Indian Journal of Science and Technology, *Vol 8(13), 54495, July 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Low-Power Hybrid Multiplication Technique for Higher Radix Hard Multiples Suppression

M. V. Jithin Kumar[1] and K. B. Jayanthi[2*]

[1]Department of Electronics and Communication Engineering, Heera College of Engineering and Technology, Kalliyode, Thiruvananthapuram - 695568, Kerala, India; mv.jithin@gmail.com
[2]Department of Electronics and Communication Engineering, K. S. Rangasamy College of Technology, Namakkal District - 637215, Tamil Nadu, India; jayanthikb@gmail.com

Abstract

This paper presents a low-power higher radix multiplication algorithm based on Radix-16 32×32 bit Modified Booth Encoder (MBE). Hard multiples are the major factors for power consumption in higher radix MBE. This paper introduces the design of a Hybrid Multiplication Technique (HMT) to suppress the hard multiples that exist in a Radix-16 32×32 bit MBE. The proposed HMT uses Radix-8 and Radix-4 encoding technique along with Radix-16 to avoid the hard multiples. Experimental results based on Synopsys SDK 90nm, 1.32V standard-cell library show that the proposed HMT reduces power consumption up to 25% and 21% in comparison with conventional Radix-16 and Radix-8 MBE respectively. HMT equipped Radix-16 MBE also gives better performance than existing techniques with respect to frequency and power.

Keywords: Hard Multiples, Higher-Radix Modified Booth Multiplier, Low-Power, Slack, Synthesis

1. Introduction

A lot of preceding digital multipliers like those used in DSP and MAC are aimed at transition or switching activity reduction to decrease power dissipation. The switching activity reduction method used in a Modified Booth Multiplier (MBM) is the Radix-4 recoding scheme[1]. Radix-4 MBE is being widely used in parallel multipliers to diminish the number of Partial Products (PPs) by a factor of two. These PPs are generated by shifting and 2's complement operation[2-4]. It does not generate hard multiples. Better savings in area and dynamic power dissipation are possible for higher word-length multipliers by raising the bit coding which is higher than Radix-4[5].

A technique that reduces number of full adders in the compression tree of signed/unsigned Modified Booth Algorithm (MBA) is described in[6]. This technique is based on the need to extend the sign bit of the PPs. Working on the extension of 2nd order MBA a novel concurrent carry save Multiplier Accumulator (MAC) architecture is shown in[7] with different computational performance.

A combination of wide-bit range Radix-4/Radix-8 architecture is proposed in[8] to optimize between the high speed and the low power dissipation in Radix-4 and Radix-8 multiplier architecture, respectively. A new encoding scheme with Multiple-Level Conditional-Sum Adder (MLCSMA) hybrid structure is proposed in[9] to improve the performance up to 25% in booth encoded parallel multiplier design. A hybrid spare-tree structure is used in[10] to implement two's complement circuit which further reduces the area to 15.8% and improve the speed up to 11.7% over the standard design. Commercial processors like multimedia and DSP have the Radix-8 multiplier design to boost their speed thereby reducing the number of PPs. The number of PPs in a digit representation is reduced by ⅔ in Radix-8 encoding. But its performance bottleneck is the generations of the terms 3X and 4X also termed as 'Hard Multiples'. This term is usually computed by adding and shifting operations (+4X=+3X+X) in a high-speed adder. In a 3X+X addition, full adders share the same input signal. This property consumes more power in hard multiples expression related to process[11].

*Author for correspondence

The proposed work optimizes the internal architecture of MBM which controls dynamic multiplier resources to match peripheral data characteristics[12]. The prime objective is power reduction by avoiding the hard multiples in a Radix-16 32×32 bit MBE along with the use of Radix-4 and Radix-8 MBE. By using Hybrid architectures, it is possible to achieve both power and delay reduction which is the strength of high-level optimization. The paper is organized as follows: Section 2 discusses the design of the proposed algorithm. Section 3 gives the synthesis and performance analysis of Hybrid Multiplier.

2. Hybrid Multiplication Technique

The regular linear array multiplication can be expressed by considering a binary integer *Multiplicand (M) and Multiplier (R),*

If $M = \sum_{i=0}^{m-1} M_i 2^i$ and $R = \sum_{j=0}^{n-1} R_j 2^i$

The product, P= $\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (M_i R_j) 2^{i+j}$

$$P = \sum_{k=0}^{m+n-1} P_k 2^k \quad (1)$$

Where '*mn*' PPs are obtained, they are produced by a set of AND gates as shown in Figure 1. Scheme[13] is an example for linear array multiplier.

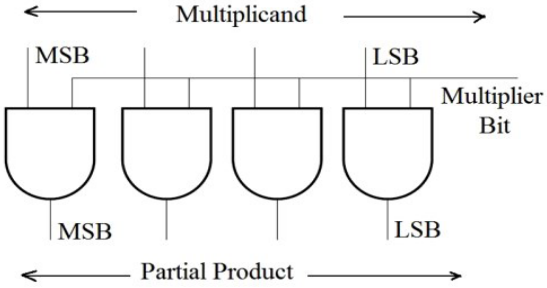

Figure 1. 4×4 bit linear array multiplier PP generator.

These 'mn' PPs consume more power. So a booth algorithm and its modification is proposed in[14]. In a 2'complement binary number with n-bit Multiplicand (M) and Multiplier (R) the multiplier bit group where 'n' is even can be represented by;

$$R = \sum_{i=0}^{\frac{n-2}{2}} (r_{2i+1} + r_{2i} - 2r_{2i+1}) 2^{2i} \quad (2)$$

$$M \times R = \sum_{i=0}^{\frac{n-2}{2}} S_i \quad (3)$$

Where, $S_i = (r_{2i+1} + r_{2i} - 2r_{2i+1}) 2^{2i}$ (4)

As shown in equation number 4, the multiplier is separated by overlapping groups of 3 bits with an appended zero (represented by Z) at the least significant bit. The group is represented in another form by S_k .

If 'n' is even, and k = N-1,

$S_k = (b_{2k+1} b_{2k} b_{2k-1})$ (5)

Where, $N-2) \leq k \leq 1$

$S_k = (b_{2k+1} b_{2k}\ Z)$, for k=0 (6)

If 'n' is odd,

$S_k = (b_{2k} b_{2k} b_{2k-1})$, for k=N-1 (7)

Where, the sign bit is repeated by b_{2k} .

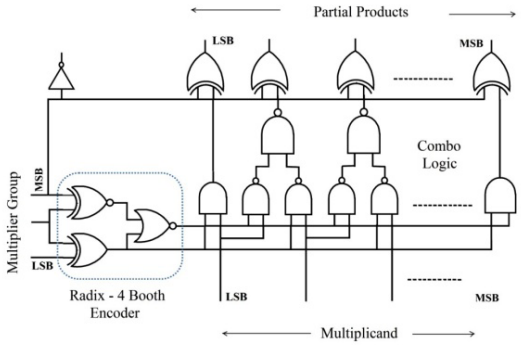

Figure 2. Radix-4 modified booth encoder and PPs generator.

The combinations of S_k are generated by a Radix-4 booth encoder as shown in Figure 2. For MBA, the multiples S_k {0, ±1, ±2} are used to generate the PPs. The final product is given as;

$$P = M \times R = M.S_{N-1} 2^{2(N-1)} + M.S_{N-2} 2^{2(N-2)} + \cdots + M.S_1 2^2 + M.S_0 2^0 \quad (8)$$

The above equation shows that each PP is shifted by 2 bits before compression to produce the correct result. This PP generation can be further reduced by higher radix

encoding techniques like Radix-8, Radix-16 etc. In Radix-8 encoding technique, the multiplier is separated by 4-bit group with an appended zero at the least significant bit. If 'n' is even and k = N-1, the Radix-8 multiplier group can be represented by;

$$S_k = (b_{3k+2} b_{3k+1} b_{3k} b_{3k-1}) \quad (9)$$

Where, $N-2) \leq k \leq 1$

$$S_k = (b_{3k+2} b_{3k+1} b_{3k}\ Z) \text{ for k=0} \quad (10)$$

If 'n' is odd,

$$S_k = (b_{3k+1} b_{3k+1} b_{3k} b_{3k-1}) \text{ for k=N-1} \quad (11)$$

Where, the sign bit is repeated by b_{3k+1}.

The combinations of b_{3k+1} are produced by Radix-8 encoder as shown in Figure 3. The multiples S_k {0, ±1, ±2, ±3, ±4} are used to generate the PPs and the final product is;

$$P = M \times R = M.S_{N-1}2^{3(N-1)} + M.S_{N-2}2^{3(N-2)} + \cdots + M.S_1 2^3 + M.S_0 2^0 \quad (12)$$

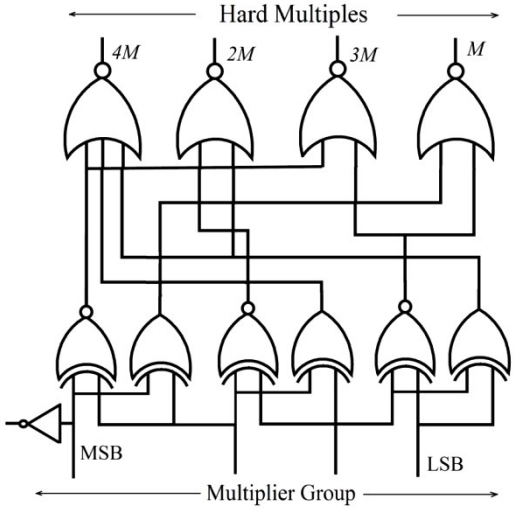

Figure 3. Radix-8 modified booth encoder.

Hence each PP is shifted by 3 bits before compression. In Radix-16 encoding technique, the multiplier is separated by a 5-bit group and an appended zero is added at the least significant bit.

If 'n' is even and k = N-1, the Radix-16 multiplier group S_k can be represented by;

$$S_k = (b_{4k+3} b_{4k+2} b_{4k+1} b_{4k} b_{4k-1}) \quad (13)$$

Where,

$N-2) \leq k \leq 1$ $S_k = (b_{4k+2} b_{4k+1} b_{4k} b_{4k-1} Z)$ for k=0 (14)

If 'n' is odd,

$$S_k = (b_{4k+2} b_{4k+2} b_{4k+1} b_{4k} b_{4k-1}) \text{, k = N-1} \quad (15)$$

The multiples S_k { $0, \pm1, \pm2, \pm3\ \pm4, \pm5, \pm6, \pm7, \pm8$ } are used to generate the PPs. The final product can be written as;

$$P = M \times R = M.S_{N-1}2^{4(N-1)} + M.S_{N-2}2^{4(N-2)} + \cdots + M.S_1 2^4 + M.S_0 2^0 \quad (16)$$

The equation numbered from (12) to (16) often generates hard multiples. In Radix-16 booth encoding, the number of PPs are reduced by $^1/_4$ [15].

The equation numbered from (12) to (16) often generates hard multiples. In Radix-16 booth encoding, the number of PPs are reduced by $^1/_4$ [15]. However these reductions in the number of PPs come at the expense of increased complexity in their generation. Specifically, the generation of hard multiples $\pm 3M, \pm 5M, \pm 6M, \pm 7M, \pm 8M$ results in increase in delay.

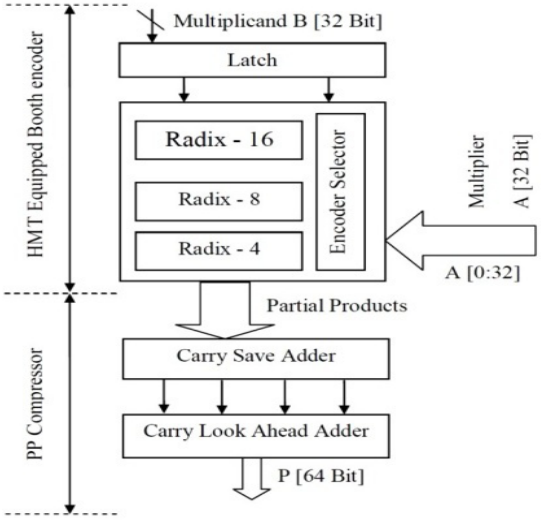

Figure 4. Block diagram of HMT scheme for 32×32 bit MBM.

To avoid these hard multiples, a HMT is proposed as shown in Figure 4. It is a block diagram which has 2 sections, namely HMT equipped booth encoder and PP compressor. Section 1 is designed with an encoder selector, Radix-4, Radix-8 and Radix-16 MBA. Section 2 uses Carry Save Adder (CSA) to compress the five main PPs from HMT block. The results from the CSA stages are fed to Carry Look Ahead based fast adder to produce the final product[16]. The PPs with two's complement format creates sign extension crisis. Instead of hard multiples, adding the PPs with different bit length in Radix-4 and Radix-8 cause sign extension problem. If the sign bit is negative, it increases the number of overhead bits in PPs. So before PP compression few modifications[15,17] have to be done in 2's complement signed MBM. The encoder selector is designed using Table 1. The Radix-16 five bits groups are separated and given to different encoders like Radix-4, Radix-8 or Radix 16 so that hard multiples will not be generated. With the help of Table 1, a Karnaugh map for the design of encoder selector can be realised as shown in Figure 5.

CDE / AB	C'D'E' 000	C'D'E 001	C'DE 011	C'DE' 010	CDE' 110	CDE 111	CD'E 101	CD'E' 100
A'B' 00	R16	R16	R16	R16	R4	R4	R4	R16
A'B 01	R4	R4	R4	R4	R8	R8	R8	R4
AB 11	R4	R4	R16	R4	R16	R16	R16	R16
AB' 10	R8	R8	R8	R8	R4	R4	R4	R8

Figure 5. 5-Variable K-map reduction for encoder selector Logic.

The equation for the different encoder selector logic is given by,

R4 Encoder = 'BD'E'+BC'DE'+A'BC'+BC'D'+B'CD+B'CE

R8 Encoder = A'BCD+A'BCE+AB'D'E'+AB'C'

R16 Encoder = ABC'DE+A'B'D'E'+A'B'C'+ABC

Figure 6 shows the encoder selector logic using basic logic gates. It consumes only 3 OR gates, 14 AND gates and 26 NOT gates. The Venn diagram for the different encoder selector using congruent ellipses in a 5-fold rotationally symmetrical arrangement is shown in Figure 7. The truth density of Radix-4, Radix-8 and Radix-16 are 43.75%, 25% and 31.25% respectively. So the maximum logic strength will be in Radix-4 selector logic.

The operation of different MBM encoder selection is performed according to Table 1 with the encoder selector logic as shown in Figure 6. In the HMT scheme Encoder selector will enable the corresponding MBA. The multiplier groups which generates hard multiples in Radix-16 encoding scheme are shifted to Radix-8 and Radix-4 encoding scheme. In Radix-16 encoding scheme the PP collection and summation are expressed by the equation;

$$S_{N-1}2^{4(N-1)} + S_{N-2}2^{4(N-2)} + \ldots + S_1 2^4 + S_0 2^0 \quad (17)$$

In Equation (15) if $S_1 2^4$ is hard multiples, it can be represented by lower radix encoders. Considering $S_1 2^4$ as the combination of any hard multiples the new PP collection and summation can be written as;

$$S_{N-1}2^{4(N-1)} + S_{N-2}2^{4(N-2)} + \ldots + \{ S_{N-1}2^{2(N-1)} + S_{N-2}2^{2(N-2)} + \ldots + S_1 2^2 + S_1 0\ 2^{\uparrow}0\} + S_0 2^0 \quad (18)$$

Instead of the PPs $S_{N-2}2^{4(N-2)}$ and $S_0 2^0$ in Radix-16, if Radix-4 and Radix-8 encoding schemes are used the equation can be represented by;

$$S_{N-1}2^{4(N-1)} + \{S_{N-1}2^{3(N-1)} + S_{N-2}2^{3(N-2)} + \ldots + S_1 2^3 + S_0 2^0\} + \ldots + S_1 2^4 + \{S_{N-1}2^{2(N-1)} + \cdots S_{N-2}2^{2(N-2)} + \ldots + S_1 2^2 + S_0 2^0\} \quad (19)$$

Thus when HMT is used in Radix-16, Radix-4 or Radix-8 or a combination of both may come. Depending on the arithmetic weight the shifting operation should be performed before PP compression. Unlike normal Radix-4 and Radix-8 recoding scheme, there is no virtual zero at the least significant bit of the multiplier in lower order Radix-8 and Radix-4 HMT recoding scheme. Consider a 32-bit binary multiplier 00011000010010100001100001001010. As per Radix-16 MBA concept and Table 1, the recoding of this multiplier binary number can be written as shown in Figure 8. The combined Radix-4, Radix-8 and Radix-16 encoding schemes are used to find the partial products.

For the above binary multiplier conventional Radix-16 encoder generates +0M, +2M multiples and hard multiples. Instead of 8M, 6M hard multiples, it is recommended to use Radix-8 encoding scheme and for +5M hard multiples, Radix-4 scheme as shown in Figure 8. Thus PP1, PP3, PP5 and PP7 are generated by Radix-8 encoding scheme, PP2 and PP6 are generated with Radix-4 MBA, PP4, PP8 and PP9 are generated by Radix-16 encoder. So the system will avoid hard multiples without any complication in the algorithm except some chain adders and shifters for the compression of lower radix PPs. This simple shifting and addition consumes

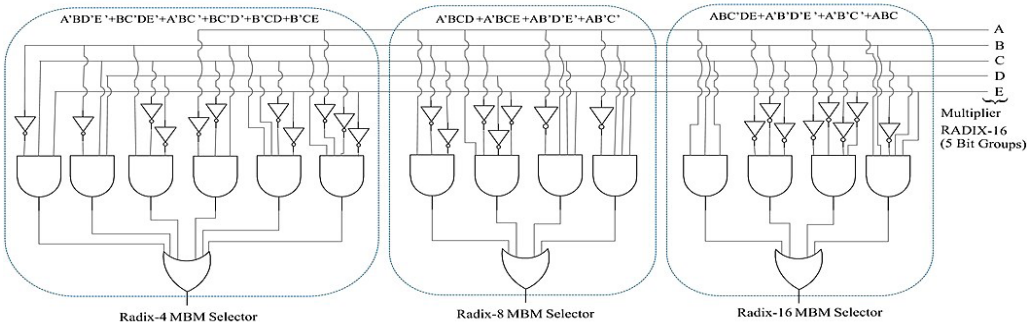

Figure 6. Encoder selector Logic for HMT Scheme.

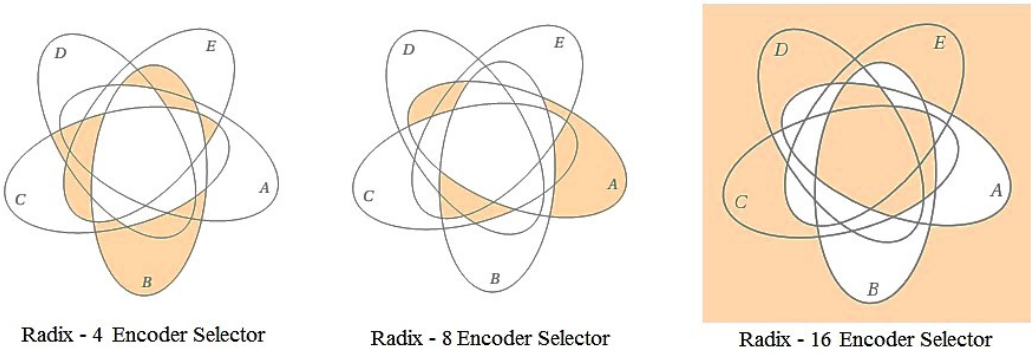

Figure 7. Venn diagram using congruent ellipses for the different Encoder selector.

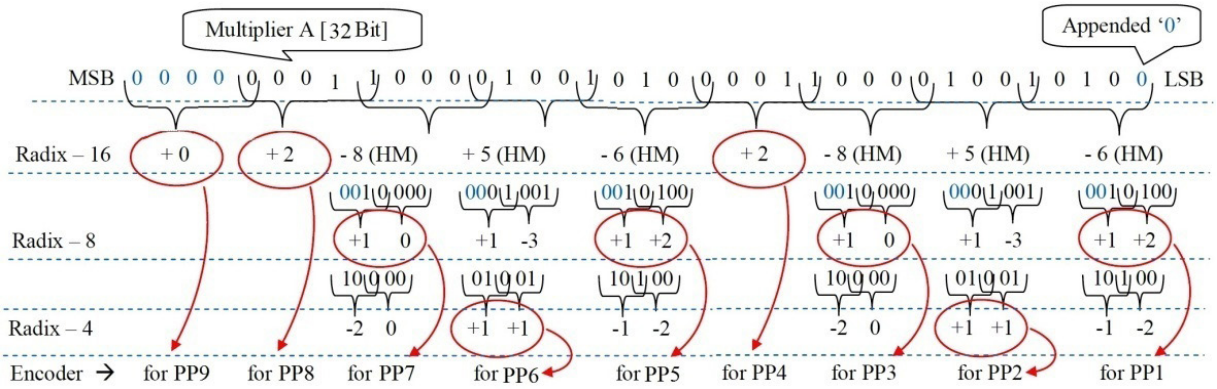

Figure 8. HMT encoder selection scheme.

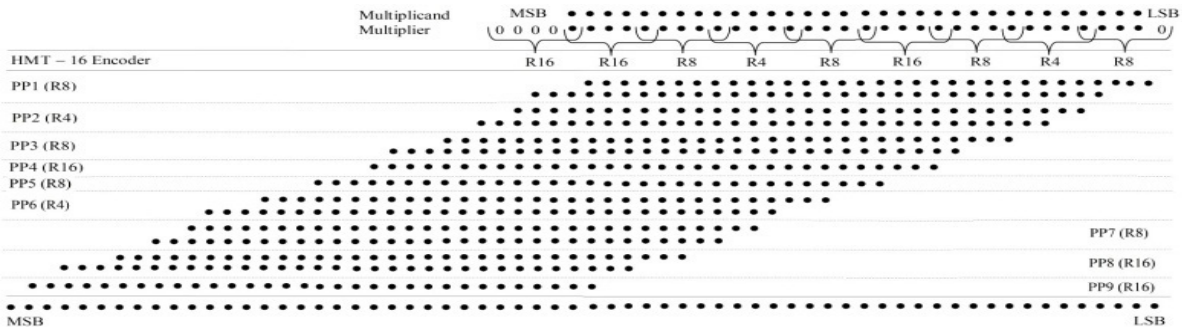

Figure 9. Dot diagram illustration of HMT-16 32×32 Bit MBM.

very less power than the conventional Radix-16 hard multiples. The PP generation method is demonstrated by the use of a dot diagram[15] as shown in Figure 9. The multiplier bit shown in Figure 8 can be interrelated with the Figure 9 multiplier dot diagram. Each PPs are generated with the corresponding encoding scheme and is represented by horizontal dotted rows. The PPs are shifted according to the arithmetic weight of the bits in the multiplier for different encoder. The final product is indicated by 64 bit length dotted rows as shown in Figure 9.

3. Synthesis and Analysis of HMT

The proposed HMT equipped Radix-16(HMT-16), conventional Radix-16, Radix-8 and Radix-4 multipliers are described for 32×32 bit multiplication in Verilog HDL and its RTL functionality verification is done in Synopsys Verilog Compiler Simulator (VCS) in the RH Linux AS v5.7 platform. The individual designs for Radix-16, Radix-8, Radix-4 and HMT-16 are synthesized and optimized by Synopsys Design Compiler (DC) for minimum attainable area and delay independently under the same synthesis design environment. Total number of cells and buf/inv (excluding interconnects) used in HMT-16 at 400MHz is 2803.Table 2 shows that the HMT technique is better than Radix-8 and Radix-16 encoding schemes. Synopsys SDK standard-cell library with operating voltage 1.32V and temperature -40 to +55 as min/max show that the proposed 32×32 bit Hybrid multiplier reduces the power consumption by 25% and 21% compared to conventional Radix-16 and Radix-8 booth encoders, respectively.encoder. The final product is indicated by 64 bit length dotted rows as shown in

A performance assessment of the proposed multiplier with some existing designs[4,7,11,18–21] gives more statistics as shown in Table 3. While comparing frequency, power and delay factor, proposed technique is superior[4,7,18–21]. If a performance based VLSI algorithm is needed, the algorithm should have less critical path and less delay. In the proposed architecture, the delay is very low and maximum frequency that can withstand the proposed circuit is 1 GHz. On comparison[11], proposed design is better in delay.

The area consumption of proposed HMT-16 in 64 × 64 bit multiplication is 27912.432 μm^2.From this it is evident that the ratio of area utilization of HMT-16 may not be very high for large word length multipliers.

So when comparing with[11] proposed design is also a better alternative. From Table 3, it is evident that the power optimization, speed and delay of HMT-16 32×32 bit multiplier is superior to any other MBA. The proposed idea can also be extended to Radix-64, as well as high Radix multipliers. In the case of Radix-64, a combination of Radix-4, Radix-8 and Radix-16 can be used along with Radix-32 MBA. Lower Radix RTL modules can be preserved and used at the time of synthesis for higher radix MBA to reduce the higher radix hard multiples. In general, power savings in multipliers are possible without compromising the system speed, provided that HMT is also a non-critical path. Also the proposed HMT-16's delay factor is better than New Testable Gate (NTG) for testable reversible logic based booth multiplier[22] which is used to build the complex logic circuits for quantum computers.

For further study HMT-16 32×32 bit MBM static time analysis is considered. It is characterized by data arrival time (decided by the maximum combinational path delay or critical path delay) and data required time (Clock period) in an algorithm. The difference of data required time and data arrival time is called slack or timing margin of the path. If slack is negative there is a timing violation on that path. Histogram views can be used to analyse the distribution of timing slack values in the design. Figure 10, 11, 12 shows the positive end-point path slack, positive path slack and Nets capacitance histograms of the entire algorithm at 400MHz, respectively. It shows that HMT equipped Radix-16 32×32 bit multiplier's positive endpoint slack is better than other conventional designs. Figure 10 shows that the number of end-point paths of HMT-16 32×32 bit MBA after the positive slack unit +2.25 is 34 out of 37 paths. This means HMT has good tolerance in constraints.

Figure 11 shows that the conventional encoders have very less positive path slack and it is +0.86for Radix-16. The positive path slack histogram for HMT technique is +2.3 and it is higher than all other conventional MBA. This makes the HMT tolerate constraints better. Figure 12 shows the maximum Nets capacitance value of different encoders using 32×32 bit MBA. For Radix-4, Radix-8 and Radix-16 encoders, the maximum capacitance values are 4.1pf, 5pf and 3pf respectively. The maximum capacitance of HMT is 1.1pf and it is less than any other MBM technique.

Table 1. HMT Encoder Selection Table

Radix 16	B_k	Encoder Selector
00000	+0	Radix-16 $S_{N-1}2^{4(N-1)}$ $S_{N-1}2^{4(N-1)}$ + $S_{N-2}2^{4(N-2)}$ $S_{N-2}2^{4(N-2)}$ + … + $S_1 2^4 S_1 2^4 + S_0 2^0$ $S_0 2^0$
00001, 00010	+1	
00011, 00100	+2	
00101, 00110	+3	Radix-4 $S_{N-1}2^{2(N-1)}$ $S_{N-1}2^{2(N-1)}$ + $S_{N-2}2^{2(N-2)}$ $S_{N-2}2^{2(N-2)}$ + … + $S_1 2^2 S_1 2^2 + S_0 2^0$ $S_0 2^0$
00111, 01000	+4	
01001, 01010	+5	
01011, 01100	+6	
01101, 01110	+7	Radix-8 $S_{N-1}2^{3(N-1)}$ $S_{N-1}2^{3(N-1)}$ + $S_{N-2}2^{3(N-2)}$ $S_{N-2}2^{3(N-2)}$ + … + $S_1 2^3 S_1 2^3 + S_0 2^0$ $S_0 2^0$
01111	+8	
10000	-8	
10001, 10010	-7	
10011, 10100	-6	
10101, 10110	-5	Radix-4 $S_{N-1}2^{2(N-1)}$ $S_{N-1}2^{2(N-1)}$ + $S_{N-2}2^{2(N-2)}$ $S_{N-2}2^{2(N-2)}$ + … + $S_1 2^2 S_1 2^2 + S_0 2^0$ $S_0 2^0$
10111, 11000	-4	
11001, 11010	-3	
11011, 11100	-2	Radix-16 $S_{N-1}2^{4(N-1)}$ $S_{N-1}2^{4(N-1)}$ + $S_{N-2}2^{4(N-2)}$ $S_{N-2}2^{4(N-2)}$ + … + $S_1 2^4 S_1 2^4 + S_0 2^0$ $S_0 2^0$
11101, 11110	-1	
11111	-0	

A 90 nm chip layout of proposed HMT-16 MBM created using Synopsys IC Compiler (ICC) is as shown in Figure 13. The Milkyway database for the HMT-16 32×32 bit MBM is created in ICC and a synthesised Netlist from Synopsys DC is loaded in it with constraints. After placement, building clock tree, routing and optimisation of HMT-16, it is noticed that 40% of the layout is consumed by Radix-16 MBM. The remaining 60% is consumed by Radix-4, Radix-8 and encoder selector logic. The Voltage drop (IR drop) of HMT-16 32×32 bit MBM before placement and routing is as shown in Figure 14. Basically IR drop depends on factors like driver type, loads, how often cell is switched, power supply network and voltage drop at each pin etc. It seems that at the time of design plan the maximum IR drop is due to encoder selector and Radix 16 logic.

One of the most widely used practices to decrease IR drop is to reduce peak voltage drop of power supplies and to add decoupling capacitor cells in the affected areas23. Decoupling capacitors can provide local charge storage points close to high instantaneous current sinks, thereby sinking instantaneous current needs from the supply pads. New generation VLSI systems are very sensitive due to the high integration and low global voltage. In this context noise margin is also an important factor for IR drop. The power grid which provides the supply and ground signals throughout the chips is one of the most important sources of noise. On-chip decoupling capacitors are attached to the power and ground network to decrease noise effects. After the placement and routing number of optimization iteration with decoupling capacitors has been carried out and the IR drop was reduced to 24%. The IR drop was improved after placement and routing as shown in Figure 15.

4. Conclusion

In this paper, an attempt has been made to design 32×32 bit HMT-16 MBM design. The paper provides a versatile design approach to reduce the dynamic power dissipation in a higher radix MBM by avoiding hard multiple components. The design is well-structured because the HMT-16 MBM uses less complex Radix-4 and Radix-8 MBA without an appended zero at the least significant bit. According to the experimental results, the design can obtain 25%, 21% reduction in power savings over the conventional Radix-16 and Radix-832×32 bit

Table 2. Power Report of Different MBA

Parameters	Radix-4	Radix-8	Radix-16	HMT-16
Frequency	400 MHz			
Operating Voltage	32 × 32 bit @ 1.32V			
Temperature	-40 to +55			
Library setup time	0.04 ns			
Clock Uncertainty	0.15 ns	0.15 ns	0.15 ns	0.15 ns
Total area	19987.8	21900.1	22198.4	22996.3
Total Dynamic Power (mW)	5.9408	7.0436	7.2765	5.8212
Cell Leakage Power (pW)	60.572	83.977	83.286	96.278
Date Arrival Time (ns)	2.21	2.10	1.58	0.32
Positive Slack	0.45 ns	0.36 ns	0.88 ns	2.24 ns

Table 3. HMT Comparisons with Existing Technique NA = Not Available

Design	[4]	[7]	[11]	[18]	[20]	[21]	[22]	Proposed HMT-16		
Feature (Bit)	32 × 32	16 × 16	64 × 64	16 × 16	16 × 16	32 × 32	32 × 32	32 × 32	32 × 32	64 × 64
MBM Radix	8	4	8	4	16	4	8	16	16	16
Technology	130 nm	90 nm	180 nm	250 nm	130 nm	180nm	180nm	90 nm	90 nm	90 nm
Frequency (MHz)	100	1000	NA	NA	50	NA	50	400	1000	400
Voltage (V)	NA	1.3	1.8	NA	1.8	1.98	NA	1.32	1.32	1.32
Power (μW)	12.1	9.00	NA	17.3	NA	20.06	6.35	5.821	6.312	11.13
Area (μm^2)	85983	30000	29086	33700	19995	99055.2	153606	22996	23261	27912
Delay (ns)	3	1	0.76	8.3	9.17	3.5	NA	0.32	0.34	0.58

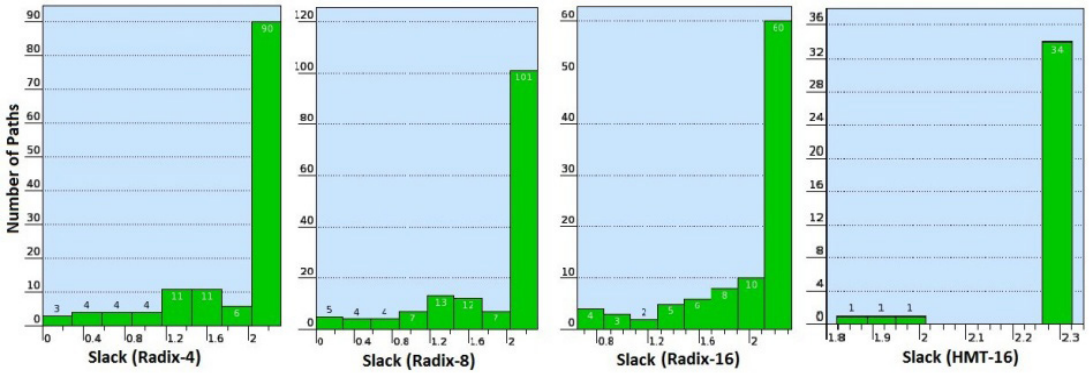

Figure 11. Path slack histogram of different encoder using 32×32 bit MBA.

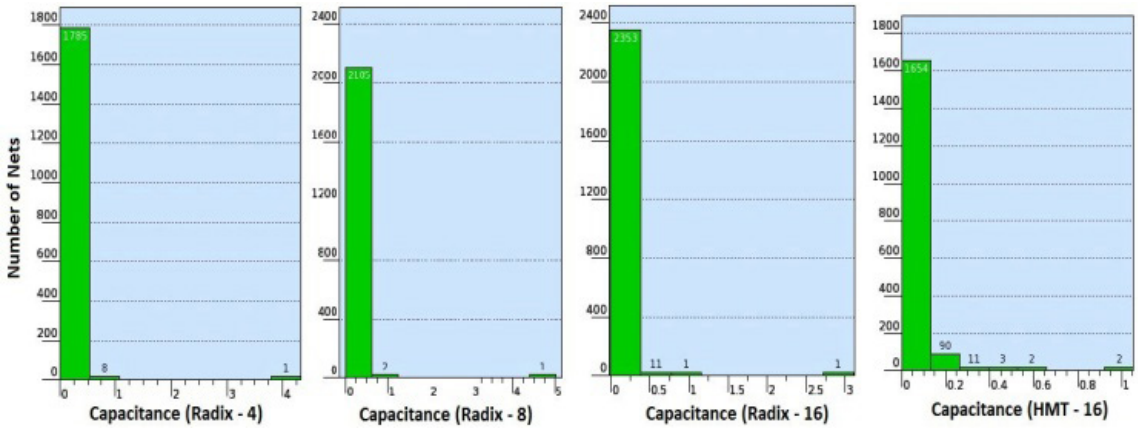

Figure 12. Nets capacitance histogram of different encoder using 32×32 bit MBA.

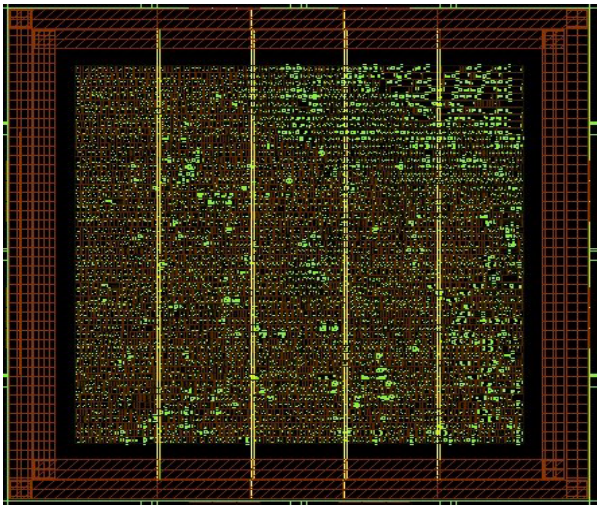

Figure 13. Chip layout of HMT equipped Radix-16 MBA.

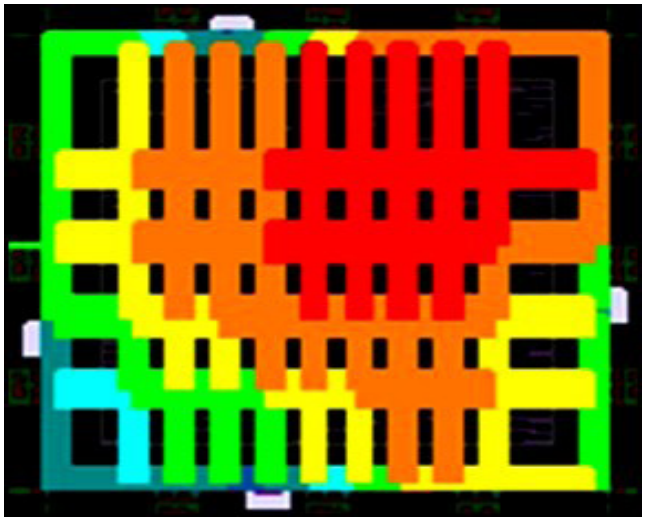

Figure 14. IR drop before placement.

MBM designs respectively. The proposed method can be used for Multimedia applications like texture coding of H.264, DSP application, Cryptographic Applications[24] and MAC unit design. The future work can be extended to Radix-64, Radix-128 and higher radix MBM with higher bit range for better power reduction and improved speed.

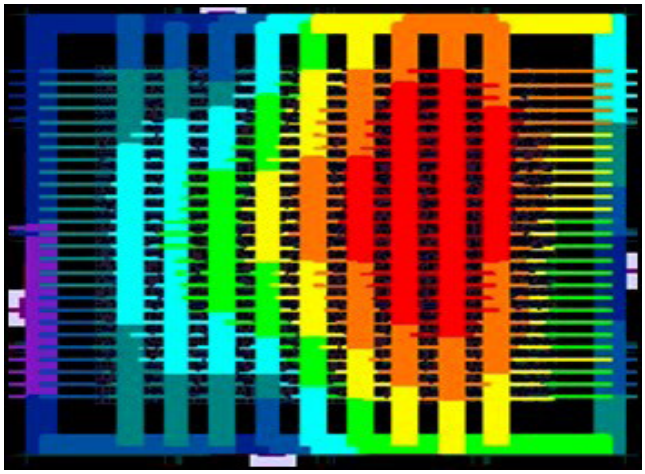

Figure 15. IR drop after placement.

5. References

1. Rubinfield LP. A proof of the modified Booth's algorithm for multiplication. IEEE Transactions on Computers.1975 Oct; 24(10):1014–15.
2. Villeger D, Oklobdzija VG. Analysis of Booth encoding efficiency in parallel multipliers using compressors for reduction of partial products. Proceedings of IEEE 27th Asilomar Conference on Computer Systems and Signals; Pacific Grove, USA. 1993 Nov. p. 781–4.
3. Villeger D, Oklobdzija VG. Evaluation of Booth encoding techniques for parallel multiplier implementation. Electronics Letters. 1993 Nov; 29(23):2016–7.
4. Wang L-R, et al. A well-structured modified Booth multiplier design. Proceedings of IEEE International Symposium on VLSI Design Automation and Test; Hsinchu, Taiwan. 2008 Apr. p. 85–8.
5. Muralidharan R, Chang CH. Fast hard multiple generators for radix-8 Booth encoded modulo 2n−1 and modulo 2n+1 multipliers. Proceedings of IEEE International Symposium on Circuits and Systems; Paris, France. 2010 Jun. p. 717–20.
6. Sangjin H, et al. Low power parallel multiplier design for DSP applications through coefficient optimization. Proceedings of 12th Annual IEEE International ASIC/SOC Conference; Washington, DC, USA. 1999 Sep. p. 286–90.
7. Hsu SK, et al. A 110 GOPS/W 16-bit multiplier and reconfigurable PLA loop in 90-nm CMOS. IEEE Journal of Solid-State Circuits. 2006 Jan; 41(1):256–54.
8. Cherkauer BS, Friedman EG. A hybrid radix-4/radix-8 low power, high speed multiplier architecture for wide bit widths. Proceedings of IEEE International Symposium on Circuits and Systems; 1996 May; Atlanta, GA, USA;P. 53–56.
9. Yeh WC, Jen CW. High-speed Booth encoded parallel multiplier design. IEEE Transactions on Computers. 2000 Jul; 49(7):692–701.
10. Wang LR, et al. A well-structured modified Booth multiplier design. Proceedings of IEEE International Symposium onVLSI Design Automation and Test; Hsinchu, Taiwan. 2008 Apr. p. 85–8.
11. Muralidharan R, Chip CH. Hard multiple generator for higher radix modulo 2n-1 multiplication. Proceedings of the 12th International Symposium on Integrated Circuits, Suntec Singapore International Convention and Exhibition Centre; Singapore, Singapore. 2009 Dec. p. 546–9.
12. Wang JP, et al. High-accuracy fixed-width modified Booth multipliers for Lossy applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2011 Jan; 19(1):52–60.
13. Pezaris SD. A 40-ns 17-bit by 17-bit array multiplier. IEEE Transactions on Computers. 1971 Mar; 20(4):442–7.
14. Min AS et al. A generalized methodology for low-error and area-time efficient fixed-width Booth multipliers. Proceedings of 47th IEEE International Midwest Symposium on Circuits and Systems; Hiroshima, Japan. 2004 Jul. p. 9–12.
15. Gary WB. Fast multiplication: algorithms and implementation [Phd Thesis]. California, United States: Stanford University; 1994.
16. Huang Z, Ercegovac MD. High-performance low-power left-to-right array multiplier design. IEEE Transactions on Computers. 2005 Mar; 54(3):272–83.
17. Roorda M. Method to reduce the sign bit extension in a multiplier that uses the modified booth algorithm. Electronic Letters. 1986 Sep; 22(20):1061–2.
18. Chen, et al. Minimization of switching activities of partial products for designing low-power multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2003 Jun; 11(3):418–33.
19. Kuang SR, Wang JP. Design of power-efficient configurable Booth multiplier. IEEE Transactions on Circuits and Systems. 2010 Mar; 57(3):568–80.
20. Chen JW Yao RH. Efficient modulo 2n+1 multipliers for diminished-1 representation. IET Circuits Devices Systems I. 2010; 4(4):291–300.
21. Vassil SD, et al. Area-efficient multipliers based on multiple-radix representations. IEEE Transactions on Computers. 2011 Feb; 60(2):189.
22. Nandal A, Vigneswaran T, Rana AK. Booth multiplier using reversible logic with low power and reduced logical complexity. Indian Journal of Science and Technology. 2014 Apr; 7(4):525–9.
23. Meng X, et al. A novel active decoupling capacitor design in 90nm CMOS. Proceedings of IEEE International Symposium on Circuits and Systems; Washington, USA. 2007 May. p. 657–60.
24. Vignesh B, Sridhar KP, Parameshwaran R. Pipelined quadratic equation based novel multiplication method for cryptographic applications. Indian Journal of Science and Technology. Apr 2014; 7(4S):34–9.