

Performance analysis of radix-4 adders

Shahzad Asif and Mark Vesterbacka

Linköping University Post Print

N.B.: When citing this work, cite the original article.

Original Publication:

Shahzad Asif and Mark Vesterbacka, Performance analysis of radix-4 adders, 2012, Integration, (45), 2, 111-120.

<http://dx.doi.org/10.1016/j.vlsi.2011.09.004>

Copyright: Elsevier

<http://www.elsevier.com/>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-76617>

Performance Analysis of Radix-4 Adders

Shahzad Asif and Mark Vesterbacka

Abstract—We present a radix-4 static CMOS full adder circuit that reduces the propagation delay, PDP, and EDP in carry-based adders compared with using a standard radix-2 full adder solution. The improvements are obtained by employing carry look-ahead technique at the transistor level. Spice simulations using 45 nm CMOS technology parameters with a power supply voltage of 1.1 V indicate that the radix-4 circuit is 24% faster than a 2-bit radix-2 ripple carry adder with slightly larger transistor count, whereas the power consumption is almost the same. A second scheme for radix-2 and radix-4 adders that have a reduced number of transistors in the carry path is also investigated. Simulation results also confirm that the radix-4 adder gives better performance as compared to a standard 2-bit CLA. 32-bit ripple carry, 2-stage carry select, variable size carry select, and carry skip adders are implemented with the different full adders as building blocks. There are PDP savings, with one exception, for the 32-bit adders in the range 8% to 18% and EDP savings in the range 21% to 53% using radix-4 as compared to radix-2.

Keywords—CMOS full adder, radix-4 adder, power-delay product, energy-delay product, carry-based adder

1. INTRODUCTION

Addition is the most basic and most frequently used operation in digital circuit design [1][2]. Due to this reason, a large number of adder architectures have been designed to accommodate the requirements of different applications. Common parameters that measure the performance of a circuit are speed, power, area, power-delay product (PDP), and energy-delay product (EDP), where the PDP represents the energy dissipation per switching event and the EDP is a commonly used measure for energy-performance trade-off [3][4]. On the basis of area and speed, we can divide the adders in three major categories. The ripple carry adder (RCA) is the slowest architecture with the least number of transistors. The carry look-ahead adder (CLA) is the fastest adder but requires very large area. The carry select adder (CSEL) is used when we need a moderate solution [5][6][7]; it is faster than RCA but slower than CLA.

Extensive research has been done to improve the performance of the full adder (FA), which is the basic

building block for large carry-based adders. In [8][9][10][11], different architectures have been presented to improve the performance of the FA in terms of speed, power, and energy. In [12], different FA architectures are analyzed and compared with respect to speed, power consumption and PDP. In [13] and [14], different radix-2 and radix-4 architectures are presented and compared for 64-bit parallel prefix adders with main emphasis on the Kogge-Stone adder.

In this work, we design a radix-4 FA instead of the more common radix-2 FA. The radix-4 FA is based on internal carry look-ahead to reduce the carry propagation path in an adder since that path is critical [3]. The main disadvantage of this architecture is a slight increase in transistor count.

With the scaling of technology and supply voltage, it is becoming more important to design the circuits having full voltage swing, which results in high noise margins. For low power applications, it is also common to reduce the supply voltage, but doing so will result in even lower noise margins [15]. Static CMOS logic provides high gain and a voltage swing equal to the supply voltage, yielding high noise margins. Hence, the CMOS logic style is a good choice for low power implementation in deep submicron technologies [16], which we have chosen for our design.

The paper is organized as follows. In Section 2, we study different FA topologies. In Section 3, two radix-4 FAs are proposed. In Sections 4 and 5, we discuss 32-bit adders that are implemented for comparison. Section 6 shows the simulation results of the adders, and the work is concluded in Section 7.

2. FA TOPOLOGY

In this work we focus on static CMOS circuits, which provide the high noise margins required for implementation in deep submicron technologies. A popular topology is the mirror FA [12], which is used as the base topology for our designs. Two different mirror FA circuits are examined.

Fig. 1 shows the typical 28-transistor mirror FA. The carry is generated first since it is on the critical path, and then it is reused in the sum generation to obtain low area.

Another mirror FA that consists of 24 transistors is shown in Fig. 2 [17]. Compared to the previous FA, the transistor

S. Asif is with the Dept. of EE, COMSATS IIT, 44000 Islamabad, Pakistan (e-mail: shahzad_comsats@yahoo.com).

M. Vesterbacka is with the Dept. of EE, Linköping University, SE-581 83 Linköping, Sweden (e-mail: markv@isy.liu.se).

count is reduced for the sum generation while the circuitry for the carry generation is the same as in Fig. 1.

Both FAs implement the same Boolean function, which is given in equation (1) and (2).

$$C_{out} = (A + B)C_{in} + A.B \quad (1)$$

$$Sum = \overline{C_{out}}.A + \overline{C_{out}}.B + \overline{C_{out}}.C_{in} + A.B.C_{in} \quad (2)$$

The circuits are designed using 45 nm CMOS technology parameters [18] and simulated in Spice with a power supply voltage of 1.1 V. Widths of all NMOS and PMOS devices are set to 0.1 μm and 0.2 μm , respectively. For power dissipation we applied pulses with time periods 6 ns, 12 ns, and 24 ns to inputs A , B , and C_{in} , respectively. An inverter is used as load on the carry and sum output in the simulations of the delay and energy dissipation, which are reported in Table 1. Widths of NMOS and PMOS for this inverter are set to 0.1 μm and 0.2 μm , respectively.

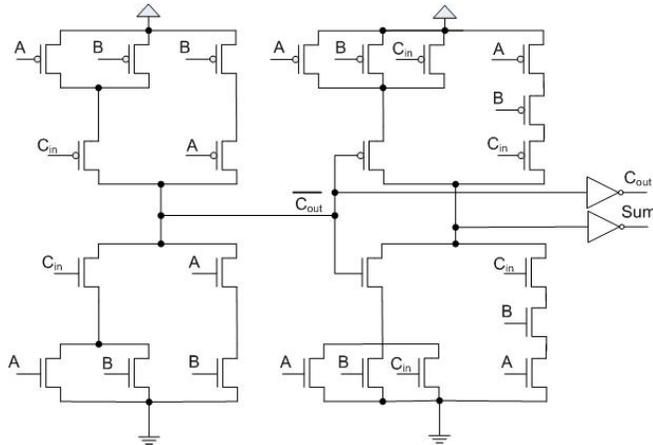


Fig 1. 28-transistor mirror FA.

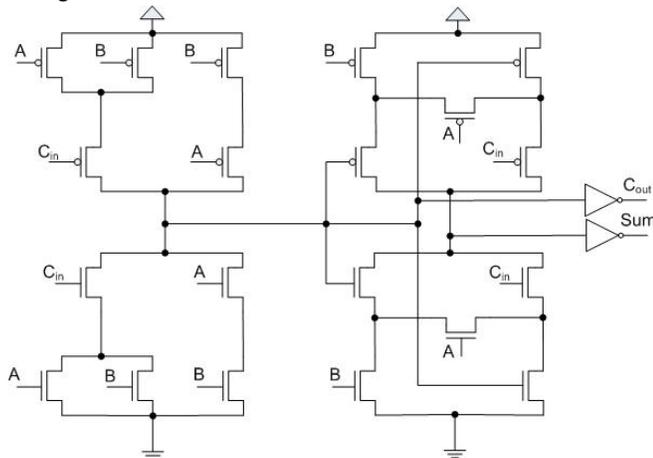


Fig 2. 24-transistor mirror FA.

From the results given in Table 1, it appears that the 28-transistor FA is better as compared to the 24-transistor FA in terms of speed and energy dissipation. However, when we build large adders using this FA, its speed tends to decrease. The reason for this is the fan-out of the carry signal. In the 28-transistor FA, C_{out} is connected to 6 transistors while in the 24-transistors FA, C_{out} is connected to only 4 transistors. This difference in fan-out causes the circuit to be slower for the 28-transistor FA.

Results from simulation of a 32-bit RCA constructed by the two different FAs are shown in Table 2. For estimating power dissipation, random inputs are generated by a pseudo-random binary sequence (PRBS) generator. Characteristics of this PRBS are discussed in Section 6.1.

It is clear from the data in Table 2 that the 24-transistor FA is better than the 28-transistor FA with respect to speed, power, PDP, and EDP when used as a building block in the 32-bit RCA.

TABLE 1
COMPARISON OF MIRROR FAS

	28-transistor mirror FA	24-transistor mirror FA
Technology	45 nm CMOS	
PMOS Width (μm)	0.2	
NMOS Width (μm)	0.1	
Supply Voltage (V)	1.1	
Delay of C_{out} (ps)	15.7	18.5
Delay of Sum (ps)	23.3	30.3
Transistor Count	28	24
Energy/Operation (fJ)	2.55	2.53

TABLE 2
32-BIT RCA USING DIFFERENT MIRROR FAS

	32-bit RCA	
	28-transistor mirror FA	24-transistor mirror FA
Technology	45 nm CMOS	
PMOS Width (μm)	0.2	
NMOS Width (μm)	0.1	
Supply Voltage (V)	1.1	
Delay of C_{out} (ps)	502	467
Delay of Sum (ps)	508	480
Transistor Count	896	768
Power Dissipation (μW)	236	226
PDP (fJ)	120	108
EDP (zJ/Hz)	61.0	52.0

3. RADIX-4 ADDERS

This section presents two different architectures for Radix-4 FA. A 2-bit standard CLA is also presented for the comparison purpose.

3.1. Basic Radix-4 FA

The main approach in this work to reduce the PDP and EDP of an adder is to reduce the delay due to carry propagation. We use the 24-transistor mirror FA as a base for our design, which is illustrated in Fig. 2.

To optimize the speed, we joined two FAs in a single block and use the carry look-ahead technique [19] to shorten the carry path within the radix-4 FA block. The block diagram of the proposed radix-4 FA is shown in Fig. 3 and the transistor schematic is shown in Fig. 4. The total transistor count is 56. Boolean equations for the two carry and two sum gates of the carry-accelerated design are given below.

$$\overline{C_1} = \overline{(A_0 + B_0)C_{in} + A_0 \cdot B_0} \quad (3)$$

$$S_0 = \overline{C_1} \cdot B_0 + \overline{C_1} \cdot C_{in} + \overline{C_1} \cdot A_0 + A_0 \cdot B_0 \cdot C_{in} \quad (4)$$

$$C_{out} = (A_0 + B_0)(A_1 + B_1)C_{in} + A_0 \cdot B_0(A_1 + B_1) + A_1 \cdot B_1 \quad (5)$$

$$S_1 = \overline{C_{out} \cdot B_1 + C_{out} \cdot C_1 + C_{out} \cdot A_1 + A_1 \cdot B_1 \cdot C_1} \quad (6)$$

The circuit for computing the least significant sum bit S_0 is the same as that of a radix-2 FA. We use the carry look-ahead technique to design the circuitry for the most significant carry bit C_{out} . Equation (5) is similar to the equation for a CLA with the difference that generate and propagate signals are generated explicitly for the CLA. To implement the function that computes the most significant sum bit S_1 with a single gate, we invert the inputs A_1 , B_1 and C_1 .

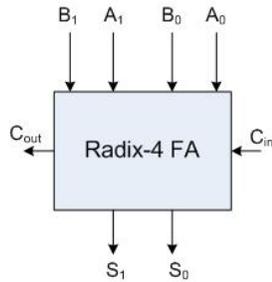


Fig. 3. Radix-4 FA block.

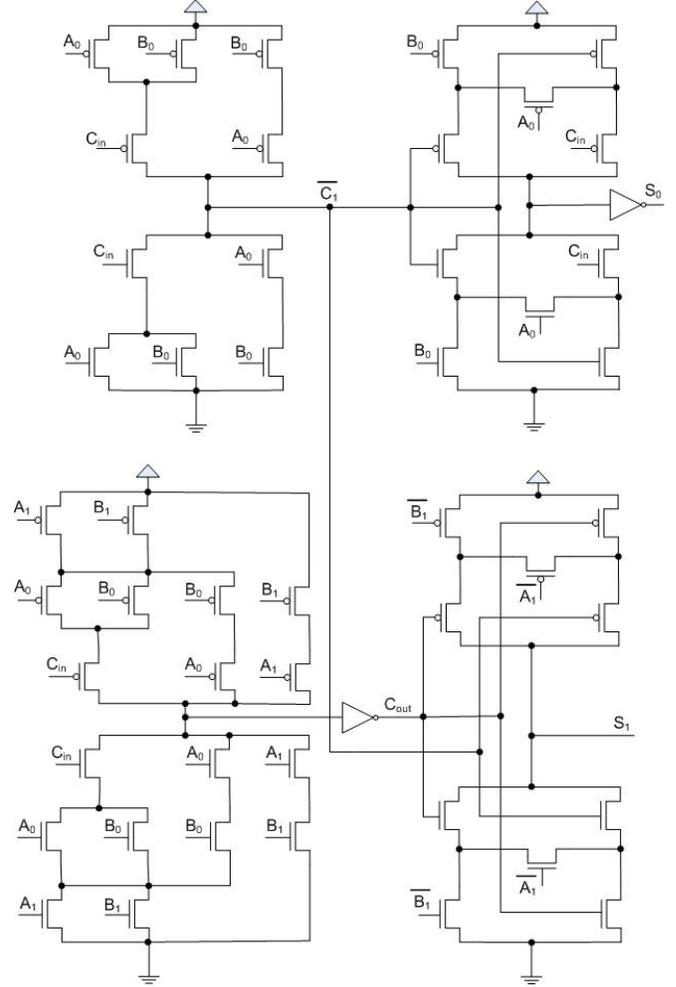


Fig. 4. Radix-4 FA.

Table 3 shows the intrinsic delay (without load) for the basic components used in the standard and proposed adders. These values are used in the optimization of the adders presented in Section 4.

TABLE 3
DELAY OF BASIC COMPONENTS

	Propagation Delay (ps)
Inverter	1.79
2:1 Multiplexer	10.7
C_{out} , Radix-2 FA	16.0
Sum, Radix-2 FA	28.5
C_{out} , Radix-4 FA	19.4
Sum, Radix-4 FA	27.0

3.2. Pre-computed Radix-4 FA

In this section, we modify the radix-4 FA presented in Section 3.1 so that the carry acceleration signals are pre-computed. By using these signals, the carry can be generated with less complex circuitry resulting in less delay. A radix-2 FA with pre-computed signals is also presented for comparison purpose.

3.2.1. Radix-2 FA with Pre-computed G/H Signals

A standard radix-2 FA with pre-computed carry acceleration signals is shown in Fig. 5. In this design the carry signal of an adder has to pass through a 6-transistor gate, which is faster than the previously designed 10-transistor circuitry.

Boolean functions for this architecture are given by equations (7)-(10). In this architecture G is computed as the generate function of a CLA, while we use the inverted kill function, denoted by H , instead of the more commonly used propagate function P .

$$G_0 = A.B \quad (7)$$

$$H_0 = A + B \quad (8)$$

$$\overline{C_{out}} = \overline{C_{in} \cdot H_0 + G_0} \quad (9)$$

$$\overline{S_0} = \overline{\overline{C_1} \cdot B_0 + \overline{C_1} \cdot C_{in} + \overline{C_1} \cdot A_0 + A_0 \cdot B_0 \cdot C_{in}} \quad (10)$$

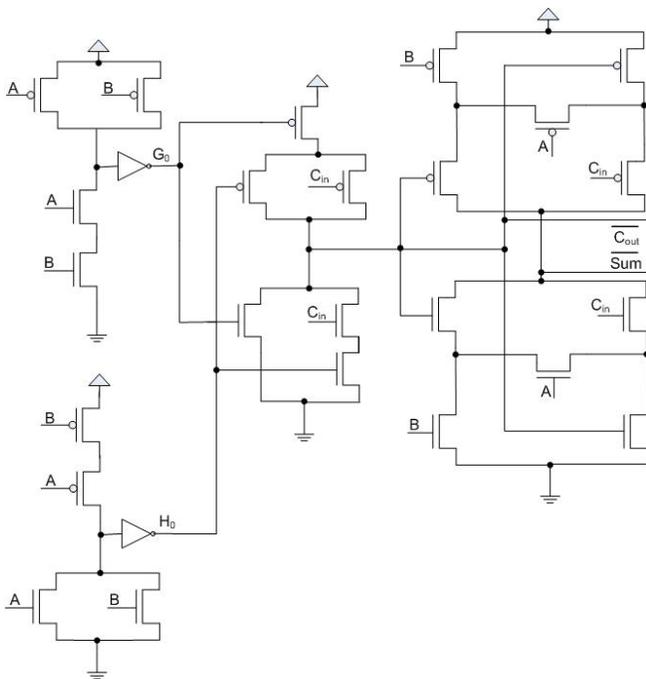


Fig. 5. Radix-2 FA with pre-computed G and H functions.

3.2.2. Radix-4 FA with Pre-computed G/H Signals

The proposed radix-4 FA with pre-computed G/H circuitry is shown in Fig. 6. Compared to the previous radix-4 FA design shown in Fig. 4, the circuitry for C_{out} is modified so that the carry of an adder has to pass through only a 6-transistor gate instead of an 18-transistor gate. Hence a speed advantage is obtained compared to the standard FA.

The Boolean equations for the circuit are

$$G = A_1 \cdot B_1 + (A_1 + B_1) \cdot A_0 \cdot B_0 \quad (11)$$

$$H = (A_1 + B_1) \cdot (A_0 + B_0) \quad (12)$$

$$C_{out} = C_{in} \cdot H + G \quad (13)$$

Equations for implementation of C_1 , S_0 and S_1 are same as given in Section 3.

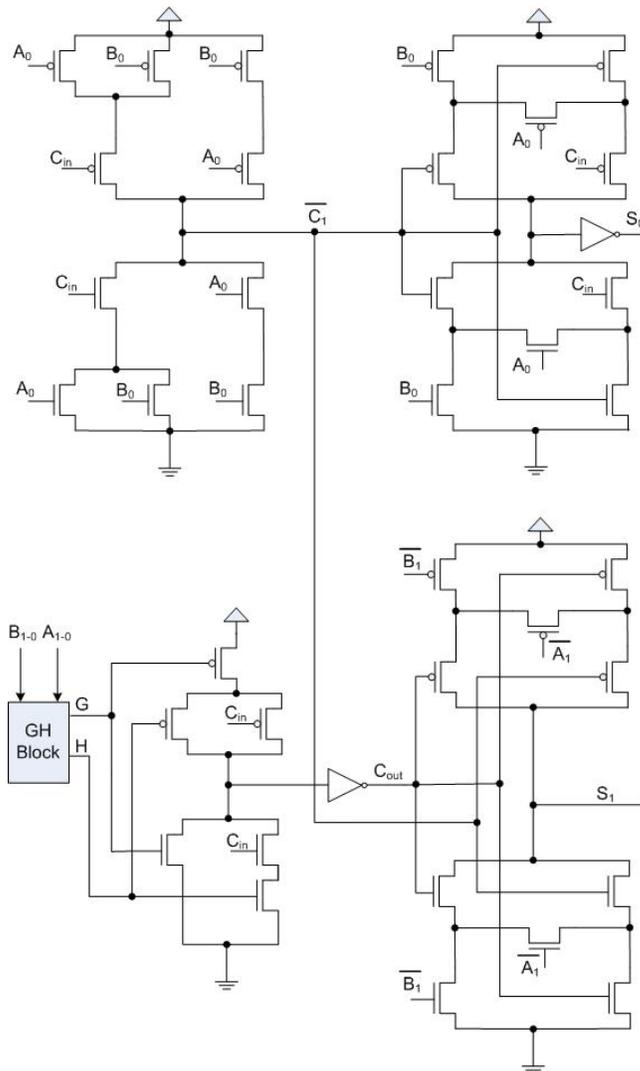


Fig. 6. Proposed radix-4 FA with explicit G/H signals.

The circuit for carry acceleration signal generation is shown in Fig. 7. A low transistor count is obtained by sharing a part of the circuit. Note the case when A_1 and B_1 are low, and A_0 and B_0 are high. In this case, the NMOS net is used to conduct a high voltage resulting in a reduced voltage swing. However, the inverters at the outputs provide full voltage swing to subsequent circuits. The circuit becomes slower by the sharing, but fortunately it does not affect the critical path. Another drawback is the reduced noise margin, which may require a full static CMOS gate for low supply voltages.

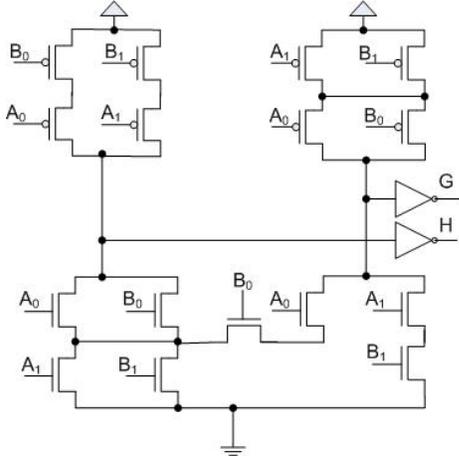


Fig. 7. G/H pre-computation circuit.

3.3. Radix-4 FA vs. Standard 2-bit CLA

A 2-bit standard CLA is implemented for comparison with the Radix-4 FAs. Block diagram for this 2-bit CLA is shown in Fig. 8.

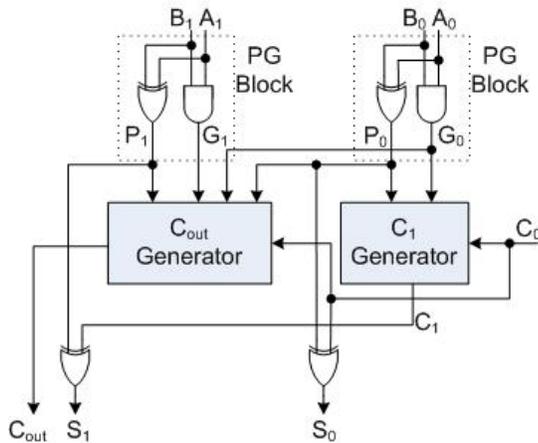


Fig. 8. Standard 2-bit CLA

The circuit is implemented using complementary CMOS logic style and is realized by equations (14) – (18).

$$G_i = A_i \cdot B_i \quad (14)$$

$$P_i = \overline{A_i} \cdot B_i + A_i \cdot \overline{B_i} \quad (15)$$

$$S_i = \overline{P_i} \cdot C_i + P_i \cdot \overline{C_i} \quad (16)$$

$$C_1 = G_0 + P_0 \cdot C_0 \quad (17)$$

$$C_{out} = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0 \quad (18)$$

This 2-bit CLA uses two PG blocks whereas pre-computed radix-4 FA uses only one pre-computation block named as ‘GH Block’. Moreover, the complexity of C_{out} circuit is reduced in Radix-4 FA by using the ‘not kill (H)’ instead of commonly used ‘propagate (P)’ signal. Table 4 shows that the radix-4 FAs are not only faster than the 2-bit CLA, but also better in terms of energy and transistor count. Comparing the pre-computed radix-4 FA with the basic radix-4 FA, the speed advantage of the former is even larger, but at the cost of slightly higher energy per operation and transistor count. The increased number of transistors will increase the area of the circuit, but since the different designs provide the same regularity, the impact on the layout complexity should be small.

TABLE 4
RESULTS FOR 2-BIT CLA AND RADIX-4 FA

	Basic radix-4 FA	Pre-computed radix-4 FA	2-bit CLA
Technology	45 nm CMOS		
PMOS Width (μm)	0.2		
NMOS Width (μm)	0.1		
Supply Voltage (V)	1.1		
Propagation Delay (ps)	32.1	24.7	26.3
Transistor Count	56	64	90
Energy/Operation (fJ)	7.18	8.57	15.07

4. IMPLEMENTATION OF 32-BIT ADDERS WITH BASIC FA

To evaluate the performance of the proposed circuit, we use it to build 32-bit carry propagation adders. Investigated adders are the ripple carry, carry select, and carry skip adders.

4.1. Ripple Carry Adder

A ripple carry adder (RCA) is used when we have area constraints, and when the speed is of little importance [6]. The standard scheme of an RCA is shown in Fig. 9.

In order to optimize the RCA delay, we can use the inverter elimination scheme in the carry path [20]. Inverter elimination schemes for radix-2 and radix-4 RCAs are shown in Fig. 10 and Fig. 11, respectively.

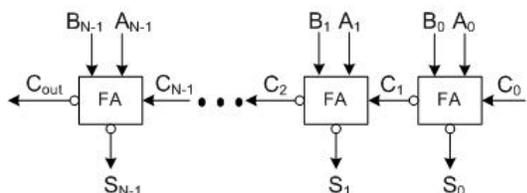


Fig. 9. Standard scheme of RCA.

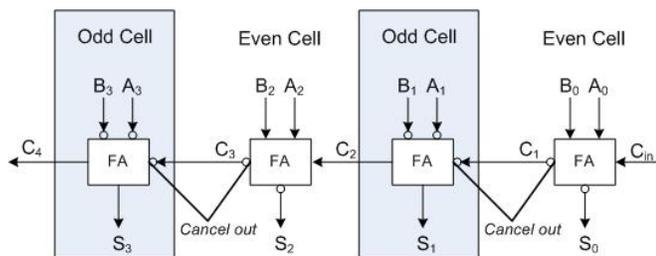


Fig. 10. Inverter elimination in carry path for radix-2 RCA.

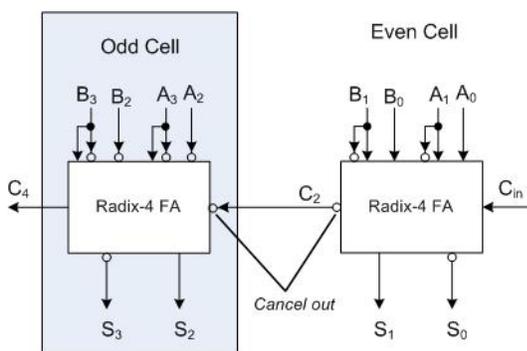


Fig. 11. Inverter elimination in carry path for radix-4 RCA.

With this scheme, the delay and power consumption of the RCA are significantly reduced as compared to when we use the standard scheme. We used both the standard and the inverter elimination scheme in our experiments, and refer to these in the following as standard scheme and inverter-eliminated scheme, respectively.

4.2. Carry Select Adder

A carry select adder (CSEL) is commonly used when we need to compromise between speed and area [2]. A CSEL can be constructed using constant block size, or a variable block size. While the implementation of constant block size is less complex, the variable block size gives higher performance than the constant block size. We implemented both types for comparison purpose.

In the case of constant block size, we implemented a 2-stage CSEL using the radix-4 FA to compare its performance with a radix-2 2-stage CSEL. The block diagram for the 2-stage CSEL is shown in Fig. 12.

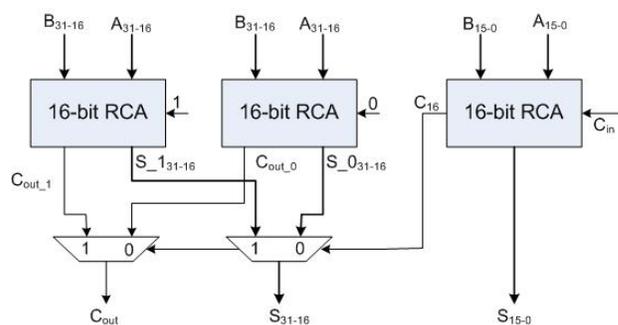


Fig. 12. 2-stage CSA.

We use the strategy presented in [2] to construct a variable block size CSEL optimized for speed. The size of first and second block is computed by equation (19).

$$M_{1,2opt} = \frac{\alpha + \beta}{\ln(1 - \beta) \cdot (2 \cdot \beta - 1)} - \frac{\alpha + \beta}{\beta} \quad (19)$$

where $M_{1,2opt}$ is rounded to nearest integer. In (19)

$$\alpha = \frac{\tau_{int}}{\tau_{carry}}, \text{ and } \beta = \frac{\tau_{FO}}{\tau_{carry}}$$

where τ_{int} is the intrinsic delay of the multiplexer, τ_{FO} is increase in multiplexer delay when fan-out is increased by 1, and τ_{carry} is the carry propagation delay of a FA.

Equation (20) is used to find the size for the remaining blocks.

$$M_i = \left\lceil \frac{M_{1,2opt} + \beta \sum_{j=3}^{i-1} M_j + (i-2)(\alpha + \beta)}{1 - \beta} \right\rceil \quad (20)$$

for $i = 3 \dots Q$, where Q is the total number of blocks in a CSEL.

The propagation delay of this optimum CSEL can be found by equation (21).

$$\tau_{PD} = t_{sel,Q} + \tau_{MUX,Q} \quad (21)$$

$t_{sel,i}$ can be found by equation (22).

$$t_{sel,i} = \max(t_{in,i-1}, t_{sel,i-1}) + \tau_{MUX,i-1} \quad (22)$$

and

$$t_{in,i} = M_i \tau_{carry}$$

Values for τ_{int} and τ_{FO} for our multiplexer are 10.7 ps and 1.93 ps, respectively.

4.2.1. Constructing an optimum CSEL using radix-2 FA

The total number of blocks required to construct the adder is represented by N , where in this case $N = 32$. Values for α and β are 0.528 and 0.0955, respectively. Block sizes for the adder are obtained from equations (19) and (20) and given in Table 5.

TABLE 5
TIMING ANALYSIS OF A NEARLY OPTIMUM CSEL

Group	$t_{in,i}/\tau_{carry} = M_i$	$t_{sel,i}/\tau_{carry}$
1	1	-
2	1	1
3	1	1.72
4	2	2.53
5	3	3.44
6	4	4.45
7	5	5.55
8	6 (7)	6.75 (6.84)
9	8	8.14

We obtain a nearly optimum CSEL [2] with $N^* = 31$. The remaining one bit has to be added in one of the blocks. Timing analysis for this nearly optimum CSEL is given in Table 5. Arrival times for input and select signals are normalized with respect to τ_{carry} .

By inspection of Table 5, we see that the remaining bit should be added to 8th block, because it has the largest difference in arrival time of input and select signal, $t_{sel,8} - t_{in,8} = 0.75$ ps. Finally, an optimum CSEL is constructed. The modified values are shown within parentheses in Table 5.

4.2.2. Constructing an optimum CSEL using radix-4 FA

In the proposed radix-4 FA, we need only 16 blocks to construct a 32-bit adder, hence $N = 16$. Values for α and β are 0.445 and 0.08, respectively.

We find the block sizes of the adder from equations (19) and (20). They are given in Table 6.

TABLE 6
TIMING ANALYSIS OF A NEARLY OPTIMUM CSEL

Group	$t_{in,i}/\tau_{carry_radix-4_FA} = M_i$	$t_{sel,i}/\tau_{carry_radix-4_FA}$
1	1	-
2	1	1
3	1	1.61
4	2	2.29
5	3	3.06
6	3 (4)	3.83 (3.90)
7	4	4.68

This is a nearly optimum CSEL with $N^* = 15$. We need to increase the size of one of the blocks to get the complete adder. The timing analysis for this nearly optimum CSEL is given in Table 6. Arrival times for input and select signals are normalized with respect to τ_{carry} .

By inspection of Table 6, we see that the remaining bit should be added to 6th block, because it has the largest difference in arrival time of input and select signal, $t_{sel,6} - t_{in,6} = 0.83$ ps. Finally, an optimum CSEL is constructed. The modified values are shown within parentheses in Table 6.

4.3. Carry Skip Adder

The carry skip adder (CSKA) is well-known for its low PDP [21]. It consists of FA blocks that can be fixed or variable size. A CSKA with a variable block size perform much better than with fixed block size. The general structure of a CSKA is shown in Fig. 13.

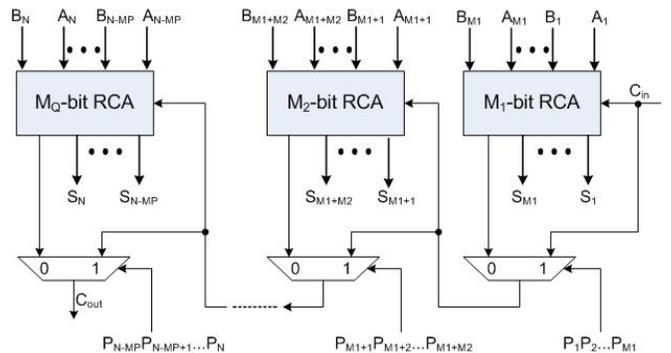


Fig. 13. Carry skip adder.

We use the strategy presented in [21] for implementation of CSKAs using the radix-2 FA, and the radix-4 FA. Optimum size for the first and last block can be found by equation (23).

$$M_{1,opt} = \frac{\alpha}{2} \quad (23)$$

where

$$\alpha = \tau_{MUX} / \tau_{carry}$$

To find the sizes for the remaining blocks, we use equation (24).

$$M_j = \lfloor M_1 + (j-1)\alpha \rfloor \quad (24)$$

Generally, sizing the blocks with this procedure results in an adder with N^* bits, which are lower than N . Then, we increase the block sizes appropriately to construct the adder with required number of bits. A CSKA obtained by this procedure is symmetrical, which means it is sufficient to find the block sizes only for left side.

Propagation delay of the designed CSKA can be found by equation (25).

$$\tau_{PD} = (2M_1 - 1)\tau_{carry} + (Q - 1)\tau_{MUX} + \tau_{sum} + \Delta\tau_{p,tot} \quad (25)$$

where

M_1 is the size for first block,

Q is the number of total blocks in the CSKA,

τ_{MUX} is propagation delay of a multiplexer,

τ_{sum} is sum propagation delay of the FA, and

$\Delta\tau_{p,tot}$ is adder delay increase with respect to the nearly optimum CSKA when some bits are added to it.

4.3.1. Constructing an optimum CSKA using radix-2 FA

The total number of blocks required to construct the 32-bit adder is $N = 32$. The value of α is 0.681. From equations (24) and (25), we find the block sizes $M_1 = 1, M_2 = 1, M_3 = 2, M_4 = 3, M_5 = 3, M_6 = 4$. We have to stop at block 6, because adding another block will result in a total number of bits for one side greater than $N/2 = 16$. This is a nearly optimum CSKA with $N^* = 28$.

Since $N - N^* = M_{Q/2}$, we add another block between the two sides with the same size as that of $M_{Q/2}$, resulting in a three bit nucleus [21] with $M_7 = 4$. Blocks on the right side are symmetrical to left side with sizes $M_8 = 4, M_9 = 3, M_{10} = 3, M_{11} = 2, M_{12} = 1, M_{13} = 1$. Thus, our optimum CSKA is complete containing 13 blocks.

4.3.2. Constructing an optimum CSKA using radix-4 FA

Using the proposed radix-4 FA, we need only $N = 16$ blocks to construct a 32-bit adder. The value of α is 0.6. The block sizes obtained by equations (23) and (24) are $M_1 = 1, M_2 = 1, M_3 = 2, M_4 = 2$. We have to stop at block 4, because adding another block will result in a total number of bits for one side greater than 8. We get a nearly optimum CSKA with $N^* = 12$.

Since $N - N^* > M_{Q/2}$, we add another block between the two sides. The size of this block is determined by equation (24), and it represents the nucleus [21] with $M_5 = 3$. Blocks on the right side are symmetrical to the left side with sizes $M_6 = 2, M_7 = 2, M_8 = 1, M_9 = 1$.

We have constructed a nearly optimum CSKA having 30 bits and now we do need to add one more FA block. These remaining 2 bits must be added to the block with the minimum $\Delta\tau_p$. Hence we check for the suitable position to insert the remaining bits to complete the 32-bit adder. This limits our search to choose from M_2 and M_4 , and symmetrically, M_8 and M_6 , respectively. $\Delta\tau_p$ for these blocks are

$$\Delta\tau_{p,2} = \Delta\tau_{p,8} = 8.4 \text{ ps}$$

$$\Delta\tau_{p,4} = \Delta\tau_{p,6} = 4.2 \text{ ps} .$$

$\Delta\tau_{p,4}$ and $\Delta\tau_{p,6}$ are lower than $\Delta\tau_{p,2}$ and $\Delta\tau_{p,8}$, so we can add another FA to either M_4 or M_6 . We add an FA to M_4 to complete our optimum CSKA with 9 blocks. The block sizes are $M_1 = 1, M_2 = 1, M_3 = 2, M_4 = 3, M_5 = 3, M_6 = 2, M_7 = 2, M_8 = 1, M_9 = 1$.

5. IMPLEMENTATION OF 32-BIT ADDERS WITH PRE-COMPUTED FA

The previous set of 32-bit adders has also been implemented with the pre-computed FAs to evaluate their performance. We use the same scheme of RCA as described in Section 4.1. However, we need to re-design the optimized CSEL and CSKA due to the change in delay of the carry signal.

5.1. Constructing an Optimum CSEL using Radix-2 and Radix-4 FA

The new values of α and β for radix-2 FA are 0.5556 and 0.1005, respectively. Repeating the procedure described in Section 4.2, we get the group sizes $M_1 = 1, M_2 = 1, M_3 = 1, M_4 = 2, M_5 = 3, M_6 = 4, M_7 = 5, M_8 = 7, M_9 = 8$, which are the same as obtained in Section 4.2.1.

The new values of α and β for the radix-4 FA are 0.63497 and 0.1149, respectively. Repeating the procedure described in Section 4.2, we get the group sizes $M_1 = 1, M_2 = 1, M_3 = 2, M_4 = 3, M_5 = 4, M_6 = 5$.

5.2. Constructing an Optimum CSKA using Radix-2 and Radix-4 FA

The new value of α for the radix-2 FA is 0.7199. Repeating the procedure described in Section 4.3, we get the group sizes $M_1 = 1, M_2 = 1, M_3 = 2, M_4 = 3, M_5 = 3, M_6 = 4, M_7 = 4, M_8 = 4, M_9 = 3, M_{10} = 3, M_{11} = 2, M_{12} = 1, M_{13} = 1$, which are same as obtained in Section 4.3.1.

The new value of α for the radix-4 FA is 0.884. Repeating the procedure described in Section 4.3, we get the group sizes $M_1 = 1, M_2 = 2, M_3 = 2, M_4 = 3, M_5 = 3, M_6 = 2, M_7 = 2, M_8 = 1$.

6. SIMULATION RESULTS

6.1. Simulation Parameters

In this section we present the simulation results of the different adders that we compare. All the circuits are simulated in Spice using 45 nm Predictive Technology Model (PTM) for metal gate/high-k [18]. A power supply voltage of 1.1 V is used for the simulations. The widths for all NMOS and PMOS transistors are set to 0.1 μm and 0.2 μm , respectively. An inverter is used as load at the outputs for all designs. The simulation parameters are given in Table 7.

TABLE 7
SIMULATION PARAMETERS

Technology	45 nm CMOS
PMOS Width (μm)	0.2
NMOS Width (μm)	0.1
Supply Voltage (V)	1.1

For measuring propagation delay we need to set the inputs such that the carry propagates from LSB to MSB [22]. We set $A_{31-0} = 1$ and $B_{31-0} = 0$ to measure the worst-case delay. Then, for t_{phl} we set $C_{in} = 0$, and for t_{plh} we set $C_{in} = 1$.

Random inputs are applied to the adders to verify the results and to estimate the power consumption at f_{max} for each design. The switching activity of the inputs A and B are 0.23 and 0.26, respectively. A pseudo-random binary sequence generator (PRBS) of Galois type is used to provide the data pattern. The generator polynomial for this PRBS is $1 + x^{25} + x^{26} + x^{30} + x^{32}$, which generates a sequence of

length $2^{32} - 1$ [23]. Two different seeds are used for the two PRBS for input A and B . The switching activity of the output is 0.23.

For simulation of leakage power, we applied two inputs. The first input is $A_{31-0} = 0, B_{31-0} = 0$ and $C_{in} = 0$, and the second is $A_{31-0} = 1, B_{31-0} = 1$ and $C_{in} = 1$.

6.2. Simulation Results for Adders with Basic FA

Table 8 shows the simulation results for the 32-bit adders with basic FA. The radix-4 RCA is 1.48 and 1.20 times faster than the radix-2 RCA using standard and inverter eliminated schemes, respectively. The PDP of the radix-4 RCA using standard scheme is 19% lower than the radix-2 RCA, but when we compare the RCAs using inverter eliminated scheme, the radix-4 RCA has 17% higher PDP as compared to the radix-2 RCA. This is because the delay of the radix-4 RCA increases due to the domination of parasitic capacitances. Comparing EDP, standard RCA yields 45% lower EDP for the radix-4 RCA compared to the radix-2 RCA, and inverter eliminated RCA yields 3% lower EDP for the corresponding case.

The radix-4 RCA with the best PDP and EDP values is implemented with standard scheme, while the radix-2 RCA with the best PDP and EDP values is implemented with inverter eliminated scheme. Comparing these designs, the radix-4 RCA with standard scheme has a PDP that is 11% higher and an EDP that is 22% lower than the radix-2 RCA with inverter eliminated scheme.

A comparison of the 2-stage CSEL adders shows that the radix-4 CSEL is 1.38 times faster, has 12% lower PDP, and 36% lower EDP compared to the radix-2 CSEL. Similarly, the radix-4 variable size CSEL is 1.13 times faster with 10% lower PDP and 21% lower EDP than the radix-2 variable size CSEL.

Similarly the radix-4 CSKA is 1.25 times faster, has a PDP 12% less, and an EDP 45% less than the radix-2 CSKA.

For the 32-bit adders with basic FA, Fig. 14(a) shows the PDP of the radix-2 and radix-4 architectures, Fig. 15(a) shows the EDP of the radix-2 and radix-4 architectures, and Fig. 16(a) shows the power dissipation due to leakage.

The leakage power of the different adders with basic FA is plotted in the graph shown in Fig. 16 (a). Comparing leakage power, the radix-2 RCA implemented with inverter elimination scheme has the lowest leakage, followed by the radix-4 RCA implemented with standard scheme, which has 12% higher leakage power dissipation than the radix-2 RCA architecture using inverter elimination scheme.

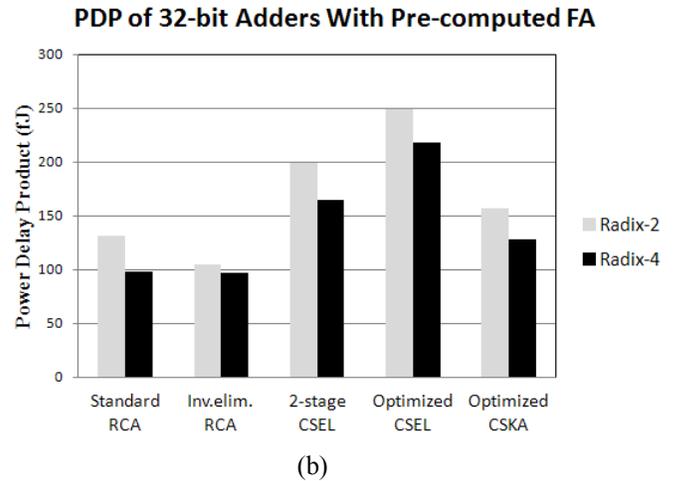
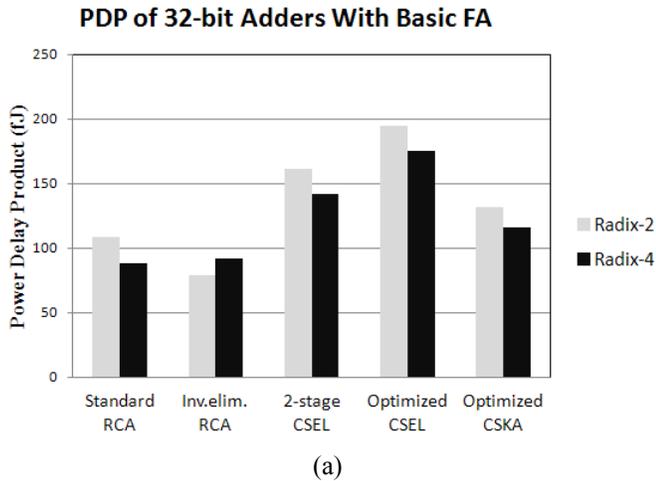


Fig. 14. PDP of radix-2 and radix-4 designs with (a) basic FA, and (b) pre-computed FA.

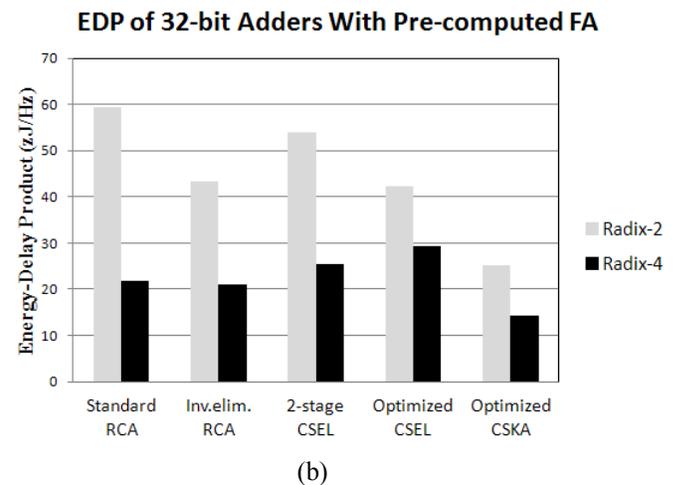
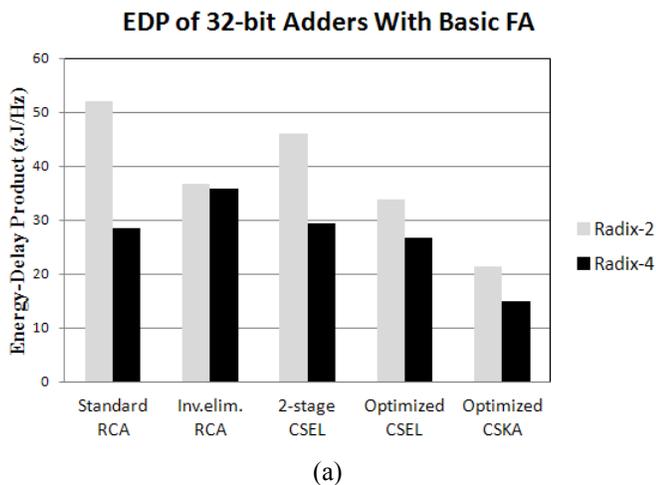


Fig. 15. EDP of radix-2 and radix-4 designs with (a) basic FA, and (b) pre-computed FA.

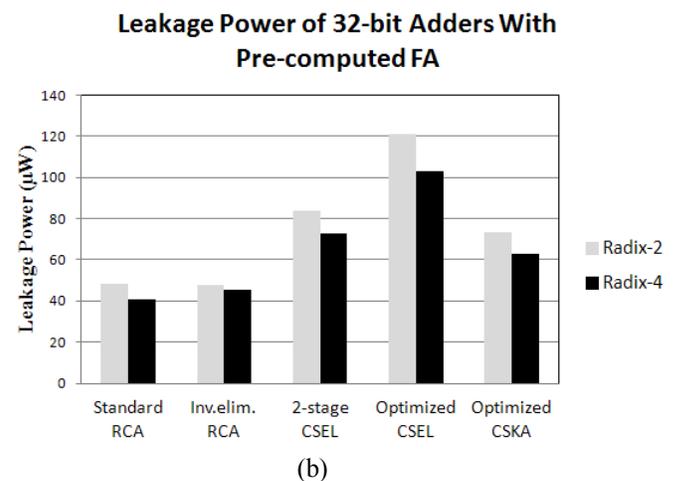
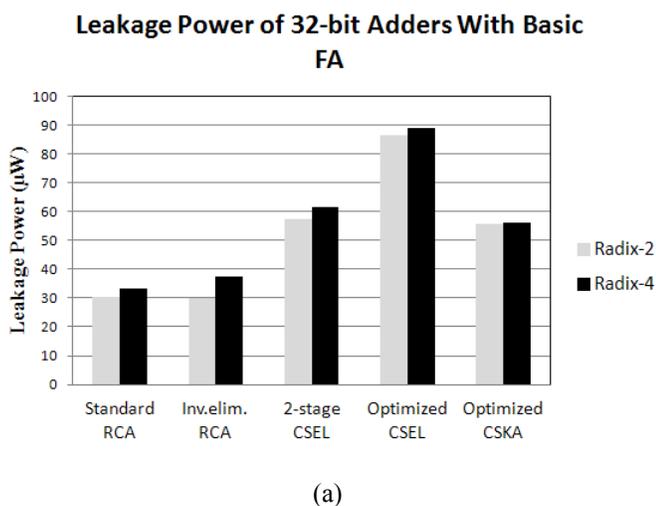


Fig. 16. Average leakage power of radix-2 and radix-4 designs with (a) basic FA, and (b) pre-computed FA.

TABLE 8
RESULTS FOR 32-BIT ADDERS WITH BASIC FA

	Radix	Delay (ps)	Device Count	Power (μ W)	PDP (fJ)	EDP (zJ/Hz)
Standard RCA	2	480	768	226	108	52.0
	4	324	896	272	88	28.5
Inv.elim. RCA	2	467	736	169	79	36.8
	4	389	928	236	92	35.8
2-stage CSEL	2	285	1356	567	162	46.1
	4	207	1536	686	142	29.4
Var. size CSEL	2	173	1980	1129	195	33.8
	4	153	2168	1144	175	26.8
Var. size CSKA	2	162	1372	817	132	21.4
	4	130	1452	890	116	15.0

The leakage power of the radix-4 2-stage CSEL is 8% higher than radix-2 2-stage CSEL. Similarly the radix-4 variable size CSEL has 3% higher leakage power dissipation as compared to radix-2 variable size CSEL.

The leakage power of the variable size CSKA is approximately the same for both the radix-4 and radix-2 designs.

6.3. Simulation Results for Adders with Pre-computed FA

Table 9 shows the simulation results for the 32-bit adders with pre-computed FA. The radix-4 RCA is 2.01 and 1.90 times faster than the radix-2 RCA using standard and inverter-eliminated schemes, respectively. Similarly, the PDP of the radix-4 RCA is 26% and 8% lower than the radix-2 RCA, and the EDP is 63% and 51% lower than the radix-2 RCA, using standard and inverter-eliminated schemes, respectively.

A comparison of the 2-stage CSEL shows that the radix-4 CSEL is 1.75 times faster, has 17% lower PDP, and 53% lower EDP compared to the radix-2 CSEL. Similarly, the radix-4 variable size CSEL is 1.27 times faster with 13% lower PDP and 31% lower EDP than the radix-2 variable size CSEL. Comparing the CSEL adders, we see that the variable size CSEL has higher PDP and EDP values than the 2-stage CSEL adders.

Similarly the radix-4 CSKA is 1.44 times faster as compared to the radix-2 CSKA. The radix-4 CSKA also has 18% lower PDP, and 43% lower EDP as compared to radix-2 CSKA.

For the adders with pre-computed FA, Fig. 14(b) shows the comparison of PDP of the radix-2 and radix-4

TABLE 9
RESULTS FOR 32-BIT ADDERS WITH PRE-COMPUTED FA

	Radix	Delay (ps)	Device Count	Power (μ W)	PDP (fJ)	EDP (zJ/Hz)
Standard RCA	2	451	1024	292	132	59.3
	4	224	1024	435	98	21.8
Inv.elim. RCA	2	414	992	253	105	43.4
	4	218	1056	444	97	21.1
2-stage CSEL	2	271	1740	736	199	54.0
	4	155	1740	1060	164	25.5
Var. size CSEL	2	171	2496	1451	249	42.4
	4	135	2404	1611	218	29.4
Var. size CSKA	2	161	1628	975	157	25.3
	4	112	1568	1143	128	14.3

architectures, Fig. 15(b) shows the comparison of EDP of the radix-2 and radix-4 architectures, and Fig. 16(b) shows the comparison of power dissipation due to leakage.

The leakage power of the different adders with pre-computed FA is plotted in the graph shown in Fig. 16 (b). Comparing the leakage power, the radix-4 RCA implemented with standard scheme has the best performance, which is 14% lower than the radix-2 RCA implemented with inverter-eliminated scheme.

The leakage power of radix-4 2-stage CSEL is 13% lower than the radix-2 2-stage CSEL. Similarly, the radix-4 variable size CSEL has 15% lower leakage power dissipation as compared to the radix-2 variable size CSEL.

The leakage power of the radix-4 CSKA is 15% lower as compared to the radix-2 CSKA.

7. CONCLUSION

In this work we propose a method of reducing the power-delay and energy-delay products of an adder by reducing the carry propagation. The method is based on the carry look-ahead technique, where we compute the carry without waiting for the carry out of the previous stage. We implement a radix-4 FA block by joining two FAs. Within this adder we used the carry look-ahead technique to compute the results for the second sum bit, and thus avoiding carry propagation. The disadvantage of this technique is a slightly larger transistor count. A standard 2-bit CLA is also implemented to show the performance improvement of radix-4 adder over 2-bit CLA in terms of speed and power dissipation. Simulation results show that the proposed radix-4 adder has lower energy dissipation per operation as compared to 2-bit CLA.

For estimating the performance, we simulate the designs in Spice using a 45 nm technology with a power supply of 1.1 V. A comparison of adders with a basic FA type shows that the radix-4 RCA, 2-stage CSEL, variable size CSEL, and CSKA have a relative PDP of +11%, -12%, -10%, and -12%, and a relative EDP of -22%, -36%, -21%, and -30% compared to radix-2 RCA, 2-stage CSEL, variable size CSEL, and CSKA, respectively. When comparing adders with a pre-computed FA type, the radix-4 RCA, 2-stage CSEL, variable size CSEL, and CSKA have -8%, -17%, -13%, and -18% relative PDP and -51%, -53%, -31%, and -43% relative EDP compared to radix-2 RCA, 2-stage CSEL, variable size CSEL, and CSKA, respectively.

Based on these results, we conclude that our proposed radix-4 FA gives better performance than a standard radix-2 FA.

A comparison of the two designs of radix-4 adders shows that the PDP is -10%, -16%, -24%, and -11%, and the EDP is +26%, +13%, -10%, and +5% for the radix-4 RCA, 2-stage CSEL, variable size CSEL, and CSKA adders with basic FA relative to the adders with pre-computed FA, respectively. Hence we can conclude that the radix-4 adders with basic FA is better than the adders with pre-computed FA in terms of PDP, but in terms of EDP the radix-4 adders with pre-computed FA has better performance on average.

REFERENCES

- [1] A. Neve, H. Schettler, T. Ludwig, and D. Flandre, "Power-delay product minimization in high-performance 64-bit carry-select adders," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 12, no. 3, pp. 235-244, March 2004.
- [2] M. Alioto, G. Palumbo, and M. Poli, "A gate-level strategy to design carry select adders," in *Proc. Int. Symp. Circuits and Systems*, May 2004, vol. 2, pp. 465-8.
- [3] J.M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective, 2nd Ed.*, Prentice-Hall, 2003.
- [4] N.H.E. Weste and D.M. Harris, *Integrated Circuit Design, 4th Ed.*, Pearson Education, 2010.
- [5] Yan Sun, Xin Zhang, and Xi Jin, "High-performance carry select adder using fast all-one finding logic," in *Proc. Second Asia Int. Conf. Modeling & Simulation*, May 2008, pp. 1012-1014.
- [6] B. Amelifard, F. Fallah, and M. Pedram, "Closing the gap between carry select adder and ripple carry adder: a new class of low-power high-performance adders," in *Proc. Sixth Int. Symp. Quality Electronic Design*, March 2005, pp. 148-152.
- [7] Sun Yan, Zhang Xin, and Jin Xi, "Low-power carry select adder using fast all-one finding logic," in *IEEE Int. Conf. System of Systems Engineering*, 2008, pp. 1-5.
- [8] R. Shalem, E. John, L.K. John, "A novel low power energy recovery full adder cell," In *Proc. Ninth Great Lakes Symp. on VLSI*, Mar. 1999, pp. 380-383.
- [9] A. Shams and M. Bayoumi, "A novel high-performance CMOS 1-bit full-adder cell," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 478-481, May 2000.
- [10] J.-F. Lin, Y.-T. Hwang, M.-H. Sheu, and C.-C.Ho, "A novel high-speed and energy efficient 10-transistor full adder design," *IEEE Trans. Circ. Syst. I: Reg. Papers*, vol. 54, no. 5, pp. 1050-1059, May 2007.
- [11] I. Hassoune, D. Flandre, I. O'Connor, and J.-D. Legat, "ULPFA: a new efficient design of a power-aware full adder," *IEEE Trans. Circ. Syst. I: Reg. Papers*, vol. 57, no. 8, pp. 2066-2074, Aug. 2010.
- [12] M. Alioto and G. Palumbo, "Analysis and comparison on full adder block in submicron technology," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 6, pp. 806-823, Dec 2002.
- [13] Frank K. Gurkaynad, Yusuf Leblebicit, Laurent Chaouatt and Patrik J. McGuinnessz, "Higher Radix Kogge-Stone Parallel Prefix Adder Architectures," in *Proc. of ISCAS 2000*
- [14] R. Zlatanovici and B. Nikolic, "Power - Performance Optimal 64-Bit Carry-Lookahead Adders," in *Proc. of the 29th European Solid-State Circuits Conference*, 2003. ESSCIRC '03.
- [15] M. Alioto, L. Pancioni, S. Rocchi, and V. Vignoli, "Power-delay-area-noise margin tradeoffs in positive-feedback MOS current-mode logic," *IEEE Trans. Circ. Syst. I: Reg. Papers*, vol. 54, no. 9, pp. 1916-1928, Sept. 2007.
- [16] M. Anis, M. Allam, and M. Elmasry, "Impact of technology scaling on CMOS logic styles," *IEEE Trans. Circ. Syst. II: Analog Digital Signal Processing*, vol. 49, no. 8, pp. 577-588, Aug. 2002.
- [17] S.M.Y. Sherazi, S. Asif, E. Backenius, and M. Vesterbacka, "Reduction of Substrate Noise in Sub Clock Frequency Range," *IEEE Trans. Circ. Syst. I: Reg. Papers*, vol. 57, no. 6, pp. 1287-1297, June 2010.
- [18] *Berkeley Predictive Technology Models (BPTM)*, 2008 [Online]. Available: <http://www.eas.asu.edu/~ptm/>
- [19] M.A. Manzoul, "Parallel CLA algorithm for fast addition," in *Proc. Int. Conf. Parallel Computing in Electrical Engineering*, 2000, pp. 55-58.
- [20] Youngjoon Kim and Lee-Sup Kim, "A low power carry select adder with reduced area," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2001, vol. 4, pp. 218-221.
- [21] M. Alioto and G. Palumbo, "A simple strategy for optimized design of one-level carry-skip adders," *IEEE Trans. Circ. Syst. I: Fundamental Theory Appl.*, vol. 50, no. 1, pp. 141-148, Jan. 2003.
- [22] K. Rawat, T. Darwish, and M. Bayoumi, "A low power and reduced area carry select adder," in *Proc. 45th Midwest Symp. Circuits and Systems*, Aug. 2002, vol. 1, pp. 467-70.
- [23] R. Ward and T. Molteno, *Table of linear feedback shift registers*, Dunedin, New Zealand: Datasheet, Department of Physics, University of Otago, 2007.