INTERNATIONAL JOURNAL OF CIRCUIT THEORY AND APPLICATIONS
Int. J. Circ. Theor. Appl. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cta.1886

Design of high-speed low-power parallel-prefix adder trees in nanometer technologies

Stefania Perri, Marco Lanuzza and Pasquale Corsonello*,†

Department of Electronics, Computer Science and Systems, University of Calabria, Arcavacata di Rende 87036 Rende, (CS) Italy

ABSTRACT

This paper presents a novel approach to design high-speed low-power parallel-prefix adder trees. Sub-circuits typically used in the design of parallel-prefix trees are deeply analyzed and separately optimized. The modules used for computing the group propagate and generate signals have been designed to improve their energy-delay behavior in an original way. When the ST 45 nm 1 V CMOS technology is used, in comparison with conventional implementations, the proposed approach exhibits computational delay with mean value and standard deviation up to 40% and 48% lower and achieves energy consumption with mean value and standard deviation up to 57% and 40% lower.

A 32-bit Brent-Kung tree made as proposed here reaches a computational delay lower than 165 ps and dissipates 147.4fJ on average.

Received 9 February 2012; Revised 19 November 2012; Accepted 21 November 2012

KEY WORDS: Parallel-Prefix adders; Brent-Kung adder tree; High-speed addition

1. INTRODUCTION

In the last few decades, the design of efficient addition circuits has received a great deal of attention, but it still remains a challenging proposition for several applications. Although hundreds of papers have been written in the past about adders, it is generally acknowledged that, when high speed is required, parallel-prefix trees are the best choice.

Besides the adder topology, also selecting the logic style plays a crucial role. In fact, on the one hand, static CMOS circuits achieve low-power consumption and high robustness; on the other hand, dynamic CMOS circuits offer higher speed and less complex networks. As deeply discussed and demonstrated in [1–7], static circuits are preferable when the performance target is relaxed, and limiting the energy consumption is the main objective. Conversely, in high-performance designs, dynamic circuits are typically required. Domino and domino-compound logics are most often employed in dynamic CMOS implementations [1, 2, 5–7]. In a domino circuit, a dynamic gate is always followed by a static inverter, whereas in a domino-compound circuit, a dynamic gate is followed by a more complex static gate.

Parallel-prefix adder trees are usually implemented by using a certain number of replicas of simple modules. Each module impacts on the overall performance of the adder depending on the chosen tree architecture. Therefore, using the same strategy for all modules might not lead to the desired result. This paper proposes a novel optimized implementation of the most recurring logic module, typically used within all types of parallel-prefix adders. Furthermore, guidelines for choosing the best dynamic

*Correspondence to: Pasquale Corsonello, Department of Electronics, Computer Science and Systems, University of Calabria, Arcavacata di Rende - 87036 - Rende (CS), Italy.
†E-mail: p.corsonello@unical.it

CMOS circuit design style are furnished, taking into account also the effects induced by process variations [8]. The latter are among major design challenges since they can significantly affect both speed and energy performances, thus causing unexpected behavior in manufactured circuits [9–13].

The basic ideas exploited in the novel implementation are: (1) reducing the complexity of the pull-down networks (PDNs) of each dynamic gate; and (2) minimizing the number of dynamic nodes within the overall structure of the generic parallel-prefix tree. When created using the ST 45 nm 1 V process technology, with respect to conventional implementations, the novel circuits exhibit mean values of computational delay and energy consumption up to 40% and 57% lower with standard deviations up to 48% and 40% lower.

The proposed design approach has been applied to the design of a 32-bit Brent-Kung parallel-prefix tree in order to prove the actual advantages it offers. Comparison with its conventional domino CMOS counterpart demonstrated that, owing to the innovations introduced, ~30% lower computational delay is achieved with ~41% lower energy consumption and ~51% lower silicon area requirement.

The paper is organized as follows: in Section II, a brief background on the parallel-prefix adder trees is provided; the novel circuits are then described in Section III; finally, characterization and comparison results are presented and discussed in Section IV.

2. CONVENTIONAL PARALLEL-PREFIX ADDER TREES

With $A = a_{n-1} \ldots a_0$ and $B = b_{n-1} \ldots b_0$ being two n-bit inputs, a parallel-prefix adder computes the sum $S = s_{n-1} \ldots s_0$ through the following three steps: (1) the preprocessing stage computes the auxiliary signals propagate and generate; (2) the carry propagation stage computes the carries, groups the propagate and generate signals r by r, with r being the radix of the adder; (3) the produced carries are then used by the final stage to calculate the sum bit s_i.

Several topologies of parallel-prefix adder trees are known, such as the Kogge-Stone [14], Ladner and Fischer [15], Ling [16], Brent-Kung [17], Han Carlson [18] and many others. As an example, the Kogge-Stone tree reaches the minimum logic depth using a very regular structure with uniform fan-out, but has a drawback in that it is very dense, thus requiring a large number of gates and wires. On the contrary, sparse trees, like the Brent-Kung, do not assure obtaining the minimum logic depth, but they save hardware resources and power, and, in any case, allow high-speed performances to be reached.

Usually, an n-bit radix-r sparse tree computes only $k = \frac{n-r}{r}$ carry signals c_x, with x being a multiple of r. Therefore, the final addition stage consists of $k + 1$ r-bit sum modules each receiving a c_x signal. In Figure 1, examples of 32-bit radix-4 parallel-prefix sparse trees are depicted, and their critical paths are highlighted with boldfaced lines. By observing these examples and others numerous sparse trees known in the literature, it can be easily seen that the basic modules needed to design a whole tree with any sparse architecture are those shown in Figure 1. The generic module within a triangle (in the following named LEV1) contains the gates required to compute propagate and generate signals at four consecutive bit positions and the circuits needed to group them as shown in equation (1), where $i = 0, 4, 8, \ldots, 24$. Note that the signal GG_0 furnishes the carry c_4 produced at the 4-th bit-position.

$$\begin{aligned}
&p_i = a_i + b_i \\
&g_i = a_i \cdot b_i \\
&\ldots \\
&p_{i+3} = a_{i+3} + b_{i+3} \\
&g_{i+3} = a_{i+3} \cdot b_{i+3} \\
&GP_{\frac{i}{4}} = p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot p_i \\
&GG_{\frac{i}{4}} = g_{i+3} + p_{i+3} \cdot g_{i+2} + p_{i+3} \cdot p_{i+2} \cdot g_{i+1} + p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot g_i
\end{aligned} \tag{1}$$

The generic module within the rhomb (named LEV2) receives four consecutive grouped propagate signals $(GP_{\frac{i}{4}+3}, GP_{\frac{i}{4}+2}, GP_{\frac{i}{4}+1}, GP_{\frac{i}{4}})$ and four consecutive grouped generate signals $(GG_{\frac{i}{4}+3}, GG_{\frac{i}{4}+2}, GG_{\frac{i}{4}+1}, GG_{\frac{i}{4}})$ as inputs and produces the signals *GGP* and *GGG* as given in equation (2).

DOI: 10.1002/cta

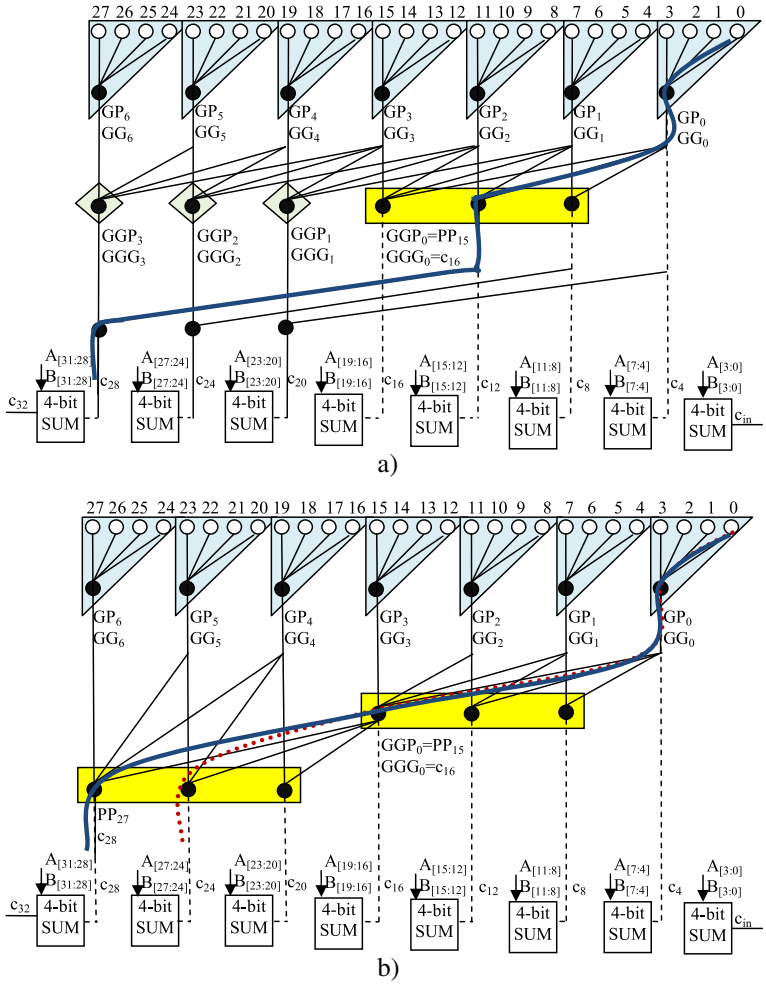


Figure 1. Examples of sparse parallel-prefix trees: a) Han Carlson; b) Brent Kung.

$$
\begin{aligned}
GGP_{\frac{i}{4}} &= GP_{\frac{i}{4}+3} \cdot GP_{\frac{i}{4}+2} \cdot GP_{\frac{i}{4}+1} \cdot GP_{\frac{i}{4}} \\
GGG_{\frac{i}{4}} &= GG_{\frac{i}{4}+3} + GP_{\frac{i}{4}+3} \cdot GG_{\frac{i}{4}+2} + GP_{\frac{i}{4}+3} \cdot GP_{\frac{i}{4}+2} \cdot GG_{\frac{i}{4}+1} \\
&\quad + GP_{\frac{i}{4}+3} \cdot GP_{\frac{i}{4}+2} \cdot GP_{\frac{i}{4}+1} \cdot GG_{\frac{i}{4}}
\end{aligned}
\tag{2}
$$

Finally, the modules within the rectangles visible in Figure 1 implement equations (3) and (4), where c_x, with $x = 8, 12, 16, 20, 24, 28$, is the carry-signal at the x-th bit-position. These circuits produce multiple outputs and in the following are named LEV3.

$$
\begin{aligned}
PP_{15} &= GP_3 \cdot GP_2 \cdot GP_1 \cdot GP_0 \\
c_{16} &= GG_3 + GP_3 \cdot GG_2 + GP_3 \cdot GP_2 \cdot GG_1 + GP_3 \cdot GP_2 \cdot GP_1 \cdot GG_0 \\
c_{12} &= GG_2 + GP_2 \cdot GG_1 + GP_2 \cdot GP_1 \cdot GG_0 \\
c_8 &= GG_1 + GP_1 \cdot GG_0
\end{aligned}
\tag{3}
$$

DOI: 10.1002/cta

$$\begin{aligned} PP_{27} &= GP_6 \cdot GP_5 \cdot GP_4 \cdot PP_{15} \\ c_{28} &= GG_6 + GP_6 \cdot GG_5 + GP_6 \cdot GP_5 \cdot GG_4 + GP_6 \cdot GP_5 \cdot GP_4 \cdot c_{16} \\ c_{24} &= GG_5 + GP_5 \cdot GG_4 + GP_5 \cdot GP_4 \cdot c_{16} \\ c_{20} &= GG_4 + GP_4 \cdot c_{16} \end{aligned} \tag{4}$$

Equations (1)–(4) are specialized for 32-bit radix-4 trees. However, they can be easily extended to different wordlengths and radices.

3. CIRCUITS IMPLEMENTATIONS

This section shows conventional domino and domino-compound gates used in radix-4 parallel-prefix adders. Then, the design strategies adopted in the new implementation are described.

3.1. Conventional designs

The conventional domino CMOS implementation of the basic modules LEV1, LEV2 and LEV3 can be easily done by using the sub-circuits illustrated in Figure 2 and following the above logic equations.

More explicitly, LEV1 is implemented by using the gates of Figures 2a, 2b, 2c and 2d and applying equation (1); to implement LEV2, the circuits illustrated in Figures 2c and 2d must be exploited as given in equation (2); finally, the module LEV3 is implemented combining the gates reported in Figures 2d and 2e as shown in equations (3) and (4). In the alternative implementations described below, a similar approach has to be used to realize the LEV1, LEV2 and LEV3 modules.

We will refer to this implementation as Conv_Dom. Transistor sizes reported in figures are referred to the ST 45 nm 1 V CMOS technology, in which the minimum transistor width is 0.12 um. The minimum-size criterion was applied to the elementary two-input OR and AND gates (Figure 2a–b), whereas the progressive transistor sizing with a 1.5 tapering factor was exploited within gates with higher fan-in and the Manchester carry-chain (MCC) (Figure 2c-e). Inverters on the dynamic nodes are minimum sized with an aspect ratio of 4/3 and include a minimum-sized keeper transistor (not drawn in the schematics). Clock buffers also illustrated in Figure 2f are sized for having their effective fan-out equal to 4 (the effective fan-out is defined as the ratio between the load capacitance and the input capacitance) [19].

Often, to simplify the PDNs, domino gates with limited stack height (LSH) are employed. In such a case, propagate and generate signals are grouped by means of the gates illustrated in Figure 3. In the following comparison, these implementations are named Conv_Dom_LSH.

An efficient alternative dynamic style is the domino-compound logic. This design style substitutes the output inverter of domino gates with more complex logic. In Figure 4, the domino-compound sub-circuits needed to implement the LEV1, LEV2 and LEV3 modules are illustrated. Below in the paper, this implementation is named Conv_CD. It should be noted that, with the exception of the circuit enclosed in the dashed box that is used only within LEV3 to compute the carry signals c_8 and c_{12}, the modules LEV2 and LEV3 use the same basic gates.

3.2. The novel designs

The novel design presented here is based on two simple conjectures: (1) the LEV1 is the most recurring stage in all parallel-prefix trees, therefore, improving its energy-delay behavior would have a significant impact on the overall circuit; (2) a unique CMOS style for all modules may not lead to an optimized design. To improve the energy-delay behavior, it should be taken into account that reducing the number of dynamic stages within each module will decrease the energy consumption, whereas, simplifying the PDNs of each dynamic stage would increase the speed performances.

The novelties introduced in the LEV1 circuit are the following: (1) the novel circuit computes only one generate and one propagate signal for each two bit positions; (2) the new LEV1 is based on logic equations that differ from the original ones previously reported in equation (1).

To understand this, let us consider the logic function implemented in the generic LEV1 module and above shown in equation (1). Four two-input OR and four two-input AND gates are required for

DOI: 10.1002/cta

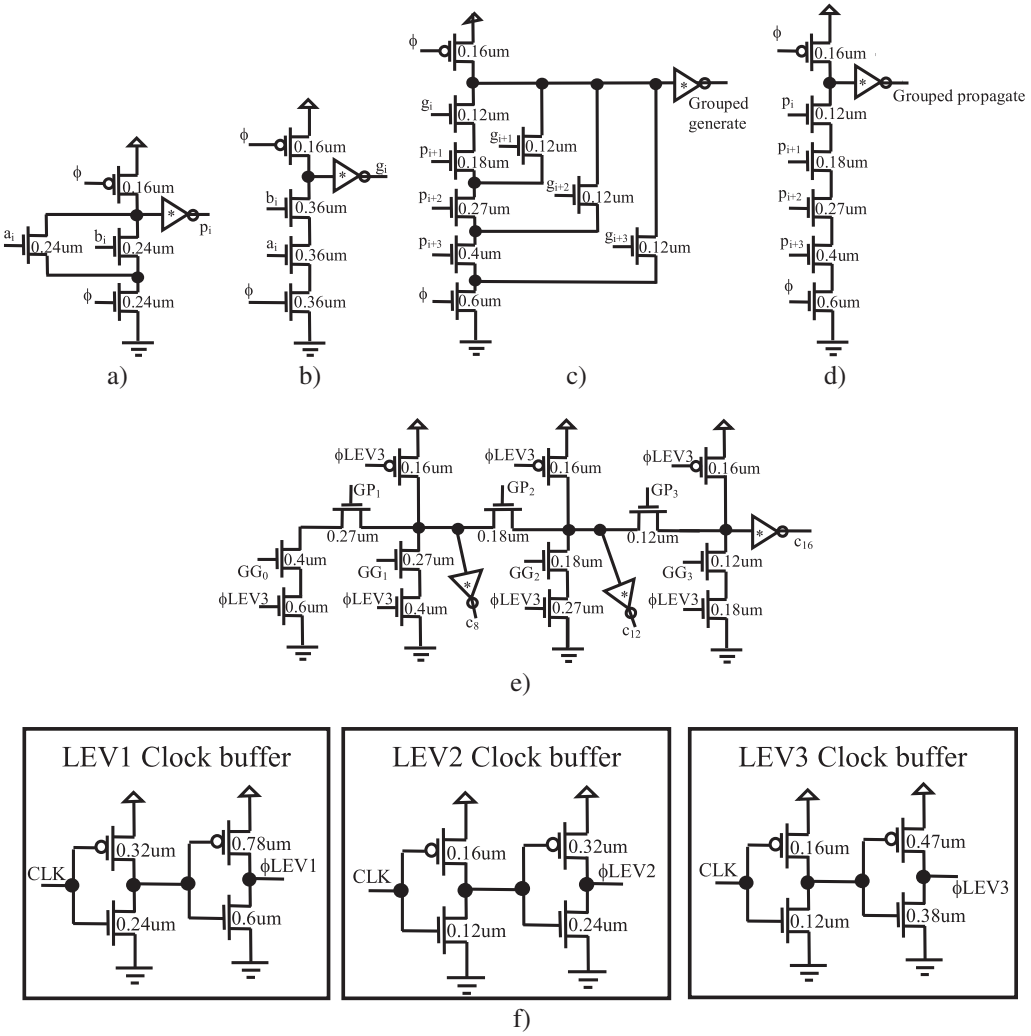


Figure 2. Conventional domino implementations of: a) the two-input OR; b) the two-input AND; c) the seven-input OR-AND; d) the four-input AND; e) the MCC; f) the clock buffers.

computing the propagate signals p_i, p_{i+1}, p_{i+2}, p_{i+3} and the generate signals g_i, g_{i+1}, g_{i+2}, g_{i+3}. Two further logic operations are then performed to compute $GG_{\frac{i}{4}}$ and $GP_{\frac{i}{4}}$ obtained grouping 4 by 4 the propagate and generate signals. Since the single propagate and generate signals are used just locally within the LEV1 module, only one propagate and one generate signal could be produced by each two-bit position, as detailed in equation 5.

$$
\begin{aligned}
\overline{P_{i+1:i}} &= \overline{(a_{i+1}+b_{i+1})\cdot(a_i+b_i)} \\
\overline{P_{i+3:i+2}} &= \overline{(a_{i+3}+b_{i+3})\cdot(a_{i+2}+b_{i+2})} \\
GP_{\frac{i}{4}} &= \overline{\overline{P_{i+3:i+2}}+\overline{P_{i+1:i}}} = P_{i+3:i+2}\cdot P_{i+1:i} \\
\overline{G_{i+1:i}} &= \overline{a_{i+1}\cdot b_{i+1}+(a_{i+1}+b_{i+1})\cdot a_i\cdot b_i} \\
\overline{G_{i+3:i+2}} &= \overline{a_{i+3}\cdot b_{i+3}+(a_{i+3}+b_{i+3})\cdot a_{i+2}\cdot b_{i+2}} \\
GG_{\frac{i}{4}} &= \overline{G_{i+3:i+2}\cdot\left(\overline{P_{i+3:i+2}}+\overline{G_{i+1:i}}\right)} = G_{i+3:i+2}+P_{i+3:i+2}\cdot G_{i+1:i}
\end{aligned}
\tag{5}
$$

DOI: 10.1002/cta

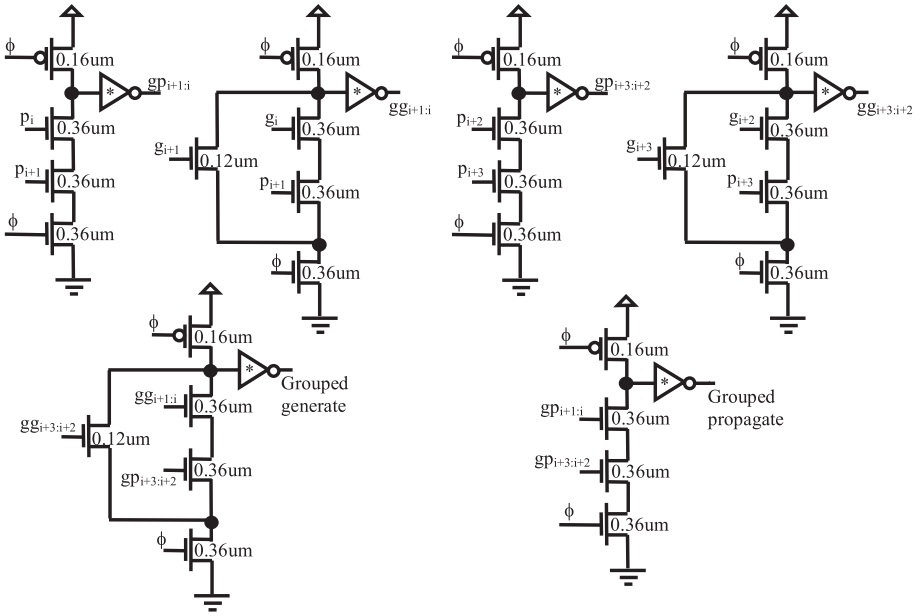


Figure 3. Conventional domino gates with limited stack height.

The novel circuit of the LEV1 module is depicted in Figure 5. There, the domino-compound logic style is used, thus in the following, this implementation is named New_CD. It can be seen that only two propagate intermediate signals, $\bar{P}_{32}$ and $\bar{P}_{10}$, and two generate intermediate signals, $\bar{G}_{32}$ and $\bar{G}_{10}$, are computed, thus reducing the overall number of dynamic nodes within the module to four. The novel circuit of LEV1 is also analyzed when a pure domino logic implementation is carried out. This implementation, named New_Dom, substitutes the static gates used in Figure 5 to compute GG_0, GP_0 with domino correspondent gates. Furthermore, minimum-sized static inverters are introduced on the nodes $\bar{P}_{10}$, $\bar{G}_{10}$, $\bar{P}_{32}$ and $\bar{G}_{32}$.

The New_CD implementation proposed here for LEV1, in comparison with the conventional implementations _Dom, _Dom_LSH and _CD requires up to 43% less transistors. This has a positive impact on the wiring resources utilization. It is worth underlining that in the New_CD implementation, the minimum-size criterion is applied at the static gates, whereas the transistors of the dynamic gates that directly receive the operands A and B are sized ensuring that the input capacitances of a parallel-prefix tree designed as proposed here are mainly unchanged with respect to the conventional implementations. As an example, it can be verified that the front-end module providing the input signals drives one 0.24 μm and one 0.36 μm wide NMOS transistors in all the referred implementations. Obviously, this choice, adopted for comparison purpose, penalizes the novel circuit in terms of computational speed since narrower transistors make its PDNs slower.

All the implementations described above are suitable for radix-4 parallel-prefix trees, but the approach holds also for higher radices that could be preferred to reach higher speed performances.

3.3. Performance analysis and comparison results

All the above described circuits have been implemented using the STMicroelectronics 45 nm 1 V CMOS process technology [20], and they were preliminarily characterized through pre-layout simulations. Computational delays and energy consumptions were analyzed in the presence of both inter-die and intra-die process variations. In order to do this, the input transitions for which the critical paths are activated were set for each characterized module. Following the traditional approach [12, 13], statistical analysis was performed through 1000 runs of Monte Carlo simulations to measure, for each output signal, standard deviation (σ_τ), mean (μ_τ) and 3-sigma delay limit (τ_W) calculated as $_\tau + 3_\tau$. The latter was referenced as a bench mark in the comparison since, as explained

DOI: 10.1002/cta

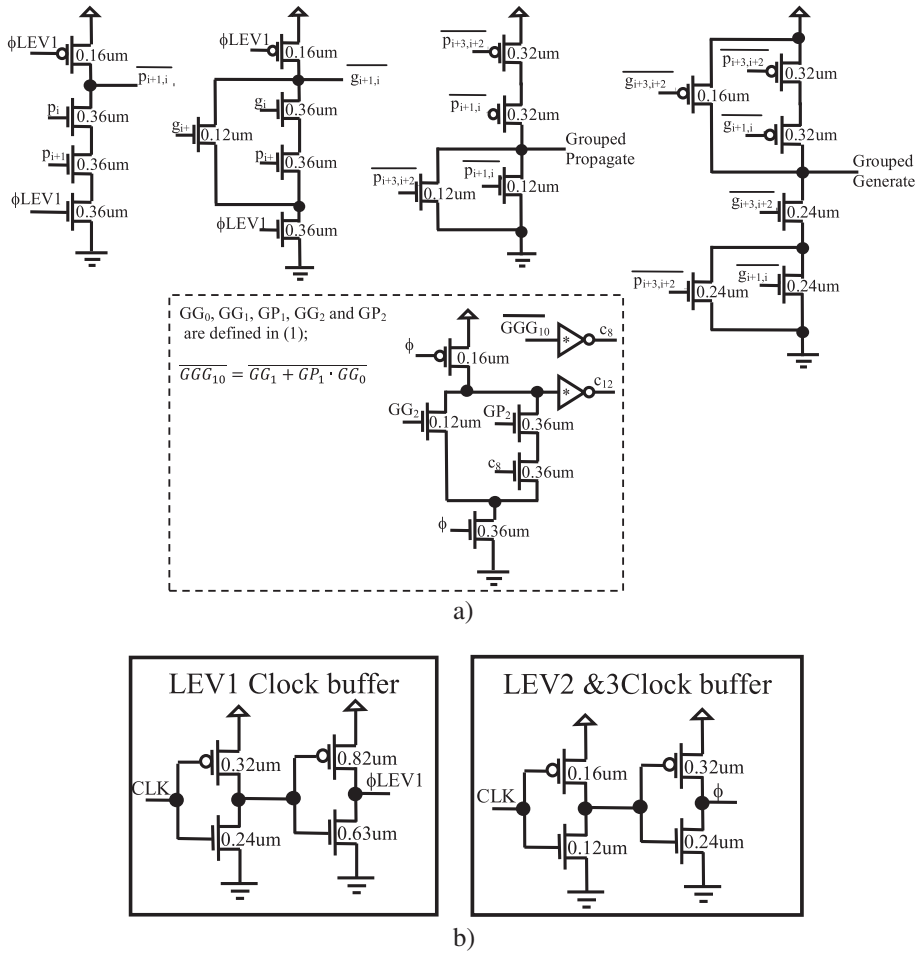


Figure 4. Conventional domino-compound implementations of: a) the basic sub-circuits; b) the clock buffers.

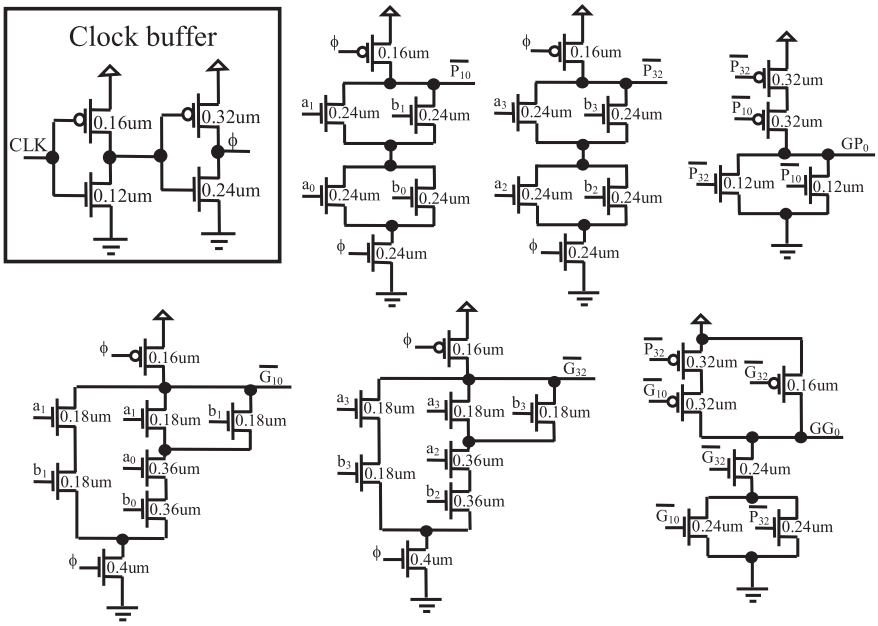


Figure 5. The new implementation proposed for the module LEV1.

DOI: 10.1002/cta

in [21], taking the effects of process variations into account, 99.7% of the samples will show a delay between $_{\tau} - 3_{\tau}$ and $_{\tau} + 3_{\tau}$.

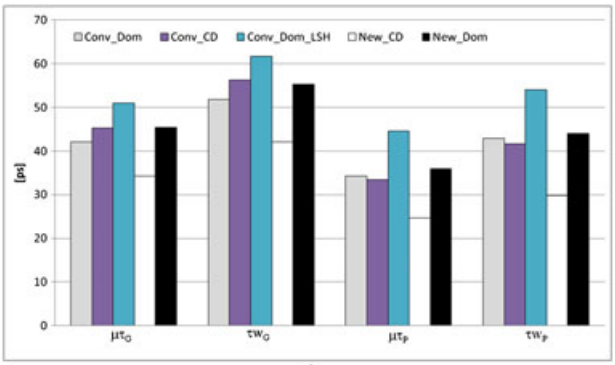
In this preliminary analysis, the nominal 1 V supply voltage and the 27 °C operating temperature were set in all simulations that were performed loading each output signal with a 0.8 fF capacitance.

The statistical analysis of the average energy consumption requires too much time, and it is unfeasible with available workstations. For this reason, we adopted the simplified Corner Analysis. In particular, exhaustive simulations were performed at the Best, Typical and Worst process conditions with a 27 °C operating temperature and 1 V supply voltage to compute mean values $_{E}$, standard deviations $_{E}$ and the 3-sigma energy limits E_W (calculated as $_{E} + 3_{E}$). Measured energy values include the contribution of the clock buffers. Simulations results obtained for delays and energy consumptions are summarized in Figure 6. In Figure 6a, mean delays and 3-sigma delay limits are illustrated for the LEV1 module. It can be observed that the fastest implementation of the module LEV1 is the New_CD proposed here.

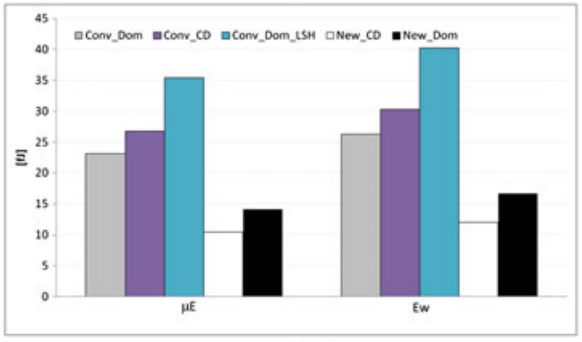
As visible from Figure 6b, the novel strategy used within LEV1 also allows the energy dissipation of the most recurring module inside a sparse parallel-prefix adder tree to be more than halved with respect to all the conventional implementations.

Figure 6 also shows that the _Dom_LSH implementation is significantly slower and dissipates more energy than competitors. This is mainly due to the fact that it contains a higher number of dynamic nodes and uses more inverters to perform the domino action, thus increasing the critical computational path.

To finalize the above analysis towards the realization of a complete parallel-prefix tree, we compare the modules behavior in the Energy-Delay space. These plots are illustrated in Figure 7. From Figure 7a, it can be seen that the New_CD LEV1 allows both minimum delay and minimum energy to be reached. Furthermore, among conventional implementations, the _Dom one shows the better energy-delay trade-off. From Figure 7b, it is clear that the _Dom LEV2 performs better in terms of energy with a very low delay penalty. Finally, as shown in Figure 7c, the _CD implementation is more suitable for speed-critical applications, whereas the _Dom circuit allows minimum energy to



a)



b)

Figure 6. Simulation results: a) computational delays; b) energy consumptions.

DOI: 10.1002/cta

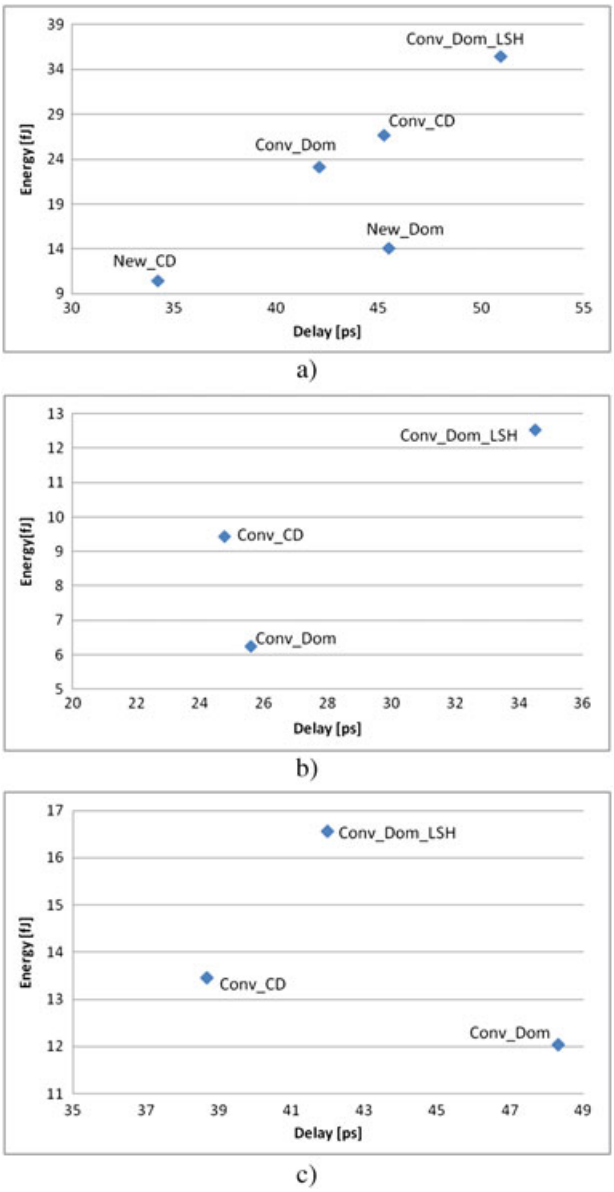


Figure 7. Energy-Delay characteristics of the module: a) LEV1; b) LEV2; c) LEV3.

be dissipated. It should also be noted that the LEV2 and LEV3 modules participate to the overall energy dissipation with lower contributions.

4. POST-LAYOUT CHARACTERIZATION

The novel LEV1 circuit was laid out using the conventional full-custom approach. For comparison purpose, conventional _Dom and _CD modules were also laid out using identical physical strategies. All layouts were organized to ensure that they are not wiring dominated.

Post-layout simulation results obtained at the 27 °C operating temperature are provided in Tables I and II. The former shows the mean delays μ_τ and the standard deviations σ_τ obtained for all the output signals of analyzed modules, whereas the latter summarizes post-layout energy and area behavior of the modules. Measurements were performed loading each output signal with a 0.8 fF capacitance. Worst delays contributing on the critical path of an entire tree are highlighted with bold characters.

DOI: 10.1002/cta

Table I. Post-layout mean delays and standard deviations measurements.

Module	Implementation	$\mu_{\tau G}$ [ps]	$\sigma_{\tau G}$ [ps]	$\mu_{\tau P}$ [ps]	$\sigma_{\tau P}$ [ps]	$\mu_{\tau c8}$ [ps]	$\sigma_{\tau c8}$ [ps]	$\mu_{\tau c12}$ [ps]	$\sigma_{\tau c12}$ [ps]	$\mu_{\tau c16}$ [ps]	$\sigma_{\tau c16}$ [ps]	σ_τ/μ_τ [%]
LEV1	CONV_DOM	**66.91**	**5.14**	49.36	3.89	-	-	-	-	-	-	7.68
	NEW_CD	**51.1**	**3.84**	37.2	2.81							7.51
LEV2	CONV_DOM	**37.89**	**3.26**	31.37	2.92	--	-	-	-	-	-	8.6
	CONV_CD	**38.3**	**2.74**	36.47	2.86							7.15
LEV3	CONV_DOM	-	-	29.69	2.73	30.75	2.77	51.03	4.3	**70.22**	**5.47**	7.79
	CONV_CD			35.92	2.75	32.2	2.56	**57.17**	**4.27**	40.62	2.84	7.47

Table II. Post-layout energy and area measurements.

Module	Implementation	μ_E [fJ]	σ_E [fJ]	*Area* [μm^2]	σ_E/μ_E [%]
LEV1	CONV_DOM	32.37	0.96	28.6	2.97
	NEW_CD	13.78	0.57	14.6	4.13
LEV2	CONV_DOM	7.68	0.24	6.6	3.13
	CONV_CD	12.32	0.37	10.32	3
LEV3	CONV_DOM	15.16	0.45	12.2	2.97
	CONV_CD	17.49	0.63	13.3	3.6

It should be noted that the New_CD implementations of the module LEV1 and Conv_CD of LEV3 are roughly 20% faster than their domino counterparts, whereas the _CD implementation of the module LEV2 mainly shows a ~16% reduction in the delay standard deviation. It is also important to observe that the novel circuit exhibits energy and area reductions higher than 50% with respect to its conventional domino counterpart. In Tables I and II, the delay and energy sensitivities to process variations are also reported. They are defined as the ratios $\frac{\sigma_\tau}{\mu_\tau}$ and $\frac{\sigma_E}{\mu_E}$. Results show that, while domino circuits are quite more delay sensitive than domino-compound counterparts, the Conv_Dom implementations achieve lower energy sensitivity. Nevertheless, the New_CD implementation exhibits energy mean value and standard deviation 57.5% and 40% lower than the Conv_Dom counterpart. Obviously, this leads to a 39% higher sigma/mean ratio.

As an example of application, the 32-bit parallel-prefix tree depicted in Figure 1b was created using the New_CD implementation for the LEV1 module and Conv_CD for LEV3. As already noted in the pre-layout results, the slowest output signal in the CONV_CD implementation of LEV3 is c_{12} instead of c_{16}. This means that the worst-case delay path of a Brent-Kung tree will involve c_{24} instead of c_{28} (dashed line in Figure 1b). The same tree was also implemented using the CONV_DOM implementation for all the involved modules. The layout of the novel parallel-prefix tree is illustrated in Figure 8. For both the laid out trees, all delay and energy measurements were taken loading each output signal with a capacitance of 0.4 fF. The latter was chosen to represent the load due to the sum block receiving the generic carry signal computed by the tree.

The timing diagrams illustrated in Figure 9 are related to the critical operation that takes place when a carry is generated at the least significant bit position (i.e. $a_0 = b_0 = 1$), and then it is propagated through all the subsequent bit positions (i.e. $a_i = 0$ and $b_i = 1$, with $i = 1, \ldots, 31$). It can be seen that the adder tree implemented using only the domino logic provides the carry signals orderly from c_4 to c_{28}. On the contrary, Figure 9b demonstrates that, as expected, the tree in which the novel circuits are employed computes the carry c_{16} in anticipation with respect to c_{12}, and the carry c_{28} well before c_{24}.

The adder trees created were characterized at an operating temperature of 75 °C under process variations through 1000 runs of Monte Carlo simulations. The resulting normal delay distributions are depicted in Figure 10. The latter shows not only that the adder tree designed exploiting the proposed approach is on average ~30% faster than the conventional counterpart, but also that 99.7% of samples exhibits a computational delay lower than 199 ps. At the same yield point, the conventional domino adder achieves a worst case delay of 274 ns. The overall characteristics summarized in Table III also prove that the newly implemented adder dissipates ~41% lower energy

DOI: 10.1002/cta

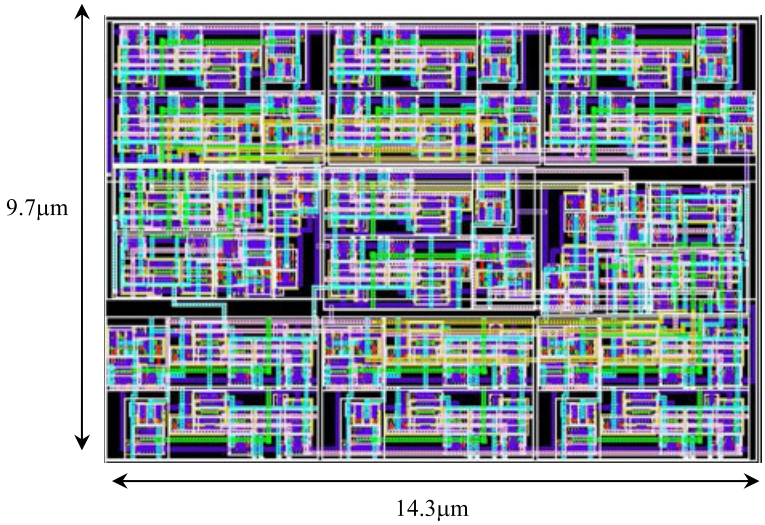


Figure 8. Layout of the 32-bit Brent-Kung tree using novel circuits.

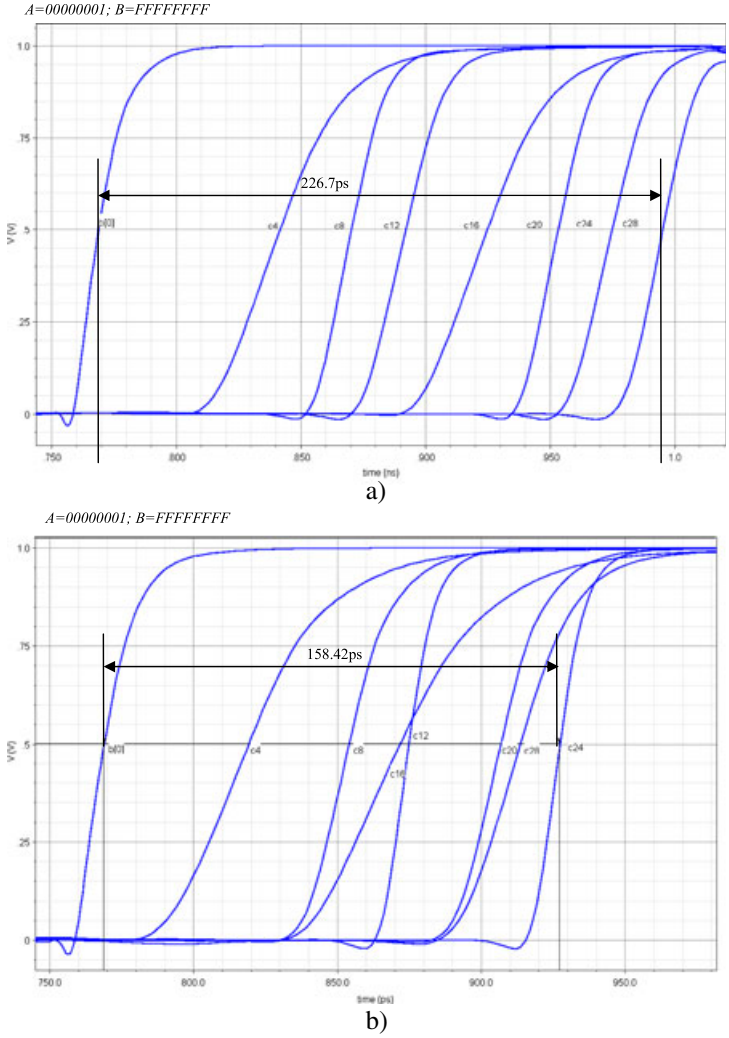


Figure 9. The critical operation performed by: a) the conventional tree; b) the proposed implementation.

DOI: 10.1002/cta

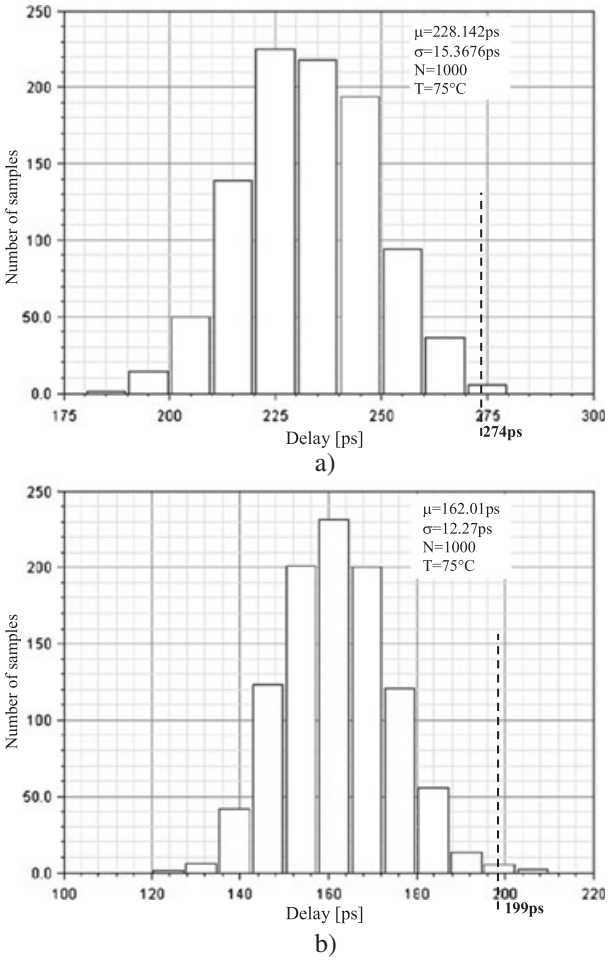


Figure 10. Normal delay distributions of the 32-bit Brent-Kung trees using: a) the conventional tree; b) proposed implementation.

Table III. Post-layout simulation results obtained for the 32-bit Brent Kung tree @75 °C.

Implementation	μ_τ [ps]	σ_τ [ps]	*Energy* TT [fJ]	*Energy* FF [fJ]	*Energy* SS [fJ]	*Area* [μm^2]
CONV_DOM	228.14	15.53	249.4	273.4	237.6	286
NEW_CD + CONV_CD	162.01	12.27	144.3	160.7	137.3	139

and saves ~51% of silicon area. For each corner, a sequence of 1000 randomly generated input has been adopted for energy measurements.

5. CONCLUSIONS

A novel design approach is presented to implement efficient sparse parallel-prefix adder trees using nanometer technologies. The basic idea exploited in the proposed designs consists of: (1) reducing the complexity of the PDNs of each dynamic gate; and (2) minimizing the number of dynamic nodes within the overall structure of the generic parallel-prefix tree.

In comparison with the conventional domino and domino-compound implementations, the circuits proposed here achieve higher speed performances and lower power consumptions.

DOI: 10.1002/cta

As an example of application, a 32-bit Brent-Kung tree was designed applying the novel approach. Statistical analysis performed taking process variations into account demonstrated that the novel adder is ~30% faster and dissipates ~41% lower energy.

REFERENCES

1. Park J, Ngo HC, Silberman JA, Dhong SH. 470 ps 64bit Parallel Binary Adder. *Proc. IEEE Symposium on VLSI Circuits*, Honolulu, Hawaii, 2000; 192–193.
2. Corsonello P, Perri S, Margala M. Efficient Addition Circuits for Modular Design of Processors-in-Memory. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2005; **52**(8):1557–1567.
3. Frustaci F, Lanuzza M, Zicari P, Perri S, Corsonello P. Designing High-Speed Adders in Power-Constrained Environments. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2009; **56**(2):172–176.
4. Das S, Khatri SP. A Novel Hybrid Parallel-Prefix Adder Architecture With Efficient Timing-Area Characteristic. *IEEE Transactions on VLSI Systems*, 2008; **16**(3):326–331.
5. Oklobdzija VG, Zeydel BR, Dao HQ, Mathew S, Krishnamurthy R. Comparison of High-Performance VLSI Adders in the Energy-Delay Space. *IEEE Transactions on VLSI Systems* 2005; **13**(6):754–758.
6. Zlatanovici R, Kao S, Nikolic B. Energy-Delay Optimization of 64-bit Carry-Lookahead Adders with a 240 ps 90 nm CMOS Design Example. *IEEE Journal of Solid-State Circuits* 2009; **44**(2):569–583.
7. Zeydel BR, Baran D, Oklobdzija VG. Energy-Efficient Design Methodologies: High-Performance VLSI Adders. *IEEE Journal of Solid-State Circuits* 2010; **45**(6):1220–1233.
8. Ghosh S, Roy K. Novel Low Overhead Post-Silicon Self-Correction Technique for Parallel Prefix Adders Using Selective Redundancy and Adaptive Clocking. *IEEE Transactions on VLSI Systems* 2011; **19**(8):1504–1507.
9. Chen TC. Where CMOS is Going: Trendy Hype vs. Real Technology. in *Proc. IEEE International Solid-State Circuits Conference*, San Francisco, USA, Vol.3, 2006; 5–9.
10. Boning D, Nassif S. Models of Process Variations in Device and Interconnect. In *Design of High-Performance Microprocessor Circuits*, Chandrakasan A (Ed), Wiley-IEEE Press: Piscataway, NJ, USA, 2000; 98–115.
11. Mukhopadhyay S, Kim K, Jenkins KA, Chuang CT, Roy K. An On-Chip Test Structure and Digital Measurement Method for Statistical Characterization of Local Random Variability in a Process. *IEEE Journal of Solid-State Circuits* 2008; **43**(9):1951–1963.
12. Da Silva DN, Reis AI, Ribas RP. CMOS logic gate performance variability related to transistor network arrangements. *Microelectronics Reliability* 2009; **49**(9-11):977–981.
13. Alioto M, Palumbo G, Pennisi M. Understanding the Effect of Process Variations on the Delay of Static and Domino Logic. *IEEE Transactions on VLSI Systems* 2010; **18**(5):697–710.
14. Kogge PM, Stone HS. A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations. *IEEE Transactions on Computers* 1973; **C-22**(8):786–793.
15. Ladner RE, Fischer MJ. Parallel Prefix Computation. *Journal of the ACM* 1980; **27**(4):831–838.
16. Ling H. High-speed binary adder. *IBM Journal of Research and Development* 1981; **25**(2-3):156–166.
17. Brent RP, Kung HT. A regular layout for parallel adders. *IEEE Transactions on Computers*, 1982; **C-31**(3):260–264.
18. Han T, Carlson DA. Fast area-efficient VLSI adders. *Proc. IEEE Symposium on Computer Arithmetic*, Como, Italy, 1987; 49–56.
19. Rabaey J, Chandrakasan A, Nikolic B. *Digital Integrated Circuits: A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 2003.
20. "CMOS045 Design Kit- Version 4.0" Central CAD & Design Solution Front End Technology & Manufacturing ST Microelectronics, February 2008.
21. Yelamarthi K, Chen CIH. Timing Optimization and Noise Tolerance for Dynamic CMOS Susceptible to Process Variations. *IEEE Transactions on Semiconductor Manufacturing* 2012; **25**(2):255–265.

DOI: 10.1002/cta