Technische Universität Braunschweig

Chair for Chip Design for Embedded Computing

FPGA Accelerated Post-Synthesis Fault Injection for RISC-V Cores

Eike Trumann, Jasper Homann, and Guillermo Payá-Vayá

{eike.trumann,jasper.homann,g.paya-vaya}@tu-braunschweig.de

Introduction

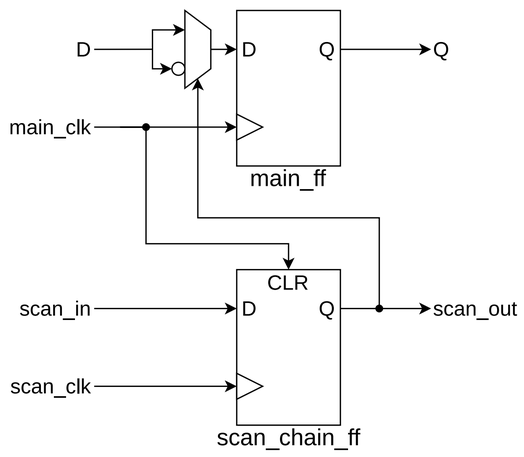
- ***Processor cores*** for space and other critical environments might suffer from radiation effects
- Processor cores exhibit different failure modes such as halts, erroneous control flow, wrong results
- Bit errors can arise from single particle hits
- Irradiation experiments are expensive and time-consuming
- Single critical bits might be missed out in experiments
- **Fault Injection** (FI) *simulation* might use long execution times, limiting possible coverage
- Different FPGA-based acceleration solutions exist
- Full fault space coverage might reveal critical components
- ***Contributions of this work:***
  1. FPGA-based FI-Acceleration using scan chains
  2. Automatic test campaign execution using FPGA-SoC
  3. `OpenFI4ASIC` tool based on our previous `FLINT` tool [1]

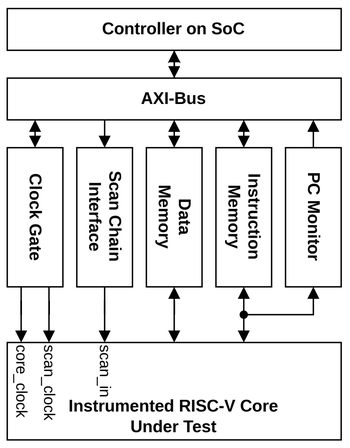
State of the Art: FPGA-based FI Tools

Method	Tools
Scan-Chain	FLINT[1]
Partial Reconfiguration	FT-UNSHADES2[1], ACME[2], SPFFI[3]

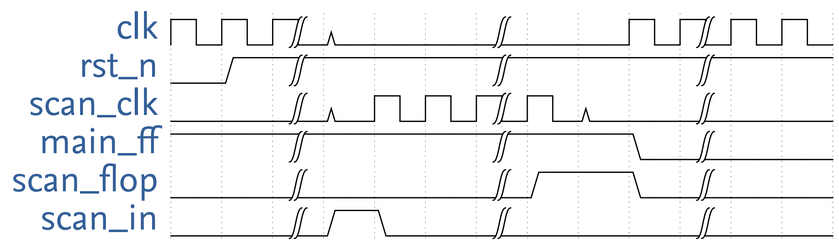
[1] https://ieeexplore.ieee.org/document/6131392 [2] https://ieeexplore.ieee.org/document/8826250
[3] https://www.academia.edu/637875/SPFFI_Simple_Portable_FPGA_Fault_Injector

Open-Source Tool: `OpenFI4ASIC`





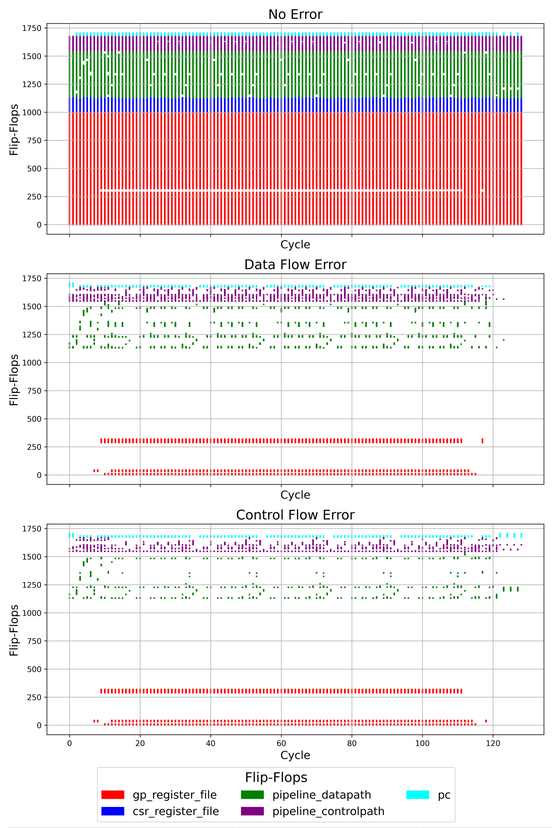
- Fault Instrumentation Tool
  - Scan chain insertion for post-synthesis Verilog netlists
  - Independent of library used
  - Independent of fault model (configurable number of bits depending on number of supported fault models)
    - Stuck-0, stuck-1 model possible [1]
- FPGA Fault Emulation Runtime
  - Control and fault classification on FPGA-SoC
  - Faster FI due to autoreset of instrumentation



Exemplary Use Case: Custom EIS-V Processor

- Processor: *EIS-V*
  - RISC-V (RV32I M-Mode)
  - 5 Stage Pipelined
  - Separate memory interfaces for IMEM and DMEM (Harvard Arch.)
  - Synthesized for NangateOpenCellLibrary using Cadence Genus
  - 1706 Flip Flops
- Sample Application: *Iterative Fibonacci*
  - All variables stored on stack
  - No elided memory accesses (all variables `volatile`)
    - → Intensively tests memory operations and forwarding
  - Program length: 37 Instructions
  - Runtime for $fib(5) = 8$: 130 Cycles

Fault to Error Rate



Category	Data Flow	Control Flow	Both Errors	No Error (Masked)
All	6.21 %	4.09 %	3.84 %	93.54 %
gp_register_file	4.44 %	4.16 %	4.14 %	95.54 %
csr_register_file	0.00 %	0.00 %	0.00 %	100.00 %
pipeline_datapath	7.41 %	2.47 %	2.06 %	92.19 %
pipeline_controlpath	17.48 %	7.24 %	6.62 %	81.90 %
pc	27.91 %	28.29 %	23.47 %	67.27 %

Speedup Emulation vs Simulation

Method	#FFs	#Cycles	#FIs	Runtime	Time/FI	Speedup
Simulation Questasim	100	100	10,000	243.985 s	24 ms/FI	1x
	1706	130	221,780	5411 s (est.)	24 ms/FI	1x
Emulation Zynq 7000	100	100	10,000	1.661 s	0.166 ms/FI	146.89x
	1706	130	221,780	43.418 s	0.196 ms/FI	124.63x

FPGA Resources' Overhead

	LUTs	LUT Utilization	Flip-Flops	Flip-Flops Utilization	Core Clock Frequency	Scan Clock Frequency
RISC-V Core	2873	5.40 %	1706	1.60 %	66.6 MHz	-
Instrumented RISC-V Core	4341	8.16 %	3413	3.21 %	66.6 MHz	400 MHz

Conclusion

- Netlist-level fault injection
- Open-source tool `OpenFI4ASIC`
- Fast control and data analysis using Zynq-7000 FPGA-SoC

https://github.com/tubs-eis/OpenFI4ASIC

[1] R. Nowosielski, L. Gerlach, S. Bieband, G. Paya-Vaya, and H. Blume, FLINT: Layout-oriented FPGA-based methodology for fault tolerant ASIC design, Proceedings - Design, Automation and Test in Europe, DATE, vol. 2015-April, pp. 297300, Apr. 2015, doi: 10.7873/DATE.2015.0278.

RISC-V in Space Workshop, 2-3 April 2025, Gothenburg, Sweden