Adv. Radio Sci., 9, 289–295, 2011
www.adv-radio-sci.net/9/289/2011/
doi:10.5194/ars-9-289-2011

Advances in Radio Science

A modified prefix operator well suited for area-efficient brick-based adder implementations

I. Rust and T. G. Noll

Chair of Electrical Engineering and Computer Systems, RWTH Aachen University, 52062 Aachen, Germany

Abstract. The implementation of integrated circuits becomes more and more difficult in the Ultra-Deep-Submicron regime due to sub-wavelength lithography issues. An approach called Brick-Based Design was recently proposed to eliminate the disadvantages of staying with the classical approach to layout design. Prefix adders are a core component in a wide variety of applications due to their high speed and regular topology. In this paper, a modified prefix operator for prefix adders is proposed which is well suited for brick-style layout implementation and, in addition, offers an increase in efficiency. The proposed operator makes it possible to use a mirror gate for the generation of both generate and propagate signals, which exhibits a forbidden input signal combination. This "forbidden state" causes an increase in power dissipation due to transient short circuit currents. The effect of the forbidden state was quantified as part of a comparison against the classical prefix operator, based on 64-bit Sklansky adders implemented in a 40-nm CMOS technology. The effects of the forbidden state were found to be well acceptable. The implementation of the adder based on the proposed prefix operator reduces the area by 29% while increasing the power by 13% compared to one based on the classical operator.

1 Introduction

The continuous reduction of feature sizes in nanometer scale technologies makes high-yield manufacturing at these technology nodes a growing challenge.

This is because classical minimum-space design rules can not guarantee the effective usage of Resolution Enhancement Techniques (RETs) that are necessary to tackle the sub-wavelength lithography problems (Jhaveri et al., 2007). As a result, the yield is compromised and the circuit performance is reduced due to variability issues (Tong et al., 2006).

A straight-forward way to solve this problem is to use Design-For-Manufacturing (DFM or "Recommended") rules, which allow to keep the classical approach to layout design, but usually cause a high area penalty.

As an alternative approach, it is possible to restrict allowed patterns for layout generation to a small subset highly optimized for lithography ("Restricted Patterning") and, on the other hand, keep or even relax the corresponding layout design rules ("Restricted Design Rules" with SRAM design rules as a well-known example (Jhaveri et al., 2007; Liebmann et al., 2009)). Using this design style, area savings of 15–25% compared to standard cells using DFM rules are expected (Goering, 2007).

Figure 1 shows a layout comparison between minimum-spacing (MS) rules and DFM rules for an operator cell used in carry prefix adders, the area saving is as high as 35%.

Note: Layout pictures do not comply to real design rules to protect proprietary information.

In 2005, Kheterpal et al. (Kheterpal et al., 2005) proposed the Brick-Based Design approach, where a small set of logic primitives is mapped on cells called bricks, which are implemented using the Restricted Patterning approach. To ensure high yield manufacturability, the brick layout must comply to the following rules (Jhaveri et al., 2007, 2009; Tong et al., 2006; Liebmann et al., 2009; Kheterpal et al., 2005; Taylor and Pileggi, 2007):

- unidirectional use of poly-, diffusion and metal layers,
- no jogs in poly layer,
- fixed pitch on poly/metal,
- high periodicity of layout structures.

Fast prefix adders based on the operator proposed by Brent-&Kung (Brent and Kung, 1982), layout shown in Fig. 1, are frequently used in a wide variety of applications due to their

Correspondence to: I. Rust
(rust@eecs.rwth-aachen.de)

Published by Copernicus Publications on behalf of the URSI Landesausschuss in der Bundesrepublik Deutschland e.V.

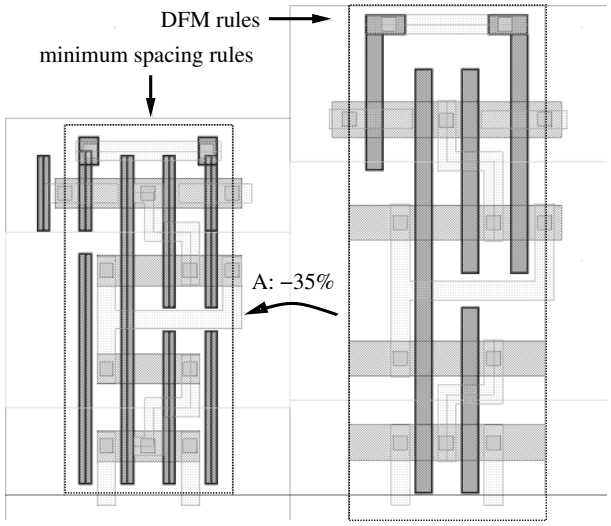

Fig. 1. B&K prefix operator: MS vs. DFM rules.

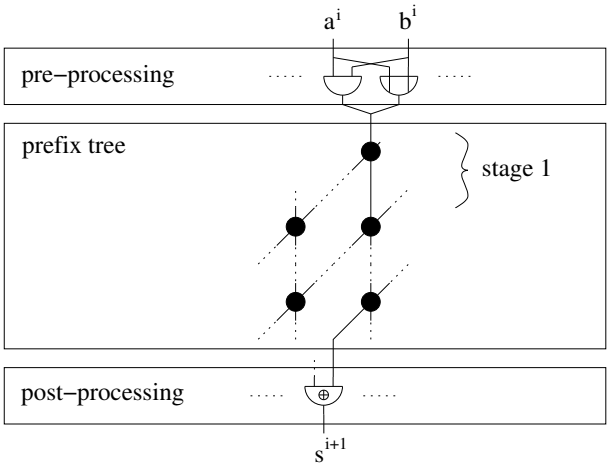

Fig. 2. General building blocks of prefix adders.

high speed and high regularity compared to carry-lookahead adders. All prefix adders consist of a pre-processing block, prefix tree and post-processing block (see Fig. 2). Different variants differ only in the topology of their prefix tree (Patil et al., 2007). The pre-processing block computes generate (g^i), transmit (t^i) and propagate (p^i) signals for every weight i based on the corresponding input bits a^i and b^i. In the prefix tree these signals are used to compute carry bits c^i in every weight i. These carrys are fed to the post-processing block which uses them together with the propagate signals to generate the sum bits s^i.

In Kheterpal et al. (2005), a Kogge-Stone adder was realized with a design-specific brick set and compared to an implementation based on a commercial 90-nm standard cell library. Results show that by using 8 different bricks, delay is only increased by 6% and area only increased by 15% compared to the standard-cell-based implementation without layout pattern restrictions despite the much more limited layout pattern set allowed in a brick based approach.

In this paper, a modified prefix operator, inherently, well suited for implementation in a brick-style fashion is proposed. Based on the properties of the brick-realization of the operator, a complete brick set for the flexible realization of arbitrary prefix adders is developed and implemented.

The paper is organized as follows: in Sect. 2 the mirror gate used for the implementation of the proposed operator is described and the motivation for its usage for brick-based adder implementations is presented. Section 3 presents the derivation of the modified prefix operator. A complete brick set for the implementation of arbitrary adders based on the proposed operator is developed in Sect. 4. In Sect. 5 a quantitative comparison between the proposed operator and the one from Brent&Kung is presented, based on 64-bit Sklansky adder implementations. The effect of the forbidden state of the proposed operator at the presence of variability is evaluated. Section 6 concludes this work.

2 Gate for brick implementation

The implementation of carry-generating gates in carry-lookahead adders using Branch-Based Logic (BBL) is a well-known and frequently used approach (Neve et al., 2002; Masgonty et al., 1991). In BBL gates, the topologies for p- and n-channel networks are created separately by forming the gate's pull-up and pull-down equations in a sum-of-product fashion (Masgonty et al., 1991).

As a result, every connection from supply to output and from output to ground forms a simple, straight transistor stack without internal branches, so intermediate parasitic capacitances are reduced to a minimum. This leads to a smaller delay and possibly to a smaller power dissipation.

The well known formula

$$c^{i+1} = g^i + t^i \cdot c^i \tag{1}$$

with

$$g^i = a^i \cdot b^i, t^i = a^i + b^i \tag{2}$$

describes the computation of the carry in bit position $i+1$ depending on the local generate and transmit condition and the incoming carry in bit position i. Equation (1) can be generalized to the form

$$G^{i:k} = G^{i:j} + T^{i:j} \cdot G^{j-1:k}, \tag{3}$$

where $G^{i:j}$ and $T^{i:j}$ denote the group carry generate and group carry propagate signals for weight i down to weight j, respectively (Zimmermann, 1997). The inverted form of Eq. (1)

$$\overline{c^{i+1}} = \overline{g^i + t^i \cdot c^i} \tag{4}$$

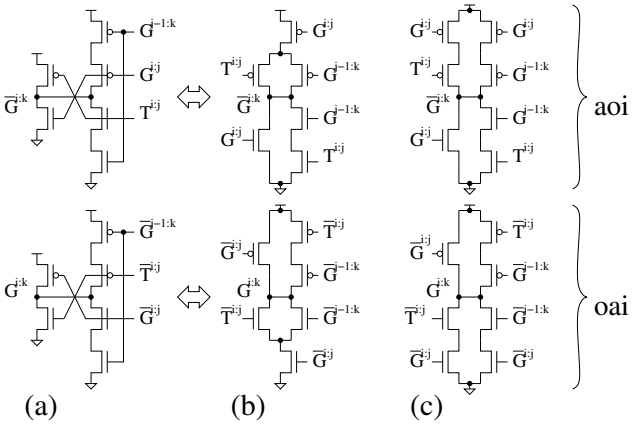

Fig. 3. Gate schematic comparison: **(a)** mirror gate, **(b)** 6-transistor complex gates, **(c)** 7-transistor complex gates.

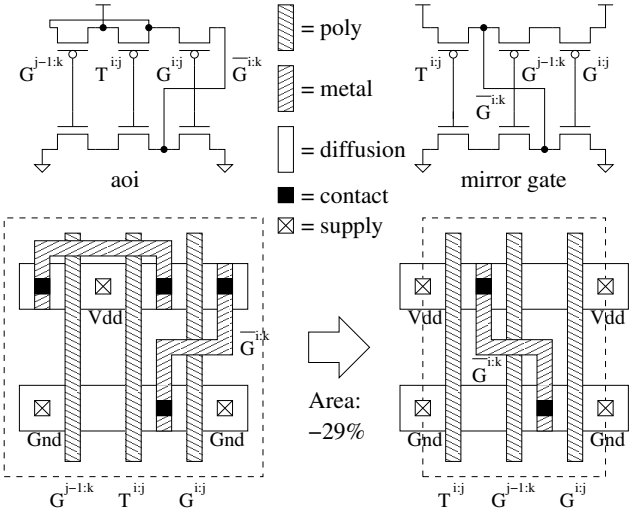

Fig. 4. Mask layout diagrams.

can be implemented using an and-or-invert gate (aoi). Because of the fact that the conditions

$$g^i = a^i \cdot b^i = 1 \text{ and } t^i = a^i + b^i = 0 \tag{5}$$

can never happen simultaneously at the outputs of the adders preprocessing block (neglecting delay effects for now), the n-channel topology of the aoi gate can also be used in the p-channel part, yielding a symmetric mirror gate which also is a BBL-style gate (Neve et al., 2002, Fig. 3). The fact represented in Eq. (5) motivates the usage of transmit conditions t^i (Weinberger and Smith, 1958) instead of the more common propagate conditions p^i (Brent and Kung, 1982) where $(g^i = 1, p^i = a^i \oplus b^i = 0)$ is possible, preventing the usage of a mirror gate.

Moreover, the dual expression

$$c^{i+1} = \overline{\overline{g^i} \cdot (\overline{t^i} + \overline{c^i})} \tag{6}$$

can be implemented using the mirror gate by interchanging g^i with $\overline{t^i}$, t^i with $\overline{g^i}$ and utilizing the absence of the input state $(\overline{g^i} = 0, \overline{t^i} = 1)$. Normally it would be necessary to implement Eq. (6) using an or-and-invert gate (oai).

Figure 3 shows the corresponding transistor schematics, depicting two alternative implementations for an oai and an aoi gate, respectively. Figure 3a shows the mirror gate with signal names corresponding to the use as a replacement for an aoi or an oai gate, respectively. Figure 3b and Fig. 3c depict different versions of complex aoi and oai gates. Figure 3c shows delay-enhanced versions which trade an additional transistor for the absence of additional branches in the n- or p-channel topologies, which is inherent in the BBL mirror gate version.

Figure 4 shows a comparison of the layout implementation of the mirror gate and the six-transistor variant of the aoi gate. The supply nodes of the mirror gate are located directly at the cell boundary in a symmetrical fashion, which is not the case for the aoi gate. This makes it possible to share the supply nodes between adjacent mirror gates, hence, reducing the area per gate significantly (−29% in 40-nm). The resulting cell rows are not only very dense, but also highly regular without any poly jog. In contrast, the seven-transistor variants of aoi and oai can also share supply contacts over cell boundaries, but not without poly jogs.

A comparison using a 90-nm technology shows that the pair delay of the mirror gate is approximately 20% smaller compared to a cascade of 6-transistor aoi and oai gates, depending on the fanout (post layout simulation/slow conditions).

3 Modified prefix operator

The main disadvantage of the described mirror gate is the existence of a so-called forbidden input state. If the forbidden input combination is applied, a short current flows from supply to ground. Taking a closer look at the mirror gate shown in Fig. 3a, it becomes clear that this happens for $T^{i:j} = 0, G^{i:j} = 1$ in the version corresponding to an aoi gate and for $\overline{T^{i:j}} = 1, \overline{G^{i:j}} = 0$ in the version corresponding to an oai gate.

As mentioned earlier, it is always possible to use the mirror gate in the first stage of the prefix graph directly after the preprocessing (see Fig. 2), because the forbidden input state is not present in the signal pairs g^i, t^i and $\overline{g^i}, \overline{t^i}$, respectively (Eq. 5) and the inputs of all prefix operators in this stage are connected only to those signal pairs (wired to the operators according to the identities $g^i = G^{i:i}$ and $t^i = T^{i:i}$).

The "●"-operator introduced by Brent and Kung (Brent and Kung, 1982) is, for any Boolean variables g, t, $\hat{g}$ and $\hat{t}$, defined as

$$(g,t)\bullet(\hat{g},\hat{t}) = (g + t \cdot \hat{g}, t \cdot \hat{t}). \tag{7}$$

Table 1. Application of "●"-operator to output of preprocessing.

g^i	t^i	g^{i-1}	t^{i-1}	$G^{i:i-1}$	$T^{i:i-1}$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	1	1	1
1	1	0	0	1	0
1	1	0	1	1	1
1	1	1	1	1	1

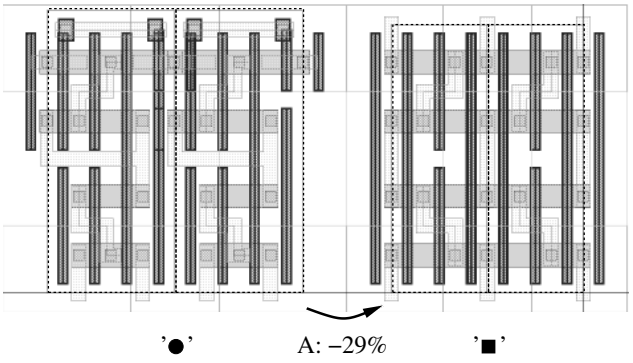

Fig. 5. ■ vs. ●: comparison of layouts (MS rules).

Using the "●"-operator on the signal pairs generated in the pre-processing block creates the forbidden state at the inputs of the following operator cells (Table 1).

The forbidden state ($G^{i:i-1}=1, T^{i:i-1}=0$) appears at the row marked in gray, because a carry would be generated at weight i, but none would be transmitted from weight $i-1$. Taking a closer look at $G^{i:i-1}$ and $T^{i:i-1}$, it is obvious that $T^{i:i-1}$ is irrelevant for the process of carry generation due to the fact that the carry is already generated because of $G^{i:i-1}=1$ (Brent and Kung, 1982).

To exploit this fact, a modified prefix operator, denoted as "■", is introduced and defined as follows

$$(g,t)\blacksquare(\hat{g},\hat{t}) = (g + t\cdot\hat{g}, g + t\cdot\hat{t}). \tag{8}$$

Comparing the truth table of the modified operator with the truth table of the "●"-operator in Table 1, the only difference is the value of $T^{i:i-1}$ in the marked row, which is one for the modified operator. So the forbidden state ($G^{i:i-1}=1, T^{i:i-1}=0$) is replaced by the allowed state ($\tilde{G}^{i:i-1}=1, \tilde{T}^{i:i-1}=1$) which is equivalent regarding carry generation. The fact that the signal pair ($\tilde{G}^{i:i-1}, \tilde{T}^{i:i-1}$) can now only take the values $(0,0)$, $(0,1)$ and $(1,1)$ (sharing this property with (g^i, t^i)) means that input and output pairs of Table 1 both do not contain the forbidden state. This is a prerequisite to use the mirror gate at any position inside a prefix tree.

The modified operator, "■", to be used as a replacement for "●" exhibits the following properties

- $c^{i+1} = \tilde{G}^i \forall i$
- "■" must be associative
- "■" must be idempotent
- $\tilde{G}^i = 1, \tilde{T}^i = 0$ must never occur

The first three properties can be proved in analogy to "●", so these proves are omitted. The last property can be proved by contradiction as follows

$$\begin{aligned}
&\tilde{G}^{i:k} = 1, \tilde{T}^{i:k} = 0 \\
\Rightarrow\ &\tilde{G}^{i:j} + \tilde{T}^{i:j}\cdot\tilde{G}^{j-1:k} = 1 \\
&\tilde{G}^{i:j} + \tilde{T}^{i:j}\cdot\tilde{T}^{j-1:k} = 0 \\
\Rightarrow\ &\tilde{G}^{j-1:k} = 1, \tilde{T}^{j-1:k} = 0 \\
j \rightarrow\ &k+2: \\
\Rightarrow\ &g^{k+1} + t^{k+1}\cdot g^k = 1, \\
&g^{k+1} + t^{k+1}\cdot t^k = 0, \\
\Rightarrow\ &g^k = a^k\cdot b^k = 1, t^k = a^k + b^k = 0 \text{ qed.}
\end{aligned}$$

4 Derivation of brick set

Now a brick set for the realization of arbitrary prefix adders is derived based on the presented modified prefix operator.

It is evident that the implementation of the operator cell(s) is crucial for the adder efficiency, which strongly suggests the layout adaption of the remaining cells to the layout of an operator cell which is implemented as efficiently as possible.

In modern deep-sub-micron CMOS technologies for which a brick-based design style is advantageous, the ratio of wiring capacitance to gate capacitance is ever increasing (Veendrick, 2000). To accommodate this fact, the bit slice width should be kept at a minimum. Therefore, and due to the shareable supply contacts of the mirror gate depicted in Fig. 4, the operator cell contains two mirror gates on top of each other as shown in Fig. 5. To show the effect of supply contact sharing, two operator cells are cascaded horizontally, which is indicated by the dashed boxes in the operator examples. Implemented in a state-of-the-art 40-nm technology, the resulting cell pitch of the proposed version is only 0.48 µm at a cell height of 1.33 µm, using minimum-sized transistors.

Figure 5 shows the comparison of "■" implemented using mirror gates and "●" using the seven-transistor aoi gate for G^i in combination with a nand-gate producing T^i. Layouts are shown for minimum spacing (MS) rules. Advantages regarding area and regularity are evident. The area saving of 29% clearly over-compensates the area increase of 15%

using the Brick-Based Design methodology which was reported in Kheterpal et al. (2005). Compared to the classical implementation based on DFM rules (Fig. 1), the area saving is as high as 54%. Typically the area of prefix adders is dominated by their prefix graph implementation. This means that a brick-oriented implementation based on "■" can be realized with a smaller area than one based on "●" in combination with the classical approach to layout generation despite the much more limited layout features available in the brick approach.

To build a complete prefix adder, the following additional cells are required:

- preprocessing:
  - generation of g^i, $\overline{g^i}$, t^i and $\overline{t^i}$, requiring nand, nor and two inverters,
  - generation of p^i from either g^i and $\overline{t^i}$ or $\overline{g^i}$ and t^i, requiring nor or nand, respectively,
- prefix graph:
  - buffer(s),
  - a single mirror-gate operator cell to use prior to post-processing,
- post-processing:
  - xor or xnor gate.

All additional bricks that were implemented are shown in Fig. 6.

In consequence of the very small cell pitch in combination with a strict horizontal well- and substrate orientation, substrate- and well contacts are not needed in every single cell. To use this as an advantage and at the same time keeping the brick set small, dedicated substrate- and well columns are used.

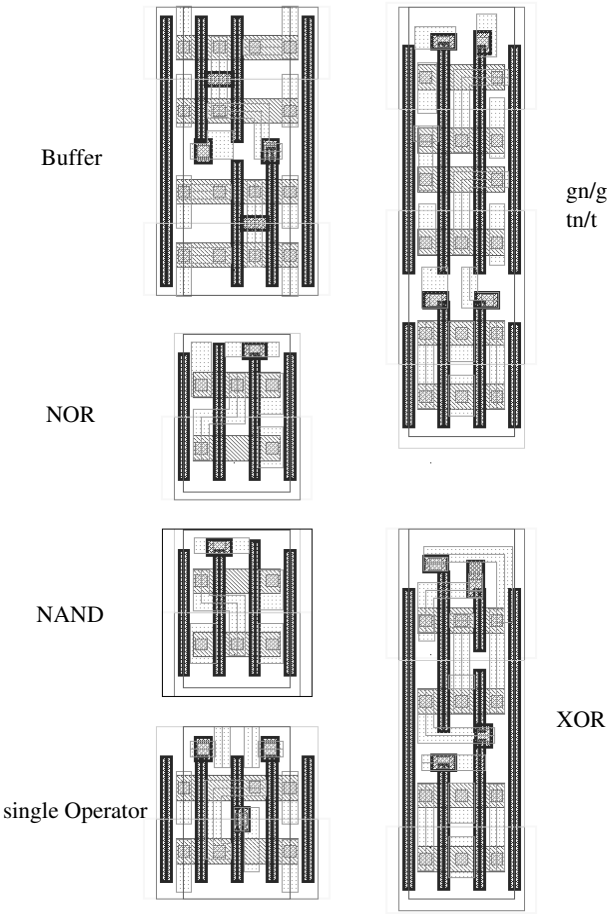

Fig. 6. Additional bricks.

5 Comparison of operators

The decision to build the quantative comparison of the proposed prefix operator with the B&K type on the Sklansky topology is based on its advantages in the ultra deep-submicron domain (Patil et al., 2007) were the modified operator is beneficial regarding a brick-style implementation. It combines the minimum possible logical depth with the minimum amount of wiring (paying the price in terms of maximum fanout), which tackles the aforementioned increasing ratio of wiring capacitance to gate input capacitance. In addition, the increased delay per gate caused by the increased ratio between threshold voltage magnitudes and supply voltage makes a small logical depth desirable.

A 64-bit Sklansky adder was implemented for each operator variant using a physically oriented design flow and minimum-space design rules. In both cases the bit slice width is determined by the width of the operator cell. Apart from the cell width, all bricks are identical in both adders except for the operators. All bricks feature minimum-sized transistors (see Fig. 6). To drive the high-fanout wires, multiple instances of minimum-sized buffers were used.

Figure 7 shows the corresponding layouts, the version based on the proposed operator exhibits an area 29% smaller than the one based on the classical Brent&Kung approach. If the influence of the larger bit slice pitch inside the conventional adder on the area of the bricks used in both versions is eliminated, the BBL-based version has still an area advantage of 23%.

While featuring almost identical slow-corner delays (slow model, 0 °C, 0.9 V) of 1121 ps ("■") and 1128 ps ("●"), the power dissipation of the BBL-based adder was about 12.5% higher (222.8 µW vs. 198.0 µW, simulated at 500 MHz under typical conditions). In terms of ATE efficiency, the BBL version is 26% better than the conventional one. If the aforementioned net logic area of the peripheral bricks is taken into account for the conventional adder, an advantage of 15% in ATE efficiency remains (Table 2). However, the adder based on "■" is 11% less efficient with respect to the

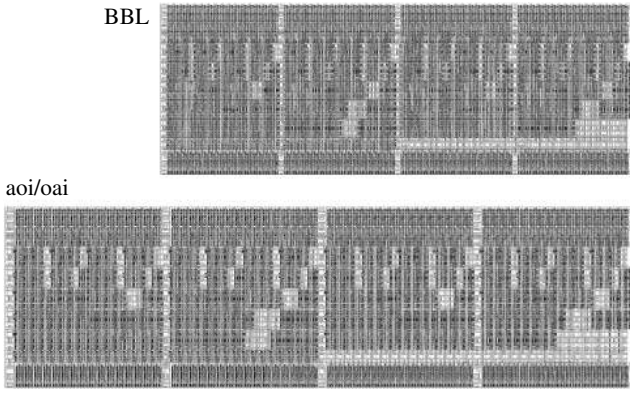

Fig. 7. 64-bit Sklansky adder: area comparison.

energy-delay-product (EDP). This is due to increased power dissipation that is discussed below.

It was already shown in Sect. 3 that the forbidden state ($G^{i:k} = 1, T^{i:k} = 0$) can not appear when ideal switching conditions are assumed. In reality, switching conditions are far from ideal, so delays between corresponding G and T signals can occur, temporarily, resulting in the forbidden state at the input of a receiving mirror cell. Compared to a regular CMOS gate under the influence of glitching, the power dissipation of the mirror gate in the forbidden state can be much higher due to the stronger short-circuit current.

To determine the fraction of the power increase caused by the denser wiring of the BBL version, an additional adder based on "■" with the bit slice pitch from the "●"-version was realized and characterized. The power dissipation of this version is still 10% higher than the one based on the classical approach, so it becomes apparent that the transient presence of the forbidden state is responsible for most of the increase.

The peak current in the supply network is an indicator for the impact of the forbidden state because of the aforementioned strong short circuit current it may cause. The results summarized in Table 2 show that the adder based on "■" exhibits a 21% higher maximum VDD current than the one occurring in the adder based on "●" (4.34 mA compared to 3.60 mA) in a 500-cycle random simulation at typical conditions.

A strong increase of supply current amplitudes can be a knock-out criterion for the usage of "■", therefore the maximum possible increase must be determined. In the considered nanometer-scale technologies the variability of the threshold voltage V_{th} can no longer be neglected and can have a large impact on delay. This makes it necessary to take a closer look at the consequences for the forbidden state (FS). BSIM version 4.5 transistor models feature a parameter to model the effects of V_{th} variability, called DELVTO (BSI, 2005). This parameter was used to model aggressive variability scenarios based on the layout-derived netlists of both adders. The research results published in Bernstein et al. (2006) show that the $\frac{3\cdot\sigma}{\mu}$ ratio of the V_{th} distribution for nominal devices at nodes below 90nm can be up to 30%.

Table 2. 64-bit adder comparison (40-nm CMOS).

	BBL	Conv.
Area/μm^2	388.2	546.2(500.9)
Delay (w.c.)/ps	1121	1128
Max. supply current/mA	4.34	3.60
Energy per cycle/μW MHz^{-1}	0.446	0.396
ATE-Efficiency (normalized)	1.26(1.15)	1

As a first step, all transistors in the netlists of both adder versions which could affect a transient occurrence of the FS were biased regarding V_{th} in a worst-case way by accelerating transitions which could possibly lead to the FS (e.g. the rise transition of $G^{i:j}$ or the fall transition of $T^{i:j}$) and at the same time slowing down transitions that are needed to exit it. All V_{th} variations had a magnitude of 30% of the respective default values. As a result, die power dissipation of the BBL adder was further increased by 6%, while the conventional version only suffered an increase by 0.5%. The current peaks were increased by 12% ("■") and 7% ("●"), respectively. Surprisingly, the power dissipations of both adders were much higher when applying the "best-case" V_{th} variation. In this case, the power increased by 12% for BBL and 18% for the conventional adder. The greater increase in the conventional version is due to the decreased fraction of power dissipation caused by the FS, which is observable at the current peaks ("■": −15%, "●": −9%). This observation shows that, at least for the considered topology and transistor widths, the effect of the FS can be clearly dominated by other sources of power dissipation. This is ephasized by the power results of five simulations based on normally-distributed random V_{th} variations covering all transistors, were the power increase compared to classical CMOS was between 13.1% and 13.2% (max. current: 18% to 22%).

To determine which maximum supply currents can be expected due to the FS, special stimuli which favor its transient occurrence were simulated. The first stimulus (i) is based on the fact that, along paths between weights, the difference in path delay between T and G can accumulate, possibly leading to FS with increased duration. On the other hand, switching activity is limited because propagate conditions need to be fullfilled. The second stimulus (ii) tries to establish the FS at the input of all operator bricks in the first row of the adder's prefix graph (Fig. 2) at once, thereby maximizing the gates in FS simultaneously. The simulations were made without V_{th} modifications as well as under the influence of worst-case V_{th} settings. Again, the conventional adder was simulated using the same setup (Table 3).

Table 3. Adder comparison: worst-case stimuli.

	BBL_{nc}	$Conv._{nc}$	BBL_{wc}	$Conv._{wc}$
$I_{max,(i)}$/mA	3.27	3.33	5.17	3.08
$E_{cyc,(i)}$/µW MHz^{-1}	0.394	0.386	0.502	0.399
$I_{max,(ii)}$/mA	6.94	6.55	9.14	7.12
$E_{cyc,(ii)}$/µW MHz^{-1}	0.541	0.541	0.547	0.528

While the supply current peaks are considerably increased in the BBL-based adder when passing over to the worst-case V_{th} distribution, the magnitudes of energy per cycle as well as peak currents are perfectly acceptable. Stimulus (i) shows a notable relative increase in energy per cycle and peak current, but both parameters have low amplitudes. Stimulus (ii), on the other hand, causes a high energy per cycle of the BBL-based adder in absolute terms, but without any important increase when applying the worst-case V_{th} distribution and without much difference to the classical implementation. Considering the variability-limiting nature of the brick-style layout approach, it can be expected that, in reality, the increase in power dissipation caused by the forbiddden state will be much lower compared to the presented values, because the results are based on extremely pessimistic assumptions regarding variability. In addition, because the brick-style approach could make the usage of restricted layout rules possible, the area savings regarding the operator cell could exceed the 54% reduction based on minimum space rules in comparison to DFM rules.

6 Conclusions

To improve the realization of arbitrary prefix-based carry-propagate adders in a brick-based design style, a modified version of the prefix operator introduced by Brent&Kung was presented. This operator is usable for any prefix tree, using a 3-input mirror gate as a replacement for the classical aoi and oai gates. It was shown that this gate is very well suited for realization as a brick. Based on the resulting operator layout, a complete brick set for prefix adder implementations was created and used to implement a 64-bit Sklansky adder. To carry out an in-depth comparison of the proposed operator with the classical one from Brent&Kung, a version of this adder based on the classical approach was also realized. Both adders were comprehensively characterized and compared, showing that the ATE efficiency of the adder based on the proposed operator is 15% higher. We have also shown that the forbidden state inherent to the modified prefix operator does not significantly increase power dissipation or current peaks.

In conclusion, the presented approach to prefix adders offers a highly efficient alternative implementation for nanometer-scale technologies where Design-for-Manufacturability and Design-for-Yield is increasingly important.

References

Bernstein, K., Frank, D. J., Gattiker, A. E., Haensch, W., Ji, B. L., Nassif, S. R., Nowak, E. J., Pearson, D. J., and Rohrer, N. J.: High-performance CMOS variability in the 65-nm regime and beyond, IBM J. Res. Dev., 50, 433–449, 2006.

Brent, R. and Kung, H.: A Regular Layout for Parallel Adders, IEEE Transactions on Computers, C-31, 260–264, 1982.

BSIM 4.5.0: Manual, available at: http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html, last accessed: 30 March 2011, 2005.

Goering, R.: PDF CEO calls for restricted layouts, available at: http://web.archive.org/web/20071201153818/www.scdsource.com/article.php%?id=32, last accessed: 30 March 2011, 2007.

Jhaveri, T., Rovner, V., Pileggi, L., Strojwas, A. J., Motiani, D., Kheterpal, V., Tong, K. Y., Hersan, T., and Pandini, D.: Maximization of layout printability/manufacturability by extreme layout regularity, Journal of Micro/Nanolithography, MEMS and MOEMS, 6, 031011, doi:10.1117/1.2781583, 2007.

Jhaveri, T., Rovner, V., Pileggi, L., Strojwas, A. J., Motiani, D., Kheterpal, V., Tong, K. Y., Hersan, T., and Pandini, D.: OPC simplification and mask cost reduction using regular design fabrics, SPIE, 7274, 727417, doi:10.1117/12.814406, 2009.

Kheterpal, V., Rovner, V., Hersan, T., Motiani, D., Takegawa, Y., Strojwas, A., and Pileggi, L.: Design Methodology for IC Manufacturability Based on Regular Logic-Bricks, Proc. DAC, 42, 353–358, 2005.

Liebmann, L., Pileggi, L., Hibbeler, J., Rovner, V., Jhaveri, T., and Northrop, G.: Simplify to survive: prescriptive layouts ensure profitable scaling to 32 nm and beyond, in: SPIE Conference Series, vol. 7275, 2009.

Masgonty, J., Arm, C., and Piguet, C.: Technology- and Power Supply-Independent Cell Library, IEEE Custom Integrated Circuits Conference, pp. 25.5.1–25.5.4, 1991.

Neve, A., Flandre, D., Schettler, H., Ludwig, T., and Hellner, G.: Design of a Branch-Based 64-bit Carry-Select Adder in 0.18 um Partially Depleted SOI CMOS, Proc. ISLPED, pp. 108–111, 2002.

Patil, D., Azizi, O., Horowitz, M., Ho, R., and Ananthraman, R.: Robust Energy-Efficient Adder Topologies, Computer Arithmetic, IEEE Symposium on, 0, 16–28, doi:http://doi.ieeecomputersociety.org/10.1109/ARITH.2007.31, 2007.

Taylor, B. and Pileggi, L.: Exact Combinatorial Optimization Methods for Physical Design of Regular Logic Bricks, Proc. DAC, 2007.

Tong, K., Rovner, V., Pileggi, L., and Kheterpal, V.: Design Methodology of Regular Logic Bricks for Robust Integrated Circuits, Proc. ICCD, 2006.

Weinberger, A. and Smith, J. L.: A Logic for High-Speed Addition, National Bureau of Standards Circular 591, pp. 3–12, 1958.

Veendrick, H.: Deep-Submicron CMOS ICs, Kluwer Academic Publishers, 2000.

Zimmermann, R.: Binary Adder Architectures for Cell-Based VLSI and Their Synthesis, PhD thesis, Swiss Federal Inst. of Technology, Zurich, 1997.