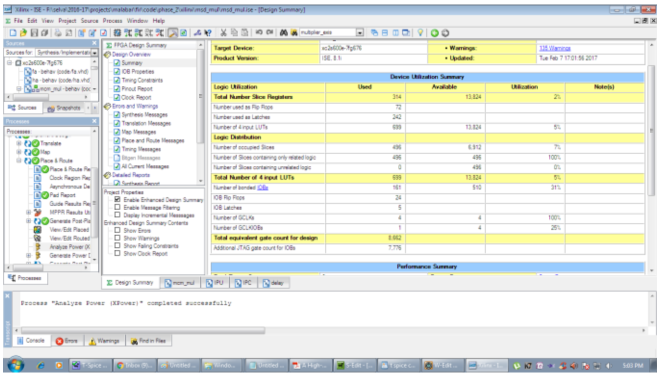
Figure 7: Graph showing power consumption of two systems

The area, power and time delay of the proposed FIR filter is analyzed using Xilinx software. Table I shows comparison between proposed method and existing method. The possibility of realization of FIR filter in transpose form configuration to achieve efficient area and delay for large order FIR filters were explored. The proposed method is also focused on the minimization of area by replacing the multiplication operations with constants by addition, subtraction, and shifting operations. Since shifts are free in terms of hardware, the MCM problem can be defined as the minimization of the number of addition/subtraction operations to implement the constant multiplications.

Table 1: Comparison of Existing and Proposed Method

Parameter	Existing method	Proposed method
	CSD Based FIR	Combined CSD and MSD Based FIR
Slices	765	496
LUTs	1066	699
Delay (ns)	14.393	4.672
Power (mW)	186	181

Application-specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involve 13% less ADP and 12.8% less EPS than that of the existing direct-from block FIR structure.



Target Device:	xc2s600e-7fg676	• Warnings:	135 Warnings
Product Version:	ISE, 8.1i	• Updated:	Tue Feb 7 17:01:56 2017

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	314	13,824	2%	
Number used as Flip Flops	72			
Number used as Latches	242			
Number of 4 input LUTs	699	13,824	5%	
Logic Distribution				
Number of occupied Slices	496	6,912	7%	
Number of Slices containing only related logic	496	496	100%	
Number of Slices containing unrelated logic	0	496	0%	
Total Number of 4 input LUTs	699	13,824	5%	
Number of bonded IOBs	161	510	31%	
IOB Flip Flops	24			
IOB Latches	5			
Number of GCLKs	4	4	100%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	8,662			
Additional JTAG gate count for IOBs	7,776			

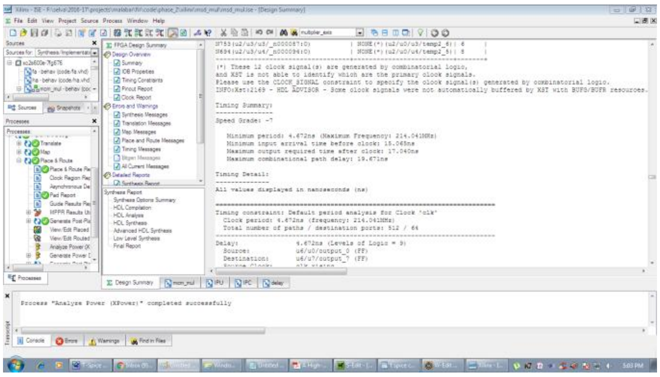
Performance Summary

```
Process "Analyze Power (XPower)" completed successfully
```

Figure 8: Synthesis report of the proposed system from Xilinx software

The synthesis report is shown in figure 7. The number of slices required for the CSD based method was 765. It is reduced to 496 when CSD based FIR filter is replaced by combined CSD and MSD based FIR filter. Correspondingly the area of the filter is also reduced. The number of look up tables required for the CSD based FIR filter was 1066 It is reduced to 699 when combined CSD and MSD based FIR

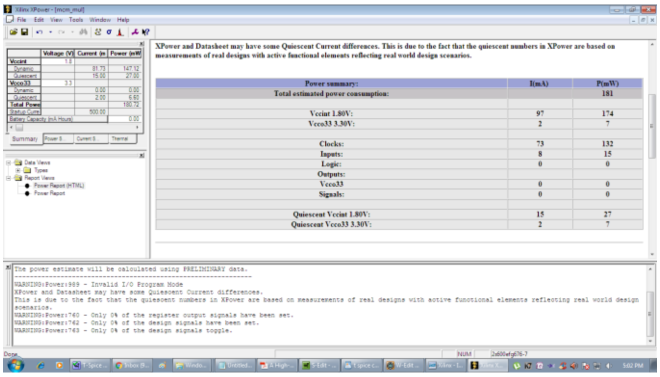
Filter is used. The time delay report is shown in figure 8. The time delay of the existing method was 14.393 ns. Using the proposed method, this time delay is reduced to 4.672. Area delay product of the proposed system is 42% less than the existing system. The energy per sample of the proposed Scheme is 40% less than the existing FIR filter structures.



```
N753(u2/u3/u3/_n000087:O)          | NONE(*)(u2/u0/u3/temp2_6)| 6      |
N694(u2/u3/u4/_n000094:O)          | NONE(*)(u2/u0/u4/temp2_5)| 8      |
-----------------------------------+------------------------------------+
(*) These 12 clock signal(s) are generated by combinatorial logic,
and XST is not able to identify which are the primary clock signals.
Please use the CLOCK_SIGNAL constraint to specify the clock signal(s) generated by combinatorial logic.
INFO:Xst:2169 - HDL ADVISOR - Some clock signals were not automatically buffered by XST with BUFG/BUFR resources.

Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 4.672ns (Maximum Frequency: 214.041MHz)
   Minimum input arrival time before clock: 15.068ns
   Maximum output required time after clock: 17.040ns
   Maximum combinational path delay: 19.671ns

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 4.672ns (frequency: 214.041MHz)
  Total number of paths / destination ports: 512 / 64
-------------------------------------------------------------------------
Delay:               4.672ns (Levels of Logic = 9)
  Source:            u6/u0/output_0 (FF)
  Destination:       u6/u7/output_7 (FF)
```

```
Process "Analyze Power (XPower)" completed successfully
```

Figure 9: Time delay report of the proposed system from Xilinx software



XPower and Datasheet may have some Quiescent Current differences. This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design scenarios.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		181
Vccint 1.80V:	97	174
Vcco33 3.30V:	2	7
Clocks:	73	132
Inputs:	8	15
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

```
The power estimate will be calculated using PRELIMINARY data.
--------------------------------------------------------------
WARNING:Power:989 - Invalid I/O Program Mode
XPower and Datasheet may have some Quiescent Current differences.
This is due to the fact that the quiescent numbers in XPower are based on measurements of real designs with active functional elements reflecting real world design
scenarios.
WARNING:Power:760 - Only 0% of the register output signals have been set.
WARNING:Power:762 - Only 0% of the design signals have been set.
WARNING:Power:763 - Only 0% of the design signals toggle.
```

Figure 10: Power report of the proposed system from Xilinx software

The power report is shown in figure 9. The total power required for the proposed system is 181mW. For the existing system using CSD based FIR filter, required power was 186mW. So the power is reduced by 5mW.

6. Conclusion

The impact of power consumption, delay, area has been successfully done. Smaller truncation error and error compensation (MCM used done). It yields considerable time saved for the compensated circuit. A high-accuracy, low-cost, and flexible fixed-width got by using this multiplier based on MCM. The possibility of realization of FIR filter in transpose form configuration to achieve efficient area and delay for large order FIR filters were explored. Application-specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involve 13% less ADP and 12.8% less EPS than that of the existing direct-from block FIR structure.

References

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[2] T. Hentschel and G. Fettweis, “Software radio receivers,” in CDMA Techniques for Third Generation Mobile Systems. Dordrecht, The Netherlands: Kluwer, 1999, pp. 257–283.

[3] E. Mirchandani, R. L. Zinser, Jr., and J. B. Evans, “A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals],” IEEE Trans. Circuits Syst. II, Analog Digital. Signal Process. vol. 39, no. 10, pp. 681–694, Oct. 1995.

[4] S. Nerurkar, K. H. Abed, R. E. Siferd and V. Venugopal, “Low power Sigma-delta Decimation Filter,” in Proceedings of 45th IEEE International Midwest Symposium on Circuits and Systems, 2002.

[5] Y. C. Lim, R. Yang, D. Li, and J. Song, “Signed-powers-of-two term allocation scheme for the design of digital filter”, IEEE Trans. Circuits Syst. II, vol. 46, pp. 577-584, May 1999.

[6] L.Naviner, J.F.Naviner, “Design and Prototyping of a □ Δ Decimator Filter for DECT Standard”, Proceedings of 43rd IEEE Midwest Symposium on Circuits and Systems, 2000

[7] Sanjit K. Mitra and James .F. Kaiser, Handbook for Digital Signal Processing, John Wiley and Sons 1993.

[8] Vinay K. Ingle and John G. Proakis, Digital Signal Processing using Matlab, Brooks/Cole 2000.

[9] O. Gustafsson and L. Wanhammar, “ILP modeling of the common sub expression sharing problem,” in Proc. 9th IEEE ICECS, Dec. 2002, vol. 3, pp. 1171–1174.

[10] S. Vijay et al., “A greedy common sub expression elimination algorithm for implementing FIR filters,” in Proc. ISCAS, May 2007, pp. 3451–3454.

[11] R. M. Hewlitt and E. S. Swartzlander, “Canonical signed digit representation for FIR digital filters,” in Proc. IEEE WorkShop SiPS, 2000, pp. 416–426.

[12] N. Sankarayya, K. Roy, and D. Bhattacharya, “Algorithms for low power high speed FIR filter realization using differential coefficients,” IEEE Trans. Circuits Syst. II, vol. 44, pp. 488–497, June 1997.

[13] F. Klass, “Semi-dynamic and dynamic flip-flops with embedded logic,” in Symp. VLSI Circuits Dig. Tech. Papers, June 1998, pp. 108–109.

[14] B. Nikolic et al., “Sense amplifier-based flip-flop,” in IEEE Int. Solid- State Circuits Conf. Dig. Tech. Papers, Feb. 1999, pp. 282–283.

[15] B. S. Kong et al., “Conditional capture flip-flop for statistical power reduction,” in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, Feb. 2000, pp. 290–291

Author Profile

Raseena .K .A has received B.Tech degree in Electronics and Communication Engineering from Calicut University, currently pursuing M.Tech degree in Applied Electronics and Communication under Kerala Technological University. Research interests include VLSI and embedded systems.

Vincy Mathew completed M.Tech and B.Tech from Karunya University. She is currently working as assistant professor in Malabar College of engineering and technology, Thrissur. Her research interests include VLSI.