Chapter 2
Hardware Faults

Abstract Chapter explains hardware faults, their origins, dependency on technology used, and some known solutions of fault toleration using error correction codes and redundancy hardware schemes. Shown that with growing density of hardware there is a risk of multiple temporary fault faults grows at order of magnitude prime concern for designers of new computer systems for safety critical application. Hardware faults occur due to natural phenomena, such as ionized radiation, variations in the manufacturing process, vibrations, etc. We present in this chapter a short introduction to hardware faults, show the typical fault types and patterns and give examples how to deal with these faults.

2.1 Introduction

Transient faults are a huge concern in silicon-based electronic components, such as SRAM, DRAM, microprocessors, and FPGA. Those are devices that have a well-documented history of transient faults mainly caused by energetic particles.

An important obstacle for safety critical systems to achieve maximum reliability is represented by the susceptibility of those systems to faults produced by radiation. In addition, as manufacturing technologies evolve, the effects of ionizing radiation become a primary concern.

Due to the reduction in size of the transistors and the reduction in critical charge of logic circuits, the susceptibility of technologies to information corruption is increasing [1–3]. A concrete example for this is given in Fig. 2.1 [1], indicating that the malfunction rate (soft error rate) is massively increasing in processor logic for decreasing manufacturing sizes.

Collisions of particles with sensitive regions of the semiconductor can change stored information in ROM/RAM and lead to logic errors, for example, in processor circuits. In this context, single event upsets (SEU) are effects induced by the strike of a single energetic particle (ion, proton, electron, neutron, etc.) in the semiconductor.

I. Schagaev and T. Kaegi-Trachsel, *Software Design for Resilient Computer Systems*, DOI 10.1007/978-3-319-29465-0_2

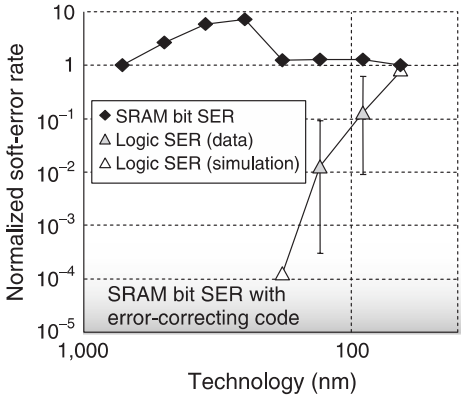

Fig. 2.1 Comparison of the SRAM bit soft-error rate (malfunctions) [1]

The charged particle passes through the semiconductor material leaving an ionized track behind leaving sufficient energy in the circuit to have an effect on a localized area of the electronic device.

Single event upsets can occur either through the impact of primary particles (e.g., direct ionization from galactic cosmic rays or solar particles) or by the secondary particles generated after the strike (indirect ionization). It affects many different types of devices, designed using various technologies resulting in data corruption, high current conditions, and transient disturbances. If such single event upsets are not handled well, unwanted functional interrupts and catastrophic failures could take place.

However, single event upset is not fully representative anymore. The number of multiple bits affected by a single event was relatively small using previous silicon technologies. Modern technologies are much more vulnerable and single event can affect multiple bits, so-called multiple bits upsets (MBU). The single event rate affecting multiple bits is expected to increase in the coming years [2].

Traditionally, two different types of techniques have been used to mitigate upsets: fault avoidance, and fault tolerance techniques. Fault avoidance techniques, usually at the device level, such as silicon on insulator or hardened memory cells, usually involve IC process changes. These techniques have drawbacks in terms of cost, chip area, and speed of operation.

Several low level fault-tolerant techniques for memories and processor registers have been used to mitigate upsets: error detection codes and error correction codes. *Hamming* codes [4] have been largely employed to detect and correct single bit upsets. However, these techniques are vulnerable to the increasing multiple bit upset ratio.

Reed-Solomon codes [5] can correct a large number of multiple faults but are not suitable for hardware implementation in terms of design complexity, additional memory required (area), and inherent latency (performance).

At a higher level, circuit-level mitigation techniques with some amount of redundancy, such as N-modular redundancy and voting [6], are frequently used.

Triple modular redundancy [7] is a very simple solution and perhaps the most implemented approach in RT safety critical applications. However, the area and therefore, also power consumption penalty is large (higher than 200 %).

Since the malfunction sensitivity of sequential and combinational logic is increasing dramatically [1, 2] and the multiple bit upset ratio of SRAM is also rising, the previous mechanisms by themselves will not provide the necessary reliability for safety critical systems.

2.2 Single Event Effects and Other Deviations

Semiconductor devices experience single event upsets in two major forms: in the form of destructive effects which result in permanent degradation or even complete failure of the device and therefore also affecting functionality (permanent fault), and in the form of nondestructive effects, causing no permanent damage (malfunctions).

However, in this sense when something irregular affects the system the question must be answered which of the two cases it is. How to deal with this deviation is the part of the HW/SSW design.

In terms of events, the most common are single event upsets and multiple cell upsets, which both belong to the single event category:

Single bit upset is a single event upsets or single bit upset, meaning, one event produces a single bit error. This type of error is very common on SRAM.

Multiple cell upset is a multiple bit upset for one event regardless of the location of the multiple bits. For example a FPGA where one routing bit gets an impact from a high energetic particle, affecting several memory positions. Hence, multiple cell upsets involves both upsets, the ones that can be corrected by error correction codes as well as those which cannot with reasonable overhead.

Multiple Bit Upset is a subset of multiple cell upsets. It is a multiple bit upset for one event that affects several bits in the same word. This type of deviation cannot be corrected by error correction codes with reasonable overhead. However, it is possible to partially avoid multiple bit upsets by using specific layout design of memory cells.

Growing density of logic elements of wafer technology, miniaturization of manufacturing processes, and high clock speeds will inevitably increase rate of so-called intermittent faults.

Inevitable variations in the manufacturing process will induce these faults at higher rate; moreover impact of these faults will be lasting up to several seconds [2], increasing complexity of recovery. In contrast to external faults such as radiation, intermittent faults are triggered by internal events, such as voltage, temperature, and timing variations.

Figure 2.2 illustrates such an intermittent fault.

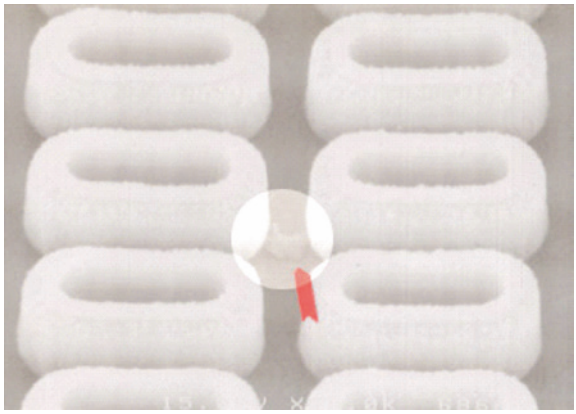

Fig. 2.2 Residue-induced intermittent fault in a DRAM chip [1]

2.3 Conclusion

This short introduction is by no means comprehensive, but illustrates that not all faults can be treated at the hardware level, especially the more complex fault types. Examples of these faults are multiple bit upsets and intermittent faults.

Both of mentioned types of faults are expected to occur more often in the future with decreasing hardware structure sizes and growing demand of higher frequencies of operation.

It is, therefore, obvious that software must assist the hardware in the fault detection and recovery process.

References

1. Baumann R (2005) Soft errors in advanced computer systems. IEEE Design Test Comput. 22(3):258–266
2. Constantinescu C. Intermittent faults and effects on reliability of integrated circuits. RAMS 2008. Annual, pp 370–374, Jan. 2008
3. Wells P., Chakraborty, K., Sohi, G.S.: Adapting to intermittent faults in future multicore systems. In: 16th International Conference Parallel Architecture and Compilation Techniques, PACT pp 431 (2007)
4. Hamming R.: Error detection and error correction codes. In: The Bell System Technical Journal, volume XXVI(9), pp 147–160 (1950)
5. Reed I, Solomon G (1960) Polynomial codes over certain finite fields. J Soc Ind Appl Math 8(2):300–304
6. Takaesu K, Yoshida T. Construction of a fault-tolerant voter for n-modular redundancy. Electronics and Communications in Japan (Part II: Electronics), 87(12):62–71, 2004
7. Birolini Alessandro. Reliability Engineering Theory and Practice. Springer, 20014. Ed.8

http://www.springer.com/978-3-319-29463-6

Software Design for Resilient Computer Systems

Schagaev, I.; Kaegi-Trachsel, Th.

2016, XIV, 214 p. 70 illus., 51 illus. in color., Hardcover
ISBN: 978-3-319-29463-6