Grenze International Journal of Engineering and Technology, Jan 2019

A 32-Bit Altered Partial Product Multiplier for Low-Power and Low-Area Applications

S. Harsha Sree[1] and S. Hanumantha Rao[2]
[1]M. Tech-VLSID Student, Department of Electronics and Communications Engineering, Shri Vishnu Engineering College for Women(Autonomous), Bhimavaram, India
Email:sagiharshasree19@gmail.com
[2]Professor, Department of Electronics and Communications Engineering, Shri Vishnu Engineering College for Women(Autonomous), Bhimavaram, India
Email:itshrao@gmail.com

***Abstract*—The concept of approximate computing gives scope for applications in various areas. Signal processing and image processing applications doesn't require exact arithmetic designs. The outputs of their huge computational units provide error tolerance where approximate computational blocks can be made use of. Computational blocks include inexact adders and multipliers at the basic level. Approximate computing develops simplified units saving complexity and power consumption compared with the exact designs. An advanced multiplier design is introduced in the paper which makes use of the concept of altered partial products using two different signals propagate and generate to replace the original partial products. Two sorts of 32-bit multipliers are explored in the paper. The 32-bit multipliers implementation is done in Verilog, simulation is performed using the Cadence Native Compiler and synthesis is accomplished using Cadence Encounter making use of 45nm technology.**

***Index Terms*— approximate computing, arithmetic designs, inexact multipliers, partial products and Cadence Encounter.**

I. INTRODUCTION

Many of the computer based arithmetic circuits inculcate digital logic designs for gaining reliability at the top. Most of the applications based on multimedia and image processing always allow imprecise and inaccurate values producing useful results. At each time, exact arithmetic blocks doesn't succeed in providing efficient designs. To develop best designs for these circuits, inexact units can be employed. Multipliers and adders stand as the building blocks for the computer arithmetic designs. So if approximation is developed using these blocks, efficient designs can be presented.

Full adders are highly used for the addition operations in approximate computing. The full adder designs are explored for approximate designs [1-3]. These designs discuss about different approximate adder blocks. For the design of approximate multipliers different truncation schemes are discussed in [4-6]. A group of partial products are truncated in these designs to generate approximate multipliers.

In the works of [7], two different types of segment based multipliers were discussed. In this, Static Segment Multiplier (SSM) is studied. In SSM, instead of multiplying two p-bit multipliers, a segment of them is

Grenze ID: 01.GIJET.5.1.508_1

chosen and multiplication is performed. Segment length can be varied. If the multiplier is 16-bit one, length of the segment can be any number less than 16. The obtained result can be extended to 32 bit by appending zeros at their respective positions. The segments are chosen based on the leading "1" bit position. If the MSB contains leading "1" bit the segment from it is chosen else it will be from LSB. From [8], inexact 2*2 multiplier designs are generated. This multiplier uses only 3-bits to represent the result. For example the result of 9 (1001) uses four bits but it represented as 7(111). This multiplier saves 1-bit position saving the number of adders used. Using these inexact 2*2 blocks higher order multipliers such as 4*4, 8*8 and 16*16 even higher orders can be designed. The multiplier discussed in [9] is Partial Product Perforation (PPP) multiplier. In this design some of the partial products are perforated. The elimination is done in a row wise manner. Since some of the partial products are eliminated, the number of adders used gets reduced providing approximation. In the discussion of [10] two types of approximate 4-2 compressors were discussed. These compressors offer less area and power when compared with the exact one.

II. Altered Partial Product Multiplier

A. Transformation of Partial Products

Generally multipliers are solved in three basic steps. They are partial product generation, reduction of partial products and adding final rows of partial products. Of these stages, partial product reduction occupies more area and power. So any approximation is concentrated on this stage. Approximation includes basic blocks such as adders and compressors. Compressors are used instead of using more number of full adders. Different compressor based designs are discussed in the works of [11-14]. In this paper, approximate half adder, full adder and 4-2 compressor are used to solve the multipliers. Basically, any arithmetic block is directly applied to the partial products but here first the partial products are transformed to new terms and are applied. Two signals termed as propagate (p) and generate (g) are used for the transformation. Fig. 1 shows the block diagram for the transformation of generated partial products to changed partial products whereas Fig. 2 shows the same for an 8-bit multiplier. The $a_{m,n}$ term indicates the generated partial products. Most of them are transformed to two new terms $p_{m,n}$ and $g_{m,n}$. The term $p_{m,n}$ is obtained using an OR gate and $g_{m,n}$ using AND gate clearly shown in Fig.1. By applying approximation to newly obtained products instead of the original partial products provides huge area and power savings.

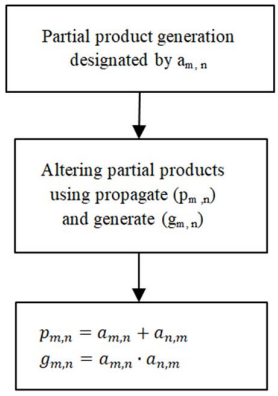

Fig 1. Block Diagram for Altering Partial Products

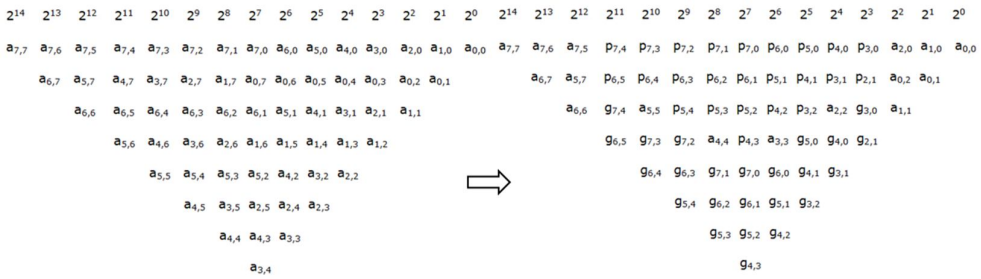

Fig 2. Transformation of Original Partial Products to New Altered Partial Products

B. Inexact Arithmetic Blocks

Compressors and adders form the basic blocks for the multiplier reduction process. This section presents approximate half adder, full adder and 4-2 compressor designs. Fig. 3 and 4 shows the block diagrams of inexact half adder and full adder. Half adder shown in Fig. 3 replaces XOR gate used in exact one using OR gate saving the delay caused by XOR gate. Full adder in Fig. 4 uses only three gates instead of five gates used in exact adder saving area and power. Fig. 5 shows the inexact 4-2 compressor design. This design has only four inputs and two outputs whereas the exact design has five inputs and three outputs.

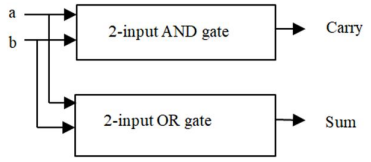

Fig 3. Block Diagram of Inexact Half Adder

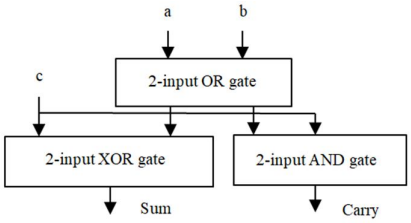

Fig 4. Block Diagram of Inexact Full Adder

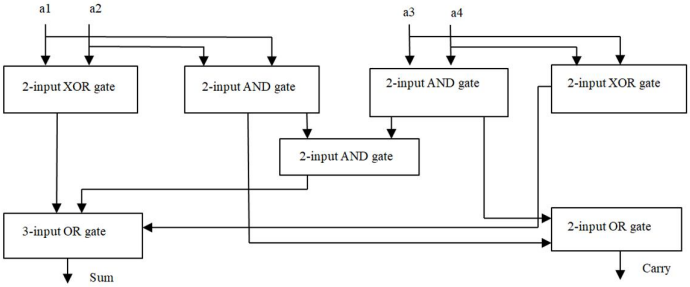

Fig 5. Block diagram of Inexact 4-2 Compressor

C. 16-Bit Altered Partial Product Multiplier

Fig. 6 and 7 show the levels 1, 2, 3 and 4 of 16-bit Multiplier ML architecture.

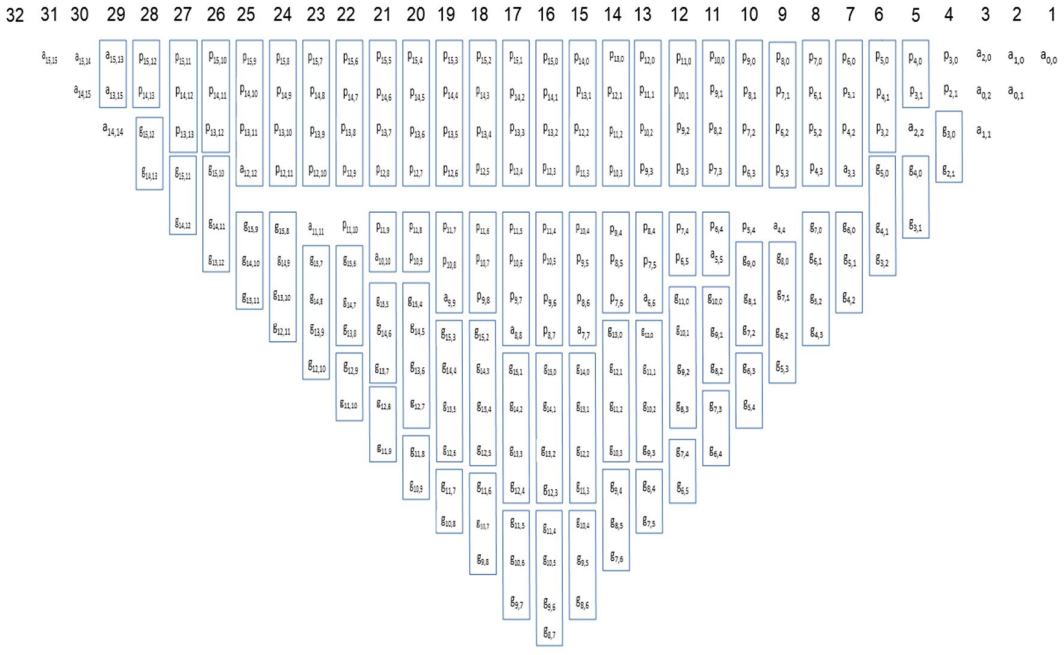

Fig 6. Stage 1 of Multiplier ML

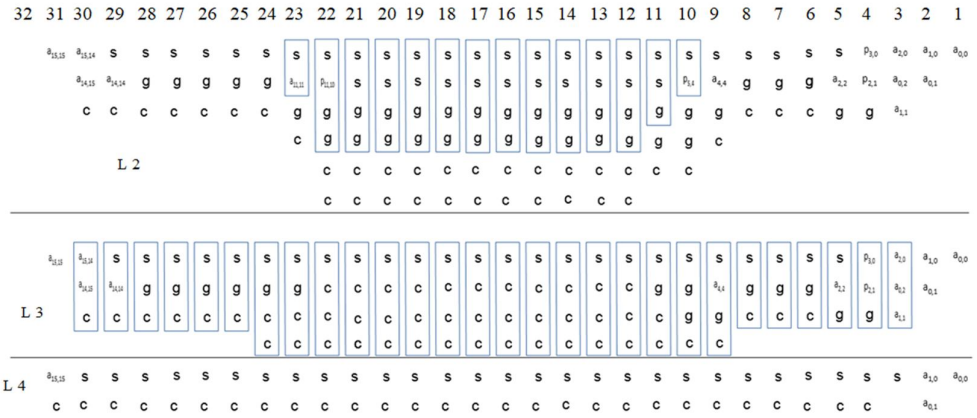

Fig 7. Stages 2, 3 and 4 of the Multiplier ML

Fig. 6 shows the level 1 architecture of 16-bit Multiplier ML. All the partial products from column 4 to column 28 are altered using the propagate and generate signals designated as p and g. All the propagate signals are grouped to form half adders, full adders and 4-2 compressors. The generate signals are solved using the OR gates. All the columns are grouped using the approximate arithmetic units. In this implementation Fig. 3, 4 and 5 showing approximate half adder, full adder and 4-2 compressor are used to group the partial products. Level 1 is solved to form level 2. The s and c represent the sum and carry signals. The g signal in level 2 represent the output of OR gates from level 1. The final two rows of the level 4 are given to the ripple carry adder to from the 32 bit output.

D. 32-Bit Altered Partial Product Multiplier

Fig. 8 and Fig. 9 show the block diagrams of 32-bit Multiplier ML and Multiplier L.

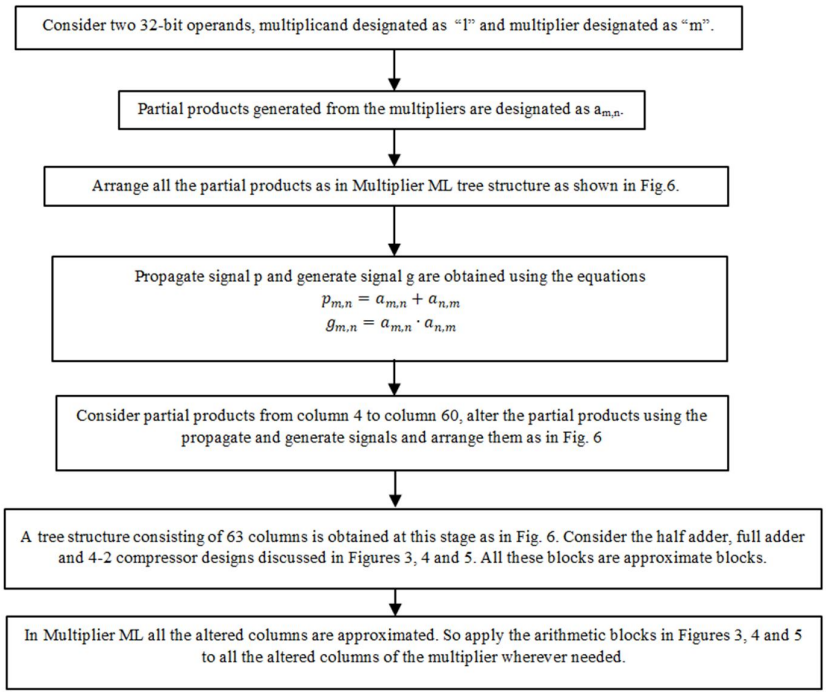

Fig 8. Block Diagram of 32-bit Multiplier ML

The 32-bit multipliers implementation flow is shown Fig. 8 and 9. Fig. 8 shows the 32-bit Multiplier ML block implementation. In this multiplier, approximate arithmetic units are applied to all the columns of the

multiplier. Fig. 9 shows the 32-bit Multiplier L block implementation. In this implementation the approximate arithmetic blocks are applied only to the least significant 31 columns of the multiplier.
Consider Fig. 8, the ML in the Multiplier ML relates to both the most significant and least significant parts of the multiplier. So approximation is applied to all the columns whereas in Fig. 9, the L in Multiplier L relates to the least significant columns of the multiplier. So approximation is applied only to the least significant columns.
Firstly, consider 32-bit multiplicand and 32-bit multiplier. The result obtained is of 64 bit. In normal multipliers approximation is applied directly to all the partial products of the multiplier. Here in this concept, the original partial products are transformed to altered partial products using the propagate and generate signals. To these altered partial products approximation is applied. In the multiplier designs, the propagate signals are grouped by using half-adder, full adder and 4-2 compressors and generate signals are grouped by using OR gates. In the design of Multiplier ML all the half adders, full adders and 4-2 compressors used were approximate blocks. In the design of Multiplier L, to group the least significant 31 columns approximate half adder, full adder and 4-2 compressor are used and for the remaining most significant columns exact units are used.

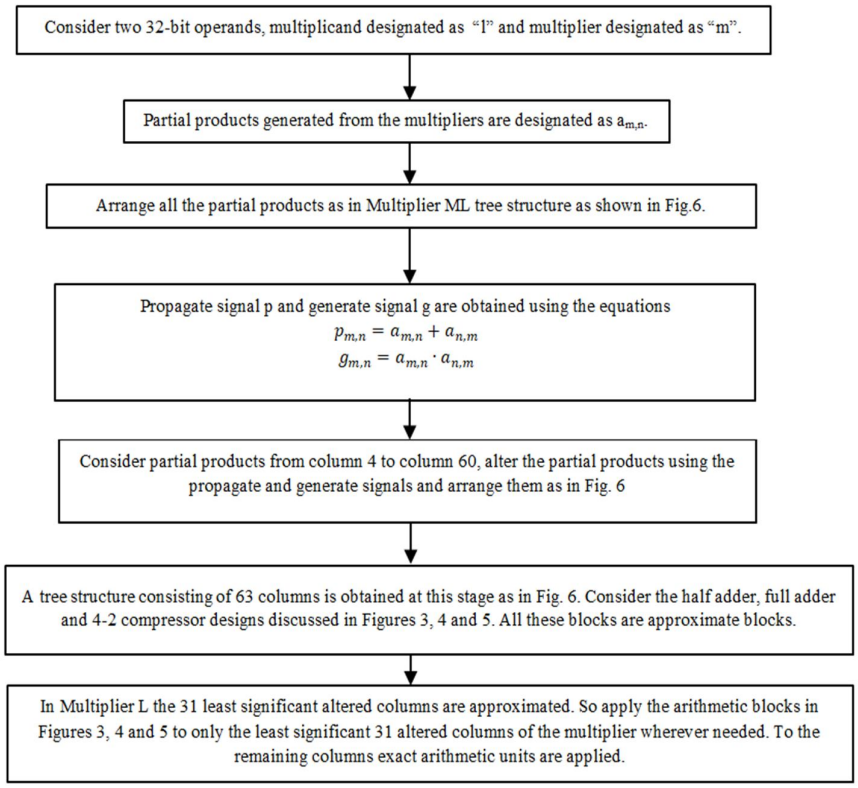

Fig 9. Block Diagram of 32-bit Multiplier L

III. Results and Discussion

Fig 10 shows the simulation result for the 32-bit Multiplier ML and Fig 11 shows the simulation results for the 32-bit Multiplier L.

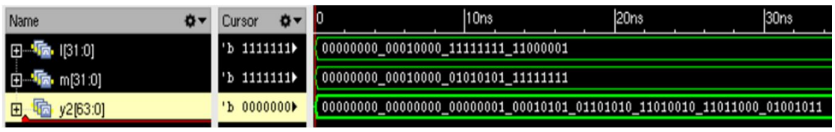

Fig 10. Simulation results for 32-bit Multiplier ML

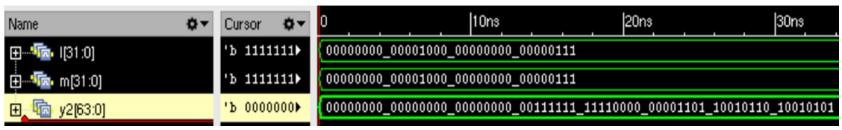

Fig 11. Simulation results for 32-bit Multiplier L

Fig. 12 and Fig. 13 show the area results for the 32-bit Multiplier ML and 32-bit Multiplier L in 45nm technology. Fig. 14 and Fig.15 show the power results of 32-bit Multiplier ML and 32-bit Multiplier L in 45nm technology.

```
  Generated by:           Encounter(R) RTL Compiler RC14.25
  Generated on:           Sep 27 2018  03:46:17 am
  Module:                 MULTIPLIER1BIT32
  Technology libraries:   slow
                          physical_cells
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
```

Instance	Cells	Cell Area	Net Area	Total Area
MULTIPLIER1BIT32	2594	5941	4780	10722
cmp99	5	11	4	15
cmp98	5	11	4	15
cmp97	5	11	4	15
cmp96	5	11	4	15
cmp95	5	11	4	15

Fig 12. Area results for 32-bit Multiplier ML in 45nm Technology

```
  Generated by:           Encounter(R) RTL Compiler RC14.25
  Generated on:           Sep 27 2018  03:49:42 am
  Module:                 MULTIPLIER2BIT32
  Technology libraries:   slow
                          physical_cells
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
```

Instance	Cells	Cell Area	Net Area	Total Area
MULTIPLIER2BIT32	2707	6432	4916	11348
cmpe99	6	14	4	18
cmpe98	6	14	4	18
cmpe97	6	14	4	18
cmpe96	6	14	4	18
cmpe95	6	14	4	18
cmpe94	6	14	4	18
cmpe93	6	14	4	18

Fig 13. Area results for 32-bit Multiplier L in 45nm Technology

```
  Generated by:           Encounter(R) RTL Compiler RC14.25
  Generated on:           Sep 27 2018  03:46:16 am
  Module:                 MULTIPLIER1BIT32
  Technology libraries:   slow
                          physical_cells
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
```

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
MULTIPLIER1BIT32	2594	258.904	1972554.318	1972813.222
cmp1	5	0.507	1310.944	1311.450
cmp10	5	0.507	1598.965	1599.472
cmp100	5	0.507	1685.392	1685.899
cmp101	5	0.507	1924.001	1924.507
cmp102	5	0.507	1751.289	1751.796
cmp106	5	0.507	1428.809	1429.316

Fig 14. Power results for 32-bit Multiplier ML in 45nm Technology

```
  Generated by:           Encounter(R) RTL Compiler RC14.25
  Generated on:           Sep 27 2018  03:49:42 am
  Module:                 MULTIPLIER2BIT32
  Technology libraries:   slow
                          physical_cells
  Operating conditions:   slow
  Interconnect mode:      global
  Area mode:              physical library
```

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
MULTIPLIER2BIT32	2707	284.459	3614068.949	3614353.407
cmpe10	6	0.692	4373.487	4374.179
cmpe100	6	0.692	8537.432	8538.124
cmpe101	6	0.692	5386.310	5387.002
cmpe102	6	0.692	10304.515	10305.207
cmpe103	6	0.692	9892.096	9892.788

Fig 15. Power results for 32-bit Multiplier L in 45nm Technology

Table I shows the synthesis results of different 16-bit multipliers in terms of area and power. All these results are compared in 180nm and 45nm technologies. Multiplier ML and Multiplier L show huge reduction in area occupied and power consumed when compared with the different multiplier implementations.

Exact Dadda Multiplier is implemented using exact 4-2 compressors. In CM1, approximation is applied to all columns of the multiplier whereas in CM2 approximation is applied to the least significant columns. SSM stands for the static segment multiplier design. In this paper a 12-bit segment is chosen to design this concept. The 24-bit result is extended to 32-bit by appending zeros at the appropriate positions. PPP stands for the partial product perforation multiplier. In this paper third and fourth rows of the partial products are eliminated and to the remaining partial products approximation is applied. UDM stands for the under designed multiplier.

Table II shows the synthesis results for the proposed 32-bit multiplier designs. Comparisons are made between Multiplier ML, Multiplier L and the exact 32-bit Dadda multipliers. All the multipliers are designed for 180nm and 45nm technologies. 32-bit Multiplier ML consumes less area and power when compared with the other multipliers. Multiplier L consumes less area and power compared with the 32-bit exact Dadda Multiplier.

TABLE I. SYNTHESIS RESULTS OF 16-BIT DIFFERENT MULTIPLIERS AND ALTERED PARTIAL PRODUCT MULTIPLIERS (MULTIPLIER ML, MULTIPLIER L)

Multiplier type	180nm Technology		45nm Technology	
	Area (μm^2)	Power (pW)	Area (μm^2)	Power (pW)
Exact Dadda Multiplier	23371	6258	3604	1659
Multiplier ML	14403	2366	2951	894
Multiplier L	16639	4347	3081	1323
Compressor based Multiplier 1(CM1)	16096	3226	2781	1035
Compressor based Multiplier 2 (CM2)	19972	5161	3262	1431
Static Segment Multiplier(SSM)	13435	4887	3921	1324
Partial Product Perforation Multiplier(PPP)	20411	5552	3162	1528
Under Designed Multiplier(UDM)	17101	5805	2797	1548

TABLE II. SYNTHESIS RESULTS OF 32-BIT MULTIPLIERS FOR 180NM AND 45NM TECHNOLOGIES

Multiplier Type	180nm Technology		45nm Technology	
	Area (μm^2)	Power (pW)	Area (μm^2)	Power (pW)
Exact Dadda Multiplier	90930	23909	15238	4992
Multiplier ML	50801	6134	10722	1972
Multiplier L	57607	14971	11348	3614

Fig. 16 and Fig. 17 show the layout designs for 32-bit Multiplier ML and Multiplier L.

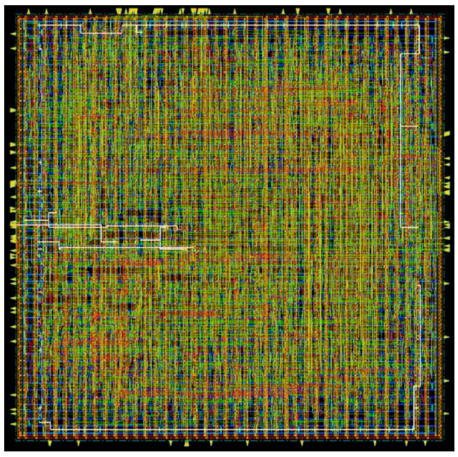

Fig 16. Layout for 32-bit Multiplier ML

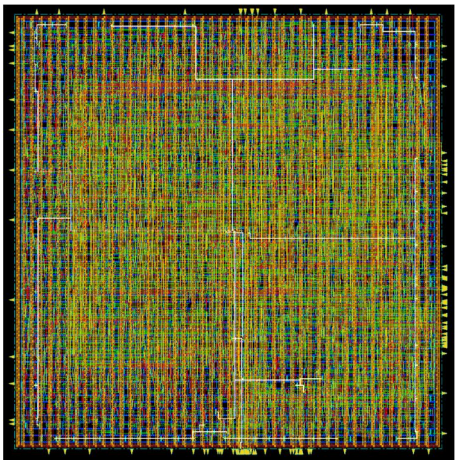

Fig 17. Layout for 32-bit Multiplier L

IV. Conclusion

Multipliers play a prominent role in the design of huge computational blocks. Multipliers are the integral part of any circuit that is built in present world. They are the building blocks of many complex designs. A circuit that consumes low power are preferred in modern days. In this fast moving generation, low power consuming devices are well appreciated and are preferred. So in this paper, a low-power and low-area 32-bit altered partial product multipliers are proposed.

References

[1] V. Gupta, D. Mohapatra, A. Raghunathan and K. Roy, "Low-Power Digital Signal Processing using Approximate Adders," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.,* vol. 32, no. 1, pp. 124-137, Jan 2013.

[2] H. R. Mahdiani, A. Ahmadi, S.M. Fakhraie, and C. Lucas, "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft Computing Applications," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 57, no. 4, pp. 850-862, Apr. 2010.

[3] J. Liang, J. Han, F. Lombardi, "New Metrics for the Reliability of Approximate and Probabilistic Adders," *IEEE Transactions On Computers,* vol. 63, no. 9, pp.1760-1771, 2013.

[4] M. J. Schulte and E. E. Swartzlander, Jr., "Truncated Multiplication with Correction Constant," VLSI Signal Processing VI, pp. 388-396, 1993.

[5] E. J. King and E. E. Swartzlander, Jr., "Data Dependent Truncated Scheme for Parallel Multiplication," *in Proceedings of the Thirty First Asilomar Conference on Signals, Circuits and Systems, pp. 1178-1182, 1998.*

[6] Z. Wang, G. A. Jullien, and W.C. Miller, "A New Design Technique for Column Compression Multipliers," *IEEE Trans. Comput.,* vol. 44, pp. 962-970, Aug. 1995.

[7] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N.S Kim, "Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,* vol. 23, no. 6, pp. 1180-1184, Jun 2015.

[8] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading Accuracy for Power in a Multiplier Architecture," *J. Low Power Electron.,* vol. 7, no. 4, pp. 33-38.

[9] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,* vol. 24, no. 10, pp.3105-3117, Oct. 2016.

[10] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and Analysis of Approximate Compressors for Multiplication." *IEEE Trans.Very Large Scale Integr. (VLSI) Syst.,* vol. 12, no. 5, pp. 522-531, May 2004.

[11] J. Gu, C. H. Chang, "Ultra Low-Voltage, Low-Power 4-2 Compressor for High Speed Multiplications," *in Proc. 36th IEEE Int. Symp. Circuits Systems, Bangkok, Thailand,* May 2003.

[12] C. Chang, J. Gu, M. Zhang "Ultra Low-Voltage Low-Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits," *IEEE Transactions on Circuits & Systems,* vol. 51, No. 10, pp. 1985-1997,Oct. 2004.

[13] M. Margala and N. G. Durdle, "Low-Power Low-Voltage 4-2 Compressors for VLSI Applications," *in Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Design,* 1999, pp. 84-90.

[14] S. Venkatachalam, Seok-Bum Ko, "Design of Power and Area Efficient Approximate Multipliers" *IEEE Transactions on VLSI systems,* vol. 25, no. 5, pp. 1782-1786.