IECON 2013

39th Annual Conference of the IEEE Industrial Electronics Society

Vienna, November 11–13, 2013

Mitigation of Single Event Upsets in the Control Logic of a Charge Equalizer for Li-ion Batteries

F. Baronti, C. Bernardeschi, L. Cassano, A. Domenici, R. Roncella, R. Saletti

Dipartimento di ingegneria dell'informazione, Università di Pisa

Overview

- A Battery Management System has been designed and implemented.
- Charge equalization is controlled by a CPLD.
- Two *ad hoc* fault-tolerant designs have been produced to mitigate the effects of SEUs in the CPLD.
- The two designs have been simulated and evaluated wrt to the TMR approach.

Charge equalization for Li-ion batteries

Lithium-ion batteries are a promising solution for energy storage in many industrial applications, such as electric transportation and smart grids.

Li-ion batteries are very sensitive to overcharge, deep discharge and operation outside the specified temperature range.

A Battery Management System (BMS) is required to guarantee the safe and effective operation of the battery.

In particular, the BMS must keep a balanced *State of Charge* among the battery cells.

Faults in programmable devices

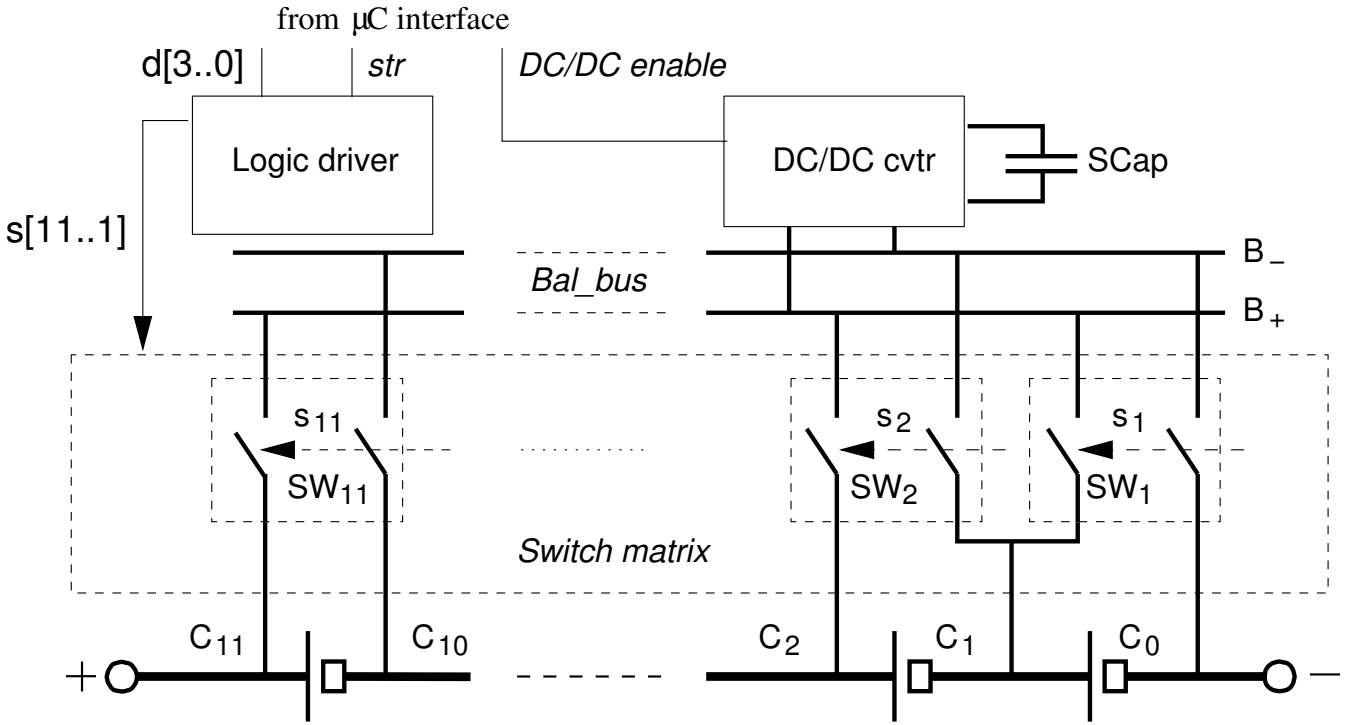
Programmable Logic Devices (PLD), such as CPLDs and FPGAs, are cost-effective building blocks for the control logic of a BMS.

However, they are subject to radiation-induced faults, called *Single Event Upsets* (SEU), which may alter their behavior.

Since a BMS is a safety-critical component, it is necessary to adopt appropriate fault-tolerance techniques to improve its reliability.

In this work, we analyze by simulation different fault-tolerant designs for the control logic of a charge equalizer.

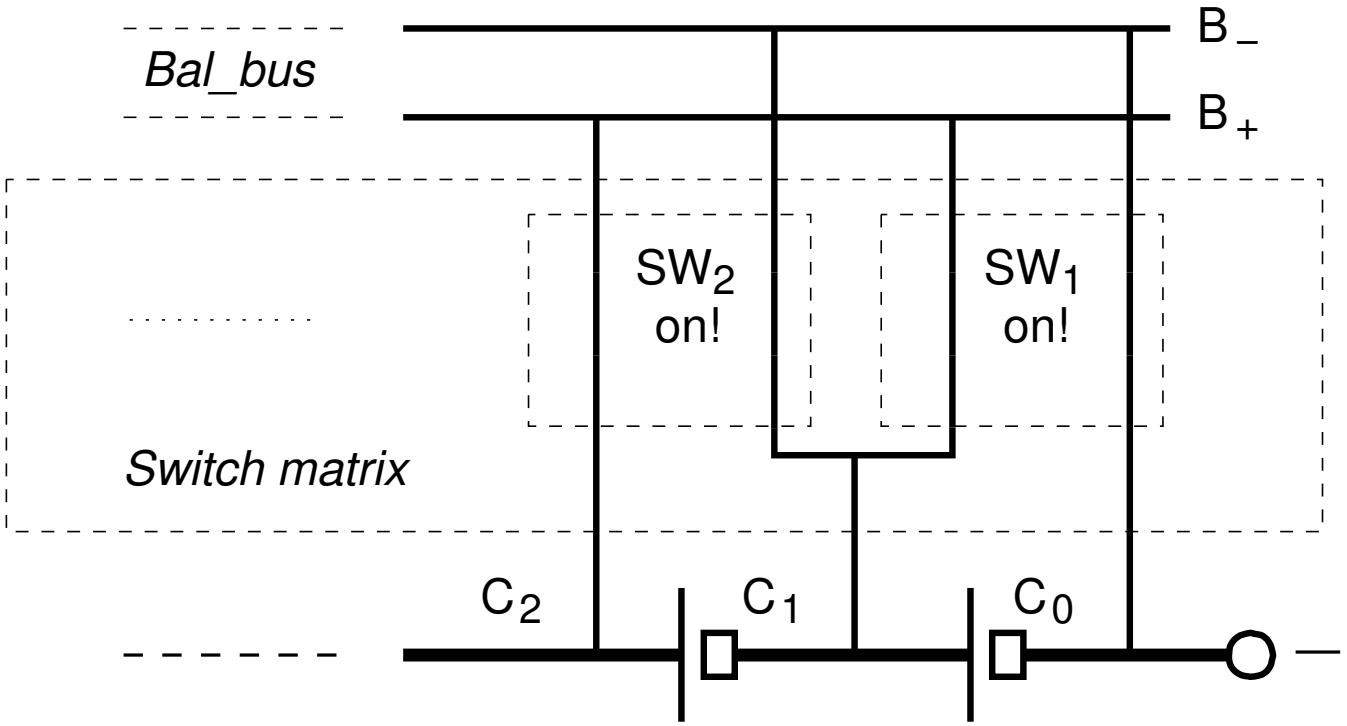
Architecture of the charge equalizer



Within a battery pack, a DC/DC converter moves charge from one cell to another to achieve charge equalization by *active balancing*.

The d[3..0] signals from the microcontroller encode the required switch configuration s[11..1], computed by the *logic driver*.

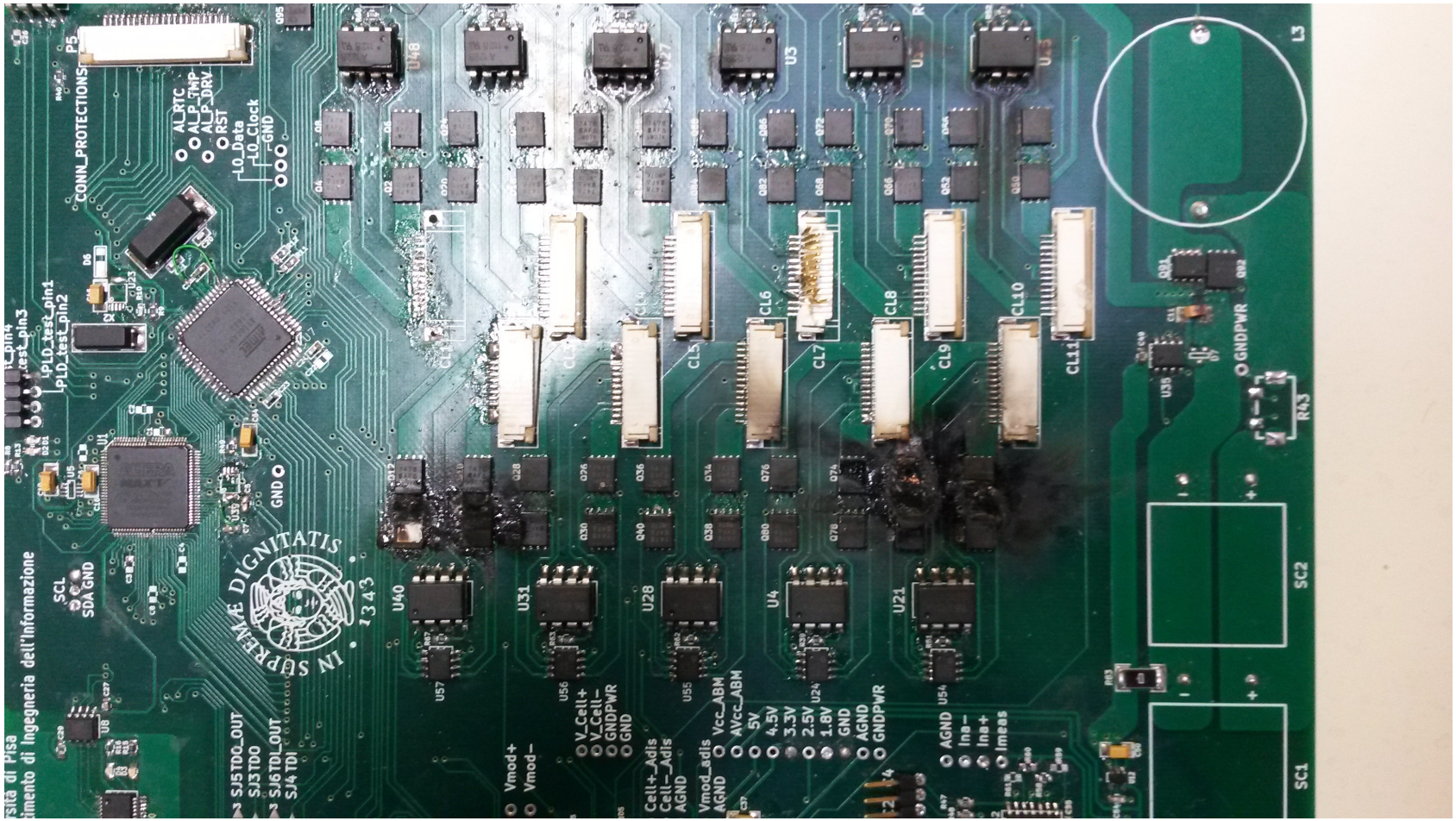
Main safety requirement



To avoid short circuits, only one DPST switch may be conducting at any time.

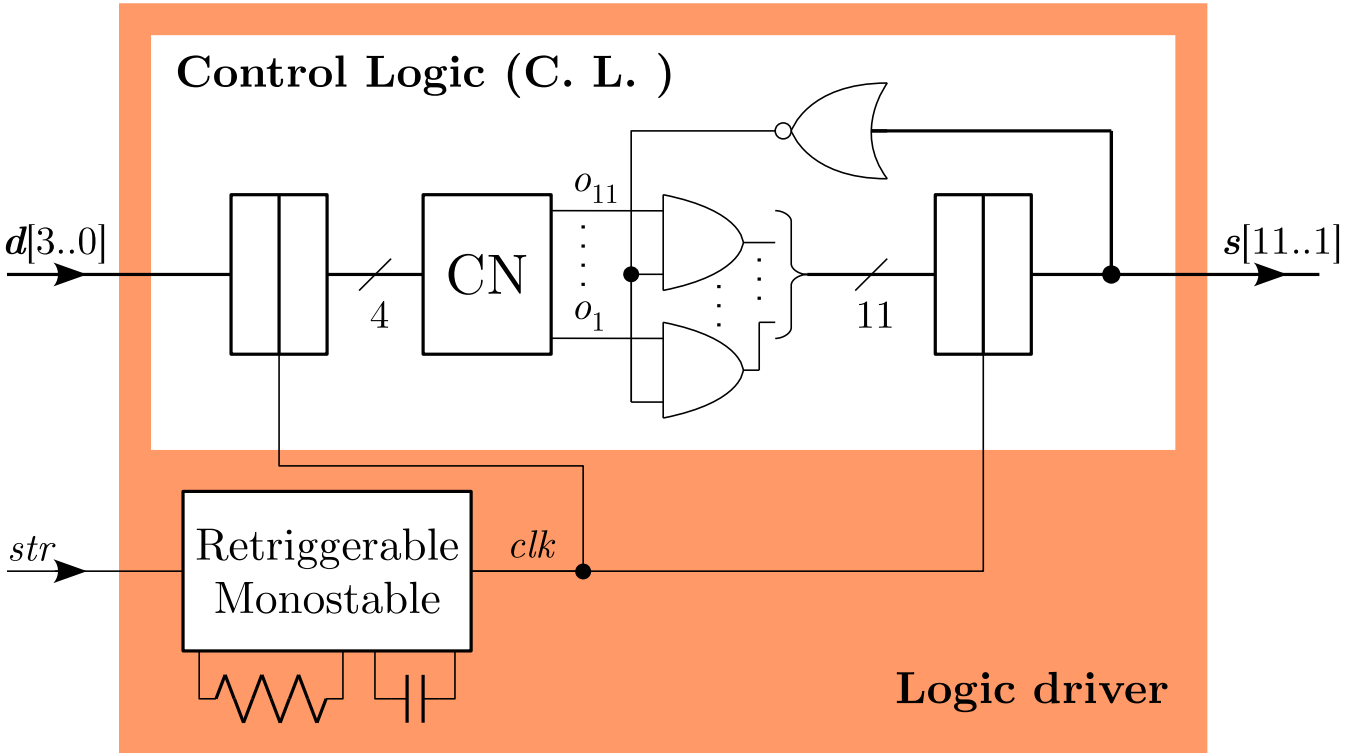
Thus, at most one of the s[11..1] signals may be asserted at any time.

Or else...


CONN_PROTECTIONS
ALARM_RTC
ALARM_TMP
ALARM_DRV
RST
LO_Data
LO_Clock
GND
PLD_test_pin1
PLD_test_pin2
SCL
SDA GND
SJ5TDO_OUT
SJ3TDO
SJ6TDI_OUT
SJ4 TDI
Vmod+
Vmod-
V_Cell+
V_Cell-
GNDPWR
GND
Cell+_Adis
Cell-_Adis
AGND
Vmod_adis
AGND
Vcc_ABM
AVcc_ABM
5V
4.5V
3.3V
2.5V
1.8V
GND
AGND
GNDPWR
AGND
Ina-
Ina+
Imeas
GNDPWR
SC1
SC2
IN SUPREMÆ DIGNITATIS
1343

Zap!

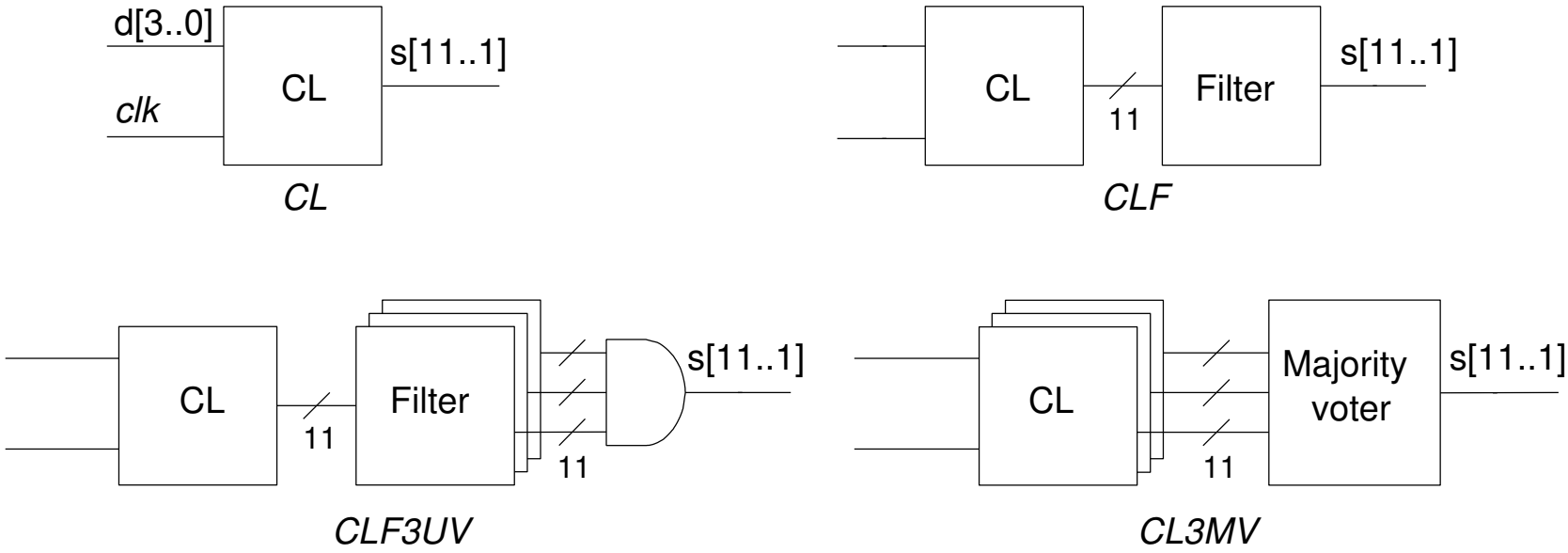
Block diagram of the logic driver



The logic driver ensures that configurations leading to short circuits are not possible.

The logic driver is a *safety critical* component.

Simulated fault-tolerance techniques



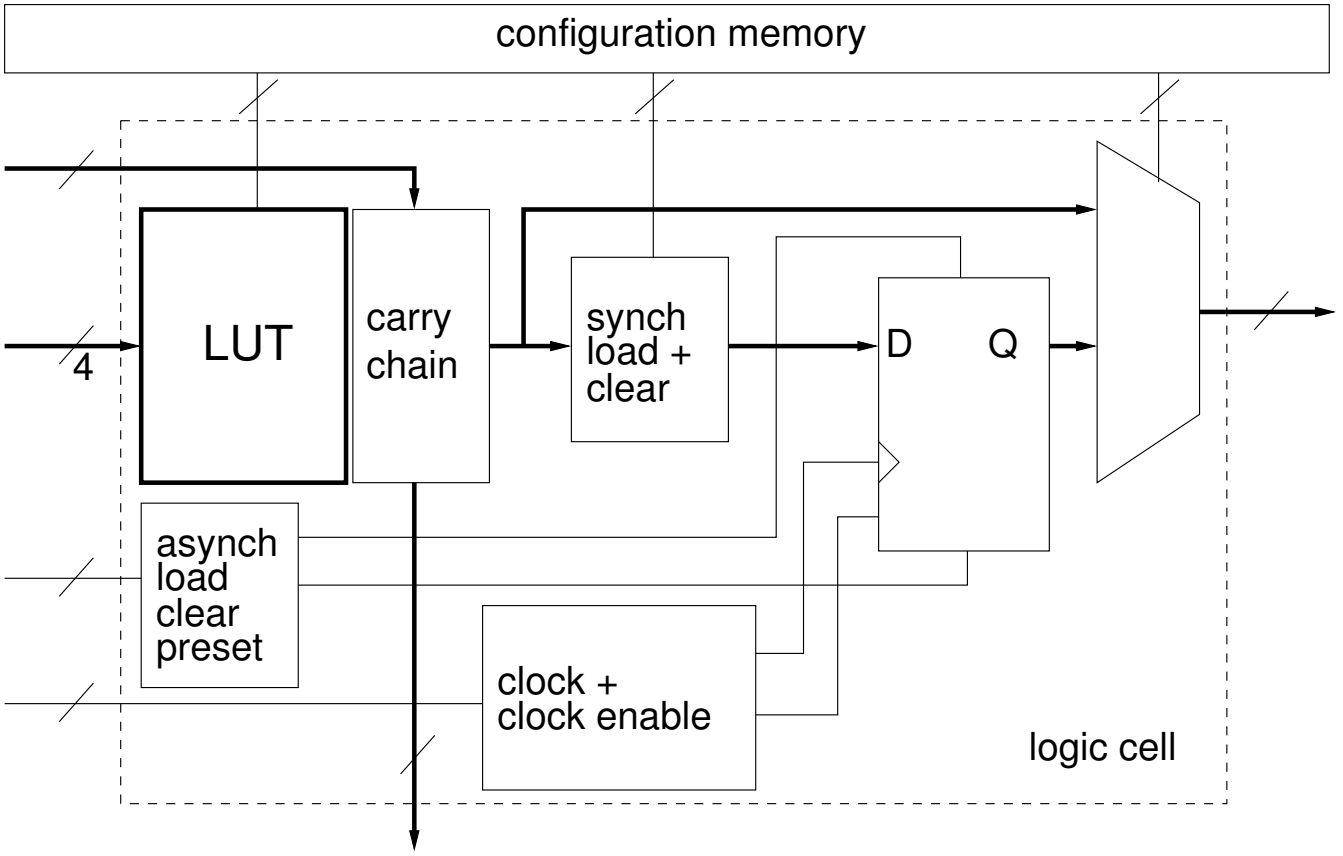
CL: the non-hardened design.

CLF: the control logic followed by the Filter block, which blocks all commands with more than one closed switch.

CLF3UV: the control logic followed by the triplicated Filter and a unanimity voter (AND gate).

CL3MV: the classical TMR technique.

A logic cell



A simplified view of an Altera Max V logic cell.

Implementation parameters

	LCs	FFs	LUTs	LUT-FF	LUTs + LUT-FF
CL	26	4	11	11	22
CLF	45	4	30	11	41
CL3MV	89	12	44	33	77
CLF3UV	94	4	79	11	90

LC: Overall number of required logic cells, divided in:

LUTs: LCs using only LUTs;
FFs: LCs using only flip-flops;
LUT-FF: LCs using both LUTs and flip-flops.

Smallest Altera MAX V CPLDs

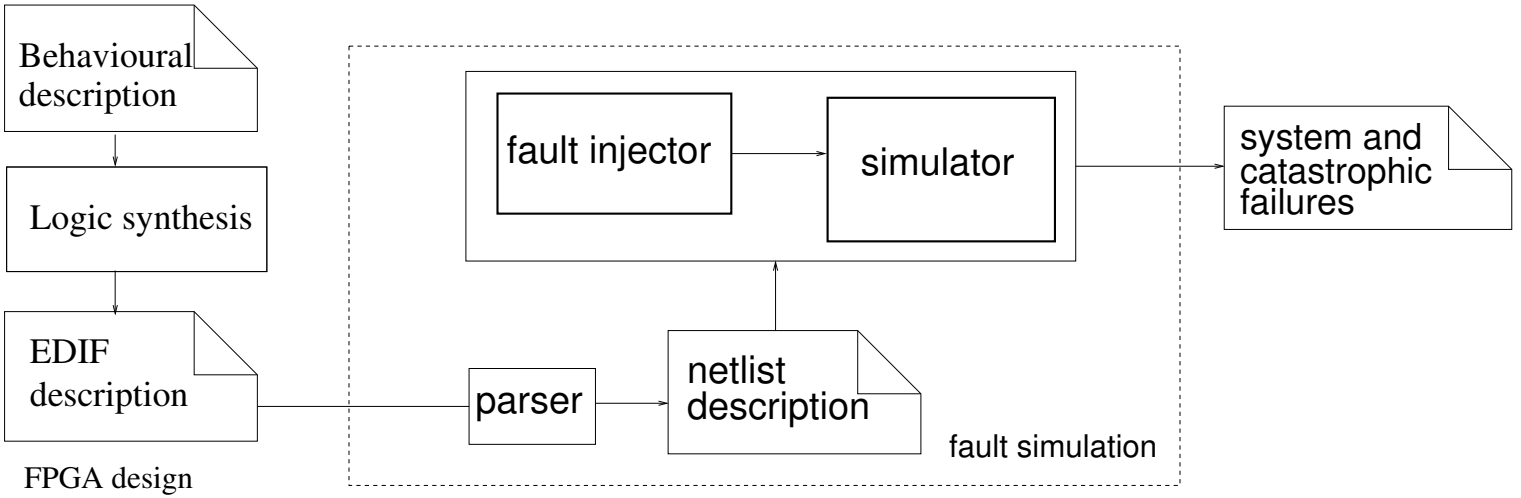
Feature	5M40Z	5M80Z	5M160Z
Logic Cells (LCs)	40	80	160
Typical equivalent macrocells	32	64	128
Maximum user I/O pins*	54	54	54
Accommodates FT schemas	no	CLF	CLF3UV

* 64-Pin EQFP

The 5M40Z device is sufficient for the basic (non-FT) design, but the 5M80Z and 5M160Z are needed for the FT designs.

However, the three devices are all available in the 64-Pin EQFP package: FT designs do not require board-level redesign.

Simulation environment



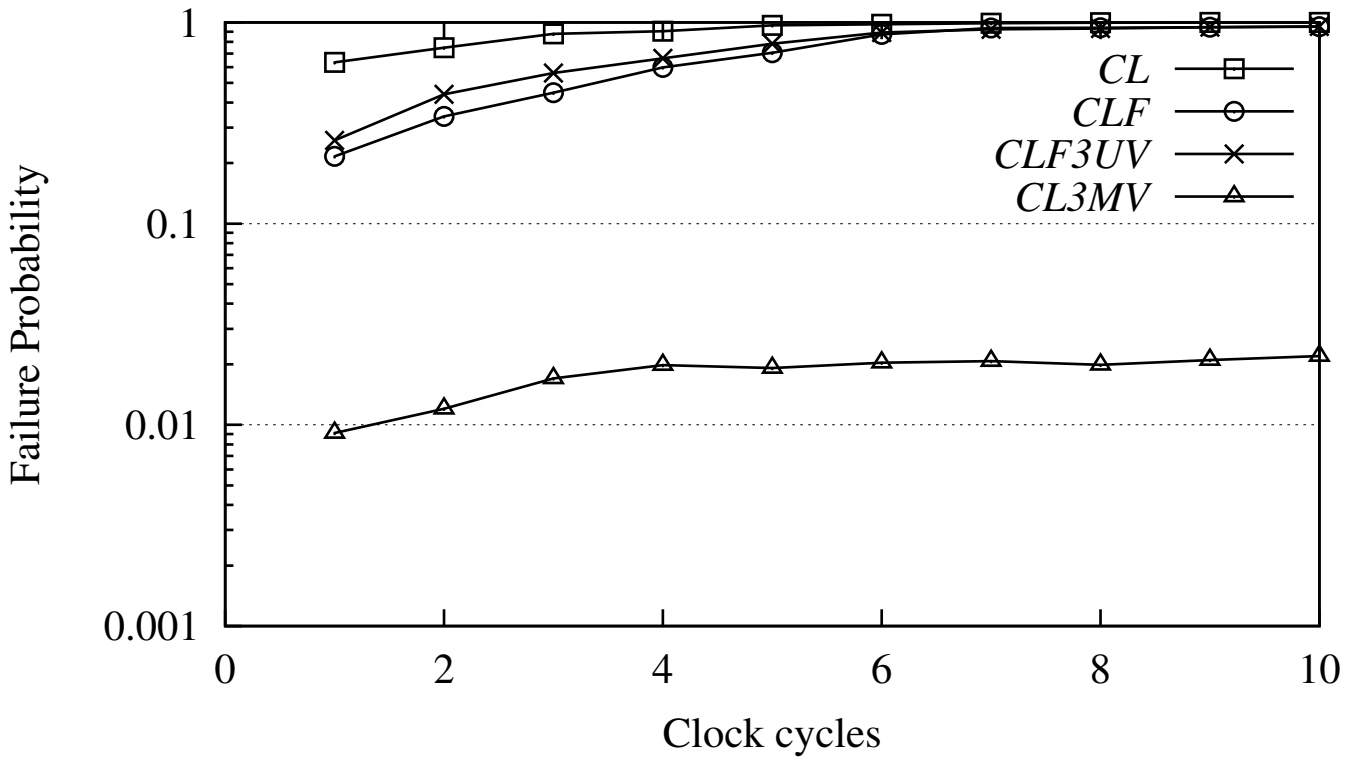
Each CL implementation has been simulated with sequences of random test vectors i clock cycles long, with i from 1 to 10.

For each value of i, 10000 simulation runs have been performed.

At each simulation run, a fault has been injected into a random location of the configuration memory.

System failures (F_i): only affect BMS performance.
Catastrophic failures (CF_i): cause physical damage.

System failure probability

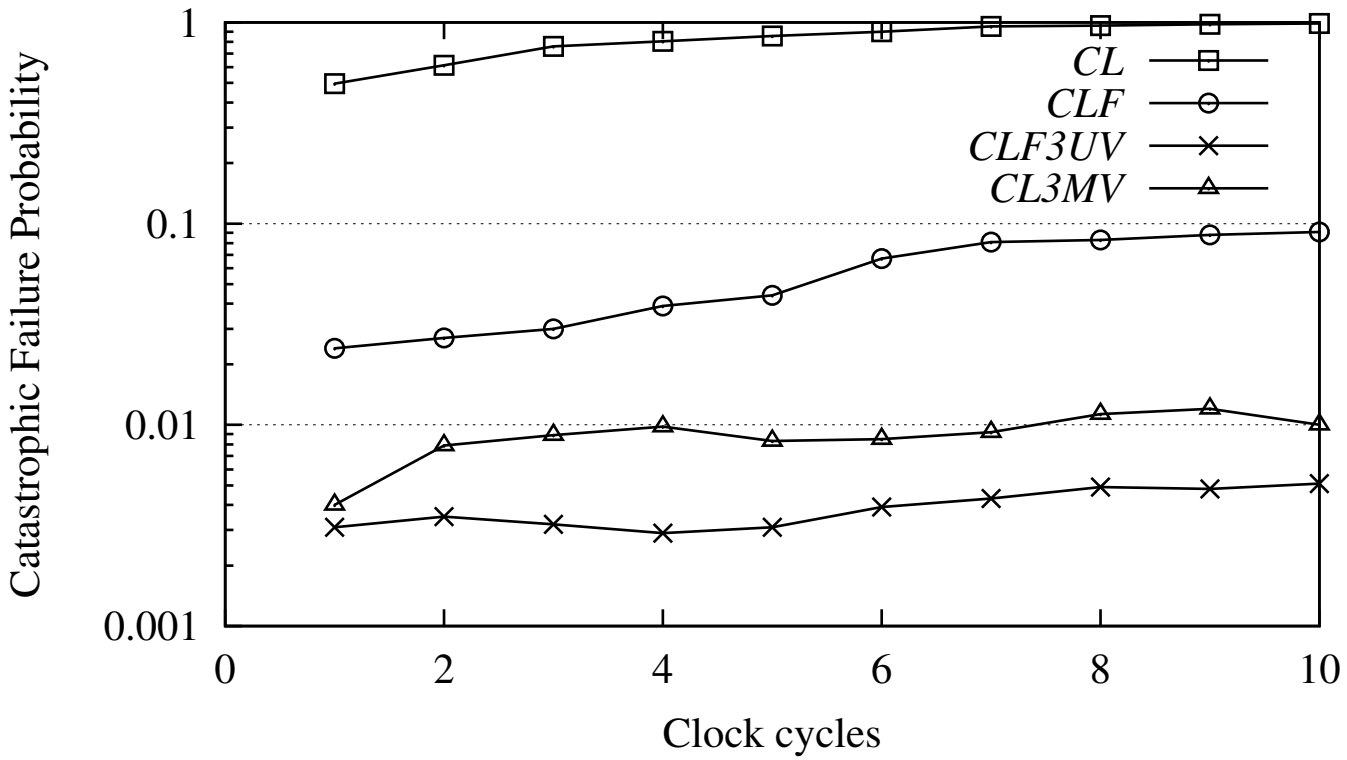


Failure probability as a function of test vector sequence length i:

$$\mathrm{FP}_i = \mathrm{F}_i / 10000$$

Triple modular redundance achieves the best results.

Catastrophic failure probability



$$\mathrm{CFP}_i = \mathrm{CF}_i / 10000$$

Triple filtering achieves the best results.

Single filtering has an intermediate performance.

Summary of simulation results

	Ovhd	FP_{10}	CFP_{10}
CL	1.0	1.00	1.000
CLF	1.7	0.95	0.090
CL3MV	3.4	0.02	0.010
CLF3UV	3.6	0.96	0.005

Ovhd: Area overhead wrt bare non fault-tolerant design.

FP_{10}: Failure probability with 10 clock cycles.

CFP_{10}: Catastrophic failure probability with 10 clock cycles.

The CLF3UV design is more effective than the CL3MV, with a small increase of area overhead.

Conclusions

Two design-specific SEU mitigation techniques for a BMS have been designed and simulated.

The two designs have been assessed for effectiveness, taking TMR as a reference.

The TMR approach is most effective at reducing the probability of *system* failures, i.e., affecting only BMS performance.

The design-specific solution using the triplicated safety filter is most effective at reducing the probability of *catastrophic* failures, i.e., leading to physical damage due to short-circuits.

The implementation of these mitigation techniques can be achieved by a moderate increase of the used logic resources.

In particular, their implementations fit on the same package as the non-mitigated design, thus requiring no change to the original board layout.

Thank you

Danke schön