

# 4x4 Bit Multiplier Designs using Different CMOS Schematics, and their Comparison

K.rana<sup>1</sup>, A.Niaz<sup>2</sup>, S.Hanif<sup>3</sup>, M.T.Ali<sup>4</sup>

<sup>2</sup> Chung-Ang University, Seoul, Republic of Korea  
<sup>1,3,4</sup> Research Associate, HITEC University, Taxila, Pakistan

<sup>3</sup> kayynat@gmail.com

**Abstract-** In this paper, low power and high speed 4x4 bit multipliers are presented. The full adder and a half adder blocks used in these multipliers are designed using adiabatic and transmission gate techniques respectively. The multiplier circuit is implemented using Dadda algorithm. This circuit is simulated in 1P-9M Low-K UMC 90nm CMMOS process technology (cadence Virtuoso). The circuit operates at clock frequency of 5.46 and 8.54 GHz and dynamic average power of 2.667 and 1.139 mW respectively, at room temperature of 27°C and 1.9V supply voltage.

**Keywords-** Dadda Tree Reduction Algorithm, Adiabatic logic, Transmission gate logic, 4x4 bit Multiplier

## I. INTRODUCTION

Operation of multiplication is widely used in all fields, e.g. digital signal processing, digital image processing and digital communication applications. Multipliers having low power delay products (PDPs) are very useful for these applications. To reduce PDP of multiplier, many approaches are used. To reduce PDP and to increase speed of the multiplier, different logics of full adder are implemented, as full adder is a main building block of the multiplier. Different existing algorithms for implementation of 4x4 bit multiplier for the optimization of delay and power are compared e.g. Dadda, Wallace, Vedic, Booth.

In our proposed design, we implemented 4x4 bit multiplier using CMOS Schematics analysis on the basis of PDP (Power delay product). Nemours multipliers have been used but power delay and area of these multipliers are moderately outsized, ur approach proposed power efficient and high performance 4x4 bit multiplier using two circuits designs. The first proposed design is Adiabatic logic based full adder and the second proposed design is transmission gate logic based full adder. By using these two logics, a reduced PDP will be obtained. Moreover, paper is described as follows. Some previously implemented multipliers are

presented in section # 02, the proposed design is presented in section # 03, results and conclusion is discussed in section # 04 and section # 05, respectively.

## II. LITERATURE REVIEW

For the sake of improvement in PDP, different logic designs are presented. PTL is very good choice for the digital circuits, so that the circuit draw low power [1]. There are different types of PTL circuits [2, 3]. These circuits draw a very small node capacitance that increases speed of the circuit, due to less number of transistors. If this logic is being used in the multiplier circuit, throughput will be increased [4]. As multipliers are based on full adder blocks, give reduced delay and the static power, hence provides energy efficient performances [1, 5]. Multipliers in which reversible logic is used will have reduced power [6, 7, 8]. An intermediate stage i.e. partial product generator is used for efficient operation. In order to reduce the partial products, Booth encoder and modified Booth encoder are used [9]. In the circuits, where speed is not an important issue (i.e. at various places/ applications where speed of the circuit does not matter, e.g. universities labs), sequential circuits are implemented. In recent past, many parallel and sequential circuits for multipliers are implemented [10]. If an asynchronous (clock independent) circuit has to be implemented, Null Conventional Logic (NCL) is used, and to increase its performance a circuit is proposed in [11]. Pass transistor logic (PTL) based multipliers gives a lower PDP [12]. An energy-delay efficient multiplier is also implemented [13]. Data driven dynamic sum logic (D3L) and reduced-split precharge-data driven dynamic sum logic (rsp-D3Lsum) adders are also presented, which gives good performance without additional power dissipation [14, 12]. Another multiplier in which row passing technique is used is also presented i.e. Bypass Multiplier, gives reduced power dissipation [15]. Vedic multiplication is another technique which gives low power dissipation as well as high speed [16]. Another multiplier technique reduces

complexity in the design as well as it also reduces area and power [17].

### III. PROPOSED APPROACH

In this paper, two possible designs for 4x4 bit multipliers are proposed. The first one is adiabatic CMOS logic based multiplier, in which adiabatic CMOS logic based Half Adder and Full adder, using Dadda tree reduction algorithm implements the multiplier. This approach will decrease the delay and hence overall PDP will also be reduced. The second implementation of the multiplier is using TG CMOS logic, in which Full Adder and Half Adder are implemented using TG CMOS logic, and through Dadda tree reduction algorithm multiplier is implemented. Using TG based Multipliers PDP will be decreased more. At the end of the paper, many of the multipliers w.r.t to their PDPs is compared. A brief review of Dadda tree reduction algorithm is presented below [12].

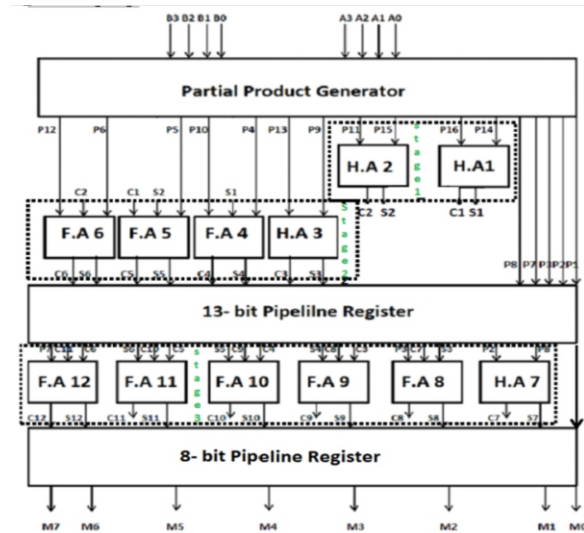


Figure 1 Block Diagram of Dadda Tree Algorithm based Multiplier

#### 3.1. Dadda Tree Reduction Algorithm

Using Dadda tree reduction algorithm, stages of the multiplier are reduced; hence the delay is also reduced. For 4X4 bit multiplication, 16 numbers of partial products are produced, which reduces the height of the tree from 4 to 2.

For every stage of addition (i.e. stage 1<sup>st</sup> to 2<sup>nd</sup> and then 2<sup>nd</sup> to 3<sup>rd</sup>) result cannot be evaluated until the previous output carry is calculated. This carry propagation will cause a large delay. To solve this problem, Dadda tree reduction algorithm is used. The first step is the rearrangement of the partial products to make a tree. The height of the tree is N i.e. 4 for 4x4 bit multiplier.

##### 3.1.1. Stages of Dadda Tree Reduction Algorithm

By the use of Dadda tree reduction algorithm, the height of the tree will be reduced from 4 to 2. But there is a limit for Dadda algorithm, i.e. in each stage, maximum reduction is 1.5 times. The height will be 3 at the 2<sup>nd</sup> last stage i.e. ( $2 \times 1.5 = 3$ ) and for the 3<sup>rd</sup> last stage, the height will be 4 i.e. ( $3 \times 1.5 = 4.5$  round off to 4). Hence, for 4x4 bit multiplier, tree height for each stage will be;

- i. 1<sup>st</sup> stage height=4
- ii. 2<sup>nd</sup> stage height=3
- iii. 3<sup>rd</sup> stage height=2

For 1<sup>st</sup> stage, height of 4 can be reduced to 3 by use of HA or FA. Only 4<sup>th</sup> column's height is 4, so it is reduced to 3 by using a HA. Output carry will propagate from 4<sup>th</sup> column to 3<sup>rd</sup> (L.H.S), i.e. the height of 3<sup>rd</sup> stage will be 4. For the reduction of 3<sup>rd</sup> column's height another HA will be required. In the same way, height of first stage will be reduced. Both HAs of 1<sup>st</sup> stage are working parallel, so they both don't wait for each other's results. Number of FAs will be  $N^2 - 4N + 3$  and the number of Has will be  $N - 1$ .

##### 3.1.1.1. Pipeline Stage 1

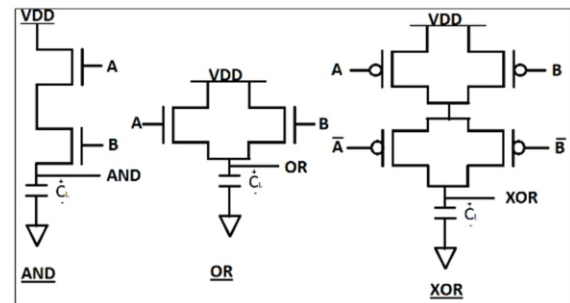


Figure 2 Comparison of different Gates

The 1<sup>st</sup> pipeline stage contains following steps;

- i. Generation of 16 partial products.
- ii. 1<sup>st</sup> Dadda stage contains 2 Has.
- iii. 2<sup>nd</sup> Dadda stage contains 1 HA and 3 FAs.
- iv. 13 bit pipeline register is used to store the results of 1<sup>st</sup> pipeline stage.

##### 3.1.1.2. Pipeline Stage 2

This stage contains following steps;

- i. 13 bit pipeline register data is read.
  - ii. 1 HA and 5 FAs are used in this stage.
- The result is stored in 8 bit pipeline register.

#### 3.2. Partial Product Generator

Partial product generator block generates partial products. AND gate is used to generate these partial products.  $N^2$  Number of partial products will be generated, so for 4x4 bit multiplier, 16 numbers of partial products are generated. So 16 numbers of AND gates will be required for this purpose. Power consumption, Delay and PDP of AND gate from the approaches is given below:

AND Gate Types	Delay (ns)	Frequency (GHz)	Power ( $\mu$ W)	PDP (fWs)
Rsp-D3Lsum [1]	0.1	10	1120	112
Adiabatic CMOS Logic	0.1	10	159	15.9
TG CMOS Logic	0.05	20	3.02	0.151

Table 1 PDP Comparison of different Schematics

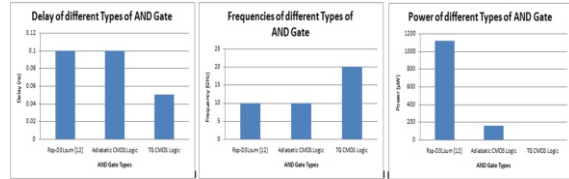


Figure 3 Comparison of Different AND Gates

Three techniques Multiplier, in which Adiabatic CMOS logic is used, would allow transfer of charge without significant heat losses [18].

### 3.2.1. Full Adder

Equations for full adder are;

$$Sum = A \oplus B \oplus C \text{ Eq.1}$$

$$Cout = A.B + B.C + C.A = A.B + C(A \oplus B) \text{ Eq.2}$$

Full Adder is implemented using two techniques i.e. Adiabatic CMOS Logic and Transmission Gate Logic.

#### 3.2.1.1. Adiabatic CMOS Logic

Using this logic, a capacitor is added to the drain of the transistors. As only one capacitor is added that is why the delay caused by this one capacitor is less than delay of pull down network). In this way, the power dissipation will be reduced and PDP will also be reduced vice versa. The gate designs and equations for this approach are given below:

$f=10\text{GHz}$	$V_{DD}=1.9\text{V}$	$C_L=278\text{yFs}$
$V_T=26\text{mV}$	$n_p=1.62$	$K_p=1.25\text{pA}$

Table 2 Parameter Values

$$C_L = \sqrt{\frac{K_p^2 n_p^2 V_T}{2 f^2 V_{DD}^3}} \left(1 - \exp\left(-\frac{V_{DD}}{2 n_p V_T}\right)\right) \left(1 + \frac{V_{DD}}{2 n_p V_T}\right) \text{ Eq.3}$$

Using above formula,  $C_L$  can be calculated [18]. The parameters required for this formula are mentioned in Table 2.

Calculations for PDP for the proposed design which is shown in Figure 5. So for its critical path, delay will be given is following cases:

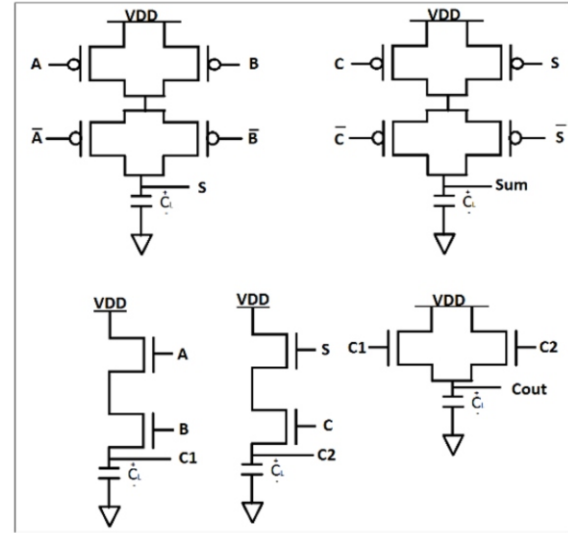


Figure 5 Full Adder Design for Adiabatic CMOS Logic

#### i. CASE I

$A=0; B=1; C=1$

In this case, for sum, A and  $\bar{B}$  are ON, that is why  $S=1$ .  $\bar{C} = 0$  and  $\bar{S} = 0$ , so that  $Sum=0$ . In this case, 4 PMOS transistors are ON, from which two are in parallel, it means, 3 PMOS transistors will be considered while calculating the delay. For carry out; B, S and C are ON, so, C1 is off and C2 is ON, so,  $Cout=1$ . It means, 4 NMOS are ON and none of them are parallel. So the total delay will be dependent upon 3 PMOS and 4 NMOS transistors.

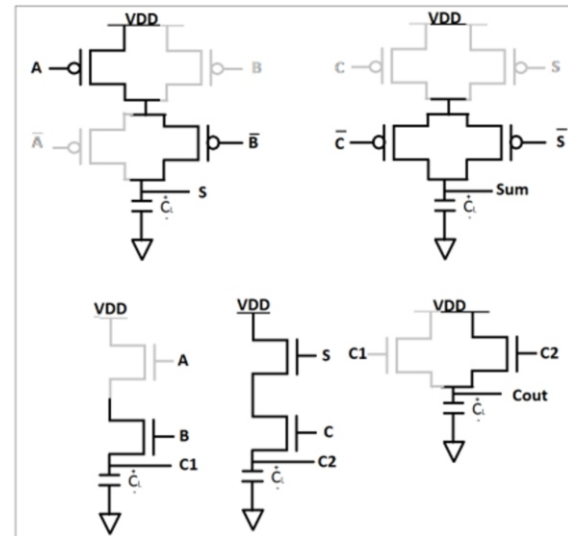


Figure 4 Case I Schematics

#### ii. CASE II

$A=1; B=0; C=1$  In this case, for sum, B and A are ON, that is why  $S=1$ .  $C = 0$  and  $\bar{S} = 0$ , so that  $Sum=0$ . In this case, 4 PMOS transistors are ON, from which two are in

parallel, it means, 3 PMOS transistors will be considered while calculating the delay. For carry out; A, S and C are ON, so, C1 is off and C2 is ON, so, Cout=1. It means, 4 NMOS are ON and none of them are parallel. So the total delay will be dependent upon 3 PMOS and 4 NMOS transistors.

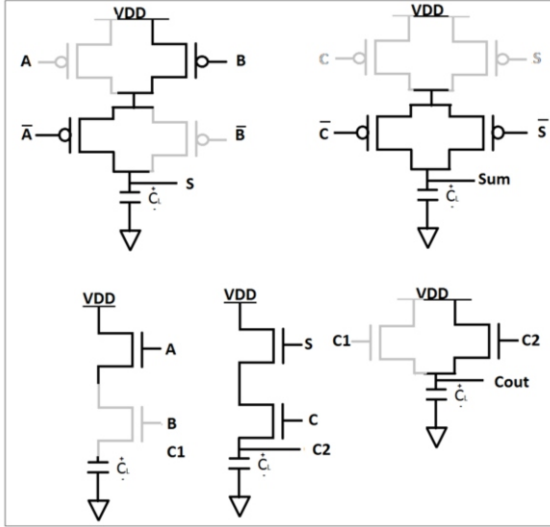


Figure 6 Case II Schematics

$$t_d = 0.7 R_{tot} C_{ox} \quad \text{Eq.4}$$

$$R_p = 2R_n \quad \text{Eq.5}$$

$$R_{tot} = 3R_p + 4R_n = 6R_p + 4R_n = 10R_n \quad \text{Eq.6}$$

Using Eq.6 and Eq.5 in Eq.4

$$t_d = 7 R_n C_{ox} = 2.6775 \text{asec}$$

Where  $R_n = 34k/W$  and  $C_{ox} = 62.5aF.WL$ .  $L=200\text{nm}$  and  $W=300\text{nm}$  are chosen for the design.

$$t_{pHL} = 0.7 R_n C_{tot} ; \quad \text{Eq.7}$$

$$t_{pLH} = 0.7 R_p C_{tot} = 1.4 R_n C_{tot} \quad \text{Eq.8}$$

$$C_{tot} = C_{ox} + C_L = C_L \text{ if } C_{ox} \ll C_L$$

$$t_{pHL} = 0.7 R_n C_L = 33.08ps$$

$$t_{pLH} = 1.4 R_n C_L = 66.16ps$$

$$t_{HL} = \frac{2.2}{0.7} t_{pHL} = 103.87ps$$

$$t_{LH} = \frac{2.2}{0.7} t_{pLH} = 207.74ps$$

Simulations results are shown in the Table 3.

Quantities	FA (Adiabatic)
Settling Time (ps)	126
Frequency (GHz)	7.93
Power Dissipation (Static) A=0000; B=0000 (μW)	0.2712
Power Dissipation (Static) A=1111; B=1111 (μW)	0.777
Power Dissipation (Dynamic) (μW)	373.6
Power Delay Product (Dynamic Power x Settling Time) (fWs)	7.03206

Table 3 Adiabatic Simulation Results

### 3.2.1.2 Transmission Gate Logic

Logic gates which are used to make a full adder are made by transmission gate logic. The designs for the gates are given in Figure 7.

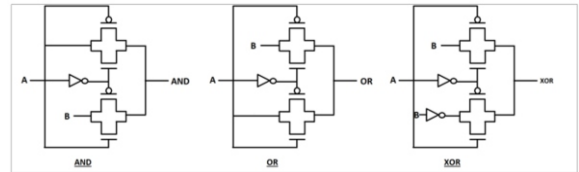


Figure 7 Transmission Gate Approach for Gates

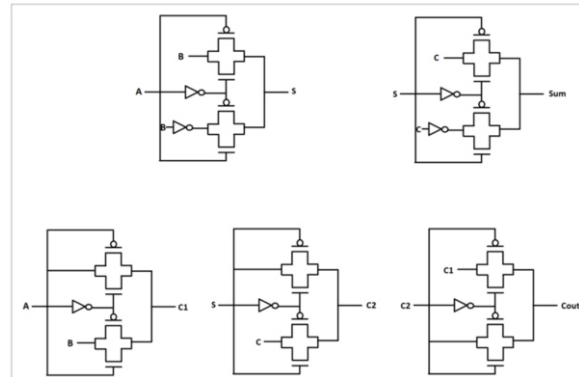


Figure 8 Full Adder Design for Transmission Gate Logic

Here's the PDP calculation for the proposed design in Figure 7.

### I. CASE I

A=0; B=0; C=0

In this case, for sum, as for A=0; S=B=0 (XOR gate).

As  $S=0$  so,  $\text{Sum}=C=0$  (XOR gate). For this case, 2 TGs are ON. For Carry out, as  $A=0$  so,  $C1=A=0$  (AND gate) and  $S=0$  so,  $C2=S=0$  (AND gate). In the last block of Cout, as  $C2=0$  so,  $\text{Cout}=C1=0$  (OR gate). For Cout case, 3 TGs are ON. Total TGs are 5 but two of them (1<sup>st</sup> TG of Sum and 1<sup>st</sup> TG of Cout) are working in parallel, it means 4 TGs are working for this circuitry. Except TGs, there are, 5 inverters are also working, inverters for A, B and C work in parallel manner and inverters for C2 and S. So, 3 numbers of inverters are working in this circuitry. It means, as a result delay is dependent upon 4TGs and 3 inverters' delays.

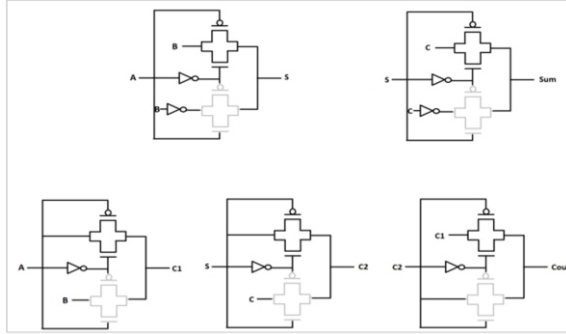


Figure 9 Case I Schematics

### ii. CASE II

$A=0$ ;  $B=0$ ;  $C=1$

In this case, for sum, as for  $A=0$ ;  $S=B=0$  (XOR gate). As  $S=0$  so,  $\text{Sum}=C=1$  (XOR gate). For this case, 2 TGs are ON. For Carry out, as  $A=0$  so,  $C1=A=0$  (AND gate) and  $S=0$  so,  $C2=S=0$  (AND gate). In the last block of Cout, as  $C2=0$  so,  $\text{Cout}=C1=0$  (OR gate). For Cout case, 3 TGs are ON. Total TGs are 5 but two of them (1<sup>st</sup> TG of Sum and 1<sup>st</sup> TG of Cout) are working in parallel, it means 4 TGs are working for this circuitry. Except TGs, there are, 5 inverters are also working, inverters for A, B and C work in parallel manner and inverters for C2 and S. So, 3 numbers of inverters are working in this circuitry. It means, as a result delay is dependent upon 4TGs and 3 inverters' delays.

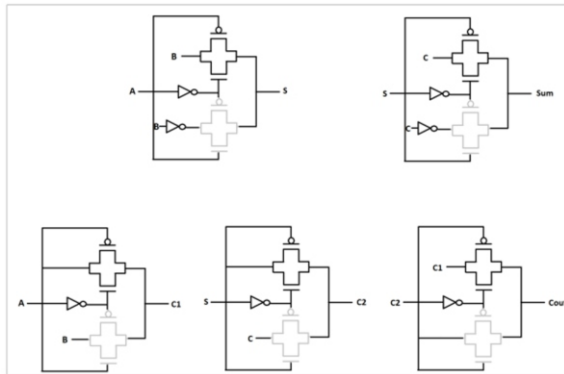


Figure 10 Case II Schematics

### iii. CASE III

$A=1$ ;  $B=1$ ;  $C=0$

In this case, for sum, as for  $A=1$ ;  $S=B=0$  (XOR gate). As  $S=0$  so,  $\text{Sum}=C=0$  (XOR gate). For this case, 2 TGs are ON. For Carry out, as  $A=1$  so,  $C1=B=1$  (AND gate) and  $S=0$  so,  $C2=S=0$  (AND gate). In the last block of Cout, as  $C2=0$  so,  $\text{Cout}=C1=1$  (OR gate). For Cout case, 3 TGs are ON. Total TGs are 5 but two of them (1<sup>st</sup> TG of Sum and 1<sup>st</sup> TG of Cout) are working in parallel, it means 4 TGs are working for this circuitry. Except TGs, there are, 5 inverters are also working, inverters for A, B and C work in parallel manner and inverters for C2 and S. So, 3 numbers of inverters are working in this circuitry. It means, as a result delay is dependent upon 4TGs and 3 inverters' delays.

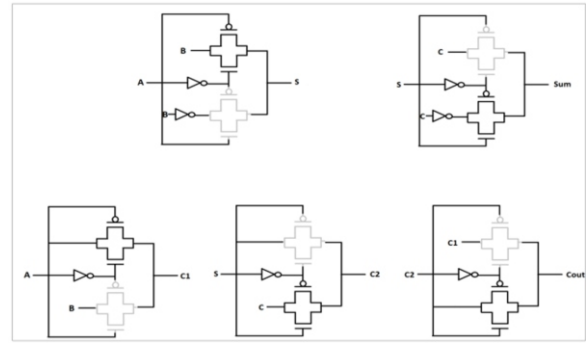


Figure 11 Case III Schematics

For transmission gate, delay is;

$$t_{dTG} = 0.7 (R_p || R_n) C_{ox} = 0.2975as \text{ Eq.9}$$

For inverter, delay is;

$$t_{pLH(inverter)} = \frac{0.7}{2} (R_p + R_n) (C_{inp} + C_{inn} + C_{oxn} + C_{oxp})$$

Where,  $C_{inp} = C_{inn} = 1.5 C_{oxp} = 1.5 C_{oxn}$ ;

$$t_{pLH(inverter)} = t_{pHL(inverter)}$$

$$\&C_{oxp} = C_{oxn}$$

$$t_{pLH(inverter)} = 2.23as \text{ Eq.10}$$

By using eq.9 and eq.10, the overall delay for the critical path is

$$t_d = 4t_{dTG} + 3t_{pLH(inverter)} = 6.98as$$

Where  $R_n = 34k/W$ ,  $R_p = 68k/W$  and  $C_{ox} = 62.5aF.WL$ .  $L=200nm$  and  $W=300nm$  are chosen for the design.

Simulations results are given in Table 4.

### 3.2.2. HalfAdder

Half adder is also made from the same principle. Simulation results are as follows:



Quantities	FA (Transmission Gate)
Settling Time (ps)	69
Frequency (GHz)	14.5
Power Dissipation (Static) A=0000; B=0000 ( $\mu$ W)	2.73
Power Dissipation (Static) A=1111; B=1111 ( $\mu$ W)	2.819
Power Dissipation (Dynamic) ( $\mu$ W)	55.81
Power Delay Product (Dynamic Power x Settling Time) (fWs femto watt seconds)	3.85089

Table 4 Simulation Result for Half Adder

Quantities	HA (Adiabatic)	HA (Transmission Gate)
Settling Time (ps)	85.3	23.5
Frequency (GHz)	11.7	42.5
Power Dissipation (Static) A=0000; B=0000 ( $\mu$ W)	0.01401	1.311
Power Dissipation (Static) A=1111; B=1111 ( $\mu$ W)	0.05408	1.24
Power Dissipation (Dynamic) ( $\mu$ W)	92.4	7.793
Power Delay Product (Dynamic Power x Settling Time) (fWs)	7.88172	0.18314

Table 5 Simulation Result for Half Adder

## VI. RESULTS

This section shows the comparison of the previous approaches with the new approach to implement multipliers.

It is clear from the results that the settling time of the adiabatic and TG approach is much less than the other multipliers as shown in Figure 12.

It is obvious from the results seen in Figure 13 that the operating frequency of the adiabatic and TG approach is much less than the other multipliers.

If we compare the overall PDPs of different multipliers, it will be seen in Figure 14 that PDP of the adiabatic and TG approach is much less than the other multipliers. If we compare TG approach and adiabatic approach, it will be seen that, the PDP of TG is lesser than adiabatic logic. As in TG logic, only body of the PMOS/NMOS is connected to VDD/ground (i.e. supply voltages).

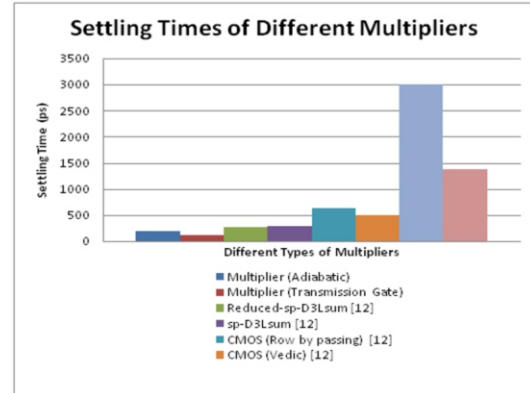


Figure 12 Settling Time Comparison

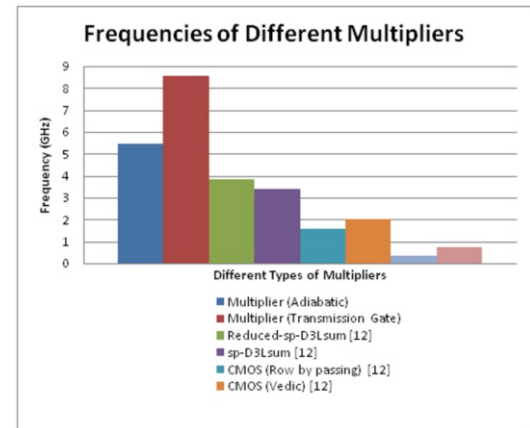


Figure 13 Frequency Comparison

But in adiabatic logic, both body and source of PMOS/NMOS are connected to VDD/ground. So, the dependency of adiabatic on supply voltage is greater than the dependency of TG on supply voltage. Due to more dependency on voltage supply, PDP of adiabatic is also greater.

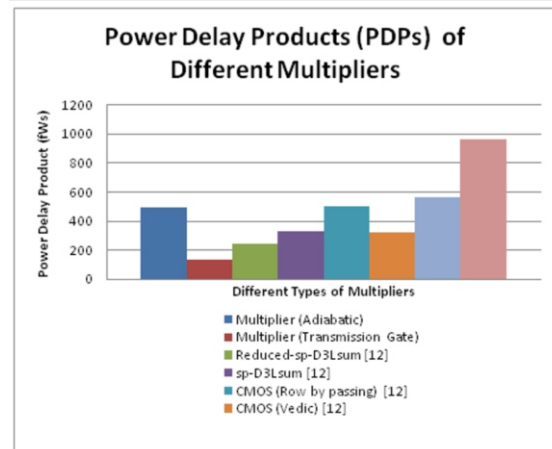


Figure 14 PDP Comparison

Comparisons of different multiplier are illustrated in table given below in Table 6.

Quantities	Multiplier (Adiabatic)	Multiplier (Transmission Gate)	Reduced- sp- D3Lsum [12]	sp- D3Lsum [12]	CMOS (Row by passing) [12]	CMOS (Vedic) [12]	CMOS (Correction Function) [12]	8-T (Array) [12]
Settling Time (ps)	183	117	262	294.2	637.9	500	3000	1380
Frequency (GHz)	5.46	8.54	3.8168	3.399	1.5676	2	0.33	0.72
Power Dissipation (Static) A=0000;B=0000 ( $\mu$ W)	121.71	26.51	59.6	1.04	79	60		
Power Dissipation (Static) A=1111; B=1111 ( $\mu$ W)	793	1072	1034	1287	1054	665		
Power Dissipation (Dynamic) ( $\mu$ W)	2667	1139	623	1100	784	642	488	697
Power Delay Product (Dynamic Power x Settling Time) (fWs)	488.061	133.263	241.826	323.62	500.114	321	564	961.86

Table 6 Comparisons of Different Multiplier

## V. CONCLUSION

High speed 4x4 bit multipliers, with two different approaches with the use of Dadda Algorithm are proposed. The first approach is implementation of Multiplier by using Adiabatic CMOS logic and the second is by using Transmission Gate Logic. Both of the multipliers are designed using 1P-9M Low-K UMC 90nm CMOS process technology. The proposed multipliers operate at frequency of 5.46 and 8.54 GHz respectively, which is improved as compared to previous approaches. The PDPs of the multipliers are reduced to 488.061 and 133.263 fWs (femto watt seconds femto= $10^{-15}$ ) respectively. These circuits work at the room temperature 27°C and 1.9V of voltage supply. As these both are very fast multipliers with very low settling times, so these can be used for most of the calculation purposes where fast calculation is required. In future we will be focusing on full pipelined structure, latency and throughput.

## REFERENCES

- [1] A. Grover, G.K. Wadhwa, N. Grover, J. Gupta, Multipliers Using Low Power Adder Cells Using 180nm Technology. in *Proceedings of ISCB1 2013*, (New Delhi, India, 2013), pp. 3–6.
- [2] N. Weste, K. Eshraghian, *Principles of CMOS Digital Design* (Pearson Education, Indiana, 2002)
- [3] R. Zimmermann, W. Fichtner, Low power logic styles: CMOS versus pass transistor logic. *IEEE J. Solid State Circuits* **32**, 1079–1090 (1997)
- [4] D. Pal, M. Chandra, M.K.Goswami, A. Saha, Novel High Speed MCML 8-Bit by 8-Bit Multiplier. In *Proceedings of ICDeCom 2011*, (Mesra, Ranchi, India, 2011), pp. 1–5 (15)
- [5] M. Linares-Aranda, M. Aguirre-Hernandez, Energy-Efficient High Speed CMOS Pipelined Multiplier. In *proceeding of CCE 2008*, (Mexico City, Mexico, 2008), pp. 460-464.9 (12)
- [6] C.H. Bennet, Logical reversibility of computation. *IBM J. Res. Dev.* **6**, 525 (1973) (5)
- [7] M.P. Frank, Reversibility for Efficient Computing. Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Jun 1999 (7)
- [8] P. Moallem, A. Vafaei, M. Ehsanpour, Design of a Novel Reversible Multiplier Circuit Using Modified Full Adder. in *Proceedings of ICCDA 2010*, (Qinhuangdao, Hebei, China, 2010), pp. V3-230–V3-234 (14)
- [9] M. Hansson, A. Alvandpour, N. Mehmood, An Energy-Efficient 32-bit Multiplier Architecture in 90-nm CMOS. in *Proceedings of Norchip 2006*, (Linköping, Sweden, 2006), pp. 35–38 (9)
- [10] M. Aktan, V.G. Oklobdzija, D. Baran, Multiplier

- Structures for Low Power Applications in Deep-CMOS, in *Proceedings of ISCAS'11*, (Rio de Janeiro, Brazil, 2011), pp. 1061–1064 (3)
- [11] M. Aktan, V.G. Oklobdzija, D. Baran, Multiplier Structures for Low Power Applications in Deep-CMOS, in *Proceedings of ISCAS'11*, (Rio de Janeiro, Brazil, 2011), pp. 1061–1064
- [12] Z. Shabbir, A.R. Ghumman, S.M. Chaudhry, r3p-D3Lsum Adder based 4x4 bit multiplier using Dadda algorithm. In *proceeding of Springer Science+ Business Media* (New York 2015)
- [13] K. Chong, T. Lin, Bah-HweeGwee, J.S. Chang, W.-G. Ho, Energy-delay efficient asynchronous logic 16×16-bit pipelined multiplier based on Sense Amplifier-Based Pass Transistor Logic. in *IEEE International Symposium on Circuits and Systems (ISCAS 2012)*, (Seoul, Korea, 2012), pp. 492–495
- [14] M. Margala, S. Purohit, Investigating the impact of logic and circuit implementation on full adder performance. *IEEE Trans. Very Large Scale Integration. (VLSI) Syst.* **20**(7), 1327–1331 (2012)
- [15] K. Kuo, C. Chou, Low power and high speed multiplier design with row bypassing and parallel architecture. *Micro electron. J.* **41**, 639–650 (2010)
- [16] Y. Bansal, C. Madhu, P. Kaur, High Speed Vedic Multiplier Designs A Review in *Proceedings of RAECS 2014*, (Chandigarh, India, 2014), pp. 1–6
- [17] S. Khan, S. Kakde, Y. Suryawanshi, VLSI Implementation of Reduced Complexity Wallace Multiplier Using Energy Efficient CMOS Full Adder. in *Proceedings of ICCIC 2013*, (Tamilnadu, India), pp. 1–4
- [18] S. Agwa, E. Yahya, Y. Ismail, Variability Mitigation Using Correction Function Technique. In *proceedings of 20<sup>th</sup> ICECS*, (Dubai, UAE, 2013)