DESIGN OF TIME AND AREA EFFICIENT VARIABLE SIZE PARTITION AND SEGMENTED AMALGAM FIR FILTER

DR. K.LAKSHMINARAYANAN[1], S. A. SIVASANKARI[2], DR.M.KARTHIKEYAN[3], DR. V. SARAVANAN[4], DR. T. SAMRAJ LAWRENCE[5]

[1] Department of Electronics and Communication Engineering, Francis Xavier Engineering, College, Tirunelveli, Tamilnadu India
[2] Associate professor, ECE Department, Koneru Lakshmaiah Education Foundation, Greenfields, Vaddeswaram
[3] Professor and Head, Civil Engineering, Dhanalakshmi Srinivasan College of Engineering, Coimbatore, India
[4,5] Assistant Professor, Department of Computer Science, College of Engineering and Technology, Dambi Dollo University, Ethiopia
[1]kyelyen@gmail.com, [2]s.a.sivasankari@gmail.com, [3]mkartik2009@gmail.com, [4]reachvsaravanan@gmail.com, [5]samrajinethiopia@gmail.com

ABSTRACT:

FIR filters are mostly exploited due to its phase linearity, less sensitivity and bounded input bounded output (BIBO) stability which makes FIR filter is suitable for Digital Signal Processing (DSP) applications. The major components in FIR filters are adder, multiplier and registers.FIR filter is categorized based on their coefficients and it is multiplied with the input signal Multiple Constant Multiplication (MCM) is preferred where the coefficients are considered as constant and it avoids the multipliers by performing the shift, add and subtract operations.MCM is used for transpose form realization whereas Single Constant Multiplication (SCM) is used for direct form realization. The hybrid FIR filter is designed using fixed size partitioning that means the value of partitioning parameter is fixed. Fixed size partitioning leads area and adder complexity. The variable size partitioning segmented hybrid FIR filter is designed by segmenting the input into two part then it performs radix2^r MCM, partial product addition and zero appending then both the inputs are added to produce output. This work is synthesized and simulated using Xilinx ISE 14.2 on Spartan3E.

KEYWORDS: Multiple Constant Multiplication, RADIX-2r, low power, high speed FIR filter

I. INTRODUCTION

Realization of digital filters can be done by two ways they are one is by convolutional (FIR) Finite impulse response filter and other is by recursion type (IIR) Infinite Impulse Response filter. Finite impulse response (FIR) filter design is an important part in digital signal processing (DSP). In many digital communication applications, speech signal processing, seismic processing and other areas of signal processing requires higher order FIR filter. Traditional FIR filters are implemented using adders but that in turn consumes more memory. So it is very important to design an FIR filter with minimized hardware utilization and delay.

1.1 Multiple Constant Multiplications

The hardware difficulty of FIR filters is solved by multiple constant multiplication (MCM). The latter is an arithmetic operation that multiplies a set of fixed-point constants C0, C1, C2, …, CM − 1 with the same fixed-point variable X. To be efficiently implemented, i.e. rapid, compact, and low power, MCM must avoid costly multipliers. The hardware alternative will be multiplier less, using only additions, subtractions, and left shifts. We assume that addition and subtraction have the same area/speed cost, and that the shift is costless since it can be realised without any gates, i.e. just by using hardwiring. Therefore, the MCM trouble is defined as the course of discovering the least number of additions, and/or the least number of cascade additions shaping the critical path.

1.2 Finite Impulse Response Filter

In signal processing a Finite Impulse Response (FIR) filter has impulse response of finite duration, because it settles to zero in finite time. The impulse response of an Nth order discrete time FIR filter lasts exactly N+ 1 sample before it settles to zero.

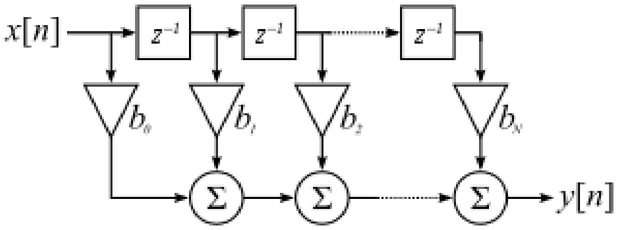


Fig.1. General Block diagram of FIR filter

$$y(n) = \sum_{i=0}^{N-1} h(i).x(n-i) \qquad (1)$$

here y(n) represents the filter output, h(i) represents filter coefficient, x(n) represents input.

Computation of (1) can be characterized by the recurrence relation.

$$Y(z) = z^{-1}(\cdots(z^{-1}(z^{-1}h(N-1)+h(N-2))+h(N-3))\cdots+h(1))+h(0)]\,X(z) \qquad (2)$$

An FIR filter features a number of useful properties. They are, It do not need feedback, i.e., that any rounding errors are not compounded by summing iterations. The same relative error occurs in each calculation. This makes an implementation simpler. It can be easily designed as linear phase by making the coefficients sequence symmetric. This is desired for phase sensitive applications. For example data communications, seismology, crossover filters and mastering. Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler. Are inherently stable, since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than times the largest value appearing in the input. Designing for linear phase can be performed easily by making the coefficient sequence symmetric. This property is usually desired for phase-sensitive applications, for instance data communications, seismology, crossover filters, and mastering. The drawback in FIR filter is it requires more computational power in a general purpose processor when compared with the IIR filter with the same selectivity and sharpness

1.3 Distributed Arithmetic

Distributed arithmetic (DA) radix is defined as the number of simultaneously computed bit sums and it is expressed as a power of two. The bit serial nature of distributed arithmetic can limit throughput. To improve throughput the basic DA algorithm can be modified to compute more than one bit –sum at a time. For example the DA radix of 2(2^1) indicates that a one bit sum is computed at a time. The DA radix of 4(2^2) indicates that 2 bit sum is computed at a time and so on. To process a more than one bit at a time, the parallelism technique would be introduced into the operation, which can improve performance at the expense of area.

II. VARIABLE SIZE PARTITIONING AND RADIX-2r BASED MCM (VPRM)

The architecture of Variable Size Partitioning and Radix-2r based MCM (VPRM) is depicted in figure 2. Variable Size Partitioning (VP) is preferred to overcome the limitations in Fixed Size Partitioning (FP) such as length of the partitioning parameter increased with the order of the filter, complexity of register and large area occupation of adders. In VP approach an efficient partitioning approach is chosen as minimum size, optimal size and maximum size. If the bit size is less than 8-bit it is termed as minimum size partitioning, if the bit size is varied between 9 to 12-bit termed as optimal size portioning otherwise called as maximum size partitioning. This partitioning approach reduces the complexity of adders and registers. It is mainly based on the sharing of common structures between the coefficients of filter.

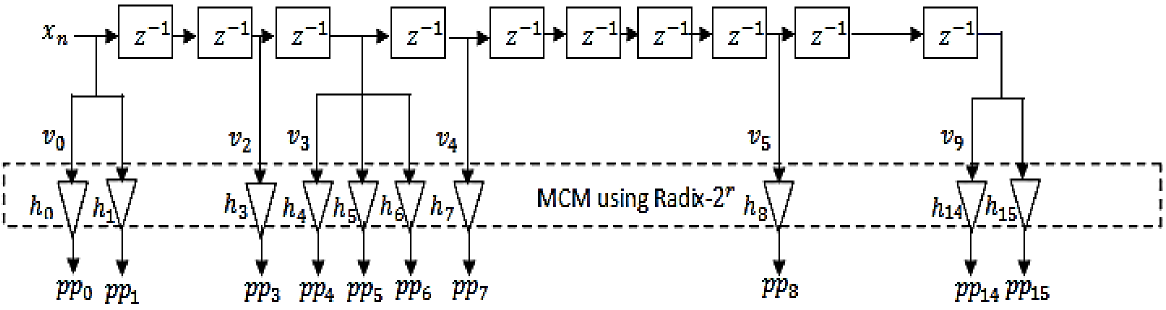

x_n
z^{-1}
v_0
v_2
v_3
v_4
v_5
v_9
MCM using Radix-2^r
h_0 h_1 h_3 h_4 h_5 h_6 h_7 h_8 h_{14} h_{15}
pp_0 pp_1 pp_3 pp_4 pp_5 pp_6 pp_7 pp_8 pp_{14} pp_{15}

Fig.2. Architecture of VPRM

Segmented FIR filter is proposed with Multiple Constant Multiplication (MCM) based on RADIX-2r and its block diagram representation is depicted in Figure2. N bit input is applied for segmentation where the inputs are segmented as two N/2 bits to reduce the complexity of FIR filter. The segmented inputs are given to the Radix-2r based MCM. In this MCM block, the partial products are obtained by performing the multiplication of input signal with multiple coefficients and these coefficients are reused by sharing technique which reduces the utilization of resources. Then it performs the addition of partial products. Appending of zeros and shifting are done in the result from the partial product addition block. Zero appended result is added with the shifted result to obtain the FIR filtered output.

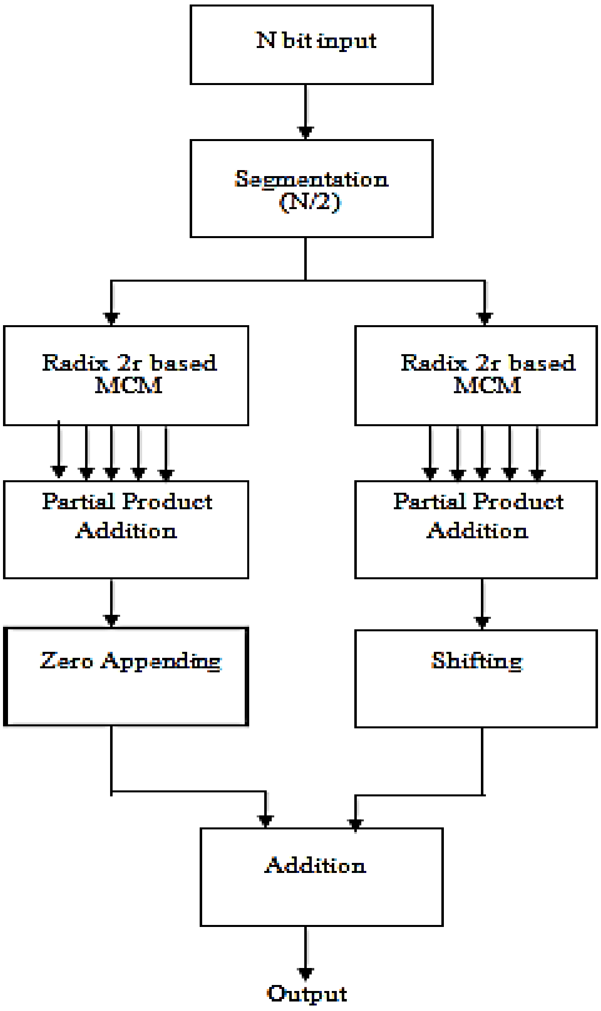


Fig.3. Block Diagram of Segmented FIR Filter

2.1 Radix-2^r Based MCM

In this proposed technique, MCM is preferred over than Single Constant Multiplication (SCM), because it performs small additions which are separated by the delay elements leads to high speed architecture and it also shared the intermediate terms between the constants. The general structure of MCM is shown in Figure 4.

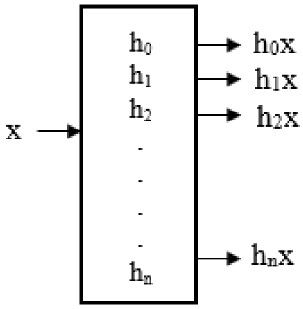


Fig.4. General Structure of MCM

The sharing of the common structure among the neighboring filter coefficients are done with the help of Radix-2r and it is shown in Figure 5. For example, if the coefficients are considered as odd coefficient set {1,3,5,7……27,29,31} the sharing is as follows.

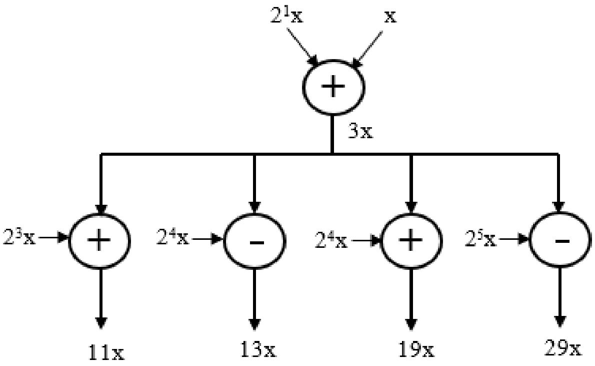


Fig.5. Common sub sharing with 3x

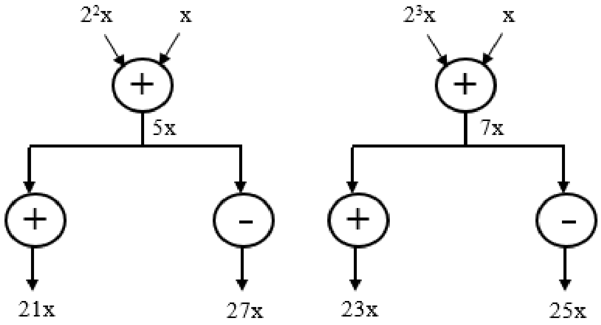


Fig.6. Common sub sharing with 5x and 7x

The products 11x, 13x, 19x and 29x are obtained by common sub sharing approach with 3x and it is shown in Figure 5. Similarly, Figure 6 shows the products 21x, 27x and 23x, 25x by using common sub sharing approach with 5x and 7x.

Example for 16-bit input

Set input value in 16 bit(ex.0000100100000101). The output Y will be obtained by the operation of,

1) The given input is segmented into two parts.

    Ex:0000100100000101

    x1=00001001; x2=00000101

2) Both parts performs radix 2^r multiplication and partial product addition.

    x1*1=pp0 x2*1=p0

    x1*3=pp1 x2*3=p1

    ….. …..

    x1*31=pp1 x2*31=p15

3) Then appending zero to the first part input.

    x3=2304

    x5=(x3,8’b0)

    =(000010010000000000000000)

4) Shifting the second part input.

    x4=1280

5) Then add the zero appending input to the shifting input by adder.

    y=x5+x4=589824+1280

    =591104

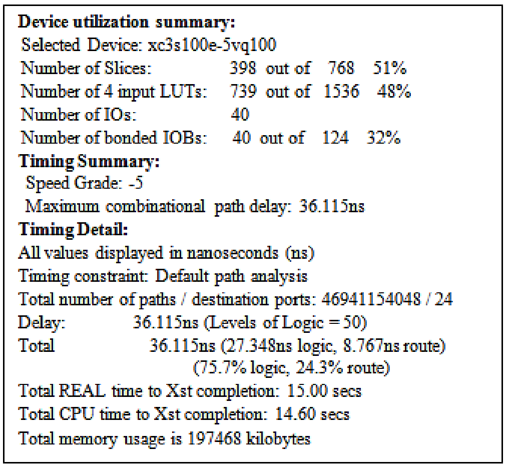
6) Take the output from adder.

Table – I: Performance Comparison Table

PARAMETER	EXISTING METHOD	PROPOSED METHOD
No of Slice Registers:	490 out of 960 51%	398 out of 960 41%
No of 4 input LUTs:	728 out of 1920 37%	739 out of 1920 38%
No of Slice Flip Flop pairs used:	240 out of 1920 12%	0
Total Memory Usage	200540KB	197468KB
Maximum combinational path delay:	36.152ns	36.115ns

III. RESULTS AND DISCUSSIONS

The Segmented FIR filter based on VPRM is synthesized and simulated using Xilinx ISE 14.2. It is synthesized in Spartan3E device with package xc3s100e-5vq100. This architecture is coded in Verilog HDL and the proposed Segmented FIR filter’s performances are evaluated in terms of area, power consumption and delay. The results showed that an FIR filter with segmentation is better than the FIR filter without segmentation.



```
Device utilization summary:
 Selected Device: xc3s100e-5vq100
 Number of Slices:           398  out of   768   51%
 Number of 4 input LUTs:     739  out of  1536   48%
 Number of IOs:              40
 Number of bonded IOBs:      40  out of   124   32%
Timing Summary:
  Speed Grade: -5
  Maximum combinational  path delay: 36.115ns
Timing Detail:
All values displayed in nanoseconds (ns)
Timing constraint: Default path analysis
Total number of paths / destination ports: 46941154048 / 24
Delay:            36.115ns (Levels of Logic = 50)
Total               36.115ns (27.348ns logic, 8.767ns route)
                         (75.7% logic, 24.3% route)
Total REAL time to Xst completion: 15.00 secs
Total CPU time to Xst completion: 14.60 secs
Total memory usage is 197468 kilobytes
```

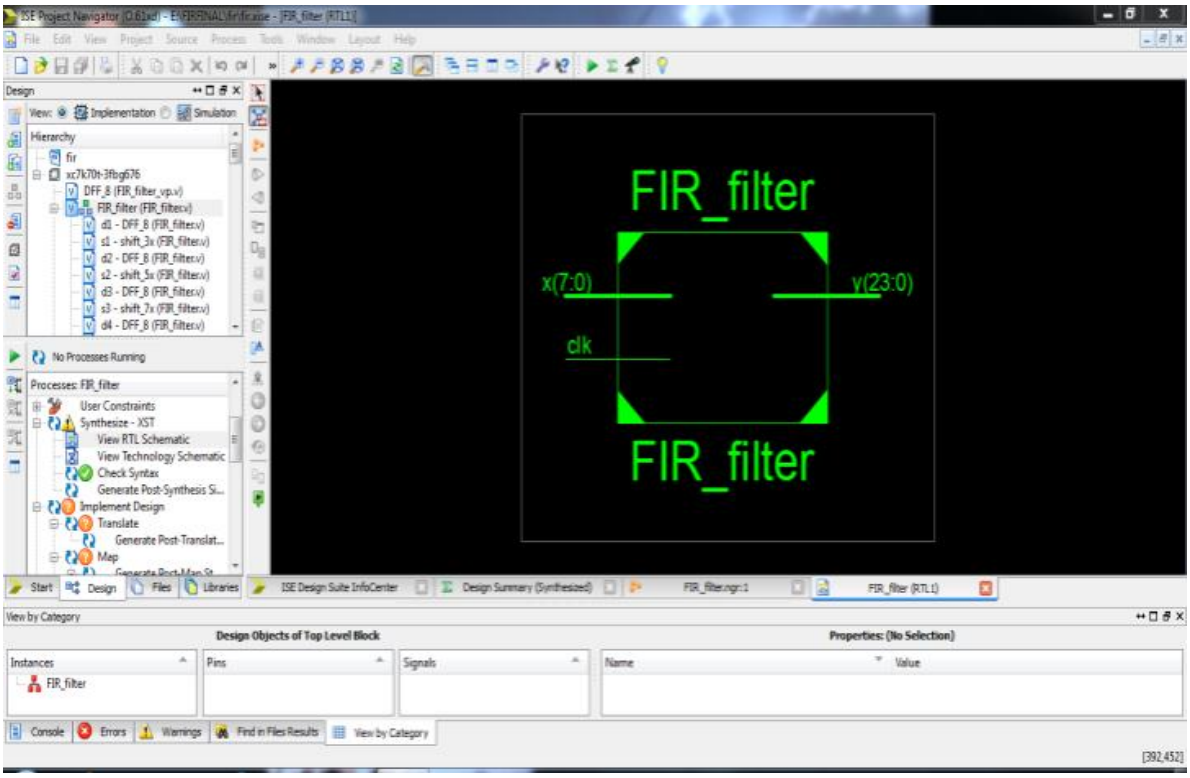
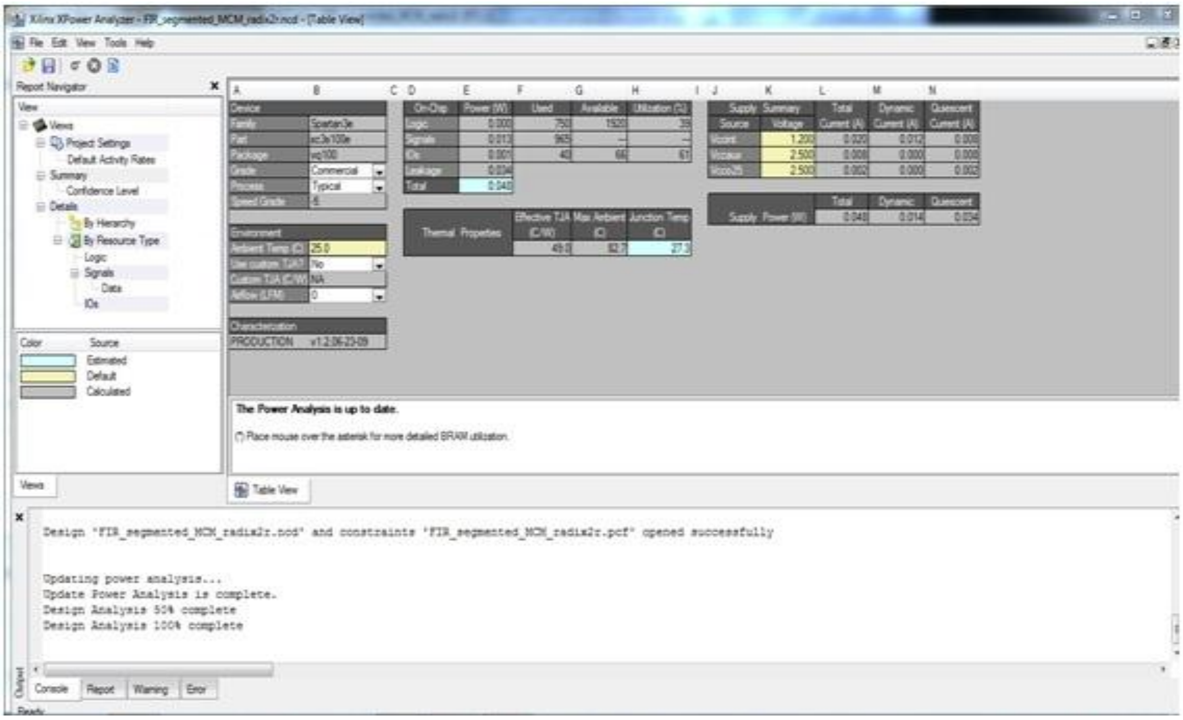

ISE Project Navigator (O.61xd) - E:\FIRFINAL\fir\firfinal.xise - [FIR_filter (RTL1)]
FIR_filter
x(7:0)
y(23:0)
clk
FIR_filter
Design Objects of Top Level Block
Properties: (No Selection)
Instances
FIR_filter
Pins
Signals
Name
Value

Fig.7. RTL View

Figure 7 shows the Register Transfer Logic (RTL) view of the finite impulse response (FIR) filter.



Device	
Family	Spartan3e
Part	xc3e100e
Package	vq100
Grade	Commercial
Process	Typical
Speed Grade	-5

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	750	1920	39
Signals	0.013	965	---	---
IOs	0.001	40	66	61
Leakage	0.034			
Total	0.048			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	49.0	82.7	27.3

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.020	0.012	0.008
Vccaux	2.500	0.008	0.000	0.008
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.048	0.014	0.034

Color	Source
▭	Estimated
▭	Default
▨	Calculated

The Power Analysis is up to date.

(*) Place mouse over the asterisk for more detailed BRAM utilization.

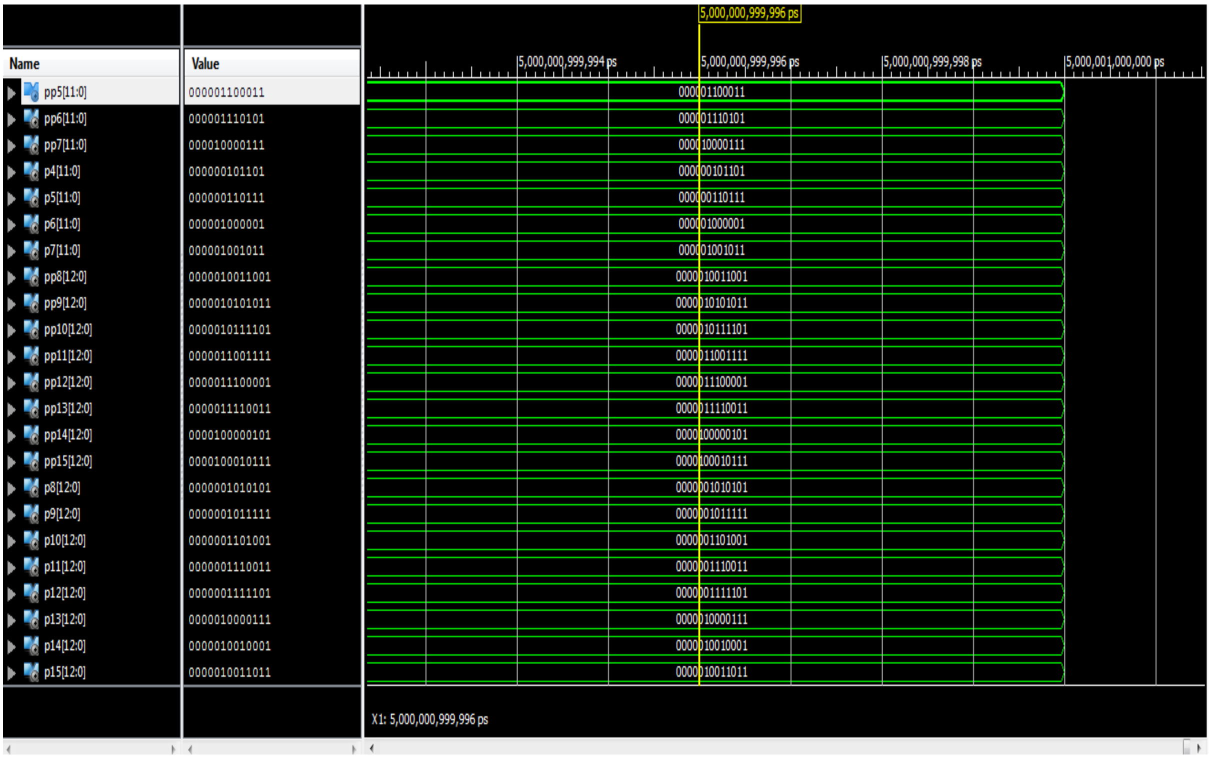
```
Design 'FIR_segmented_MCM_radix2r.ncd' and constraints 'FIR_segmented_MCM_radix2r.pcf' opened successfully


Updating power analysis...
Update Power Analysis is complete.
Design Analysis 50% complete
Design Analysis 100% complete
```

Ready

Fig.8. Dynamic Power Calculation

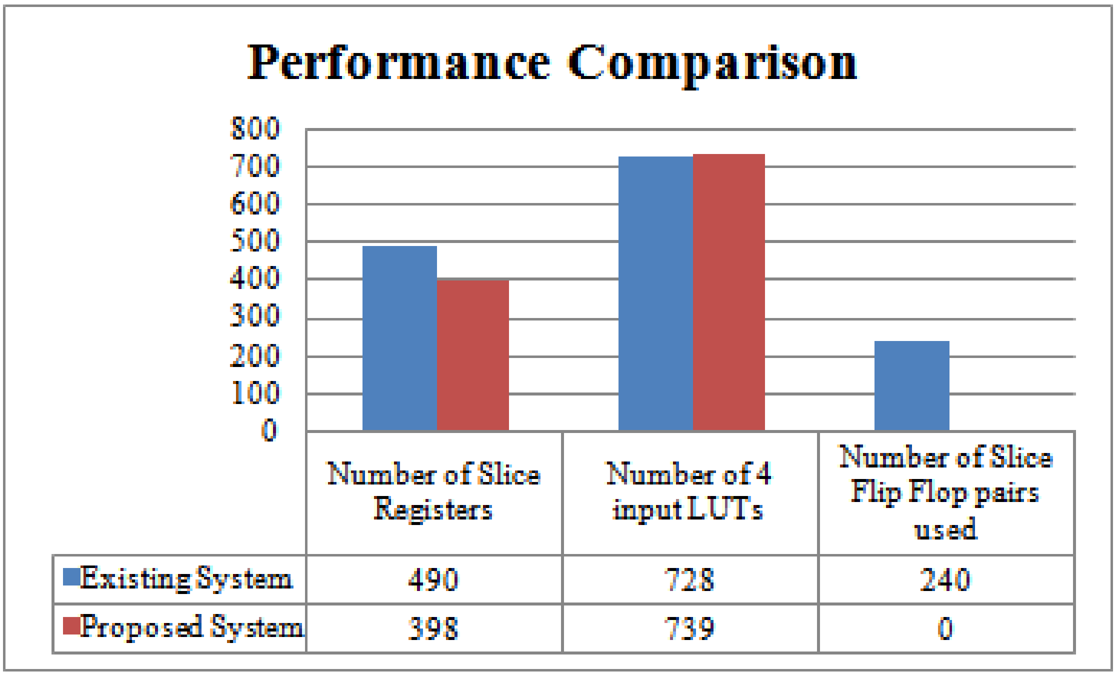
Figure 8 shows the dynamic power calculation of the designed FIR filter



Name	Value
pp5[11:0]	000001100011
pp6[11:0]	000001110101
pp7[11:0]	000010000111
p4[11:0]	000000101101
p5[11:0]	000000110111
p6[11:0]	000001000001
p7[11:0]	000001001011
pp8[12:0]	0000010011001
pp9[12:0]	0000010101011
pp10[12:0]	0000010111101
pp11[12:0]	0000011001111
pp12[12:0]	0000011100001
pp13[12:0]	0000011110011
pp14[12:0]	0000100000101
pp15[12:0]	0000100010111
p8[12:0]	0000001010101
p9[12:0]	0000001011111
p10[12:0]	0000001101001
p11[12:0]	0000001110011
p12[12:0]	0000001111101
p13[12:0]	0000010000111
p14[12:0]	0000010010001
p15[12:0]	0000010011011

Fig.9. Output Waveform

Figure 9 shows the resulting waveform of the finite impulse response (FIR) filter. Here x is the 16-bit inputsignal, x1, x2 is the 8-bit segmented inputs, pp0 to pp15 and p0 to p15 are partial products obtained from the MCM, y is FIR filter output.



	Number of Slice Registers	Number of 4 input LUTs	Number of Slice Flip Flop pairs used
🟦 Existing System	490	728	240
🟥 Proposed System	398	739	0

Fig.10. Performance Comparison

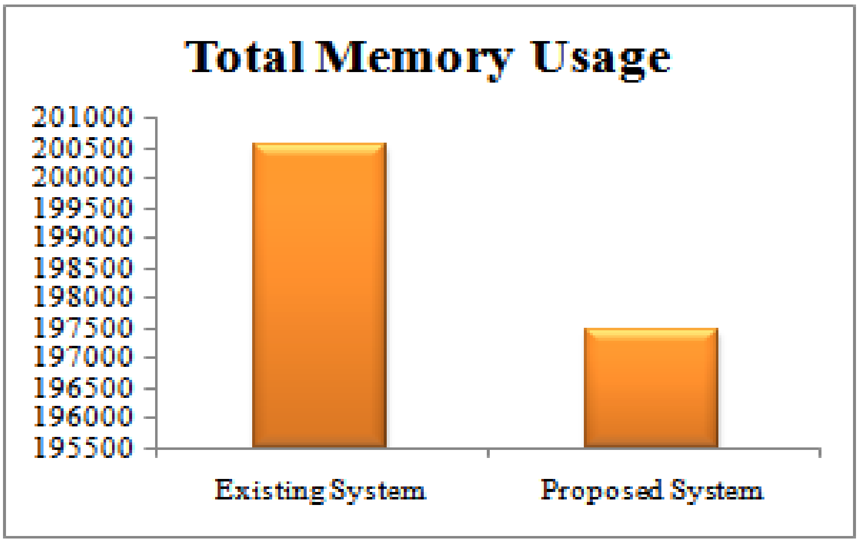


Fig.11. Comparison on Memory Utilization

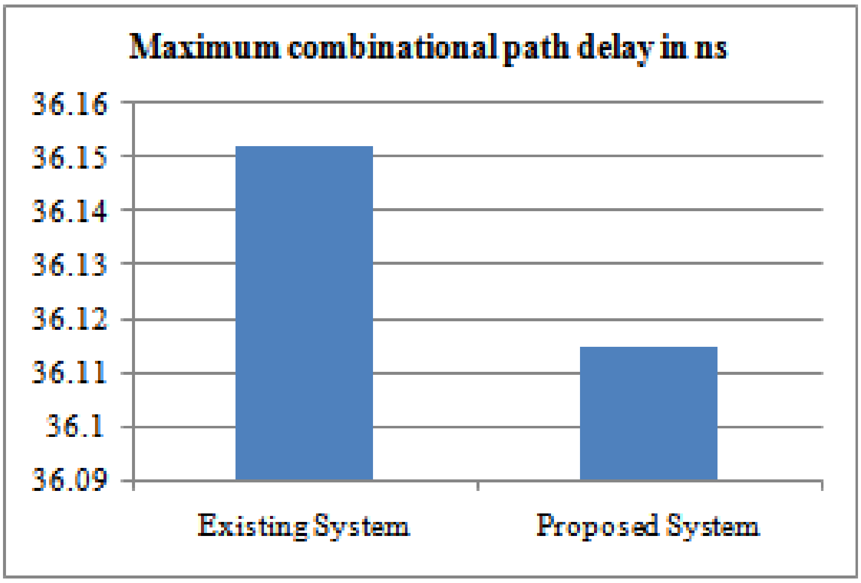


Fig.12.Comparison on Maximum Combinational path delay

IV. CONCLUSION

This project proved that Radix-2^r is one of the leading MCM heuristics, with a simple segmentation of input achieves the best results in speed and adder complexity. This has been clearly demonstrated by the MCM through segmentation of input. The FIR filter achieves the combinational path delay of 36.115; this is less compared to an existing system. This required less memory space to store the values compared to an existing system. Thus this is an area superiority and faster than the existing system.

REFERENCES

[1] A. K. Oudjida, A. Liacha, M. Bakiri and N. Chaillet, "Multiple Constant Multiplication Algorithm for High-Speed and Low-Power Design," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 2, pp. 176-180, Feb. 2016.

[2] Aggarwal, Anurag &Satija, Astha&Nagpal, Tushar. (2013). FIR Filter Designing using Xilinx System Generator. International Journal of Computer Applications, pp 37-41.

[3] Aksoy, L., Günes, E.O., Flores, P.: ‘Search algorithms for the multiple constant multiplications problem: exact and approximate’, Microprocess. Microsyst., 2010, 34, (5), pp. 151–162

[4] ApoorvaAggarwal.F., Tarun Kumar Rawat.S.: “The design of digital FIR filter using evolutionary and swarm optimization techniques,” IEEE Trans.Comput.,vol.64,no.5,pp.1502-1505, May 2017

[5] Basant Kumar MohantyPramod Kumar Meher,” A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications” IEEE transactions on VLSI, volume. 24, issue. 2, February 2016.

[6] Durga Prasad, Abhilash, Subrahmaneswara Rao, “Design of Efficient FIR filter with EDBNS multiplier using Transpose method for various Applications” Volume 5, Issue 4, International Journal of Engineering Development and Research, pp 1182 -1188, 2017.

[7] Feng, F., Chen, J., Chang, C.H.: ‘Hypergraph based minimum arborescence algorithm for the optimization and reoptimization of multiple constant multiplications’, IEEE Trans. Circuits Syst. I, 2016,doi: 10.1109/TCSI.2015.2512742

[8] Liacha, A., Oudjida, A.K., Ferguene, F., et al.: ‘A variable Radix-2r algorithm for single constant multiplication’. Proc. Int. IEEE (NEWCAS) Conf. New Circuits and Systems, Strasbourg, France, June 2017, pp. 1–4 DOI: 10.1109/ NEWCAS.2017.8010156

[9] Liu, L. S. DeBrunner, V. DeBrunner and K. Johansson, "Internal quantization in FIR filters implemented using Multiple Constant Multiplications," 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2010, pp. 1284-1287.

[10] Lou, X., Yu, Y.J., Meher, P.K.: ‘Fine-grained critical path analysis and optimization for area-time efficient realization of multiple constant multiplications’, IEEE Trans. Circuits Syst. I Reg. Papers, 2014, 62, (3), pp. 863–872

[11] M. N. Ahuja and P. Palsodkar, "Design of Efficient Add/Shift Algorithm for Multiple Constant Multiplication," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, 2015, pp. 801-805.

[12] Mahesh, R., Vinod, A.P.: ‘A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters’, IEEE Trans. Comput.-Aided Des. Integer. Circuits Syst., 2008, 27, (2), pp. 217–229

[13] Nagaria, Shuchi& Singh, Anushka&Niranjan, Dr. Vandana. “Efficient FIR filter design using Booth Multiplier for VLSI applications”. International Conference on Computing, Power and Communication Technologies, September 2018.

[14] O. Gustafsson, J. O. Coleman, A. G. Dempster and M. D. Macleod, "Low-complexity hybrid form FIR filters using matrix multiple constant multiplication," Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004., Pacific Grove, CA, USA, 2004, pp. 77-80 Vol.1.

[15] Oudjida, A.K., Chaillet, N.: ‘Radix-2r arithmetic for multiplication by a constant’, IEEE Trans. Circuits Syst. II Exp. Brief, 2014, 61, (5), pp. 349–353

[16] Thong, J., Nicolici, N.: ‘An optimal and practical approach to single constant multiplication’, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 2011, 30, (9), pp. 1373–1386