Designing High-Speed Adders in Power-Constrained Environments

Fabio Frustaci, Marco Lanuzza, Paolo Zicari, Stefania Perri, *Member, IEEE*, and Pasquale Corsonello, *Member, IEEE*

***Abstract*—Data-driven dynamic logic (D3L) is very efficient when low-power constraints are mandatory. Unfortunately, this advantage is typically obtained at the expense of speed performances. This paper presents a novel technique to realize D3L parallel prefix tree adders without significantly compromising speed performance. When applied to a 64-bit Kogge–Stone adder realized with 90-nm complementary metal–oxide–semiconductor (CMOS) technology, the proposed technique leads to an energy-delay product that is 29% and 21% lower than its standard domino logic and conventional D3L counterparts, respectively. It also shows a worst case delay that is 10% lower than that of the D3L approach and only 5% higher than that of the conventional domino logic.**

***Index Terms*—Clock-precharged dynamic logic, data-driven dynamic logic (D3L), data-precharged dynamic logic, parallel prefix adder.**

I. Introduction

ADDITION is a fundamental operation in any digital system and can significantly influence the overall achievable performances [1]. For this reason, novel high-speed adders are highly desirable.

The speed performances of addition circuits can be improved by optimizing both the top-level structure and the circuit implementation [2]–[4]. However, it is worth noting that the adder topology, together with the used logic and transistor-sizing criterion, also significantly affects energy dissipation [5]–[7]. For example, a very high speed is reached when parallel prefix adders are implemented with dynamic domino logic. In this case, the advantages offered by the logarithmic depth tree structure are emphasized through fast dynamic logic, which, as a drawback, requires a clock distribution system to correctly run. As demonstrated in [8], the power dissipation, owing to the clock distribution network in a dynamic system, can range from 20% up to 45% of the overall consumed power. This provides crucial information for the design of efficient digital circuits, in which achieving low-power dissipation is also an important issue. To limit the power consumption of the clock distribution system, the data-driven dynamic logic (D3L) [9] and the clock- and data-precharged dynamic logic (CDPDL) [10] have recently been proposed. These techniques completely or partially remove the clock distribution system required within conventional dynamic circuits, thus leading to significantly lower energy consumption. Unfortunately, this advantage is obtained at the expense of a nonnegligible penalty in speed performances.

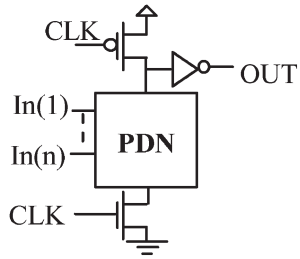

Fig. 1. Generic n-type domino gate.

This paper presents a new technique for exploiting the energy-saving advantages offered by D3L without paying significant performance penalty with respect to the conventional domino logic. As a sample application, a new data-precharged dynamic structure is presented for a 64-bit Kogge–Stone parallel prefix adder. When implemented with STMicroelectronics 90-nm 1-V complementary metal–oxide–semiconductor (CMOS) technology, the novel adder exhibits an energy-delay product that is 29% and 21% lower than those of the standard clock-precharged domino logic and the conventional D3L implementations, respectively.

This paper is organized as follows: In Section II, a brief background is given. In Section III, a comparison between the traditional clock-precharged domino logic and the D3L described in [9] is performed. Finally, Section IV describes the new data-precharged domino adder and furnishes some interesting comparison results.

II. Background

Conventional dynamic CMOS circuits operate using a sequence of precharge and evaluation phases based on a clock input. During the precharge phase, the output signal is forced to a predefined value, independently of the input data. On the contrary, during the evaluation phase, the output signal depends on the received inputs.

Among the dynamic logics known in the literature, domino logic is the most widely used in high-performance microprocessors due to its speed and area characteristics [11].

An n-type (p-type) domino circuit executes the precharge phase when the clock signal is low (high) and the evaluation phase when the clock is high (low). Fig. 1 shows the generic n-type domino gate.

Manuscript received June 23, 2008; revised September 30, 2008. First published February 10, 2009; current version published February 25, 2009. This paper was recommended by Associate Editor A. Brambilla.

The authors are with the Department of Electronics, Computer Science and Systems (DEIS), University of Calabria, 87036 Arcavacata di Rende, Italy (e-mail: p.corsonello@unical.it).

Digital Object Identifier 10.1109/TCSII.2008.2010187

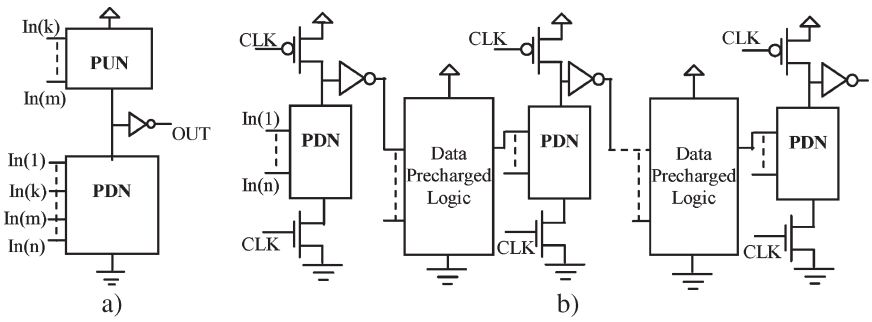

Fig. 2. Generic n-type gates realized with (a) D3L and (b) CDPDL.

The possibility of directly controlling the sequence of the precharge and evaluation phases with the input data, instead of the clock signal, has been examined in [9] and [10]. The latter work proposed dynamic circuits in which the clock distribution system is eliminated or significantly simplified. The generic n-type circuit implemented using the D3L presented in [9] and the CDPDL introduced in [10] is shown in Fig. 2. The basic concept exploited in [9] consists of replacing the clocked pull-up p-channel metal–oxide–semiconductor (PMOS) of the conventional domino logic circuit with a pull-up network (PUN) driven by some specific data inputs. As shown in Fig. 2(b), this concept is also applied in [10] by alternating data-precharged and conventional clock-precharged stages.

Here, referring to the D3L gates, the signals inputted to the pull-down network (PDN) are called the evaluation inputs, whereas the signals inputted to the PUN are called the precharge inputs. From Fig. 2(a), it can be observed that the precharge inputs $In(k),\ldots,In(m)$ are a subset of the evaluation inputs $In(1),\ldots,In(n)$ and that they need to satisfy the following two conditions: 1) During the precharge phase, the PDN is off, the PUN is certainly turned on, and the output node is charged to Vdd. 2) During the evaluation phase, the output node is eventually discharged to 0 by the PDN without any contention with the PUN.

The most evident benefits offered by the D3L circuits over their conventional domino counterparts are the elimination of the clock distribution system, which allows a significant reduction in the power consumption, and the elimination of the clocked n-channel metal–oxide–semiconductor (NMOS) transistor, which reduces the evaluation path. Unfortunately, two main disadvantages arise, i.e., the input lines of the D3L circuits have higher capacitances and the precharge phase is no longer simultaneous for all the gates since, during this phase, a propagation path exists through cascaded stages. To prevent the precharge phase from becoming slower than the evaluation phase, PMOS transistors wider than the minimum are required in the PUN, thus causing power dissipation overhead. Obviously, using the CDPDL style, the aforementioned problems are limited, but the advantages of having data-precharged stages cannot completely be exploited since a clock distribution system is still necessary.

Each of the design styles previously described has its advantages and disadvantages. Therefore, establishing which design is the most efficient depends on the specific application. In the following, the design of a fast adder and its optimization are presented as a case study conducted to compare domino and D3L logics. The adder design is a representative application since addition is often the main speed-limiting operation in digital systems.

III. Design of a Fast Adder: A Case Study

Among the existing addition circuits, such as ripple–carry, carry–select, and carry–look-ahead, logarithmic parallel prefix adders are the most widely used for reaching a high speed. With $A=a_{n-1},\ldots,a_0$ and $B=b_{n-1},\ldots,b_0$ being two n-bit inputs, the parallel prefix adders calculate the sum $S=s_{n-1},\ldots,s_0$ through the following three-stage operation: 1) The preprocessing stage computes the carry bit generate and carry bit propagate signals $g_i=a_i\cdot b_i$ and $p_i=a_i\oplus b_i$, respectively, for each bit position $i=0,\ldots,n-1$. 2) The carry propagation stage produces carry signals c_i. 3) The final stage calculates sum bits $s_i=p_i\oplus c_i$. As well known, the second stage is the most time critical. In fact, it computes the grouped generate and grouped propagate signals defined in

$$\begin{aligned} G_{i,j} &= g_i + g_{i-1}\cdot p_i + \cdots + g_j\cdot p_i\cdot p_{i-1}\cdot\cdots\cdot p_{j+1} \\ P_{i,j} &= p_i\cdot\cdots\cdot p_j \end{aligned} \tag{1}$$

with $j=0,\ldots,n-1$ and $j<i$, and then calculates generic carry $c_i=G_{i-1:0}$ using the associative dot operator "$\bullet$" defined in

$$\begin{aligned} (G_{i,k},P_{i,k}) &= (G_{i,j},P_{i,j})\bullet(G_{j-1,k},P_{j-1,k}) \\ &= (G_{i,j}+P_{i,j}\cdot G_{j-1,k},P_{i,j}\cdot P_{j-1,k}) \end{aligned} \tag{2}$$

where $k<j<i$.

The different ways of exploiting the associative dot operator result in different tree adder architectures, such as the Kogge–Stone, Brent–Kung, Ling, Ladner–Fischer, Han–Carlson, and Knowles adders. Among them, the Kogge–Stone adder was chosen, because it has a very regular structure with the minimum logic depth and the minimum fan-out [12]. According to the Kogge–Stone algorithm, (2) is specialized as

$$\begin{aligned} &G^0_{i,j}=g_i \qquad P^0_{i,j}=P_i \\ &G^m_{i,j}=G^{m-1}_{i,j} \quad P^m_{i,j}=P^{m-1}_{i,j}, && \text{when } i=2^{m-1} \\ &\left.\begin{aligned} G^m_{i,j} &= G^{m-1}_{i,k}+P^m_{i,k}\cdot G^{m-1}_{k-1j} \\ P^m_{i,j} &= P^{m-1}_{i,k}\cdot P^{m-1}_{k-1j} \end{aligned}\right\}, && \text{when } i\geq 2^{m-1} \end{aligned} \tag{3}$$

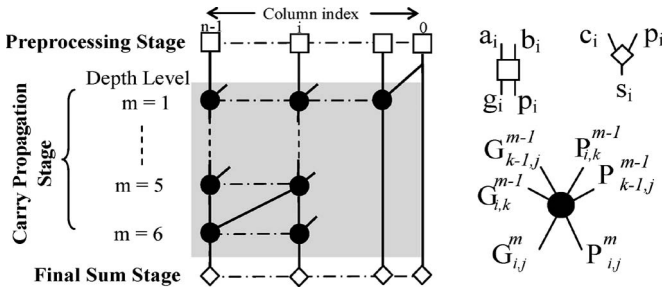

Fig. 3. 64-bit Kogge–Stone adder.

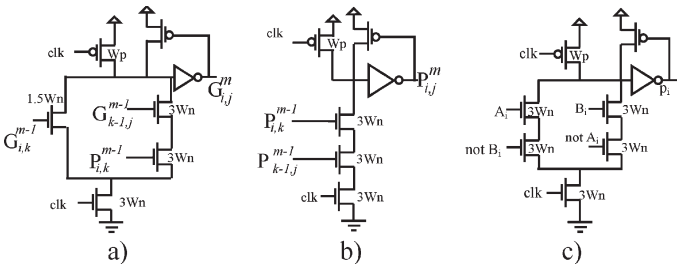

Fig. 4. Domino gates. (a) Grouped generate. (b) Grouped propagate. (c) Propagate.

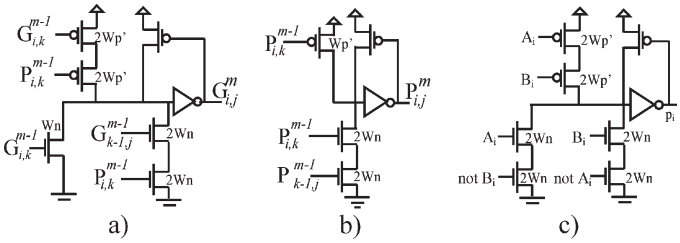

Fig. 5. D3L gates. (a) Grouped generate. (b) Grouped propagate. (c) Propagate.

where m is the depth level of the adder tree, i is the column index, and

$$m = 1, \ldots, \lceil \log_2 n \rceil$$
$$k = i - 2^{m-1} + 1$$
$$j = \begin{cases} i - 2^m + 1, & \text{when } i > 2^m - 1 \\ 0, & \text{otherwise.} \end{cases}$$

Equation (3) leads to the 64-bit Kogge–Stone adder shown in Fig. 3.

This circuit was realized using the ST 90-nm 1-V CMOS technology with the domino and D3L gates shown in Figs. 4 and 5. According to (3), the D3L gate shown in Fig. 5(a) can exploit $G_{i,k}^{m-1}$ and $P_{i,k}^{m-1}$ as the precharge inputs, whereas the AND gate in Fig. 5(b) uses just $P_{i,k}^{m-1}$ as its precharge input. In the realized implementations, generate signals g_i are computed using the AND gates organized as in Figs. 4(b) and 5(b). Finally, the gates used for computing the final sum bits s_i are not shown in Figs. 4 and 5 since they are simple two-input static XOR gates.

The NMOS transistors in the PDNs of both implementations were equally sized to make the generic evaluation path equivalent to an NMOS transistor that is $W_n = 0.3$ μm wide. To properly size the PUNs of the D3L gates, it was taken into account that the use of wider PMOS transistors not only reduces the precharge time but also increases the power consumption. As a result of the parametric simulations, parameter W_p' was set to 1.4 μm, which is the minimum value assuring a precharge delay that is not greater than the evaluation delay along the critical path of the entire adder.

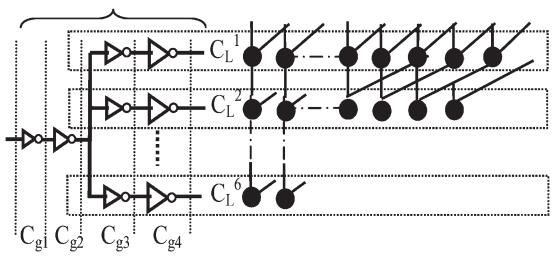

Fig. 6. Clock buffer tree used in the domino implementation.

A width W_p that is equal to 0.15 μm was used for the clocked PMOS transistors of the domino gates. This reduced channel width can be used since, in the domino gates, the precharge operation occurs in parallel. All the gates shown in Figs. 4 and 5 use a minimum-sized (0.12-μm) PMOS keeper transistor and a skewed static inverter with a skew ratio of 4 (PMOS-width/NMOS-width ratio), which assures high-speed performances [13].

To distribute the clock signal to the dynamic gates used in the domino implementation, the two-level clock buffer tree shown in Fig. 6 was purposely designed. It can be seen that, due to the regularity of the tree adder structure, the different paths existing for the clock distribution can well be balanced. The logical effort method [14] was used for sizing the inverter chains of the clock buffer. In particular, the following expression was applied:

$$\frac{C_{g2}}{C_{g1}} = \frac{6C_{g3}}{C_{g2}} = \frac{C_{g4}}{C_{g3}} = \frac{C_L^m}{C_{g4}} \tag{4}$$

where C_L^m is the load capacitance due to the clocked transistors within the mth depth level in the adder tree, with $m = 1, \ldots, 6$, and C_{gl} is the gate capacitance of the inverter at the lth stage of the clock buffer, with $l = 1, \ldots, 4$. Finally, to take into account that the considered domino and D3L implementations have different input capacitances, the input signals were accordingly buffered before driving the preprocessing stages. Energy and delay measurements were performed for the realized adders, loading each output signal with a 2.5-fF capacitance (i.e., the input capacitance of a D-type flip-flop) and taking into account the parasitic resistances and capacitances extracted from the layouts.

The comparison results obtained for different process corners and temperatures are summarized in Table I. They show that the D3L adder achieves energy dissipation that is $\sim$20% lower than that of the domino circuit, in which the parallel prefix adder and the clock buffer dissipate $\sim$68% and $\sim$32%, respectively, of the total energy. However, the D3L implementation is also $\sim$17% slower than its domino counterpart, thus leading to a substantially unchanged energy-delay product (E^*D).

TABLE I
DOMINO LOGIC VERSUS D3L (VALUES ARE NORMALIZED WITH RESPECT TO THE DOMINO LOGIC)

		Precharge delay	Evaluation delay	Total Energy(1)	Energy of the clock buffer [% of Total energy]	E*d
TT 25°C	Domino	1	1	1	31%	1
	D3L	1.2	1.16	0.78	0	0.91
SS 125°C	Domino	1	1	1	27%	1
	D3L	1.17	1.17	0.82	0	0.96
FF -55°C	Domino	1	1	1	27%	1
	D3L	1.18	1.18	0.81	0	0.96

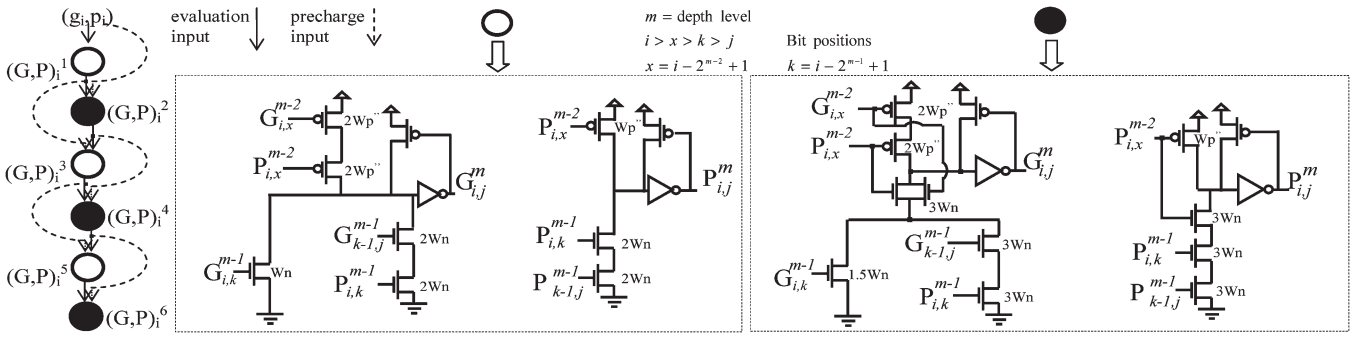

Fig. 7. Generic column of the proposed tree structure.

IV. THE NOVEL APPROACH

The new technique proposed here allows exploiting the advantages offered by the data-precharged design style without compromising speed performances, thus actually improving the energy-delay product. The presented approach exploits the possibility of using, as the precharge inputs, signals that are not included in the evaluation inputs received by the PDN. To explain how this is possible, (3) is recursively expanded, as given in the following:

$$\begin{aligned}
G_{i,j}^{m} &= G_{i,k}^{m-1} + P_{i,k}^{m-1} \cdot G_{k-1,j}^{m-1} \\
&= \left(G_{i,x}^{m-2} + P_{i,x}^{m-2} \cdot G_{x-1,k}^{m-2}\right) + P_{i,x}^{m-2} \\
&\quad \cdot P_{x-1,k}^{m-2} \cdot \left(G_{k-1,y}^{m-2} + P_{k-1,y}^{m-2} \cdot G_{y-1,j}^{m-2}\right) \\
&\;\;\vdots \\
&= g_i + p_i \cdot g_{i-1} + p_i \cdot p_{i-1} \cdot g_{i-2} + \cdots \\
&\quad + p_i \cdot p_{i-1} \cdot \cdots \cdot p_{j+1} \cdot g_j \\
P_{i,j}^{m} &= P_{i,k}^{m-1} \cdot P_{k-1,j}^{m-1} \\
&= P_{i,x}^{m-2} \cdot P_{x-1,k}^{m-2} \cdot P_{k-1,y}^{m-2} \cdot P_{y-1,j}^{m-2} \\
&\;\;\vdots \\
&= p_i \cdot p_{i-1} \cdot \cdots \cdot p_{j+1} \cdot p_j
\end{aligned} \tag{5}$$

with $j < y < k < x < i$. This makes it evident that the generic dot operator at the mth depth level of the ith column of the adder tree receives the signals generated at the $(m-1)$th depth level as the evaluation inputs and can receive the signals generated at any preceding depth level as the precharge inputs. Conventional D3L gates use the signals coming from the $(m-1)$th level as both evaluation and precharge inputs. From (5), it can be seen that the precharge phase would be performed in parallel if signals g_i and p_i were used as the precharge inputs of the dot operators inside the ith column of the adder tree. Unfortunately, in this way, the load driven by the preprocessing stage would drastically increase, and the evaluation inputs of the dot operators would come from a tree level that is deeper than those generating the precharge inputs. As a consequence, the PUNs could be switched on before the PDNs are switched off, thus generating nonnegligible short-circuit currents.

An alternative approach to the solution previously described consists of using signals $G_{i,x}^{m-2}$ and $P_{i,x}^{m-2}$ (with $x = i - 2^{m-2} + 1$) as the precharge inputs of the generic dot operator at the mth depth level inside the ith column, thus halving the propagation path occurring during the precharge phase, with respect to the conventional D3L implementation. The resulting adder topology remains unchanged, except for the physical distribution of the precharge signals. In this case, the dot operators inside the ith column are classified as "*odd*" and "*even*" nodes, depending on their depth level m. In Fig. 7, white and black nodes are used to distinguish odd and even dot operators. It is worth noting that the precharge inputs of the generic odd (even) dot operator are the outputs of the preceding odd (even) operator. In other words, as shown in Fig. 7, the input data of the dot operator at level m are the output signals produced by the dot operators of level $m-1$, whereas its precharge inputs are the output signals produced at level $m-2$. For the first node in the ith column, the precharge and evaluation inputs are the same, i.e., g_i and p_i. This approach does not introduce contentions between the PUNs and the PDNs. In fact, for the generic odd dot operator, it can be seen that, during the precharge phase, the operands A and B of the whole adder are set to 0. Thus, all the signals g_i, p_i, $G_{i,x}^{m-2}$, and $P_{i,x}^{m-2}$ are low, and the dynamic nodes of the gates in Fig. 7 are precharged. During the subsequent evaluation phase, the dynamic nodes can be discharged through the PDNs turned on. In such a case, the PUNs are certainly off, because, as shown in (5), either $G_{i,x}^{m-2}$ or $P_{i,x}^{m-2}$ is set to 1. Contentions between the PUNs and the PDNs are prevented since both precharge and evaluation inputs come from a propagation path that is $m-2$ depth levels long, thus arriving at the gate input after the same delay. On the

TABLE II
COMPARISON RESULTS

		Precharge delay [ps]	Evaluation delay [ps]	Total Energy(1) [pJ]	Energy of the clock buffer [pJ]	E*d [pJ*ps]	Total leakage current [uA]	Leak. current of the clock buffer [uA]	Area [um^2]
	Domino	360	370	6.4	4.4	2368	5	0.5	11000
TT	**D3L**	425	430	5	0	2150	5.2	0	14000
25°C	**New**	383	390	4.3	0	1677	4.8	0	12000
	Domino	510	520	6.41	1.73	3333	26	2.65	11000
SS	**D3L**	605	611	5.26	0	3213	26.4	0	14000
125°C	**New**	553	560	4.71	0	2637	24.3	0	12000
	Domino	250	255	6.3	1.73	1603	1.54	0.16	11000
FF -	**D3L**	293	300	5.12	0	1536	1.59	0	14000
55°C	**New**	270	270	4.6	0	1242	1.48	0	12000

contrary, the generic even node, with $m = 2, 4, 6$, receives the precharge inputs in anticipation with respect to its evaluation inputs. This happens, because the precharge inputs are formed through $m/2 - 1$ depth levels, whereas the evaluation inputs are formed through $m/2$ depth levels. To avoid short-circuit paths, the PDNs of the even nodes contain extra NMOSs driven by the precharge inputs, as shown in Fig. 7.

The main benefit of the proposed solution is halving the precharge propagation path (only three, instead of six, depth levels) with a negligible increase in the overall transistor count, with respect to the D3L implementation.

The proposed adder was implemented with the ST 90-nm 1-V CMOS technology and has been compared to the domino and D3L adders described in the previous section. In addition, for the novel adder, parameter W_n was set to 0.3 μm, thus assuring an equal resistive path for the PDN of all the dot operators. The PMOS transistors inside the PUNs were sized, making the adder precharge propagation delay not greater than the evaluation propagation delay.

Exploiting the reduction of the propagation path occurring during the precharge phase, the PMOS transistors in the PUNs can have channel widths smaller than their D3L counterparts. In particular, parameter $W_p'' = 0.5$ μm was used. It is expected that, in this way, the power consumption is also reduced with respect to the conventional D3L implementation.

The postlayout measurements are summarized in Table II, which shows that, depending on the process corners, the proposed adder saves $27\% \div 33\%$ of the energy consumption with respect to the domino adder, without compromising speed performances. In fact, the novel adder is just $5\% \div 8\%$ slower. When compared to its D3L counterpart, the new architecture is $\sim$10% faster and $\sim$14% less energy consuming. Table II also shows that the approach proposed here offers an energy-delay product that is $21\% \div 29\%$ lower than that of conventional implementations. In addition, the leakage current of the proposed adder is lower than its counterparts due to the smaller transistor sizes. The "even" nodes of the new adder have one extra NMOS in the evaluation path, compared to the generic node of the D3L scheme, thus enhancing the stack effect.

It is worth noting that, the proposed data-precharged strategy can also be applied to implement other kinds of parallel prefix adders. Moreover, it can be exploited, together with other previously published low-power strategies [5]–[7] known in the literature.

V. CONCLUSION

In this paper, a new technique to efficiently exploit data-precharged dynamic circuits has been presented. When applied to the design of a 64-bit Kogge–Stone adder, the proposed approach halves the precharge propagation path with respect to the traditional D3L design style and allows a smaller sizing of the precharging PMOS transistors. As a consequence, the new strategy leads to an energy-delay product that is 29% and 21% lower than those of the traditional domino and D3L logic styles.

REFERENCES

[1] R. Shalem, E. John, and L. K. John, "A novel low power energy recovery full adder cell," in *Proc. 9th Great Lakes Symp. VLSI*, Ypsilanti, MI, Mar. 4–6, 1999, pp. 380–383.

[2] S. Mathew, M. Anders, R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 130-nm address generation unit with 32-bit sparse-tree adder core," *IEEE J. Solid-State Circuits*, vol. 38, no. 5, pp. 689–695, May 2003.

[3] J. Kim, K. Lee, and H.-J. Yoo, "A 372 ps 64-bit adder using fast pull-up logic in 0.18-μm CMOS," in *Proc. IEEE ISCAS*, Island of KOS, Greece, May 21–24, 2006, pp. 13–16.

[4] S. Kao, R. Zlatanovici, and B. Nikolic, "A 240 ps 64 b carry-lookahead adder in 90 nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 6–9, 2006, pp. 1735–1744.

[5] R. Zlatanovici and B. Nikolic, "Power-performance optimization for custom digital circuits," in *Proc. PATMOS*, Leuven, Belgium, Sep. 2005, pp. 404–414.

[6] D. Patil, O. Azizi, M. Horowitz, R. Ho, and R. Ananthraman, "Robust energy-efficient adder topologies," in *Proc. 18th IEEE Symp. Comput. Arith. ARITH*, Montpellier, France, Jun. 25–27, 2007, pp. 16–28.

[7] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 754–758, Jun. 2005.

[8] H. Kawaguchi and T. Sakurai, "A reduced clock-swing flip-flop (RCSFF) for 63% power reduction," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, pp. 807–811, May 1998.

[9] R. Rafati, S. M. Fakhraie, and K. C. Smith, "Low-power data-driven dynamic logic (D^3L)," in *Proc. IEEE ISCAS*, Geneva, Switzerland, 2000, pp. 752–755.

[10] J. R. Yuan, C. Svensson, and P. Larsson, "New domino logic precharged by clock and data," *Electron. Lett.*, vol. 29, no. 25, pp. 2188–2189, Dec. 9, 1993.

[11] M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[12] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Comput.*, vol. C-22, no. 8, pp. 786–793, Aug. 1973.

[13] A. Solomatnikov, D. Somasekhar, K. Roy, and C.-K. Koh, "Skewed CMOS: Noise-immune high-performance low-power static circuit family," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 4, pp. 469–476, Aug. 2002.

[14] I. Sutherland, R. Sproull, and D. Harris, *Logical Effort*. San Francisco, CA: Morgan Kaufmann, 1999.