Approximate TSV-based 3D Stacked Integrated Circuits by Inexact Interconnects

Mahmoud S. Masadeh
Computer Engineering Department
Yarmouk University
Irbid 21163, Jordan

***Abstract*—Three-Dimensional Stacked Integrated Circuit (3D-SICs) based on Through-Silicon Vias (TSVs) provide a high-density integration technology. However, integrating pre-tested dies requires post-bond interconnect testing, which is complex and costly. An imperfect TSV-based interconnect indicates a defective chip that should be rejected. Thus, it increases the yield loss and test cost. On the other hand, approximate computing (AC) is a promising design paradigm suitable for error-resilient applications, e.g., processing sensory-generated data, by judiciously sacrificing output accuracy. AC perform inexact operations and accepts inexact data. Thus, introducing AC into 3D-SICs will significantly ameliorate the efficiency of design approximation. Therefore, this work aims to *increase the yield and reduce the test cost by accepting 3D-SICs with defected interconnects as approximate 3D-SICs*. This work considers 3D-SICs, where the sensor is stacked on logic (CPU) which is stacked on memory (DRAM). Then, use the memory-based interconnect testing (MBIT) approach to detect and diagnose the faulty interconnect. Based on the detected fault location and type, and for a maximum allowed error, some sensory 3D-SICs with defected LSBs interconnects are accepted and used in error-resilient and data-intensive applications. Targeting data lines only, 50% of the defected interconnects, i.e., least significant bits (LSBs), were accepted as approximate. Thus, the proposed work was able to significantly increase the yield. Two applications, i.e., ECG signal compression and detecting of their R peaks, demonstrated the effectiveness of using a sensory device with a faulty data line in its least significant 8-bits. The approximate ECG signals have a compression rate higher than the exact with negligible (around 0.1%) reduced accuracy.**

Keywords*—Approximate computing; Three-Dimensional Integrated Circuit (3D IC); Through-Silicon Via (TSV); testing; approximate communications; approximate interconnect; yield; energy efficiency

I. Introduction

Three-Dimensional Stacked Integrated Circuit (3D-SICs) based on Through Silicon Vias (TSVs) are emerging among industry and research groups. 3D-SIC is a package with a vertical stack of naked dies which are interconnected utilizing Through-Silicon Vias (TSVs) [1]. TSVs are electrical nails that are etched into the back-side of a thinned-down die, which permit that die to be vertically interconnected to another die. TSVs provide short vertical connections with reduced latency, low capacitance, and low inductance compared to wire-bonds. Thus, TSVs allow for more interconnects between dies with high speed and low power dissipation [2].

The feature-size scaling is becoming difficult and expensive. Moreover, the semiconductor industry is continuously demanding more functionality, bandwidth, and performance at smaller sizes, power dissipation, and cost. Thus, TSV-based 3D-SICs are the promising solution for such requirements [3]. 3D-SICs is a continuation of Moore's Law, which is called *more than Moore's law*. This design paradigm delivers various benefits such as reduced power consumption, reduced footprint, high bandwidth communication, low latency between dies, high transistor density per volume unit, and heterogeneous (e.g., logic, memory, radio frequency (RF) circuits, analog circuits, and sensors) integration [4].

The TSV-based 3D-SICs are promising products for various applications, e.g., the Internet of Things (IoT) and Bio-Medical applications [5]. These applications encompass a tremendous number of mobile and sensory devices, which continuously generate a tremendous quantity of data with redundant and noisy parts. Thus, these data can be processed approximately due to their intrinsic error-resiliency. Similarly, the data could be generated approximately.

Approximate Computing (AC) [6] is an emerging computing paradigm, among both industry and academia, that utilizes the intrinsic resiliency property of Recognition, Mining and Synthesis (RMS) applications. AC provides various benefits such as reducing computation speed, power consumption, and storage space, while achieving an acceptable output quality for various error-resilient applications [7]. Numerous approximation techniques, e.g., voltage over-scaling, approximate arithmetic units, approximate memory, and approximate communication, gained significant interest. However, AC is still immature research direction and does not have standards yet.

Similar to 2D ICs, those TSV-based 3D-SICs require manufacturing testing to meet the expected customer quality. The test operation is executed once at the beginning of the field operation of the IC. Thus, assuming the dies are fault-free, a faulty TSV that could represent a data line, address line, or control line, mandates discarding the whole 3D-SIC. Moreover, workload features could change for an operating IC. Thus, dynamic faults such as Electromigration (EM) should be considered during the operational lifetime. Therefore, to increase the yield and void rejecting an IC with a defected interconnect, *this work proposes accepting TSV-based 3D-SICs with defected interconnects and considering it as an approximate 3D-SICs*. Moreover, extra TSVs could be used to replace the defected interconnects that represent the most significant bits (MSB) of the data and address lines. On the other hand, the error that is caused by the least significant bits (LSB) of the data and address lines can be tolerated without TSV replacement. Extra TSVs are not targeted in this work due to their extra overhead.

The goals while manufacturing DRAM chips differ from those of logic chips, where DRAM designers target reduced area and refresh needs while logic designers target high performance with reduced energy. For best performance, DRAM and logic chips are manufactured individually based on different technology before integration. Thus, wide-IO memory-on-logic are realized as stacked-die applications.

This paper considers 3D-SICs, where the sensor is stacked on memory (DRAM) which is stacked on logic (CPU). Then, use the well-known memory-based interconnect testing (MBIT) approach to detect and diagnose the faulty interconnect. Based on fault location and type, and for a maximum application-dependent acceptable error, some defected 3D-SICs are accepted as approximate. Then, used in error-resilient and data-intensive applications, which tremendously increase the yield rate and reduce test cost.

The rest of this paper is arranged as follows. Section II explains near-sensor computing with various forms of integration. Section III demonstrates TSV fabrication steps, their possible defects and faults, as well as their fault models. The most relevant related work is explained in Section IV. Our proposed methodology is highlighted in Section V. In Section VI, as a case study on ECG signal, we evaluate the proposed methodology and then accept a 3D-SIC with an inexact TSV-based data line. Section VII highlights some of the future directions and concludes the paper.

II. Near-Sensor Computing

The number of sensory devices is expected to reach 75 billion by 2025 and 125 billion by 2030 [8]. They generate a huge amount of repetitious and unformed data. Usually, sensing and processing nodes have different functional requirements and varied manufacturing technology. Moreover, for data sensing, a noisy analog domain is utilized while the data is processed digitally on *von Neumann* computing devices. Thus, sensed data should be transferred from the sensing to the processing node. Therefore, various issues related to response time, data storage, data security, communication bandwidth, and energy consumption should be considered.

There are various forms of integration technologies for near-sensor computing including 3D monolithic, planer SoC, 3D heterogeneous, and 2.5D chiplet integration [9]. In a 3D monolithic integration, the system typically combines various functional layers of sensor, memory and processors in a 3D stacked structure via interlayer vias. For a planer SoC integration, the functional units are integrated with a planar wire connection. However, in 3D heterogeneous integration, different functional units are fabricated individually on different wafers. Then, integrated with advanced packaging technologies, such as TSVs, die-to-die, die-to-wafer and wafer-to-wafer interconnects. This work targets TSVs-based interconnects. For 2.5D integration, the chiplets with specific functions are connected through an interposer, which is a compromise between 2D and 3D packaging integration.

The unprecedented explosion of sensory-generated data and its usage in real-time applications mandates adopting a data-centric approach instead of a computing-centric approach. This enables a system with high performance and energy efficiency. Near-Sensor computing is the solution to provide efficient processing of sensory data with minimal data movement or transformation. In near-sensor computing, the operations of data generation, collection, and processing are performed closed to the sensory devices. The *conventional processing* of sensory-generated data includes data sensing, conversion from analog to digital, storing in memory, transmitting data to the processing unit, then data processing. These steps cause high latency and power consumption. However, the processing units in near-sensor computing reside beside sensors and process data at sensor nodes. Thus, the combination of sensing and computing functions reduces data movement. The sensory computing system performs data processing at two different levels of abstraction, i.e., low and high levels, as described next.

Low-Level Near-Sensor Processing: It removes the undesirable noise from the raw sensory-generated data and includes data filtering, noise suppression and feature enhancement, which are local operations. Such processing ameliorates the computational workload and improves the efficiency of high-level processing. It aims to optimize the features of the raw data. Usually, low-level filtering utilizes circuits located between the sensing devices and high-level processing units.

High-Level Near-Sensor Processing: It comprises the cognitive process that enables the identification of the input signals. It includes recognition, classification and localization. The authors of [10] presented a near-sensor CNN accelerator for image recognition where data processing is close to the sensors. With a near-sensor design, the energy consumption and speed of operation are 60X and 30X, more efficient, respectively, compared to related work. In [11], the authors showed that utilizing 3D stacked ICs (rather than 2D) for near-sensor NN accelerators provides high bandwidth, reduced energy consumption, and low latency of data transfer. Thus, this work targets 3D stacked ICs with inexact TSV-based interconnects.

Near-sensor computing is more complicated than near-memory computing because it includes a huge sensory-generated data of various types. Planer integration of sensors and processing units on a limited area reduces the reserved footprint for sensors. Thus, 3D integration, where sensors are mounted on the top layer while processing units are arranged on the bottom layers, will provide complete exposure for high fill factor. The short distance between the sensing and processing units delivers a high communication bandwidth and low latency. Thus, this work focuses on 3D-SICs by TSV.

III. Interconnect Fault Models

For TSV-based interconnects, this section explains the main used terminology, the basic stages of TSV fabrication, their possible defects, faults, and their fault models.

A. Terminology

Here, we explain various keywords that are used in the rest of the paper. A *defect* is an unintended difference between the implemented hardware and the intended design, emerged from the manufacturing process, e.g., open and bridge defects. The probability of defects in ICs grows with reduced feature size. *Failures* are the physical manifestation of the defect. Defects are generally modeled at a higher conception level

by *faults*, e.g., Stuck-at-Zero (SA0) and Stuck-at-One (SA1). Various defects may be represented with the same fault. A collection of faults with identical properties are grouped in a *fault model*, which should accurately reflect the behavior of defects; as they are used for generating and evaluating test patterns [12]. Faults can be detected by applying a series of *test vectors*; the obtained test responses are compared with golden fault-free responses. The fraction of detectable faults which is called *fault coverage* (FC) indicates the quality of the test.

B. TSV Fabrication Steps

The main manufacturing steps for TSVs, which are cylindrical copper nails, are shown in Fig. 1. These main steps are (*1*) etching of TSV holes: It should be vertical and uniform with a high aspect ratio, (*2*) oxidation: deposition of oxide to isolate the etch from the surrounding semiconductor, (*3*) barrier seed: a barrier layer of metals is deposited before filling the etch with copper. It will prevent the diffusion of the metal into the oxide, (*4*) plating: use copper or tungsten for filling which should be void-free, where the operation of filling should produce minimal stress to avoid warpage, and (*5*) chemical mechanical polishing (CMP): remove the extra layer on the top of the filling. Then, the TSV is ready.

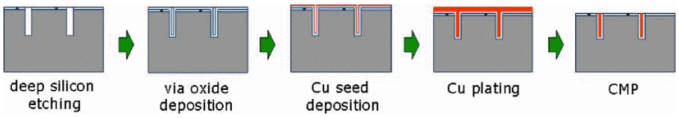

Fig. 1. TSV Fabrication Steps

TSVs can be organized into three classes, based on their fabrication time, during the IC manufacturing process: (*1*) via-first: TSV is fabricated before the front-end logic (FELO, transistor). However, it is more suitable for wafer handing rather than die and it requires adding constraints on design rules of transistor scaling, (*2*) via-middle: TSV is fabricated after the front-end logic (FELO) and before the Back-end of the line (BEOL), which is metal layers deposition, thinning, dicing, and assembly, and (*3*) via-last: TSV is fabricated after the IC fabrication process and before dicing and assembly. It has the lowest TSV fabrication process while being applicable for die and wafer stacking. However, during the manufacturing process, various reasons could cause a defect in the TSV, which are described next.

C. Interconnect Defects, Faults and Fault Models

The various manufacturing steps of TSVs are inherent sources of interconnects defects. There are various defects related to TSV including incomplete fill, pinhole, cracks, TSV misalignment with μ-bumps, TSVs Pinch-off, missing contacts between TSVs and the transistors, and Crosstalk between various TSVs [13].

Fig. 2 shows a general classification of interconnect fault models, which can be static or dynamic. Moreover, a defect can cause a single line or a multi-line fault. SA0 and SA1 are single-line static faults. However, wired-AND and wired-OR are multi-line static faults. Path delay fault (PDF) and path open fault (POF) are single-line dynamic faults. Whenever

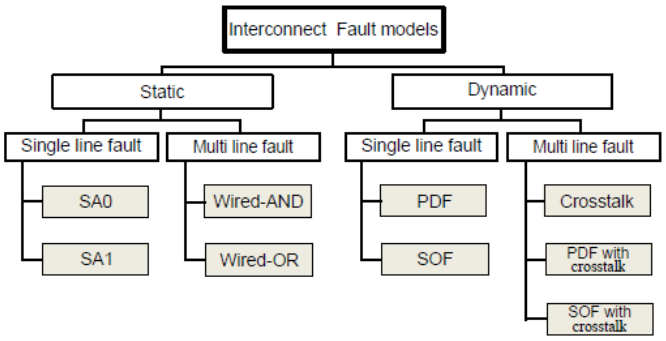

Fig. 2. Classification of Interconnect Fault Models [14]

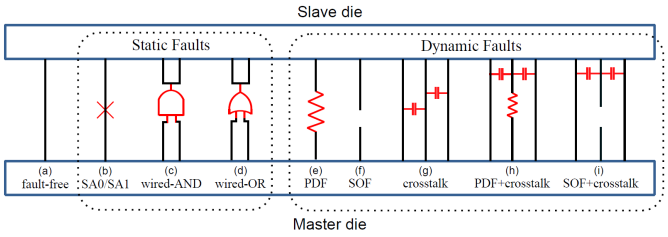

Fig. 3. Static and Dynamic Interconnect Faults [14]

crosstalk is introduced, we will have a multi-line dynamic faults. The interconnect faults are depicted in Fig. 3, including:

1) Stuck-at-Fault (SAF): Has two types which are stuck-at-0 (SA0) and stuck-at-1 (SA1) as depicted in Fig.3(b).

2) Bridge Fault: Simple bridge faults include wired-AND and wired-OR faults. Complex bridge faults also exist, such as (A dominate-AND B) where wire A is fault-free and wire B takes the value A $\cap$ B.

3) Path Delay Fault (PDF): A partial open line defect increases the line delay, e.g., rising or falling delay time (Fig.3(e)).

4) Stuck Open Fault (SOF): This is caused by a completely open line defect (Fig.3(f)).

5) Crosstalk Fault: As shown in Fig.3(g), faults on victim lines are caused by crosstalk from aggressive neighbours. Several crosstalk faults exist as described by the Maximum Aggressor (MA) fault models such as (1) glitch-up, (2) glitch-down, (3) falling delay, and (4) rising delay. Each fault has a specific behavior, while it represents the same phenomena.

6) Path Delay Fault (PDF) with Crosstalk: As shown in Fig.3(h), this is a compound fault where faults due to partial resistive opens are affected by crosstalk from neighbors.

7) Stuck Open Fault (SOF) with Crosstalk: As shown in Fig.3(i), this is a compound fault where faults due to complete open lines are affected by crosstalk from neighbors.

We will show the effects of these faults on TSV-based interconnect and the 3D-SIC as a whole.

IV. Related Work

There is a considerable number of publications that investigate approximate computing, IC testing, and 3D stacked

ICs. However, the portion of the research in approximate computing and hardware design that considers interconnects is scarce. Next, we introduce the most relevant work regarding approximate communication and approximate TSVs.

Recently, researchers investigating various techniques of *approximate communication* for approximate computing. They target network-on-chip (NoCs), aiming for reduced power consumption and latency. The proposed techniques rely on: 1) *lossy compression*: compress each packet and reduce its quality before transmission in order to reduce traffic intensity [15], 2) *value-prediction*: forecast data based on its locality to reduced the transmitted data [16], and 3) *protection-based*: approximate data by protecting the critical part to lower the cost of error correction [17]. These techniques significantly enhance performance and energy consumption. However, controlling the quality of communication is still a significant point. In [18], the authors proposed a hardware-based quality management framework for approximate communication to minimize the time needed for the approximation level calculation. Thus, they presented a new NoC design that observes the application error and adjusts the data approximation level accordingly.

Data transmission across chip interconnects requires a significant amount of time and energy. Thus, the authors of [15] proposed a framework for approximate bus architecture, which is conscious of approximable data. The proposed framework utilizes a light compression technique. For 0.5% quality loss at the application level, the proposed framework achieved a 29% performance improvement. In [19], the authors proposed a framework to reduce power consumption and communication latency of NoCs by incorporating a quality control method and data approximation to reduce packet size. For that, error-resilient variables are identified by analyzing the source code. When transmitting error-resilient variables, a lightweight lossy compression technique is utilized to significantly reduce packet size. In a closely-related work, the same authors explored, in another work, the possibility of using Reinforcement Learning (RL) to manage data quality [20].

The authors of [21], confirmed that the energy consumption of manycore is influenced by data movement, which demands energy-efficient and high-bandwidth interconnects. Towards this direction, they declared that *integrated optics* is an encouraging solution to control the bandwidth limitations of electrical interconnect. However, integrated optics with low-efficiency lasers have high power overhead. Thus, the authors of [21] proposed using *low-power optical signals* to transmit the least significant bits of floating-point numbers. Accordingly, their proposed design has 42% laser power reduction for image processing applications. Similarly, the authors of [22] presented a technique to design scalable *approximate nanophotonic interconnects*. Thus, enhance the interconnect energy efficiency by adjusting the transmission robustness to the application requirements. They achieved a 53% power reduction for output errors of 8%.

The authors of [23] proposed a runtime dynamic Built-In Self-Repair (BISR) technique to improve runtime reliability. For that, they used a test scheme to identify runtime and manufacturing defects. Then, replace defective TSVs with neighbour fault-free TSVs. However, each TSV has its test circuit which causes a large area and power overhead. The authors of [24] showed that testing of 3D-SICs is a challenge due to their complex structure. After stacking, the power and ground TSVs are connected to a grid that makes their testing a challenging task. Thus, they proposed a built-in self-test (BIST) architecture for power and ground TSVs. The proposed BIST enhances reliability by testing for full-open, pin-hole, and bridge faults. However, the proposed BIST introduces hardware overhead with low test coverage.

Previously, various works proposed approximate ICs by designing exact ICs and accepting the defective with minimal fault coverage as approximate ICs [25]. Others proposed designing approximate ICs, and accepting a defective approximate IC if the manufacturing error is within the acceptable approximation error [4]. However, the proposed chips were 2D, not 3D and the approximation is for the logic while considering the interconnect as fault-free. To the authors' knowledge, none of the previous works proposed using ICs with *defective interconnects* as approximate ICs nor targeted designing approximate 3D-SICs, which we propose here. This work mainly targets the communication interconnect itself, i.e., the TSV, as a hardware component. Our proposed idea is a simply different and efficient way. We test and diagnose the faulty TSV-based interconnect with zero area overhead, the ability to detect static and dynamic faults with at-speed testing, and a short test execution time. Then, the output quality of the defected 3D-SIC with defected interconnects is analyzed for a given quality metric. Based on that, some defected 3D-SICs are accepted as approximate ones. Thus, the yield is increased.

V. Proposed Methodology

In this section, we provide a detailed explanation of the proposed methodology. First, we explain Interconnect's built-in self-repair (IBISR). Then, revise memory-based interconnect testing (MBIT). Consequently, the assumed TSV layout is explained because multi-line faults are position-dependent. Next, how a faulty TSV-based data line could be considered approximate is explained.

A. Interconnect Built-In Self-Repair (IBISR)

The researcher of [26] proposed architecture of test and repair of a defect of TSV in 3D-IC, where BIST structure detects a defective TSV. Then, neighbours of the proposed BISR structure isolate and repair the defective TSV. This enhances the yield with an area overhead. The authors of [27] introduced a novel approach for repairing the deficient TSVs in 3D-ICs where interconnect built-in self-test (IBIST) is utilized. Then, the obtained results from IBIST provoke the repairing of defective TSV based on the given BISR structure. They employ repetitious TSV and the time-division multiplexing access (TDMA) in the case of multi defective TSV. However, the high fault rates and TSV footprint make the spare-based repair solutions inadequate [28]. In this work, to keep zero area overhead, we will not introduce interconnect repair. However, it is under investigation for closely related future work.

B. Memory based Interconnect Test (MBIT)

The authors of [14] proposed a Memory Based Interconnect Test (MBIT) approach for 3D-SICs where memory is stacked on logic by testing interconnects through memory read and write operations. MBIT solution can complete at-speed testing

and *diagnosis* and is able to detect all static and dynamic faults. Moreover, MBIT has zero area overhead and allows flexible patterns to be applied. The required test time is much lower than traditional based solutions such as Boundary Scan, but is three times slower than hardwired BIST solutions. However, BIST solutions have a large area overhead and cannot apply flexible patterns. Utilizing MBIT, the minimum set of test patterns required to detect all static and dynamic faults are the patterns to detect *PDF with crosstalk* and *SOP with crosstalk*. We assume a single fault at a time where the number of data lines is L_d = 16, and the number of address lines L_a = 16. We simulate memory test patterns, for a memory die stacked on a logic die that consists of a MIPS64 processor, by using the MIPS64 simulator in [29]. The simulator can handle a maximum of L_d = 64-bit data lines and L_a = 12-bit address lines (lowest 3 bits are byte offset).

C. TSV Layout

The TSV lines represent address, data, and control lines. Also, it include ground and power lines between stacked dies. Testing for multi-line dynamic faults requires knowing the exact layout of the address and data lines. For clarification, we assume a regular TSV array of size 4 × 4 to demonstrate how to generate test patterns for multi-line dynamic faults. Thus, knowing the exact layout is required to accurately analyze the 3D-SIC performance. We assume that a TSV victim is affected by the closest neighbour, i.e., 1^{st} aggressor model. Thus, as shown in Figure 4, a victim TSV (group 1) could be affected with a maximum of 8 aggressors (group 2, 3, and 4).

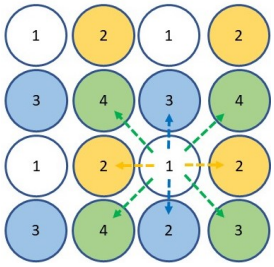

Fig. 4. TSV Layout and Grouping

The JEDEC Solid State Technology Association defends open standards for the microelectronics industry [31]. It provided a standard for stackable Wide-I/O Mobile DRAMs which describes the logic-memory interface for functional and mechanical characteristics, widening the conventional 32-bit DRAM interface to 512 bits. Figure 5 shows the interface for JEDEC Wide-I/O with 1200 connections, where each channel has 300 interconnections. Each channel consists of 6 rows by 50 columns. JEDEC's Wide-I/O interface includes four memory channels, each with 128 bi-directional data lines. Moreover, each channel has 51 control and address signals. Thus, the layout of the interconnections is given where a faulty TSV would be affected by the adjacent ones.

D. The Proposed Methodology

The authors of [32] designed and implemented back-illuminated CMOS image sensors (CIS) (BICIS) with TSV-based bonding between the 3 layers. The number of TSVs for connecting pixel substrate and DRAM substrate is about 15,000 and about 20,000 for connecting the DRAM substrate and the logic substrate. Thus, the fabrication of 35000 TSV could result in defective ones, which will reduce the yield. Therefore, we propose to accept defected TSVs that still provide accepted quality. Algorithm 1 shows the proposed methodology (as a list of steps) for accepting a 3D-SIC with defected TSV-based interconnects as *approximate 3D-SIC*.

Algorithm 1 The Proposed Methodology for Approximate TSV-Based 3D-SICs with Inexact Interconnects

Result: 3D-SIC with inexact interconnects
Manufacture Wafer#1 (CPU);
Manufacture Wafer#2 (DRAM);
Manufacture Wafer#3 (Sensor);
Test and Dice Wafer#1 (CPU);
Test and Dice Wafer#2 (DRAM);
Test and Dice Wafer#3 (Sensor);
Stack the 3 dies by TSVs;
Perform Interconnect Test and Diagnosis;
if *No fault in Address, Data, or Control lines* **then**
    3D-SIC is accepted as Exact; ▷ This is the main goal
else
    if *Control line if faulty* **then**
        3D-SIC is Rejected;
    end if
    if *Data line if faulty* **then**
        Analyze the effect of error based on a given error metric and fault model;
        if *The effect of error is acceptable* **then**
            3D-SIC is Accepted with Faulty Data line; ▷ Data line investigated in this work
        else
            3D-SIC is Rejected;
        end if
    end if
    if *Address line if faulty* **then**
        Analyze the effect of error based on a given error metric and fault model;
        if *The effect of error is acceptable* **then**
            3D-SIC is Accepted with Faulty Address line; ▷ Address line will be investigated in future work
        else
            3D-SIC is Rejected;
        end if
    end if
end if

The wafers of the CPU, DRAM, and Sensor are manufactured then diced. The CPU, DRAM, and Sensor chips are tested at the wafer level and at the die level. Dies stacking is performed through TSV fabrication between the dies. Then, we test the TSV-based interconnects, i.e., MBIT, which implies applying the full list of test patterns. It will detect all possible faults, i.e., various static and dynamic faults. If the obtained test response matches the expected fault-free response for all applied test patterns, the tested 3D-SIC is exact with 100% fault coverage. That represents the ideal case. However, when there is a mismatch between the obtained test response and the expected fault-free response, the tested 3D-SIC is defective, and it should be rejected.

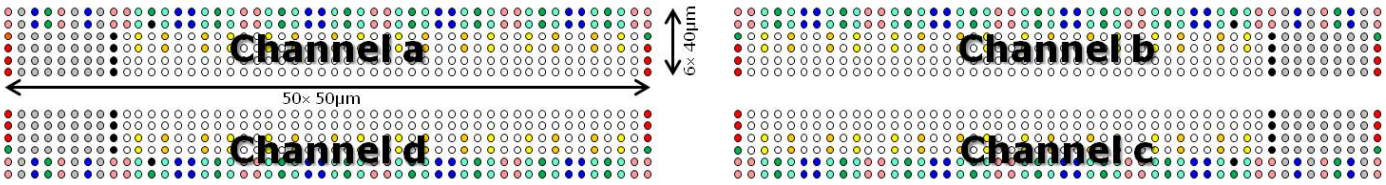

Fig. 5. The Interface for JEDEC Wide-I/O [30]

For a 3D-SIC with defected interconnect, *we perform interconnect diagnosis to identify the exact location and type of the defect*. Rather than discarding the defective 3D-SIC and reducing the yield, we propose to accept some defects based on its location and the used error metric. If a *control line* is identified to be faulty, the operation of the chip will be indeterministic, where it could perform read operation rather than write. Thus, we propose to reject the IC whenever a control line is defective. If a *data* is identified to be faulty, we evaluate its effect on the quality of the final results. If an *address line* is identified to be faulty, this is similar to a faulty address decoder, which is considered as a future work.

In approximate computing, the maximum acceptable error depends on the application, the applied inputs, and user preferences [33]. For that, different error metrics could be used for accuracy evaluation [34] [35], including: (1) Error Rate (ER): which is the percentage of erroneous outputs among all outputs, (2) Error Distance (ED): the arithmetic difference between the exact output and the approximate output for a given input, (3) Mean Error Distance (MED): the average of ED values for a set of outputs obtained by applying a set of inputs, and (4) Relative Error Distance (RED): which is the ratio of ED to the exact output. *Next, we explain how a 3D-SIC with a faulty TSV-based data line could be accepted as approximate IC based on various error metrics*.

E. Faulty Data Line

The number of data lines is L_d = 16 and a faulty data line will be denoted as D_n, for 0 $\leq$ n $\leq$ 15. We assume that the data lines have a normal distribution, where the probability of any line to have a value of 0 or 1 are equal, i.e., $P_{D_n}(0) = P_{D_n}(1)$ = 0.5. Under the assumption that a single fault could occur at a time [12], the error magnitude is 2^n for a faulty D_n data line. The acceptability of a 3D-SIC with a defected interconnect as an approximate one depends on the position of faulty *data line* and the used *accuracy metric*. Next, we explain different error metrics with various fault models:

1: Fault Model is SAF:

Error Metric is ED: For SA0 the data line is always 0, i.e., $P_{D_n}(0)$ = 1, and $P_{D_n}(1)$ = 0. Similarly, for SA1 the data line is always 1, i.e., $P_{D_n}(0)$ = 0, and $P_{D_n}(1)$ = 1. The error magnitude is 2^n for a faulty D_n data line with the assumption of a single fault at a time. Thus, we accept the 3D-SIC as approximate when $2^n >$ ED, and reject it when $2^n \leq$ ED. For large acceptable error, i.e., ED, more chips are accepted as approximate ones. Thus, the yield is increased. When the faulty data line (D_n) is located in the MSB of the design, e.g., 8 $\leq$ n $\leq$ 15, the error magnitude would be large. Thus, the defective chips are rejected, which reduces the yield. On the other hand, when the faulty data line (D_n) is located in the LSB of the design, e.g., 0 $\leq$ n $\leq$ 7, chips are accepted as approximate ones since their error magnitude is small, i.e., $2^7 >$ ED.

Error Metric is ER: The ER indicates the ratio of erroneous outputs among all outputs. A SAF data line, i.e., SA0 or SA1, will give the expected value for 50% of the time and an erroneous result for the rest of the time. Thus, the ER is 50% and the 3D-SIC with defected interconnect is accepted when the allowed ER $\leq$50% and rejected when the ER $>$ 50%.

Error Metric is MED: Under the assumption of one fault at a time the MED metric equals the ED. For multiple errors, MED is given by Equ. 1, where the average ED for a set of faulty data lines is evaluated. The ED for a single data line ranges from $2^{15} = 32K = 32768$ to $2^0 = 1$, based on its location.

$$MED = \frac{1}{16}\sum_{n=0}^{15} ED_n = \frac{1}{16}\sum_{n=0}^{15} 2^n \tag{1}$$

Error Metric is RED: Under the assumption of a single fault at a time the RED for a faulty data line is 1, while it is 0 for an exact interconnect.

2: Fault Model is Bridge Fault (Wired-AND, Wired-OR):

TABLE I. INTERCONNECT VALUE FOR BRIDGE FAULT (WIRED-AND, WIRED-OR)

Exact		A AND B		A OR B	
A	**B**	**A**	**B**	**A**	**B**
0	0	0	0	0	0
0	1	0	**0**	**1**	1
1	0	**0**	0	1	**1**
1	1	1	1	1	1

The bridge fault will give a final value based on: 1) its type; wired-AND or wired-OR, and 2) the value of its neighbour. As shown in Table I, faulty data line with wired-AND will give 0 for 75% of the time and 25% for the rest of the time. Similarly, a faulty data line with wired-OR will give 0 for 25% of the time and give 1 for the rest 75% of the time. Thus, we notice that the bridge fault is mapped to SAF with ER of 25%.

3: Fault Model is Path Delay Fault (PDF):

The dynamic fault of PDF for less than a clock cycle will not cause the circuit to fail. Thus, the data line will deliver an exact value.

4: Fault Model is Stuck Open Fault (SOF):

The SOF represents a completely open line. The floating data line is assumed to have a stable value of 0, a stable value of 1, or changes from 1 to 0. Thus, eventually, the SOF could be equivalent to SAF.

5: Fault Model is Crosstalk:

Figure 4 shows the physical layout of TSVs assuming the 1^{st} aggressor model, where the victim is affected only by the closest neighbour aggressors. Generally, any K^{th} aggressor model can be used, where *K* indicates the maximum TSV distance between victim and aggressors. The authors of [36] showed that restricting *K* to 1 is sufficient.

5.1: PDF with Crosstalk:

A transition at the victim, e.g., from 1 to 0, will be affected by the opposite transition, e.g., from 0 to 1, at the neighbours. Thus, the effect of crosstalk is similar to PDF.

5.2: SOF with Crosstalk:

Detecting SOF with crosstalk requires causing a transition on the victim while keeping the aggressors unchanged. The effect of this model is equivalent to SAF.

F. Possible Repair Scheme

Post-bond interconnect testing for memory stacked on logic requires special consideration since: 1) the stacked dies have different fabrication labs, 2) memory providers are unwilling to incorporate DFT such as JTAG for interconnect testing, and 3) the used DFT can not provide high coverage for dynamic faults. Generally, TSV repair depends on having extra TSVs. However, to avoid extra hardware we will not use extra TSVs nor perform TSV repair.

VI. Case Study

In this section, we evaluate the proposed methodology which accepts a 3D-SIC with an inexact TSV-based data line. Thus, consider it as approximate 3D-SIC, and utilize it in error-resilient applications where reduced accuracy is tolerated.

Biosignal is a human body variable that can be measured and monitored where it provides information on the health status of individuals. Wearable devices sense and process different crucial signs, e.g., electroencephalography (EEG), Electrocardiography (ECG), electrooculogram (EOG), and electromyography (EMG), and send the data to the cloud or to a smartphone. Various biomedical applications accept minor errors or small quality degradation in the values of the biosignal. Electrocardiogram (ECG) is a non-invasive examination that records and shows the electrical activities produced by heart muscle during a cardiac cycle. The ECG test is a standard clinical mechanism for analyzing abnormal heart rhythms and assessing the general condition of a heart. As shown in Figure 6, each ECG cycle consists of 5 waves called PQRST. A complete ECG is recorded using 10 electrodes capturing 12 leads (signals) to get a total picture of the heart. Next, we explain R-peaks detection of Electrocardiography (ECG) signals and its compression assuming the least significant 8-bits are faulty due to inexact TSV-based data line.

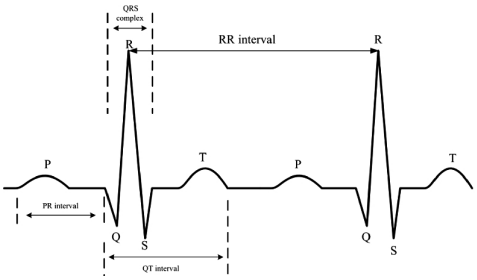

Fig. 6. Parts of an ECG Signal

A. Detecting R-Peaks of ECG Signal

ECG is one of the most critical diagnostic tools for different cardiac diseases. Fast automated detection of the P wave, QRS complex, and T wave is necessary for the early detection of cardiovascular diseases (CVDs). The detection of R-peak is important in all kinds of electrocardiogram (ECG) applications. Utilizing the approach proposed in [37], we performed R peak detection for 32 ECG recordings of the MIT-BIH arrhythmia [38]. For that, we use three parameters, i.e., true-positive (TP), false-negative (FN), and false-positive (FP). TP represents the number of correctly detected R peaks while FN is the number of missed R peaks. FP is the number of noise spikes erroneously classified as R peaks. Utilizing these parameters, we computed various statistical measures including Accuracy (Acc), Precision (positive predictability), Sensitivity/Recall (Se), and F1-Score, as given in the following equations.

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$Precision\ (P) = \frac{TP}{TP + FP} \tag{3}$$

$$Recall/Sensitivity\ (Se) = \frac{TP}{TP + FN} \tag{4}$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

Table II shows the various obtained accuracy metrics, which indicate the high performance of the R peak detection methodology. For the same ECG signals, we created an approximate version of it. For that, the various points of each ECG are approximated by randomly setting one of the least significant 8-bits to zero. This emulates the behaviour of a faulty data line (with SA0 fault) of a sensory device for recording ECG signals.

Stuck-at-0 fault at the least significant data bits did not change the number of total beats, i.e., TP + FN, since the R peak have a high magnitude value. R peak detection of approximate ECG signals missed 57 peaks and classified 123 noise spikes are R Peaks, i.e., FN=57 and FP=123. However, regardless of these false the *accuracy* decreased insignificantly from 99.88% to 99.74%. Similarly, prediction precision, recall, and F1-score reduced insignificantly with less than 0.1%. Thus, a faulty bit in the least significant 8-bits of data line will have

TABLE II. PERFORMANCE OF R PEAK DETECTION METHOD USING ECG EXACT AND APPROXIMATE DATA

	Total beats (TP+FN)	TP (beats)	FN (beats)	FP (beats)	Accuracy	Precision	Recall	F1-Score
Exact ECG	70453	70448	5	77	99.88	99.89	99.99	99.94
Approx ECG	70453	70396	57	123	99.74	99.82	99.91	99.86

reduced effect at application level. Various machine learning-based models could be used as a classifier to detect the QRS complex [39] [40], which are considered as future direction.

B. Compression of Biomedical Signals

Figure 7 shows the architecture for wearable ECG monitoring. The biosignals are acquired, filtered, digitized, *compressed*, and transmitted to the smartphone or cloud server for analysis. The distinct features are obtained, then the classification process detects anomalies. Reducing the amount of transmitted data, through discarding the least significant bits or/and data compression, extends the battery lifetime of mobile devices. Data compression helps to supply the required low-power wireless connection with a slightly large bandwidth.

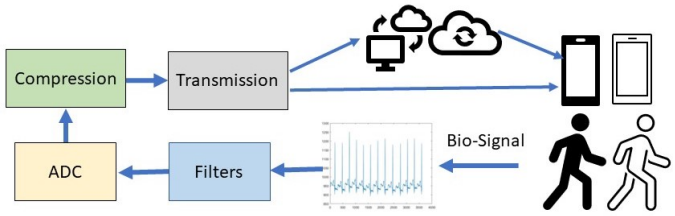

Fig. 7. IoT-Based Wearable ECG Monitoring System

MIT-BIH cardiac arrhythmia database is a widely used database in recent years [38]. MIT-BIH database was supplied by the Massachusetts Institute of Technology with 48 records each is 30 minutes in length. Utilizing the approach proposed in [41], we performed ECG compression for 32 ECG recordings of the MIT-BIH arrhythmia. Then, for the same ECG signals, formed an approximate version of it. Thus, the different points of each ECG are approximated by randomly setting one of the least significant 8-bits to one. This mimics the SA1 fault of a sensory device for recording ECG signals. To assess the ECG signal compression, various metrics are used such as:

Compression Rate (CR): measures the degree of data compression and expressed as given in Eq. 6. Thus, the highest is the best.

Root Mean Squared Error (RMSE): It is a metric for specifying the similarity between two sets, i.e., the original and compressed signal, as expressed in Equ. 7, where *y* is the original signal, *x* is the compressed signal, and *n* is the number of samples of the signal. Thus, the lowest is the best.

$$Compression\ Rate\ (CR) = \frac{Uncompressed\ Size}{Compressed\ Size} \quad (6)$$

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2} \quad (7)$$

TABLE III. RESULTS OBTAINED FOR 32 MIT-BIH RECORDS

	CR	RMSE	Accuracy
Exact ECG	51.7	3.54	99.89
Approx ECG	53.8	6.21	97.91

As shown in Table III, the CR of the exact ECG signal is 51.7 and it is enhanced to 53.8 for the approximate ECG signals. Moreover, the RMSE is 3.54 for the exact ECG signal and it is increased to 6.21 for the approximate ECG which still very acceptable. This work aims to have a high-performance classifier on the compressed signal, both exact and approximate ECG. Thus, the decompressed ECG after lossy compression is classified and detected based on a supporting vector machine (SVM) classifier. The accuracy is 99.89 for the original signal which is reduced insignificantly to 97.91 for the approximate ECG signal. We notice the increase in the compression ratio while keeping the performance of classification of the compressed signal. Thus, a 3D-SIC with a sensory device where the least 8-bits of a data line are faulty can be easily accepted in various applications.

VII. CONCLUSION

Near-sensor computing is a well-known approach to designing efficient hardware for intelligent sensory processing. Data processing at sensory nodes provides a reduced area and time with efficient energy consumption. Thus, it is suitable for real-time and data-intensive applications. However, low-level and high-level near-sensor processing mandates new integration forms and processing algorithms utilizing emerging devices. Although near-senor processing is promising with a great future potential, most of the existing work is still in the development stage and confined to specific applications. This work proposes accepting 3D-SICs with defected TSV-based interconnects as *approximate 3D-SICs*. For this purpose, a sensory device is stacked on a memory die which is stacked on a logic die. To specify if the tested IC is acceptable, context-aware testing is required. Then, a faulty IC is investigated to detect its useability as an approximate one. To evaluate the effectiveness of using a sensory device with a faulty data line in its least significant 8-bits, we performed two applications on ECG signals. First, detecting R peaks of ECG signals then compressing the ECG signals. Both applications demonstrated the usability of the faulty data line in the LSBs of a sensory device. The obtained accuracy metrics, i.e., compression rate, root mean square error, accuracy, precision, recall, and F1-score, showed that a 3D-SIC with a sensory device where the least 8-bits of a data line are faulty can be easily accepted in various applications with enhanced yield.

REFERENCES

[1] P. Garrou, C. Bower, and P. Ramm, *Handbook of 3D integration, volume 1: technology and applications of 3D integrated circuits*. John Wiley & Sons, 2011.

[2] M. Taouil, M. Masadeh, S. Hamdioui, and E. J. Marinissen, "Interconnect test for 3D stacked memory-on-logic," in *Design, Automation Test in Europe Conference Exhibition*, 2014, pp. 1–6.

[3] E. J. Marinissen, *Testing 3D Stacked ICs Containing Through-Silicon Vias*. New York, NY: Springer New York, 2011, pp. 47–74.

[4] M. Masadeh, O. Hasan, and S. Tahar, "Approximation-Conscious IC Testing," in *30th International Conference on Microelectronics*, 2018, pp. 56–59.

[5] U. K. et al., "8Gb 3D DDR3 DRAM using through-silicon-via technology," in *IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, 2009, pp. 130–131,131a.

[6] M. Masadeh, O. Hasan, and S. Tahar, "Comparative Study of Approximate Multipliers," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, 2018, pp. 415–418.

[7] ——, "Input-Conscious Approximate Mmultiply-Accumulate (MAC) unit for Energy-Efficiency," *IEEE Access*, vol. 7, pp. 147 129–147 142, 2019.

[8] T. P. Truong, H. T. Le, and T. T. Nguyen, "A reconfigurable hardware platform for low-power wide-area wireless sensor networks," in *Journal of Physics: Conference Series*, vol. 1432, no. 1. IOP Publishing, 2020, p. 012068.

[9] F. Zhou and Y. Chai, "Near-sensor and in-sensor computing," in *Nature Electronics*, vol. 3, 2020, pp. 664–671.

[10] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "Shidiannao: Shifting vision processing closer to the sensor," in *2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, 2015, pp. 92–104.

[11] T.-H. Hsu, Y.-C. Chiu, W.-C. Wei, Y.-C. Lo, C.-C. Lo, R.-S. Liu, K.-T. Tang, M.-F. Chang, and C.-C. Hsieh, "Ai edge devices using computing-in-memory and processing-in-sensor: From system to device," in *2019 IEEE International Electron Devices Meeting (IEDM)*, pp. 22.5.1–22.5.4.

[12] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI test principles and architectures: design for testability*. Elsevier, 2006.

[13] M. Masadeh, "Interconnect Testing for 3D Stacked Memories," Delft University of Technology, Delft, Netherlands, 2013.

[14] M. Taouil, M. Masadeh, S. Hamdioui, and E. J. Marinissen, "Post-bond interconnect test and diagnosis for 3-d memory stacked on logic," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1860–1872, 2015.

[15] J. R. Stevens, A. Ranjan, and A. Raghunathan, "Axba: An approximate bus architecture framework," in *Proceedings of the International Conference on Computer-Aided Design*, 2018, pp. 1–8.

[16] A. Perais and A. Seznec, "Bebop: A cost effective predictor infrastructure for superscalar value prediction," in *IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2015, pp. 13–25.

[17] Y. Chen, M. F. Reza, and A. Louri, "Dec-noc: An approximate framework based on dynamic error control with applications to energy-efficient nocs," in *IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 480–487.

[18] Y. Chen and A. Louri, "An online quality management framework for approximate communication in network-on-chips," in *Proceedings of the ACM International Conference on SuperComputing*, 2019, pp. 217–226.

[19] ——, "An approximate communication framework for network-on-chips," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1434–1446, 2020.

[20] ——, "Learning-based quality management for approximate communication in network-on-chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3724–3735, 2020.

[21] J. Lee, C. Killian, S. L. Beux, and D. Chillet, "Approximate nanophotonic interconnects," in *Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip*. ACM, 2019.

[22] J. Lee, C. Killian, S. Le Beux, and D. Chillet, "Distance-Aware Approximate Nanophotonic Interconnect," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 27, no. 2, nov 2021.

[23] D. K. Maity, S. K. Roy, and C. Giri, "Built-In Self-Repair for Manufacturing and Runtime TSV Defects in 3D ICs," in *IEEE International Test Conference India*, 2020, pp. 1–6.

[24] D. Han, Y. Lee, S. Lee, and S. Kang, "Hardware Efficient Built-in Self-test Architecture for Power and Ground TSVs in 3D IC," in *International SoC Design Conference (ISOCC)*. IEEE, 2021, pp. 101–102.

[25] Z. Jiang and S. K. Gupta, "An ATPG for threshold testing: Obtaining acceptable yield in future processes," in *International Test Conference*. IEEE, 2002, pp. 824–833.

[26] M. Benabdeladhim, A. Fradi, and B. Hamdi, "Interconnect BIST based new self-repairing of TSV defect in 3D-IC," in *International Conference on Engineering MIS (ICEMIS)*, 2017, pp. 1–4.

[27] M. Benabdeladhim, W. DGHAIS, F. ZAYER, and B. Hamdi, "An Efficient Fault Tolerance Technique for Through-Silicon-Vias in 3-D ICs," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 7, pp. 264–270, 2018.

[28] M. Nicolaidis, V. Pasca, and L. Anghel, "I-BIRAS: Interconnect Built-In Self-Repair and Adaptive Serialization in 3D Integrated Systems," in *16th IEEE European Test Symposium*, 2011, pp. 208–208.

[29] "WinMIPS64," last accessed February 7, 2022. [Online]. Available: http://indigo.ie/ mscott/

[30] S. Deutsch, B. Keller, V. Chickermane, S. Mukherjee, N. Sood, S. K. Goel, J.-J. Chen, A. Mehta, F. Lee, and E. J. Marinissen, "DfT architecture and ATPG for Interconnect tests of JEDEC Wide-I/O memory-on-logic die stacks," in *IEEE International Test Conference*. IEEE, 2012, pp. 1–10.

[31] "Wide I/O Single Data Rate (JEDEC Standard JESD229). JEDEC Solid State Technology Association." last accessed March 10, 2022. [Online]. Available: http://www.jedec.org.

[32] Y. Kagawa and H. Iwamoto, "3D Integration Technologies for the Stacked CMOS Image Sensors," in *International 3D Systems Integration Conference (3DIC)*, 2019, pp. 1–4.

[33] M. A. Laurenzano, P. Hill, M. Samadi, S. Mahlke, J. Mars, and L. Tang, "Input Responsiveness: Using Canary Inputs to Dynamically Steer Approximation," in *SIGPLAN Conference on Programming Language Design and Implementation*. ACM, 2016, p. 161–176.

[34] Z. Vasicek and L. Sekanina, "Evolutionary design of approximate multipliers under different error metrics," in *International Symposium on Design and Diagnostics of Electronic Circuits Systems*, 2014, pp. 135–140.

[35] M. Masadeh, O. Hasan, and S. Tahar, "Error analysis of approximate array multipliers," *arXiv preprint arXiv:1908.01343*, 2019.

[36] R. Weerasekera, M. Grange, D. Pamunuwa, H. Tenhunen, and L.-R. Zheng, "Compact modelling of Through-Silicon Vias (TSVs) in three-dimensional (3-D) integrated circuits," in *IEEE International Conference on 3D System Integration*, 2009, pp. 1–8.

[37] J.-S. Park, S.-W. Lee, and U. Park, "R peak detection method using wavelet transform and modified shannon energy envelope," *Journal of healthcare engineering*, vol. 2017, 2017.

[38] "MIT-BIH Arrhythmia Database Directory," last accessed March 25, 2022. [Online]. Available: https://archive.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm

[39] M. Masadeh, O. Hasan, and S. Tahar, "Machine learning-based self-compensating approximate computing," in *IEEE International Systems Conference (SysCon)*, 2020, pp. 1–6.

[40] M. Masadeh, Y. Elderhalli, O. Hasan, and S. Tahar, "A Quality-assured Approximate Hardware Accelerators–based on Machine Learning and Dynamic Partial Reconfiguration," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 17, no. 4, pp. 1–19, 2021.

[41] L. Zheng, Z. Wang, J. Liang, S. Luo, and S. Tian, "Effective compression and classification of ecg arrhythmia by singular value decomposition," *Biomedical Engineering Advances*, vol. 2, p. 100013, 2021.