# Area and Power Efficient Weighted Partitioning For Fast MCM Calculations

Deepak Tiwari[1], Prof. Prashant Purohit[2]

[1]*Mtech Scholar,* [2]*Research Guide*

*Department of Electronics and Communication Engineering, LNCT, Bhopal*

***Abstract - ASICs stand for application specific integrated circuit are the conventional solution for superior applications, but the high development costs and time-to- market factors prohibit the deployment of such solutions for certain cases. DSP processors offer high programmability, but the successive execution nature of their architecture can unfavorably influence their throughput execution. FPGAs are somewhere in between ASICs and DSPs, provides programmability and improve performance through parallelization. At present all proposed plans for FFT utilize ROMs or memory for complex twiddle multiplications. Proper strategies must be followed to eliminate the need of multipliers in FFT design. One of the most frequently used and significant method to eliminate the multipliers used in FFT design is using New Distributed Arithmetic. In this work improved partitioning method for natural numbers by weighted theory is improved with efficient adder using Koge Stone Adder, which significantly reduces the power consumption and area utilization of the design of the application.***

***Keywords - Koge Stone Adder, LUTs, Power, MCM, Weighted Partitioning.***

## I. INTRODUCTION

There is market demand to manufacture efficient digital signal processing (DSP) and digital image processing (DIP) based devices. An efficient DSP/DIP system should be of minimum cost, minimum power consumption, and maximum processing speed. Examples of DSP/DIP systems are wireless communication, machine vision, and telescopes. The reasons for seeking efficient DSP/DIP implementation are shown by considering three different applications.

Two common themes can be extracted in the DSP/DIP systems mentioned above; first they require an efficient DSP/DIP implementation, and second the core computational operation is the multiple constant multiplications (MCM) such as the FIR and IIR filters. The MCM implements the dot-product operation of a signal vector with a vector of fixed coefficients. Since the MCM is a core operation of most DSP/DIP systems, implementing an efficient MCM operation is critical to building efficient DSP/DIP systems. Three objectives for the design engineer when looking for efficient MCM operation are:

1. Minimize the use of resources.
2. Minimize the critical path length.
3. Minimize power consumption.

On the other hand, the existence of a long critical path in the MCM circuit increases the chance of glitch occurring which increases the dynamic power consumption. Dynamic power consumption of the CMOS device is due to the switching of transistors from one logic state to another and to the charging of external load capacitance. Therefore, minimizing critical path(s) length minimizes dynamic power consumption and maximizes the processing speed simultaneously. Where, the critical path in the MCM is the longest path that the signal passes through from the input to the output.

Using a digital multiplier to implement the MCM operation contradicts with achieving these objectives. This is because the digital multiplier consumes a large area, high power, and has a long delay (critical path). In other words, the digital multiplier is a bottleneck element in DSP/DIP systems. Therefore, it is important to remove the multiplication operation from the MCM operation to increase its efficiency when implemented on ASICs and FPGAs. Removing the explicit multiplication operation from the implementation of MCM makes the latter a multiplierless. In this case, the multiplication operation is substituted with simpler ones of add/subtract and shift. It is required to develop an optimization procedure that works on the multiplierless MCM to maximize the achieving of the above objectives. There are several optimization procedures in the literature that tackle the MCM problem.

## II. SYSTEM MODEL

### *a. Distributed Arithmetic*

Multiply and Accumulate (MAC) is one of the basic blocks used in many digital signal processing systems. The general structure of a MAC unit consists of a multiplier, an adder and a shifter. Elimination of multiplier in MAC unit can be made possible by using algorithms such as Distributed Arithmetic (DA).

The multiplication operation is replaced by a mechanism that generates partial products and then sums the products

together. The key difference between distributed arithmetic and standard multiplication is in the way the partial products are generated and added together. Since its introduction, distributed arithmetic has been widely adopted in many digital signal processing applications, including but not limited to digital filtering discrete cosine transform discrete Fourier transform .

A block diagram of the full table implementation is shown in Fig. 2.1.

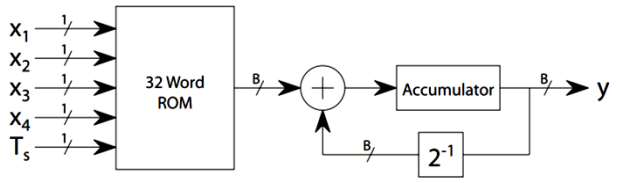


Figure 2.1 Block diagram of a 4tap DA with full table.

The ability of distributed arithmetic to reduce a multiply operation into a series of shifts and additions yields great potential for implementing various DSP systems at a significantly reduced area. However, this reduction in area comes at the cost of increased power and decreased throughput. This trade-off among area, throughput, and power has generated substantial research into making DA-based designs a more viable alternative to the standard multiply-accumulate designs for certain applications.

*b. Multiplier Less Multiplication*

It also illustrates the advantage of removing the explicit multiplication operation even from a single multiplier. Assume an MCM with N coefficients, $w_0; \cdot \cdot ; w_{N-1}$, and a signal variable $x_n$ multiplies these coefficients, where the subscript n is the discrete time variable. Consider the multiplication of the coe-fficient $w_0$ by $x_n$ given by $w_0x_n$. If the signal wordlength equals L = 4 bit (assume binary representation) and the coefficient wordlength equals 3 bit, their binary multiplication is shown in Figure 2.2. The symbol x<n,j> represents the jth bit of the binary representation of xn, similarly for w<0,j>. Each bit multiplication of the form w<0,i>x<n,j> implemented using logical AND operation.

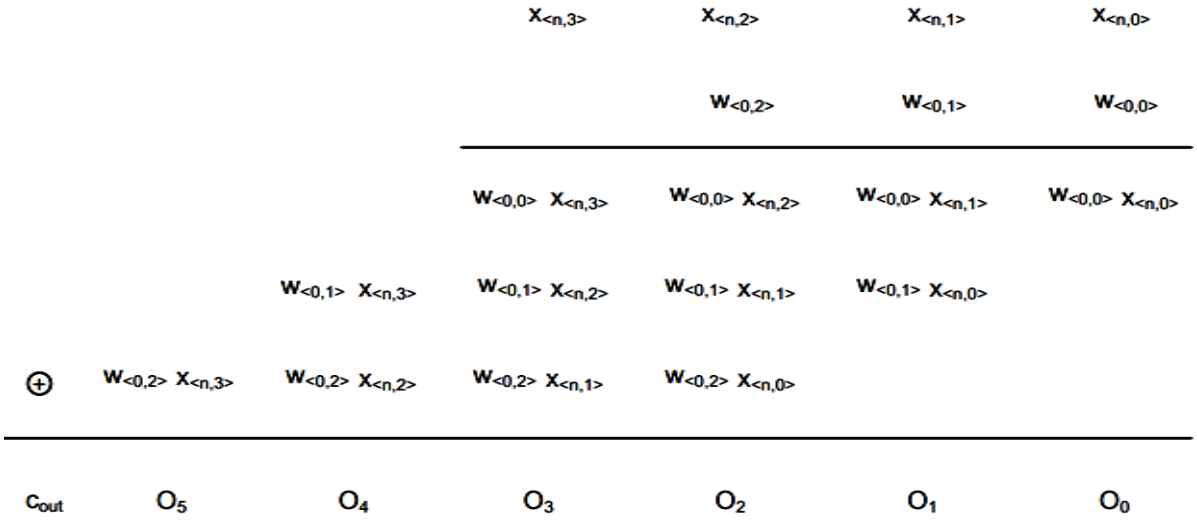


Figure 2.2 Illustration of binary multiplication of $w_0$ with $x_n$.

In figure 2.2 $w_0$ wordlength equals L = 3 bits and $x_n$ wordlength equals L = 4 bits. The symbol ⊕ is the logical exclusive OR operation.

The MCM is found in most computationally intensive DSP/DIP such as FIR lters, IIR filters, correlators, DSP transforms, edge detection, etc. Therefore implementing an efficient MCM operation is a major concern in the design of low cost, high-speed, and low-power DSP/DIP systems. To solve the multiplier problem, the MCM is designed to be a multiplication free (multiplierless) by substituting the explicit multiplication with operations of shift and add/subtract.

## III. PROPOSED METHODOLOGY

To solve the multiplier problem and to perform high speed execution of multiplication operation this work presents a FPGA implementation of Fast MCM Calculations using area and power efficient weighted partitioning algorithm. Figure 3.1 shows the top module of proposed work implemented in Xilinx 13.1 design suite. Let an un-weighted diagram G with V vertices and E edges and given a number k, the Graph Partitioning issue is to isolate the V vertices into k parts to such an extent that the quantity of edges interfacing vertices in various parts is optimized given the condition that each part contains generally a similar number of vertices. If the graph is weighted, i.e. the vertices and edges have weights associated with them; the problem requires the sum of the weights of the edges connecting vertices in different parts to be minimized given the condition that the sum of the weights of the vertices in each part is roughly the same.

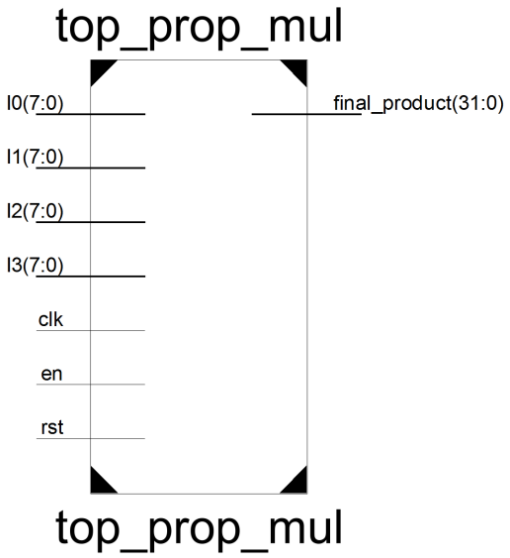


Figure 3.1 Top Module of the Proposed Architecture

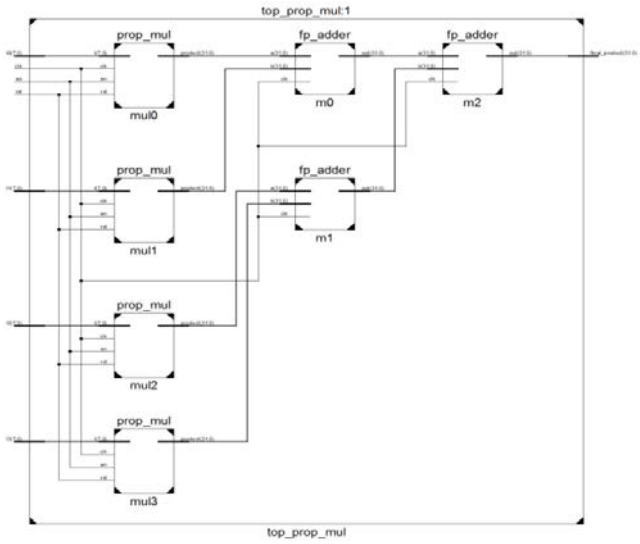


Fig.3.2 Sub-modules of Proposed Architecture

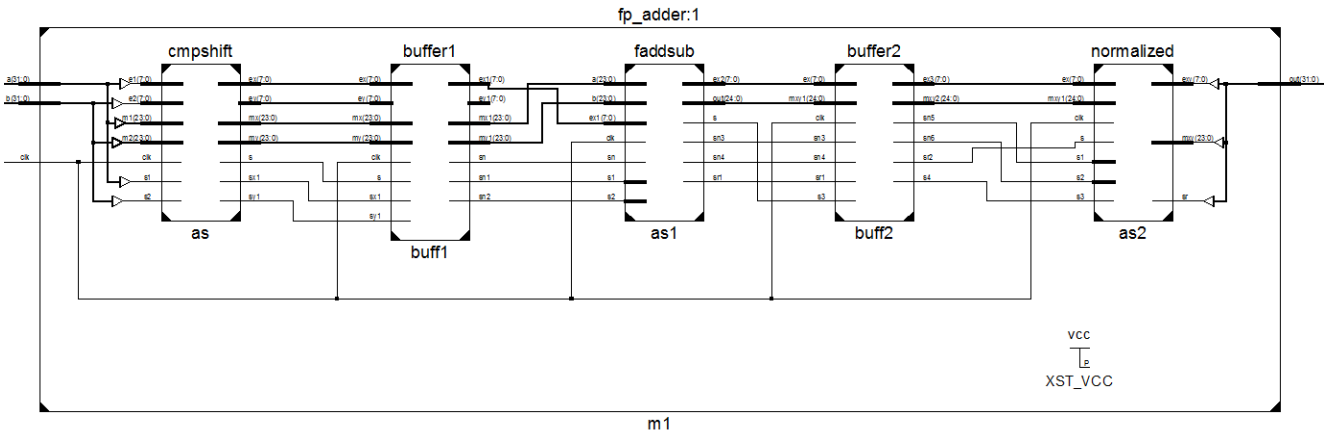


Figure 3.3 Propose Koge Stone Adder RTL Schematic.

The issue can be mnimized into that of division where the graph is split into two sections and after that each part is additionally cut up utilizing a similar methodology recursively.

Figure 3.3 demonstrate the usage of Kogge-stone adder. It can be arranged a parallel prefix adder since generate and the propagate signals are pre figured. In a tree-based adder, carries are created in tree and fast calculation is achieved to the detriment of expanded zone and power. The main advantage of this design is that the carry tree reduces the logic depth of the adder by essentially generating the carries in parallel.

The important advantage of using KSA is its mutual exclusive nature. If an fault is detected in the one-portion of the carry tree, the other half can be used to register the conveys for both the even and odd carries.

Figure 3.2 Show the sub module of proposed method and Figure 3.3 shows the RTL Schematic of proposed Koge Stone Adder.

## IV. SIMULATION RESULTS

Simulation and results analysis of proposed algorithm has carried out in Xilinx ISIM simulator. The performance of proposed work has been evaluated based on following simulation parameters such as are in terms of number of number of slice registers and LUTs. Power dissipation in proposed design has calculated in Xilinx Xpower analyzer and delay of circuit in ns. The parameters are calculated based on simulation of proposed design in Xilinx and results are compared with existing results. Simulation screen of XILINX Project View GUI with Device Utilization Summary has shown in Figure 4.1.

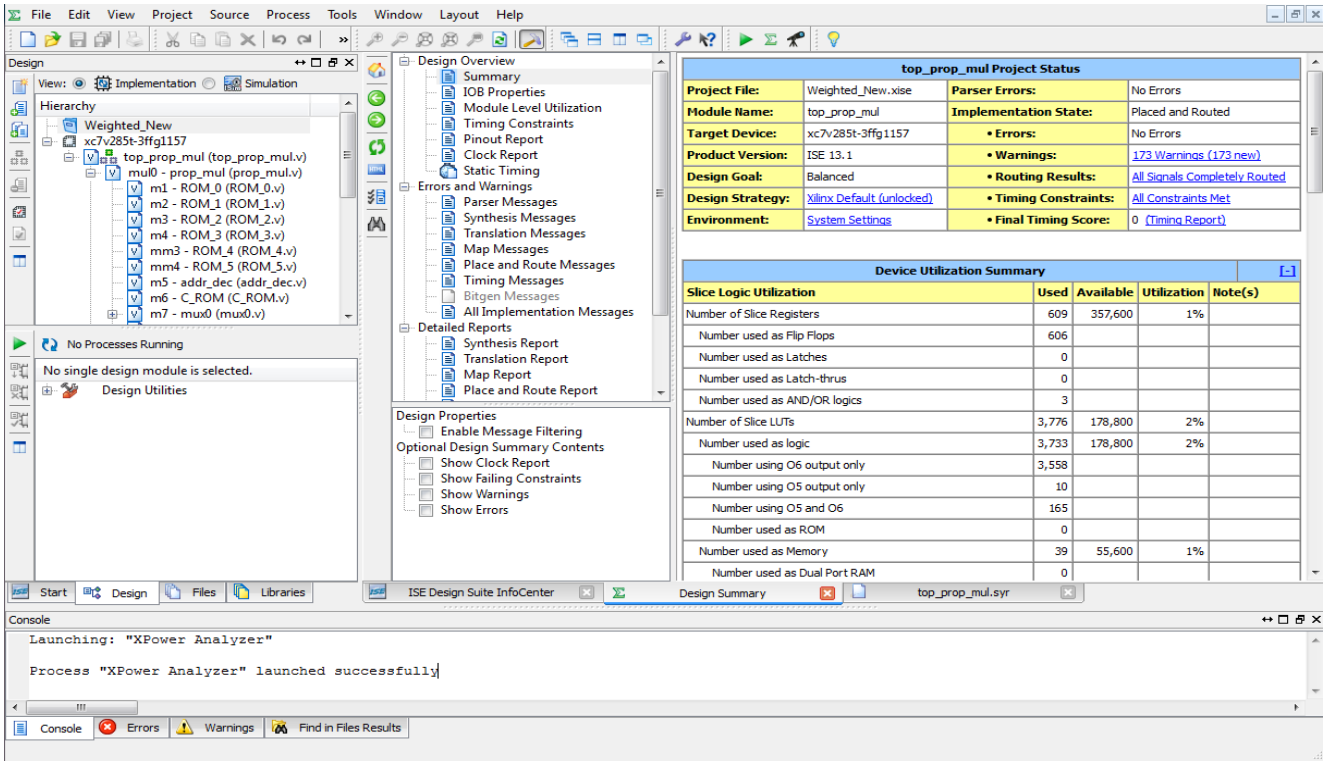


Figure 4.1 XILINX Project View GUI with Device Utilization Summary.

The Performance Comparison among Previous and Proposed Architectures are shown in Table 4.1. The summary of power utilization on Xilinx Xpower analyzer screen is shown in figure 4.2.

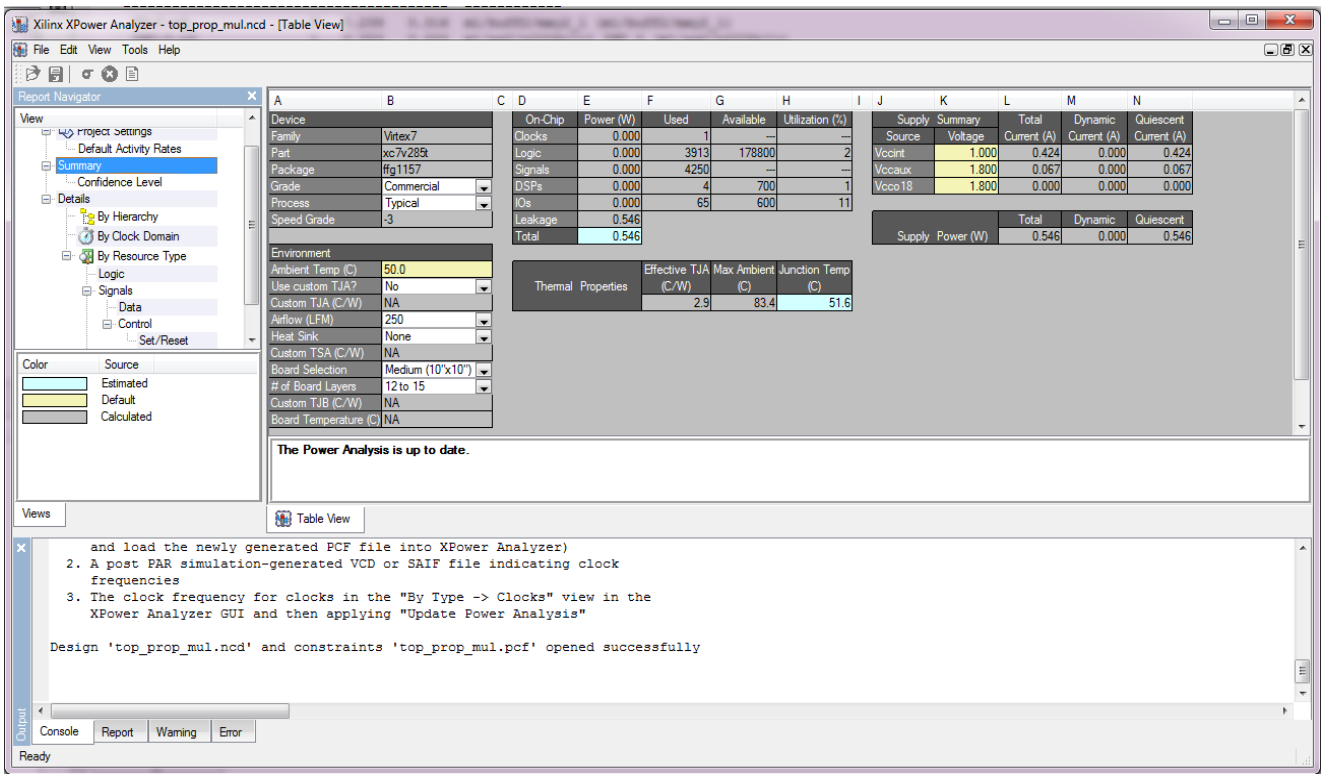


Figure 4.3 XILINX XPower Analyzer - Power Utilization.

```
Device utilization summary:
---------------------------
Selected Device : 7v285tffg1157-3

Slice Logic Utilization:
 Number of Slice Registers:             606  out of  357600     0%
 Number of Slice LUTs:                 3883  out of  178800     2%
    Number used as Logic:              3844  out of  178800     2%
    Number used as Memory:               39  out of  55600      0%
       Number used as SRL:               39

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   4049
   Number with an unused Flip Flop:    3443  out of   4049    85%
   Number with an unused LUT:           166  out of   4049     4%
   Number of fully used LUT-FF pairs:   440  out of   4049    10%
   Number of unique control sets:         5

IO Utilization:
 Number of IOs:                          67
 Number of bonded IOBs:                  65  out of    600    10%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     32     3%
 Number of DSP48E1s:                      4  out of    700     0%
```

Figure 4.3 Device Utilization Summaries.

Table 1: Performance Comparison among Previous and Proposed Architectures

| Parameters | Previous Architecture[1] | Proposed Architecture |
|---|---|---|
| Platform | Virtex 7 | Virtex 7 |
| Power (W) | 2.317 | 0.546 |
| LUTs | 8654 | 3883 |

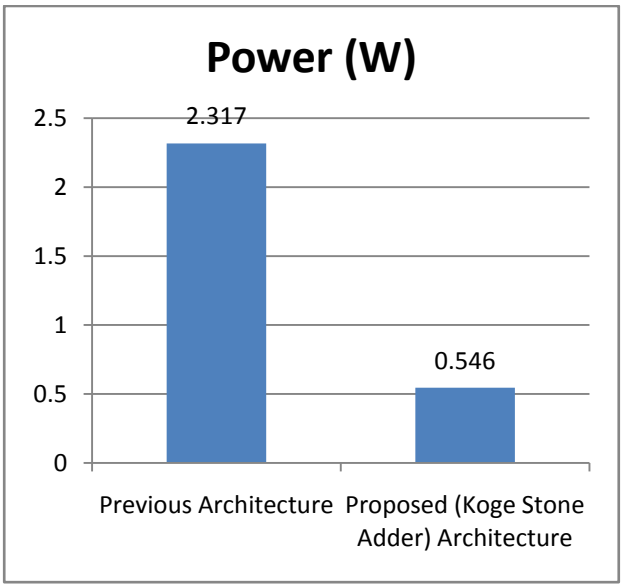


Figure 4.4 Comparison of Power (W)

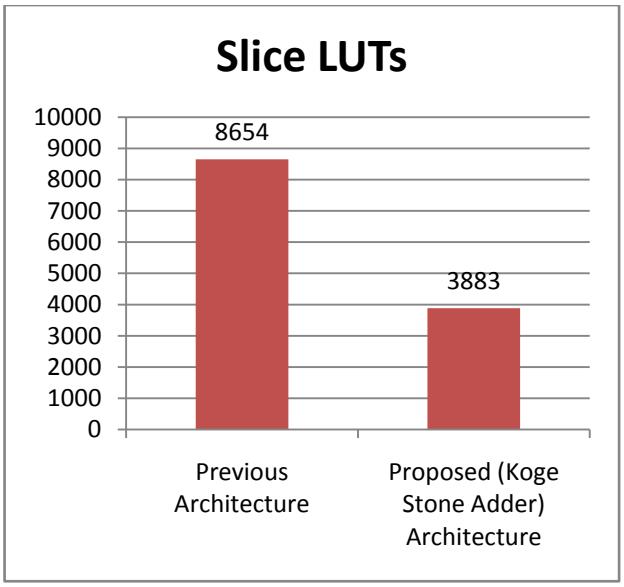


Figure 4.5 Comparisons of Slice LUTs.

## CONCLUSION

In this work, implementation and synthesis of proposed graph partition algorithm in order to partition a multiple constant multiplier design has done using distributed arithmetic. Graph Partitioning is an essential problem since it finds broad applications in many designing domains, including scientific computing, VLSI design and task scheduling. One important application is the reordering of sparse matrices. The fundamental objective of this work is a new MCM algorithm that achieves significantly better results than previous methods based on comparison and results analysis. The area and power efficient weighted partitioning for fast MCC calculations framework

developed in this investigation should be useful for further research in this following domain. One direction could be to improve the heuristic, which currently combines A-distances in a trivial way. Another direction would be to use our framework to optimize MCM blocks with respect to other criteria such as critical path or to also minimize for the number of shifts required.

## REFERENCES

[1]. G. D. Licciardo, C. Cappetta, L. Di Benedetto and M. Vigliar, "Weighted Partitioning for Fast Multiplierless Multiple-Constant Convolution Circuit," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 1, pp. 66-70, Jan. 2017.

[2]. F. L. T. Maggioni, H. Oprins, D. Milojevic, E. Beyne, I. De Wolf and M. Baelmans, "3D-convolution based fast transient thermal model for 3D integrated circuits: methodology and applications," 2015 31st Thermal Measurement, Modeling & Management Symposium (SEMI-THERM), San Jose, CA, 2015, pp. 107-112.

[3]. M. A. Hatwar, T. G. Panse and S. Pothuri, "Design of common sub expression elimination algorithm in fast Fourier transform," 2015 International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, 2015, pp. 1703-1707.

[4]. M. Garrido, F. Qureshi and O. Gustafsson, "Low-Complexity Multiplierless Constant Rotators Based on Combined Coefficient Selection and Shift-and-Add Implementation (CCSSI)," in IEEE Transactions on Circuits and Systems I: Regular Explorations, vol. 61, no. 7, pp. 2002-2012, July 2014.

[5]. R. Jeni, T. Selvin Retna Raj and R. Solomon Roach, "Design of high performance folded DIF FFT architecture using MMCM approach with Hcub algorithm," 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, 2013, pp. 715-719.

[6]. S. Shen, W. Shen, Y. Fan and X. Zeng, "A Unified 4/8/16/32-Point Integer IDCT Architecture for Multiple Video Coding Standards," 2012 IEEE International Conference on Multimedia and Expo, Melbourne, VIC, 2012, pp. 788-793.

[7]. J. Hazra, "An efficient design technique of circular convolution circuit using Vedic Mathematics and McCMOS technique," 2012 International Conference on Computer Communication and Informatics, Coimbatore, 2012, pp. 1-5.

[8]. J. Hazra, D. Kayal, A. Dandapat and C. K. Sarkar, "Design of a high speed low power linear convolution circuit using McCMOS technique," 2011 International Conference on Multimedia, Signal Processing and Communication Technologies, Aligarh, 2011, pp. 276-279. S.

[9]. L. Chen, “VLSI implementation of an adaptive edge-enhanced image scalar for real-time multimedia applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 9, pp. 1510–1522, Sep. 2013.

[10]. F. C. Huang, S. Y. Huang, J. W. Ker, and Y. C. Chen, “High perfor- mance SIFT hardware accelerator for real-time image feature extraction,” IEEE Trans. Circuit Syst. Video Technol., vol. 22, no. 3, pp. 340–351, Mar. 2012.

[11]. G. D. Licciardo, A. D’Arienzo, and A. Rubino, “Stream processor for real- time inverse tone mapping of full-HD images,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 11, pp. 2531–2539, Nov. 2015.

[12]. M. Vigliar and G. D. Licciardo, “Hardware coprocessor for stripe-based interest point detection,” US Patent 20 130 301 930, Nov. 14, 2013.

[13]. K. K. Parhi, VLSI Signal Processing Systems: Design and Implementation. New York, NY, USA: Wiley, 2007.

[14]. B. C. Paul, S. Fujita, and M. Okajima, “ROM-based logic (RBL) design: A low-power 16 bit multiplier,” IEEE J. SolidState Circuits, vol. 44, no. 11, pp. 2935–2942, Nov. 2009.