ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

Study and Analysis of Multiplier Units for Low Power MAC Units

Athulya T K, Aathira Mohandas, Abhaya Baby, Alina M T[1], Prajeesh P A[2]

UG Student, Dept. of ECE, ASIET, Kalady, Ernakulam, Kerala, India [1]

Assistant Professor, Dept. of ECE, ASIET, Kalady, Ernakulam, Kerala, India [2]

ABSTRACT: The Multiplier accumulator unit (MAC) unit finds its use in microprocessors, logic circuits and digital signal processors. The major problem phased by industry now is the increased power consumption by these units. This paper intends to bring out a detailed analysis of various multiplier units implemented within MAC units. Also it aims to portray a comparison based on the analysis of the existing systems. A detailed comparison between Vedic multiplier, booth multiplier and modified booth multiplier is brought about. The ultimate aim is to bring forward the best possible multiplier unit.

KEYWORDS: Multiplier, ModelSim ISE, VLSI, Xilinx.

I. INTRODUCTION

Multipliers play an important role in today's digital signal processing and various other applications. Multiplication is one of the most critical operations in many computational systems. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI (Very Large Scale Integrated Circuit) implementation. The growing popularity of portable and multimedia devices such as video phones, note books in which multipliers play the important role. Multiplication is an important part of real-time digital signal processing (DSP) applications ranging from digital filtering to image processing.

Lowering down the power consumption and enhancing the processing performance of the circuit designs are undoubtedly the two important design challenges of wireless multimedia and DSP applications, in which multiplications are frequently used for key computations, such as Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), quantization, and filtering. All multiplication methods share the same basic procedure - addition of a number of partial products. A number of different methods can be used to add the partial products. The simple methods are easy to implement, but the more complex methods are needed to obtain the fastest possible speed [1].

Multipliers have large area, long latency and consume considerable power. Reduction of power consumption makes a device reliable. Therefore, low power multipliers with high clock frequencies play an important role in today's digital signal processing. Most DSP systems incorporate a multiplication unit to implement algorithms such as convolution and filtering. Multiplications are basic arithmetic operations used virtually in all applications involving digital signal processing. Multiplication can be considered as a series of repeated additions. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better. Multipliers are classified by the format in which words are accessed namely:

A. Serial multiplier
B. Parallel multiplier
C. Serial-parallel multiplier

II. BOOTH MULTIPLIER

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. It is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. Booth algorithm is a method that will reduce the number of multiplicand multiples. First of all, booth recoding has to be performed on either of the two values to be multiplied as per in Fig.1. Booth's algorithm examines adjacent pairs of bits of the *N*-bit multiplier *Y* in signed two's complement representation, including an implicit bit below the least significant bit, $y_{-1} = 0$. For each bit y_i, for *i* running from 0 to *N*-1, the bits y_i and y_{i-1} are considered. Where these two bits are equal, the product accumulator *P* is left unchanged. Where $y_i = 0$ and $y_{i-1} = 1$, the multiplicand times 2^i is added to *P*; and where $y_i = 1$ and $y_{i-1} = 0$, the multiplicand times 2^i is subtracted from *P*. The final value of *P* is the signed product.

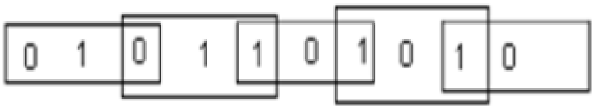


Fig.1: Booth Recoding

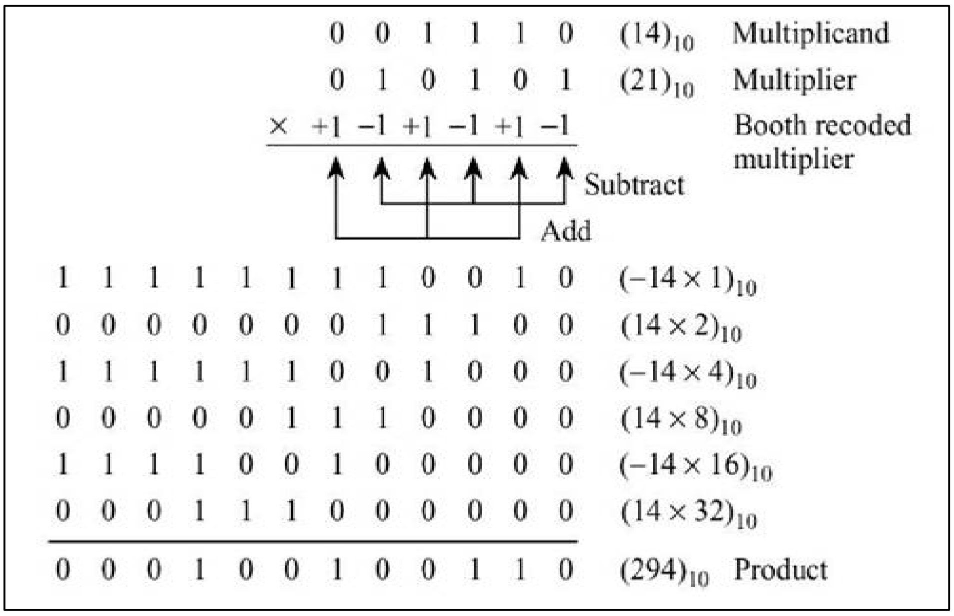


Fig. 2: Booth Multiplication

The above Fig.2 shows a booth multiplication between operands 14 and 21. The algorithm is often described as converting strings of 1's in the multiplier to a high-order +1 and a low-order –1 at the ends of the string. When a string runs through the MSB, there is no high-order +1, and the net effect is interpretation as a negative of the appropriate value. Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values *A* and *S* to a product *P*, then performing a rightward arithmetic shift on *P*. Let **m** and **r** be the multiplicand and multiplier, respectively; and let *x* and *y* represent the number of bits in **m** and **r**.

1. Determine the values of *A* and *S*, and the initial value of *P*. All of these numbers should have a length equal to $(x + y + 1)$.

- A: Fill the most significant (leftmost) bits with the value of **m**. Fill the remaining $(y + 1)$ bits with zeros.
- S: Fill the most significant bits with the value of (−**m**) in two's complement notation. Fill the remaining $(y + 1)$ bits with zeros.
- P: Fill the most significant *x* bits with zeros. To the right of this, append the value of **r**. Fill the least significant (rightmost) bit with a zero.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization   Vol. 5, Special Issue 4, March 2016

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

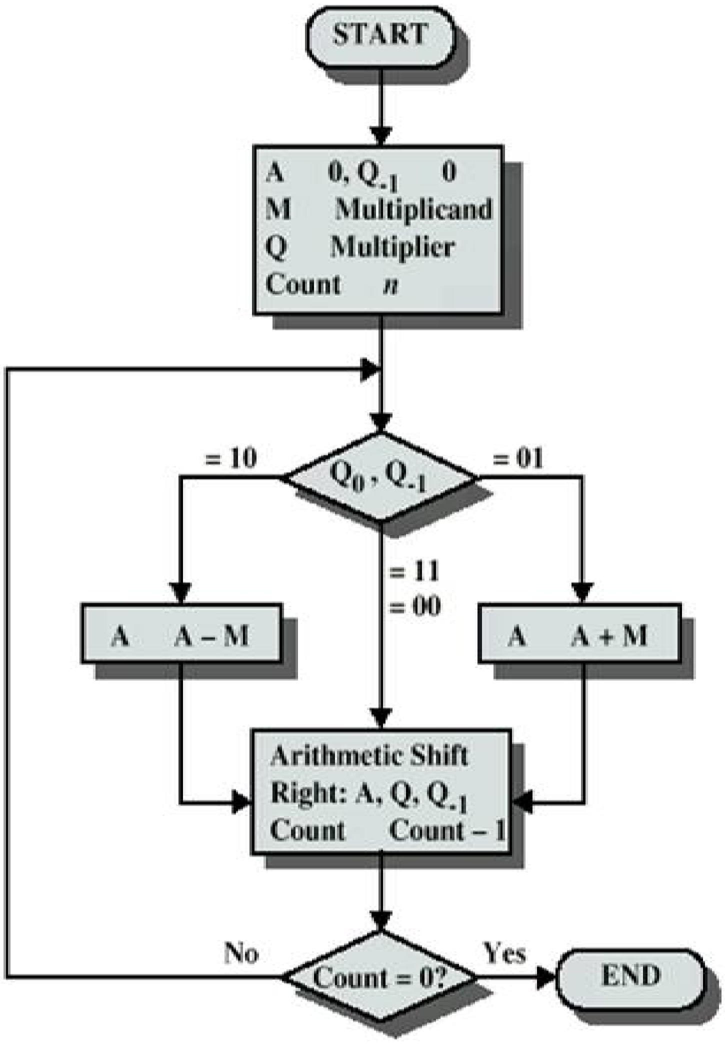


Fig. 3: Booth Algorithm flowchart

2. Determine the two least significant bits of *P*.
- If they are 01, find the value of *P* + *A*. Ignore any overflow.
- If they are 10, find the value of *P* + *S*. Ignore any overflow.
- If they are 00, do nothing. Use *P* directly in the next step.
- If they are 11, do nothing. Use *P* directly in the next step.
3. Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let *P* now equal this new value.
4. Repeat steps 2 and 3 until they have been done *y* times.
5. Drop the least significant (rightmost) bit from *P*. This is the product of **m** and **r**.

III. MODIFIED BOOTH MULTIPLIER

Modified Booth Algorithm (MBA) is the method commonly used for achieving high speed multiplication and also reduce number of partial products. Also by using radix-4, radix-8, radix-16 and radix-32 booth Encoding technique, the partial products are further reduced, which increases complexity and improves the performance. Also by using MBA algorithm speed, has been increased [1].
Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is possible to reduce the number of partial products by half, by using the technique of radix-4 Booth

recoding. The basic idea is that, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, we only take every second column, and multiply by ±1, ±2, or 0, to obtain the same results.

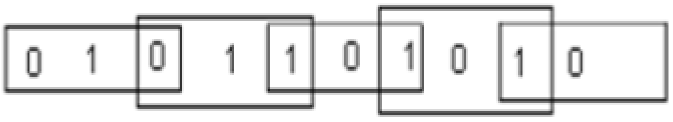


Fig. 4: Grouping of bits for recoding

TABLE 1: MODIFIED BOOTH ENCODING TABLE

X (i+1)	Xi	X(i-1)	Action
0	0	0	O x Y
0	0	1	1 x Y
0	1	0	1 x Y
0	1	1	2 x Y
1	0	0	-2 x Y
1	0	1	-1 x Y
1	1	0	-1 x Y
1	1	1	0 x Y

The advantage of this method is the halving of the number of partial products. To Booth recode the multiplier term, we consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping starts from the LSB and the first block only uses two bits of the multiplier [4]. Fig.4 shows the grouping of bits from the multiplier term for use in modified booth encoding. Each block is decoded to generate the correct partial product. The encoding of the multiplier Y, using the modified booth algorithm, generates the following five signed digits, -2, -1, 0, +1, +2. Each encoded digit in the multiplier performs a certain operation on the multiplicand, X, as illustrated in Table 1.

IV. VEDIC MULTIPLIER

Vedic multiplication method based on one of the sutra of Urdhva Tiryakbhyam [3]. In this method S0 is vertical product of bit A0 &B0,S1the sum ofA0*B1and A1*B0and tens bit of previous sum (S0). All the multiplication is done with the same process. The ten’s and the hundred’s position bit are forward as the carry to next addition. Only the S7 bit of answer will comes from the tens bit S6 in the 4 bit multiplication process. In the n bit multiplication process the 2n-1 bit of answer is calculated like the same way as S7 bit. Let us consider two inputs X1X0 and Y1Y0, each of 2 bits. P3P2P1P0 represents each bit of the final computed product.

X1 X0 ×
Y1 Y0
———————————
X1Y0 X0Y0 X1Y1 X0Y1
————————————
P3 P2 P1 P0

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization *Vol. 5, Special Issue 4, March 2016*

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

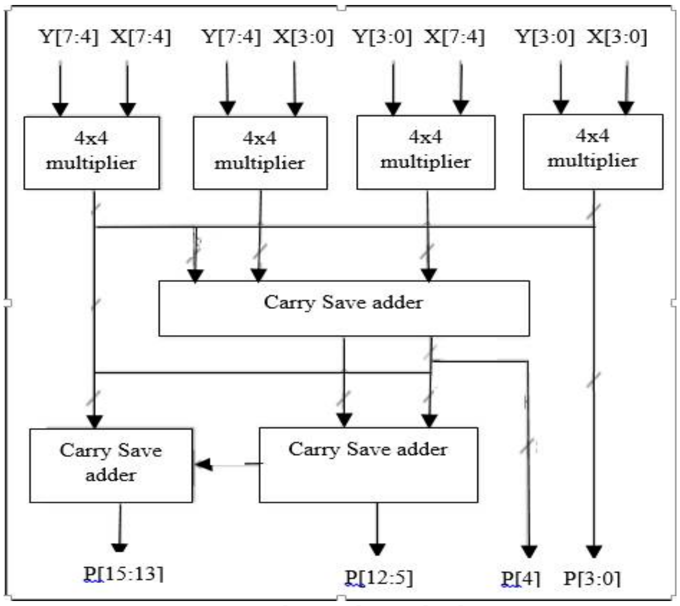


Fig. 5: 8 bit Vedic multiplier

In Vedic mathematics, the vertical multiplication of bits X0 and Y0 gives product P0. The product term P1 is obtained by the addition of crosswise bit multiplication i.e. X1 & Y0 and X0 & Y1. P2 is addition of the vertical product of bits X1 & Y1 along with the carry generated from the previous addition during P1. P3 output is the carry generated from the previous calculation of P2. Fig. 1 shows the module that generate 4 outputs from the two 2-bit inputs by using AND gates and half adder and it is known as 2×2 multiplier block. This is similar to shift and add technique however, to further improve the performance of the multiplier, divide and conquer strategy is used for higher bits.

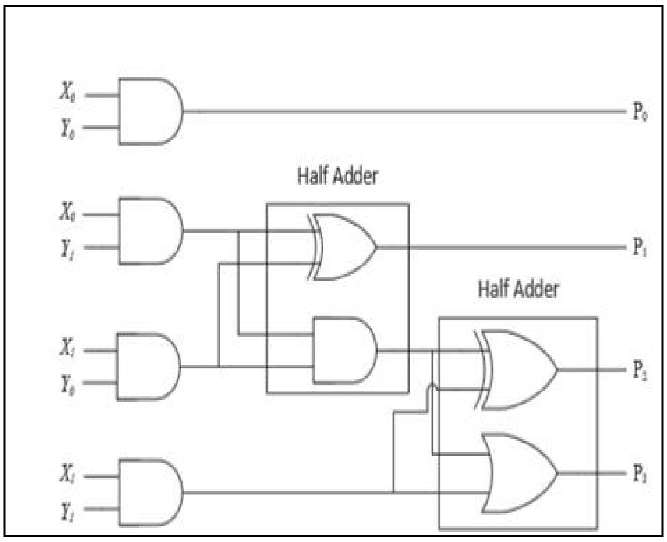


Fig. 6: 2 bit Vedic multiplier

Now, let us analyze for a 4×4 multiplication that gives output product as P7P6P5P4P3P2P1P0. The multiplicand and multiplier are decomposed equally as XH=X3X2 and XL=X1X0 for X and YH=Y3Y2 and YL=Y1Y0 for Y, where H and L represents higher and lower order bits of X and Y . Fig.5 shows the 4 bit multiplication by taking two bit at a time and using 2×2 multiplier block. According to Vedic mathematics, initially vertical multiplication of XL & YL is carried out, which gives partial product outputs p0[3 : 0] [2].
Then, the middle one shows crosswise multiplication of two, 2 × 2 multiplier with inputs XH & YL and XL & YH, generates p1[3 : 0] and p2[3 : 0] partial product outputs respectively Finally, the last block inputs XH & YH multiplies

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

An ISO 3297: 2007 Certified Organization Vol. 5, Special Issue 4, March 2016

National Conference on Signal Processing, Instrumentation and Communication Engineering (SPICE' 16)

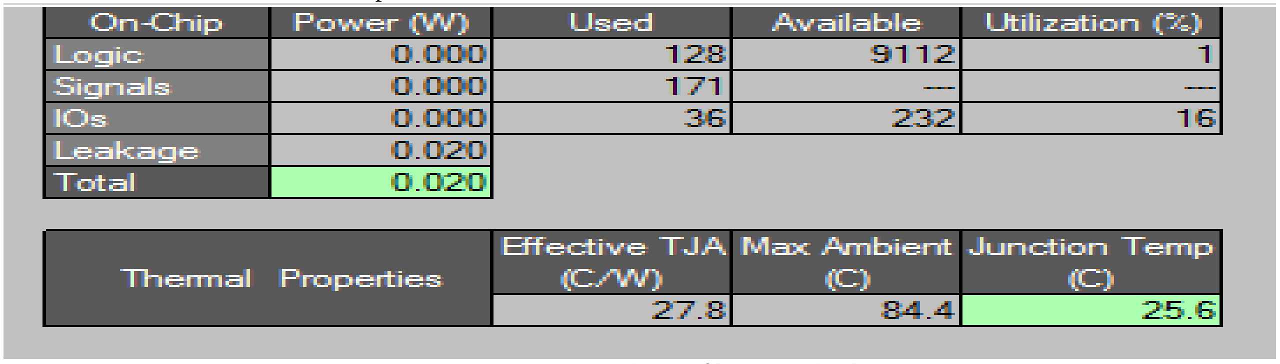
Organized by

Dept. of ECE, Mar Baselios Institute of Technology & Science (MBITS), Kothamangalam, Kerala-686693, India

vertically and generates the partial product outputs as p3[3 : 0]. The first two final product outputs P0 and P1 are same as that of the partial products p0[0] and p0[1]. The remaining product terms are obtained by using three conventional adders as shown in Fig. 3. The inputs for adder1 are {p3[3 : 0],00} and {00,p2[3 : 0]} and for adder2 are p1[3 : 0] and {00,p0[3 : 2]}. The outputs of adder1 and adder2 are given inputs to the adder3, that generates the final product terms P[7 : 2]. Thus, the final product terms are obtained by parallel multiplication using sub multiplier blocks and addition as P7P6P5P4P3P2P1P0, that can reduces the delay of the multiplier.

V. RESULT AND DISCUSSION

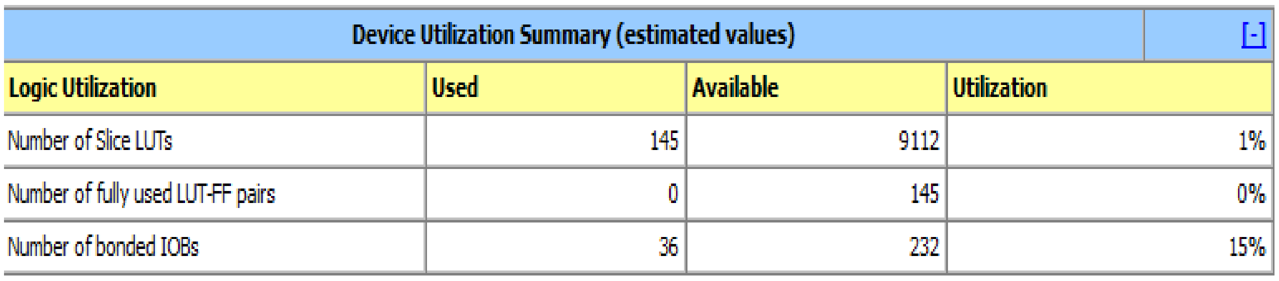
The below shown is the result of resource utilization and power analysis of booth multiplier, modified booth and Vedic multiplier. From the above three analysis report we can make out that the resource utilization for the modified booth is almost half of that of a booth multiplier.



On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	128	9112	1
Signals	0.000	171	---	---
IOs	0.000	36	232	16
Leakage	0.020			
Total	0.020			

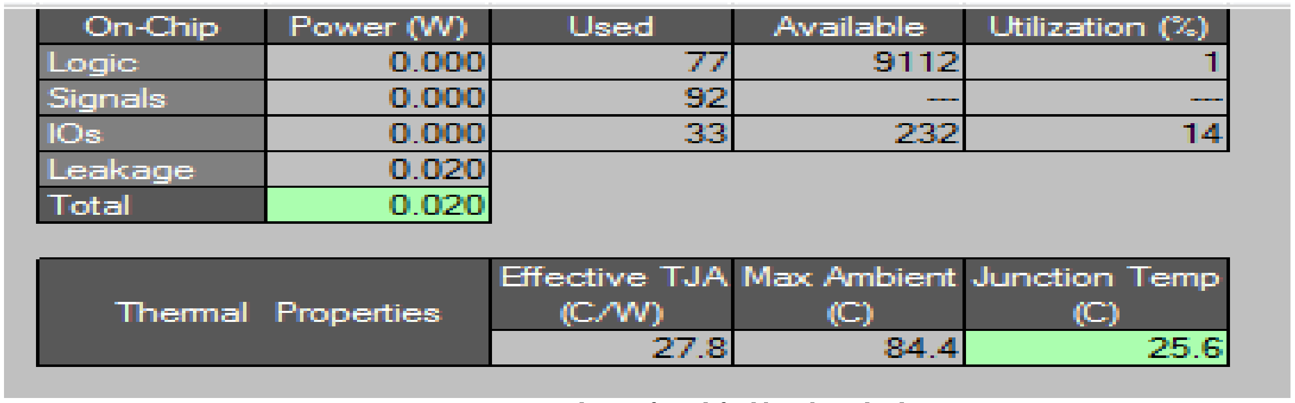
Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	27.8	84.4	25.6

Fig. 7: Power analysis of booth multiplier



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	145	9112	1%
Number of fully used LUT-FF pairs	0	145	0%
Number of bonded IOBs	36	232	15%

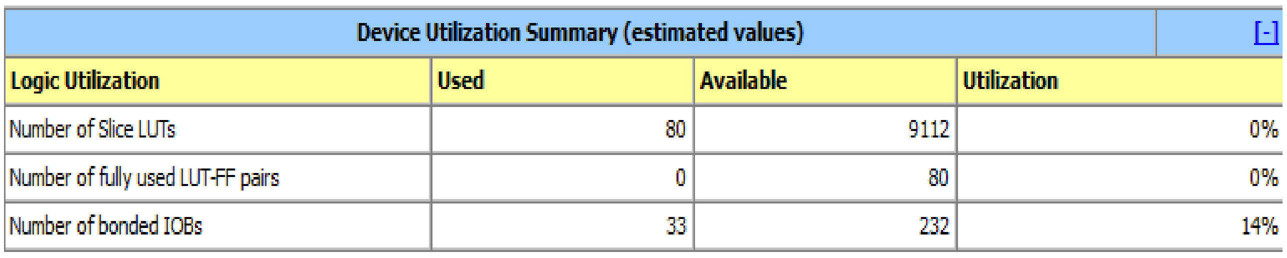
Fig. 8: Resource utilization of booth multiplier



On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	77	9112	1
Signals	0.000	92	---	---
IOs	0.000	33	232	14
Leakage	0.020			
Total	0.020			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	27.8	84.4	25.6

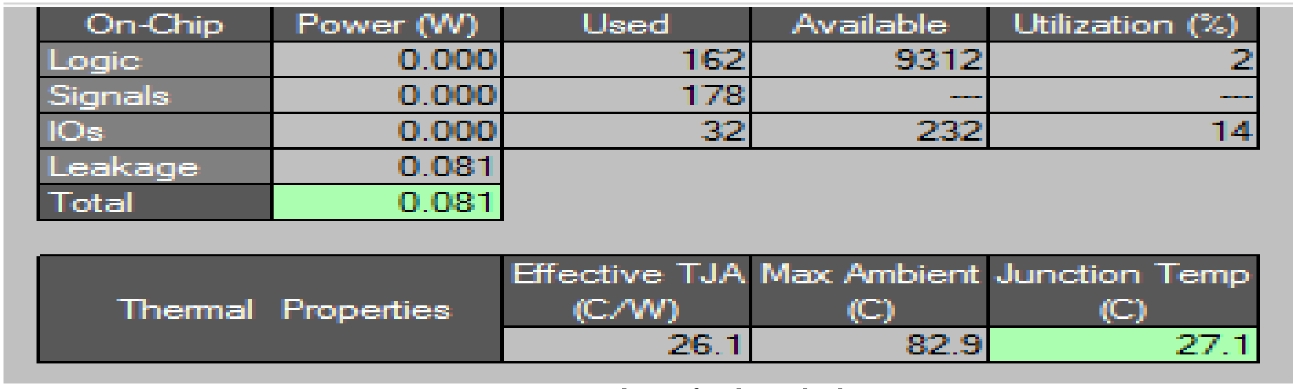
Fig. 9: Power analysis of modified booth multiplier



Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	80	9112	0%
Number of fully used LUT-FF pairs	0	80	0%
Number of bonded IOBs	33	232	14%

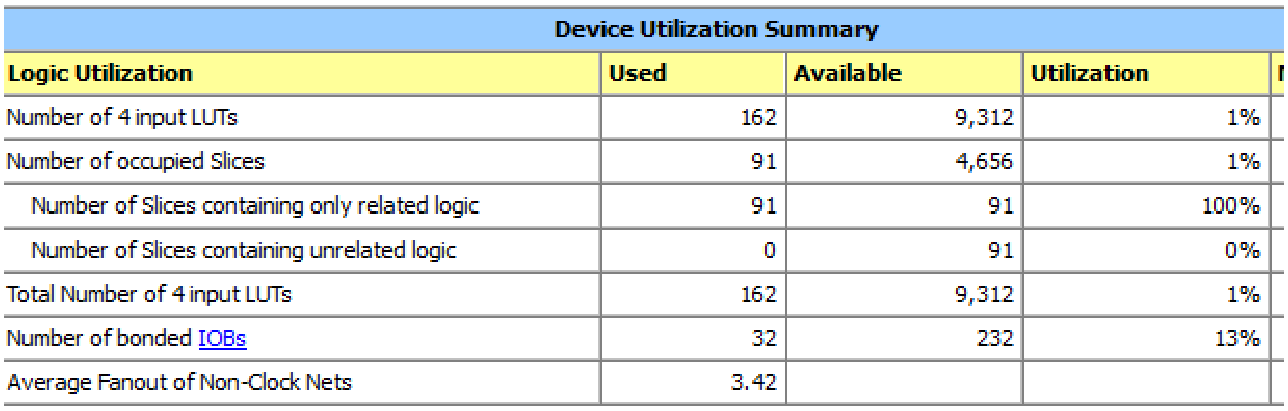
Fig. 10: Resource utilization of modified booth multiplier



On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	162	9312	2
Signals	0.000	178	---	---
IOs	0.000	32	232	14
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Fig. 11: .Power analysis of vedic multiplier



Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	162	9,312	1%
Number of occupied Slices	91	4,656	1%
Number of Slices containing only related logic	91	91	100%
Number of Slices containing unrelated logic	0	91	0%
Total Number of 4 input LUTs	162	9,312	1%
Number of bonded IOBs	32	232	13%
Average Fanout of Non-Clock Nets	3.42		

Fig. 12: Resource utilization of vedic multiplier

The least number of IOBs utilized is in the case of Vedic multiplier. Thus modified booth multiplier is seen to have better performance compared to any other multipliers considered here.

VI. CONCLUSION AND FURTHER ENHANCEMENT

The multiplier units can further be made low power by including an additional unit along with it called Spurious Power Suppression Technique (SPST) unit. This unit avoids the unnecessary addition happening in the multiplier stage that is, if the MSBs are all zeros that part of the computation is directly avoided and the output values are generated as per the circuit set as per five cases considered [1][5][6].

REFERENCES

[1] M. Bhagya Lakshmi Devi, Jala.Vijaya Kumar, "Implementation of Power Optimized Multiplier using SPST Technique", *International Journal of Advanced Technology and Innovative Research* Volume. 06, IssueNo.09, October-2014, Pages: 969-974.

[2] Poornima M, Shivaraj Kumar Patil, Shivukumar , Shridhar K P , Sanjay H, "Implementation of Multiplier using Vedic Algorithm," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-2, Issue-6, May 2013

[3] Garima Rawat, Khyati Rathore, Siddharth Goyal, Shefali Kala and Poornima Mittal, "Design and Analysis of ALU: Vedic Mathematics Approach," *International Conference on Computing, Communication and Automation* (ICCCA2015), ISBN:978-1-4799-8890-7/15/$31.00 ©2015 IEEE.

[4] Rasika Nigam, Jagdish Nagar, "VLSI implementation of modified Booth Algorithm," *International Journal of Engineering and Technical Research (IJETR)* ISSN: 2321-0869, Volume-3, Issue-4, April 2015.

[5] Kuan-Hung Chen and Yuan-Sun Chu, "A Spurious-Power Suppression Technique for Multimedia/DSP Applications", *IEEE transactions on circuits and systems*—i: regular papers, vol. 56, no. 1, January 2009

[6] G. Sasi, "Design of Low Power / High Speed Multiplier using Spurious Power Suppression Technique (SPST)," *IJCSMC*, Vol. 3, Issue. 1, January 2014, pg.37 – 41