Design and Analysis of Low Leakage 1-Bit Adder for Inexact Computing

Savinder Kaur[1], Candy Goyal[2]
[1]*Electronics and Communication Section, Yadavindra College of Engineering, Talwandi Sabo, India.*
[2]*Electronics and Communication Section, Yadavindra College of Engineering, Talwandi Sabo, India.*
[1]*Department of Electronics and Communication Engineering, Punjabi University, Patiala, India*
[2]*Department of CSE Sant Longowal Institute of Engineering & Technology, Longowal, India*
[1]savi.rai70@gmail.com
[2]candygoyal@pbi.ac.in

***Abstract*— In most of the digital applications faster speed and low power are the basic requirements. Adders are the basic elements which are widely used in digital systems. Carry select adders (CSLA) are the fastest adder used in many advanced arithmetic circuits to perform the fast calculations in arithmetic circuits. To enhance the speed and to reduce the leakage power we have used approximation technique in CSLA. This work evaluates the comparison of the conventional 1-bit full adder with proposed 1-bit approximate adder in terms of average power, PDP, delay, leakage power and number of transistors. All the Circuits are designed using Tanner EDA tool v13.0 and simulation are performed on H-Spice tool at 500MHz frequency using 45nm technology at supply voltage of 1.1V. Result shows there is improvement in delay, leakage power and number of transistors.**

***Keywords*— CSLA (carry select adders), RCA (ripple carry adder), DSP (digital signal processing), MA (mirror adder), FA (full adder), VLSI (very large scale integration), CMOS (complementary metal oxide semiconductor), PDP (power delay product).**

I. INTRODUCTION

DSP blocks are the backbone of most of the modern multimedia applications. Most of the DSP processors used the basic adder and multipliers (which is also constructed using adders) to build the basic building block of the multiplier. Low power and high speed are the imperative requirement of all digital systems. In digital VLSI systems, speed of the addition is limited by time required to propagate a carry through the adder [3]. The sum of each bit of the basic adder is made sequentially only after the previous bit position has been summed and carry propagated in the next position. The CSLA is preferred in digital VLSI systems to diminish the problem of carry propagation delay by generating the multiple carries and then choose a carry to generate the sum. Though CSLA uses large area because it uses the multiplexer and multiple blocks of RCA [10].

In CSLA, one n- bit adder is being divided into m-parts. It is made of two RCAs and one multiplexer. The addition of two n-bits is done with two ripple carry adders. RCAs provide the calculation of two numbers is twice. It assumes one time the carry input signal is zero (C_{in} = 0) and other time the carry input is 1 (C_{in} = 1). Once the right value of C_{in} is known, then the correct value of C_{out} and the sum is chosen with multiplexer. In the CSLA, the n- bit inputs a0, a1, a2 . . , an−1 and b0, b1, b2 . . . ,bn−1 are divided into sets of length m. In CSLA adder the size m of each set is 4 bits. [12]

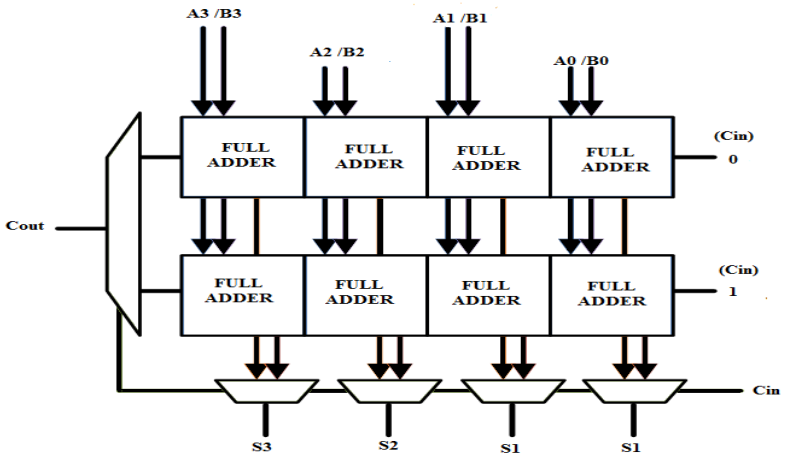

Fig.1 Block diagram of carry select adder

Fig. 1 shows the basic block diagram of CSLA, where the each block size is 4. Two 4-bit ripple carry adders perform the operation on the corresponding input applied to them and these blocks are multiplexed together. Since one RCA block assumes a carry input of 0 (C_{in} = 0) and other RCA block assumes a carry input of 1 (C_{in} = 1). To select which adder had the correct assumption is selected with multiplexer [9].

II. Conventional And Existing Full Adder Using Approximation

In this section we have discussed the various methods for designing the approximate FA cells. Since the MA is the used to implement the FA (full adder).

A. *Approximate Mirror Adder* :-

In this section, the different types of approximation are used with minimum number of transistors. Since the series connected transistors provide large delay by removing the number of transistors to provide the faster charging and discharging of node capacitances. With removal of transistors there is decrease in the dynamic power [2].

$$P_{dynamic} = \alpha C_L V^2_{dd} f \quad (1)$$

Where α is the average number of switching transitions per unit time. C_L is the load capacitance for charging and discharging. V_{dd} is supply voltage and f is frequency. This result shows the low power dissipation and another advantage is reduced area. Now, let us discuss the conventional MA implementation [2].

1.) Conventional CMOS: - Fig. 2 shows the transistor level schematic of conventional MA, it is the basic method to implement the FA. It uses total 24 transistors. This is not based on complementary CMOS logic. But it provides an opportunity to design an approximate version with removal of selected transistors [11].

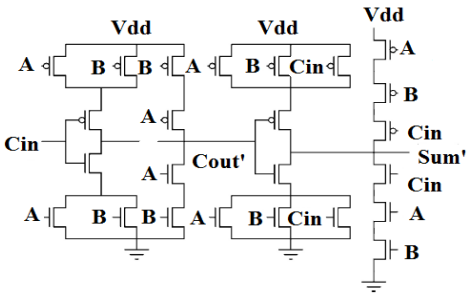

fig. 2 Conventional CMOS adder [2]

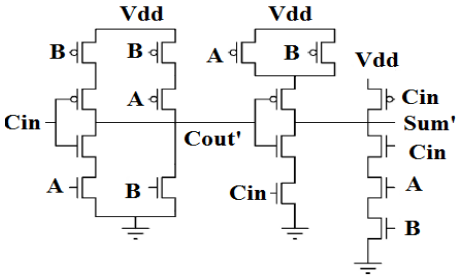

fig. 3 Approximation MA 1 [2]

2.) Approximation MA 1:-To get the approximation with less number of transistors, the removal of transistors from the conventional schematic of CMOS (fig. 2) is one by one. We have to ensure any input combination of A, B, C does not provide any short circuit or open circuit and provide the criteria that gives minimal number of errors in FA truth table. It has 8 less transistors compared to conventional MA schematic. The truth table of FA shows Sum = C_{out} for 6 cases out of 8, except the input combination of A=0, B=0, C=0 and A=1, B= 1, C=1. Table I shows output result has 1 error in C_{out} and 3 errors in Sum. The tick mark shows output of FA is match with accurate output and a cross denotes an error.

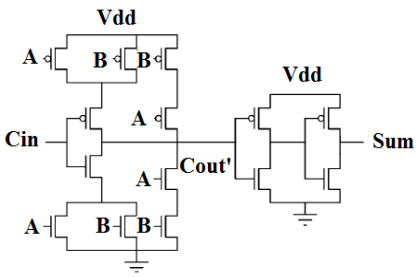

fig. 4 Approximation MA 2 [2]

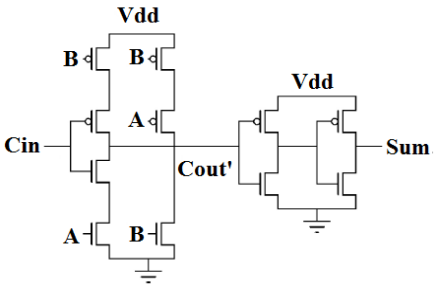

fig. 5 Approximation MA 3 [2]

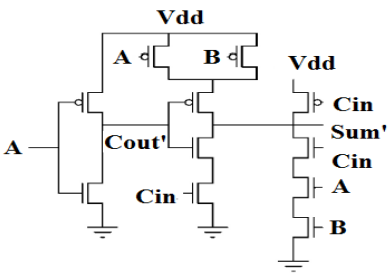

fig. 6 Approximation MA 4 [2]

3.) Approximation MA 2:- The truth table of FA shows Sum = complement C_{out1} for 6 cases out of 8, except the input combination of A=0, B=0, C=0 and A=1, B= 1, C=1. The C_{out} of conventional MA is calculated in 1st stage. It is the easy method to compute the Sum = complement of C_{out1}. However, we can introduce the buffer block after C_{out} (see in fig.4). Table I shows output result has total 2 errors, Sum has 2 errors and C_{out} has 0 errors.

4.) Approximation MA 3:- MA approximation 3 shows in fig. 5. The further calculations are done using approximation 1 and 2. Here Sum has 3 errors and C_{out} has only 1 error.

5.) *Approximation MA 4:-* The MA approximation 4 shows in fig. 6. The FA truth table shows that C_{out} = A for 4 cases out of 8 cases. Similarly the C_{out} = B for 6 cases out of 8 cases. It gives the 3 errors in Sum and 2 errors in $C_{out.}$. In all these approximations C_{out} is computed by using the inverter with complement of C_{out} as input signal.

III. 1-Bit Proposed Approximate Adder

It provides the low leakage power, delay and reducing the number of transistors by using approximation technique. In conventional approximation adder, minimum number of transistors is 15 and has very high leakage and average power. It uses only 8 numbers of transistors and gives less leakage power and delay. So this approximation can apply for CSLA [6].

G. 1-bit approximation_1:- Fig. 7 shows the 1-bit approximation_1. It is constructed of total 8 transistors. The proposed 1-bit approximation_1 shows C_1 = B for 7 cases out of 8 cases, except A = 1, B = 0, C =1. Table I shows the output has total 4 errors, 1 error in C_{out} and 3 errors in Sum.

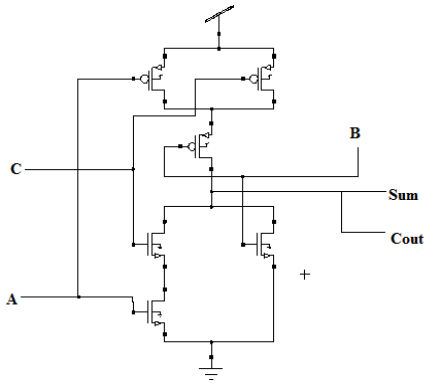

fig. 7 1-bit approximation_1

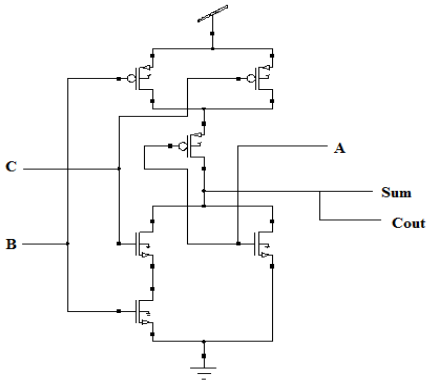

fig. 8 1-bit approximation_2

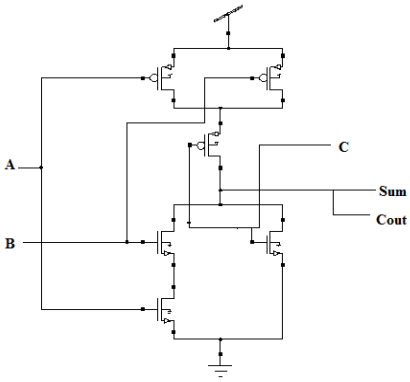

fig. 9 1-bit approximation_3

H. 1-bit approximation_2:- Fig. 8 shows the 1-bit approximation_2. The proposed 1-bit approximation_2 shows C_2= A for 7 cases out of 8 cases, except A =0, B =1, C =1. Table I shows it has total 4 errors, 1 error in C_{out} and 3 errors in Sum.

I. 1-bit approximation_3:- Fig. 9 shows the 1-bit approximation_3. The proposed 1-bit approximation_3 shows C_3 = C for 7 cases out of 8 cases, except A =1, B =1, C =0. Table I shows it has total 4 errors, 1 error in C_{out} and 3 errors in Sum.

TABLE I: *TRUTH TABLE FOR EXISTING AND PROPOSED APPROXIMATION ADDER*

Inputs			Accurate outputs		Existing 1-bit approximate adder outputs								Proposed 1-bit approximate adder outputs					
			MA adder		Approxi-mation 1		Approxi-mation 2		Approxi-mation 3		Approxi-mation 4		Approxi-mation_1		Approxi-mation_2		Approxi-mation_3	
C	A	B	Sum	C_{out}	S1	C1	S2	C2	S3	C3	S4	C4	S1	C1	S2	C2	S3	C3
0	0	0	0	0	0 √	0 √	1×	0 √	1 ×	0 √	0 √	0 √	1 √	0 √	1 ×	0 √	1 ×	0 √
0	0	1	1	0	1 √	0 √	1 √	0 √	1 √	0 √	1 √	0 √	0 ×	1 ×	1 √	0 √	1 √	0 √
0	1	0	1	0	0×	1×	1√	0 √	1 √	1 ×	0 ×	0 √	1 √	0 √	0 ×	1 ×	1 √	0 √
0	1	1	0	1	0 √	1 √	0 √	1 √	0 √	1 √	1 ×	0×	0 ×	1 √	0 √	1 √	0√	1 √
1	0	0	1	0	0×	0 √	1 √	0 √	0 √	0 √	0 ×	1 ×	1 √	0 √	1 √	0 √	0 ×	1 ×
1	0	1	0	1	0 √	1 √	0 √	1 √	1 ×	1 √	0 √	1 √	0 √	1 √	0 √	1 √	0 √	1 √
1	1	0	0	1	0 √	1 √	0 √	1 √	0 √	1 √	0 √	1 √	0 √	1 √	0 √	1 √	0√	1 √
1	1	1	1	1	1 √	1 √	0×	1 √	0 ×	1 √	1 √	1 √	0 ×	1 √	0 ×	1 √	0 ×	1 √
Total number of errors in output			0		3		2		4		5		4		4		4	

IV. Simulation Results

The conventional and proposed approximation adder design is compared in terms of leakage power, delay, PDP and number of transistors. The proposed approximation technique is having lesser number of transistors and lesser leakage power. All the simulations are performed at 45nm technology using 1.1V at 500MHz frequency. All the schematic are designed using Tanner

tool v13.0 and the simulation waveforms are on H-spice tool. The approximation technique is applied to reduce the number of transistors and PDP. Table I shows the result of C_{out} and Sum. Table I shows the result of C_{out} and Sum. The output result of conventional approximate adder and proposed approximate adders are shown in fig 10(a)-10(d).

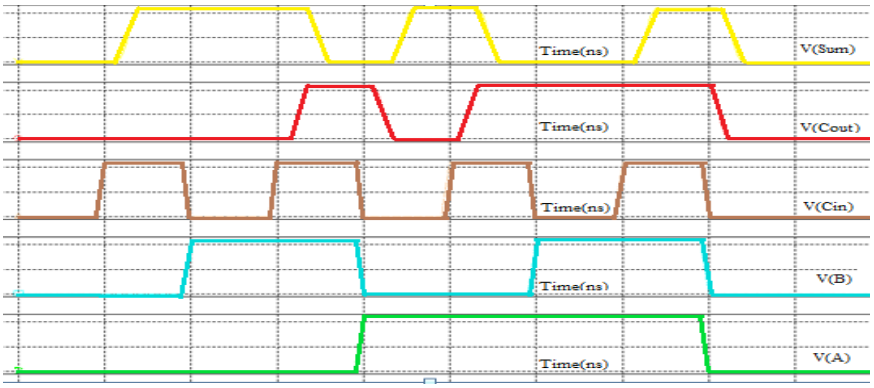

fig. 10(a) Simulation waveform of 1-bit conventional CMOS adder [2]

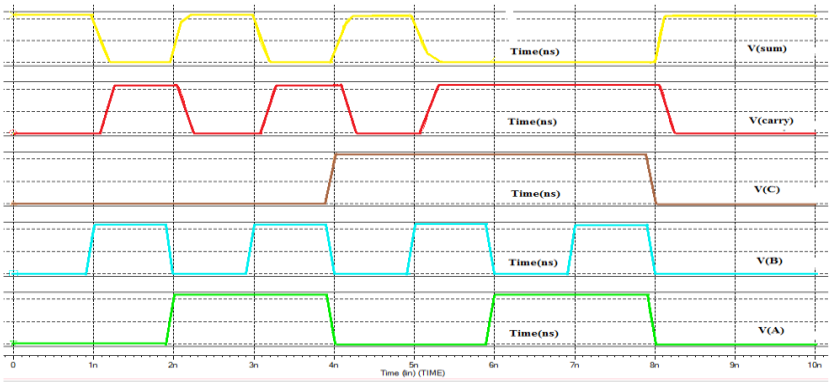

fig. 10(b) Proposed 1-bit approximation_1

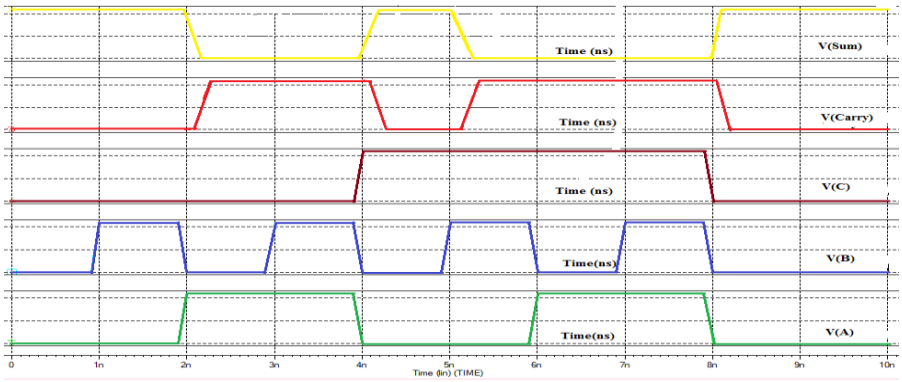

fig. 10(c) Proposed 1-bit approximation_2

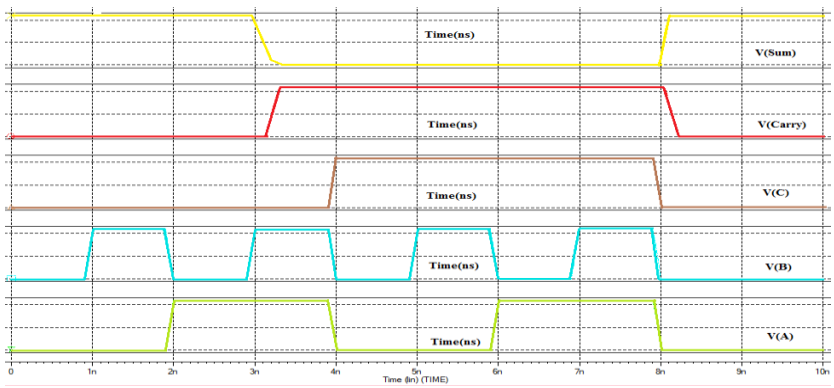

fig. 10(d) Proposed 1-bit approximation_3

TABLE II: TRANSISTOR SIZE OF EXISTING AND PROPOSED APPROXIMATE ADDER

	Name of the approximation adder	No. of transistors used	No.of approxi-mation used
Existing approximate adder [11]	Conventional CMOS	28	0
	MA approximation 1	24	3
	MA approximation 2	16	2
	MA approximation 3	13	4
	MA approximation 4	15	5
Proposed approximate adder	1-bit approximation_1	8	4
	1-bit approximation_2	8	4
	1-bit approximation_3	8	4

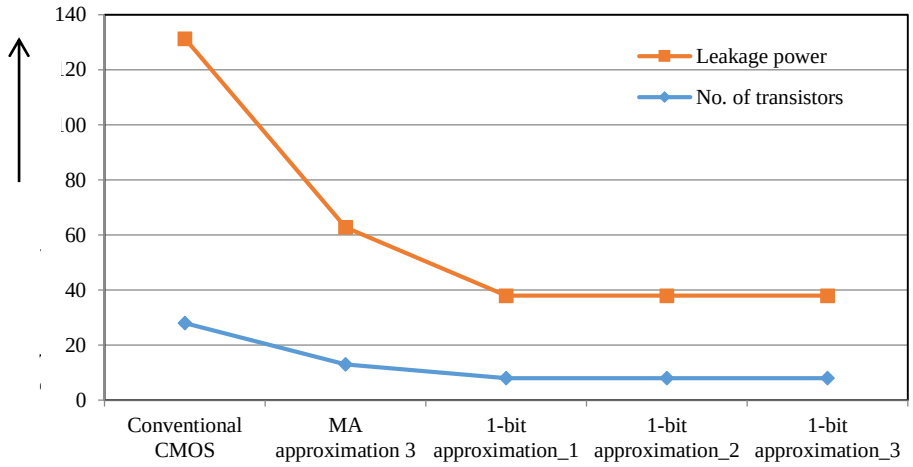

fig. 11 Comparison of number of transistor and leakage power of 1-bit conventional approximate adder and proposed approximate adder.

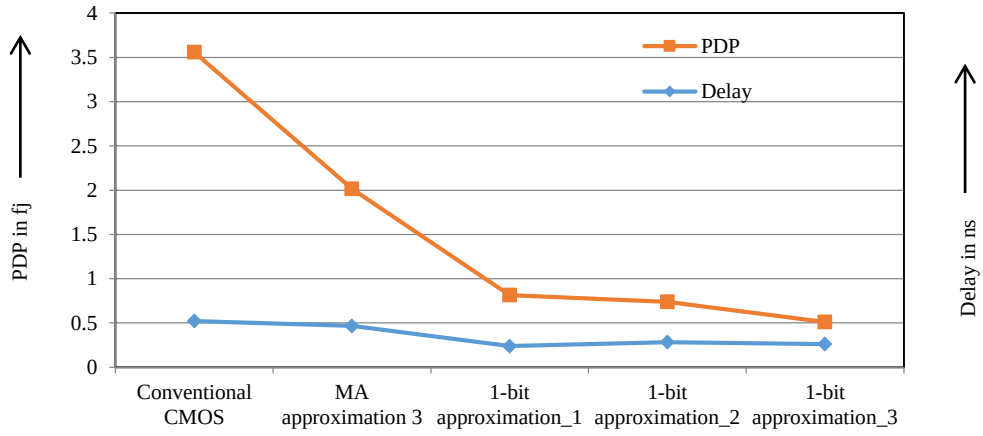

fig. 12 Comparison of delay and PDP of 1-bit conventional approximate adder and proposed approximate adder.

Fig. 11 shows that the graph between number of transistors and leakage power of conventional approximate and proposed approximate adder. The proposed adder design uses only 8 numbers of transistors and it has minimum 29.95 pw leakage power. Fig. 12 shows the graph between delay and PDP of conventional approximate and proposed approximate adder. It has minimum

delay is 0.239 ns and power delay product is 0.250 fj using 1.1V supply voltage. These graphs show, the proposed adder is better than the existing approximation adder.

TABLE III: COMPARISON OF 1-BIT EXISTING AND PROPOSED APPROXIMATE ADDER.

	Name of adder	No. of transistor used	Average power (µw)	Leakage power (pw)	Peak power (µw)	Delay (ns)	PDP (fj)
Existing approximate adder [2]	Conventional CMOS	28	5.797	103.28	84.98	0.523	3.036
	MA approximation 1	24	4.394	78.55	77.95	0.282	1.241
	MA approximation 2	16	3.494	73.45	51.26	0.316	1.107
	MA approximation 3	13	3.312	49.75	43.38	0.467	1.549
	MA approximation 4	15	3.350	50.95	51.59	0.416	1.396
Proposed approximate adder	1-bit approximation_1	8	2.411	29.95	38.92	0.239	0.576
	1-bit approximation_2	8	1.603	29.95	35.30	0.284	0.455
	1-bit approximation_3	8	0.960	29.95	35.32	0.261	0.250

V. CONCLUSION

Three 1-bit approximate adders at 45nm technology are designed in this paper. To optimize the leakage power and delay approximation is used. It reduces the number of transistors. Result shows that proposed adder has 39.79% improvement in leakage power, 42.02% improvement in PDP and 15.24% improvement in delay as compared to existing approximation adder. So we can use proposed 1-bit approximate adder in CSLA (carry select adder) for advanced arithmetic applications.

REFERENCES

[1] Z. Nan and W. Haomin, "A new design of the CMOS full adder," *IEEE J. Solid-State Circuits*, vol. 27, no. 5, pp. 840–844, 1992.

[2] V. Gupta, "Low-Power Digital Signal Processing Using Approximate Adders," *... -Aided Des. ...*, vol. 32, no. 1, pp. 124–137, 2013.

[3] A. Sudsakorn, S. Tooprakai, and K. Dejhan, "Low Power CMOS Full Adder Cells," no. 2, pp. 12–15, 2012.

[4] M. Agarwal, "A New Design of Low Power High Speed Hybrid CMOS Full Adder," pp. 448–452, 2014.

[5] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate XOR/XNOR-based adders for inexact computing," *Proc. IEEE Conf. Nanotechnol.*, pp. 690–693, 2013.

[6] M. B. Damle, S. S. Limaye, and M. G. Sonwani, "Comparative Analysis of Different Types of Full Adder Circuits," vol. 11, no. 3, pp. 1–9, 2013.

[7] A. T. Tran and B. M. Baas, "Design of an energy-efficient 32-bit adder operating at subthreshold voltages in 45-nm CMOS," *ICCE 2010 - 3rd Int. Conf. Commun. Electron.*, no. August, pp. 87–91, 2010.

[8] Z. Yang, J. Han, and F. Lombardi, "Transmission gate-based approximate adders for inexact computing," *Proc. 2015 IEEE/ACM Int. Symp. Nanoscale Archit. NANOARCH 2015*, vol. 0, pp. 145–150, 2015.

[9] A. Chakraborty and S. N. Pradhan, "International Journal of Electronics A new design of adder circuit for low power and high speed in sub-threshold region," no. December 2014, pp. 37–41, 2013.

[10] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 20, no. 2, pp. 371–375, 2012.

[11] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," *Proc. Int. Symp. Low Power Electron. Des.*, pp. 409–414, 2011.