Indian Journal of Natural Sciences www.tnsroindia.org.in ©IJONS

Vol.14 / Issue 81 / Dec / 2023 *International Bimonthly (Print) – Open Access* *ISSN: 0976 – 0997*

RESEARCH ARTICLE

FPGA based Comparative Performance Analysis of 'n' Bit Multiplier

S.Bharathi[1]*, Y.Sudha[1] and G.Ravi Kishore[2]

[1]Assistant Professor, Department of EEE, Malla Reddy Engineering College, (Affiliated to Jawaharlal Nehru Technological University) Secunderabad, Hyderabad, Telangana, India.
[2]Assistant Professor, Department of ECE, Vidya Jyothi Institute of Technology, (Affiliated to Jawaharlal Nehru Technological University) Hyderabad, Telangana, India.

Received: 22 Aug 2023 Revised: 25 Sep 2023 Accepted: 28 Nov 2023

***Address for Correspondence**
S.Bharathi
Assistant Professor,
Department of EEE, Malla Reddy Engineering College,
(Affiliated to Jawaharlal Nehru Technological University)
Secunderabad, Hyderabad, Telangana, India.
Email: barati.reddy10@gmail.com

This is an Open Access Journal / article distributed under the terms of the **Creative Commons Attribution License** (CC BY-NC-ND 3.0) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. All rights reserved.

ABSTRACT

In this research, a brand-new adaptive multiplexer-based multiplication technique for an n-bit multiplier is proposed. It can be utilized in a variety of digital image processing applications, including those for processing satellite images, processing images for use in medicine. This proposed Field Programmable Gate Array (FPGA)based 'n'-bit multiplier uses a multiplexer and adder circuit. This reduces the power's area, latency, and leakage. In comparison to other current methods like the Constant Correction Truncation (CCT) scheme, Variable Correction Truncation (VCT) scheme, and Pseudo-carry Compensation Truncation (PCT) scheme, the suggested new adaptive multiplexer-based multiplication approach performs multiplication. In terms of area and delay, the suggested adaptive multiplexer-based multiplier performs up to 30% better than other existing multipliers. In comparison to other existing methods, the suggested method uses around 45% less dynamic power and generates a smaller amount of leakage power. The proposed system's average error value is 28% lower than that of the other systems already in use. Three devices from the XC3S1500-5fg320 FPGA family are used to implement the multiplier. The adaptive multiplexer-based multiplier now performs better all around.

Keywords: multiplier, multiplexer, FPGA, Area, Delay and leakage power

INTRODUCTION

A digital circuit using combinational logic called a "binary multiplier" is used to multiply two numbers. It frequently appears in a variety of contexts, including

- Digital image processing
- Computers
- High speed calculators
- Digital signal processing
- Mobiles …..etc.,

When compared to addition and subtraction, multiplication is a complex operation. It requires the multiplicand and multiplier, two numbers. Similar to how you multiply decimal values, binary multiplication works. Because binary multiplication simply uses binary values (0 and 1), computation of such operations is simpler. The following equations illustrate the binary multiplication rules.

$$0 \times 0 = 0 \quad (1)$$
$$0 \times 1 = 0 \quad (2)$$
$$1 \times 0 = 0 \quad (3)$$
$$1 \times 1 = 1 \quad (4)$$

Let us consider the example of 4 bit numbers as shown in figure (1). The multiplier value is 1011 and multiplicand value is 1010. This type of multiplication generates partial products depending on the multiplier's bit count. The value of all partial products is added to determine the final product value. Combinational multiplier or array multiplier is the name given to the process in which four shifts and additions are employed to multiply four-bit binary numbers. To improving the performance of combinational multiplier circuit to introduces the Parallel Binary Multiplier Circuit (PBMC). Let us consider the example of two bit PBMC is shown in figure (2)

Here the multiplicand is A1 A0 and multiplier is B1 B0
The partial products are evaluated by multiplication of the each bit of the multiplicand and multiplier. PBMC method was implemented using combinational logic circuits. AND gate was used to perform multiplication operation and adder circuit was used to perform addition operation. Similarly 4 bit PBMC example is shown in figure (3)

Here the multiplicand is A3 A2 A1 A0 and multiplier is B3 B2 B1 B0
The implementation of 4 bit PBCM by using combinational logic circuit as shown in figure (4)

RELATED WORK

A method for quick augmentation with a basic computerised circuit was illustrated by Lobby et al. The calculation entails registering erroneous two-fold logarithms, adding or subtracting the logarithms, and calculating the anticipated antilogarithm of the result. Even though the calculation uses an advanced channel, processing a single item is not any simpler than with exhibit multipliers. In order to increase the accuracy of a logarithmic multiplier that relies on Mitchell's computations for computing logarithms and antilogarithms, McLaren presented a method. The proposed approach only relies on the paired division segments of two multiplicands, which provides space and force sparing, to make the error in the output. Patricio Bulic et al., in order to achieve discretionary exactness through an iterative process, Patricio Bulic et al. devised an iterative logarithmic multiplier and examined several multipliers in a logarithmic number framework. The error remedy was applied in accordance with the necessary augmentation. This strategy's limitation is the combinational postponement that increases with each new remedy circuit.

A method based on blended whole number straight programming was presented by Davide De Caro and colleagues to achieve perfect co-productive qualities that reduce relative estimate error while using fewer nonzero bits for the co-proficient. By making a few moves, the equipment usage recognises the duplicate and grows, avoiding the use of all multipliers. Complex equipment for low-accuracy use is the result of this method. Abed and Sifred came up with the idea for and created a 32-piece, original double to parallel logarithm converter CMOS. The converter is constructed using a pure combinational logic plan and determines approximated logarithm in a single clock cycle.

The logarithmic converter's speedy Leading One Detector (LOD) circuit and modified logarithmic shifter operate quickly and efficiently. Because of logarithmic approximations, the errors are known. To study the complexity of the computation, Kowada et al. devised reversible circuits for Karastuba's calculation. The various methods of disposing of waste were discussed and compared to Bennet's plans. This circuit can be used in reversible computers, which have the advantage of being very energy-efficient. CPUs and GPUs were examined by Bryson R. Payne et al. The analysis shows that picture handling convolution functions more effectively than CPU. In order to deal with giving the advanced Rivest-Shamir-Adleman (RSA) processors dependent on high-radix Montgomery multipliers fulfilling various requirements, such as circuit region, working time, and resistance against side channel attacks, Atsushi Miyamoto et al. proposed a fundamental configuration method. The suggested method combines three information way structures using various middle-of-the-road structures to provide improved information ways matching the requirements. The multiplier is not the best choice for calculations involving cryptography.

Effective FPGA use of bit-equal blended Karatsuba-Of man multipliers (KOM) and Galois Field GF were introduced by Pack Zhou et al (2m). As compared to previous Application Specific Integrated Circuit (ASIC) dialogues, the common statement sharing and the multidimensional nature investigation on odd-term polynomials were familiarised with to achieve a lower entryway bound. Utilizing four or six information query tables on FPGAs prolonged the test. A useful calculation for registering decimal logarithms using 64-piece coasting point number-crunching was proposed by Ramin Tajallipour and colleagues. The calculation relies on digit-by-digit iterative calculation, which excludes the use of bend fitting, query tables, decimal to twofold transformation, and division operations. A low-normal-mistake adaptive PCT plot was proposed by Chip-Hong Chang and Ravi Kumar Satzoda and compares to other truncation algorithms.

The suggested method achieves lower normal. Equipment construction and FPGA-based equal engineering for conventional and shorter multipliers were introduced by Muhammad Rais et al. Comparing the suggested multiplier to conventional multipliers, it takes up less space.

The product of two n bit number is given by

$$P = A\,B \qquad (5)$$

Where A = an-1 an-2 ……..a1 a0 is multiplicand and B = bn-1 bn-2 ……. b1 b0

For multiplexer based multiplier An-1 = an-2 an-3 an-4 ……..a0 and Bn-1 = bn-2 bn-3 bn-4 ……..b0, the product value is given by

$$P = \{An-1 + 2^{n-1}an-1\}\{Bn-1 + 2^{n-1}bn-1\} \qquad (6)$$

To reduce this equation we get

$$P = \sum_{i=0}^{n-1}(ai\,.\,bi)2^{2i} + \sum_{i=1}^{n-1} Mi\,2^{2i} \qquad (7)$$

Where Mi = ai Bi + Ai bi, Mi can be used to implement the multiplexers with select lines are ai and bi.
The adaptive compensation circuit and fixed biassed technique are proposed to reduce the truncation error. By keeping the multiplexer's most important columns, it is possible to decrease the hardware. The result of the n-bit truncation is given by

$$Pt = \sum_{i=r}^{n-1} Mi\,2^{i-n} + \sum_{i=\left[\frac{r+1}{2}\right]}^{r-1} Mti\;2^{r-n} + \sum_{i=\left[\frac{r+1}{2}\right]}^{n-1}(ai\,.\,bi)\,2^{2i-n} + \sum_{i=0}^{r-1}(ai\,.\,bi)\,2^{r-n} + 2^{-1} \qquad (8)$$

Where r= n-k
Mti = ai . Bti + Ati . bi
Bti = bi-1 bi-2 bi-3 ……br-i
Ati = ai-1 ai-2 ai-3 ……ar-i

For a 8:1 multiplexer, however the output z depend on select signals s1, s2 &s3 and inputs i1, i2, i3, i4, i5, i6, i7 & i8 is given by

$z = \overline{s1}\overline{s2}\overline{s3}i1 + \overline{s1}\overline{s2}s3i2 + \overline{s1}s2\overline{s3}i3 + \overline{s1}s2s3i4 + s1\overline{s2}\overline{s3}i5 + s1\overline{s2}s3i6 + s1s2\overline{s3}i7 + s1s2s3i8$ (9)

The probability of z may be 1 or o therefore Pz = 0.5.

The multiplexer based array multiplication product value is given by

$P = \sum_{i=0}^{n-1}(ai \,.\, bi)\, 2^{2i} + \sum_{i=1}^{n-2} Mi\ 2^{i} - Mn - 1\, 2^{n-1}$ (10)

Reduction error

Elimination of the reduction error based on the selection of multiplexer for partial product evaluation. The number of multiplexers used for the implementation also reduced.

$$E\,(mux) = \left\lceil \frac{i}{2} \right\rceil Pz\, 2^{i-n} \quad (11)$$

$$Ered = \begin{cases} 2^{-n} P0\,, \ i = 0 \\ \sum_{i=2}^{n-k-1} \left(\left\lceil \frac{i}{2} \right\rceil Pz \left(\frac{n}{2} \right) + Pzn + P0 \right), \ \forall\ even\ i \\ \sum_{i=1}^{n-k-1} \left\lceil \frac{i}{2} \right\rceil Pz \left(\frac{n}{2} \right) 2^{i-n}\,, \ \forall\ odd\ i \end{cases} \quad (12)$$

Rounding error

The value of P0 initially 0.25, if i increase it quickly changed as 0.5 therefore the expected value of error is given by

$Eround = 0.5 \sum_{i=n-k}^{n-1} 2^{i-n}$ (13)

Total error value for the before compensation is given by

$Etot = -(Eround + Ered)$ (14)

The expected value of the compensation is given by

$C = 0.5 \left\lceil \frac{n-k}{2} \right\rceil Pc\, 2^{-k}$ (15)

The average error value is given by

$Eavg = Etot + C$ (16)

RESULT AND DISCUSSION

Our proposed method compared with the existing methods such as PCT, VCT and CCT. We consider the following parameters for discussion

- Average error value
- Delay
- Area
- Dynamic power
- Leakage power

The parameter practical values are listed in the following table and corresponding graph is plotted as shown in the following figure.

CONCLUSION

The efficiency of the proposed adaptive multiplexer based multiplier can be improved tremendously. In this proposed method with low mean square error value when compared with the other existing methods. The area of utilization is directly propositional to the number of bits, if n is small the area of utilization also less. The delay in nano seconds, compared to the other existing methods the proposed method provide 30% better results. The leakage

power in nano Watts, the proposed method reduces tremendously. It can be used in various applications like digital signal processing, digital image processing, biometric applications and medical image processing.

REFERENCES

1. U. M. Baese, *Digital Signal Processing With Field Programmable Gate Arrays*. Berlin, Germany: Springer, 2004.
2. V. Mahalingam and N. Ranganathan, "An efficient and accurate log- arithmic multiplier based on operand decomposition," in *Proc. VLSI Des.*, Jan. 2006, p. 6.
3. N. Yoshida, E. Goto, and S. Ichikawa, "Pseudorandom rounding for truncated multipliers," *IEEE Trans. Computers*, vol. 40, no. 9, pp. 1065–1067, Sep. 1991.
4. J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of low-error fixed width multipliers for DSP applications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 6, pp. 836–842, Jun.99.
5. M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," in *Proc. IEEE Workshop VLSI Signal Process. VI*, Oct. 1993, pp. 388–396.
6. E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel multipliers," in *Proc. 31st Asilomar Conf. Signals, Syst. Comput.*, Nov. 1997, vol. 2, pp. 1178–1182.
7. M. J. Schulte, J. E. Stine, and J. G. Jansen, "Reduced power dissipa- tion through truncated multiplication," in *Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Des.*, Mar. 1999, pp. 61–69.
8. L. -D. Van and C. -C. Yang, "Generalized low-error area-efficient fixed- width multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.
9. Y. C. Lim, "Single-precision multiplier with reduced circuit com- plexity for signal processing applications," *IEEE Trans. Comput.*, vol. 41, no. 10, pp. 1333–1336, Oct. 1992.
10. C. H. Chang, R. K. Satzoda, and S. Sekar, "A novel multiplexer based truncated array multiplier," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Kobe, Japan, May 2005, pp. 85–88.
11. K. Z. Pekmestzi, "Multiplexer-based array multipliers," *IEEE Trans. Comput.*, vol. 48, no. 1, pp. 15–23, Jan. 1999.
12. T. B. Juang and S. F. Hsiao, "Low-error carry-free fixed-width multi- pliers with low-cost compensation circuits," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 6, pp. 299–303, Jun. 2005.
13. K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified Booth multiplier," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 5, pp. 522–531, May 2004.
14. R. K. Satzoda, C. H. Chang, and T. Srikanthan, "Monte Carlo statistical analysis for power simulation in Synopsys design compiler," presented at the Synopsys Users' Group Conf., Singapore, Jun. 2006 [Online].
15. R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A Monte Carlo ap- proach for power estimation," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 1, no. 1, pp. 63–71, Mar. 1993.
16. D. Salomon, *Data Compression—The Complete Reference*, 2nd ed. New York: Springer-Verlag, 2000.
17. Patricio Bulic, Zdenka Babic and Aleksej Avramovic, "A Simple Pipelined Logarithmic Multiplier," IEEE Conference on Computer Design, pp. 230-240, December 2010.
18. E L Hall, D D Lynch and S J Dwyer III. "Generation of Products and Quotients using Approximate Binary Logarithms for Digital Filtering Applications," IEEE Transactions on Computers, Vol. C-19, No. 2, pp. 97-105, February 1970.
19. K H Abed and R E Sifred, "CMOS VLSI Implementation of a Low-Power Logarithmic Converter," IEEE Transactions on Computers, vol. 52, no. 11, pp. 1421-1433, November 2003.
20. K H Abed and R E Sifred, "VLSI Implementation of a Low-Power Antilogarithmic Converter," IEEE Transactions on Computers, Vol. 52, No. 9, pp. 1221-1228, September 2003.

Table 1 Comparison of average error value

	n=8		n=12		n=16		n=32		n=64		n=128	
	K=2	K=3	K=2	K=3	K=2	K=3	K=2	K=3	K=2	K=3	K=2	K=3
PM	0.0412	0.0214	0.0341	0.0212	0.0426	0.0125	0.0486	0.0115	0.0526	0.0196	0.0572	0.0218
PCT	0.0576	0.0476	0.0180	0.0157	0.0623	0.0155	0.0612	0.0132	0.0721	0.0432	0.0762	0.0542
VCT	0.0615	0.0313	0.0625	0.0313	0.0621	0.0312	0.0682	0.0691	0.0872	0.0781	0.0896	0.0871
CCT	0.0595	0.0605	0.0625	0.0621	0.0631	0.0623	0.0782	0.0796	0.0962	0.0896	0.0986	0.0924

Table 2 Comparison of Delay (ns)

	n=16		n=32		n=64		n=128	
	K=2	K=3	K=2	K=3	K=2	K=3	K=2	K=3
PM	3.05	3.08	4.68	4.92	5.68	5.75	7.21	7.23
PCT	4.04	4.03	7.21	7.98	8.63	8.61	13.21	14.62
VCT	4.94	5.11	9.69	9.67	12.86	12.95	18.65	19.71

Table 3 Comparison of Area (µm²)

	n=16		n=32		n=64		n=128	
	K=2	K=3	K=2	K=3	K=2	K=3	K=2	K=3
PM	19624	19867	41826	42678	72126	73167	102787	120427
PCT	21641	23584	71887	74468	112846	113284	168267	169827
VCT	27249	27971	96618	103204	142681	152863	196826	204862

Table 4 Comparison of Dynamic power (µW)

	n=16		n=32		n=64		n=128	
	K=2	K=3	K=2	K=3	K=2	K=3	K=2	K=3
PM	172	196	621	657	732	754	976	982
PCT	161	180	545	592	635	671	696	699
VCT	157	176	540	586	623	652	681	683

Table 5 Comparison of Leakage power (nW)

	n=16		n=32		n=64		n=128	
	K=2	K=3	K=2	K=3	K=2	K=3	K=2	K=3
PM	94	98	212	224	387	397	420	476
PCT	102	113	324	351	556	572	720	738
VCT	113	141	448	486	682	697	867	892

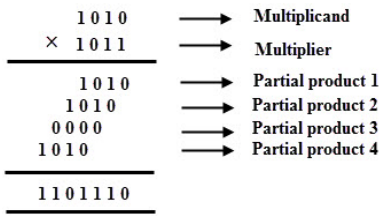


Figure 1 Example for 4 bit multiplication

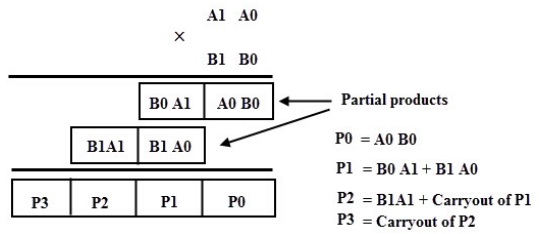


Figure 2 Two bit PBMC

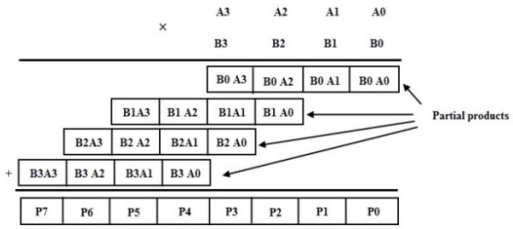


Figure 3 Four bit PBMC

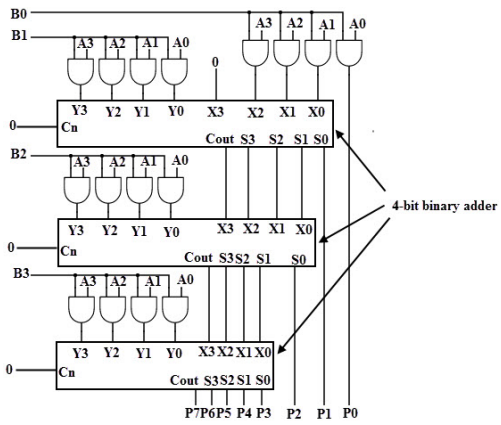


Figure 4 Four bit PBMC by using combinational logic circuit

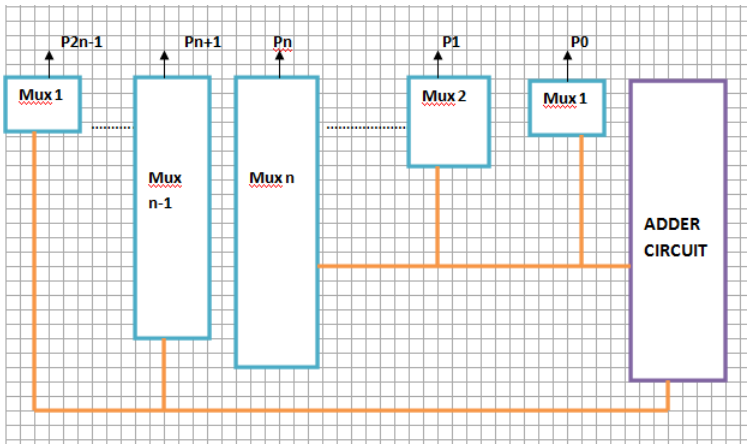


Figure 5 Adaptive Multiplexer based n bit Multiplier (AMM)

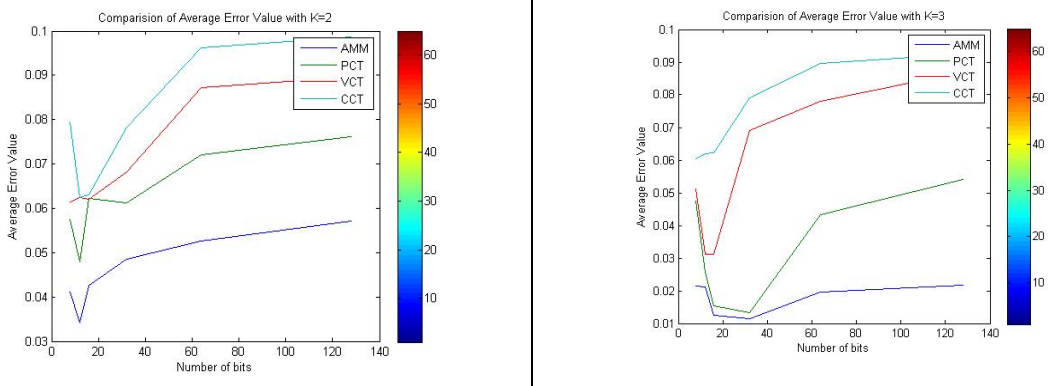
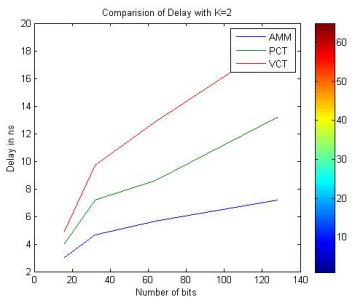


Figure 6 Comparison of average error value



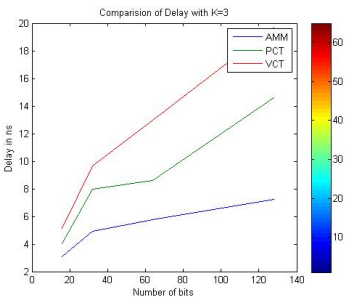
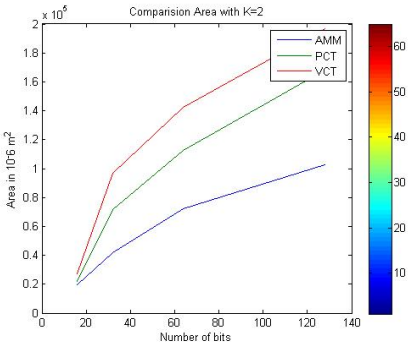


Figure 7 Comparison of Delay (ns)



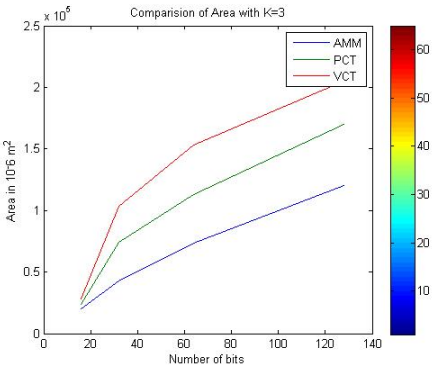
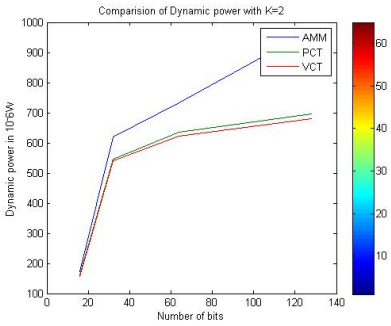


Figure 8 Comparison of Area (µm²)



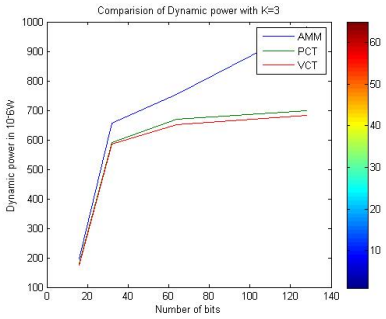
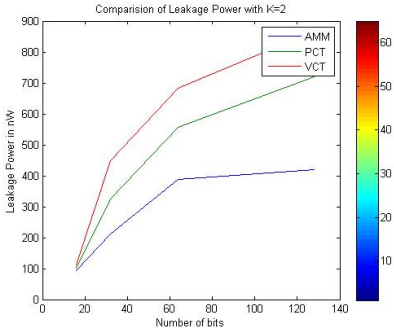


Figure 9 Comparison of Dynamic power (µW)



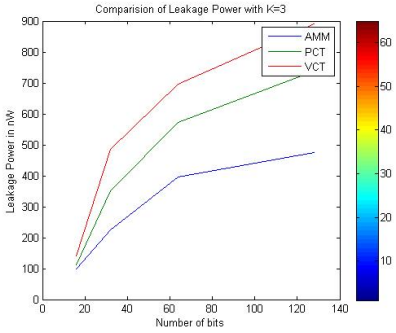


Figure 10 Comparison of Leakage power (nW)