3DFAR: A Three-Dimensional Fabric for Reliable Multi-Core Processors

Javad Bagherzadeh, and Valeria Bertacco
University of Michigan, {javadb, valeria}@umich.edu

***Abstract*—In the past decade, silicon technology trends into the nanometer regime have led to significantly higher transistor failure rates. Moreover, these trends are expected to exacerbate with future devices. To enhance reliability,several approaches leverage the inherent core-level and processor-level redundancy present in large chip multiprocessors. However, all of these methods incur high overheads, making them impractical.**

In this paper, we propose 3DFAR, a novel architecture leveraging 3-dimensional fabrics layouts to efficiently enhance reliability in the presence of faults. Our key idea is based on a fine-grained reconfigurable pipeline for multicore processors, which minimizes routing delay among spare units of the same type by using physical layout locality and efficient interconnect switches, distributed over multiple vertical layers. Our evaluation shows that 3DFAR outperforms state-of-the-art reliable 2D solutions, at a minimal area cost of only 7% over an unprotected design.

I. Introduction

Over the past five decades, continued scaling of silicon fabrication technology has led to an exponential increase in transistor budgets, leading to drastic performance improvements. However, deep sub-micron technology also poses unique challenges: processors in aggressively scaled technologies are more susceptible to permanent transistor failures at runtime, often due to wearout phenomena. Moreover, beyond 2021 it will no longer be economically viable for companies to continue to shrink transistors' dimensions. Instead, chip manufacturers will be forced to turn to other solutions to boost performance, possibly novel device technologies that are likely to suffer from even more disruptive reliability issues. Thus, unless reliability concerns are addressed by effective design solutions, manufacturing yields and silicon chip lifetime expectancy will soon be drastically compromised, while future device technologies may be nonviable from the start. Today, several solutions address reliability and fault tolerance in processors; they can be grouped into software management or hardware level techniques. The former usually present high latency in fault detection and slow performance in recovery [1]. On the other hand, hardware approaches are often based on providing spare units or using inherent redundancy within a time domain or a sharing infrastructure. However, approaches that provide spare components, such as N-modular redundancy methods [2], StageNet [3], BulletProof [4], *etc*., are associated with hardware and power overheads, which can be quite considerable in the case of a many-core processor.

Our solution, called a 3-Dimensional FAbric for Reliable multicores (3DFAR), proposes to use monolithic 3D fabrics to stack corresponding hardware units from distinct cores above each other, and leverages inter-core redundancy to provide a reliable architecture. We place equivalent resources within short vertical distance from each other, and provide low overhead and fast communication infrastructure using Monolithic Interlayer Vias (MIVs) [5]. Compared to other existing 3D integration technologies, such as wirebonding, interposers, TSVs, *etc*., monolithic 3D technology is the only solution that can enable ultra fine-grained vertical integration of devices and interconnects, thanks to the extremely small size of inter-tier vias (typically 50nm in diameter).

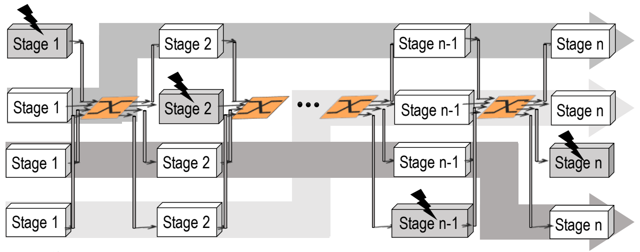


Fig. 1: **Schematic of a 3DFAR multi-core architecture**: corresponding pipeline stages are stacked vertically, while specialized crossbar units are inserted between each pair of stages. With 4-faults shown, a 2D CMP would be disabled, while 3DFAR reconfigures to connect healthy units as shown with the wideband lines, providing the computing power of 3 cores.

In our architecture, illustrated in Figure 1, we replace the direct connections at each pipeline stage boundary by a crossbar switch, so that each stage may connect to subsequent stages from other layers. By adaptively routing around failed stages, we can salvage working units and performance effectively. In developing our solution, we investigated multiple of interconnect switch structures, evaluated our solution on the physical design of a 4-core in-order processor and compared it against several state-of-the-art 2D reliable architectures. We found that 3DFAR provides consistent performance improvements over these solutions, at an area cost of only 7% for interconnects and MIVs. In summary, we make the following contributions:

- A novel sparing-based, reliable solution for multicore processors, specialized for 3D fabrics. Our solution entails only a 4% performance impact over an unprotected 2D design.
- A new method to connect corresponding hardware resources on a vertical layout, which does not require any buffering or complex routing. Through our method, we can dynamically create and adapt pipelines of healthy resources.
- An analysis of the proposed interconnect solutions and their performance when varying the number of 3D design layers.

II. Proposed Architecture

3DFAR is based on a fine-grained pipeline design for multicore processors, which can be dynamically configured to route instructions only through functioning hardware components. Instead of pushing instructions through paths fixed at design time, 3DFAR relies on inter-stage crossbar switches to form execution pipelines dynamically, enabling graceful performance degradation when facing increasing transistor failures. Specifically, we replace the direct connections at each pipeline stage boundary with **interconnect switches** to create a network of resources, so that each pipeline stage is connected to all instances of the subsequent stage. In our architecture, illustrated in Figure 1, each stage may connect to subsequent stages from other layers. By adaptively routing around failed stages we can salvage working units and performance effectively. When a fault occurs, the victim unit (that is, pipeline stage) is isolated, and an identical unit from another core, laid out on another layer of the 3D fabric, is used to advance the execution. As a result, the pipeline executing

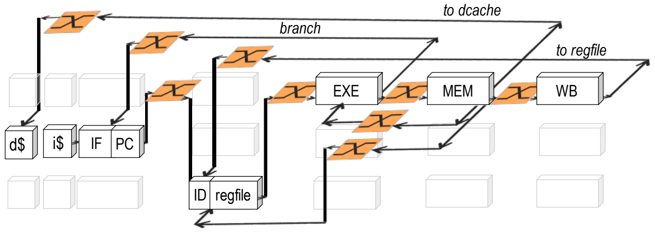


Fig. 2: **Interconnect switches deployed in a 5-stage in-order pipeline.** Due to physical locality and short propagation delays through the switches, backward and forward execution paths remain unbuffered and unmodified, making the 3DFAR solution viable over a wide range of architectures.

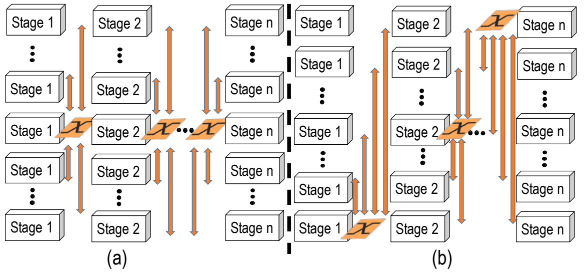


Fig. 3: **Middle-layer and vertically distributed interconnect.** a) The area overhead is concentrated in the middle layers for the middle-layer solution, while b) the vertically distributed design distributes switches on every layer.

the application may comprise elements from several vertical layers, connected together to form a logical processor core.

Fault detection. Our proposed solution targets the repair of a faulty system, so that it remains available, possibly providing lower performance, even in presence of faults. A number of works in recent years have presented efficient mechanisms to perform fault detection, even at fine granularity, *e.g.*, [4,6]. We assume a framework similar to those, where the occurrence of a fault is detected via hardware or firmware mechanisms and localized to a single pipeline stage.

Latency of interconnect switches. 3DFAR cross-layer switches do not require any buffering, thus simplifying their design and control requirements. Moreover, it is possible to use switches to connect pipeline stages both on forward and backward paths. Prior solutions, *e.g.*, [3], suffer from performance and complexity impacts introduced by buffered switches. However, since propagation delays on vertical MIVs are minimal (about 100x times faster than in 2D layouts) due to much shorter distances, we can avoid buffering by accommodating a small increase in clock cycle length (<5%). Because of this low interconnect latency, it is straightforward to deploy 3DFAR in a wide range of stacked processor architectures: Figure 2 provides an example of crossbar switches inserted in a 5-stage in-order pipeline with data forwarding paths: 5 switches are introduced to advance computation between pipeline stages, 3 are used for data and control forwarding, and one last switch connects the memory stage to the integrated local cache.

Number of design layers. There are contrasting goals in determining the ideal number of layers in a 3DFAR design: on one hand the more the layers the more spare units are available, and thus the stronger the robustness of the solution. On the other hand, too many layers may be impractical and may negatively affect the latency required for traversing the vertical dimension of the design to reach a spare unit, as well as the size of crossbar switch. Thus, a large many core system would need to be logically partitioned into smaller 3DFAR islands. To evaluate the viable number of cores in a island, we conducted experiments over a range of island sizes.

Thermal issues. Heat dissipation is a key concern in monolithic 3D-ICs. A few recent solutions discussed and addressed this problem, proposing advanced cooling technologies, within the layers and around the peripheries: incorporating materials with high thermal conductivity, such as graphene, to aid in heat removal [7] or adding advanced convective structures, such as a metallic nanomesh [8].

III. Interconnect switch design

In designing our interconnect switches, we took into account the number of vertical MIVs required for each interconnect, the propagation delay entailed, which in turn affects the nominal operating frequency of the system, and the silicon area overhead for each silicon layer. Note that the overall area overhead is the one imposed by the largest silicon layer.

The **number of vertical MIVs** necessary to connect all vertical pipeline stages depends on the specific architecture of the system. For the microarchitecture used in our examples and depicted in Figure 2, a total of 1,106 signals must be connected to and from other layers. Each pipeline stage uses a varying number of input signals, ranging from 68 for the connection between write-back stage and the register file, to 336 signals connecting decode to execute. The **propagation delay** between connections depends on how many vertical layers a signal must cross to go from a source pipeline stage through the crossbar and then to the destination stage. Finally, the **silicon area overhead** was estimated based on the size of individual MIVs, as reported in [9] (0.5μm TSV) and the area of the crossbar switch. In light of these factors, we developed three design solutions, described in the sections below. Finally, an important factor associated with using MIVs is their **reliability**, as the failure of a single MIV may cause unpredictable effects that could lead to system failure. Yield and reliability improvements are usually achieved through a range of redundancy techniques and sparing, investigated in recent years, along with several diagnosis and repair mechanisms [10].

Middle-layer interconnect. This solution, illustrated on the left side of Figure 3, minimizes and equalizes the latency introduced by the interconnect switches. By placing all the switches in the middle layer, all signals travel no more than 2×traverse_delay(#layers/2) to go from one layer, through the switch and to the destination layer. When the number of layers is even, switches can be placed on the two middle layers in an alternating fashion, without affecting the overall latency impact. Note that the middle layer must accommodate all MIVs incoming from other layers of the design. However, they can be aligned so that the same surface can be used for MIVs coming from above and from below. Thus, with this solution, the middle layer requires space for 1,106·$n/2$ MIVs, where n is the number of layers. In addition, we estimated the area of the crossbars to be approximately half the area required by MIVs (thus, 1,106/2= 553 MIV-equivalent area units). Thus, in first approximation, this solution requires an area overhead equivalent of 1,659·$n/2$ MIV-equivalent area units. If the number of layers is even, crossbar switches can be partitioned over two layers, and the area cost is reduced to 1,106 + 553/2 = 1382·$n/2$ MIV-equivalent area units.

Vertically distributed interconnect. To distribute the area overhead over all the vertical layers of the design, we explored a solution where each interconnect switch is placed on a different layer, on a rotating fashion. This approach minimizes area imbalance at the granularity of one switch. Note, however, that the number of signals connecting two stages of a pipeline varies for each stage; thus each switch entails a different area

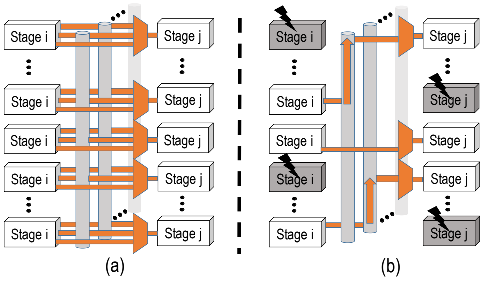


Fig. 4: **The vertical bus interconnect** a) uses multiplexers to select which input to route to a pipeline stage. b) Example of a possible fault scenario.

overhead (see the right side of Figure 3). The interconnects located at the bottom and top layers experience the longest wire delays, that is, the time to traverse all vertical layers, so they are best placed at the input of pipeline stages which can afford more timing slack. The area overhead is highly dependent on the specific switch locations: for each layer, it includes the area of the local switch(es), that of the MIVs passing through, and the areas of the MIVs incoming to the local switch, from above or below (which can be overlapped). In general, middle layers host the most MIVs, approximately 1,106·$n/2$, (n being the number of layers) and likely one or a few switches. The area overhead is smaller than in the prior solution, as the crossbars are distributed over multiple layers.

Vertical bus interconnect. This solution leverages a bus-style architecture, where vertical links run across the entire height of the design, and each layer uses a set of multiplexers to select its inputs among one of the vertical bus lines or the prior stage on the same layer, as shown in Figure 4.a). The advantage of this solution is that only unidirectional MIVs are required, since signals are switched directly at their destination layer. Note that the propagation delay for this solution depends on the location of the faulty unit: signals must simply propagate from a faulty layer to that of the spare unit. In the worst case, this is the delay of crossing all the layers, as for the vertically-distributed interconnect switch. The area overhead is uniformly distributed among all layers, indeed each layer is simply augmented with a set of multiplexers, and the vertical signals to route are half than in previous solutions. We estimated the area of the selector multiplexers to be approximately 1/4 that of an MIV, for each signal. Thus, with a vertical bus structure, each layer must accommodate (553+138)·b = 691·b MIV-equivalent area units, where b is the number of vertical buses between each stage.

Note that we do not need to route as many vertical bus lines as the number of layers in the design. In fact, as more units in a stage become faulty, the need to transfer data to healthy units in the subsequent stage decreases as well: only one bus line is needed after stage i, when only one unit, or all but one units, is faulty. In contrast, when half of the stages i and half of the $i+i$s are faulty, we may need as many as $\lfloor$#layers/2$\rfloor$ vertical buses. Figure 4.b) illustrates a possible scenario. Based on the analyses provided above, this interconnect entails the least area overhead at no extra cost in latency.

IV. 3DFAR SYSTEM-LEVEL OPERATION

The 3DFAR architecture is capable of replacing any pipeline stage with a spare resource from another layer. We assume that the control inputs of the crossbar switches are connected to a few register bits, which in turn can be programmed via the 3DFAR firmware routine. Upon the detection of a fault, first the new faulty pipeline is suspended and all processes in execution are swapped out of context by the operating system. Then the firmware routine computes how many dynamic pipelines can still be setup with the current failures map, and programs the interconnect switches accordingly. The operating system then takes over to reschedule all the processes in execution, based on the healthy pipelines remaining.

design	freq. (MHz)	area (μm^2)	switch area	power (mW)	CPI
unprotected 2D	745	160,000	0%	201	0.402
2D w/switches	434	161,217	12%	222	0.402
StageNet	691	161,992	19%	274	0.561
3DFAR	714	41,234	7%	204	0.402

TABLE I: **Key system parameters for all solutions considered.** 3DFAR, stacked four layers deep, achieves almost optimal performance at much lower area and power cost than all other solutions.

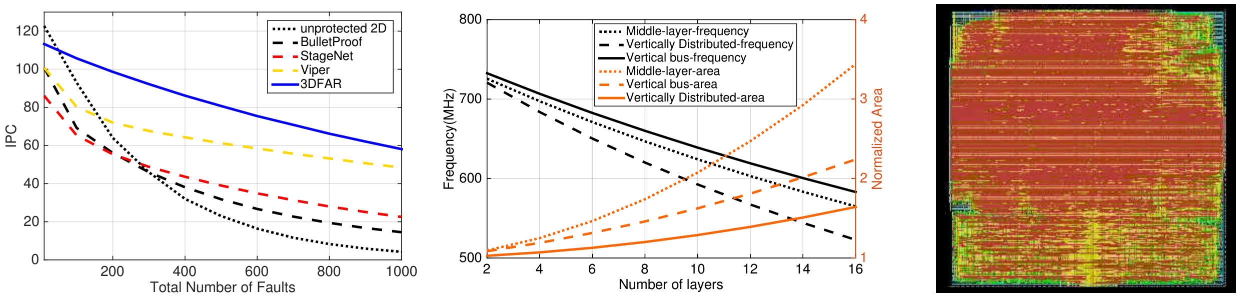
Faulty storage. To address situations where faults occur in register storage or caches, we equip each register and cache line with a few ECC bits [11], which allow to detect and correct one fault. After the first detection, we save the content of the register/ cache line and remove it from the available resources. We also count on having at least two read ports for each register file and cache unit, so that we can isolate them as soon as the one before last read port fails.

V. EXPERIMENTAL EVALUATION

To evaluate 3DFAR we deployed it on a 4-cores, 5-stage in-order pipeline (see Figure 2) implementing a subset of the Alpha instruction set architecture, equipped with separate instruction and data, 8KB two-way caches, with 1 cycle hit latency We augmented it with the vertical bus interconnect solution discussed in Section **??**, synthesized it on an IBM 45nm technology with Synposys' Design Compiler, and placed and routed it with Cadence's Encounter. To create the 3D layout (see Figure 5.c). we followed the specifications and design rules recommended in [9], and evaluated power and timing through SPICE simulations. We also considered three baseline designs: *unprotected 2D* is a 4-core processor with no reliability protection, laid out in a 2x2 matrix formation. *2D w/switches* is augmented with interconnect switches placed at the center of the matrix to minimize wire lengths, as recommended in [3]. We also implemented a buffered interconnect solution, *StageNet*, as specified in [3]. Finally, all systems were evaluated by executing a suite of 12 test programs overall executing for approximately 100,000 dynamic instructions. Table I reports our measurements for key system parameters in absence of faults, for all the solutions described. It can be noted that 3DFAR provides the same average clock cycles per instructions (CPI) as the unprotected 2D design, although its operating frequency is 4.1% lower at 714Mhz. In contrast, the 2D w/switches solution suffers from significant clock frequency slow down, while StageNet's CPI is 39% worse than 3DFAR, compounded with a slow down in clock frequency.

Fault model. Our fault model injects permanent transistor failures into any design component and any layer, proportionally to the area of the unit. Once a pipeline unit is hit by a fault, we disable the entire unit and trigger a dynamic reconfiguration via the 3DFAR firmware. If a fault hits an interconnect switch, we disable the unit connected to the output of that switch. If a fault hits the pipeline's control logic, we disable the entire pipeline. We assume that MIVs are implemented reliably (Section III): in Table I we accounted for one spare MIV every 100 [10]. To attain statistical confidence, we repeated each experiment on faulty processors 10,000 times, using different random seeds.

Performance in presence of faults. In Figure 5.a) we compared the robustness of 3DFAR against a number of recent reliability solutions: Viper [1], StageNet [3], BulletProof [4] and the basic unprotected 2D design. The plot evaluates the



(a) (b) (c)

Fig. 5: **Experimental evaluation and physical layout.** a) Performance of 3DFAR with a varying number of faults. b) Frequency and area of 3DFAR for a varying number of 3D layers. c) Layout of one 3D layer including a complete core (no cache) and all vertical bus switches.

performance of each solution in instructions-per-cycle (IPC) up to 1,000 concurrent faults. We considered only area-equivalent implementations of each solution, considering a budget of 2B transistors, similarly to the analysis in Figure 9 of [1]. With this budget, one could implement 128 *unprotected 2D* cores, 40 *Viper* pipelines, 22 *3DFAR clusters*, each 8 layers deep, 27 *BulletProof*, or 30 *StageNet* pipelines (the latter two having a fault-free throughput equivalent to about 4 in-order cores). We considered an equal MTBF for all designs, so the plot reflects also the solutions' life-time performance.

3DFAR provides better performance than all other solutions beyond 38 faults. *unprotected 2D* has the best performance when fault-free, but quickly degrades to the worst option at 298 faults. Note how the compact area footprint and the limited latency cost of 3DFAR deliver a significant IPC boost even over Viper. This advantage, however, starts to thin out beyond 800 faults. We believe this is due to the benefits of the Viper's decentralized control logic, which provides enhanced reconfiguration flexibility. On the other hand, 3DFAR's approach is orthogonal to Viper, and the two solutions could be easily integrated.

3DFAR cluster size. To ascertain the maximum number of layers that can be efficiently stacked together to form a cluster, we evaluated reliability and overhead over varying cluster sizes. Increasing cluster size improves reliability as there will be more available spare units, but it negatively affects area footprint and interconnect's propagation delay, which in turn impacts system's frequency. Figure 5.b) reports our findings for all three interconnect solutions discussed, showing that vertical bus switches provide the best performance.

VI. Related Work

Recent works for processor reliability have focused on unit sparing, exploiting natural redundancy in VLIW cores [4], introducing logic to enable dynamic reconfiguration around faulty pipeline stages [1,3], or sparing at the core level [12]. All these solutions assume or introduce an underlying fault detection mechanism(*e.g.*, BIST, software-based fault detection, *etc.*), similarly to 3DFAR. StageNet [3] is the most similar to 3DFAR conceptually; however, our solution provides a much more efficient unit-isolation mechanism, which leverages the 3D layout and a novel and efficient interconnect switch. Viper [1] also entails a completely distributed control logic solution. However, it comes with a number of limitations typical of distributed-control architectures and, as a result, its performance and scalability compares poorly against traditional chip multi-processors. Note that 3DFAR is complementary to Viper.

Research in reliability leveraging 3D layouts has also been explored. The authors of [13] investigate the concurrent execution of a program on two separate layers in a 3D design for fault detection, by using idle resources in the second layer. The authors of [14] propose a checker processor stacked vertically over a main processor. In this context, 3DFAR provides a complete recovery solution with extremely graceful performance degradation, compared to the limited and specialized approaches mentioned above.

VII. Conclusions

We presented 3DFAR, a novel reliability solution for multi-core processor designs, which leverages the system's natural redundancy to provide robustness to transistor failures. We exploit spatial locality of equivalent compute units to design efficient interconnect switches, with extremely low area footprint and minimal propagation delay, because of their innovative design and short vertical distances. Our evaluation indicates that 3DFAR greatly outperforms several state-of-the-art solutions, at any fault rate, when implemented with area-equivalent resources. At no-faults, 3DFAR requires only 7% more silicon (4-stacked) and it is 4% slower than an unprotected 2D design, while it outperforms StageNet by over 40%.

Acknowledgements. This work was supported by C-FAR, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

[1] A. Pellegrini, J. Greathouse, and V. Bertacco, "Viper: Virtual pipelines for enhanced reliability," in *Proc. ISCA*, 2012.

[2] I. Koren and S. Su, "Reliability analysis of N-modular redundancy systems with intermittent and permanent faults," *IEEE Trans. on Computers*, vol. 28, no. 7, 1979.

[3] S. Gupta, S. Feng, A. Ansari, and S. Mahlke, "StageNet: A reconfigurable fabric for constructing dependable CMPs," *IEEE Trans. Computers*, vol. 60, no. 1, 2011.

[4] K. Constantinides, S. Plaza, J. Blome, B. Zhang, V. Bertacco, S. Mahlke, T. Austin, and M. Orshansky, "BulletProof: a defect-tolerant CMP switch architecture," in *Proc. HPCA*, 2006.

[5] S. Panth, S. Samal, Y.Yu, and S. Lim, "Design challenges and solutions for ultra-high-density monolithic 3D ICs," in *Proc. S3S*, 2014.

[6] K. Constantinides, O. Mutlu, T. Austin, and V. Bertacco, "A flexible software-based framework for online detection of hardware defects," *IEEE Trans. on Computers*, vol. 58, no. 8, 2009.

[7] P. Emma, A. Buyuktosunoglu, M. Healy *et al.*, "3D stacking of high-performance processors," in *Proc. HPCA*, 2014.

[8] M. Barako, Y. Gao, A. Marconnet *et al.*, "Solder-bonded carbon nanotube thermal interface materials," in *Proc. ITHERM*, 2012.

[9] D. Kim, S. Kim, and S. Lim, "Impact of nano-scale through-silicon vias on the quality of today and future 3D IC designs," in *Proc. SLIP*, 2011.

[10] L. Jiang, F. Ye, Q. Xu, K. Chakrabarty, and B. Eklow, "On effective and efficient in-field TSV repair for stacked 3D ICs," in *Proc. DAC*, 2013.

[11] R. Swarz and D. Siewiorek, *Reliable computer systems: design and evaluation*, 3rd ed. A.K.Peters/CRC Press, 1998.

[12] S. Hari, M. Li, P. Ramachandran, B. Choi, and S. Adve, "mSWAT: Low-cost hardware fault detection and diagnosis for multicore systems," in *Proc. MICRO*, 2009.

[13] S. Safiruddin, M. Lefter, D. Borodin, G. Voicu, and S. Cotofana, "Zero-performance-overhead online fault detection and diagnosis in 3D stacked integrated circuits," in *Proc. NANOARCH*, 2012.

[14] N. Madan and R. Balasubramonian, "Leveraging 3D technology for improved reliability," in *Proc. MICRO*, 2007.