Fast Single Event Upset Detection Technique for Processors Using an All-Digital Flow

Stefan De Raedemaeker
stefan.deraedemaeker@kuleuven.be

Karel Appels
karel.appels@kuleuven.be

Jeffrey Prinzie
jeffrey.prinzie@kuleuven.be

ESAT-ADVISE Research Lab, KU Leuven University, Belgium

February 2025

Abstract

The increasing interest in space exploration has elevated the demand for advanced data processing and autonomous operations. With artificial intelligence and advanced digital signal processing gaining popularity in commercial products, spacecraft systems must ensure accurate behavior in harsh environments. On-board computers are sensitive to single event transients and single event upsets inducing state flips that propagate through logic gates. This work presents a method for detecting single event transients and single event upsets in flip-flops immediately after they occur by using a double flip-flop sampling technique. This technique does not involve any timing delay on the data path, making it useful for high performance and high reliability systems in space. These upsets can be reported at the processor to take further mitigation actions, such as roll-back, pipeline flush, and system reset. These reporting signals can be used for error rate estimation and real-time monitoring of the radiation levels in space. Its circuit design is fully automated and implemented only using digital design flow. The design constraints required for implementation of double sampling flip-flops are elaborated on in this work. Implemented in 180 nm bulk CMOS technology and tested at 100 MHz, the double sampling technique effectively detected single event effects, making this technique a promising solution for enhancing spacecraft system reliability in space environments.

1 Introduction

Increasing interest and access to space exploration boost the number of spacecrafts launched each year. The expectations and requirements for advanced data processing and autonomous operations in space are growing annually, certainly with the popularization of artificial intelligence and powerful digital signal processing, available in commercial products. These growing expectations demand accurate and predictable behavior of the on-board electronic control and processing systems in harsh space environments. Over time, system complexity has increased and numerous processing circuits are now integrated on a single system-on-chip (SoC). This also means that microcontrollers, microprocessors, and digital processing units need to increase their processing power and reliability to maintain safe control of the spacecraft. However, the performance scaling of commercial digital electronics cannot always be followed in space applications due to reliability concerns and a performance gap nowadays exists between commercial processors and space grade systems.

On-board computers are affected by high energy and charged particles originating from the sun's solar wind, coronal mass ejection, and other cosmic events. The radiations effects become more prominent with technology scaling and increasing transistor count density. When a single energized particle strikes a circuit, an

unwanted state flip may occur due to charges generated in the logic, leading to a single event transient (SET), which can propagate trough the logic gates of a circuit. When this SET reaches a sequential cell such as a flip-flop and is captured on its active clock edge, the SET fault is stored in that sequential cell, leading to a single event upset (SEU). These upsets can disrupt the configuration and operation of the electronic device, and will propagate to successive sequential cells throughout the circuit, possibly leading to a single event functional interrupt (SEFI). [1]

Over the years different methods have been developed to mitigate the effects of SEUs. At the circuit level, triplication of logic and sequential cells is often used to create redundancy if presumably only one of those cells is affected by a SET or SEU. A voter after those triplicated cells can select the value that finds agreement between two of the three cells, as first proposed by John Von Neumann in 1956. [2]

Although TMR proves to be an effective technique for protecting digital circuits against SEUs, it adds a significant overhead in power consumption, area usage and path delay by more than a factor of three, especially if the data-path is also triplicated to resolve SETs in the logic [3] [4].

Another circuit-level strategy for hardening flip-flops against SEUs is the use of the dual interlocked storage cell (DICE) circuits, where the cell state can be restored by two interleaved pairs of redundant nodes within the cell [5].

TMR and DICE methods aim at detecting and restoring the correct state in a sequential cell after a SEU. This work proposes a method of only detecting arriving SETs and SEU in a flip-flop, and reporting this upset to the system, which can handle appropriately.

2 Methodology

In a flip-flop, two possible faults can occur: either (i) a SET arrives at the D-input of the flip-flop, which is subsequently captured upon the rising clock edge, or (ii) a SEU occurs in the flip-flop itself. In both cases, a fault is introduced in the flip-flop, which possibly leads to malfunctions. As such, our goal is twofold: detect arriving SETs and detect SEUs in the flip-flop itself. The proposed double sampling technique is inspired from well known Razor flip-flops [6].

2.1 Double Sampling Flip-Flop

The method relies on two flip-flops: a main flip-flop and a shadow flip-flop, as shown on Figure 1. The main flip-flop is used in the regular data-path, and determines the overall setup timing of the design. The shadow flip-flop will capture the data slightly later than the main flip-flop. This capture time offset between both flip-flops should be larger than the SET pulse width. As such, when one flip-flop has captured the SET, the other flip-flop will not have captured it. This results in different signals at the output of both flip-flops, which can be detected by an xor-gate. The xor-gate generates an error signal indicating an SEU has occurred in this cell. Furthermore, if an SEU directly occurs in the main flip-flop, it can also be detected by the xor gate.

As shown on Figure 1, only the main flip-flop resides in the data-path, the shadow flip-flop and the xor gate are not. The main advantage of this technique is that the timing of the data path is not affected, thus eliminating the performance penalty found with TMR and DICE techniques [7].

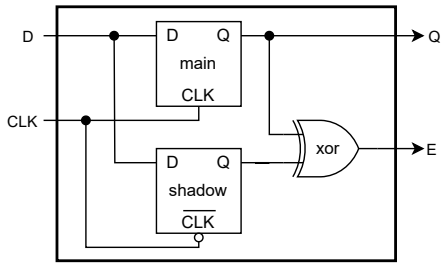

Figure 1: Schematic of a double sampling flip-flop

The waveform of the double sampling flip-flop (DSFF) is illustrated in Figure 2.

In Cycle 1, normal operation is observed with the data input low. Both the main and shadow flip-flops store the same value, resulting in a low XOR comparison output.

In Cycle 2, a SET event occurs near the rising clock edge. While the main flip-flop captures incorrect data, the shadow flip-flop

correctly records the input. This discrepancy is detected by the xor gate, which flags the error in the following clock cycle, causing the error signal (E) to be generated.
In Cycle 3, a SEU affects the main flip-flop, causing its output (Q main) to go low, while the shadow flip-flop remains unaffected. As in the previous case, the XOR gate detects the inconsistency.
Finally, Cycle 4 demonstrates normal operation again, where the error signal returns to low.

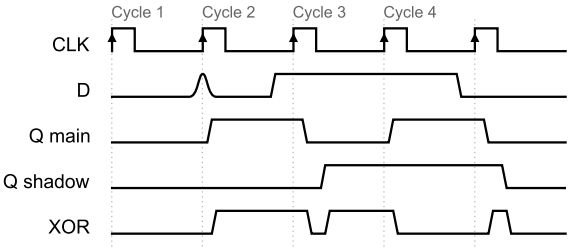

Figure 2: Waveform of the double sampling flip-flop detecting single event upsets

2.2 Clock Generator

In the implemented DSFF, we utilized a positive-edge-triggered main flip-flop and a negative-edge-triggered shadow flip-flop. This design allows for adjustable capture delay between the two flip-flops by modifying the duty cycle of the clock signal. A clock generator, incorporating taps on a delay line (as shown in Figure 3), enables the generation of varying duty cycles to fine-tune this delay.

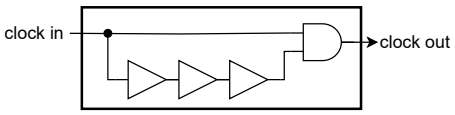

Figure 3: Schematic of a clock generator with a fixed duty-cycle

2.3 Implementation

The double-sampling technique can be seamlessly integrated into a fully digital design by replacing each sensitive flip-flop with a DSFF. However, incorporating DSFFs requires careful timing analysis both within the DSFF cell and across multiple cells.
To ensure proper operation, multi-cycle path timing constraints must be applied to the shadow flip-flop in all DSFFs. These constraints assure proper setup and hold requirements, as verified by the static timing analysis tool. Figure 4 illustrates the setup and hold constraints required for the DSFF.

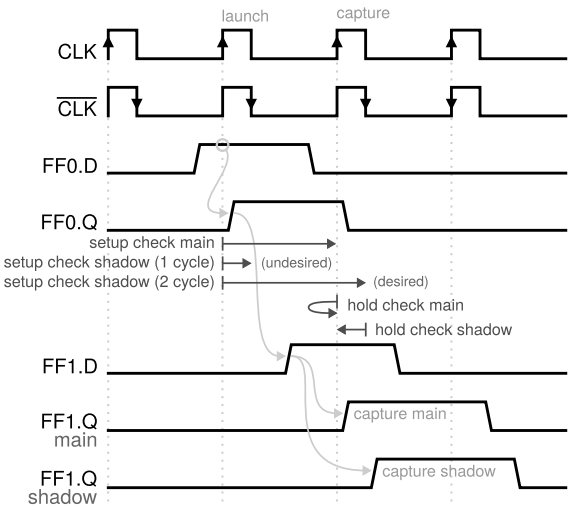

Figure 4: Setup and hold timing with double sampling flip-flops

Without proper constraints, the setup check for shadow FF1 incorrectly expects new data (FF1.D) at the negative clock edge immediately following the launching positive clock edge from FF0.Q. This is undesirable because the shadow flip-flop captures in this case the previously launched data one full clock cycle too early.
To correct this, a multi-cycle constraint can be applied to the shadow flip-flop, enforcing a two-cycle multi-cycle path for the timing check. This shifts the capturing edge one clock cycle later, ensuring that the new data arrives at the correct time. As a result, the hold check for the shadow flip-flop extends back to the previous rising edge, inherently satisfying the hold constraint.
In a specific use case, the following Synopsys Design Constraints (SDC) may apply:

```
create_clock -name clk
             -period 10ns
             -waveform {0 0.5ns}
             [get_pin clockGen/out]
set_multicycle_path 2
             -to [get_pin *shadow*/Q]
```

In this example, a 100 MHz clock with a 500 ps active-high pulse width is constrained, allowing SETs up to 500 ps to be detected. Additionally, a multi-cycle path is defined

for the shadow flip-flop. These constraints provide all the necessary information for the synthesis tool to accurately implement the timing of the module, ensuring correct operation.

3 Results

The proposed technique of using DSFFs to detect SETs and SEUs was tested with both RTL simulations and verified in silicon on 180 nm bulk cmos technology at 100 MHz. The test chip contains a large scan chain of DSFFs, combined with a representative data-path in between. The test setup stimulates the scan chain with random data and monitors the output and error flags of the flip-flops to check if all errors are correctly flagged. The chip was tested with heavy ion and two-photon absorption stimulation to determine the sensitivity of the technology and determine the effectiveness of the DSFF technique. The two photon absorption technique additionally provided spatial insight in the sensitivity of the cells.

Using a 1271 MeV Xe ion source with a linear energy transfer of 62.5 MeV·cm²/mg, about 5% of the SEUs were not reported. When we compared this with the result of two photon absorption testing, all induced errors (originating from both SEUs and SETs) were successfully reported. This is because the heavy ion source also stimulated the unprotected I/O pads (like clock and data), where the undetected SEUs were likely generated.

4 Future Work

We are working within the reserach group on integrating the double sampling flip-flops in a processor system. We use the open-source 32-bit RISC-V Ibex processor core which was initially designed by ETH Zurich and the University of Bologna, and now maintained by lowRISC. This Ibex core has a two-stage pipeline with an additional third write-back stage. [8]

To integrate the double sampling flip-flops in the processor pipeline, we follow the next steps which are depicted in Figure 5. First we take the HDL code of the Ibex

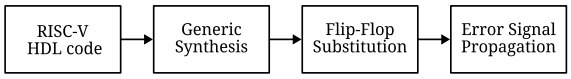

Figure 5: Implementation of double sampling flip-flops in a RISC-V processor

RISC-V core written in SystemVerilog, and perform elaboration or generic synthesis on the design. This allows for extracting a list of all flip-flops in the processor core. From this list of flip-flops we can can select the instances we want to substitute with the double sampling flip-flops. Not all flip-flops need to be substituted depending on the level of detection and protection wanted in the processor. One could substitue all synchronization flip-flops in the pipeline after each stage, or only substitue a smaller selection. After substitution of the selected flip-flops with the double-sampling variant, we propagate the error signals to the interrupt controller of the Ibex core. We use a compression tree with or-gates to get one error interrupt signal from all double sampling flip-flops for each stage. You can see a simplified example block diagram of a two-stage pipeline processor such as the Ibex RISC-V core in Figure 6.

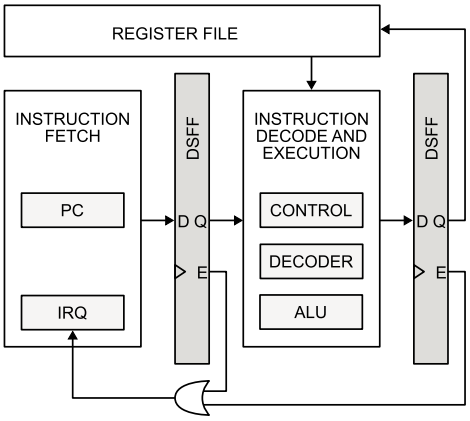

Figure 6: RISC-V two-stage pipeline with EDFFs connected to the interrupt controller.

This would allow for the interrupt handler to perform roll-back of the internal state of the processor, if the processor periodically stores the state of the registers in RAM. Roll-back allows re-execution of the instructions from a well-known state after a SEU was detected in the pipeline.

Pseudo-code for such roll-back scenario is provided below:

```
void checkpoint(void) {
    store_pc();
    store_regs();
}

void irq_handle_seu(void) {
    load_regs();
    load_pc();
}

int main(void) {
    // do work
    checkpoint();
    // do work
    checkpoint();
    ...
}
```

After each block of important work performed in the code, a `checkpoint` function will store the program counter `pc` and all registers `regs` to RAM. When an SEU is detected by the double sampling flip-flops, the interrupt request handler function `irq_handle_seu` will reload all previously saved registers and restore the program counter. If the interrupt request handler function is not called, the checkpoint function will overwrite the values in RAM.

5 Conclusion

This work shows a fully automated detection method using a double sampling based flip-flop, which is able to detect SEUs and SETs. Integrating these double sampling flip-flops into a digital flow is easy, and even timing critical paths can be replaced by these flip-flops without introducing additional delays in the data-path. All detected errors must be handled at the system level, for instance through an interrupt, which can be further handled in software. In a practical implementation, the different error sources can be further separated or grouped to assign different importance to each group. Furthermore, the circuit provides real-time monitoring capabilities to understand and further predict the error rates in a system.

References

[1] V. Ferlet-Cavrois, L. W. Massengill, and P. Gouker, "Single event transients in digital cmos—a review," *IEEE Transactions on Nuclear Science*, vol. 60, no. 3, pp. 1767–1790, 2013. DOI: 10.1109/TNS.2013.2255624.

[2] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata studies*, vol. 34, no. 34, pp. 43–98, 1956.

[3] D. G. Mavis and P. H. Eaton, "Soft error rate mitigation techniques for modern microcircuits," in *2002 IEEE International Reliability Physics Symposium. Proceedings. 40th Annual (Cat. No. 02CH37320)*, IEEE, 2002, pp. 216–225.

[4] P. Balasubramanian and N. E. Mastorakis, "Power, delay and area comparisons of majority voters relevant to tmr architectures," *arXiv preprint arXiv:1603.07964*, 2016.

[5] F. Mori, M. Ebara, Y. Tsukita, J. Furuta, and K. Kobayashi, "Intrinsic vulnerability to soft errors and a mitigation technique by layout optimization on dice flip flops in a 65-nm bulk process," *IEEE Transactions on Nuclear Science*, vol. 68, no. 8, pp. 1727–1735, 2021.

[6] D. Ernst, N. S. Kim, S. Das, *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.*, IEEE, 2003, pp. 7–18.

[7] K. Appels, R. Weigand, W. Dehaene, and J. Prinzie, "In-situ single-event effects detection in 22 nm fdsoi flip-flops," *IEEE Transactions on Nuclear Science*, 2024.

[8] *Ibex documentation*. [Online]. Available: https://ibex-core.readthedocs.io/en/latest.