International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

https://dx.doi.org/10.12785/ijcds/110122

# VLSI Architectures of Booth Multiplication Algorithms – A Review

**B Hareesh**[1], **John Moses C**[2] **and MVV Prasad Kantipudi**[3]

[1]*Electronics and Communication Engineering, Sreyas Institute of Engineering, Hyderabad, India*
[2]*Electronics and Communication Engineering, Sreyas Institute of Engineering, Hyderabad, India*
[3]*Department of Electronics and Telecommunication Engineering, Symbiosis Institute of Technology,*
*Symbiosis International (Deemed University), Pune 412115, India*

*Received 16 May 2020, Revised 13 Sep 2021, Accepted 13 Nov. 2021, Published 9 Jan. 2022*

**Abstract:** The Booth multiplication scheme plays a major role in designing signed multiplier using multiplier encoder and by decreasing the number of intermediate products. Both radix-4 and radix-8 Booth encoding schemes are widely used due to simple and fast respectively. Multiplier is one of the basic as well as an important part in arithmetic unit of many high- performance operations like digital signal processing (DSP) and digital image processing (DIP) and other high-performance central processing unit (CPU) operation. In the past decade numerous ways of Booth multiplier circuits have been implemented by using different application specific integrated circuit (ASIC) technology like Taiwan semiconductor manufacturing technology (TSMC) 45 nm and 65 nm complementary metal oxide semiconductor (CMOS) process and some of the implementations have been proposed by field programmable gate array (FPGA). This work analyses the very large-scale integration (VLSI) characteristics such as area utilization, power consumption and speed of operation of different types of implementation of Booth multiplication scheme. Based on the exhaustive examination on Booth multiplication scheme, it is noticed that the recent implementation of approximate computing-based and modified two's complementor-based multiplication algorithms outperform other multiplication schemes. Further, the VLSI technology using ST Microelectronics (STM) 28 nm and TSMC 45 nm CMOS processes beat the other implantation schemes by providing less-area and power as well as high-speed of multiplication, respectively.

**Keywords:** : Booth Encoder, FPGA, Multiplier, Radix-8, Low-power design, ASIC

## 1. Introduction

Very Large-Scale Integration (VLSI) technology is used for producing application specific integrated circuit (ASIC) for any kind of signal processing and multimedia applications. In modern multimedia applications, the performance, power utilization and area of the chip are important factors for fabrication the circuit. Low-power utilization and less-area are the most important consideration for designing multimedia devices [1]. The low-energy devices are widely used in many digital signal processing applications like discrete cosine transform (DCT), finite impulse response filter [2], fast Fourier transform (FFT), speech recognition, computer vision and biomedical imaging [3]. It is also used in internet-of-things (IoT) - based devices [4], pattern recognition and machine learning [5]. Energy consumption and area utilization are most important factors in multimedia application due to portable and for extending battery life and reducing operation cost [6]. Multiplication using finite field multiplication on GF (2m) is mostly utilized in more fields like crypto processors [7] remote health care systems [8]. Elliptic curve cryptographic (ECC) is widely used for smart phones and e-commerce [9]. Further, energy-efficient, or low-power multiplication schemes can also be used to implement data mining, data analytic, image processing applications and common communication devices like antenna.

Multiplication takes an important role in all arithmetic processors of any computational devices and it is overly expensive and generally it utilizes complex operations and time consuming. The general multiplication algorithm used addition and shift operations to carry out its operation. The structure of any multiplication algorithm is divided into three major steps such as generating partial products, dropping partial products, and adding the different stages partial product. The partial products are generated by using a multiplicand and with and without recoded multiplier. The final addition operation is mostly performed by high-performance adders like carry save adder. The performance of the multiplication is normally based on number of partial products or the length of multiplier operand [1]. Different types of partial product addition schemes are used such as

*E-mail address: : hareeshb@gmail.com,erjohnmoses@gmail.com,prasad.kantipudi@sitpune.edu.in*

Dadda tree, Wallace tree and carry save [10]. Multipliers are vital arithmetic elements in advanced products, for two main purposes. Primarily, they are categorized by complicated logic model, and it decides the demand of energy for data processing elements of advanced microprocessors. Again, it computes intensive calculations to perform the desired operations[11].

Booth multiplication algorithm treats both signed and unsigned numbers uniformly. The main purpose of Booth algorithm is used to reduce number of partial products by reducing number of multiplier bits. To reduce the number of multiplier bits, it used different recoding techniques such as radix-2, radix-4 and radix-8. Over the recent years, different researchers proposed different kind of architecture to perform booth multiplication with different speed, area occupation and power consumption [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. Among three parameters energy utilization and area occupation get more importance to fabricate chips for consumer electronic devices like mobile handset. This research aims to analyse the VLSI characteristics of different recent architectures of Booth algorithm and identifies suitable low-energy architecture for further optimization.

The remaining portion of this manuscript is organized as follows. Section 2 describes the different recent developments on Booth multiplication algorithms. Section 3 gives the problems in the existing methods. Section 4 gives the objectives of this project work. Section 5 illustrates the proposed method for implementing Booth algorithm and Section 6 provides the work schedule of this project work.

## 2. Architectures Of Multiplication Algorithms

Over the past decade several Booth multiplication architectures have been implemented for different applications like digital signal processing, digital image processing and multimedia operations. This section briefs the characteristics of different recent architectures of Booth multiplication algorithm.

Kuang and Wang (2010) presented an energy-efficient configurable Booth multiplier by deactivating the redundant switching activities. Further the power reduction is achieved by truncating the output product. This multiplication algorithm is synthesized by using Synopsis Design Compiler with the TSMC 0.13 $\mu$m complementary metal oxide semiconductor (CMOS) technology. This scheme saves power but due to its complexity it increases the delay and area from conventional multiplication scheme [12].

Seo and Kim (2010) realized the hardware architecture of parallel multiplier-accumulator using radix-2 modified Booth algorithm for high-performance arithmetic operations. The hardware architecture of this algorithm is implemented by using different techniques such as 90 nm, 130 nm, 180 nm and 250 nm CMOS processes. The multiplier with 90 nm CMOS process utilizes 1819 gates. By comparing with other CMOS processes this 90 nm CMOS process utilizes a smaller number of gates [13].

Muralidharan and Chang (2011) proposed modulo 2n - 1 multiplier for residue number system (RNS) [14]. The hardware architecture of this multiplier is implemented by TSMC 0.18 $\mu$m CMOS process by saving both area and power utilization for the RNS with different word length. To perform modulo 2n - 1 multiplication, this architecture utilizes 267099 $\mu m^2$ area, 51.71 mW dynamic power.

Chen et al (2011) applied a Generalized Probabilistic Estimation Bias (GPEB) fixed-with multiplication algorithm to perform DCT. This multiplication scheme saves 18 % of area with 0.8dB peak signal to noise ratio (PSNR).The hardware architecture of this multiplication scheme is implemented by using 0.18 $\mu$m technology with the power consumption of 15.7 mW at the operating frequency of 55 MHz. [15].

Chen and Chang (2012) applied an adaptive conditional-probability estimator (ACPE) for large-length Booth multiplier with fixed width. The ACPE gives varying information about column width (w) to reduce area utilization. This multiplication scheme is analysed by applying for discrete cosine transform (DCT) which saves 14.3 % of area with less peak signal to noise ratio (PSNR) as compared with post-truncated Booth multiplication scheme [16].

Ramkumar and Kittur (2013) implemented a modified Booth encoder (MBE) multiplier using the technique of partitioning the partial products and a low-complexity hybrid adder for digital systems [17].The ASIC design of this multiplier is performed by using Cadence tool with TSMC 65nm CMOS technology. The 32-bit MBE multiplier utilizes chip area of 15721 $\mu m^2$ and 5.58 $\mu$W power for the delay of 2.19 ns.

Muralidharan and Chang (2013) also deigned another muti-modulus multiplier scheme by using radix-4 and radix-8 Booth encoding technique for residue number system (RNS) [18]. To, reduce area, this method uses a technique of reusing hardware resources. This multiplier architecture is realized by using Synopsis Design Compiler with TSMC 0.18 $\mu$m CMOS process. The area delay and power consumption by this architecture are tabulated as 288814 $\mu m^2$, 5.14 ns and 2494 $\mu$W respectively to perform radix-2k multiplication for RNS when k =3 and n = 64.

Chen et al 2013 designed and implemented an energy-efficient variable-latency speculating Booth multiplier (VLSBM) to increase the functionality in a gloomy process [19]. To reduce complexity the VLSBM uses a technique of portioning the partial products into least significant part as well as most significant part. To improve the performance of operation, it uses pipelining techniques and to reduce the path delay, it uses carry prediction technique. The hardware architecture of this method is realized by CMC 90 nm CMOS process. It saves 25.4 % of energy and 7 % of area in multimedia applications like object detection and JPEG compression. Choi et al 2014 used a hybrid radix-4/-8 truncation-based Booth multiplier for graphical processing unit (GPU) applications. This hybrid logic is introduced by sharing the common logics of both radix-4 and radix-8 encoding scheme. This reduces power consumption by 60.7 % in mobile multimedia applications [20].

Chen 2014 presented a fixed-width Booth multiplier using multi-level conditional probability (MLCP) for accuracy-tuning and to compensate the truncation fault in digital signal processing applications [21].The hardware architecture of this fixed-width Booth multiplication algorithm is implemented by using TSMC 0.18 $\mu$m technology. The 16 x 16 Booth multiplications by using fixed width utilize 2.9 K gates with 10 ns path delay and 8.3 mW power.

Jiang et al 2015 introduced an energy-efficient with high-performance radix-8 Booth multiplication algorithm by using approximation technique. This multiplication algorithm uses an approximate 2-bit addition for generating a triple multiplicand without carry propagation. This multiplication algorithm outperforms the exact Booth multiplication scheme in terms of hardware utilization. The hardware architecture of this algorithm is implemented by using STM 28 nm CMOS technology for FIR filter operation [22].

An area-efficient with low-power consumption Booth multiplication algorithm is provided by Hardidas and George (2016) for finite impulse response (FIR) filter design. This method realizes a spanning tree-based Booth multiplication algorithm to reduce the area of the FIR filter. This design is implemented by Xilinx and MATLAB Simulink tools. By using modified spanning tree adder, this method reduces the area by 29.10 % from conventional FIR filter. Further it reduces the power consumption by 3.03 % [23].

Liu et al 2016 designed an approximate radix-4 Booth Multiplier for error-tolerant applications using 45 nm CMOS technology. This design uses an approximate Wallace tree architecture for the accumulation of partial products. This technique improves the power-delay-product by 59% by introducing inexact terms in the truth table. Further, this design is applied for image processing systems [24].

Mirhosseini et al (2016) designed a radix-8 modulo 2n + 1 multiplier by reducing the number of bias terms and by using a parallel prefix architecture for calculating carries lone used for odd positions [25] .The hardware structure of this multiplication algorithm is implemented by using Cadence RTL compiler with TSMC 65 nm process by reducing area time product as compared with the other radix-8 multiplier. This multiplication method uses 16691.8 $\mu m^2$ area and 17.38 mW power for the operation with 1.318 ns delay.

Zhang and He (2017) proposed a fixed-width Booth multiplier by using Booth encoded sign-digit-based conditional probability (BSCP) assessment to accuracy on digital signal processing applications [26]. This multiplication algorithm utilizes error distribution and multiplexer-based estimation. This multiplication algorithm is implemented by Synopsis Design Compiler with 32 nm CMOS process. A 16-bit multiplication using sign-based restricted probability works with 1.95 ns delay for the utilization of 452.8 $\mu$W power.

Patil and Kulkarni (2018) implemented a multiply accumulation unit (MAC) using Rdix-4 Booth encoding for digital signal processing (DSP) system. This method is implanted on field programmable gate array (FPGA) (Spartan 6-XC6LX9-2TQG144). This implementation outperforms the conventional pipeline-based MAC unit in terms of delay [27].

Liu et al 2018 proposed a design of approximate multiplication using approximate Booth encoder for error-tolerant applications by using CMOS 45 nm technology. This method uses a modified Radix-4 modified Booth encoder, redundant binary approximate 4:2 compressor and a redundant binary (RB) to normal binary (NB) converter. This approximate Booth multiplication is used for FIR filter application and this method reduces power consumption by 64 % [28].

Xue et al 2018 presented a low-power-delay-product radix -4 8x8 multiplier by using CMOS 90 nm technology. This method uses a modified low-complexity binary-to-two's complement converter and multiplexer in one the stages in multiplication instead of conventional adder. By using this low-complexity adder the delay is reduced and due to the low-delay it generates a low-power-delay product [29].

Radix-4, serial - parallel multiplier is designed by Moss et al (2018) for improving the performance of different applications like filtering, machine learning and neural network-based systems [30]. This multiplier is realized by Intel Cyclone V FPGA for 32- bit and 64-bit operations. This multiplication systems work on another data - path with two different sub-circuits and so it is called as two-speed multiplier. The sub-circuit consumes 292 logic elements on FPGA and 2.23 mW power with the delay of 3.9 ns for 64-bit operation. The second-sub-circuit utilizes 159 logic elements with 3.18 mW power consumption and 45.05 ns delay.

Barrio et al 2018 presented a partial carry-save radix-8 Booth multiplier for data-paths by splitting the overall operations into different number of fragments to perform in parallel for improving the performance of multiplication operation [31]. The on-the-fly correcting radix-8 multi-speculative multiplier with nine fragments (OMSM-B-8k-9) is implemented by using Synopsis Design Compiler with 65 nm CMOS process. The 32-bit OMSM-B-8k-9 increases the performance of multiplication with the delay of 1.04 ns. This multiplication scheme outperforms the previous radix-

4 OMSM.

Another approximate Booth Multiplier is proposed by Venkatachalam et al (2019) by using radix-4 encoding scheme. This method reduces both the complexity on partial product generation as well as the complexity on partial product summation. This method outperforms the other approximate Booth multiplication algorithms by reducing 56 % of area and 46 % of power consumption on 32-bit multiplication operation. This multiplication algorithm is synthesized by using Synopsis Design Compiler with TSMC 65 nm process [32].

Double- least significant bit two's complement multiplier is proposed by Leon et al (2019) to reduce power and area utilization from the conventional radix-4 Booth multiplier. The hardware architecture of this multiplication algorithm is implemented by using Synopsis Design Compiler with TSMC-90 nm, TSMC 65-nm and TSMC45 nm CMOS processes [33]. Among the different standard CMOS processes, the 45nm CMOS process utilizes less area with reduced delay. The power delay product of this multiplication scheme using TSMC 45nm is 4411.95 ($\mu$ W.ns).

## 3. Comparative Analysis

This section provides a comparison of different characteristics of various Booth multiplication scheme such as Booth multiplier using TSMC 0.13-$\mu$m CMOS, TSMC 0.18$\mu$m CMOS, STM 28nm CMOS, 45nm CMOS, 90 nm CMOS and FPGA technologies. Table 1 illustrates the characteristics of different Booth multiplication architectures such as configurable Booth multiplier (CBM), parallel multiplier-accumulator (PMA), modulo 2 n -1 multipliers (M2 n - 1), modulo 2n + 1 multiplier (M 2n + 1 M), different arithmetic-based multiply-accumulate unit (DABMAU), approximate redundant binary multiplier (ARBM), fixed–width Booth multiplier (FWBM), approximate Booth Multiplier (ABM), spanning tree-based modified Booth multiplier (STBMBM), HPM-based Signed multiplier (HPMBSM), multi-modulus multipliers (MMM), VLSBM, truncated multiplier (TM),approximate redundant binary multiplier (ARBM), binary to 2's complement-based Booth multiplier (B2CBBM), serial-parallel multiplier (SPM), Hybrid Radix-4/8 truncated multiplier (HTM), OMSM-B8-k9, and DLSB. Among the various technologies, TSMC 0.18$\mu$m and 45nm CMOS techniques are widely used to implement Booth multipliers.

As shown in Table 1 most of the Booth multiplication algorithms used for various multimedia operations like digital signal processing (DSP), dital image processing (DIP) and network security. FWBM [15] is used for discrete cosine transform (DCT) with the PSNR of 53.7 dB. HTM [20] is utilized for joint photographic experts group (JPEG) with the PSNR of 47.44 dB. FWBM [21] is suitable for fast-Fourier transform (FFT). Both ABM and STBMBM [23] are used for finite impulse responnse (FIR) filter. SPM [30] is used for implementing filter as well as meachine learning (ML) algorithms. DABMAU [27] is applied for desigining discrete wavelet transform (DWT) and B2CBBM [29] is suitable for implementing graphical processor units (GPU). Furthermore, the multiplication algorithms that are used for DSP and DIP are use approximate technique as the approximate alogoritms require simple acircuits with less accuracy on multiplication. The other mulplication techniques such as HPMSBM [17], MMM [18], M 2n + 1 M [25] and DLSB [33] are utilized for different arithmetic operations with higher accuracy. OMSM-B8-K9 [31] is specifically used to design datapath unit.

Fig. 1 to Fig. 6 illustrate that different characteristics of various Booth multiplication algorithms and their comparison. Fig. 1 demonstrates that the TSMC 0.18 $\mu$m technology is used a greater number of times to implement multiplication operation. However, in recent years the CMOS technologies 65 nm, 40 nm and 90 nm are widely used to provide high-performance and low-area multipliers [29], [31], [32] and [33] as compared with other CMOS process like 0.18 $\mu$m.

The different proposals of multipliers have also been implanted by different width on final products such as 16-bit, 32-bit and 64-bit. Both 16-bit and 32-bit multiplication schemes are widely used by comparing with other bit sizes. Figure 2 and Fig. 3 shows the delay and power utilization characteristics of different 32-bit multipliers, respectively.

Based on Fig. 2, it is identified that the recently developed 16-bit multiplier B2CBBM [29] utilizes less power as compared with another recently developed multiplier FWBM [26]. The high-performance B2CBBM [29] is implemented by TSMC 90 nm CMOS process. This multiplier reduces the complexity of circuit operation by using low-complex two's complementor.

From Fig. 3, it is found that the recently developed multipliers ABM [24] and B2CBBM [29] use less power as compared with another moderately power consumed multiplier FWBM [26]. Therefore, based on Figure 2 and Figure 3 it is confirmed that the recently developed 16-bit multiplier B2CBBM [29] is suitable for high-performance and low-power applications.

The high-performance and low-power Booth multiplication architecture [29] uses four stages of multiplication as in convention Booth multiplication using 2's complementor. This multiplication scheme uses four stages as in convention 2's complementor-based Booth encoding. But it utilizes a smaller number of operations. In the first stage it uses an optimized 2's complementor and 9-bit addition/subtraction unit instead of using 15-bit addition/subtraction unit and it also eliminates the need of a shifter. The second, third and fourth stages use same architecture by using radix-4 encoder, 9-bit 3-to-1 multiplexer and 9-bit addition/subtraction unit.

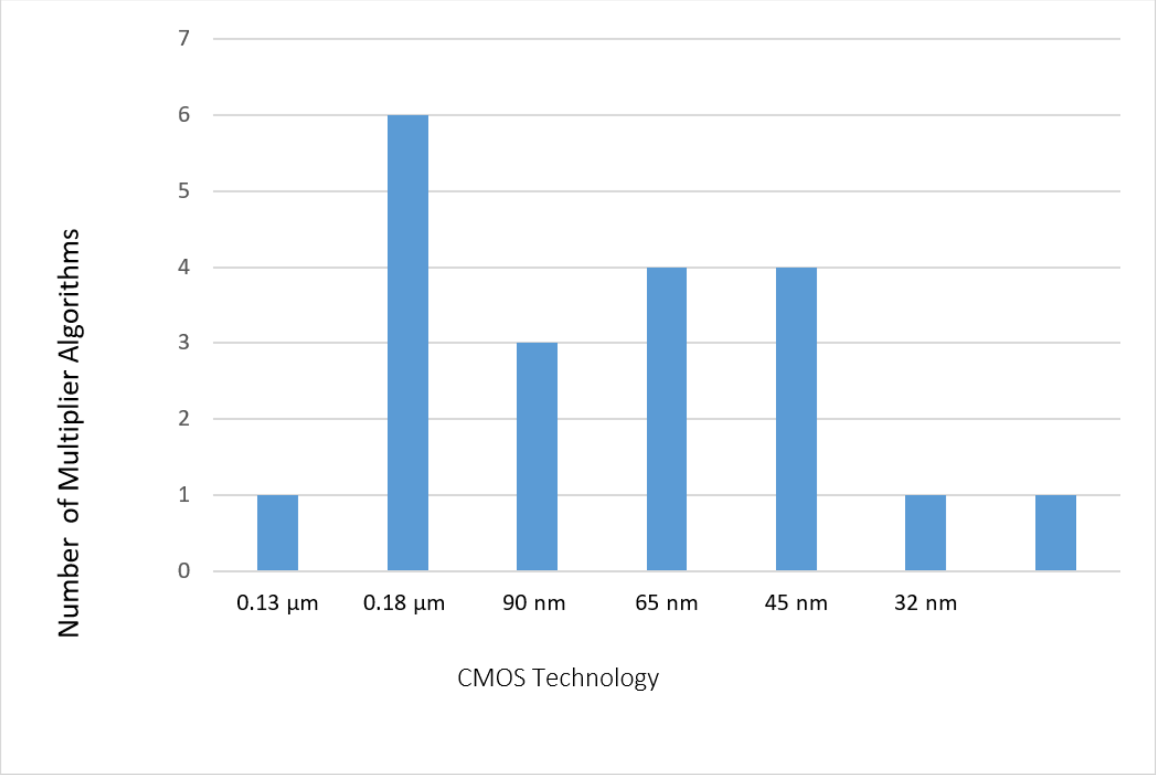

Figure 1. Different CMOS technologies and their usage to implement multiplier circuit

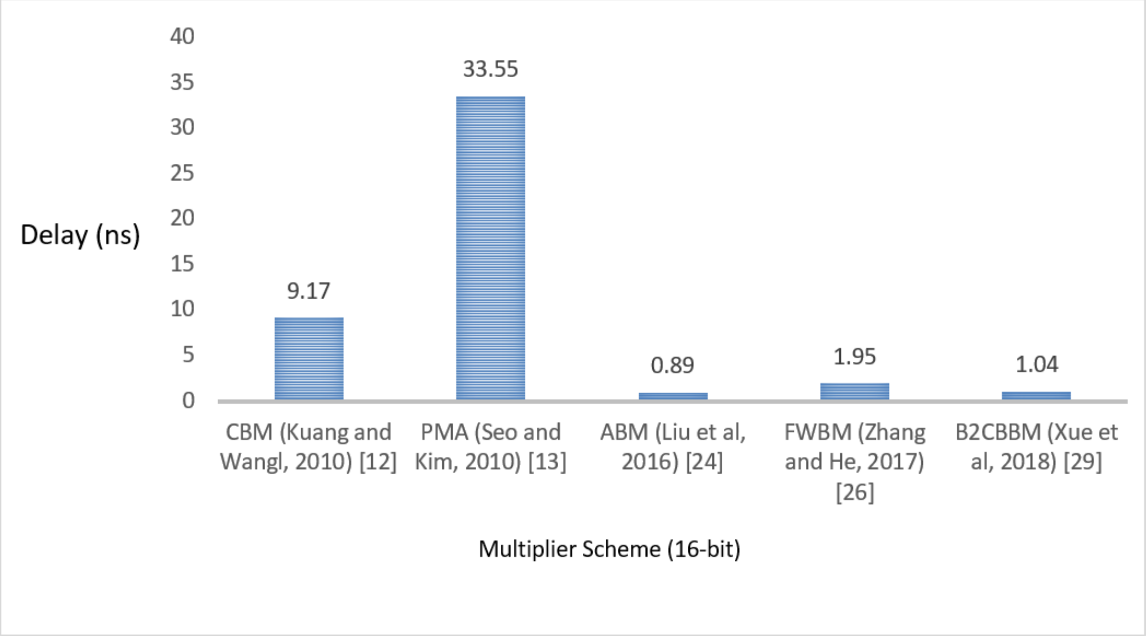

Figure 2. Comparison of data-path delay of different 16-bit multiplication algorithms

TABLE I. COMPARISON OF VLSI CHARACTERISTICS OF DIFFERENT BOOTH MULTIPLICATION ALGORITHMS

| # | Multiplication Scheme | Technology used / Device used | Area/ No. of Gates | Power | Delay | Applications |
|---|---|---|---|---|---|---|
| 1 | CBM [12] | TSMC 0.13 $\mu$ m CMOS | 19995 $\mu m^2$ | 0.861 mW | 9.17 ns | DSP |
| 2 | PMA [13] | 90 nm CMOS | 1819 gates | NA | 33.55 ns | DSP |
| 3 | $M2^n$ -1M [14] | TSMC 0.18 $\mu$m CMOS | 267099 $\mu m^2$ | 51.71 mW | 5.13 ns | Cryptography |
| 4 | FWBM [15] | GPEB 0.18 $\mu$m 1P6M CMOS | 18 k | 15.7 mW @ 55MHz | NA | DIP/DCT 53.7 dB |
| 5 | FWBM [16] | ACPE TSMC 0.18 $\mu$m CMOS | 18 k | 17.6 mW @ 55MHz | NA | DIP/DCT |
| 6 | HPMBSM [17] | TSMC 65 nm | 15721 $\mu m^2$ | 5.58 mW | 2.19 ns | Arithmetic |
| 7 | MMM [18] | TTSMC 0.18 $\mu$m CMOS | 288814 $\mu m^2$ | 2494$\mu$W | 5.14 ns | Arithmetic |
| 8 | VLSBM [19] | UMC 90 nm CMOS | 15652 $\mu m^2$ | NA | 1.50 ns | Communication/Cryptography |
| 9 | HTM [20] | TSMC 0.18 $\mu$m CMOS | 111690 $\mu m^2$ | 1.22 mW@ 10MHz | 3.91 ns | JPEG Compression /Object detection 47.44 dB |
| 10 | FWBM [21] | MLCP TSMC 0.18 $\mu$m | 2.9 K | 8.3 mW | 10 ns | FFT/DCT |
| 11 | ABM [22] | STM 28 nm CMOS | 419 $\mu m^2$ | 206.8 mW | 2.07 ns | FIR Filter |
| 12 | STBMBM [23] | FPGA/ Spartan 3E | 770 (No. of 4 i/p LUT) | 0.032 W | 1.933 ns | FIR Filter |
| 13 | ABM [24] | TSMC 45 nm CMOS | 1788 $\mu m^2$ | 437.4 $\mu$W | 0.89 ns | DIP/Filtering |
| 14 | ($M2^n$ + 1 M) [25] | TSMC 65 nm | 16691.8 $\mu m^2$ | 17.38 mW | 1.318 ns | Arithmetic |
| 15 | FWBM [26] | BSCP 32 nm CMOS | NA | 452.8 $\mu W$ | 1.95 ns | DSP |
| 16 | DABMAU [27] | TSMC 45 nm CMOS | 513 (slice registers) | NA | 248.231 MHz | DWT/DCT |
| 17 | ARBM [28] | TSMC 45 nm CMOS | 7683.4 $\mu m^2$ | 1.99 mW | 1120 ps | DIP/Filtering |
| 18 | B2CBBM [29] | TSMC 90 nm CMOS | NA | 435.9 $\mu W$ | 1.04 ns | DSP/GPU |
| 19 | SPM [30] | Intel Cyclone V FPGA | 292 (LEs) | 2.23 mW | 3.9 ns | Filtering/ML |
| 20 | OMSM-B8-k9 [31] | TSMC 65 nm | NA | NA | 1.04 ns | Datapath |
| 21 | ABM [32] | TSMC 65 nm | 1830 $\mu m^2$ | 0.7 mW | NA | JPEG Compression |
| 22 | DLSB [33] | TSMC 45 nm | 5256 $\mu m^2$ | 4411.95 ($\mu$W.ns) | 0.67 ns | Arithmetic |

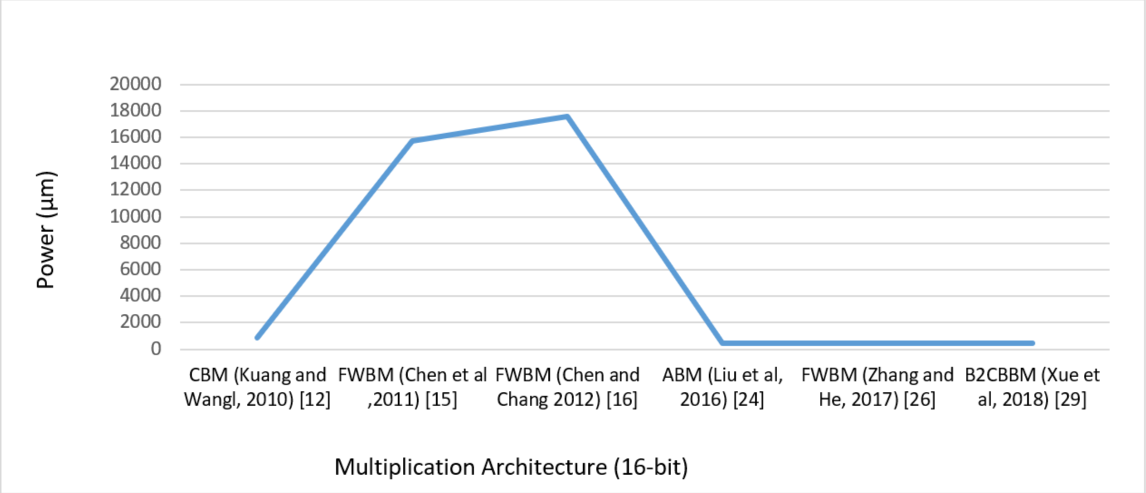

Figure 3. Evaluation of power improvement on different16-bit multiplication algorithms

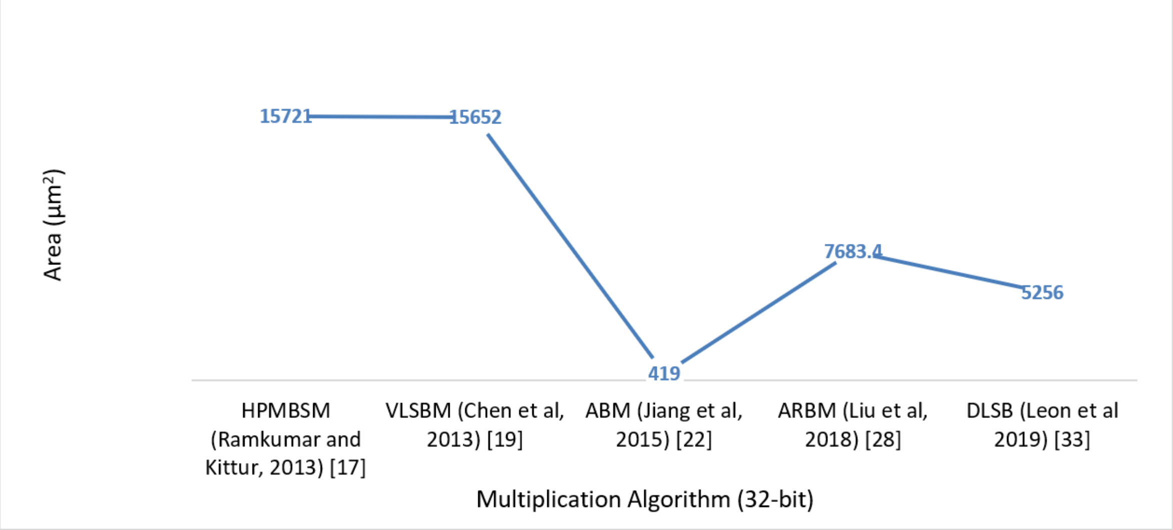

Figure 4. Investigation of area utilization by various32-bit multiplier architectures

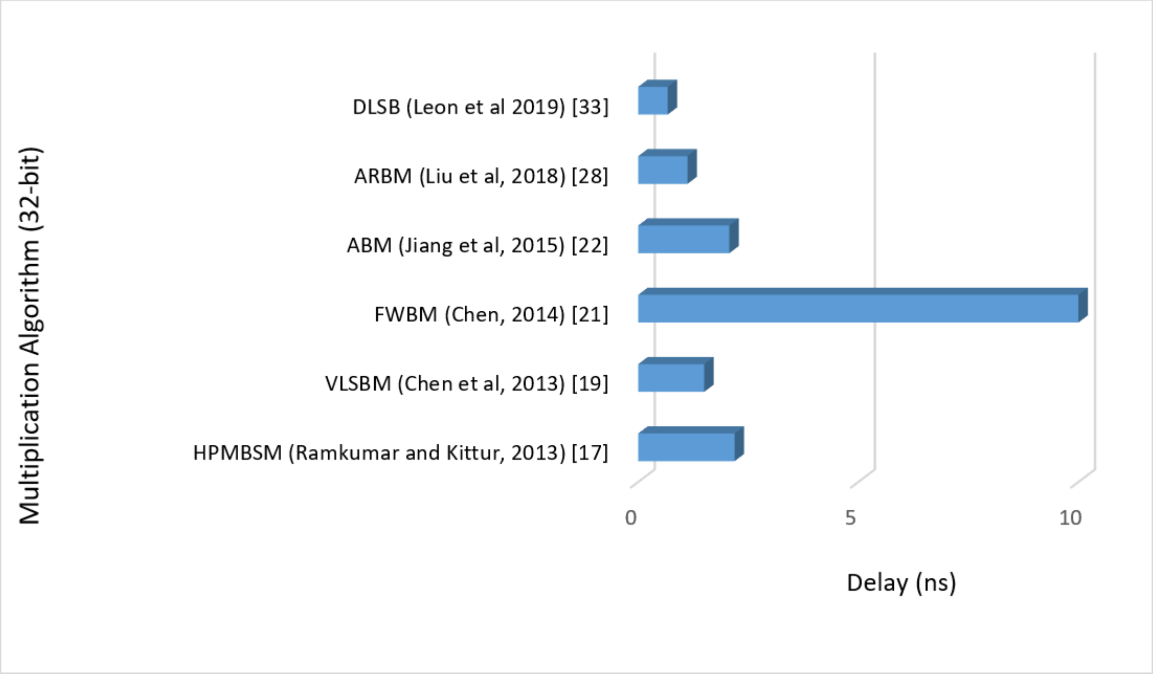

Figure 5. Delay characteristics of different 32-bit multiplication algorithms

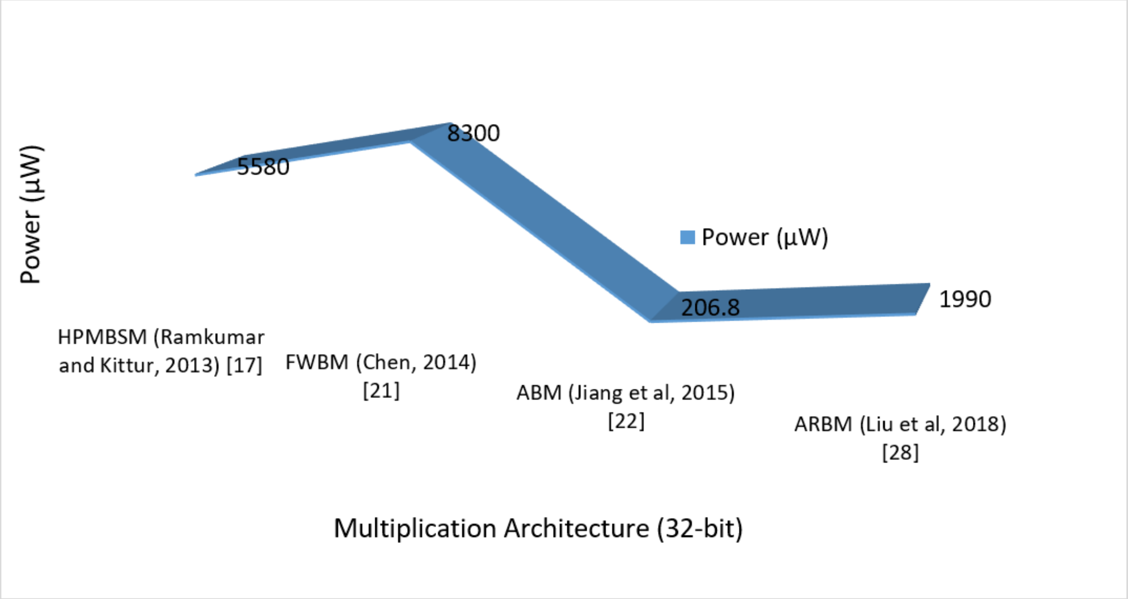

Figure 6. Power utilization by various 32-bit multiplication algorithms

The optimized 2's complementor outperforms the conventional 2's complementor by using additional number gates.

Similarly, Fig. 4, Fig. 5 and Fig. 6 illustrate that area usage, delay characteristics and power consumption of different 32-bit multiplier implementations, respectively. An investigation performed to identify an area efficient 32-bit multiplier (Figure 4) shows that an approximate Booth multiplier ABM [22] utilizes less are as compared with recently developed DLSB [33] and other previous multipliers. However, the very recently developed DLSB [33] utilizes moderate area by comparing with other existing methods. So, further analysis is performed to analyse the other circuit characteristics like delay and power consumption by the recently developed multipliers.

As shown in Fig. 5 the evaluation on characteristics of different 32-bit multipliers demonstrate that an area efficient multiplication algorithm DLSB[33] consumes very little time to provide the final product as compared with recent ARBM [28] and previously developed 32-bit multipliers. However, the recently developed ARBM [28] takes moderate and less time as compared with other existing multiplication scheme.

## 4. Conclusion

In this article, different kind of Booth multiplication algorithms and their applications are discussed and their VLSI characteristics such as area utilization, power consumption and speed of operations are investigated. The different kind of Booth algorithms have been examined such as configurable and fixed width Booth algorithm as well as exact and approximate Booth multiplication algorithms. Both fixed width and exact multiplication algorithms are used for various accurate operations like cryptosystems and other scientific arithmetic operations. The reconfigurable and approximate multiplication algorithms are widely used in image processing, multimedia applications and machine learning applications. Among the different radix encoding schemes, radix-4 is widely used due to its moderate- complex and high-speed of operation. Based on the examination, this investigation result suggests that the Radix4-based multiplication using binary to two's complement converter is suitable for high-performance with moderate power consumption. Further the power consumption can also be reduced by using adaptive binary to two's complement converter and by using low-complexity adder structure. Furthermore, the high-speed multimedia applications can be performed by high-performance Double- least significant bit two's complement multiplier. An area-efficient application can also be designed and be implemented by approximate computing scheme like approximate 2-bit addition for generating a triple multiplicand without carry propagation. Additionally, it recommends that STM 28nm CMOS technology for area-efficient as well as low-power applications and TSMC 45 nm technology for high-speed applications. Further research is being performed on designing and implementing low-power multiplication scheme for image processing and multimedia applications with different width of final product by reducing the complexity of binary to two's complement converter. To reduce complexity on Booth multiplication, it is suggested that resource redundancy technique and it is useful for developing an area-efficient Booth multiplication by using smaller number of components.

## References

[1] R. Shanmuganathan and K. Brindhadevi, "Comparative analysis of various types of multipliers for effective low power," *Microelectronic Engineering*, vol. 214, pp. 28–37, 2019.

[2] I. Qiqieh, R. Shafik, G. Tarawneh, D. Sokolov, S. Das, and A. Yakovlev, "Significance-driven logic compression for energy-efficient multiplier design," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 3, pp. 417–430, 2018.

[3] P. Sakellariou and V. Paliouras, "Application-specific low-power multipliers," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2973–2985, 2016.

[4] Y.-H. Chen, C.-Y. Li, and T.-Y. Chang, "Area-effective and power-efficient fixed-width booth multipliers using generalized probabilistic estimation bias," *IEEE Journal on Emerging and selected topics in Circuits and Systems*, vol. 1, no. 3, pp. 277–288, 2011.

[5] Q. Shao, Z. Hu, S. N. Basha, Z. Zhang, Z. Wu, C.-Y. Lee, and J. Xie, "Low complexity implementation of unified systolic multipliers for nist pentanomials and trinomials over $gf(2^m)$," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 8, pp. 2455–2465, 2018.

[6] M. S. Kim, A. A. Del Barrio, L. T. Oliveira, R. Hermida, and N. Bagherzadeh, "Efficient mitchell's approximate log multipliers for convolutional neural networks," *IEEE Transactions on Computers*, vol. 68, no. 5, pp. 660–675, 2018.

[7] J. Xie, P. K. Meher, X. Zhou, and C.-Y. Lee, "Low register-complexity systolic digit-serial multiplier over $gf(\hat{2}m)$ based on trinomials," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 773–783, 2018.

[8] C. W. Chiou, H. W. Chang, W.-Y. Liang, C.-Y. Lee, J.-M. Lin, and Y.-C. Yeh, "Low-complexity gaussian normal basis multiplier over gf (2m)," *IET Information Security*, vol. 6, no. 4, pp. 310–317, 2012.

[9] S. Malek, S. Abdallah, A. Chehab, I. H. Elhajj, and A. Kayssi, "Low-power and high-speed shift-based multiplier for error tolerant applications," *Microprocessors and Microsystems*, vol. 52, pp. 566–574, 2017.

[10] Z. Yuejun, D. Dailu, P. Zhao, W. Pengjun, and Y. Qiaoyan, "An ultra-low power multiplier using multi-valued adiabatic logic in 65 nm cmos process," *Microelectronics Journal*, vol. 78, pp. 26–34, 2018.

[11] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi, "Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 9, pp. 2856–2868, 2018.

[12] S.-R. Kuang and J.-P. Wang, "Design of power-efficient configurable booth multiplier," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 3, pp. 568–580, 2009.

[13] Y.-H. Seo and D.-W. Kim, “A new vlsi architecture of parallel multiplier–accumulator based on radix-2 modified booth algorithm,” *IEEE Transactions on very large scale integration (vlsi) systems*, vol. 18, no. 2, pp. 201–208, 2009.

[14] R. Muralidharan and C.-H. Chang, “Radix-8 booth encoded modulo 2{n}−1 multipliers with adaptive delay for high dynamic range residue number system,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 5, pp. 982–993, 2010.

[15] Y.-H. Chen, C.-Y. Li, and T.-Y. Chang, “Area-effective and power-efficient fixed-width booth multipliers using generalized probabilistic estimation bias,” *IEEE Journal on Emerging and selected topics in Circuits and Systems*, vol. 1, no. 3, pp. 277–288, 2011.

[16] Y.-H. Chen and T.-Y. Chang, “A high-accuracy adaptive conditional-probability estimator for fixed-width booth multipliers,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 3, pp. 594–603, 2011.

[17] B. Ramkumar and H. M. Kittur, “Faster and energy-efficient signed multipliers.” *VLSI Design*, 2013.

[18] R. Muralidharan and C.-H. Chang, “Radix-4 and radix-8 booth encoded multi-modulus multipliers,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 11, pp. 2940–2952, 2013.

[19] S.-K. Chen, C.-W. Liu, T.-Y. Wu, and A.-C. Tsai, “Design and implementation of high-speed and energy-efficient variable-latency speculating booth multiplier (vlsbm),” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 10, pp. 2631–2643, 2013.

[20] S. Choi, G. Kim, H.-J. Yoo, and B.-G. Nam, “Hybrid radix-4/-8 truncated multiplier for mobile gpu applications,” *Electronics Letters*, vol. 50, no. 23, pp. 1680–1682, 2014.

[21] Y.-H. Chen, “An accuracy-adjustment fixed-width booth multiplier based on multilevel conditional probability,” *IEEE transactions on very large scale integration (VLSI) systems*, vol. 23, no. 1, pp. 203–207, 2014.

[22] H. Jiang, J. Han, F. Qiao, and F. Lombardi, “Approximate radix-8 booth multipliers for low-power and high-performance operation,” *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2638–2644, 2015.

[23] G. Haridas and D. S. George, “Area efficient low power modified booth multiplier for fir filter,” *Procedia Technology*, vol. 24, pp. 1163–1169, 2016.

[24] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, “Design of approximate radix-4 booth multipliers for error-tolerant computing,” *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, 2017.

[25] S. M. Mirhosseini, A. S. Molahosseini, M. Hosseinzadeh, L. Sousa, and P. Martins, “A reduced-bias approach with a lightweight hard-multiple generator to design a radix-8 modulo 2{n}+1 multiplier,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, pp. 817–821, 2016.

[26] Z. Zhang and Y. He, “A low-error energy-efficient fixed-width booth multiplier with sign-digit-based conditional probability estimation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 2, pp. 236–240, 2017.

[27] P. A. Patil and C. Kulkarni, “Multiply accumulate unit using radix-4 booth encoding,” in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2018, pp. 1076–1080.

[28] W. Liu, T. Cao, P. Yin, Y. Zhu, C. Wang, E. E. Swartzlander, and F. Lombardi, “Design and analysis of approximate redundant binary multipliers,” *IEEE Transactions on Computers*, vol. 68, no. 6, pp. 804–819, 2018.

[29] H. Xue, R. Patel, N. Boppana, and S. Ren, “Low-power-delay-product radix-4 8* 8 booth multiplier in cmos,” *Electronics letters*, vol. 54, no. 6, pp. 344–346, 2018.

[30] D. J. Moss, D. Boland, and P. H. Leong, “A two-speed, radix-4, serial–parallel multiplier,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 769–777, 2018.

[31] A. A. Del Barrio, R. Hermida, and S. Ogrenci-Memik, “A combined arithmetic-high-level synthesis solution to deploy partial carry-save radix-8 booth multipliers in datapaths,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 742–755, 2018.

[32] S. Venkatachalam, E. Adams, H. J. Lee, and S.-B. Ko, “Design and analysis of area and power efficient approximate booth multipliers,” *IEEE Transactions on Computers*, vol. 68, no. 11, pp. 1697–1703, 2019.

[33] V. Leon, S. Xydis, D. Soudris, and K. Pekmestzi, “Energy-efficient vlsi implementation of multipliers with double lsb operands,” *IET Circuits, Devices & Systems*, vol. 13, no. 6, pp. 816–821, 2019.

**B. Hareesh** was graduated in Electrical and Electronics Engineering in 2016 from Swami Vivekananda Institute of Engineering and Technology, Jawaharlal Nehru Technological University Hyderabad, Telangana. Currently he is doing his post- graduation degree in Embedded Systems in the Department of Electronics and Communication Engineering of Sreyas Institute of Engineering and Technology, Hyderabad. His research interests are VLSI design and computer architecture.

**C. John Moses** was graduated in Electronics and Communication Engineering in 1996 from Manonmaniam Sundaranar University, Tirunelveli. He obtained his M.E. degree from Madurai Kamaraj University, Madurai in 1999, specializing in Applied Electronics. He obtained Ph. D. degree from Anna University, Chennai for his research work on “Some Studies on Realization of Image Interpolation Algorithms in FPGA” in 2017.

His area of specialization is Information and Communication Engineering. He has 2.9 years of industrial experience. He is in the teaching profession for the past nineteen years. Currently, he is working as Associate Professor of Electronics and Communication Engineering in Sreyas Institute of Engineering and Technology, Hyderabad. He has twenty international publications in Journals, more than thirty-five papers in International/National Conferences to his credit. He has acted as coordinator for many workshops, Seminars, International Conferences, and International Assignments like MoU with foreign universities. He visited various technical universities in Malaysia and organized student exposure programs in association with Universiti Sains Malaysia, Penang, Malaysia. He is a senior member of IEEE, Life Member of ISTE and Professional Member of IET and ACM. He received the Outstanding Branch Counselor 2013 award and IEEE MGM award 2019 from IEEE, USA.

**MVV Prasad Kantupudi** received his B. Tech (ECE) & M. Tech (Digital Electronics and Communication Systems) degrees from Jawaharlal Nehru Technological University, Kakinada. He received his Ph.D. from BITS, VTU, Belgum. Currently he is working as an Associate Professor in the Dept. Of E&TC, Symbiosis Institute of Technology, Pune. Previously he was worked as a Director of Advancements for Sreyas Institute of Engineering & Technology, Hyderabad, and as an Associate Professor with RK University, Rajkot having teaching experience around 10 years. His current research interests are in Signal Processing with Machine Learning, Education and Research. He is recognized as a technical resource person for Telangana state by IIT Bombay Spoken tutorial team and he conducted Key Training Workshops on Open-Source Tools for education, Signal Processing and Machine Learning focused topics, Educational Technology, etc. He has authored and co-authored many papers in International Journals, International Conferences and National Conferences