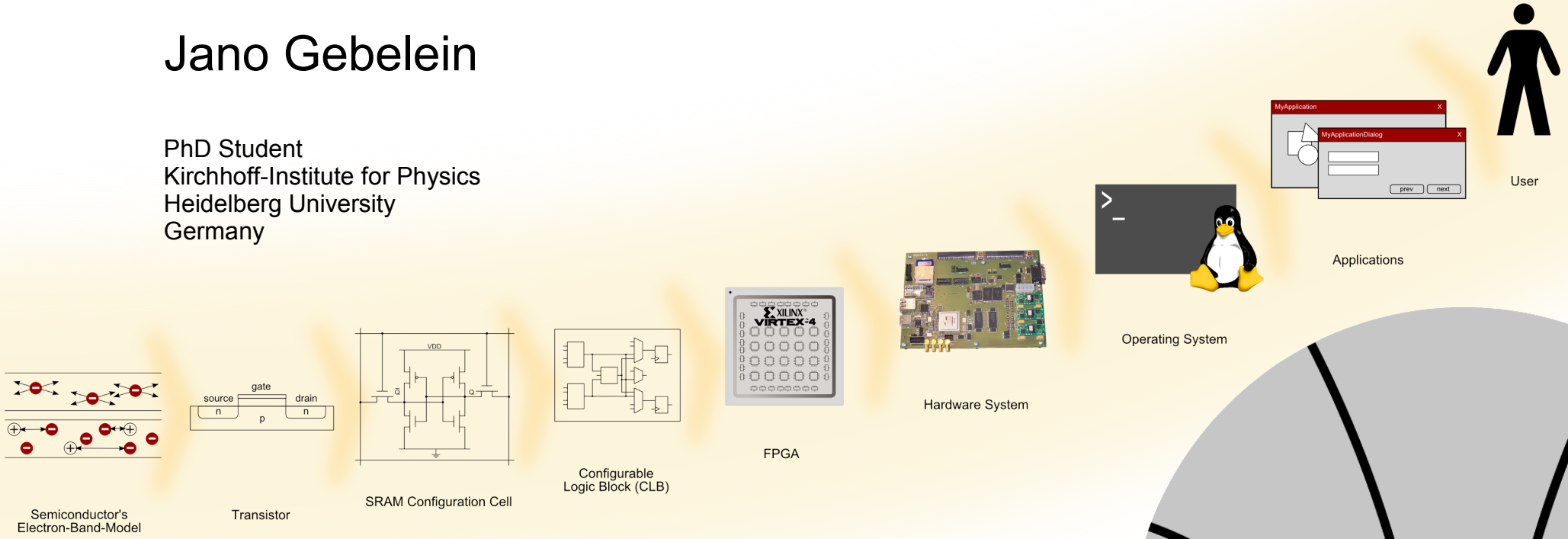
„An approach to system-wide fault tolerance for FPGAs“

Jano Gebelein

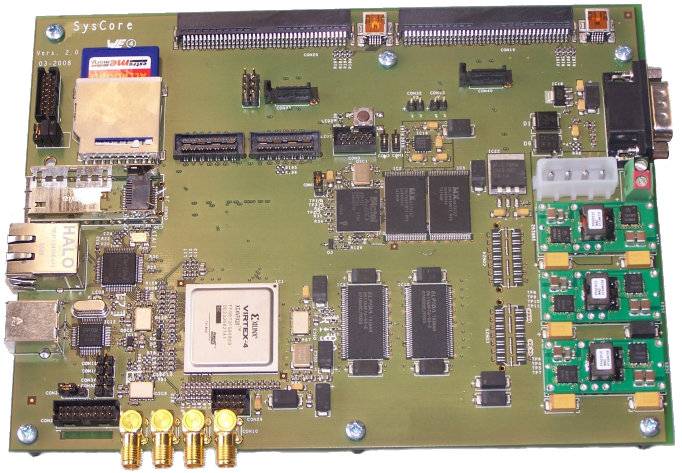
PhD Student
Kirchhoff-Institute for Physics
Heidelberg University
Germany

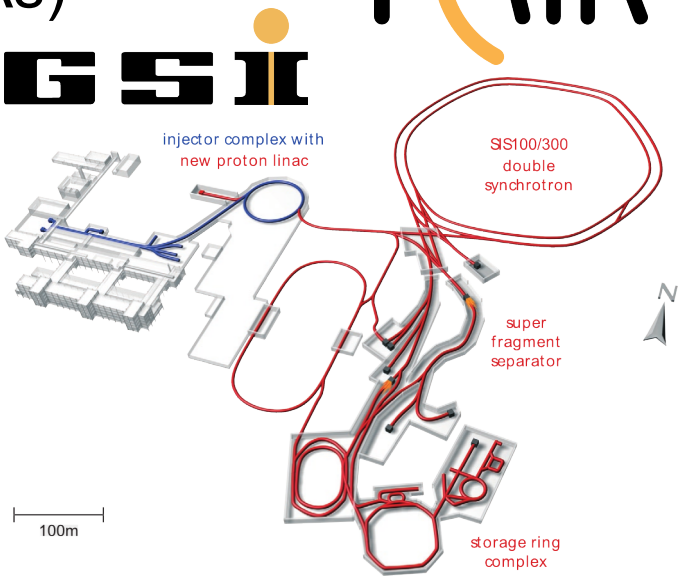


11. September 2009
ESA ESTEC, Noordwijk

Kirchhoff-Institute for Physics (Reconf. Hardware)

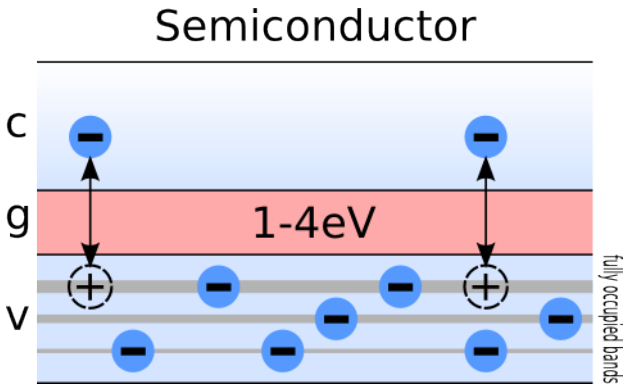
- years of experience in detector electronics
  - CERN (Geneva)
    ALICE Electron Trigger
  - GSI → FAIR (Darmstadt)
    CBM Read-Out-Controller (Xilinx FPGAs)





Ionizing Radiation Effects

- ionizing particles increase # of electron-hole pairs by scattering electrons along their way
  - pairs try to recombine immediately, but
  - electric fields (e.g. powered transistors) hinder recombination
    - electron-hole pairs are separated
    - additional carriers in material (electrons & holes)
  - this leads to
    - Single Event Effects
      - destructive errors (SEL, SEBO, SEGR)
      - non-destructive errors (SEU, SET)
    - Cumulative Effects (Displacement, TID)



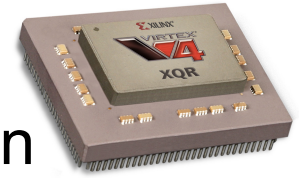
CMOS fault-tolerance in brief

- a lot of conventional CMOS techniques already exist
  - Heavy Ion Tolerant (HIT) cells [BV93]
  - Single Event Resistant Topology (SERT) [SM00]
  - Dual Interlocked Storage Cell (DICE) [CNV96]
  - ...
- not applicable to FPGA hardware circuits
  - conflict with reprogrammability feature
  - circuit design becomes too expensive

[BV93] Bessot, Velazco, “Design of SEU-hardened CMOS memory cells: the HIT cell”, RADECS 93
[SM00] Shi, Maki, “New design techniques for SEU immune circuits”, NASA Symposium on VLSI Design, Nov, 2000
[CNV96] Calin, Nicolaidis, Velazco, “Upset hardened memory design for submicron CMOS technology”, Nuclear Science Volume 43 1996

FPGA fault-tolerance in brief

- physically hardened chips

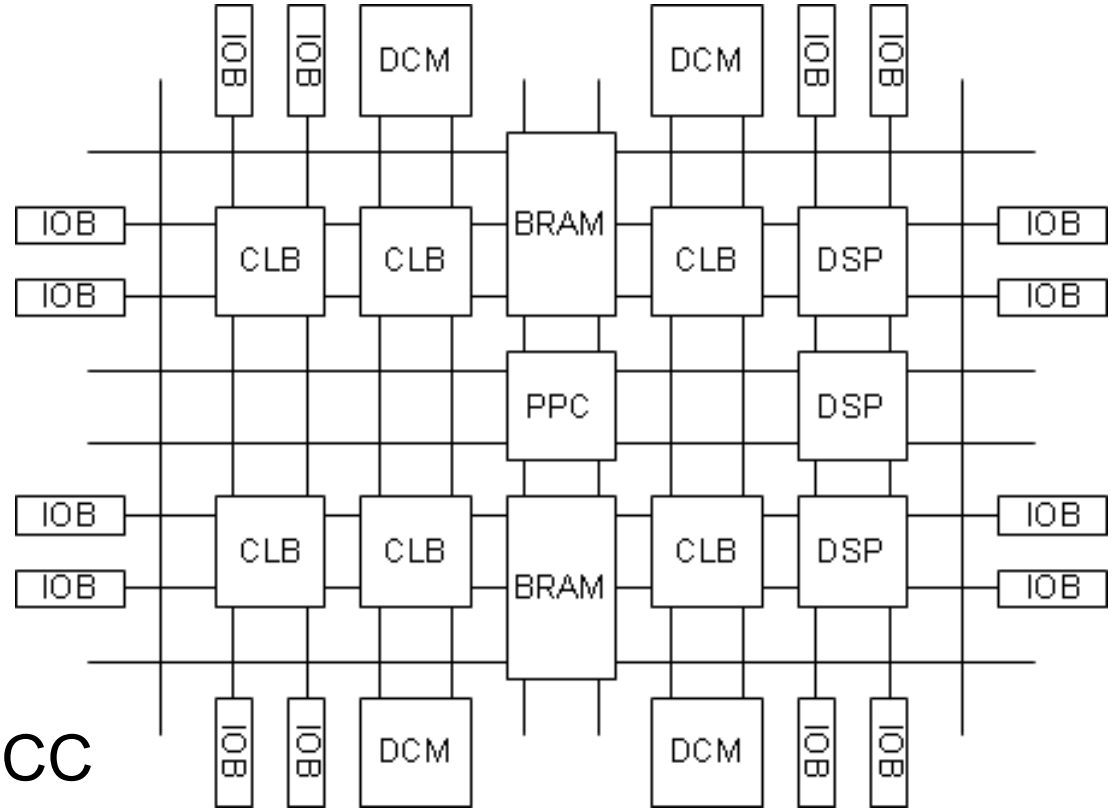


  - ceramics and advanced silicon
  - excessive shielding increases weight and size
- slightly modified CMOS architecture
- triple device redundancy
  - tripled costs, power supply, setup size

- fault tolerant design
  - no limiting constraints
  - unused chip area used for additional security features

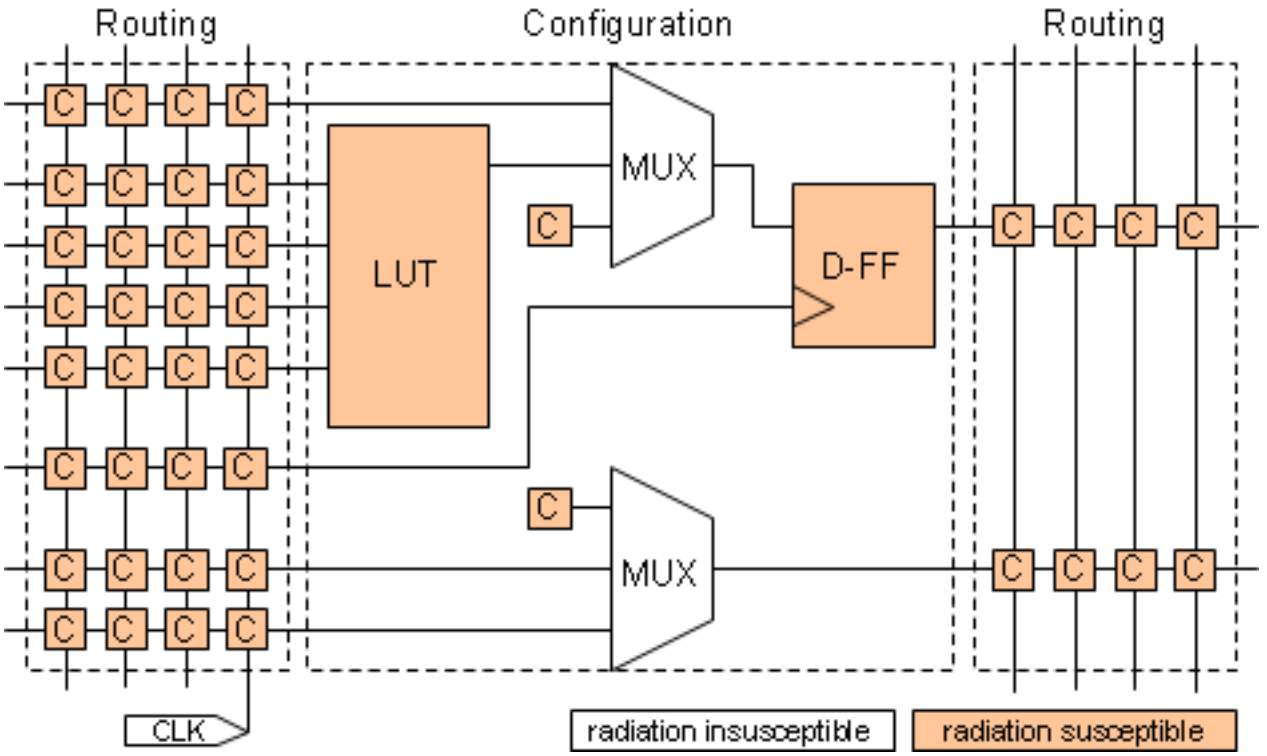
Xilinx FPGA

- SRAM based → runtime reprogrammable, but radiation susceptible
  - DCM: clear signaling, skew elimination
  - IOB: buffered inputs, grouped in banks for different standards
  - PPC: may offer embedded PowerPC
  - BRAM: SRAM-based memory with build-in-ECC on Virtex4, Virtex5
  - CLB: combinatorial logic, shift registers or RAM ff



Upset risks for FPGA components ctd.

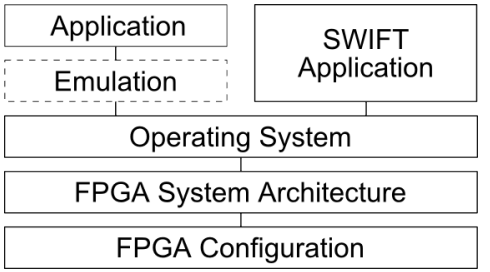
- CLB + Routing Radiation Susceptibility

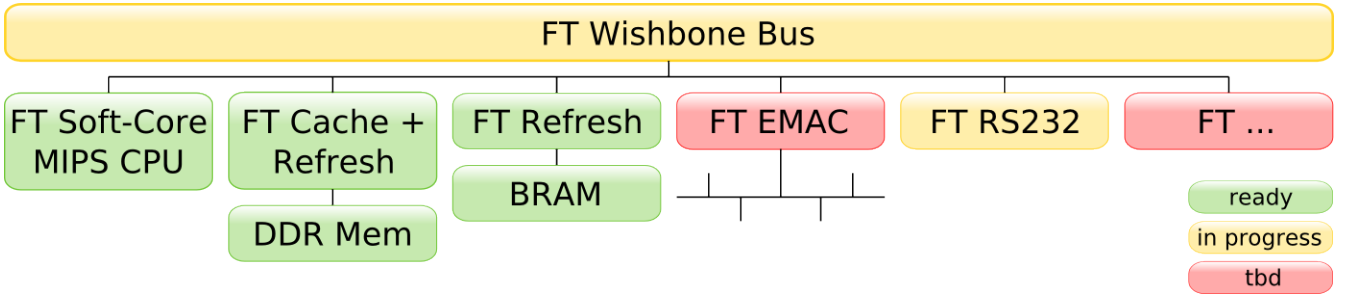
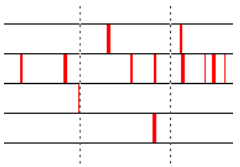


Designing SEU-tolerant circuits

- automated
  - TTM ++, area consumption --, power consumption --
  - everybody's tool, no additional knowledge about FT required
  - TMR tools under development:
    - Partial TMR Tool (BLTmr) - Mike Wirthlin (BYU)
    - Xilinx TMR Tool (XTMR)
    - and others
- manually
  - TTM --, area consumption +, power consumption ++
  - extremely time-consuming
  - best optimization and fault tolerance results (designers know about their critical code patterns)

Approach to system-wide fault-tolerance



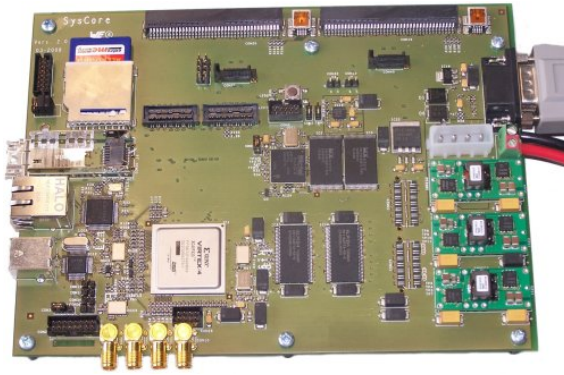
Layer	Implementation
Application	regular end-user applications (maybe SWIFT applications)
Emulation	optional emulation of SWIFT features [RCVR05]
Operating System	handles error signals, Fastboot, MIPS Compiler, MIPS SWIFT Compiler
System Architecture (HDL)	
FPGA Bit-Level	configuration scrubbing via SelectMAP configuration management via Actel 

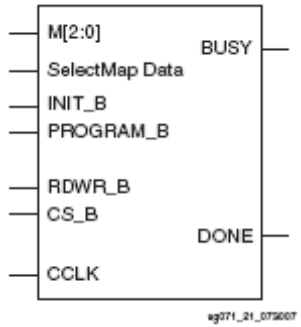
SWIFT - Software implemented fault-tolerance

[RCVR05] Reis, Chang, Vachharajani, Rangan, August, Mukherjee, „Software-Controlled Fault Tolerance“, ACM Trans. on Architecture and Code Optimization 2005

How to secure configuration matrix

- eyecatcher: “Blind Scrubbing”
  - continuous refresh resets configuration memory (including errors) without performing a chip reset (exclusive Xilinx feature) → „dynamic reconfiguration“
  - BRAM, FF, PPC untouched
  - refresh cycle less than a second
  - SysCore: Actel ProASIC 3 + Flash Memory
    - Actel connected to Virtex SelectMAP and 2x4MB Flash memory
    - Intended: file system on flash to select uploaded configuration file dynamically

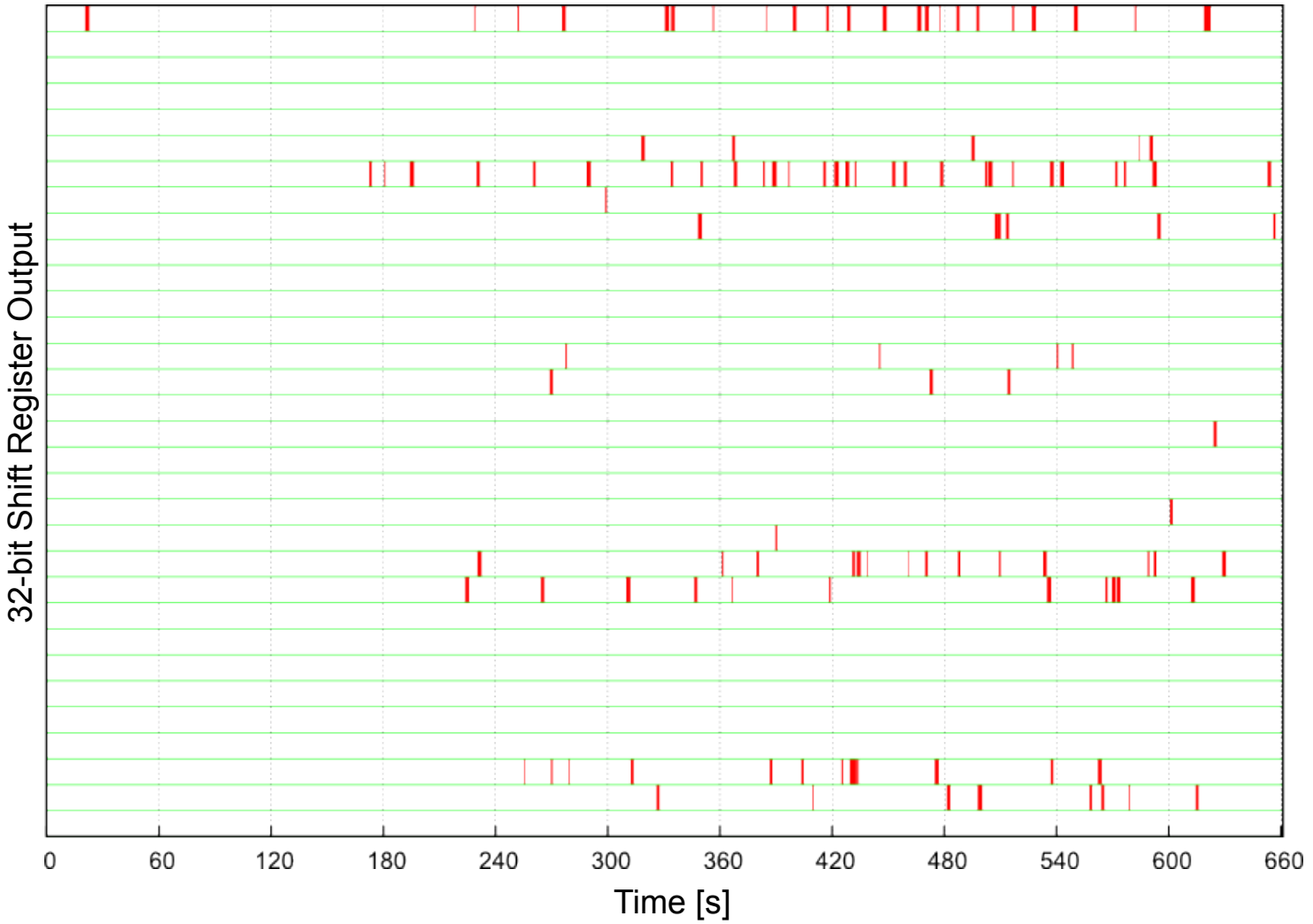




- watch out: do not use LUT as distributed RAM (SLICEM) or as shift registers, leave this for BRAM and Flip-Flops

How to secure configuration matrix ctd.

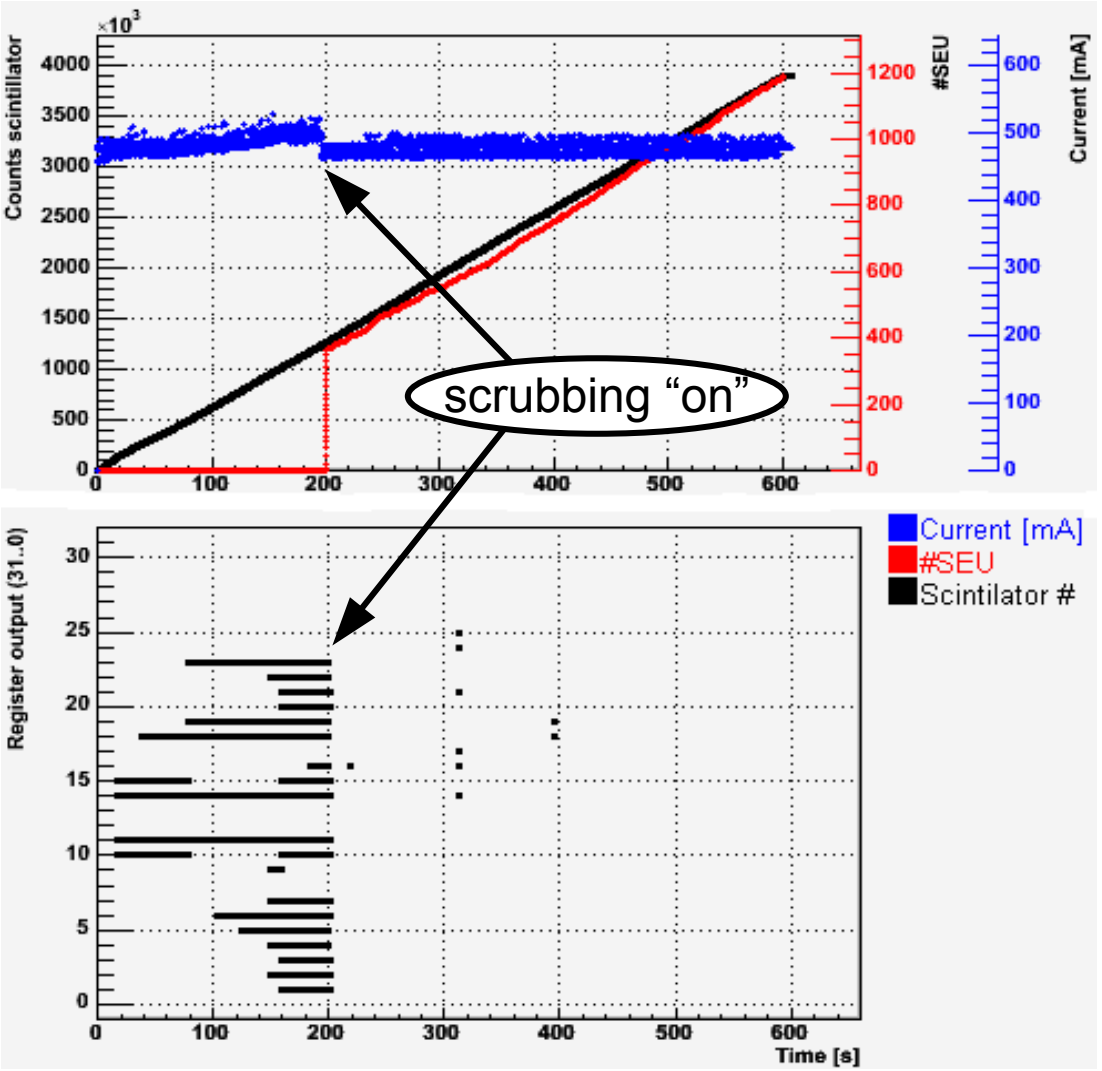
- validation results



How to secure configuration matrix ctd.

- counts <200:
  transistor threshold increases
- count 200:
  scrubbing turned on
- counts >200:
  scrubbing continuously holds current at constant values

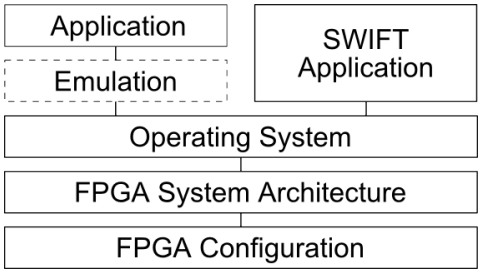
routing shortage reduced



[Røe]

[Røe] Ketil Røed, PhD thesis (to be published), private communication

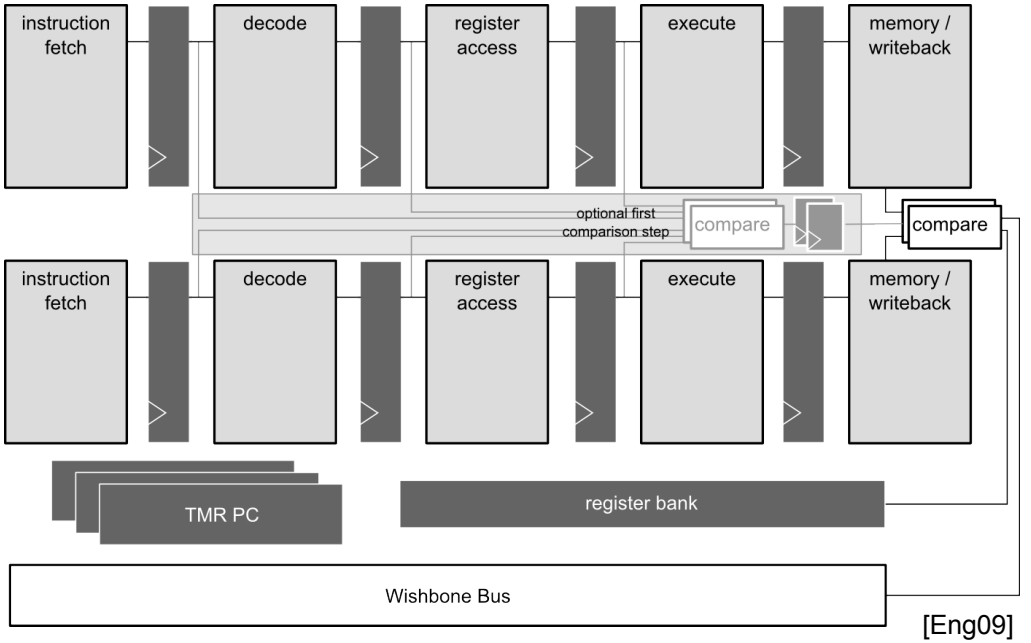
How to secure System Architecture



- mitigation techniques
  - double module redundancy for functional units
    → can wait for the next reconfiguration cycle
  - triple module redundancy for dynamic data
    → unrecoverable data has to be kept valid
  - Parity/CRC error detection/correction in data paths and buses
    → prevent data pipelining failures
  - fault tolerant state machines (hamming-based state encoding with neighbored states have fixed/minimal Hamming distance)
    → detect illegal state crossings
- intended: maximum fault tolerance at minimum size

How to intelligently secure CPU

- pipeline stages are doubled
  - comparison of all stages before memory writeback
  - in case of difference: reset PC to last valid address and invalidate all following calculations



[Eng09]

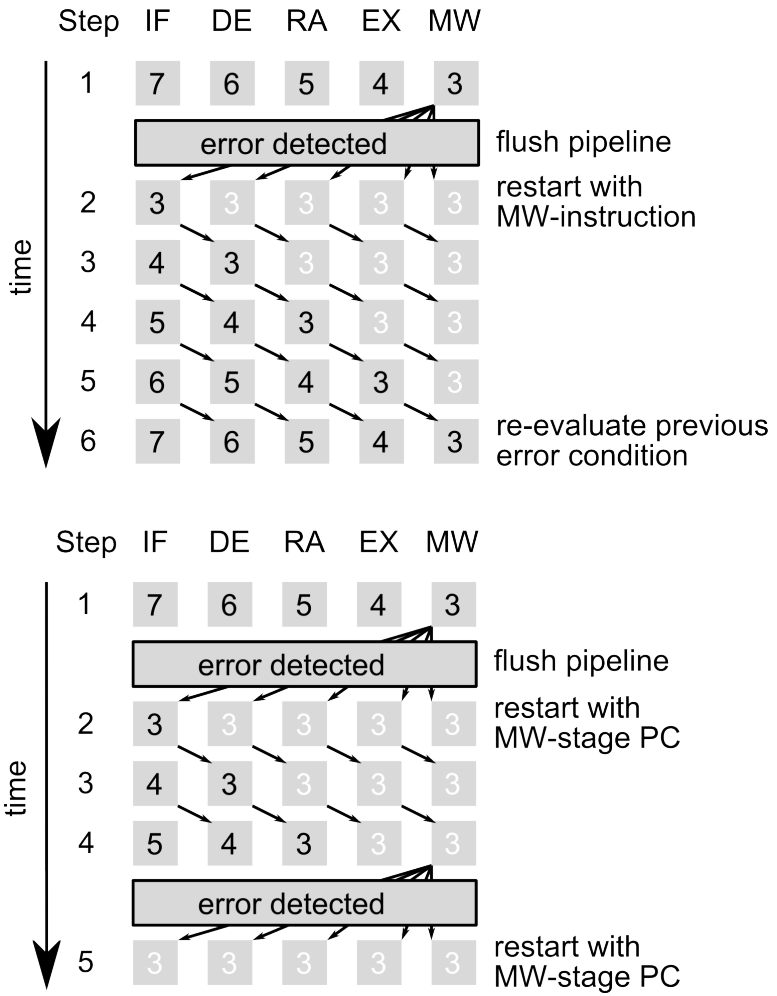
- Program Counter is tripled (data has to be kept valid)
  - in case of error or watchdog: reset cpu (disables deadlocks)

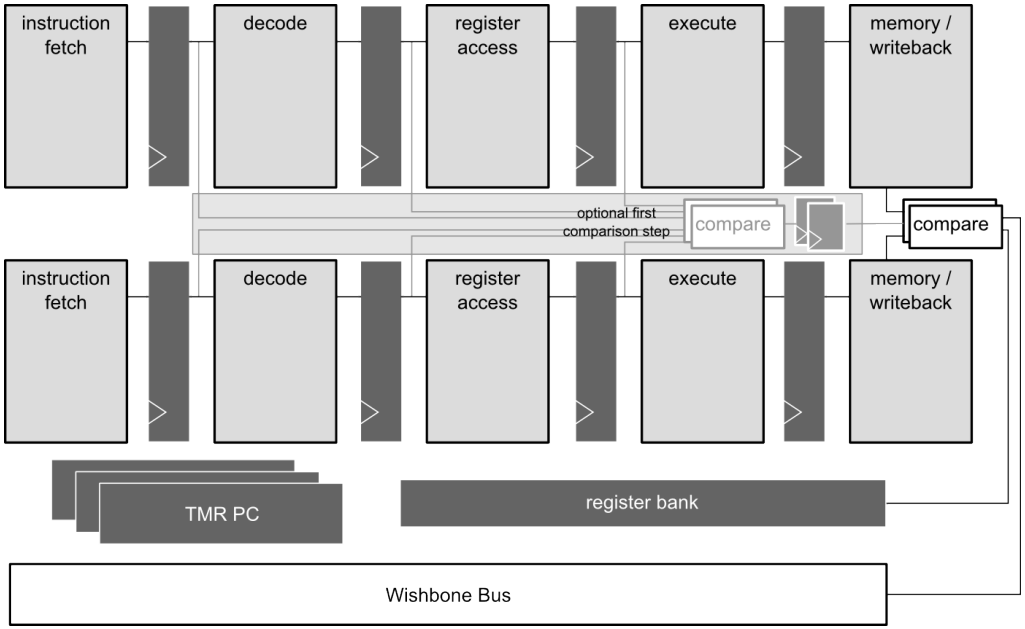
- keep in mind: Xilinx series 6 doesn't provide PowerPC

[Eng09] Engel, “Development of a Fault Tolerant Softcore CPU for SRAM based FPGAs”, diploma thesis, 2009

How to intelligently secure CPU ctd.

- Error Handling:





more at: IEEE proceedings FPL09 (P1.8) “An approach to system-wide fault tolerance for FPGAs”

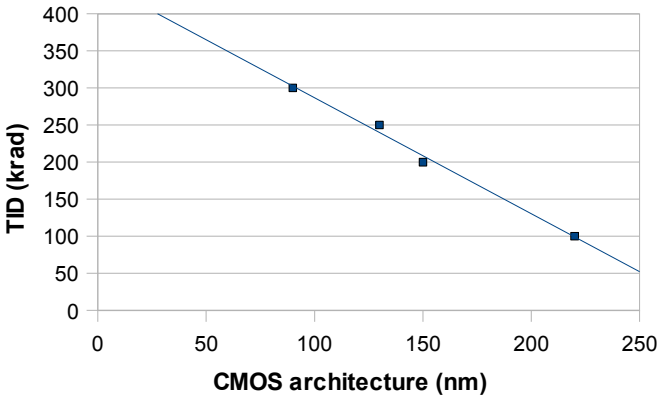
Beamtest

- Beam parameters:
  - Ru96 ions at 1,69 GeV → LET: $3{,}3 \cdot 10^{12}$ eV·cm^2/kg (Bethe-Bloch)
  - flux at FPGA: $1{,}4 \cdot 10^{10} \leq \Phi \leq 4{,}21 \cdot 10^{10}$ ions/cm^2
- $e^- = 1{,}602 \cdot 10^{-19}$ C [J/V]
- A = 1 cm^2 (approx. XC4VFX20)
- TID rate = $100 \cdot \text{LET} \cdot \Phi \cdot e^- / A$ = 740 krad to 2,23 Mrad
- exceeds max. Virtex4 TID of 300 krad [FDLH08]
  → scrubbing may have saved the chip

[FDLH08] Fabula, DeJong, Lesea, Hsieh, "The Total Ionizing Dose Performance of Deep Submicron CMOS Processes", MAPLD 2008

CMOS architecture and TID

- TID susceptibility for Xilinx Virtex (MIL-STD-883 testing method 1019 at full dose rate)
  - Virtex 220 nm 100 krad [FDLH08]
  - Virtex-II 150 nm 200 krad [FDLH08]
  - Virtex-II Pro 130 nm 250 krad [FDLH08]
  - Virtex-4 90 nm 300 krad [FDLH08]
  - Virtex-5 65 nm ~340 krad
  - Virtex-6 40 nm ~380 krad

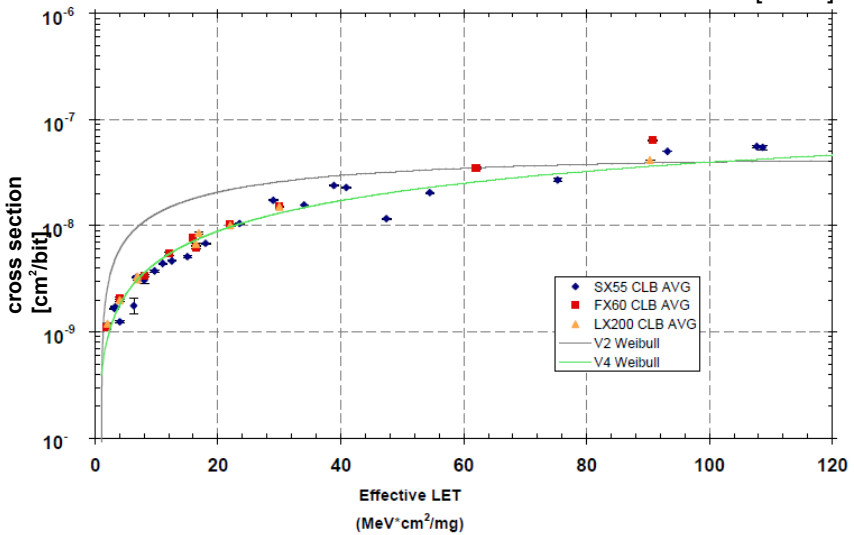


90nm and 65nm test transistors “appear capable of operating through TID stress well in excess of 1 Mrad(Si) with proper design margins” [FDLH08] (= reduced Timing) [Sch96]

[FDLH08] Fabula, DeJong, Lesea, Hsieh, “The Total Ionizing Dose Performance of Deep Submicron CMOS Processes”, MAPLD 2008
[Sch96] Schwank, “Space and Military Radiation Effects in Silicon-on-Insulator Devices”, 1996
[MIL-STD-883] http://www.dscc.dla.mil/Programs/MilSpec/listDocs.asp?BasicDoc=MIL-STD-883

Single Event Effect calculations

[AA08]

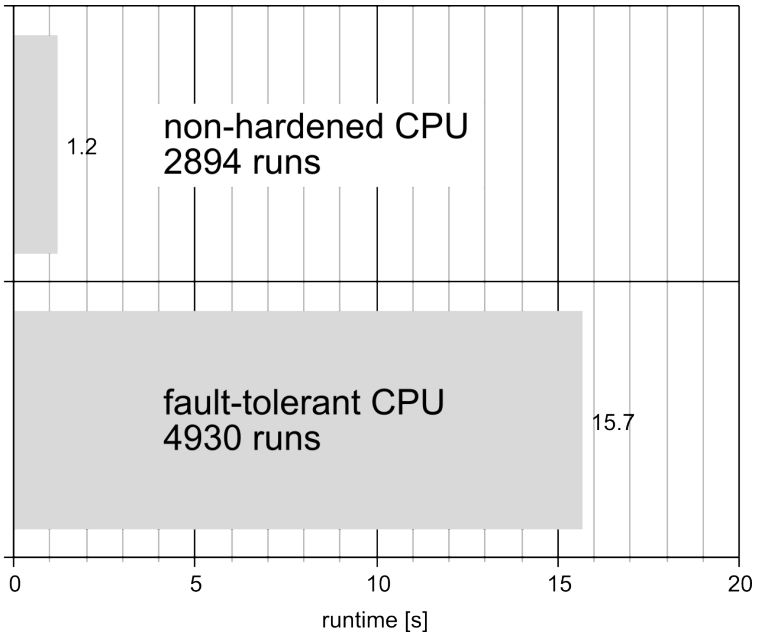


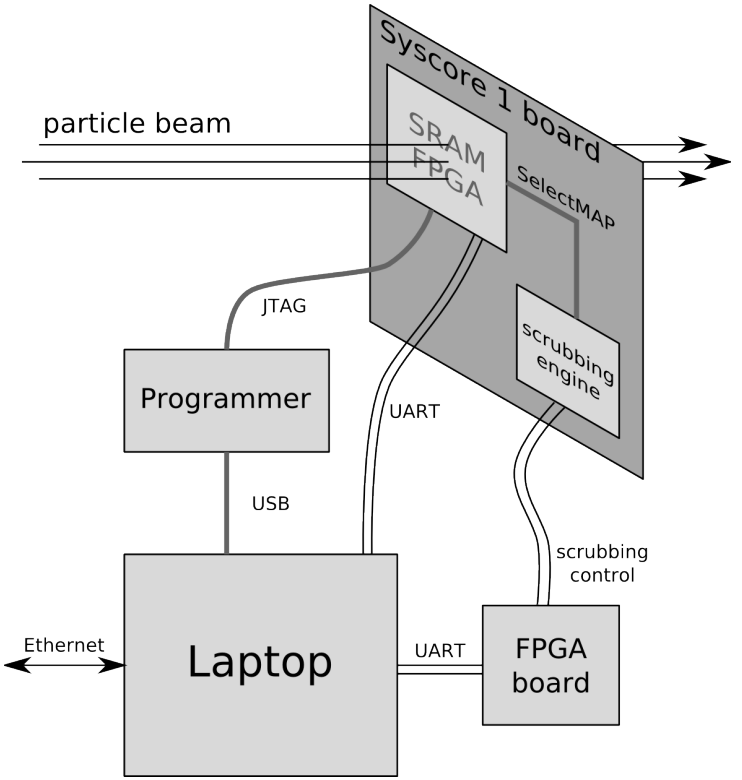
- #SEU = σ · Φ · ω / A
  - σ - cross section [cm^2/bit]
    - depends on LET
    - given by Weibull Fit: varys by $2-4 \cdot 10^{-9}$ cm^2/bit
  - Φ - particle flux [1/spill] = $1.7\text{-}5.0 \cdot 10^5$ ions/(cm^2·spill)
  - ω - design density [bit] = 7.242.624 (XC4VFX20)
  - A - FPGA chip area [cm^2] = 1 cm^2 (approx. XC4VFX20)
- #SEU per particle = σ · ω / A = 0,014-0,029
- #SEU per 15s spill = σ · Φ · ω / A = 2k to 15k

[AA08] Adell, Allen, “Assessing and Mitigating Radiation Effects in Xilinx FPGAs”, 2008

How to intelligently secure CPU ctd.

- CPU Test results
  - GSI FOPI beamtime (3 weeks long-term test)
    - 96Ru (Z=44, 42+); 1.69 GeV
    - $5 \cdot 10^6$ ions / 15s spill
    - #SEU per 15s spill: 2k to 15k





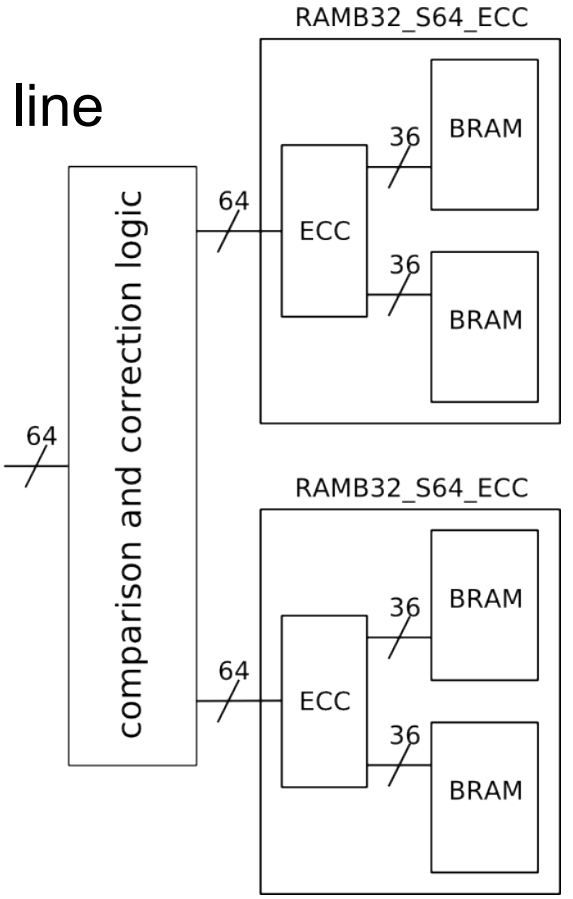
How to secure simple BRAM blocks?

- BRAM
  - made of SRAM cells
  - therefore susceptible to radiation
- increased susceptibility
  - SEU cross-section higher than for CLBs (Virtex4) [GKS+04]
  - maybe manufacturing issues: BRAM cells use smaller channels, thinner oxide, less metal than CLBs [GKS+04]
  - #MBUs for Virtex4 = 3·VirtexII = 69·Virtex [QGK+05]
- mitigated chance of hit
  - general designs use more CLB configuration bits than BRAM bits → theoretically balanced SEU cross-section

[GKS+04] George Koga Swift Allen Carmichael Tseng, “SEUs in Xilinx Virtex-4 FPGA Devices”, 2004
[QGK+05] Quinn Graham Krone Caffrey Rezgui Carmichael, “Radiation-Induced Multi-Bit Upsets in Xilinx SRAM-Based FPGAs”, 2005

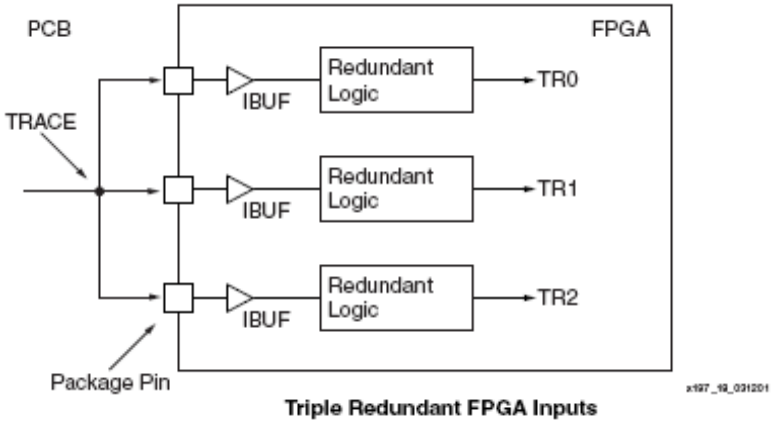
How to secure simple BRAM blocks? ctd.

- current approach:
  - each 2 parallel ECC BRAM contain identical data
    - DMR implementation, but
    - ability to correct 2+ errors in a single word line
  - each 64bit line checked for errors via continuous loop
    - single error corrected and rewritten immediately
    - double error fetched from second ECC BRAM (max security)

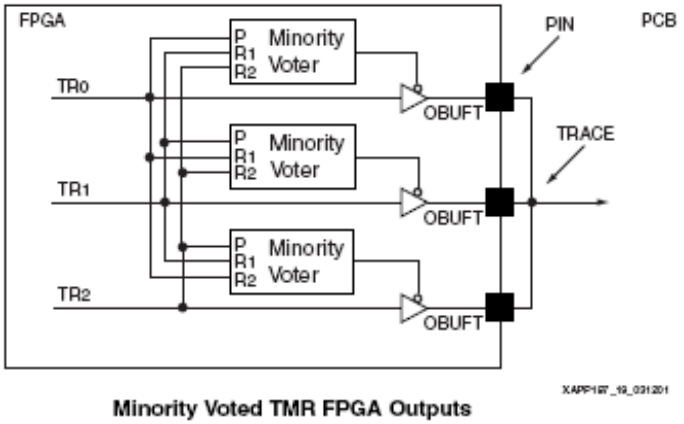


How to secure FPGA I/O Buffers (IOB)

- can be secured by TMR [xapp197]
  - combinational logic tripled
  - I/O pins tripled and hard-wired outside FPGA (no external logic required)



Triple Redundant FPGA Inputs



Minority Voted TMR FPGA Outputs

- I/O Buffers are not that critical [RWCG02]
  - just 1 of 324 IOB configuration bits and 2 two-bit combinations are able to flip an IOB behavior

[RWCG02] Rollins, Wirthlin, Caffrey, Graham, „Reliability of Programmable Input/Output Pins in the Presence of Configuration Upsets“, 2002

Upset risks for FPGA components

- Configurable Logic Blocks (Routing, LUT, MUX) ✓
- Embedded Block RAM (no distributed SLICEM memory) ✓
- Flip-Flops in combinational logic ✓
- XtremeDSP Slices (DSP48) ✗ → temporal Redundancy ✓
- Power PC (esp. internal Cache) ✗ → FT Soft Core CPU ✓
- Digital Clock Managers (DCM) ✗ → temporal sampling ✓
- I/O Buffers (IOB) → 3x hard-wired ✓

  → everything is feasible

  → Virtex6/Spartan6 without PPC → instead more Slices

✓ feasible
✗ untouchable

Lessons Learned

- creating individual fault tolerant designs without TMR has to be done manually and is very time consuming
- what we get:
  - instant fault tolerant system with default components
  - CPU standard MIPS architecture
  - Linux compatibility
  - maximum fault tolerance

- PowerPC in Virtex is obsolete → SoftCore CPU **is** required


That's all Folks!
Questions? Please get in personal contact with me!