Shodhasamhita : Journal of Fundamental & Comparative Research
Vol. VIII, No. 1(X) : 2021
ISSN: 2277-7067

A SUGGESTED DESIGN OF XOR HYBRID ADDER USING EDA TOOLS WITH LOW POWER AND LOW AREA

[1]P.Sravan Kumar Reddy, [2]G.Surendra Babu, [3]T.Rajitha, [4]R. Prapulla Kumar
[1,2,3]Dept of Electronics and Communication Engineering, Sree Venkateswara College Of Engineering, Nellore (Dt), Andhra Pradesh, India.
[4]Dept of Computer Science and Engineering, Sree Venkateswara College Of Engineering, Nellore (Dt), Andhra Pradesh, India.

ABSTRACT
To execute computer arithmetic, Adder has applications in graphics processing units, micro controllers, and digital calculators. The downside with the ripple carry adder is that there will be a greater delay but a smaller area since only after the preceding carry is known can the sum and carry be computed. We now need to decrease both the Area and the power. We designed the circuit at the transistor level, and both the sum and carry outputs underwent thorough delay analyses. The Tanner EDA tool has been utilized in this study to develop the suggested adder employing hybrid logic 0.25um technology. There were fewer transistors (6N-2). As compared to the existing design, the proposed concept reduced power usage by around 28.6%. Power delay product for about 16%wassaved for the proposed design when compared to the existing design.
Keywords: Hybrid logic, power, Area, adder

1. INTRODUCTION

A total of 26 transistors were used in the design of the adder, of which 12 were required for execution and 14 for the sum output [1]. It has been claimed that the adder circuit, which consists of 26 transistors without gates at the sum and carry outputs, performs the best of all the proposed circuits [2]. The adder was designed using a hybrid logic style where a combination of logic types may be employed to produce minimal power and great performance [3]. Due to the necessity for complete swing in CMOS to obtain large noise margins for logic 0 and logic 1, adders with just 14 transistors were proposed [4]. Energy delay product is a significant issue in VLSI [5], which is why adder was introduced for applications requiring energy-efficient arithmetic. Adder was proposed with low power making sure fewer transistors will turn on while the circuit has a 0-1 transition [6]. Adder was proposed for tree structures with 0.180 um implementation technology and their performances were compared [7]. Adder was proposed with low voltage high performance, as if the supply voltage is less then performance degrades but using hybrid structure low delay was possible with low voltage [8].New design methodologies were proposed for high-speed with low-voltage for 1 bit CMOS Full Adder circuits [9]. A high speed 8 transistor full adder design was proposed using unique 3 transistor XOR gates [10]. Hybrid adder has been designed with full noise margins at every internal and external nodes of design as in [11].Adder was designed using CMOS logic with 26number of devices as in[12].Full adder logic based comparator is designed as in [13].Now we need to design a adder by implementing our own. technique such that functional behavior of the circuit should be correct but off course by keeping in mind the design constraints like area as well as delay as these are our concern. We have designed adder by using hybrid logic style.

HFA17T Adder[2]

Initially , Xor logic is designed and then used not gate to design Xnor logic which is the prime requirement to design carry and sum parts of logic. For this nine transistors are required and eight more are required to pass the logic required for functional correctness of the two outputs. It takes three levels to produce the output. Four transistors will be on at any point of time in the third level giving rise to activity factor of 50%. It took for about seven transistors to design xor logic with full output voltage swing. Since xnor/xor are not

parallelly designed, delay will be more in this design. Power delay product is also more in this design as reported in the literature. This design successfully got operated till 8-bit addition but failed to operate from 16-bit and so on due to lack of buffer sat the output nodes of sum and carry.

HFA20T Adder [2]

Parallelly, Xor and Xnor logic is designed and then these are used to design the logic for sum and carry outputs. For this, twelve transistors are required and eight more transistors are required to pass the logic which is needed for logical correctness of the two outputs. Four transistors will be on at any point of time in the second level giving rise to activity factor of 50%. Delay will be less in this design because it requires only two levels to generate the output with first level needs Xnor/Xor and the second level needs pass transistors. This design successfully got operated till 8-bit addition but failed to operate from 16-bit and so on due to lack of buffers at the output nodes of sum and carry.

HFA–B-26TAdder[2]

Parallelly, Xor and Xnor logic is designed and then these are used to design the logic for sum and carry outputs.For this, twelve transistors are required and fourteen more transistors are required to transmit the logic by using transmission gates which is needed for logical correctness of the two outputs. Four transistors will be on at any point of time in the second level giving rise to activity factor of 50%. Power consumption is more as there are more number of transistors in the circuit. This design successfully got operated till 64-bit addition due to insertion of inverting buffers at the output nodes of both sum and carry outputs . It has given the best results as reported in the literature.

HFA–NB-26T Adder [2]

Parallelly, Xor and Xnor logic is designed and then these are used to design the logic for sum and carry outputs. For this, twelve transistors are required and fourteen more transistors are required to transmit the logic by using transmission gates which is needed for logical correctness of the two outputs. Four transistors will be on at any point of time in the second level giving rise to activity factor of 50%. Power consumption is more as there are more number of transistors in the circuit. This design successfully got operated till 64-bit addition due to insertion of inverting buffers at the input nodes of all the transmission gates. It has given the better results as reported in the literature.

HFA19T ADDER [2]

Initially , Xor logic is designed and then used not gate to design Xnor logic which is the prime requirement to design carry and sum parts of logic. For this nine transistors are required and two more are required to pass the logic required using pass transistors and six more for transmitting the logic using transmission gates. And two more for complemented carry input to acheive functional correctness of the two outputs. It takes three levels to produce the output. Four transistors will be on at any point of time in the third level giving rise to activity factorof 50%. It took for about seven transistors to design xor logic with full output voltage swing. Since xnor/xor are not parallelly designed, delay will be more in this design. Power delay product is also more in this design as reported in the literature. This design successfully got operated till 16-bit addition but failed to operate from 32-bitandsoondue to lack of buffers at the output nodes of sum and carry.

HYBRID ADDER

Adder is going to add 4 bits of A(A3 to A0) and 4 bits of B(B3 to B0) and gives 5-bit result out of those 4-bitswill be sum bits and 1-bit will be carry bit. we can design the circuit for adder which by default gives high speed but at the cost of Area. Now let us see how can we design the circuit for 4-bit adder coming from the least significant bit side that is A0 and B0, there will be Carry in. Now just a XOR circuit is needed to design sum output and two copies will be used to do so, odd input XOR is logically equalto odd input XNOR. To design carry output, as the logic of carry says if A and B are high then output is„ 1‟, also when cinis„1‟andxor is „1‟ then also the output is „1‟. This will set up a pull down network arrangement since all the input conditions are set to„1‟and finally a inverter is needed to get the value of „1‟.

XOR has the output„0‟ for two input combinations and the output„1‟ for two input combinations. In order

for the outputs to operate fully, two individual PMOS and two individual NMOS transistors must drive them. Coming to the carry portion, inputs are logic "1" and output should likewise be logic "0," therefore there is no contradiction because NMOS can drive solid logic 0.

When inputs are logic 0, outputs should likewise be logic 0, which creates a contradiction that forces us to utilise a not gate in order to obtain a full swing. In a hybrid design, solid logic levels might not come from power rails but rather from the input, ensuring that logic levels 1 and 0 are driven by PMOS and NMOS, respectively. As XOR has equal logic 0 and logic 1 at the output, it should have two PMOS and two NMOS when designing the sum output.

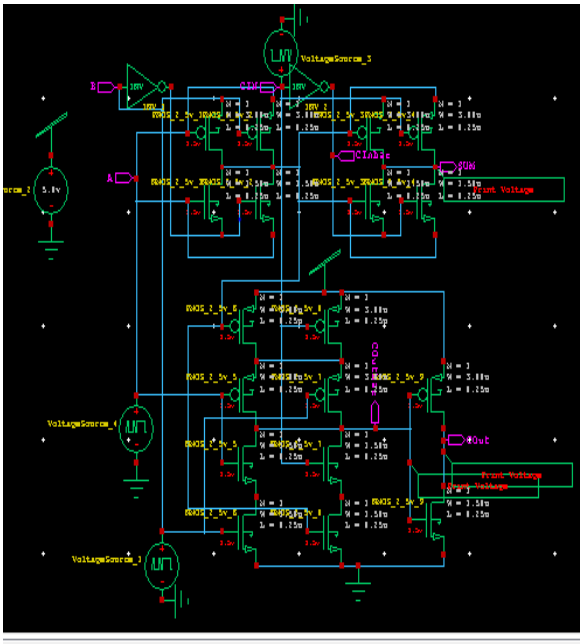

Figure1 Circuit for full adder

Figure 1 shows a full adder Circuit which has three inputs and two outputs out of which 12 transistors are needed for sum output and 10 transistors are needed for carry output. Coming to the sum output , there are four paths to provide the output for XOR and the same cell is replicated to design sum output. Coming to the carry output, pull down network has two paths and pull-up network has four paths in the first level and a not gate is needed to generate the output in the second level.

Critical path for carry output needs four transistors. Critical path for sum needs three transistors. Delay can be expected to be more for the carry output due to more number of transistors in the critical path. Total transistor count is 22 for the least significant stage and20 for the rest of the stages.

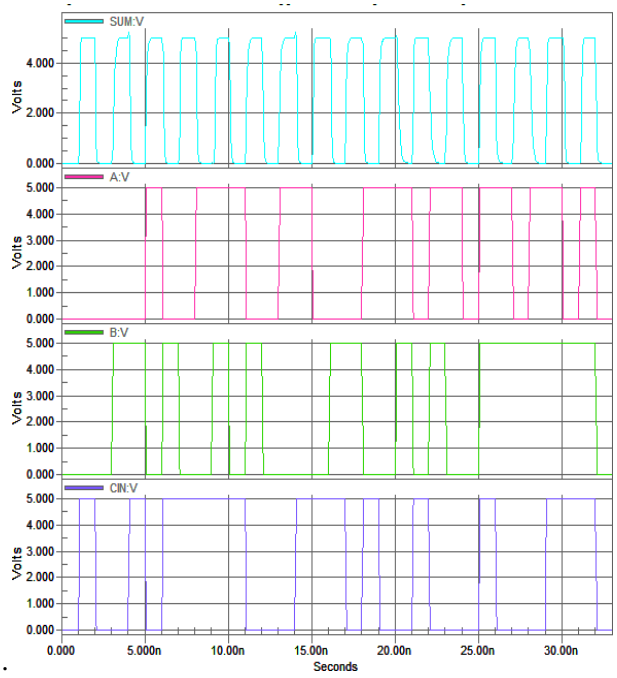

Figure 2 Delays for sum outputs

Figure 2 shows a sum out waveform for all possible events which can occur with three primary inputs and that events purely depends on the truth table of sum output, since to calculate delay we need both logic"0" andlogic"1" and the propagation delay is measured as input changing 50% to the output changing 50% of theoperatingvoltage.Everypossibleeventofsumoutputthedelayismeasuredandcomparedwiththeconventionalhybridadder.

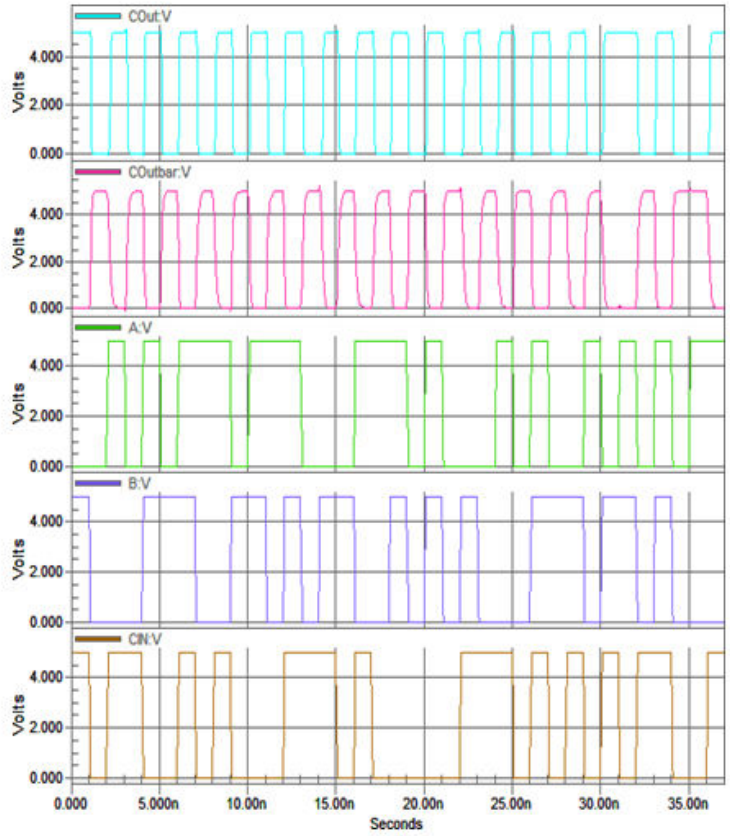

Figure.3 Delays for carry outputs

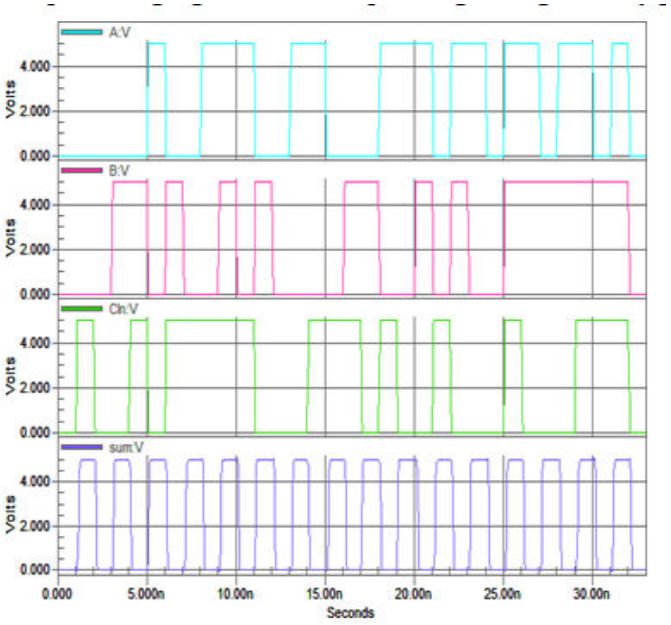

Figure 3 shows a carry out and complemented carry out waveform for all possible events which can occur with three primary inputs and that events purely depends on the truth table carry output, since to calculate delay we need both logic"0" and logic"1" and the propagation delay is measured as input changing 50% to the output changing 50% of the operating voltage. Every possible event of carry output the delay is measured and compared with the conventional hybrid adder.

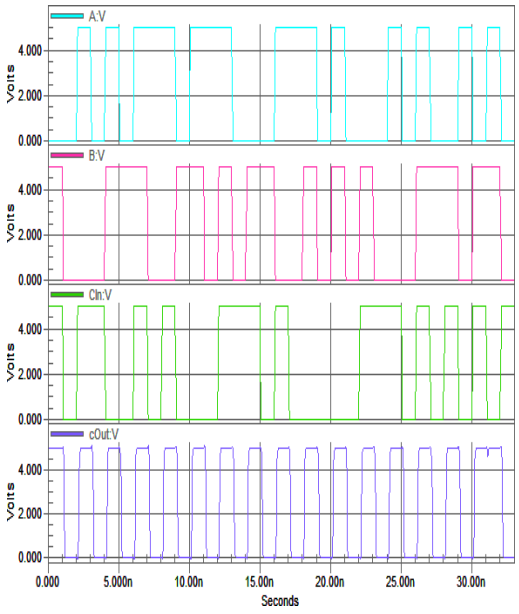

Figure 4 Delays for carry outputs

Figure 4 depicts a carry out waveform for all events that can happen with three main inputs and that solely depend on the truth table carry output because logic 0 and logic 1 are required to calculate delay and the propagation delay is calculated as the operational voltage shifting 50% from the input to the output. Every potential carry output event's delay is measured, and it is then contrasted with the suggested hybrid adder.

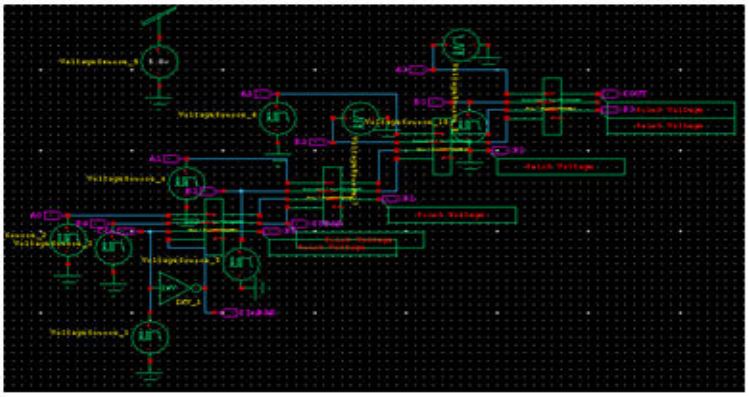

Figure.5 Block diagram for 4-bit full adder

Figure 5 shows the 4-bit adder block diagram, where four1-bit adders were connected in cascade and it produces four carry outputs.

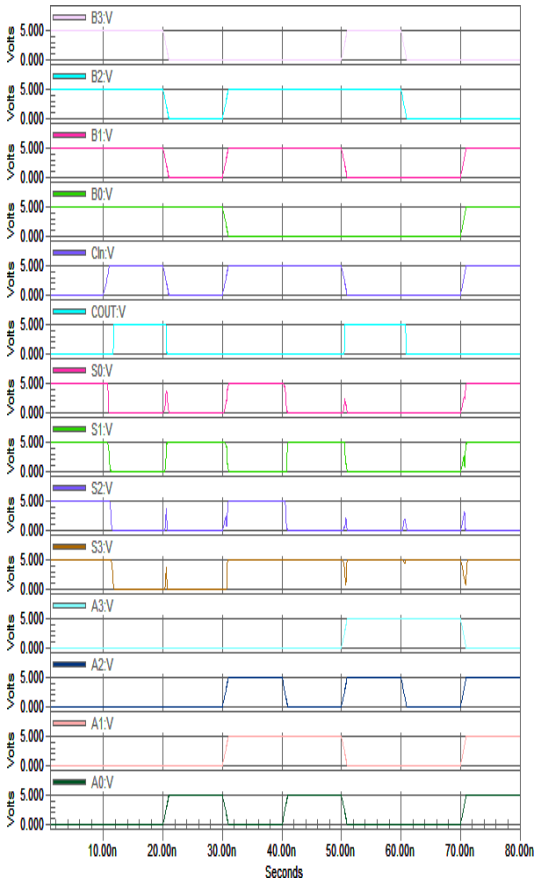

Figure.6 Output wave form for 4-bit full adder

Figure 6 shows the 4-bit adder wave form, where the delay is maximum for carry out as expected. Test vectors are applied such that it will activate the critical path of the adder. Circuit run time is for 80ns with rise , fall time as 1n sand pulse wid this 10ns.

PERFORMANCE ANALYSIS

Implementation of adder has been done using tanner tools with technology 250nm with supply voltage of 5V.

Table I Delay analysis for count

s.no	Input sequence	HFA-B-26T[2](NS) count	Proposed design-22T (NS) count
1	3-1	0.142	0.156
2	5-1	0.162	0.157

3	6-4	0.186	0.103
4	7-0	0.163	0.150
5	7-4	0.150	0.114
6	0-3	0.150	0.219
7	0-5	0.151	0.172
8	2-6	0.146	0.096
9	0-7	0.142	0.080

When the input combination is 011 and the prior input is 000, the delay is 0.219 and it is the critical path, according to Table I, which displays the delays for randomly applied input sequences of cout. Coming back to the current design, the delay is 0.186 ns and it is the critical route when the input combination is 100 and the prior input is 110.Therefore, 33 ps delay is preferable for traditional over proposed.

TABLEII DELAYANALYSISFORSUM

S.No	**Input sequence**	**HFA –B-26T[2] (NS) (Sum)**	**Proposed design(NS) (Sum)**
1	0-4	0.135	0.036
2	0-7	0.137	0.042
3	3-1	0.150	0.066
4	6-2	0.125	0.087
5	6-7	0.137	0.050
6	2-0	0.145	0.043
7	7-3	0.151	0.129
8	7-5	0.118	0.093
9	2-6	0.177	0.124
10	4-6	0.177	0.176

When the input combination is 110 and the preceding input is 100, Table II shows the delays for randomly applied input sequences of sum, and it is stated that for the suggested design, the delay is 0.176 ns and it is on the critical path. Regarding the current architecture, the critical path has a delay of 0.177 ns when the input combination is 110 and the prior input was 010 or 100. Therefore, compared to the standard design, our proposed architecture performs better with a 1 ps delay.

TABLE III POWERANALYSIS

Run time	**8NS**	**33NS**	**37NS**
Proposed design	1.705982mw	2.024553mw	2.424798mw
Existing design[2] HFA –B-26T	2.453412mw	3.161704mw	3.543291mw

Table III displays the power consumption figures for various circuit run times. Run time is 8ns when inputs are supplied as numbers between 0 and 7.Runtime of 33 ns is required to activate all potential events for the sum output. The carry output will take 37ns to activate all conceivable events. According to reports, the proposed architecture outperforms the conventional one in every scenario.

TABLE IV POWER DELAY PRODUCT ANALYSIS

DESIGN	DELAY	POWER	POWER DELAY PRODUCT
Proposed design	219ps	2.5mw	547.5
Existing design[2] HFA –B-26T	186ps	3.5mw	651

Table IV compares the delay, power, and power delay products, and the proposed design performs better when it comes to the power and power delay products than the typical design for delays. The power results are based on table 4, where the average power results for the proposed and existing ones were taken. The results of the delay are based on tables 2 and 3, and the maximum value is taken into account for both the existing and planned delays for the sum and carry.

CONCLUSION

Hybrid adder is extensively used where quick processing is needed especially in graphical processing unit. Performance analysis was done for all transitions which were events for sum and carry in any condition which is possible. For this appropriate test vectors are generated to simulate the events. Proposed Hybrid logic has proved that it outperforms the conventional adder with respect to the average power consumption for different run times. In the similar fashion for N-bit the advantage can be considered. Proposed adder was designed with only 20 number of transistors , but LSB stage needs 22 number of transistors and this makes the total transistor count to 22+ (N-1)*20 where for the conventional adder it is 26*N. So in total (6N-2) number of transistors was reduced. Delay was better in the conventional design and it is 15% better than proposed design. When compared to the old design, the proposed design reduced power consumption by around 28.6%.When compared to the old design, the proposed design reduced power delay by around 16%.

REFERENCES

1. N. S. Kim et al., “Leakage current: Moore’s law meets static power, “Computer, vol. 36, no. 12, pp. 68–75, Dec. 2003.
2. 2. N. H. E. Waste and D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Boston, MA, USA: Addison-Wesley, 2010.
3. 3. S. Goel, A. Kumar, and M. Bayoumi, “Design of robust, energy-efficient full adders for deep-sub micrometer design using hybrid-CMOS logic style,” IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 14, no. 12, pp. 1309–1321, Dec. 2006.
4. 4. H. T. Bui, Y. Wang, and Y. Jiang, “Design and analysis of low-power10-transistor full adders using novel XOR-XNOR gates,” IEEE Trans.Circuits Syst. II, Analog Digit. Signal Process. vol. 49, no. 1, pp. 25–30, Jan. 2002.
5. 5. S. Timarchi and K. Navi, “Arithmetic circuits of redundant SUT-RNS,” IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 2959–2968, Sep. 2009.
6. 6. J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
7. 7. D. Radhakrishnan, “Low-voltage low-power CMOS full adder,” IEEProc. -Circuits, Devices Syst., vol. 148, no. 1, pp. 19–24, Feb. 2001.
8. 8. K. Yano, A. Shimizu, T. Nishida, M. Saito, and K. Shimohigashi, “A 3.8-ns CMOS 16×16-b multiplier using complementary pass-transistor logic,” IEEE J. Solid-State Circuit’s vol 25, no. 2, pp. 388–395, Apr. 1990.
9. 9. A. M. Shams, T. K. Darwish, and M. A. Bayoumi, “Performance analysis of low-power 1-bit CMOS

full adder cells," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 10, no. 1, pp. 20–29, Feb. 2002.

10.10.N. Zhuang and H. Wu, "A new design of the CMOS full adder," IEEEJ. Solid-State Circuits, vol. 27, no. 5, pp. 840–844, May 1992.

11.11. N. Weste and K. Eshraghian, Principles of CMOS VLSI Design. New York, NY, USA: Addison-Wesley, 1985.

12.12.P. Bhattacharyya, B. Kundu, S. Ghosh, V. Kumar, and A. Dandapat, "Performance analysis of a low-power high-speed hybrid 1-bit full adder circuit," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 23, no. 10, pp. 2001–2008, Oct. 2015.

13.13.M. Vesterbacka, "A 14-transistor CMOS full adder with full voltage-swing nodes," in Proc. IEEE Workshop Signal Process. Syst. (Sips), Oct. 1999, pp. 713–722.

14.14.H. Naseri and S. Timarchi, "Low-Power and Fast Full Adder by Exploring New XOR and XNOR Gates," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 26, no. 8, pp. 1481-1493, Aug. 2018.