3D Stacked Wide-Operand Adders: A Case Study

George R. Voicu, Mihai Lefter, Marius Enachescu, Sorin D. Cotofana
Faculty of EE, Mathematics and CS
Delft University of Technology
Delft, The Netherlands
{G.R.Voicu, M.Lefter, M.Enachescu, S.D.Cotofana}@tudelft.nl

Abstract—**In this paper, we address the design of wide-operand addition units in the context of the emerging Through-Silicon Vias (TSV) based 3D Stacked IC (3D-SIC) technology. To this end we first identify and classify the potential of the direct folding approach on existing fast prefix adders, and then discuss the cost and performance of each strategy. Our analysis identifies as a major direct folding drawback the utilization of different structures on each tier. Thus, in order to alleviate this, we propose a novel 3D Stacked Hybrid Prefix/Carry-Select Adder with identical tier structure, which potentially makes the manufacturing of hardware wide-operand adders a reality. Such an N-bit carry select adder can be implemented with K identical tier stacked ICs, where each tier contains two N/K-bit fast prefix adders operating in parallel according to the computation anticipation principle. Their carry-out signals are cascaded through TSVs in order to perform the selection of the sums accordingly, which results in a delay with the asymptotic notation of $O(\log(N/K) + K)$. To evaluate the practical implications of direct folding and of the hybrid prefix/carry-select approaches we perform a thorough case study of $65\,\text{nm}$ CMOS 3D adder implementations for different operand sizes and number of tiers, and analyze various possible design tradeoffs. Our simulations indicate the hybrid prefix/carry-select approach can achieve speed gains over 3D folding based designs of between 29% and 54%, for 512-bit up to 4096-bit adders, respectively. Even though 3D folding requires less real estate, when considering a more appropriate metric for 3D design, i.e., delay-footprint-cost product, the hybrid prefix/carry-select approach substantially outperforms the folding one and provides delay-footprint-cost reductions between 17.97% and 94.05%.**

Index Terms—**Adders, Cryptography, Three-dimensional integrated circuits, Through-silicon vias.**

I. Introduction

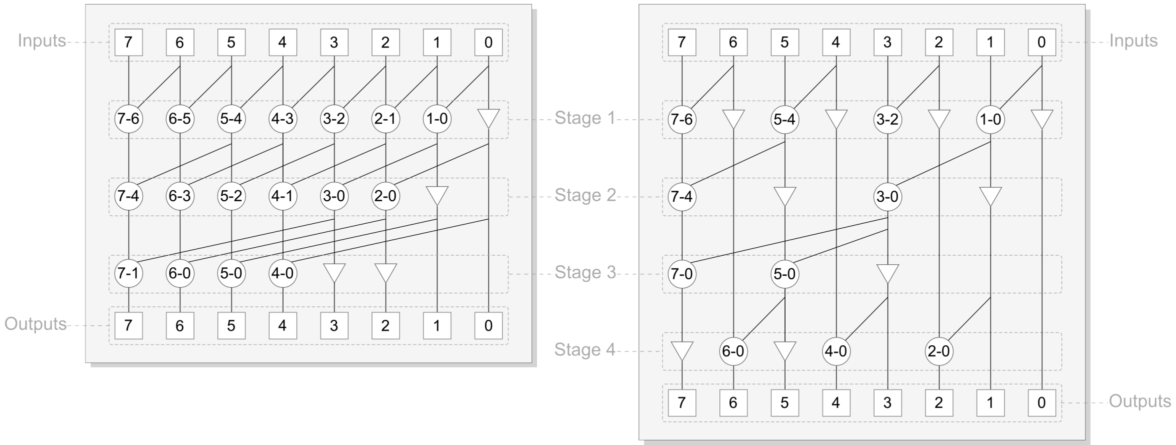
Today, almost any computing device has stringent requirements in terms of security, coming from privacy concerns, restricted content, restricted access, etc. Data encryption is one solution to address this, and public-key cryptography [1] is a fundamental and widely used encryption system. For example, RSA [2] is the dominant cryptographic algorithm used in key exchange in secure communications over the Internet. The security of any cryptographic system is proportional with the encryption key length, so the larger the key is, the better. Currently, 1024-bit keys are considered sufficient for RSA algorithms, but continuous advances in raw computation power or integer factorization theory will require the increase of this value even further, to fulfil the application security demands.

Large key length cryptography relies on intensive utilization of arithmetic operations on wide-operands. Traditionally, these operations are implemented in software on cryptographic co-processors since hardware only wide-operand arithmetic units are impractical, when making use of the current mainstream planar Integrated Circuits (ICs) fabrication technologies.

As an alternative to planar technology, 3D Stacking Integrated Circuits (3D-SIC) technology has emerged as a solution in improving the performance of a circuit by reducing the global wire-length and the design footprint [3]. The idea behind the 3D-SIC approach is to partition a large design in several smaller parts, and to implement each of them on a separate die. The dies are stacked and bonded together, and signals travel between the tiers in the stack using special vias, i.e., Through Silicon Vias (TSVs). In this way, blocks placed in the planar case far away from each other and connected by long global wires can now be stacked on top of each other and communicate through the short and low-resistance TSVs.

In this paper we investigate the implications of using 3D-SIC technology in designing efficient wide-operands adders, to be potentially included in cryptographic coprocessors. We first identify direct folding strategies of fast adder designs, i.e., prefix tree adders, and provide a generalization and a classification for partitioning an N-bit operand width adder on K-tiers. We theoretically analyze in terms of cost and performance the 3D folding classes. Given that our analysis identifies the utilisation of different structures on each tier as a major drawback of the direct folding due to major augmentations of the manufacturing cost, we subsequently address this issue and propose a 3D Stacked Hybrid Prefix/Carry-Select Adder. Each tier contains two identical N/K-bit fast prefix adders that compute in parallel the sums corresponding to a carry-in signal of both high and low value. Subsequently, the carry-out signals are transmitted, from the least significant tier towards the most significant tier, through TSVs in order to perform the selection of the sums accordingly. Since the layout of each tier can now be the same, the manufacturing costs are diminished.

To evaluate the implications of wide-operand adders in the context of 3D stacking, we perform a thorough case study of $65\,\text{nm}$ CMOS 3D adder implementations with operand widths varying from 512 to 4096 bits, and number of tiers in the range of 2 to 16 tiers. The new design space dimensions introduced by 3D stacking, i.e., the number of available tiers and the adder partitioning (3D folding) strategy, create new trade-off opportunities. As our simulations indicate, there is a clear delay vs. number of tiers trade-off, and, the optimal number of tiers,



(a) Kogge-Stone.

(b) Brent-Kung.

Figure 1: 8-bit Parallel Prefix Adder.

i.e., the 3D stack height providing the smallest delay, grows with the increase of the adder's width. We considered the following metrics in our evaluation: delay, footprint, footprint-delay product, and footprint-delay-cost product. Based on our simulation experiments we can conclude that the 3D stacked hybrid prefix/carry-select approach is overall faster than the 3D folding approach, with delay improvements of up to 29% for a 512-bit adder and up to 54% for a 4096-bit adder. In terms of footprint, the folding approach requires less chip real estate, which however, has a very high manufacturing cost. This is induced by the fact that folded 3D adders are formed out of tiers implementing different circuits, and not of the same type as it is the case for the 3D stacked hybrid prefix/carry-select designs. Based on the footprint-delay-cost product, which is more appropriate to capture the complexity of a 3D implementation, we can conclude that the hybrid prefix/carry-select approach is more suitable for 3D stacked integration, achieving a reduction of the footprint-delay-cost product over 3D folded adders between 17.97% and 94.05%.

The remainder of the paper is organized as follows. Section II motivates our work and gives a brief overview of relevant state of the art literature. In Section III different straightforward implementations of folded 3D adders are classified and analyzed. In Section IV a novel 3D stacked hybrid prefix/carry-select adder with identical tier structure is proposed. Section V presents an experimental design space exploration of various trade-offs in terms of delay, footprint, and cost for 3D folded and hybrid wide-adders. Finally, Section VI concludes this paper.

II. Motivation and related work

Addition is the primary mechanism to implement more complex arithmetic operations, e.g., multiplication, division, etc. If addition is slow or expensive, all other operations suffer in speed and/or cost. It is well known that carry propagation is the limiting factor for the performance of any adder with fixed-radix number representation [4]. Carry-Lookahead Adders (CLAs), in which the computation of independent carries is done in parallel and in advance of the sum computation, can significantly speed-up the addition and most of the currently implemented high-performance adders make use of a parallel prefix carry computation scheme, which is a particular case of carry-lookahead.

Among the prefix calculation schemes, Kogge-Stone [5] and Brent-Kung [6] represent the two extremities of the theoretical design space interval determined by the area-delay trade-off [4]. Figure 1 depicts the 8-bit carry prefix graph for these two adder types. The squares on the first row compute the propagate and generate signals for each bit position, while the circles are carry operator cells (also known as carry-merge cells), in which combined propagate and generate signals are computed. The Kogge-Stone approach offers the fastest result (fewer stages) at the expense of the largest number of carry operator cells, while Brent-Kung has the lowest number of carry operator cells, but with larger fan-out, and more propagation stages on the critical path.

While in theory various prefix strategy combinations are possible, when it comes to operations on wide numbers, the area-delay design space range of prefix adders (or of any fast adder for that matter) is reduced. The dominant factor in the adder speed shifts from the computation delay to the communication delay. The large number of carry-merge cells increases the distance between two connected cells, which in turn demands long and dense wires [7]. The solution to address the signal loss on such long wires is to instantiate additional buffers. However, this introduces a delay degradation and the routing congestion problem gets even worse.

This positive feedback loop makes the delay of wide adders to grow more than linear when the operand width is doubled, as opposed to small-operand adders. This trend can be clearly observed in the delay plots for Kogge-Stone and Brent-Kung adders in Figure 2. The values are obtained from simulations under worst-case conditions of sign-off implementations in a commercial low-power $65\,\mathrm{nm}$ CMOS technology. We note that the 4096-bit Kogge-Stone adder could not be successfully routed, thus the presented delay value is an optimistic one. In

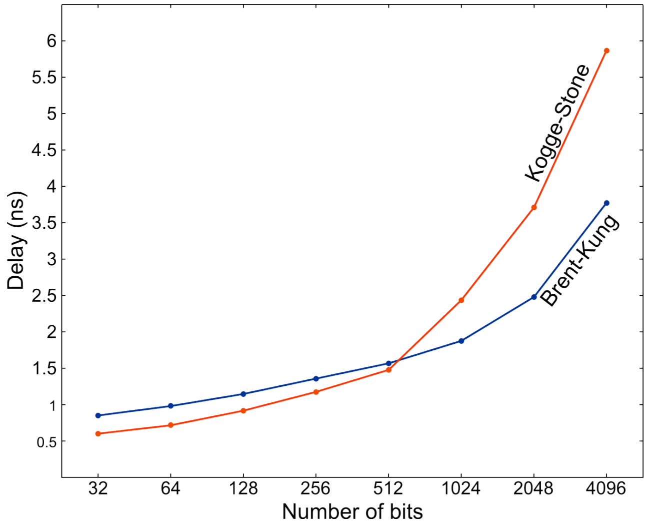


Figure 2: Propagation delay of planar prefix adders.

addition to the considerable decrease in performance, a wide adder implemented in a planar fashion has a higher fabrication cost due to the decrease in manufacturing yield caused by the large footprint and dense routing.

Thus, as a consequence of the aforementioned drawbacks, a high-performance wide-operand adder is impractical to implement in current mainstream planar IC fabrication technologies. An alternative to this is the use of 3D Stacking Integrated Circuits (3D-SIC) technology. One of the promises of this emergent technology is the reduction of global wire-length, which in turn leads to an increase in the speed of a wire-dominated circuit.

Various ways to partition prefix tree adders for 3D stacking were devised and, and several case studies for small-width adders were performed. Vaidyanathan et al. proposed in [8] to split the prefix tree in-between computation stages and estimated a maximum $4\times$ wire length reduction for 32-bit Kogge-Stone adders on 3 tiers. Puttaswamy and Loh [9] use a 2-tier bit-splitting approach where carry-merge cells corresponding to odd operand bits in a Kogge-Stone adder reside on one tier, and cells for even operand bits reside on the other die. For sparser prefix trees like Sklansky and Brent-Kung adders adjacent processing nodes are stacked. However, in the case of Brent-Kung adders the splitting of the inverse carry tree from the last stages of computations is not discussed.

A variation of the previously mentioned bit-splitting partitioning strategy for Kogge-Stone adders is used by Ouyang et al. [10], with the first carry merge stage performing ternary operations instead of binary ones, in order to match a 3-tier stacking technology. The direction of carry forwarding in the first merge stage is claimed to be flipped in order to conveniently generate the final sum, but the implications of this change on the correct functionality of the adder are not presented. Thus, we will only consider classic Kogge-Stone prefix adders as a discussion vehicle in the next section to classify the 3D partitioning strategies and analyze them in the context of wide-operand adders.

III. 3D Partitioning of Parallel Prefix Adders

The straightforward way to design a 3D stacked fast adder is to take an existing planar prefix adder, partition it, and fold the resulting partitions such that each one is placed on a separate die. Based on the existing examples found in the literature and presented in the previous section, we generalize the 3D partitioning strategies of an N-bit parallel prefix adder on a K-tier stack as follows:

1) *Stage Folding*: the carry-merge cells in each stage are placed on one tier, as suggested in [8],
2) *Bit Interleaving*: the carry merge cells on each and every K-th column in the prefix graph are placed on the same tier, as suggested in [9].

In addition to these, we also identify a third type of partitioning:

3) *Bit-slice Folding*: the carry merge cells on every N/K consecutive columns from the prefix graph are placed on the same tier.

Figure 3 graphically depicts an example of how an 8-bit Kogge-Stone adder can be divided across a 4-tier stack according to the three identified types of partitioning strategies. We note that even though Stage Folding of an N-bit Kogge-Stone adder demands $K = 1 + \log_2 N$ layers, any number of tiers can be accommodated if we group several stages together on the same die. In the figure we use only 3-tiers since the carry prefix tree has only 3 stages.

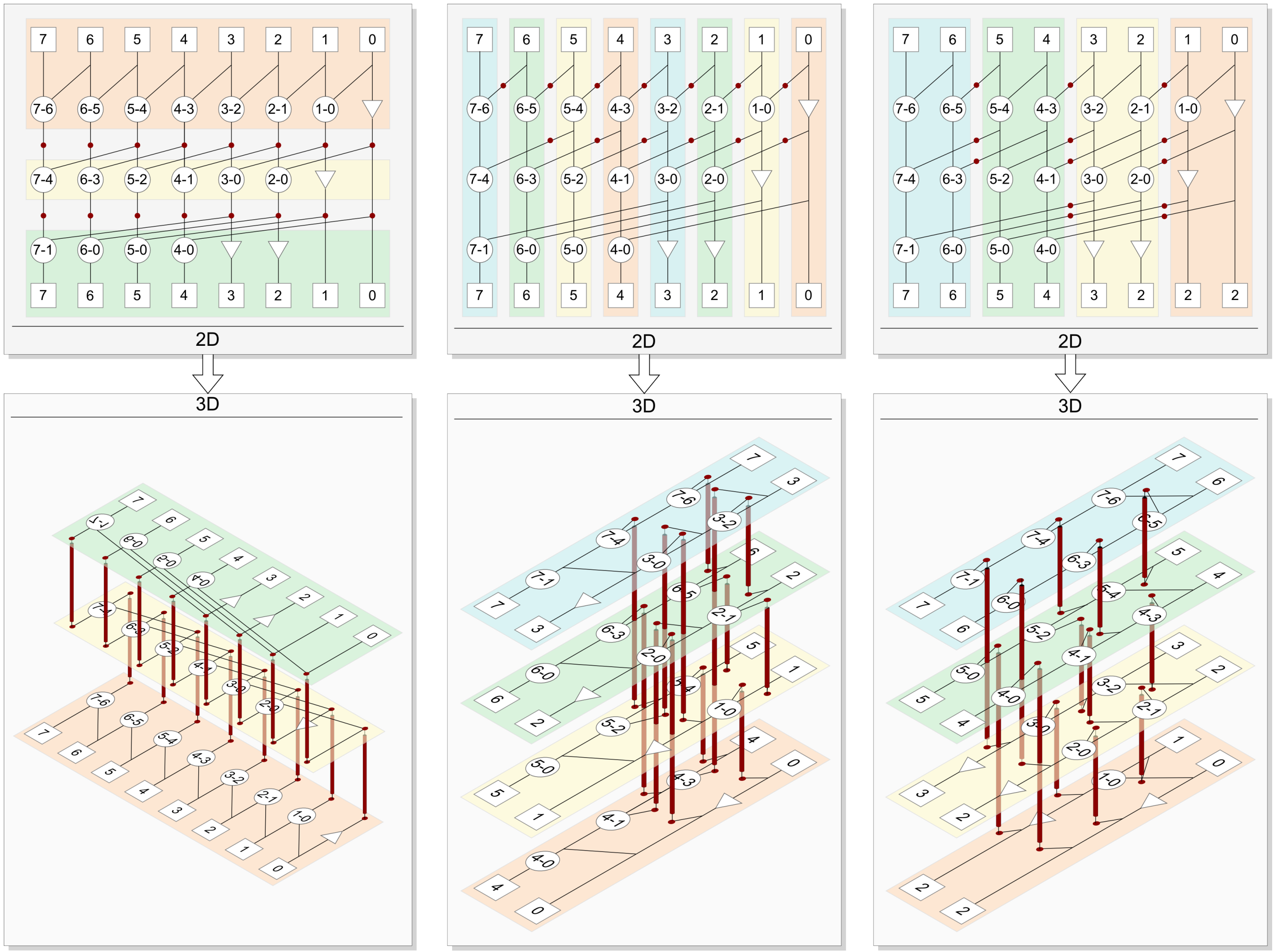
The partitioning should strive to reduce the long interconnects in the carry network, therefore clusters of communicating carry-merge cells should be placed on the same die. Based on this observation, other partitioning strategies in which random carry-merge cells are placed on each tier will most likely not produce better results.

For wide-operand adders, the number of TSVs has a direct influence on the overall performance of the adder. The reasons for this are twofold: i) the area occupied by a TSV makes the interconnect wires to increase in length, and, ii) the large parasitic capacitance between the TSV and the silicon substrate necessitates the placement of a high strength driving buffer before it, with large area and propagation delay. For comparison, a carry-merge operator cell synthesized in a commercial 65 nm low power CMOS technology takes $4.68\,\mu\text{m}^2$, while the minimum predictions for TSV diameter and pitch are $0.8\,\mu\text{m}$ and $1.6\,\mu\text{m}$ [11], respectively.

Each of the TSVs drawn in the figure stand for a (generate,propagate) pair. The Stage Folding strategy also needs to transmit the initial generate signal computed for each bit from the bottom tier to the top one, to compute the final sum bit. The number of TSVs in the k-tier stack adders from Figure 3 is given by the following equations:

$$N_{TSV_a}(N,K) = 3N(K-1),$$

$$N_{TSV_b}(N,K) = N_{TSV_c}(N,K) = 2\left(N\log_2 K - \sum_{i=0}^{\log_2(K-1)} 2^i\right).$$



(a) Stage Folding. **(b) Bit Interleaving (3D BI).** **(c) Bit-Slice Folding (3D BS).**

Figure 3: 3D Kogge-Stone Adders Partitioning Strategies.

In case of K-tier Stage Folding, the number of TSVs is constant between every two adjacent layers, equal with N, the size of the adder operands. The number of TSVs is the highest in this case, but they only connect neighboring tiers. The Bit Interleaving and Bit-Slice Folding techniques require a significantly lower number of TSVs, but some of them run through several tiers, thus increasing congestion on those tiers.

The delay of the adder for all three partitioning strategies from Figure 3 is given by the following equations:

$$D_a(N,K) = D_{CM} \cdot log_2 N + (K-1) \cdot D_{TSV_a} + D_{wire_a},$$
$$D_b(N,K) = D_{CM} \cdot log_2 N + log_2 K \cdot D_{TSV_b} + D_{wire_b},$$
$$D_c(N,K) = D_{CM} \cdot log_2 N + log_2 K \cdot D_{TSV_c} + D_{wire_c},$$

where D_{CM} is the delay of one carry-merge computational cell, D_{TSV_x} is the delay introduced by the TSV driving buffer, and D_{wire_x} the delay on the interconnect wires. It can be seen that for Bit Interleaving and Bit-Slice Folding some of the TSVs on the critical path traverse multiple tiers, and the longest TSVs on the critical path have to cross the entire k-tier stack. Since longer TSVs have proportionally larger parasitic capacitance (and hence delay), they either demand for the utilization of larger driving buffers, or as an alternative they have to be split in shorter TSVs. Thus, if we note with D_{TSV} the delay introduced by a buffer driving a short TSV between two adjacent tiers the delay equations become:

$$D_a(N,K) = D_{CM} \cdot log_2 N + (K-1) \cdot D_{TSV} + D_{wire_a},$$
$$D_b(N,K) = D_{CM} \cdot log_2 N + (K-1) \cdot D_{TSV} + D_{wire_b},$$
$$D_c(N,K) = D_{CM} \cdot log_2 N + (K-1) \cdot D_{TSV} + D_{wire_c}.$$

As can easily be deduced from the above equations, the partitioning strategy does not have a direct influence on the propagation delay, but rather an indirect influence by affecting the interconnect delay on the wiring. The effect of this influence is even more important in case of wide operand adders, with large area and long wiring. A theoretical comparison of the actual delay of the three partitioning scenarios is difficult to make owing to the difficulty in estimating the interconnect wire delay, D_{wire_x}, which is dependent on the physical layout of every tier. Nevertheless, some conclusion can be made by analysing the TSV distribution in the stack and the area of the most congested tier, which determines the footprint of the entire 3D stack.

The Stage Folding strategy (D_a) has the largest footprint, given by the tier including the first computation stage, with at least N carry-merge cells. The Bit Interleaving strategy (D_b) breaks connections across dies in the early stages of computation, thus the TSVs are more clustered in that region of the layout. The end of every tier has a structure similar with a Kogge-Stone N/K-bit adder, with the cells in the last stages of computations placed far apart. In contrast, the Bit-Slice Folding strategy (D_c) recursively breaks inter-cell connections starting with the long connections in the last stages of computations, thus the TSVs are spread over the entire layout. The way in which the folding happens is also more advantageous since in each tier only the carry-merge cells in the first stages of computations are interconnected, where the wires are shorter. The footprint of the Bit-Slice Folding strategy, with the upper tier having a full matrix of $N/K \cdot \log_2 N$ carry-merge cells is larger than the one of Bit Interleaving strategy.

Bit Interleaving and Bit-slice Folding are essentially the two extreme cases of bit-splitting the computation tree. Hence, any other particular instances of bit-splitting results in a footprint and area characteristics in between these two extreme cases.

When compared with the delay of a planar implementation of a wide-operand prefix adder, the presented 3D folding techniques provide a length reduction of critical wires, since a large area is now distributed over many tiers. On the other hand, the addition of TSVs increases the occupied area and the routing congestion, and in the same time adds a delay penalty on the driving buffers. The three partitioning strategies presented can be applied to any prefix adder. However, since the main benefit of 3D partitioning of wide operand width adders resides in wirelength reduction, the Kogge-Stone adder architecture, having the highest number of wire tracks [12], is expected to gain the most from using the 3D technology.

IV. 3D Stacked Hybrid Adder

Besides the widely used metrics of delay and footprint, the cost metric is another one of high importance. In 3D-stacking technology the manufacturing cost equation has an additional parameter with a heavy weight, namely whether the tiers are identical or not. It is a well-known fact that the price of lithography masks substantially contributes to the manufacturing cost of integrated circuits, especially in the deep sub-micron technologies [13]. Therefore, if the tiers are not the same, the manufacturing cost is almost multiplied with the number of tiers.

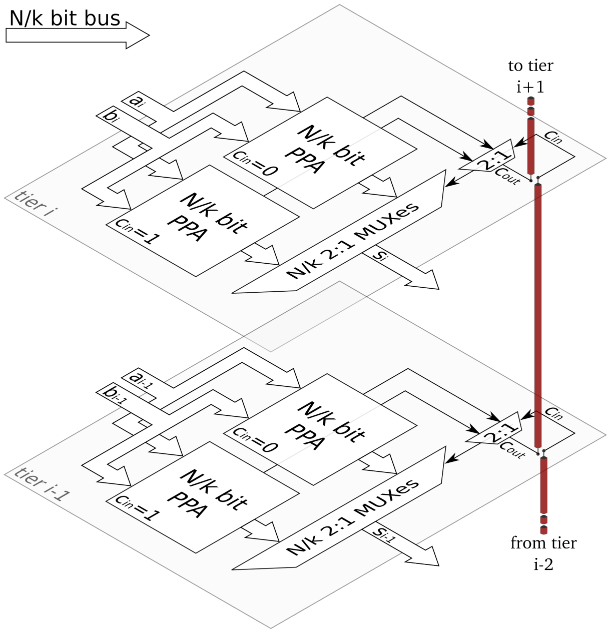


Figure 4: 3D Stacked Hybrid Prefix/Carry-Select Adder.

All partitioning scenarios presented in the previous section suffer from the same drawback: they induce a heterogeneous stack structure, as they require a different design in each tier. Silicon tiers with identical layout can be designed if we place unused carry-merge cells instead of the feed-through buffers (the triangles in Figure 1) for the lower bits of the prefix graph, and we add configuration logic to select the proper functionality on each tier. The additional overhead can be tolerated for small sized adders, but in the case of large operand widths, as the ones targeted by our investigation, the area lost and the delay penalty can become prohibitively high.

Thus, in order to alleviate this shortcoming, we propose a novel 3D Stacked Hybrid Prefix/Carry-select Adder with identical tier structure, which potentially makes manufacturing of hardware wide-operand adders a reality. An N-bit adder can be implemented on a K identical tier stacked IC with the structure depicted in Figure 4. Each tier contains two N/K-bit fast parallel prefix adders (PPA) and $N/K+1$ 2:1 multiplexers. The selection between the two sum outputs in tier i, with $i = 2, ..., N/K$ is given by the carry-out signal from tier $i-1$, transmitted through one TSV.

The critical path crosses one of the adders on the first tier and the multiplexing chain at the outputs of each tier, resulting in an asymptotic delay in the order of $O(\log_2 \frac{N}{K} + K)$, which can be further reduced to $O(\log_2 N)$ at the expense of some hardware overhead if the multiplexing chain is designed by following the look-ahead principle [4]. The footprint of the 3D stacked IC structure is given by the area of the two N/K-bit prefix adders, $N/K + 1$ 2:1 multiplexers, and a TSV, which is substantially smaller than the one of an N-bit prefix adder.

To optimize the propagation delay, the 2:1 multiplexers that select between the two sums can be implemented with transmission-gates. Transmission-gates need also inverted select signals, but having an inverter on each tier would increase the delay. Thus, it is more advantageous to also compute the negated carry out signal and pass it to the next tier. In such a case, the number of TSVs between every two neighboring tiers grows to two, one for the carry out and one for the inverted carry out.

The computation delay of an N-bit hybrid adder with K tiers can be expressed as follows:

$$D_H(N,K) = D_{PP}(\frac{N}{K}) + K \cdot D_{MUX} + (K-1)D_{TSV},$$

where $D_{PP}(x)$ is the delay of an x-bit parallel prefix adder, and D_{MUX} the delay of a 2:1 multiplexer. Increasing the number of tiers reduces the operand width of the parallel prefix adders, and hence its delay, but increases the delay contribution of the multiplexers and inter-tier TSVs. This trade-off can be optimized in terms of delay by finding the K value for which the differential of the delay function is equal with zero:

$$\frac{\Delta D_H(N,K)}{\Delta K} = 0$$

$$\frac{\Delta D_{PP}(\frac{N}{K})}{\Delta K} + D_{MUX} + D_{TSV} = 0.$$

If we do not take into account the wire interconnects and consider the prefix adder as being Kogge-Stone with the delay equal with $D_{KS}(\frac{N}{K}) = D_{CM} \cdot \log_2 \frac{N}{K}$, D_{CM} - delay of a carry-merge cell, the optimal number of tiers that results in the lowest possible delay of the hybrid adder is:

$$K_{opt} = \frac{D_{CM}}{(D_{MUX} + D_{TSV})\ln 2} = \frac{1.44 D_{CM}}{D_{MUX} + D_{TSV}}. \quad (1)$$

This means that for given silicon and TSV technologies there is always an optimal number of tiers for which the 3D Hybrid Adder has the lowest delay. The delay of the 3D Hybrid Adder is comparable with the one of 3D direct folded adders from Section III, the difference between them being dependent on the number of tiers K, and the adder size N.

If we consider the footprint metric, for the 3D Hybrid Adder we have to use two instances of N/K-bit prefix adders, while 3D direct folded adders rely on a structure larger than an N/K-bit prefix adder. However, depending on the TSV dimensions, the 3D Hybrid Adder footprint might be lower than the one of an N-bit 3D folded adder. Regardless of this, the 3D Hybrid Adder is more advantageous to manufacture due to the reduced number of TSVs and the easy alignment during bonding of identical dies. Thus, disregarding the TSV area overhead, the 3D Hybrid Adder has a larger footprint than the 3D folded one, when both use the same architecture for the prefix adders. However, from the manufacturing cost perspective, the 3D Hybrid Adder has a cost approximately K times lower than a 3D folded adder, since all tiers are identical, and only one set of lithography masks is needed.

V. Case Study

In this section we present a study of the following wide-operand width adders: 2D prefix, 3D folded prefix (from Figure 3), and 3D hybrid prefix/carry-select (from Figure 4), with respect to the metrics presented at the beginning of Section IV, and combinations of them, i.e., delay, footprint, delay-footprint product, and delay-footprint-cost product. Moreover, we identify the trends for different operand widths and number of available stacked tiers, which create various design trade-off opportunities.

Even though the theoretical advantages of the novel 3D Hybrid Adder are obvious, the evaluation of the actual impact of such a 3D-Stacked implementation on the overall delay, e.g., TSV delay, wire delay, crosstalk delay, and on the footprint is not straightforward. The validation of the 3D designs containing TSVs require additional implementation steps when compared with the normal timing closure sign-off flow, i.e., design partitioning per tier, TSVs insertion, and design aligning. Taking into consideration that for the proposed 3D Hybrid family each tier is identical, we note that in this case the design effort is reduced to designing one tier only, while for the other 3D folded prefix adders this does not hold true. Furthermore, in all 3D design cases, we expect to achieve latency and cost benefits due to wire-length reduction.

A. Implementation Methodology

From the three direct 3D folding strategies presented in Section III, we do not consider Stage Folding adders since all the input bits and outputs bits are on one tier, the lowest and the uppermost, respectively. For wide operand widths this becomes a considerable hindrance in a real-life design, since the floorplan will be pin congested, and most likely impractical. The remaining three 3D adder families considered by this case study, i.e., Bit Interleaving adders, Bit-Slice Folding adders, and 3D Hybrid adders, have the input and output bits equally distributed on all tiers, with input bits having the same weight being on the same tier.

For our study, we implemented in structural parameterized VHDL one tier of the 3D Hybrid adder, and the most congested tier of a 3D Bit Interleaving adder and a 3D Bit-Slice Folded adder. The hardware description was synthesized using Cadence RTL Compiler [14] for all considered combinations of operand bit-widths and number of tiers. For synthesis we use a commercial 65 nm technology with a wide variety of standard cells, including optimized complex gates, e.g., full and half adders. Moreover, we take advantage of the RTL Compiler hierarchical design option in order to ensure that the prefix tree architectures are maintained throughout technology mapping. Furthermore, we continue our implementation using Cadence Encounter Digital Implementation (EDI) flow [14] as follows. We perform floorplaning for each design, also taking into consideration the associated TSV footprint. We model the TSV area to be 5 μm^2 including keep-out zone [11], [15]. We continue with place and route using the bottom four routing metals, and we finalize each layout with a sign-off analysis, obtaining the propagation delay and footprint for worst case timing conditions, i.e., 1.08 V supply voltage, $-40\,^{\circ}$C temperature, and slow device models.

For each adder we add to the obtained propagation delay the delay of the TSV driving buffer, equal with the delay of a buffer with strength 8X, according with [15]. We mention that in the case of the 3D Hybrid adder the TSV driving buffer can be embedded in the useful computation performed by the multiplexer present before the TSV, by resizing its gates. The use of the same approach for 3D folded adders requires a higher design effort, since the tiers are not of the same length,

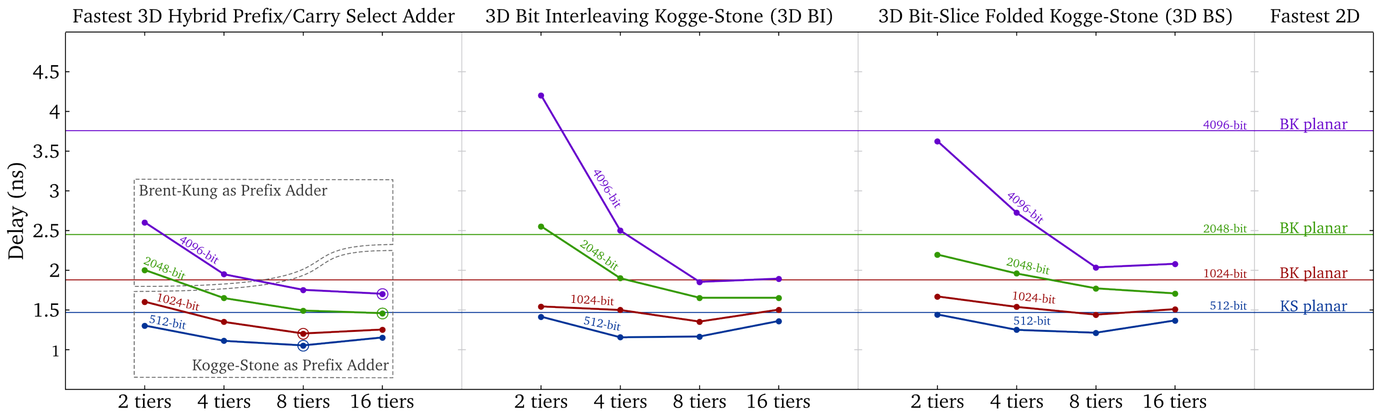


Figure 5: Delay Comparison.

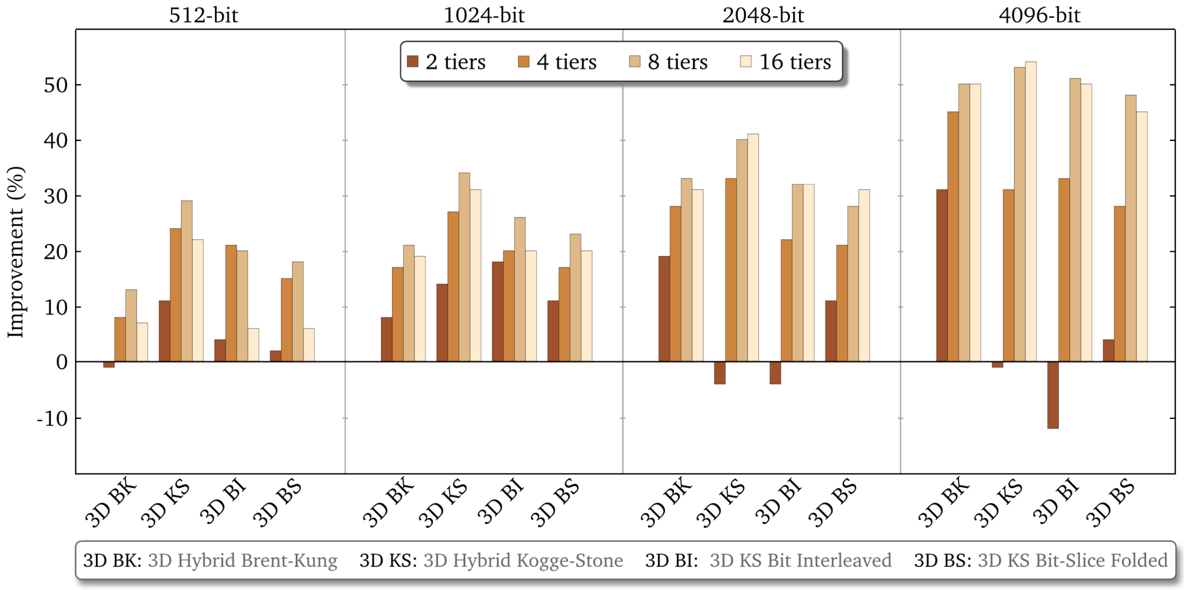


Figure 6: Delay Comparison Relative to the Fastest Planar Prefix Adder.

and every TSV is driven by separate carry merge cells, with logic gates with different driving strengths.

B. Experimental Results

An absolute delay comparison of the adder families is presented in Figure 5 for operand widths from 512-bits to 4096-bits, stacked on 2, 4, 8, and 16 tiers. In addition, the horizontal lines represent the fastest planar Kogge-Stone (KS) or Brent-Kung (BK) adder for the same range of operand widths. We also selected the fastest version between Kogge-Stone and Brent-Kung as the local prefix adder in the 3D Hybrid adder in all cases (on the left side). The Bit Interleaved (3D BI) and Bit-Slice Folded (3D BS) 3D adders are depicted in the middle and right part of the figure, respectively. As previously explained at the end of Section II, we remind that the 3D folded adders considered throughout this section, i.e., 3D BI and 3D BS, are both Kogge-Stone adders.

The 3D Hybrid plots confirm the trade-off between the numbers of tiers and the delay, hence the existence of the optimum number of tiers with respect to delay, defined by Equation (1). The same trade-off is also present for 3D folded prefix adders. In all implementations the optimum number of tiers with respect to delay increase when the operand width is doubled. This trend is more apparent in Figure 6, presenting the delay improvement in percentages (negative values indicate delay degradations) over the fastest planar prefix adder for a fixed operand width. For large operand widths, i.e., 2048-bit and 4096-bit, even though the delay is reduced by 3D partitioning for the Bit Interleaving strategy (3D BI) and the Bit-Slice Folding one (3D BS), the design on each tier is still too large and with a wire-dominated delay, giving a too high total delay. By looking at Figure 1, it can be observed that in order for any 3D wide-adder to have a viable delay it needs to process on each tier no more than 512-bits. Thus, for large operand widths (4096-bit) and a sufficient number of tiers (at least 8), all of the considered adders offers comparable delay improvements. As expected, the speed

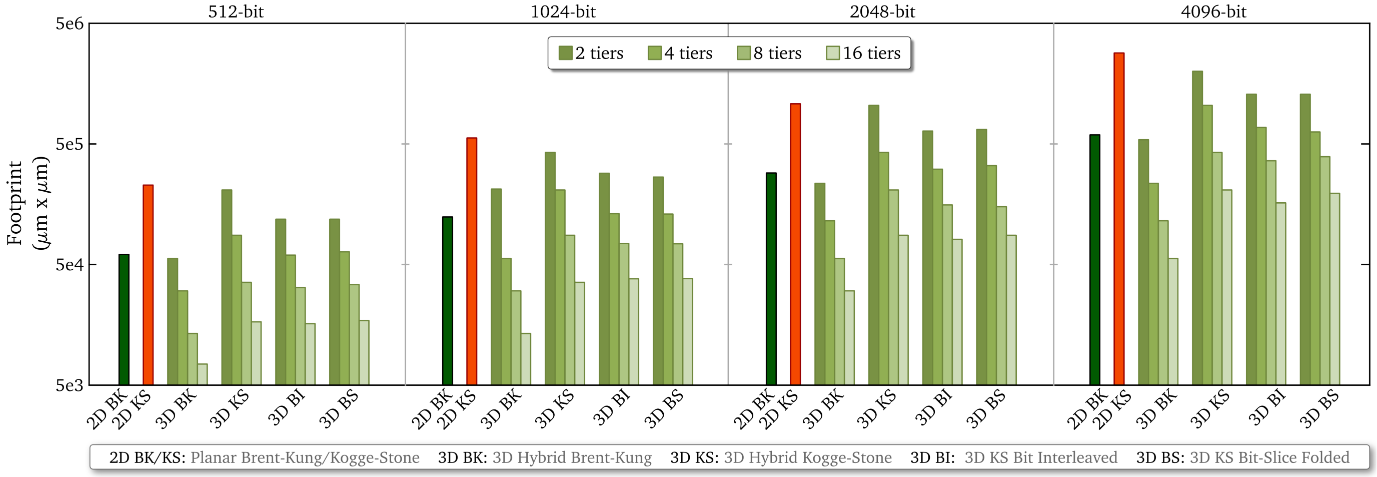


Figure 7: Footprint Comparison.

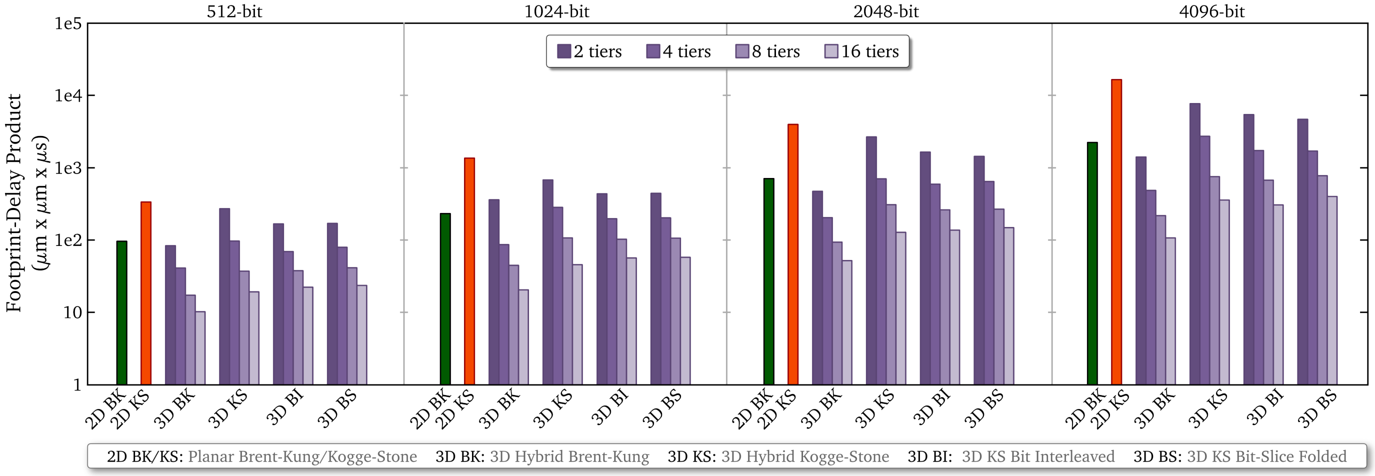


Figure 8: Footprint-Delay Product Comparison.

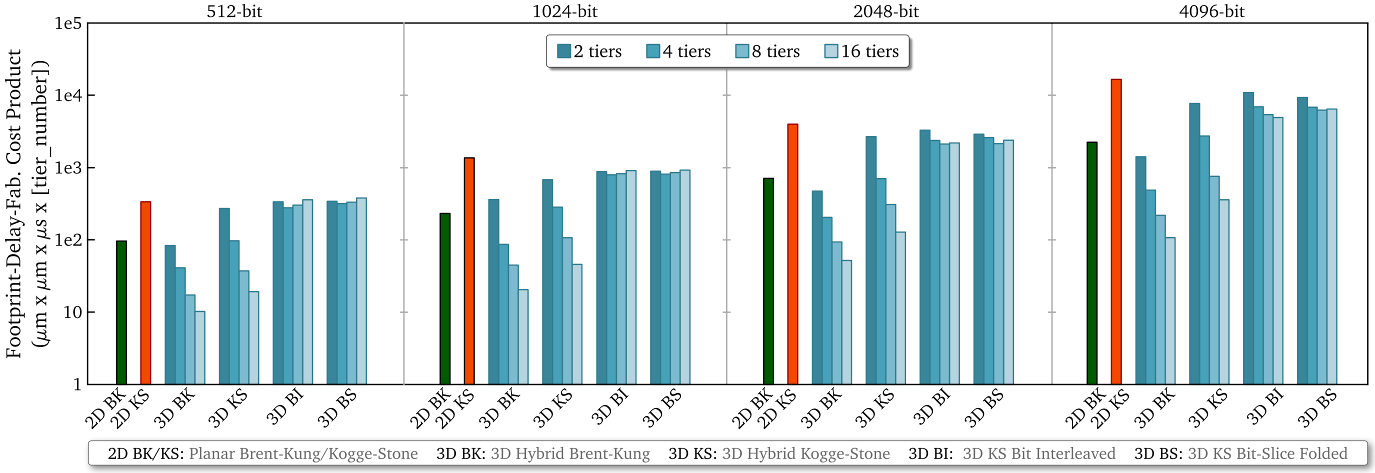


Figure 9: Footprint-Delay-Cost Product Comparison.

improvement for 3D Hybrid adder with Brent-Kung (3D BK) is less than 3D Kogge-Stone (3D KS) adders, since the Brent-Kung prefix tree has less wiring tracks. The figure clearly indicates that the 3D Hybrid KS adders always outperform the other solutions in terms of delay improvement for all the considered operand widths.

Figure 7 presents the footprint for the considered 2D and 3D addition units, on a logarithmic scale. Overall, by far, the 3D BK hybrid adder represents the best choice in all cases, i.e., number of tiers and operand width, when footprint is the metric of interest. However, in all cases the 3D KS hybrid adder has a footprint value slightly larger than, but in the same order of magnitude as the one of the corresponding 3D BI and 3D BS equivalent counterparts. Finally, we can observe that the 3D KS, 3D BI, and 3D BS footprints are lower than the "classic" 2D BK one only for stacking on 8 and 16 tiers, for all the considered operand widths.

Figure 8 presents the 3D equivalent of the widely used area-delay product metric, i.e., the footprint-delay product. We can observe that the tendencies we identified in Figure 7 still hold true. Thus even though the 3D BK hybrid adder is almost always the slower design for all considered operand widths (see Figure 6) it is the most effective one, as its gets the best out of its resources.

Even though for 2D chips the area-delay metric accurately captures the effectiveness of a design, when multiple tiers are stacked using a wafer-2-wafer stacking [3], the mask-set fabrication cost dominates the manufacturing costs when a large number of different tiers are stacked. Thus, we plot in Figure 9 the footprint-delay-cost product, where *cost* represents the number of tiers with different implementations (equivalent with the number of different wafers). As explained in Section IV, the 3D BK and the 3D KS Hybrid designs are not affected by this metric due to the fact that all tiers are identical.

3D BI and 3D BS implementation both suffer significant degradation due to the *cost* metric. Practically, the gain in delay and footprint is almost canceled for all the 3D BI and 3D BS implementations, when compared with their 2D counterparts. Moreover, the number of tiers becomes now an irrelevant factor in their design space as all 3D folded adders exhibit an almost equal delay-footprint-cost product value. We observe that the best choice in terms of footprint-delay-cost metric are the novel 3D BS and 3D BK implementations presented in Section IV. With the increase of the number of tiers they even become more attractive, due to the fact that the footprint-delay-cost parameter linearly decreases when the number of tiers is linearly increased.

VI. Conclusions

In this paper we investigated the implications of using 3D Stacking IC technology in designing efficient wide-operand adders, to be potentially included in cryptographic coprocessors. We classified direct folding strategies applicable to 3D fast adder designs, and analyzed the cost and performance for each of them. Since our study indicated that direct folding suffers from a major drawback related to a large manufacturing cost overhead due to the fact that the tiers are not identical, we addressed this issue by proposing and evaluating a 3D Hybrid Prefix/Carry-Select Adder, with identical layout on every tier, and a reduced number of TSVs. We performed a 3D wide-operand adders design space exploration with regard to delay, footprint, and cost metrics and analyzed various folded and hybrid 65nm CMOS 3D designs. The simulation results indicated that the 3D hybrid adder is faster than adders constructed based on direct 3D folding strategies, providing a delay improvement of up to 29% for 512-bit adders and up to 54% for 4096-bit adders. In terms of footprint, the folding approach requires less chip real estate, but with a very high manufacturing cost, owing to the fact of having different layouts on every tier. Based on the footprint-delay-cost product, which is more appropriate to capture the complexity of a 3D implementation, we concluded that the hybrid prefix/carry-select approach is more suitable for 3D stacked integration, achieving a reduction of the footprint-delay-cost product over 3D folded adders between 17.97% and 94.05%.

References

[1] J. Katz and Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Boca Raton: CRC PRESS, 2007.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[3] P. Garrou, *Handbook of 3D integration: Technology and Applications of 3D integrated circuits*. Weinheim: Wiley-VCH, 2008.

[4] B. Parhami, *Computer arithmetic*. New York: Oxford University Press, 2009.

[5] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 786–793, 1973.

[6] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Transactions on Computers*, vol. 100, no. 3, pp. 260–264, 1982.

[7] Z. Huang and M. D. Ercegovac, "Effect of wire delay on the design of prefix adders in deep-submicron technology," in *Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, vol. 2, 2000, pp. 1713–1717.

[8] B. Vaidyanathan, W. L Hung, F. Wang, Y. Xie, V. Narayanan, and M. J Irwin, "Architecting microprocessor components in 3D design space," in *International Conference on VLSI Design*, 2007, pp. 103–108.

[9] K. Puttaswamy and G. H. Loh, "The impact of 3-dimensional integration on the design of arithmetic units," in *IEEE International Symposium on Circuits and Systems*, 2006.

[10] J. Ouyang, G. Sun, Y. Chen, L. Duan, T. Zhang, Y. Xie., and M. J Irwin, "Arithmetic unit design using 180nm TSV-based 3D stacking technology," in *IEEE International Conference on 3D System Integration*, San Francisco, CA, 2009, pp. 1–4.

[11] "ITRS - Interconnect," International Technology Roadmap for Semiconductors, Tech. Rep., 2011.

[12] D. Harris, "A taxonomy of parallel prefix networks," in *Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2004*, Nov. 2003, pp. 2213–2217.

[13] R. F. Pease and S. Y. Chou, "Lithography and other patterning techniques for future electronics," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 248–270, 2008.

[14] "Cadence Design Systems," 2013. [Online]. Available: http://www.cadence.com/us/pages/default.aspx

[15] C. L. Yu, C. H. Chang, H. Y. Wang, J. H. Chang, L. H. Huang, C. W. Kuo, S. P. Tai, S. Y. Hou, W. L. Lin, E. B. Liao, and others, "TSV process optimization for reduced device impact on 28nm CMOS," in *Proc. Symp. VLSI Technol. Dig. Tech. Papers*, vol. 8, 2011, p. 1.