

A General Approach for Highly Defect Tolerant Parallel Prefix Adder Design

Soumya Banerjee and Wenjing Rao*

ECE Department, University of Illinois at Chicago, Chicago, IL 60607, USA
 {sbaner8, wenjing}@uic.edu

Abstract—This paper proposes a highly defect tolerant Parallel Prefix Adder (PPA) design. Motivated by the inherent defect tolerance capability displayed in a Kogge Stone Adder (KSA), this paper identifies the key elements that can be applied to make the general PPA’s defect tolerant: 1) the Generate and Propagate computing hardware is divided into disjoint groups, such that defects in one group will not “contaminate” the computation carried out by the other groups; 2) redundant copies of the results for each group can be derived cost-effectively from the other disjoint groups. This approach provides flexibilities for a defect tolerant PPA design on both the number of groups and the type of Sub-Adder structure to be adopted. As is verified by the simulation results, the proposed scheme not only offers a general way of constructing highly defect tolerant PPA’s, but also opens up a large number of pareto-front design choices, considering the objectives of reliability, hardware and performance.

I. INTRODUCTION

The scaling down of device dimensions into the nanometer range is likely to result in significantly higher defect rates during the manufacturing process of IC’s [1]. With significantly increased defect rates, defect tolerance mechanisms are necessitated to guarantee a reasonable yield. Post manufacturing reconfiguration techniques to bypass defects are already applied in memory systems and FPGA’s [2] [3]. However, such low-cost defect tolerance techniques rely heavily on the relative independence of operations of the homogeneous components, such as LUT and memory cells. Logic systems, on the other hand, usually constitute heterogeneous components with strong dependencies among each other. This makes it hard to realize fine-grained, low-cost defect tolerance schemes for a high level of defect rate.

In this paper, we focus on the design of highly defect tolerant Parallel Prefix Adders (PPA). Multiple papers have addressed the reliability issues in general adders. For instance, defect tolerant designs are proposed in [4], [5] where an adder can be configured for its approximate computation, according to the amount of accuracy required in an application. [6] applies quadruple time redundancy for an adder to tolerate error occurrences. The work in [7] applies Triple-Modular Redundancy with a voter to guarantee the correct result. [8] uses a mechanism to stretch the clock period to accommodate the critical delay paths, with minor performance sacrifice. Even though these designs have a relatively low hardware cost, they mostly assume a low level of defect occurrences, and cannot scale when defect rates are high.

Among the various adders, PPA provides a general form to represent a wide range of adder design choices [9]. Reliable PPA designs have mostly been done on the particular form of Kogge Stone Adders (KSA), due to its inherent regular

structure and hardware redundancy. For performance purposes, the hardware of a typical KSA is divided into two disjoint groups of the even bits and the odd bits. This provides a natural way to make the defects *isolated*: errors caused by the defects in one group will not affect the results produced by the other group. Furthermore, the in-built redundancy in a KSA allows each group to be capable of generating the results for the other group, with a small hardware and time overhead. Based on these features, defect tolerance mechanisms are proposed in [10] [11] for KSA’s, where as long as one of the two groups is defect-free, all the computations are done using the defect-free half of the adder to ensure correctness.

This paper presents a general approach for defect tolerant PPA that is no longer limited to the special case for KSA. The proposed approach is a scalable solution along two dimensions: the number of disjoint hardware groups (beyond two), and the variety of PPA structures that can be adopted (beyond KSA). This provides enhanced reliability, as well as opens up a large number of design choices for defect tolerant PPA designs.

II. PRELIMINARIES

A. Parallel Prefix Adders (PPA)

A PPA [9] performs addition in 3 steps:

1. **Bit-level Generate and Propagate computation:** each input bit pair A_i and B_i is used in the following expressions to compute the bit-level Generate (g_i) and Propagate (p_i) signal pair:

$$(g_i, p_i) = (A_i \cdot B_i, A_i \oplus B_i)$$

2. **Block-level Generate and Propagate computation:** from each bit-level g_i and p_i , a block level $G_{i:j}$ and $P_{i:j}$ can be calculated by:

$$(G_{i:j}, P_{i:j}) = (G_{i:k_1}, P_{i:k_1}) \circ (G_{k_1:k_2}, P_{k_1:k_2}) \circ (G_{k_2:k_3}, P_{k_2:k_3}) \circ \dots \circ (G_{k_n:j}, P_{k_n:j})$$

when $i \geq k_1 \geq k_2 \geq k_3 \geq \dots \geq k_n \geq j$.

The “ \circ ” operator is defined as:

$$(G_{i:k}, P_{i:k}) \circ (G_{k:j}, P_{k:j}) = (G_{i:k} + P_{i:k} \cdot G_{k:j}, P_{i:k} \cdot P_{k:j})$$

where $i \geq k \geq j$.

Based on the above formula, all the $G_{i:0}$ and $P_{i:0}$ are calculated for $i \in [0, n]$. Since the “ \circ ” operator is associative, the entire operation for $G_{i:j}$ and $P_{i:j}$ computation can be performed in any arbitrary order, which opens up various designs with different parallelism, delay and hardware costs.

3. **Carry and Sum computation:** after all the $G_{i:0}$ and $P_{i:0}$ signals are computed, the corresponding Carry C_i and Sum S_i for all the bits can be computed in parallel, using $G_{i-1:0}$:

*This work is supported by NSF Grant CNS-1149661

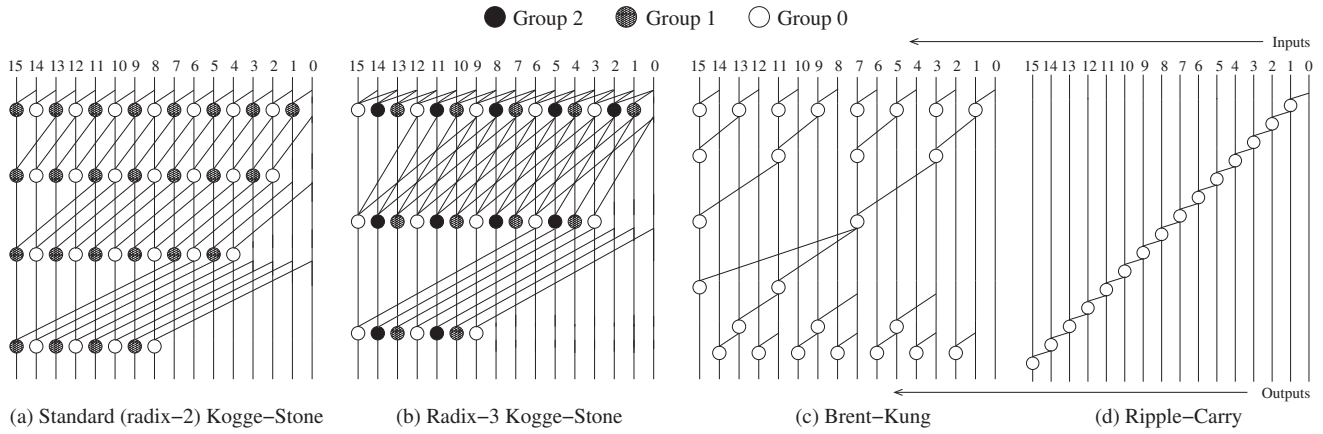


Fig. 1: The GP computation stages of various 16-bit PPA's

$$C_i = G_{i-1:0}$$

$$S_i = p_i \oplus C_i$$

For a PPA, stage 2 is the dominant part in both hardware cost and delay. Since Generate and Propagate (GP) information is computed for every bit, each relies on all the preceding bits from $(i - 1)$ to 0. Therefore the entire circuit is dominated by a large network of Generate and Propagate blocks. Various PPAs have distinct architectures for this stage. Figure 1 shows this stage of 4 PPAs, with each of the circles representing one “o” operation, or a GP Block.

B. Motivation: Beyond Defect Tolerant KSA

Figure 1(a) shows a typical (radix-2) 16-bit Kogge-Stone Adder (KSA), where the bits are divided into two disjoint hardware groups of even (Group 0) and odd (Group 1) bits, illustrated by the circles of two different patterns. As a result, any defect in one group will have their erroneous results constrained within the same group, thus not contaminating the results of the other group.

Moreover, it turns out that each group can derive a redundant copy of the results of the other group, i.e., the corresponding GP signals with a small extra cost. Such a

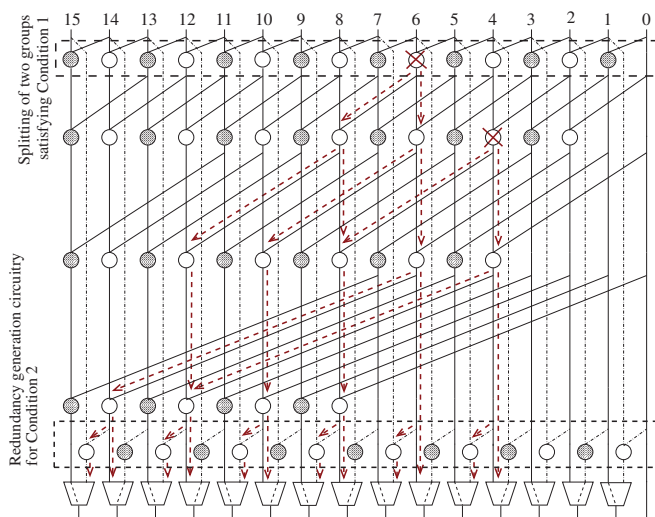


Fig. 2: The defect tolerant KSA design proposed in [10]

scheme is proposed in [10], with its structure shown in Figure 2. This design ensures the correct result when all the defects are within one group. In such a case, the defect-free group is used to generate the results for both, with the additional row of extra GP Blocks at the bottom. For example, bit 14 has two copies of $(G_{14:0}, P_{14:0})$ generated: one from the regular computation of the even group (which bit 14 belongs to), another by performing $(G_{14:14}, P_{14:14}) \circ (G_{13:0}, P_{13:0})$, obtained from bit 13, part of the odd group. When the defect is in the even group, as is shown in the figure, the first copy will be erroneous. However, the second copy, computed from the odd group, remains correct. This copy of the result is then selected by configuring the MUXes at the output end.

Essentially, the KSA structure satisfies two key conditions, which make it possible to achieve defect tolerance at such a low cost: **Condition 1**: the formation of *hardware disjoint groups*, each carries out half of the GP signal calculations independently; and **Condition 2**: the *replacing capacity* of one group's result for deriving a redundant copy of the other groups' results, thanks to the arrangement of interleaving bits into groups.

Extending the defect tolerance framework to other PPA's such as Brent-Kung (BKA), Ripple-Carry (RCA), shown in Figure 1, is not straightforward, as they do not easily satisfy the two conditions presented in KSA. For this reason, we need to take a careful look into the structure that makes such conditions hold. From Figure 2, it can be observed that Condition 1 for KSA is essentially enabled by the first level of GP Blocks (denoted as *Group-Split*) only. Then, all the communications are strictly contained within each group. Therefore, this first level of GP Blocks of a KSA can in fact be extracted out, to satisfy Condition 1 for other general PPA structures. Condition 2, it turns out, is also not limited to the specific structure of KSA's. As long as the groups are formed by interleaving bits, and an additional level of GP Blocks (denoted as *Redundancy-Generation*) are added at the end, each group can guarantee to generate a redundant copy of results for all the other groups. The structure between those two levels do not really impose any constraints for either Condition 1 or Condition 2, thus are customizable.

Based on these two observations, Figure 3(a) illustrates a radix-2 KSA in a reorganized form, highlighting the three stages, where the two *Sub-Adder* structures in the middle

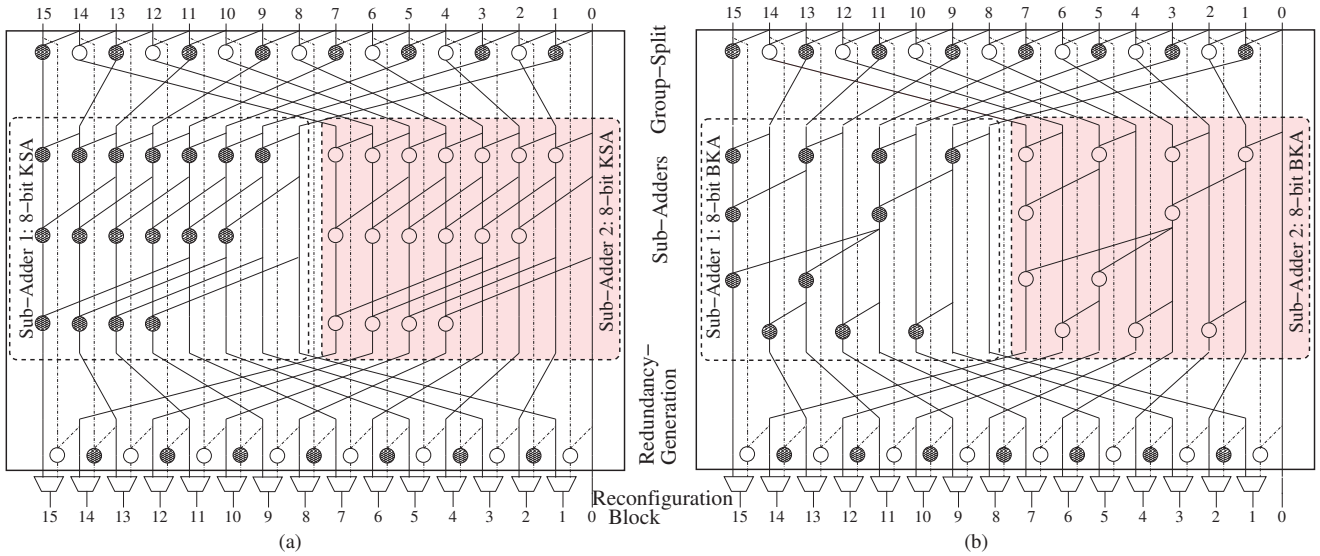


Fig. 3: (a) Radix-2 KSA with the presence of Sub-Adders, and (b) using Brent-Kung Adders (BKA's) as Sub-Adders

take the form of two 8-bit KSAs. It is worth noting that, the Sub-Adders are only responsible for the computation of GP information for the corresponding bits of the same group. However, each is identical to the form of an 8-bit KSA, except that the interleaved bits (even and odd bits) are taken in each as inputs, instead of the consecutive ones as in a regular 8-bit KSA.

Thus, the two groups of a defect tolerant KSA, (as the two Sub-Adders) being independent from one another, can be replaced in block by another adder structure without affecting the functionality or the defect tolerance ability. As is shown in Figure 3(b), the corresponding Sub-Adders are replaced by Brent-Kung Adders (BKA's). Such a replacement can be done in general with any PPA structure, thus opening up new choices in defect tolerant designs.

Under a high defect rate, chances are exceedingly low that all the defects will be concentrating within only one group out of two. Therefore, the number of groups needs to be increased to enhance the likelihood of having at least one defect free functional group. Such an expansion on the number of groups can be achieved by exploring higher radices [12]. The radix of a PPA is defined by how many maximum “ \circ ” operations are done inside a single GP Block. A radix-3 KSA example is shown in Figure 1(b), with 3 disjoint groups: Group 2 (bit Mod 3 = 2), Group 1 (bit Mod 3 = 1) and Group 0 (bit Mod 3 = 0). Defect tolerance can be achieved in this case, by letting each group generate two additional results for the other two groups. With every bit having 3 copies of the result, at least one defect free group will ensure the success of the defect tolerance mechanism. Overall, increasing the number of disjoint groups raises the level of defect tolerance.

III. DEFECT TOLERANT PPA DESIGN

Figure 4 shows a generalized example of the proposed design with 3 groups in a 24-bit PPA. The Group-Split stage resembles the first level of a radix-3 KSA, which guarantees that the computations of the 3 groups will be carried out by disjoint hardware. In the subsequent levels, the three groups (8-bit Sub-Adders) independently carry out the computations of

GP signals. Their specific structure can be of any PPA design. In the end, at the Redundancy-Generation stage, two additional GP Blocks are added to each bit for the redundant results for the other two groups. As an example, bit 12 (belonging to Group 0) generates a copy of its result from its own group's hardware. In addition, one redundant copy is derived by: $(G_{12:12}, P_{12:12}) \circ (G_{11:0}, P_{11:0})$ from the result of Group 2, and another redundant copy $(G_{12:12}, P_{12:12}) \circ (G_{11:11}, P_{11:11}) \circ (G_{10:0}, P_{10:0})$ is derived from the result of Group 1. As long as one group is defect-free, the correct result for every bit can be guaranteed by configuring the MUXes at the output end.

In the overall design, the extra hardware needed in the Group-Split stage and the final Redundancy-Generation stage is determined by the number of groups in the design. The type of Sub-Adders is determined independent of the group number. The entire defect tolerant PPA structure can thus be characterized by two parameters: the number of groups, and the type of the Sub-Adders.

We can formalize the design approach with k -groups by dividing the framework into the following multiple stages:

Group-Split: This stage takes the same form as the first level of a radix- k KSA, where bit i interacts with bits $(i-1)$, $(i-2)$, $(i-3)$, till $(i-k+1)$ to generate $G_{i:i-k+1}$ and $P_{i:i-k+1}$. Thus each bit $(i \geq (k-1))$ requires a k -input GP Block at this stage. This stage forms the k groups, which will be independent of each other with disjoint hardware.

Sub-Adders: Once the groups are formed, all the bits are computed via a corresponding Sub-Adder: bit i will be grouped with bits $(i \pm k)$, $(i \pm 2k)$ and so on, to be processed by the $(i \bmod k)^{th}$ Sub-Adder. For k groups, there will be k Sub-Adders. At this stage, the Sub-Adders can be of any PPA design. For bit i in a group (Sub-Adder), the following operations are performed to compute $G_{i:0}$ and $P_{i:0}$:

$$\begin{aligned} (G_{i:0}, P_{i:0}) &= (G_{i:i-k+1}, P_{i:i-k+1}) \\ &\circ (G_{i-k:i-2k+1}, P_{i-k:i-2k+1}) \\ &\circ (G_{i-2k:i-3k+1}, P_{i-2k:i-3k+1}) \circ \dots \circ (G_{i-mk:0}, P_{i-mk:0}) \end{aligned}$$

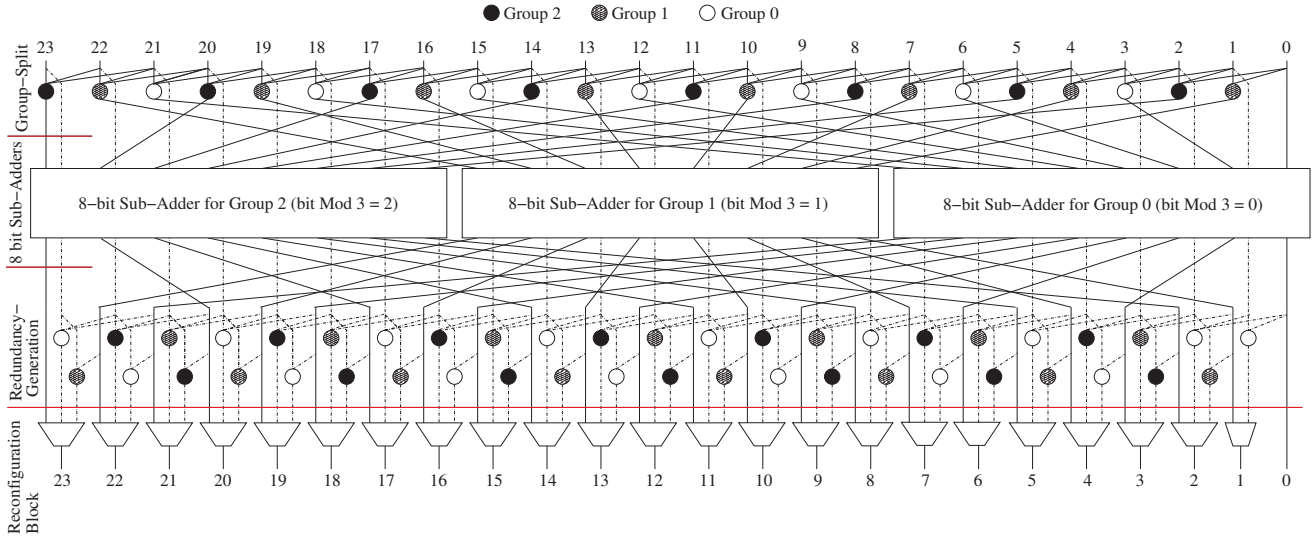


Fig. 4: An example of the proposed adder design approach with 24-bits and 3 groups

In this equation, $(i - mk)$ is any bit where $(i - mk) < k$ and $(i - mk) \geq 0$. The bits $i, (i - k), (i - 2k), \dots (i - mk)$ belong to the same group, such that $(i - mk) < k$ and $(i - mk) \geq 0$, are part of the same group. The corresponding Sub-Adder is responsible for the computation of $G_{i:0}$ and $P_{i:0}$.

Redundancy-Generation: At this stage, each bit generates $(k - 1)$ additional results by utilizing the extra GP Blocks. Specifically, bit i performs the following operations to generate redundant copies of its results i_1, i_2, i_3 to i_{k-1} :

$$\begin{aligned}
 i_1 &= (G_{i:i}, P_{i:i}) \circ (G_{i-1:0}, P_{i-1:0}) \\
 i_2 &= (G_{i:i}, P_{i:i}) \circ (G_{i-1:i-1}, P_{i-1:i-1}) \circ (G_{i-2:0}, P_{i-2:0}) \\
 &\dots \\
 i_{k-1} &= (G_{i:i}, P_{i:i}) \circ (G_{i-1:i-1}, P_{i-1:i-1}) \circ \dots \\
 &\circ (G_{i-k+2:i-k+2}, P_{i-k+2:i-k+2}) \circ (G_{i-k+1:0}, P_{i-k+1:0})
 \end{aligned}$$

Once all the results are available, the correct result can be chosen to guarantee the functionality of the adder.

IV. EXPERIMENTAL RESULTS

We carried out the simulation for the proposed defect tolerant PPA design with various instances using a C program. All the analyses are done on 64-bit adders, considering defects at the transistor level. This model assumes that any defective transistor will result in erroneous behavior is on the pessimistic side, and covers most of the interconnect defects as well. We analyze the reliability of various PPA implementation at the Sub-Adder stage, and the associated hardware requirements, with KSA, Han-Carlson (HCA), Ladner-Fischer (LFA), Brent-Kung (BKA) and Ripple-Carry (RCA) as the Sub-Adders.

The proposed PPA design is compared with a standard N-Adder defect tolerance approach: one out of N adders is selected by a MUX to be used for defect tolerance. Thus, as long as one of the adders is defect free, the defects can be tolerated. In this way, an N-Adder approach is comparable to an N -group configuration of the proposed approach, except that the redundancy is organized “externally” in an N-Adder approach, rather than “internally” as in the proposed approach.

The N-Adder approach is similar in hardware cost to the N-Modular Redundancy (NMR) approach. However, an NMR approach requires the majority of the adders to function properly for the correct result to be voted out, to deal with dynamically occurring faults. An N-Adder approach, on the other hand, requires only a single adder to be defect free to bypass defects that have been diagnosed.

In both the proposed approach and the N-Adder design, a block of MUXes is needed to select the correct output. The correctness of such a reconfiguration block is crucial for the entire defect tolerance scheme. Thus we assume that in both the cases, it is guaranteed reliable without any defects, possibly by adopting highly reliable devices.

To maintain the generality in the proposed approach for devices with any dimension, we evaluate the quality of the proposed design approach with the following generic parameters:

Reliability is calculated by randomly generating a set of defective devices, and determining if the defect can be tolerated in the adder. This is done for 1000 times, and the reliability is determined by the ratio of the successful cases to the total number of cases (1000) considered.

Performance (and also **Delay**) is indicated by the sum of fan-ins of the GP Blocks along the critical path. From [12], it is known that an n -input GP Block usually has less than n/m times more delay than an m input GP Block, when $n > m$. In this paper, we evaluate the performance of GP Blocks via the lower bound case of their number of inputs. Thus, we consider 2-input GP Block 1.5 times faster than 3-input GP Blocks and 2 times faster than 4-input GP Blocks, etc. In this way, the sum of fan-ins of the GP Blocks along the critical path indicates an upper limit of the delay, or the lower limit of performance. As an example, a 64-bit 1-group RCA has a fan-in sum of the GP-Blocks along the critical path equal to 126 (= 63×2), whereas, a 64-bit 4-group RCA has the same fan-in sum of 34 (4 for Group-Split stage and 30 for Sub-Adder stage)+ additional degradation due to Redundancy-Generation stage. The possible values for additional degradation can be either 0 for no defect condition, or 2, 3, or 4 depending upon

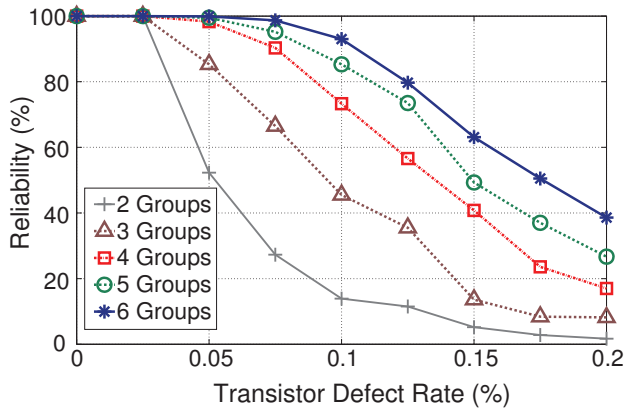


Fig. 5: Reliability Analysis for the proposed design with 64-bit Brent-Kung Adder

the number of defective groups. Thus, the fan-in sum along the critical path can be either of these values: 34, 36, 37, or 38.

Required **hardware** is measured by the total number of transistors required for CMOS implementation. Basically, an i -input GP Block will require $(6 \times i) + 2$ number of transistors (including both PMOS and NMOS transistors).

In this section, by an adder with certain number of groups, we mean the type of adder used in the Sub-Adder section of the proposed structure. As an example, by a 64-bit RCA with 4 groups, we mean that the structure has an input of 64-bits, with 4 groups, each of type RCA of 16-bits.

A. Reliability Analysis

Figure 5 shows the reliability analysis for various 64-bit BKA's. Each reliability curve is plotted with defect rates at the transistor level, shown for 2 to 6 groups. Reliability (represented in percentage) is given by the ratio of the number of instances where the defects are tolerated, to the total number of trials.

Figure 5 verifies the effect of increasing the number of groups, which consistently delivers enhanced reliability. As an example, for BKA with transistor level defect rates of 0.1%, 2 groups can only tolerate the defects for less than 20% of the time. Reliability is increased to more than 70%, when the number of groups is increased to 4. It can be further enhanced to more than 90% if the number of groups is 6. This trend of enhanced reliability due to increased number of groups is generally observed to be true for all types of Sub-Adders.

Figure 6 depicts the reliability comparison between N-Adder and the proposed approach for 64-bit BKA's. The proposed scheme is superior to the N-Adder approach in both cases of $N = 3$ and 5. This is because for an N-Adder approach, a single defect is enough to deem an adder unusable. In the proposed approach, since the Redundancy-Generation stage is significantly large, any defect there will leave more room for the actual adder to perform without any error affecting it. We will show that the reliability gap between N-Adder (and thus NMR) and the proposed approach for higher performance adders to be even greater.

B. Expansion of Reliable PPA Design Space Choices

In Figure 7, various designs from the proposed approach is shown together with the N-Adder based approach to illustrate

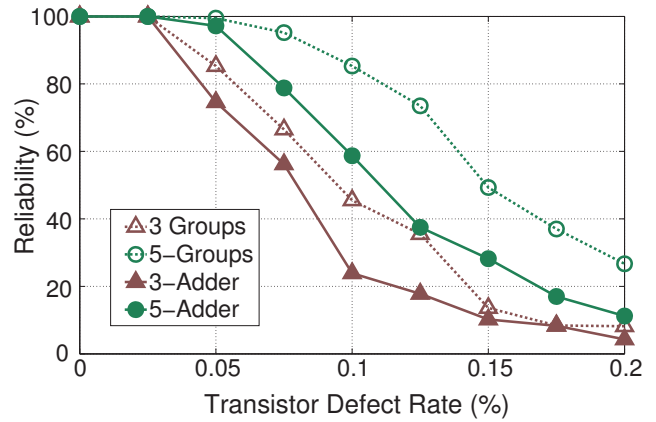


Fig. 6: Reliability comparison between the proposed design versus the N-Adder approach with 64-bit Brent-Kung Adder

their positions in the design trade-off space. Comparison is done by considering three metrics: reliability, transistor number, and delay (indicated by the sum of fan-ins of the GP Blocks along the critical path).

Figure 7(a), (b) and (c) show the design of PPA's that can meet the reliability requirements of $> 50\%$, $> 70\%$, and $> 90\%$, respectively, under a defect rate of 0.1%.

Overall, for the N-Adder approach, only simple types of adders such as RCA and BKA can meet the reliability requirements, while higher performance adder designs (such as KSA, HCA etc) fail to deliver even $> 50\%$ reliability. In fact, no other N-Adder approaches other than RCA can deliver a reliability of $> 70\%$. The proposed scheme, however, can deliver a number of solutions with RCA, BKA, LFA, HCA and KSA that meet various reliability requirements. Overall, the low delay, high performance region is dominated by the proposed design approach, while the N-Adder approach occupies the complementary region where the delay is high and the hardware requirement is low. Overall, the proposed design approach expands the design space by delivering highly reliable PPA's of a variety of types for any given defect rate.

C. Delay Variation in the Proposed Scheme

The proposed approach results in some performance variations over a non defect-tolerant single group PPA:

1) An n -bit k -group design with the same type of adder structure at the (parallelly operating) Sub-Adder stage will have each of the Sub-Adder to be (n/k) bits wide, and effectively reducing the delay due to parallelism, compared to the same adder structure of an original n -bit implementation.

2) With the increasing number of groups, the number of inputs to each GP Block in the Group-Split stage increases, thus adding delay to the critical path. In addition, the presence of higher input GP Blocks at the Redundancy-Generation stage also introduces extra delay, particularly for tolerating a higher number of defects.

For lower-performance adders (such as RCA), factor 1 is dominant due to reduction in delay in linear scale, while the performance degradation is negligible in comparison. It is evident from Figure 7, a 4-Adder approach has a delay of > 120 , while the comparable 4-group approach has a delay of < 40 , effectively delivering 3-fold performance enhancement. For higher performance adders, factor 2 is dominant, since

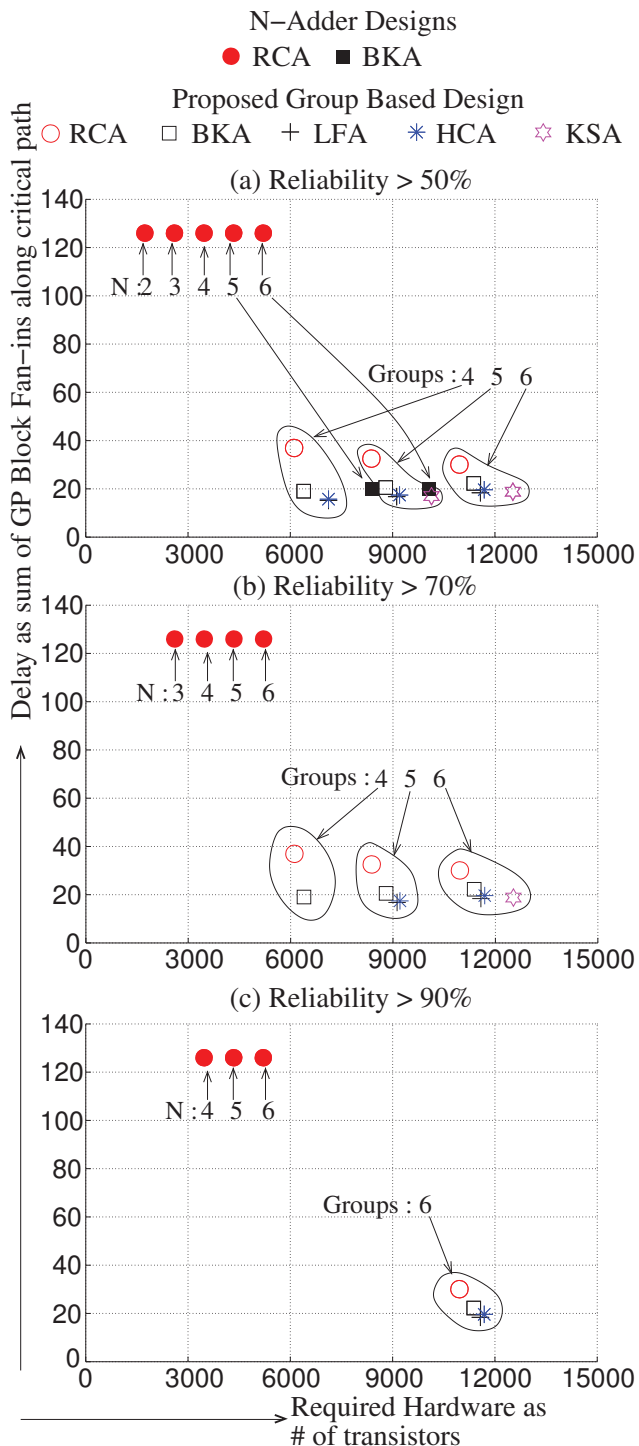


Fig. 7: Design trade-off points for various 64-bit adders at 0.1% defect rate.

the reduction in the total delay due to factor 1 occurs in a logarithmic scale. However, as is shown in Figure 7, the performance degradation for BKA, LFA, HCA and KSA are negligible with the increasing number of groups, in the proposed approach.

V. CONCLUSION

A generalized defect tolerant architecture for PPAs is proposed in this paper, delivering significantly enhanced reliability against a high level of defect rates. The architecture is constructed by gaining insights of the two unique conditions displayed in KSA, and extending it to the general PPA structures. This not only makes it possible to enhance defect tolerance capabilities, but also brings a plethora of design choices to the table of defect-tolerant adder designs. Simulation results show that the increase in the number of groups provide increased defect tolerance capabilities with reasonable hardware overhead. The proposed approach also demonstrated the superiority in reliability comparing to a regular N-Adder approach, as well as its complementary positions in the design-trade-off space.

REFERENCES

- [1] ITRS, "International Technology Roadmap for Semiconductors, Emerging Research Devices", 2013.
- [2] W.P. Shi and W.K. Fuchs, "Probabilistic Analysis and Algorithms for Reconfiguration of Memory Arrays", in *IEEE Trans. on Computer Aided Design*, volume 11, no. 9, pp. 1153–1160, Sept 1992.
- [3] F. Hatori, T. Sakurai, K. Nogami, K. Sawada, M. Takahashi, M. Ichida, M. Uchida, I. Yoshii, Y. Kawahara, T. Hibi, Y. Saeki, H. Muroga, A. Tanaka and K. Kanzaki, "Introducing Redundancy in Field Programmable Gate Arrays", in *Custom Integrated Circuits Conference, 1993 Proceedings of the IEEE*, pp. 7.1.1–7.1.4, 1993.
- [4] A.B. Kahng and S. Kang, "Accuracy-Configurable Adder for Approximate Arithmetic Designs", *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pp. 820–825, June 2012.
- [5] R. Ye, T. Wang, F. Yuan, R. Kumar and Q. Xu, "On Reconfiguration-oriented Approximate Adder Design and Its Application", in *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*, pp. 48–54, 2013.
- [6] W.J. Townsend, J.A. Abraham and E.E. Swartzlander, "Quadruple Time Redundancy Adders", *Defect and Fault Tolerance in VLSI Systems, 2003. Proceedings. 18th IEEE International Symposium on*, pp. 250–256, November 2003.
- [7] B.W. Johnson, "Design and Analysis of Fault Tolerant Digital Systems", *Addison Wesley Publishing Company*, 1989.
- [8] S. Ghosh, P. Ndai, S. Bhunia and K. Roy, "Tolerance to Small Delay Defects by Adaptive Clock Stretching", *On-Line Testing Symposium, 2007. IOLTS 07, 13th IEEE International*, pp. 244–252, July 2007.
- [9] Behrooz Parhami, *Computer Arithmetic Algorithms and Hardware Designs*, Oxford University Press, New York, 2nd Edition, 2010.
- [10] P. Ndai, Shih-Lien Lu, Dinesh Somesekhar and K. Roy, "Fine Grained Redundancy in Adders", *Quality Electronic Design, 2007. ISQED '07. 8th International Symposium on*, pp. 317–321, March 2007.
- [11] S. Ghosh, P. Ndai and K. Roy, "A Novel Low Overhead Fault Tolerant Kogge-Stone Adder Using Adaptive Clocking", *Design, Automation and Test in Europe, 2008. DATE '08*, pp. 366–371, March 2008.
- [12] F.K. Gurkayna, Y. Leblebici, L. Chaouati and P.J. McGuiness, "Higher Radix Kogge-Stone Parallel Prefix Adder Architectures", *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 5, pp. 609–612, May 2000.