TaMaRa:
An automated triple modular redundancy EDA flow for Yosys

Matt Young

31 July 2024

University of Queensland

Table of contents

1. Background
2. TaMaRa methodology
3. Current status & future
4. Conclusion

Background

About me

TODO photo

Matt Young (he/him), 21 years old from Brisbane, Australia.

Graduated Bachelor of Computer Science earlier in 2024 from the University of Queensland.

Currently studying Bachelor of Computer Science (Honours) at UQ, which includes a one year research thesis.

Passionate about digital hardware design, embedded systems, high performance/low-level software/hardware. Looking to in future take up a PhD, and eventually research/work in the area of CPU/GPU/ASIC design, or FPGAs, or similar.

Motivation

Fault tolerant computing is important for safety critical sectors (aerospace, defence, medicine, etc.)

Motivation

Fault tolerant computing is important for safety critical sectors (aerospace, defence, medicine, etc.)

For space-based applications, Single Event Upsets (SEUs) are very common

- Must be mitigated to prevent catastrophic failure
- Caused by ionising radiation striking transistors on a digital circuit

Motivation

Fault tolerant computing is important for safety critical sectors (aerospace, defence, medicine, etc.)

For space-based applications, Single Event Upsets (SEUs) are very common

- Must be mitigated to prevent catastrophic failure
- Caused by ionising radiation striking transistors on a digital circuit

Even in terrestrial applications, SEUs can still occur

- Must be mitigated for high reliability applications

Motivation

Fault tolerant computing is important for safety critical sectors (aerospace, defence, medicine, etc.)

For space-based applications, Single Event Upsets (SEUs) are very common

- Must be mitigated to prevent catastrophic failure
- Caused by ionising radiation striking transistors on a digital circuit

Even in terrestrial applications, SEUs can still occur

- Must be mitigated for high reliability applications

ASICs and FPGAs commonly deployed in space (and on Earth)…

Motivation

Fault tolerant computing is important for safety critical sectors (aerospace, defence, medicine, etc.)

For space-based applications, Single Event Upsets (SEUs) are very common

- Must be mitigated to prevent catastrophic failure
- Caused by ionising radiation striking transistors on a digital circuit

Even in terrestrial applications, SEUs can still occur

- Must be mitigated for high reliability applications

ASICs and FPGAs commonly deployed in space (and on Earth)… but protection from SEUs remains expensive!

Triple Modular Redundancy

TODO describe TMR

Triple Modular Redundancy

TMR can be added manually...

Triple Modular Redundancy

TMR can be added manually…

but this is another time consuming and error prone step in the *already* complex design process.

Triple Modular Redundancy

TMR can be added manually…

but this is another time consuming and error prone step in the *already* complex design process.

Let's automate it!

TaMaRa methodology

Concept

Implement TMR as a pass in an EDA synthesis tool.

- Integrated with the rest of the flow, easy to use
- Fully automated

Concept

Implement TMR as a pass in an EDA synthesis tool.

- Integrated with the rest of the flow, easy to use
- Fully automated

Goal: Pick any design, of any complexity, “press a button” and have it be rad-hardened.

Concept

Implement TMR as a pass in an EDA synthesis tool.

- Integrated with the rest of the flow, easy to use
- Fully automated

Goal: Pick any design, of any complexity, “press a button” and have it be rad-hardened.

Yosys [1] is the best (and the only) open-source, research grade EDA synthesis tool.

Concept

Implement TMR as a pass in an EDA synthesis tool.

- Integrated with the rest of the flow, easy to use
- Fully automated

Goal: Pick any design, of any complexity, “press a button” and have it be rad-hardened.

Yosys [1] is the best (and the only) open-source, research grade EDA synthesis tool.

- Proprietary vendor tools (Synopsys, Cadence, Xilinx, etc) immediately ignored as they can’t be extended

Existing works

Very important prior work done by J. M. Johnson and M. J. Wirthlin [2] at BYU.

Existing works

Very important prior work done by J. M. Johnson and M. J. Wirthlin [2] at BYU.

TODO algorithm

The TaMaRa algorithm

TODO

Technical implementation

Currently implemented in C++20, using CMake and Ninja build tools.

Technical implementation

Currently implemented in C++20, using CMake and Ninja build tools.

Load into Yosys: `plugin -i libtamara.so`

Technical implementation

Currently implemented in C++20, using CMake and Ninja build tools.

Load into Yosys: `plugin -i libtamara.so`

TMR is implemented as two separate commands: `tmr` and `tmr_finalise`

Technical implementation

Currently implemented in C++20, using CMake and Ninja build tools.

Load into Yosys: `plugin -i libtamara.so`

TMR is implemented as two separate commands: `tmr` and `tmr_finalise`

Run `tmr` after synthesis, but before techmapping.

Technical implementation

Currently implemented in C++20, using CMake and Ninja build tools.

Load into Yosys: `plugin -i libtamara.so`

TMR is implemented as two separate commands: `tmr` and `tmr_finalise`

Run `tmr` after synthesis, but before techmapping.

Run `tmr_finalise` just before techmapping (ensuring no more optimisation passes will run).

Current status & future

Current status

TODO

Mostly focused around literature reviews, scoping out the problem, formulating requirements, etc.

That being said, I also have a skeleton Yosys plugin loading.

The future

This will be implemented for my Honours thesis over the next 1 year.

- Honours is kind of of like mini masters, it's an Australia-specific thing.
- Supervised by Assoc. Prof. John Williams (former PetaLogix, Xilinx)

The future

This will be implemented for my Honours thesis over the next 1 year.

- Honours is kind of of like mini masters, it's an Australia-specific thing.
- Supervised by Assoc. Prof. John Williams (former PetaLogix, Xilinx)

Ideally, TaMaRa will be released open-source under the MPL 2.0.

- Pending university IP shenanigans, but there is a good chance of being allowed to open-source it.

Conclusion

Summary

TODO

Bibliography

[1] D. Shah, E. Hung, C. Wolf, S. Bazanski, D. Gisselquist, and M. Milanovic, "Yosys+nextpnr: an Open Source Framework from Verilog to Bitstream for Commercial FPGAs," *CoRR*, 2019, [Online]. Available: http://arxiv.org/abs/1903.10407

[2] J. M. Johnson and M. J. Wirthlin, "Voter insertion algorithms for FPGA designs using triple modular redundancy," in *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*, in FPGA '10. ACM, Feb. 2010. doi: 10.1145/1723112.1723154.

Any questions?