# Soft-errors in embedded systems

Shahzaib Waseem
*B.Eng. Electronics Engineering*
*Hochschule Hamm-Lippstadt*
Lippstadt, Germany
shahzaib.waseem@stud.hshl.de

***Abstract*—Soft errors are transient faults in digital circuits caused by external radiation or even internal radiation from radioactive material embedded chips. They have been a major barrier to the reliability of embedded systems for a long time. The consequences of these soft errors have grown with the miniaturization and increased density of devices, especially in safety-critical applications such as space, automotive, and health. This paper proposes a comprehensive analysis of soft errors, showcasing in particular their causes, impact, estimation techniques, and mitigation approaches. Primary sources include high-energy cosmic rays and alpha particle emissions from packaging materials, while shrinking transistor sizes increases vulnerability to said soft-errors. The effects of soft errors on data integrity and system reliability are reviewed, highlighting error types such as Silent Data Corruption (SDC) and Multiple Bit Upsets (MBUs). Estimation techniques, such as Architectural Vulnerability Factor (AVF) and fault injection methods, are presented as fundamental tools to quantify soft error risk. In regard to mitigating these risks, the paper looks into process-level improvements such as Silicon On Insulator (SOI), architectural solutions like ECC and redundancy, and software-based approaches using checkpoint and rollback. Finally, it also points out radiation shielding materials for systems operating in high-exposure environments. By investigating these dimensions, this work highlights the importance of an integrated approach towards soft error mitigation and assurance of system robustness in modern embedded applications.**

***Index Terms*—soft-errors, cosmic rays, neutrons, alpha particles, radioactive, SDC, DUE, AVF, shielding, SOI**

## I. Introduction

Embedded systems are abundant in today's society - they power critical applications across all industries such as aerospace, healthcare, automotive, telecommunications, etc. As these systems become tinier, more complex, and denser, they become subject to reliability challenges. Among the most important challenges are that of soft-errors, which are temporary faults that can disrupt data integrity and the output of subsystems, without causing any permanent damage in the hardware. This section will briefly introduce soft-errors, emphasize their important in embedded systems, and provide an overview of the structure of this paper. Before the topic of soft-errors is explored further, an understanding of hard-errors is beneficial [1].

Hard-errors are permanent faults that physically damage hardware components. The most common cause of these errors is wear-and-tear of the embedded system within which they occur. Other causes include manufacturing defects, and power surges (in those systems that rely on AC power from wall-sockets). Hard-errors typically require replacement or repair of damaged parts or components, as the affected components cannot recover on their own - the damaged components are unable to function as they should, and are no longer fit or suitable for use in the system. Some examples of hard-errors included damaged transistors, faulty memory cells, and short circuits. Although both hard and soft-errors negatively impact system functionality, hard-errors are unrecoverable without manual intervention [1].

Soft-errors, or transient faults, are temporary disturbances in a system's memory cells or logic-circuits. Their sources may be external, or internal - more on this will be explained in upcoming sections. While it's true that soft-errors disappear on their own, they still risk system stability, at least for a amount of time worth considering. Table 1 summarizes and provides an overview of the comparison between soft-errors and hard-errors. Below are key reasons why soft-errors are categorized as important in embedded systems [1]:

1) Reliability in critical applications
    a) Aerospace (e.g., communication systems, satellites, space probes, autopilot, etc.)
    b) Healthcare (e.g., CT scanning, MRI machines, surgery robots, etc.)
    c) Automotive (e.g., braking systems, self-driving, air bags, etc.)
2) Increased vulnerability in modern electronics
    a) Transistor compacting: Smaller transistors have lesser amount of charge, making them more susceptible to bit flips
    b) Increase in density: Densely packed transistors (more transistors per meter squared) increases the probability of soft-errors
3) Most important effects of soft-errors
    a) Data disruption; affecting system outputs and computations.
    b) Systems restarting after freezing or crashing, especially if the mechanism to handle soft-errors is not implemented.

There is especially high impact on high-altitude devices or systems from soft-errors. Systems such as satellites, airplanes, etc. are usually above the level of some atmospheric layers. These atmospheric layers protect us not only from harmful

Ultra Violet (UV) rays from the Sun, but also from radiation or emission that causes soft-errors. Being located above these atmospheric layers means that these high-altitude devices don't get the filtered radiation that reaches the planet's surface, but instead the raw, less-filtered version. These conditions demand and require high-altitude devices to have extraordinary standards of protection and mitigation against soft-errors. Faulty values and temporary faults are especially intolerable for systems that operate at these high altitudes, because a failure occurring in them would certainly lead to loss of life and destruction [1].

TABLE I
COMPARISON OF SOFT ERRORS AND HARD ERRORS

| Characteristic | Soft Errors | Hard Errors |
|---|---|---|
| Nature | Temporary | Permanent |
| Cause | Ionization | Hardware damage |
| Impact on Hardware | Low, no repair | High, repair required |
| Examples | Bit flips,<br>transient faults | Damaged parts,<br>short-circuits |
| Fixes | ECC bits, process<br>tech., etc. | Repair,<br>redundancy |

## II. Sources of soft-errors

The origin of soft-errors can be classified as either external, or internal. External soft-errors mainly refers to events that take place in a system's surrounding area, due to our environment, or rather the way our universe works. Internal sources refers to the material used to construct ICs, electronic components, etc. Regardless of the source, they trigger disturbances at a microscopic level (atomic level), that cause transistors to change their state unintentionally - a bit flip occurs. This section explores the sources of soft-errors, mainly being cosmic rays, and packaging-material emissions [1].

### *A. Cosmic rays*

Cosmic rays are groups of high-energy particles entering the Earth's atmosphere from celestial bodies in outer space, as a result of chaotic events and activities such as stars exploding (supernova), or from Hawking radiation in black holes, etc. Even after traveling thousands of light years, upon reaching the Earth's atmosphere, they can penetrate electronic devices, and cause soft-errors. The following list includes the composition of said high-energy cosmic rays [1]:

1) Protons:
   the most common and highest in proportion in cosmic rays. While they are high in energy, they don't directly cause soft-errors as much as neutrons
2) Neutrons:
   especially with high kinetic energy, can penetrate electronic devices relatively easily, and are the major cause of soft-errors through cosmic rays
3) Muons and pions:
   although their role in causing soft-errors is relatively lower, they are strongly responsible for the chain reaction that occurs at the Earth's atmosphere that allows emission of other particles that do cause soft-errors

A series of events occurs from the moment cosmic rays hit the Earth's atmosphere, which gives rise to a general mechanism leading to soft-errors due to cosmic rays. When cosmic rays reach Earth's atmosphere, they undergo chemical and physical reactions with the atmospheric particles, including neutrons, that end up reaching ground level. High-energy neutrons can penetrate semiconductor material, and come into contact with silicon atoms [1]. This is a violent collision involving an immense amount of kinetic energy transfer from the high-energy neutrons to the silicon atoms in the silicon lattice. Subsequently, this causes electrons in the valence band of the hit silicon atoms to gain enough energy to reach the conductance band, thereby changing the total number of free electrons in the silicon. Silicon is the material with which transistors are made, and free-electrons in silicon become the basis for the function of said transistors. Modern transistors such as CMOS transistors rely on the voltage of their 'gate' to change their state. When there is enough voltage in the gate of a given transistor, the transistor goes into 'active' mode. Otherwise, it stays in the 'cut-off region' where it does not allow current to pass through it [1]. Given that voltage is defined as the potential difference between two nodes in a circuit, and that the potential of a node is determined by its number of charges (electrons or holes), high-energy neutrons can give rise to changes in the voltage of the gate of transistors by adding or taking away the charge at the gate of the transistor. This is what is known as a 'bit flip'. The process of changing the distribution or proportion of charges is also called 'ionization'. Figure 1 visualizes how ionization occurs in silicon in a transistor [1].

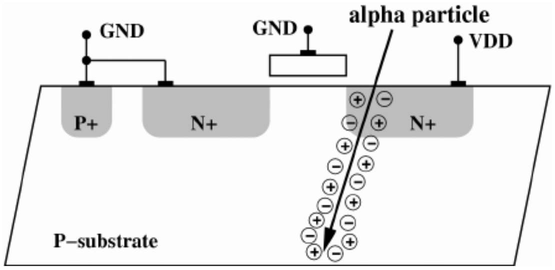

Fig. 1. Example of a transistor under ionization effect from radiation [1]

### *B. Alpha particles*

Radioactive materials are in abundance all around the world, and radioactive decay from them gives off radiation that is yet another source of soft-errors. Radioactive atoms are those that possess too many or too few neutrons in their nucleus [2]. Their tendency to become stable causes them to release radiation in the form of alpha, beta, and gamma particles. Alpha particles in particular are of interest, because their chemical composition of two protons and two neutrons, along with their high kinetic energy in radioactive radiation, makes them a source of soft-errors. Unstable isotopes that emit a lot of radioactive radiation are said to be highly active [2]. Examples of such highly active elements include Uranium and

Thorium, along with their derivatives. Another example of an unstable isotope is Pb-210, which is a Lead isotope found in solder. As mentioned earlier, soft-errors can arise from an external source or an internal source. Since solder is the sole tool used to connect conductors in electrical circuits, solder bumps from within IC packaging become the internal source of soft-errors. However, this is not the only internal source of soft-error [2]. The EMC (Epoxy Mold Compound) used in the packaging of ICs is also a source of alpha particles that strikes silicon atoms from within. Both solder-bumps and EMCs are primarily issues resulting from how Flip-chip technology or Ball Grid Array (BGA) packaging is made. Modern electronics require more complex and denser electronic circuits to be fit in small IC packages, and this requires a high number of connection points or terminals to be made available for connection through exposed copper protruding from the IC. Flip-chip or BGA IC packaging takes advantage of the hidden bottom layer of ICs by utilizing its vast surface area to place these protruded copper heads. Though clever, it comes at a cost of being a source of alpha particles. [2]Although alpha particles themselves carry low penetration power such that a piece of paper is able to absorb and stop them, there is no way to put pieces of paper within the IC without it burning or affecting the functionality of the IC. However, this is not implying that there is no way to prevent these alpha particles from within the IC to cause soft-errors, as there will be methods discussed in the upcoming sections that explain how to deal with these internal sources of alpha particles [2].

## III. Locations of soft-errors in embedded systems

With the explanation of soft-errors and their sources covered, the exact location of sof-errors in embedded systems needs to discussed, or the exact components in embedded systems within which soft-errors take place. The two main locations or components are:

1) Memory cells
2) (Digital) Logic circuits

Memory cells are essentially the individual transistors from which memory devices are made. Memory devices in electronics are those digital logic circuits that allow for storing and retrieving of bits [10]. The two types of such digital logic circuits are:

1) Latches
2) Flip-flops

Latches are storage elements used in logic circuits to store one single bit, which is a state of a transistor. Latches are level-triggered, meaning that their stored bit value changes based on the input given to them as long as an enable signal is active. That is to say, when the enable signal is at a level of 1 or HIGH, they will store whatever input they receive, and when the enable signal is at a level of 0 or LOW, they will ignore whatever input they receive, and not store anything [10]. Flip-flops are the other type of storage elements used in logic circuits to store one single bit. Flip-flops are edge-triggered, meaning that their stored bit value changes based on the input given to them only when their input clock signal either changes from a logic-level LOW or 0 to a logic-level HIGH or 1, or vice versa [4] Both latches and flip-flops are made with transistors, and as mentioned in the section of ’Sources of soft-errors’, transistors are subjected to ionization from varying sources, which gives rise to soft-errors. [10] [4]

Logic circuits also use transistors as their fundamental building blocks, just like in memory cells. The term ’logic circuits’ is simply an umbrella term and short-form for all kinds of digital logic circuits. Logic circuits perform logical operations in a computer or embedded system. Examples of logic circuits and their logical operation are summarized in the table below:

TABLE II
Mapping of Digital Logic Circuits to Operations [4]

| Logic Circuit | Operation |
|---|---|
| AND Gate | Outputs 1 if all inputs are 1 |
| NOT Gate | Inverts the input |
| Multiplexer (MUX) | Selects one input based on control signals |
| Demultiplexer (DEMUX) | Routes one input to multiple outputs |
| Decoder | Converts binary input into a single active output |
| Encoder | Converts multiple active inputs into binary output |
| Half Adder | Adds two bits; outputs sum and carry |
| Full Adder | Adds three bits; outputs sum and carry |

Although logic circuits have been described separately from memory cells in this section, they are not independent of each other. Memory cells developed with either latches or flip-flops are simply another type or category of logic-circuits, just in a particulary unique arrangement. Latches and flip-flops use the concept of ’feedback’, which the other logic circuits don’t use. Therefore, it is often preferrable to talk about memory cells separately from logic circuits. Next, the types and effects of soft-errors will be discussed.

## IV. Types, effects, and estimation techniques of soft-errors

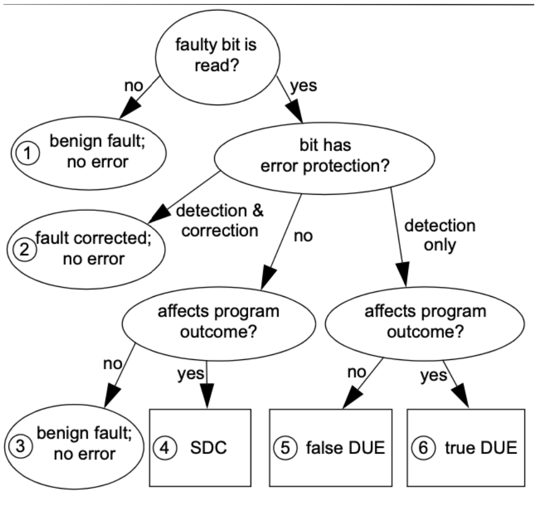

Fig. 2. Analysis of effects of soft-errors [6]

Soft-errors are mainly of two types [6]:

1) Single Bit Upset (SBUs)
2) Multiple Bit Upsets (MBUs)

Single Bit Upsets (SBUs) occur when a single bit in a memory cell or logic circuit flips from a 0 to a 1, or vice versa. SBUs are the dominant type of soft-errors, because only one bit needs to be affected. Multiple Bit Upsets (MBUs) on the other hand, take place relatively less frequently, since MBUs are defined as multiple bits being affected, as the name suggests [6]. This happens because, mathematically speaking, the probability of multiple transistors facing ionization is much lesser than that of a single transistor being ionized. This is further supported by the fact that ionization such as alpha particles lose their kinetic energy for every unit of distance they move through a silicon material [6]. In order for multiple bits to be affected by this alpha particle, it would not only need to travel towards multiple transistors, but also have enough energy to introduce electron-hole pairs, where each instance of this ionization takes away a significant amount of energy from said alpha particle. The same also applies for the other sources of soft-errors [6].

Soft-errors have mainly two types of effects [6]:

1) Silent Data Corruption (SDCs)
2) Detected Unrecoverable Errors (DUEs)

As figure 2 describes, Silent Data Corruption or SDCs occur when a fault is not detected. This is especially dangerous in applications where safety is critical, and where silent or undetected faults can lead fatal circumstances [6]. Detected Unrecoverable Errors or DUEs are when a fault occurs, and it is detected. Referring to figure 2, this leads to variations in what type of DUE would occur. If the detected error can be fixed, then it is a false DUE. Only when the detected error cannot be fixed anymore, i.e., when the detected error is unrecoverable, you get a true DUE [6].

SDCs and DUEs can be quantified using standard metrics such as [6]:

1) Mean Time Between Failures (MTBFs)
2) Failure In Time (FITs)

Mean Time Between Failures or MTBFs represent the average time that passes between one failure and the next. It is essentially describes an approximate amount of grace time a system has before it is expected to have undergone at least one failure [6]. Failures in Time or FITs represent the number of failure that occur in a given unit of time. A standard seen and used in literature is 1 billion hours ($10^9$ hours) [6]. FIT can be understood as the frequency of failures, while MTBFs can be seen as the period of failures. This is further supported by the fact that a larger MTBF value implies a system is safer from soft-errors, and a smaller MTBF value implies that a system is less safe. On the other hand, a smaller FIT value implies that a system is safer from soft-errors, while a larger FIT value indicates that a system is less safe [6]. Therefore, it can be deduced that MTBFs and FITs are inversely proportional to each other, just like how the concept of frequency and period is in mathematics and physics .

Soft error rate estimation is very important for a reliable design of embedded systems especially for safety-critical applications. There are different ways to measure the probability and consequences of soft errors, allowing engineers to plan a targeted mitigation strategy [6].

A well-known metric is the Architectural Vulnerability Factor (AVF) which can be described as the probability that a bit flip in a particular piece of hardware will create an observable error in the output of a program running on the system. AVF is calculated based on how an application behaves and whether the erroneous data is used during execution. For instance, if a bit in unused memory is flipped and does not propagate up to the program output, then that event has an AVF = 0 [6].

This metric is useful in the prioritization of protection for critical components, such as caches, registers, and control units, where errors are most likely to propagate and cause system-level faults [6].

A related metric to AVF is Architecturally Correct Execution (ACE), the ratio of clock cycles in which a system component executes without being affected by soft errors. Essentially, it means that the functional requirements of the system are still fulfilled as expected [6].

ACE is a metric representing how often a system does what it is supposed to do, even in the presence of possible defects. The higher the percentage of ACE, the lower the number of vulnerabilities and, consequently, the lower the AVF. Together, AVF and ACE provide a complete framework for understanding and quantifying a system's vulnerability to soft errors [6]. Along with these metrics, several fault injection methods are widely used to estimate the robustness of a system against soft errors. The main approaches for fault injection are either simulation-based or hardware-based [6]. Simulation-based fault injection involves inserting synthetic faults into the virtual model of the system under study and thereafter evaluating the possible mitigation measures. For instance, engineers can inject bit flips in memory or logic circuits while running programs to test the system's response [6]. On the other hand, hardware fault injection uses physical devices to mimic real-life soft error conditions, such as voltage fluctuations or radiation emitters. All these techniques are indispensable in industries like aerospace and automotive, where it is required to make sure that the systems have been tested for operation under the most extreme environmental conditions [6].

AVF and ACE metrics, together with fault injection techniques, allow engineers to understand the behavior of embedded systems in the presence of faults. The tools aid not only in identifying the vulnerable components but also in guiding the effective mitigation strategies to enhance the reliability of the system [6].

## V. Soft-error mitigation techniques

Soft error mitigation is an integral part of the dependability of embedded systems, which is more crucial in safety-related fields like aerospace, automotive, and healthcare. Mitigation strategies essentially give rise to a multi-layered approach,

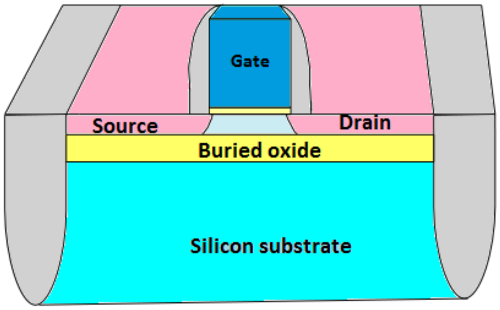

Fig. 3. Illustration of an example of fully-depleted SOI [11]

including improvements at the hardware level, optimization at the architectural level, software-based solutions, and physical protection.

One such approach is the use of process technology solutions, like Silicon On Insulator (SOI) [12]. SOI technology places an insulating layer between the silicon substrate and the active semiconductor region . This, in effect, reduces the interaction of radiation with the bulk silicon and consequently significantly lowers the generation of electron-hole pairs—one of the major sources of soft errors [12]. There exist two major SOI flavors: partially depleted SOI, which provides moderate performance and soft-error-resistance improvements, and fully depleted SOI with much better energy efficiency and robustness. New packaging materials also help in soft-error mitigation. Take, for example, the modern solder bumps and epoxy compounds designed to contain lesser radioactive isotopes like Pb-210, which would then lead to lesser alpha particles being emitted from the packaging materials [12].

At the architectural level, systems often include error detection and correction mechanisms to mitigate the effects of soft errors [7]. For example, parity bits and Error Correction Codes (ECC) are often used in memory systems. Parity bits can detect single-bit errors by indicating whether the number of 1s in a block of data is even or odd. With ECC, this capability is extended to correct single-bit errors and detect multi-bit errors, such as those caused by Multiple Bit Upsets (MBUs) [7]. Advanced ECC implementations, such as Single Error Correct, Double Error Detect (SECDED), have become standard in applications requiring the highest levels of data integrity, from data centers to space missions [7].

Another architectural solution is redundancy, implemented through techniques like redundant multi-threading and lockstepping [9] [8]. Redundant multi-threading is when the same operations are performed simultaneously on different threads or cores and their results are compared to find the differences caused by soft errors [8].Lockstepping is similar, but involves executing the same process on two processors simultaneously and comparing their results at every clock cycle [9]. Such techniques are very important in fields like real-time and safety-critical systems, such as avionics and medical equipment, where errors need to be detected and corrected in time.

Additionally, software-based strategies really take part in soft error mitigation [3]. The classic example of checkpoint and rollback mechanisms where the system periodically saves its state into a safe place, on error, the system resumes operation from the last known good state—this way, the integrity of data is maintained without a total reboot of the system [3]. Another important example can be represented by the algorithmic error detection techniques analyzing computational output anomalies and discarding results probably affected by errors. These techniques complement the hardware and architectural solutions with an additional dimension of fault tolerance.

Finally, radiation shielding is a physical barrier to soft errors that occur as a result of radiation [5]. Materials like polyamide or nylon showcase significant effectiveness in alpha-particle shielding, particularly for high-performance and radiation-prone environments such as space and nuclear facilities [5]. This is especially important and useful for protecting against alpha particles that emit from within the PCB or IC, such as from solder bumps in flip-chip technology ICs, and from the epoxy mold compound in IC packaging [5]. Current research and literature in finding better candidates for radiation-shielding, such as lightweight grapheme-based films, aims at increasing protection while maintaining the performance, thus making them suitable for aerospace and mobile electronic applications [5].

Combining both techniques offers a way to decrease soft errors significantly and increases the potential reliability of a system. Each approach in this respect targets different vulnerabilities, and their combined application guarantees comprehensive fault tolerance in embedded systems.

## VI. Conclusion

Soft errors in embedded systems are increasingly becoming a concern as devices have to be smaller, faster, and power-efficient. These transient faults, mainly caused by cosmic radiation and material impurities, threaten the reliability of systems used in safety-critical applications such as aerospace, healthcare, and automotive industries. The causes and effects of soft errors are discussed; in particular, the paper highlights significant vulnerabilities such as silent data corruption (SDC) and multiple bit upsets (MBUs). Engineers now can use metrics such as the Architectural Vulnerability Factor (AVF) and fault injection techniques to assess the probability and effects of such errors, which enables them to make better-informed design decisions.

In addition, it has presented a combination of process-level improvements and architectural enhancements, coupled with software strategies and physical shielding to mitigate the risks. Technological solutions, such as Silicon On Insulator (SOI), Error Correction Codes (ECC), checkpoint and roll-back mechanisms, and advanced radiation shielding materials, provide strong and effective defense mechanisms against soft errors. These solutions can be incorporated into ensuring high reliability in embedded systems even within radiation-prone environments.

As technology continues to advance, future research will have to concentrate on further lessening the susceptibility of smaller and more complex devices to soft errors. To be able to maintain reliability in upcoming embedded system generations, innovations are required in materials science, error detection algorithms, and fault-tolerant architectures. Soft error resilience will continue to be one of the fundamental aspects of system design for making sure that critical applications function safely and dependably.

## References

[1] Robert Baumann. “Soft errors in advanced computer systems”. In: *IEEE design & test of computers* 22.3 (2005), pp. 258–266.

[2] Robert C Baumann. “Soft errors in advanced semiconductor devices-part I: the three radiation sources”. In: *IEEE Transactions on device and materials reliability* 1.1 (2001), pp. 17–22.

[3] Nicholas S. Bowen and Dhiraj K Pradham. “Processor- and memory-based checkpoint and rollback recovery”. In: *Computer* 26.2 (1993), pp. 22–31.

[4] Brian Holdsworth and Clive Woods. *Digital logic design*. Elsevier, 2002.

[5] Chaitali V More et al. “Polymeric composite materials for radiation shielding: a review”. In: *Environmental chemistry letters* 19 (2021), pp. 2057–2090.

[6] Shubhendu S Mukherjee, Joel Emer, and Steven K Reinhardt. “The soft error problem: An architectural perspective”. In: *11th International Symposium on High-Performance Computer Architecture*. IEEE. 2005, pp. 243–247.

[7] Riaz Naseer and Jeff Draper. “DEC ECC design to improve memory reliability in sub-100nm technologies”. In: *2008 15th IEEE International Conference on Electronics, Circuits and Systems*. IEEE. 2008, pp. 586–589.

[8] Isil Oz and Sanem Arslan. “A survey on multithreading alternatives for soft error fault tolerance”. In: *ACM Computing Surveys (CSUR)* 52.2 (2019), pp. 1–38.

[9] Emre Ozer et al. “Error correlation prediction in lockstep processors for safety-critical systems”. In: *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE. 2018, pp. 737–748.

[10] Abu Sebastian et al. “Memory devices and applications for in-memory computing”. In: *Nature nanotechnology* 15.7 (2020), pp. 529–544.

[11] SignOff Semiconductors. *Silicon On Insulator ( SOI )*. URL: https://signoffsemiconductors.com/silicon-on-insulator-soi/. ()

[12] Rahul Kr Singh, Amit Saxena, and Mayur Rastogi. “Silicon on insulator technology review”. In: *International Journal of Engineering Sciences & Emerging Technologies* 1.1 (2011), pp. 1–16.