A Structured Approach for Optimizing 4-Bit Carry-Lookahead Adder

Wei Cheng[1,*] and jianping Hu[2]

[1]*Institute of Micro-Nano Electronic Systems, Ningbo University, Zhejiang 315211, China*

[2]*Faculty of Information Science and Technology, Ningbo University, Zhejiang 315211, China*

Abstract: This paper presents a comparative research of low-power and high-speed 4-bit full adder circuits. The representative adders used are a ripple carry adder (RCA) and a carry-lookahead adder (CLA). We also design a proposed carry-lookahead adder (PCLA) using a new method that uses NAND gate for modification which helps in reducing the power-delay product (PDP) for high performance applications. To yield more realistic rise and fall times in the simulations, layouts have been made in a 0.13 µm process for the RCA circuit, CLA circuit and PCLA circuit. The layouts designed were simulated by HSPICE based on 130 nm CMOS technology at 1.2 V supply voltages. Four sets of frequencies were operated: 10 MHz, 50 MHz, 100 MHz and 500 MHz with 50% duty cycle in different technology corner models. A comprehensive comparison and analysis were also carried out to test the performance of the adders. The three adders also yield different performances in terms of power consumption, PDP, and area. The simulation results of this research are expected to help designers to select the appropriate 4-bit adder cell that meets their specific applications.

Keywords: Carry-lookahead adder (CLA), low power adder, power-delay product, ripple-carry adder (RCA).

1. INTRODUCTION

The performances of the very large scale integration (VLSI) systems determine the quality of electronic equipment. Now, there is an ever-increasing number of mobile electronic devices such as mobile handsets and all tablet computers and embedded devices, the VLSI circuits low power design is becoming increasingly important.

Addition is a basic arithmetical operation in almost all of the equipment, and optimizing the efficiency of addition is a constantly attractive research topic. Propagation delay, power consumption and power-delay product (PDP) are the significant quality measure parameters for most of full adder systems, and the full adder would affect the system. This paper considers three types of 4-bit adders, a ripple carry adder (RCA), a carry-lookahead adder (CLA), and a proposed carry-lookahead adder (PCLA). With the development of CMOS technology and improvement of the feature size, low-power circuit imposes strict restrictions on power consumption while still demanding high operation speeds as required by real system. So we take the power consumption and power-delay product (PDP) as the primary standard in this paper and downplay the importance of the area occupied in the circuit.

In order to improve its PDP, the designer may make some of the tradeoffs through the circuit design styles, architecture and algorithm optimized of the adder. There are

conventional implementations with different circuit styles that have been used in the past to design full-adder cells [1] and are used for comparison in this paper. Although, they all have similar logic expression, the way of producing the carry and the propagation delay are completely different. The circuit Structure and arithmetic of adders basically influence the speed, power dissipation, and PDP. The RCA uses one logic style for the circuit while CLA uses more than one logic style for its implementation. The CLA significantly reduces the propagation delay, and PCLA more apparent low. The aim of this study is to analyze the performance of different kinds of adders, what's more, according to the optimization of the algorithm, design a proposed carry-lookahead adder (PCLA) based on the inverse logic.

The remainder of this paper is organized as follows. In section 2 presents the corresponding model description and principle analysis, the layout generation of the adder circuits, respectively. We analyze the full adder cells post-layout simulations in section 3 and compare these adder cells based on speed, power consumption, power delay product, and area. Finally, in section 4 performance comparisons and conclusions are presented.

2. MODEL DESCRIPTION AND PRINCIPLE ANALYSIS

In recent years, different forms of circuits have been proposed to implement adder cells [2-4]. Now we choose adders with a wide spectrum of structure and complexity which make it meaningful to compare their performance in terms of propagation delay, power consumption, and PDP. The adders range from the simple but redundancy (linear time) ripple

*Address correspondence to this author at the Institute of Micro-Nano Electronic Systems, Ningbo University, Zhejiang 315211, China;
Tel: 008613685817208; Fax: 86-574-87600945;
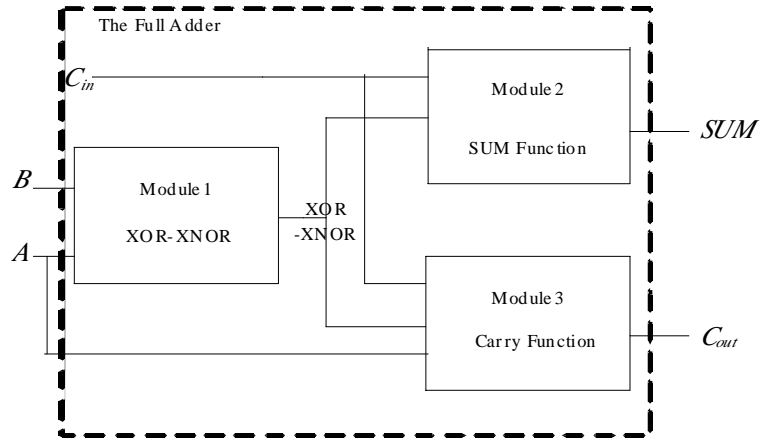E-mails: chengwei20071118@163.com, 875039437@qq.com

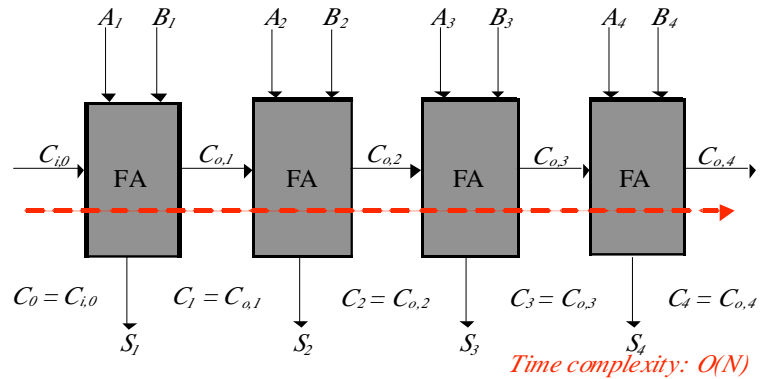**Fig. (1).** Building module of full adder cell.

Fig. (2). 4-bit ripple-carry adder.

carry adder to the fairly complex but extremely fast ($O(\log_2 N)$ time) blocked carry-lookahead adder [5]. Although they all have the same function, their critical path of carry chain and the time complexity are different, the loads on the outputs and intermediate nodes are different, and the transistor count significantly different.

2.1. Ripple-Carry Adder

The simplest way to implement the full-adder circuit is to take the logic equation and translate them directly into circuit. The typical full-adder function can be described as follows.

$$Sum = A \oplus B \oplus C_{in} \tag{1}$$

$$C_{out} = A(\underline{A \oplus B}) + C_{in}(A \oplus B) \tag{2}$$

There are standard implementations for the full-adder cells which are used for the fundamental unit of a RCA. We may take these adders into consideration [6-8]. From the equation (**1-2**) we can know most adders' logic expression are based on two XOR circuits: one to generate H (XOR) and $\overline{H}$ (XNOR), and the other to generate the Sum output function. The C_{in} is not only used to carry-in bit but have the effect of the multiplexer.

Let's rewriting equation as

$$H = A \oplus B \tag{3}$$

$$Sum = H \oplus C_{in} \tag{4}$$

$$C_{out} = A\,\overline{H} + C_{in}H \tag{5}$$

From the equation (**3-5**), it's clear that if we optimize the generation of H and $\overline{H}$, this can significant enhance the performance of the full adder cell. A block diagram of the full adder cell and its building block is shown in Fig. (**1**) [9].

As mentioned above, a full adder (FA) is the critical component unit of RCA which generates a sum bit and a carry bit. The 4-bit RCA is shown in Fig. (**2**). In Fig. (**2**) the FA is a conventional complementary CMOS structure that is realized using 28 MOSFETs. The complementary CMOS logic circuit has the advantage of layout regularity and stability at low power. The adder operations at full voltage swing.

Just as the Fig. (**2**) shows, a building block 4-bit ripple-carry adder can be constructed by cascading 4 full-adder circuits in series.

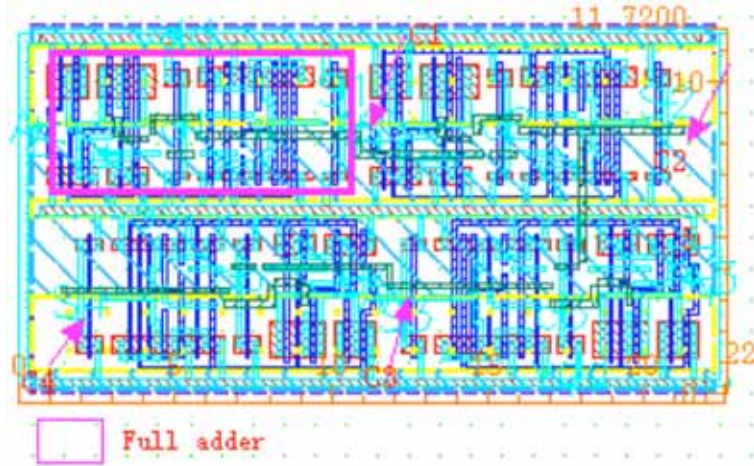The layout of 4-bit RCA has been made in a 0.13 nm process just as shown in Fig. (**3**).

Fig. (3). Layout of 4-bit ripple-carry adder. Die size equal to 22.12 * 11.72 μm^2.

For some input signals, no rippling effect occurs at all, while in the worst case, the carry has to ripple all the way from the least significant bit position to the most significant bit position. The carry finally consumed in the last stage to produce the sum. The RCA is concise and useful when N is small, however, when N is large, RCA is redundancy since the maximum carry propagation time is proportional to N. Thus, RCA isn't the optimal choice used for VLSI. It is far more important to optimize carry china than sum, since the latter has only a minor influence on the total value of time-delay [10].

We define $G_i = A_i B_i$ and $. = . \oplus .$ that in order to better analysis. After substituting the G_i and P_i into the equation of multi-bit adder, when $G_i = A_i B_i$ is true if a carry is generated by the full adder and $. = . \oplus .$ is true if the carry is propagated by the full adder from C_{i-1} to C_i.

Since the C_{out} signals of each full adder is in the critical path. Taking the 4-bit RCA for example, the expression for critical path is shown in equation (**6-9**).

$$C_1 = G_1 + P_1 C_0 \tag{6}$$

$$C_2 = G_2 + P_2 C_1 \tag{7}$$

$$C_3 = G_3 + P_3 C_2 \tag{8}$$

$$C_4 = G_4 + P_4 C_3 \tag{9}$$

We can calculate the time-delay of every gate: NAND is T, NOR is T, NOT is 0.5T. Then every time adds an adder needs 2T after the G_i and P_i come out. The 4-bit RAC needs 2T*4=8T. For N-bit ripple-carry adder, the longest time of carry chain is 2N*T. The adders' speed will largely depend upon the carry chain, so optimize the carry chain is the critical factor in the design.

2.2. Carry-Lookahead Adder

When designing even fast adders, it is essential to get around the rippling effect of the carry. The carry-lookahead (CLA) principle offers a possible way to ensures that a carry-bit will be generated at C_{out} (C_i) independent of C_{in} (C_{i-1}), respectively, as shown in equation (**10-13**).

$$C_1 = G_1 + P_1 C_0 \tag{10}$$

$$C_2 = G_2 + P_2 G_1 + P_2 P_1 C_0 = G_{2:1} + P_{2:1} C_0 \tag{11}$$

$$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_0 = G_{3:1} + P_{3:1} C_0 \tag{12}$$

$$C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_0$$
$$= G_{4:1} + P_{4:1} C_0 \tag{13}$$

The adder produces all carry needs 4T after the G_i and P_i come out, so the 4-bit carry-lookahead adder needs less time than ripple-carry adder.

The relative to equation (**10-13**) corresponding figure, using logic gates are shown in Fig. (**4**).

The carry out of a 4-bit block can be computed using only the block generate and propagate signals for each 2-bit section. We can order the generate and propagate function as a pair ($G_{i:j}$, $P_{i:j}$). A new Boolean expression can be used, just as shown in equation (**14**).

$$(G_i, P_i) \bullet (G_{i-1}, P_{i-1}) = (G_i + P_i G_{i-1}, P_i P_{i-1}) \tag{14}$$

With the new Boolean expression, a 4-bit adder can be re-written as

$$(C_4, 0) = (G_4, P_4) \bullet (G_3, P_3) \bullet (G_2, P_2) \bullet (G_1, P_1) \bullet (C_0, 0) \tag{15}$$

$$(G_{4:1}, P_{4:1}) = (G_4, P_4) \bullet (G_3, P_3) \bullet (G_2, P_2) \bullet (G_1, P_1)$$
$$= (G_{4:3}, P_{4:3}) \bullet (G_{2:1}, P_{2:1}) \tag{16}$$

So the equation (**17**) can be derived from equation (**15**)

$$(C_i, 0) = (G_i, P_i) \bullet (G_{i-1}, P_{i-1}) \searrow \searrow (C_0, 0) \tag{17}$$

We can choose the maximum fan-in for our logic gates and then build a hierarchical carry chain as shown in Fig. (**5**).

We can use this building block to construct P/G signals for any width operands. The following figure shows how this works for 4-bit operands using 2-input logic gates. By ex-
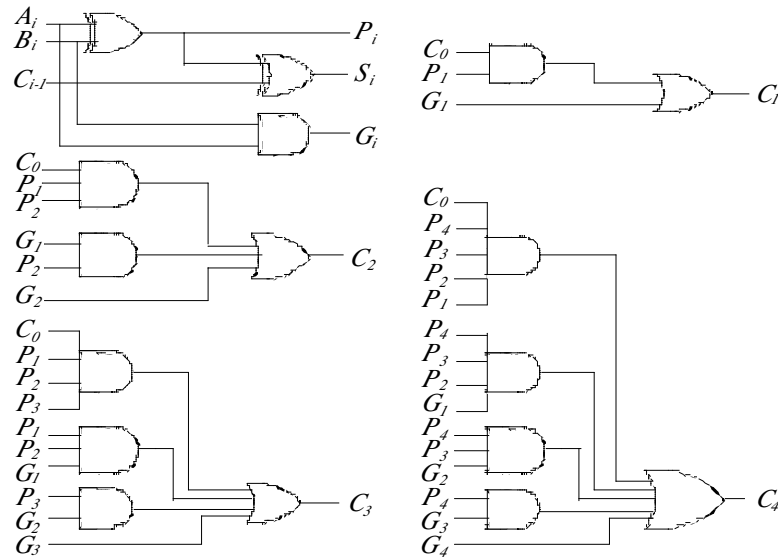
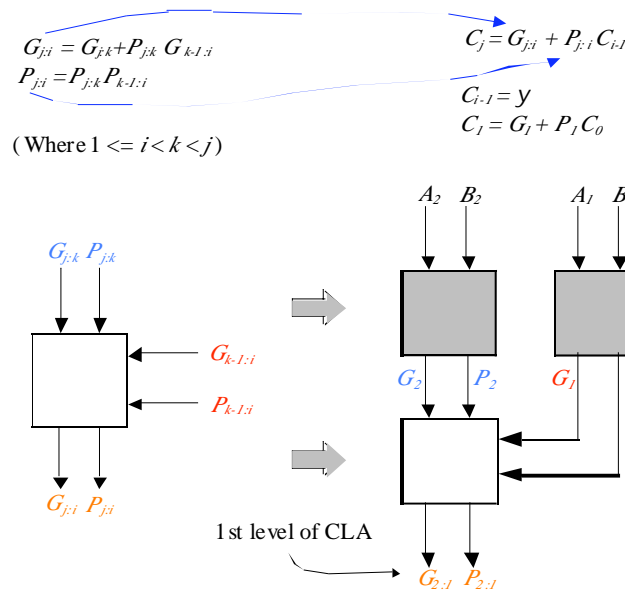**Fig. (4).** Carry chain of the carry-lookahead adder.

$$G_{j:i} = G_{j:k} + P_{j:k}\, G_{k-1:i}$$
$$P_{j:i} = P_{j:k}\, P_{k-1:i}$$

(Where $1 <= i < k < j$)

$$C_j = G_{j:i} + P_{j:i}\, C_{i-1}$$

$$C_{i-1} = y$$
$$C_1 = G_1 + P_1 C_0$$

Fig. (5). Hierarchical building block.

ploit the Fig. (**5**), a tree can be constructed that effectively computes the carries and sums is shown in Fig. (**6**). The most important factor is that the output carry of an *N*-bit adder can be computed in $O(\log_2 (N))$ time.

Brent and Kung opened the way to a class of carry-lookahead adders based on a binary-tree structure as shown in Fig. (**7**) [11].

The CLA algorithm was first raised by Weinberger and Smith [12]. Various carry-lookahead adders are presented, among them, one shown in Fig. (**8**) is a conventional type.

In summary, the CLA is based on a tree structure to reduce the time complexity to $O(\log_2 (N))$ where *N* is the number of bits. The CLA compute the values C_i using propagate/generate trees in parallel for all ranks instead of trying

to propagate them as fast as possible. For example at rank *i*, a carry-out equal to 1 occurs in the following cases:

·Rank *i* generates a carry-out equals to 1 (*i.e.*, $g_i = 1$).

·Or rank *i* propagates a carry generated at rank *i-1* (i.e., $p_i = g_{i-1} = 1$).

·Or ranks *i* and *i-1* propagate a carry generated at rank *i-2* (i.e., $p_i = p_{i-1} = g_{i-2} = 1$).

·Or ranks *i* to 1 propagate the adder carry-in C_0 equal to 1 (i.e., $p_i = p_{i-1} = \ldots = p_2 = p_1 = c_0 = 1$).

Therefore, all the carry-in bits can be computed using the relation is shown in equation (**18**):

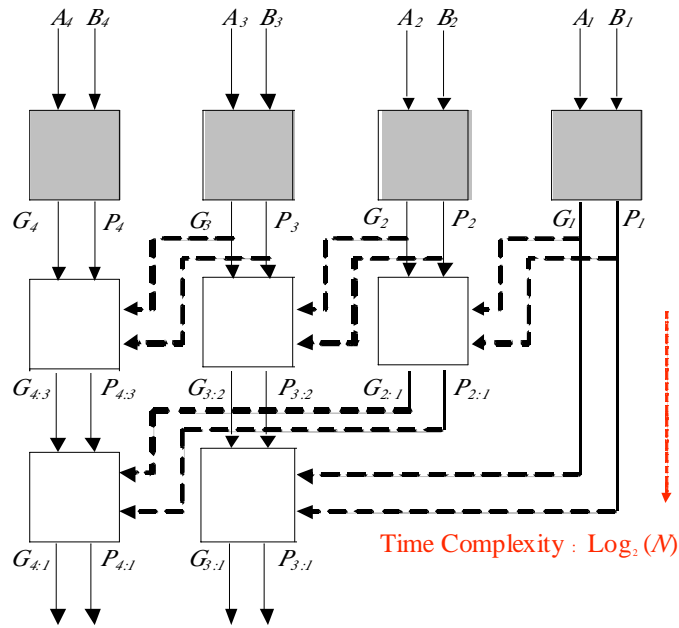$$C_i = g_i + g_{i-1} p_i + g_{i-2} p_i p_{i-1} + \searrow \searrow \tag{18}$$

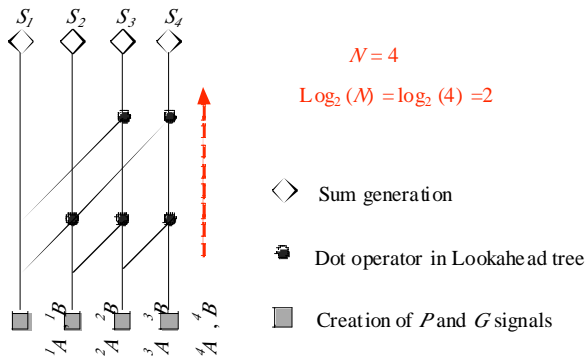**Fig. (6).** Carry path of the 4-bit carry-lookahead adder.

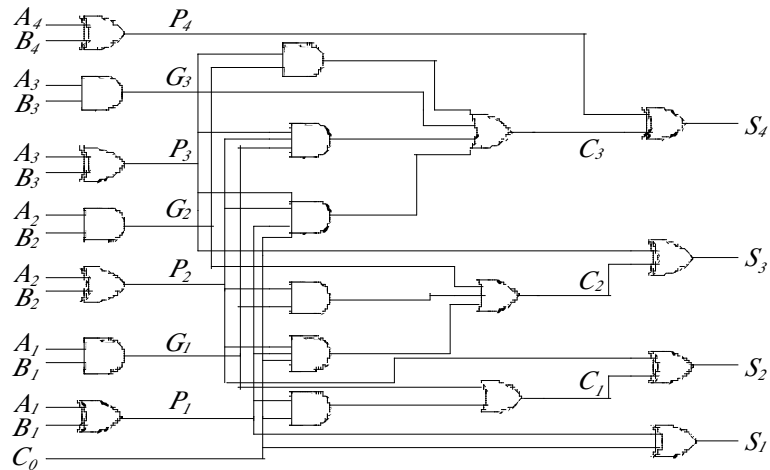**Fig. (7).** The structure of the tree adder.

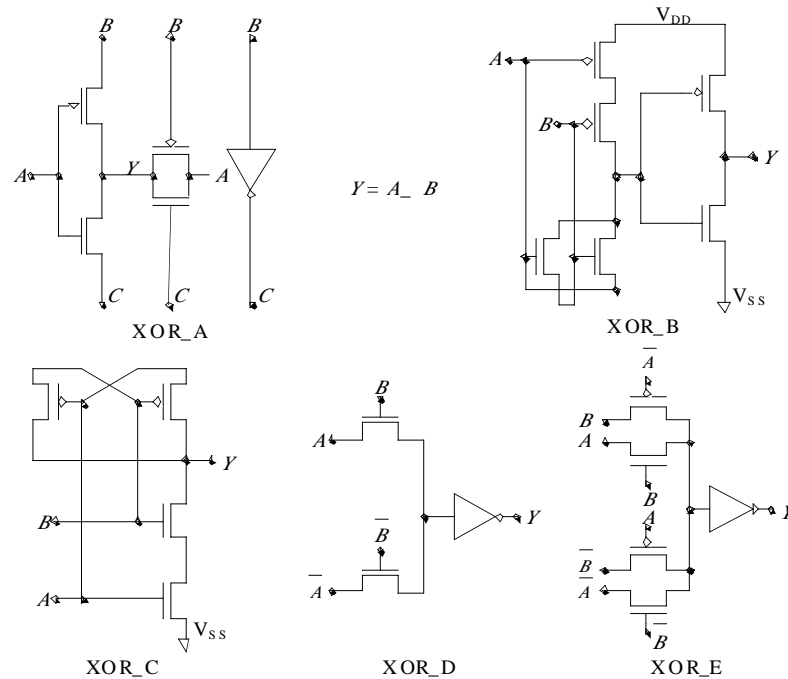**Fig. (8).** 4-bit Carry-lookahead adder.

$Y = A \_ B$

Fig. (9). Implementation comparison of XOR circuits.

The addition is performed in three stages for all rank *i*:

1. Generation of propagate and generate signals (g_i, p_i).
2. Generation of the carry signals using equation (**18**).
3. Parallel computation of $S_i = A_i \oplus B_i \oplus C_{i-1} = P_i \oplus C_{i-1}$.

2.3. Circuit Optimization of XOR Gate

The proper choice of the logic style can significantly improve different aspects of the performance of a 4-bit full adder cell. Several different logic styles of XOR circuits are designed and shown in Fig. (**9**) which is the most basic gate in an adder circuit.

In the following paragraphs, the complementary CMOS logic, complementary pass-transistor logic (CPL), double pass-transistor logic (DPL), transmission gate, all of which belong to the sets of the static logic, are used as a basis for comparison. The characteristic of circuit are briefly described as follows.

(1). Complementary CMOS Logic

XOR_B in Fig. (**9**) uses complementary CMOS structure (pull-up &down networks) has 6 transistors and involves minimum design risk. One of the advantages of the complementary CMOS cell is high noise margins and stable operation at low voltages. The layout of CMOS gates is symmetrical due to the complementary transistor style. The disadvantages of cell is the large number of PMOS which results in significant area overhead, more power consumption and high input loads, what's more, the high input capacitance produces an unwanted additional delay.

(2). Pass-Transistor Logic (PTL)

XOR_C and XOR_D in Fig. (**9**) uses PTL has 4 transistors and 8 transistors, respectively. The XOR_C in Fig. (**9**) uses single pass transistor logic which uses nearly half as many transistors and minimizes the propagation delay. However, it is not suitable in all applications due to a weak drivability at the output for A = B = 0. Furthermore, it's not operation well below 1.4V supply voltages for low power application. XOR_D uses complementary pass transistor logic to realize an XOR function. However, the outputs of the nMOSFET pass-transistor network suffer from threshold voltage drop, which results in the incomplete turn-off of pMOSFET's in the inverters. This may in large power consumption. This structure could require buffer to achieve desirable outputs.

(3). Double Pass-Transistor Logic (DPL)

XOR_E in Fig. (**9**) has 10 transistors include two inverters uses double Pass-transistor logic in which both NMOS and PMOS logic network are used [13]. The advantages of this gate avoids the nMOSFET threshold voltage drop issue of the CPL design and eliminate the power consumption, However, the drawback of this gate is a large area because of the PMOS used

(4). Transmission Gate Logic

XOR_A in Fig. (**9**) has only 6 transistors and its complements as a selected signal to produce the output through the inverter or transmission gate. It keeps full swing operation and alleviates the power dissipation.

After a comprehensive comparative study, we choose the Transmission Gate XOR cell to instead of the XOR gate in PCLA by considering the trade-off between the power and speed.
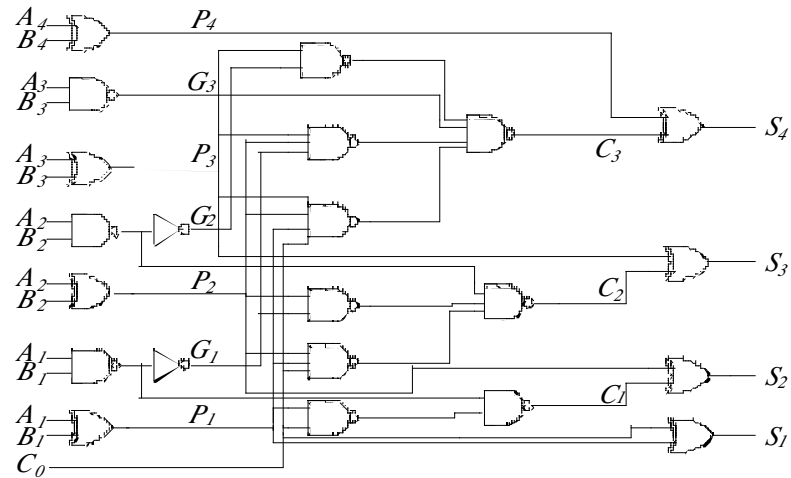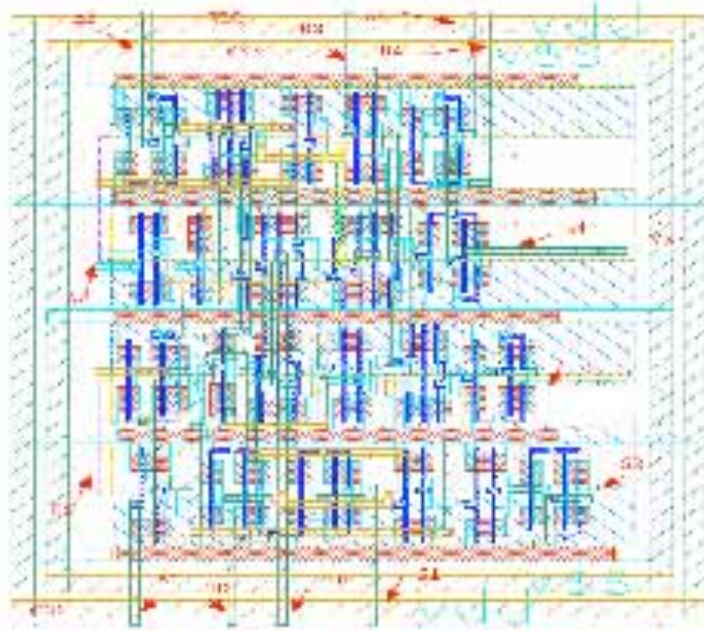
Fig. (10). Proposed Carry-lookahead circuit.

Fig. (11). The layout of the proposed Carry-lookahead adder. Die size equal to 17.69 * 18.54 μm^2.

2.4. Proposed Circuit

In this section a new method for modifying the CLA is proposed. Conventional CLA is constitute by AND, OR, and XOR logic gates. The proposed carry-lookahead adder (PCLA) uses NAND gates to instead of the AND in CLA circuit, it can improve the speed of CLA and decrease the area of CLA. Let i be the number of bit. The input variables A_i, B_i, C_i, and output S_i are bits of augends, addend, carry, and sum at stage i, respectively. The carry of the any stage can be expressed as shown in equation (**19**).

$$C_i = \overline{\overline{G_i}\,\overline{P_i C_{i-1}}} \tag{19}$$

From equation (**19**), the carry outputs of each stage be listed in the following as shown in equation (**20-23**):

$$C_1 = \overline{\overline{C_1}} = \overline{\overline{G_1 + P_1 C_0}} = \overline{\overline{G_1}\,\overline{P_1 C_0}} \tag{20}$$

$$C_2 = \overline{\overline{C_2}} = \overline{\overline{G_2 + P_2 G_1 + P_2 P_1 C_0}} = \overline{\overline{G_2}\,\overline{P_2 G_1}\,\overline{P_2 P_1 C_0}} \tag{21}$$

$$C3 = \overline{\overline{C3}} = \overline{\overline{G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_0}}$$
$$= \overline{\overline{G_3}\,\overline{P_3 G_2}\,\overline{P_3 P_2 G_1}\,\overline{P_3 P_2 P_1 C_0}} \tag{22}$$

$$C_4 = \overline{\overline{C_4}} = \overline{\overline{G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_0}}$$
$$= \overline{\overline{G_3}\,\overline{P_3 G_2}\,\overline{P_3 P_2 G_1}\,\overline{P_3 P_2 P_1 C_0}} \tag{23}$$

By using the equation (**20-23**), the 4-bit PCLA can be implemented easily with NAND logic gate and offers high-speed and low-power consumption at 1.2V supply voltages, just as shown in Fig. (**10**).

The designed construction of PCLA circuit is similar to CLA circuit. All of the components are implemented with NAND gates except for the outputs of P and S, which are

Two inverters are provide a situation similar realistic condition

when measuring the power consumption of circuit,
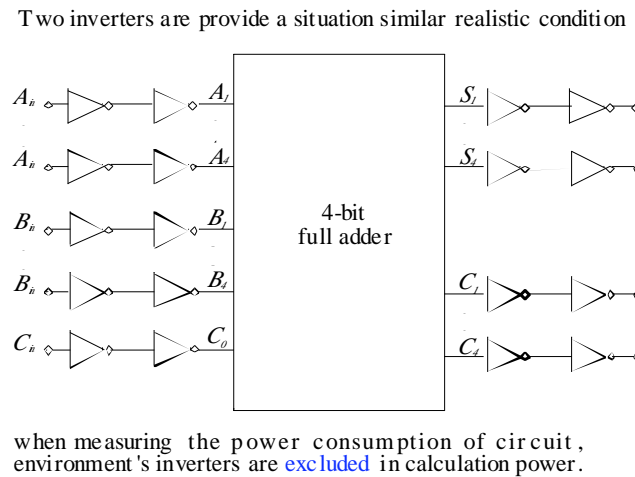environment's inverters are excluded in calculation power.

Fig. (12). Simulation environment.

implemented with XOR gates. The layout was made for the PCLA which used SMIC 130nm technology as shown in Fig. (**11**).

The metal lines are placed horizontally at the top and the bottom for the power supply (V_{dd}) and ground (V_{ss}). After the layout design, the simulation can be tested. Taking the implementation area into consideration obtained from the layouts, it can be present that the area of cells can significant affects the performance of systems.

3. RESULTS AND DISCUSSIONS

3.1. Simulation Environment

All the circuits are designed in Cadence VIRTUOSO environment using CMOS design kit. The netlists of all adders are extracted and post-layout simulations are carried out at 27°C with an input frequency of 10 MHz, 50 MHz, 100 MHz, and 500 MHz, respectively. By optimizing the transistor size of all adders considered, the Power-delay product (PDP) can be set to achieve minimum as far as possible.

A circuit responds differently to different technology corner model. So we use five corners to nearly cover all possible corner models. Each model is simulated 4 times using frequencies at 10 MHz, 50 MHz, 100 MHz, and 500 MHz. Thus, for each adder, 20 HSPICE simulations run (5 technology corner models * 4 frequencies) are execute. This present a total of 60 simulation runs are comparison.

For a simulation, 50 complete periods are given. The average power of every adder is taken from the beginning of the second period to the end of the fiftieth period. In order to avoid transient glitches, the testing can not include the first period.

In this paper, the time-delay is defined as the maximum delay which is associated with the longest path is measure for *SUM* and C_{out} output in the circuit, the value of time-delay is the fastest edge of all input signals to the output signal, the Power-delay product (PDP) is the significant quality measure parameter of the efficiency and a compromise be-

tween power dissipation and speed for CMOS circuits [14]. This value is calculated from worst-case delay multiplied with average power consumption is given as equation (**24**).

$$PDP = Power_{average} \times Delay_{worst\text{-}case} \tag{24}$$

To produce more realistic performance in the simulation, two CMOS inverters are added to inputs and output nodes, respectively, the complete simulation environment is shown in Fig. (**12**). The circuit signals are probed at the outputs of the input inverters and at the inputs of the output inverters [15]. The cell was simulated by HSPICE based on 130 nm CMOS technology at 1.2 V supply voltages.

3.2. Simulation Results and Comparison

For each transition, all bits of addends, augends and carry are set to 1. The delay is measured from 50% of the input voltage swing to 50% of the output voltages swing. The maximum delay is taken as the time-delay. The results show that the worst case delay happens when a carry generated at the least significant bit position propagates all the way to the most significant bit position. The simulated data of time-delay are summarized in Table **1,** Table **2,** Table **3,** and Table **4**.

Fig. (**13**) shows a comparison of all adders for power consumption at TT technology corner model.

As we can be seen, the power consumption of all adders increases as frequency increases, the high power due to high switching activity. The graph shows that the best adder which consumes the least power is PCLA whereas the CLA have the largest power consumption due to the transistor count. Table **1-4** show the time-delay at four different frequencies are almost equal, and the RCA have the largest delay value due to its long level on the critical paths. The simulation results indicate an actual trend that the delay of RCA will increased with the level of carry chain. However, the PCLA is faster than CLA is attributed to the new XOR cell (XOR_A) and inverter logic structure (equation **19**).

We need make a tradeoff between power and delay, the results which indicate the fastest adder is not always the most

Table 1. Performance Comparison of 4-Bit Full Adders at 10 MHz.

Cell Name	#tr.	Time - Delay (ps)					Power(μw) At 10MHz		PDP (TT) (μw*ns)
		TT	SS	FF	SF	FS	Avg.	max.	
RCA	112	291.8	370.7	242.2	292.9	299.6	2.89	3.08	0.84
CLA	134	267.7	336.3	224.9	270.5	273.1	3.71	3.82	0.99
PCLA	114	221.8	244.7	151.6	180.9	202.1	1.62	1.70	0.36

Table **1** shows the transistor count, power consumption, time-delay, and PDP comparison profile for the three different adders at 10 MHz.

Table 2. Performance Comparison of 4-Bit Full Adders at 50 MHz.

Cell Name	#tr.	Time - Delay (ps)					Power(μw) At 50 MHz		PDP (TT) (μw*ns)
		TT	SS	FF	SF	FS	Avg.	max.	
RCA	112	290.9	369.9	241.43	291.8	298.4	14.41	15.31	4.16
CLA	134	265.8	334.9	218.5	269.9	270.2	19.16	20.11	5.08
PCLA	114	186.1	244.7	151.4	180.8	201.6	8.11	8.44	1.49

Table **2** shows the transistor count, power consumption, time-delay, and PDP comparison profile for the three different adders at 50 MHz.

Table 3. Performance Comparison of 4-Bit Full Adders at 100 MH.

Cell Name	#tr.	Time - Delay (ps)					Power(μw) At 100 MHz		PDP (TT) (μw*ns)
		TT	SS	FF	SF	FS	Avg.	max.	
RCA	112	290.4	370.4	241.2	291.4	298.1	28.65	30.50	8.39
CLA	134	265.3	333.9	223.6	268.8	270.7	36.44	37.34	9.65
PCLA	114	185.9	243.9	151.3	180.9	201.6	16.14	16.83	3.00

Table **3** shows the transistor count, power consumption, time-delay, and PDP comparison profile for the three different adders at 100 MHz.

Table 4. Performance Comparison of 4-Bit Full Adders at 500 MHz.

Cell Name	#tr.	Time - Delay (ps)					Power(μw) At 500 MHz		PDP (TT) (μw*ns)
		TT	SS	FF	SF	FS	Avg.	max.	
RCA	112	289.4	366.8	242.8	289.5	297.8	141.96	151.80	40.83
CLA	134	263.5	330.3	221.9	267.2	268.5	182.20	186.50	48.04
PCLA	114	185.2	243.5	150.8	180.1	200.5	84.13	80.30	14.84

Table **4** shows the transistor count, power consumption, time-delay, and PDP comparison profile for the three different adders at 500 MHz.

power-efficiency circuit. PDP is summarized in Fig. (**14**) to evaluate the comprehensive performance of all adder circuits at different frequencies. The PDP of the PCLA circuit is improved up to 60%-70% and 55%-65% as compared with CLA circuit and RCA circuit, respectively. The PDP of CLA has a disadvantage over RCA in allfrequencies we simulated.

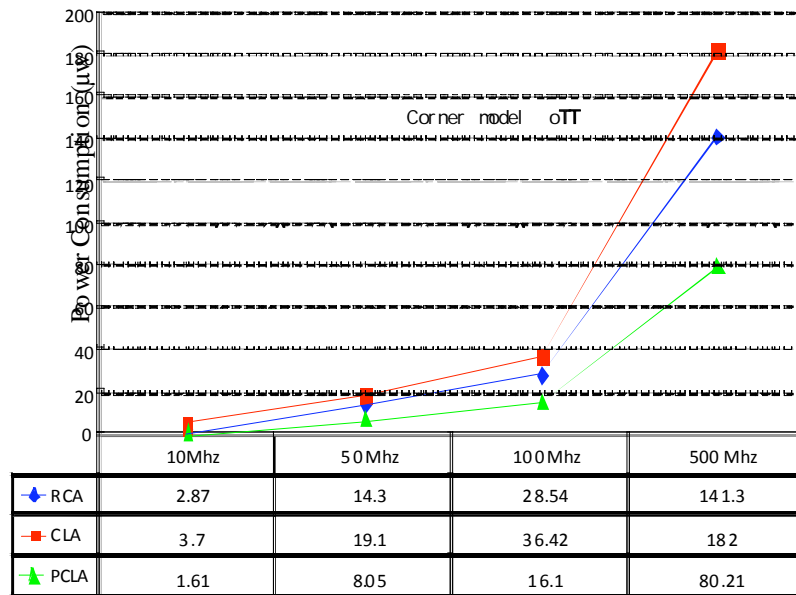The reduction in power consumption is not sufficient to balance the increase in delay when the basic unit is done

	10Mhz	50Mhz	100Mhz	500 Mhz
RCA	2.87	14.3	28.54	141.3
CLA	3.7	19.1	36.42	182
PCLA	1.61	8.05	16.1	80.21

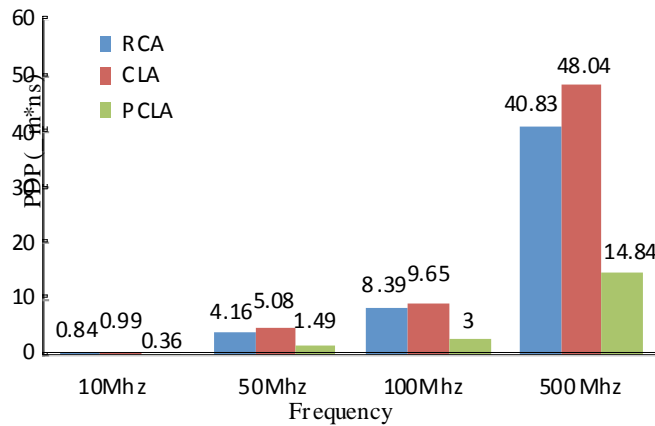**Fig. (13).** Comparison of power consumption at different frequencies.

Fig. (14). Comparison of PDP at different frequencies.

using CLA. Overall, the PCLA is an excellent alternative in this paper for PDP-efficient designs.

CONCLUSION

The performance of many complicated circuits is tightly dependent on the performance of the full adder circuit that has been used.

This paper presented a methodology that uses NAND gates to optimize the arithmetic and a XOR gate with transmission gate logic structure for designing adders to minimize the PDP for high performance applications, the corresponding layouts were simulated by HSPICE based on 130 nm CMOS technology at different frequencies condition. The comprehensive simulation shows that PDP of the PCLA circuit is improved up to 60%-70% and 55%-65% as compared with CLA circuit and RCA circuit, respectively. Thus, the PCLA is good candidate to build these large systems.

CONFLICT OF INTRREST

The authors confirm that this article content has no conflict of interest.

REFERENCES

[1]   S. Kajihara, and T. Sasao, "On the adders with minimum tests", *Test Symposium*, pp. 10-15, November 1997.
[2]   A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-Power CMOS digital design", *IEEE Journal of Solid-State Circuit.*, vol. 27, pp. 473-483, 1992.
[3]   M. Vesterbacka, "A 14-transistor CMOS full adder with full voltage-swing nodes", *IEEE Workshop on Signal Processing Systems.*, pp. 713-722, 1999.
[4]   N. Zhuang and H. Wu, "A new design of the CMOS full adder," *IEEE Journal of Solid-State Circuit*, vol. 27, pp. 840-844, 1992.

[5]   T. Lynch, and E.E. Swartzlander, "A spanning tree carry lookahead adder", *IEEE transaction on Computers*, vol. 41, pp. 931-939, 1997.

[6]   A.M. Shams, and M.A. Bayoumi, "A structured approach for designing low power adders", *Computers Signals Systems & Amp.* vol. 1, pp. 757-761, 1997.

[7]   Jan M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital integrated circuits, a design perspective*, Prentice-Hall. Inc, 1996.

[8]   V. Pudi and K. Sridharan, "low complexity design of ripple carry and brent–kung adders in QCA", *IEEE transactions on Nanotechnology*, vol. 11, pp. 105-119, 2012.

[9]   A. Weinberger and J. L. Smith, "A logic for high speed addition", *National Bureau of Standards Circular*, pp. 3-12, 1958.

[10]   M. S. Lau, K. V. Ling, Y. C. Chu, and A. Bhanu, "Modeling of Probabilistic Ripple-Carry Adders", 5th *IEEE International Symposium on Electronic Design, Test & Applications*, 2010; pp. 201-206.

[11]   A. Neve, H. Schettler, T. Ludwig, and D. Flandre, "Power-delay product minimization in high-performance 64-bit carry-select adders", *IEEE transaction VLSI Systems*., vol. 12, pp. 235-244, 2004.

[12]   D. Radhakrishnan, "Low-voltage low-power CMOS full adder", *IEEE Proceeding*, *Circuits. Devices and Systems*, vol. 148, no. 1, pp. 19-24, 2001.

[13]   A. Sudsakom, S. Tooprakai, and K. Dejhan, "Low power CMOS full adder cells", *ECTICON*, pp. 1-4, 2012.

[14]   K. Dongho, and T. Ambler, "Low power carry-lookahead adder by using dependency between generation and propagation", in *Proceeding IEEE Devices, Circuits and Systems*., pp. C68/1-C68/6, 2000.

[15]   M. Aguirre-Hernandez, and M. Linares-Aranda, "CMOS full-adders for energy-efficient arithmetic applications", *IEEE Transaction on VLSI Systems*, vol. 19, pp. 718-721, 2011.