The HCCStar ASIC for the ATLAS ITk silicon strip detector: design and verification

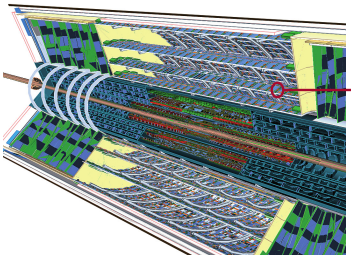
Ben Rosser

University of Pennsylvania

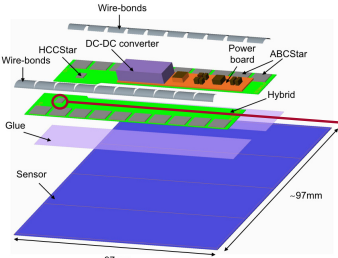
July 11, 2021

Introduction to HCCStar Version 1

- **Hybrid Controller Chip (HCCStar)**: ASIC being developed for ITK strip detector.
- Major operational problems seen when irradiating the first HCC prototype ("V0").
- New revision ("V1") currently being designed with improved radiation protection:
  - Increased mitigation against radiation-induced **single event effects (SEEs)** in digital logic.
  - Extensive **SEE simulation campaign** to verify mitigation works before fabricating the chip.

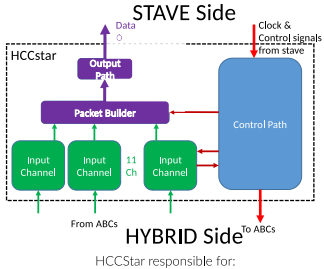


Inner Tracker: Pixels surrounded by strip modules.



ASICs (application specific integrated circuits) in strips:

- ABCStar: digitization.
- HCCStar: controls group of ABCStars (hybrid).



HCCStar responsible for:

- Sending control signals, triggers to ABCStars.
- Receiving data from each ABC in parallel.
- Combining ABC data into single output packet.

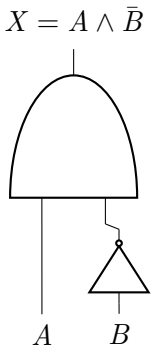
ATLAS ITk Strip TDR

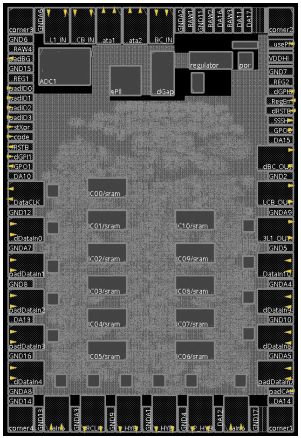
Digital Design

- HCC logic written at **register transfer level** in **Verilog** hardware description language.
- Verilog source code then compiled down into logic gates in **synthesis** process.
- Logic gates then mapped onto physical ASIC layout in **place and route** (PNR) process.

```
// Verilog source code
module example;
    input wire a;
    input wire b;
    output wire x;

    // x = a AND (NOT B)
    assign x = a && !b;
endmodule
```





Single Event Effects in ASICs

- Radiation can cause different kinds of **single event effects** (SEEs) in digital logic.
- SEEs can cause data corruption or errors that may require a reset to fix.
- Worried about two types of SEEs in the HCC: **upsets** and **transients**.

Single Event Upsets (SEUs):

- Ionizing particle causes change of state.
- Affects **flip-flops** (Verilog `reg` type).
- **Permanent** change of state; lasts until corrected.
- **Existing mitigation** in prototype HCCStar design: attempted to protect critical area, like configuration registers and state machines.

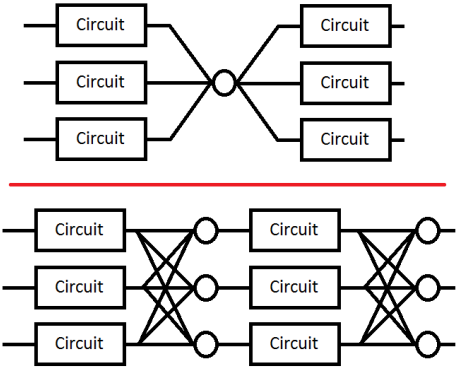
Single Event Transients (SETs):

- Voltage pulse propagating through circuit causing signals to invert.
- Affects **logic gates** (modelled using Verilog `wire` type).
- **Temporary** pulse; inversion lasts for a short time ($< 1\,\mathrm{ns}$).
- **Very limited existing mitigation** in prototype HCCStar design.

Triplicating Digital Logic

- **Triplication**, formally **triple modular redundancy** (TMR), used to protect against SEEs.
- Three copies of logic, use best-out-of-three majority voting to correct errors.

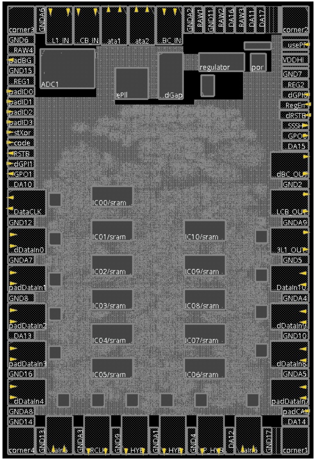
- Logic must be spaced far enough apart so one particle can't cause two SEEs at the same time.
- HCC V0 had some triplication of key flip-flops, like configuration registers and state machines.
- For protection against SETs: need to triplicate **all logic**– including majority voters– not just flip-flops.
- Full triplication **expensive**; three logic paths take up lots of space on physical chip.
- CERN tmrg tool used for automatic triplication.

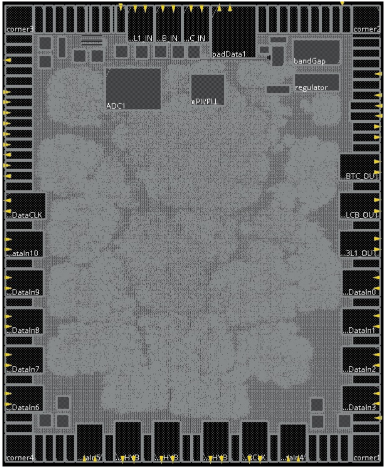


Wikipedia

HCCStar V1 Design Changes

- HCC now **mostly** triplicated:
  - Physics data not fully protected in input channels.
  - Event ID is protected: SEEs can only corrupt one event.
- Some trade-offs necessary to reach this level of protection:
  - Memory buffers made smaller.
  - Input channel SRAMs reimplemented as FIFOs.
- Deglitcher circuits added for additional SET protection.
- Even with changes, could not route triplicated design without **increasing chip width**.

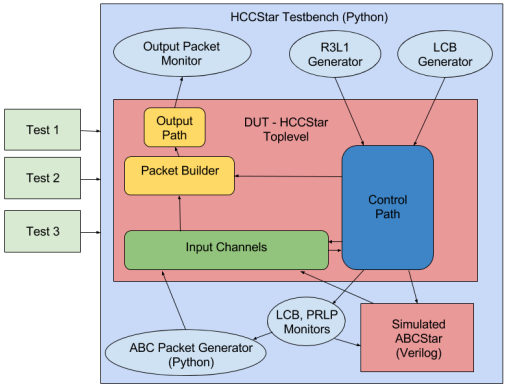




HCC widened from $3560\,\mu\mathrm{m}$ (V0, left) to $4226\,\mu\mathrm{m}$ (V1, right) in order to fit triplicated logic.

HCCStar Verification using Cocotb

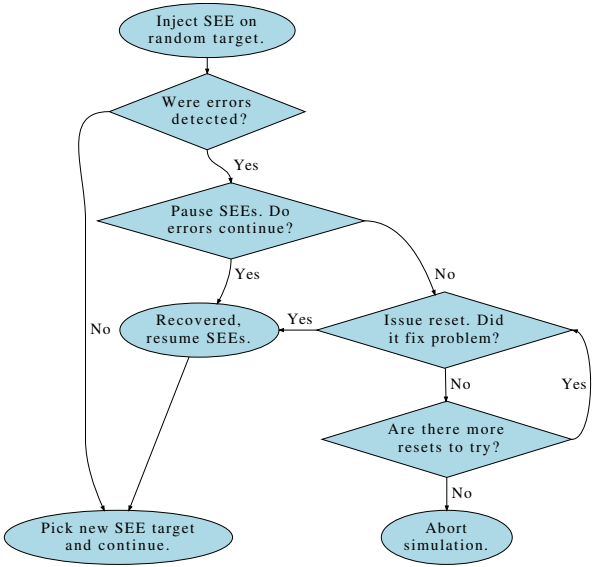
- Digital logic verification done with **Python**-based testbench powered by cocotb.
- HCC runs on Cadence Xcelium simulator, controlled via cocotb library from Python.



- Testbench built for V0, updated for V1:
  - Python implementations of serial protocols.
  - Python model of ABCStar data flow can generate packets in response to commands.
  - Tests are fully self-checking; full suite ran nightly for continuous integration.
- Testbench used for both **HCC-only** and **hybrid-level** simulations:
  - Real ABC used instead of Python model.
  - Used to simulate data flow with realistic HL-LHC conditions.
  - For V1, added support for module-level simulations with AMAC too.

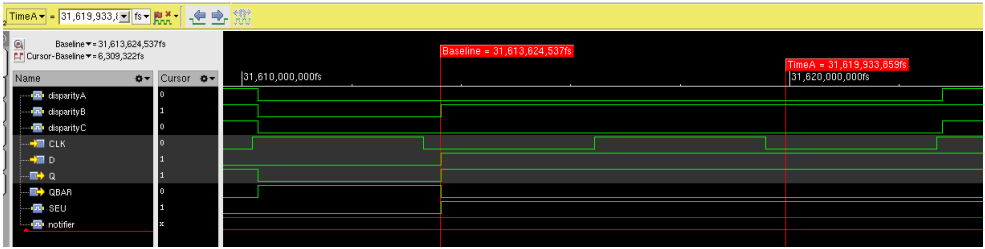
Rapid SEE Simulations

- Cocotb testbench used to build rapid SEE simulation framework for HCC V1.
- Inject SEEs **as fast as possible**, maximize chances of finding problems.
- Run simple loop to probe chip behavior.
  - HCC-only: read out physics data and registers continuously.
  - Hybrid/module-level: run realistic data flow simulations, inject SEEs on HCC.
  - Run over post-PNR, gate-level design.
- **Measure severity** of any detected issues:
  - First see if chip recovers without a reset.
  - Attempt "soft" reset (no reconfiguration) before full "hard" reset.



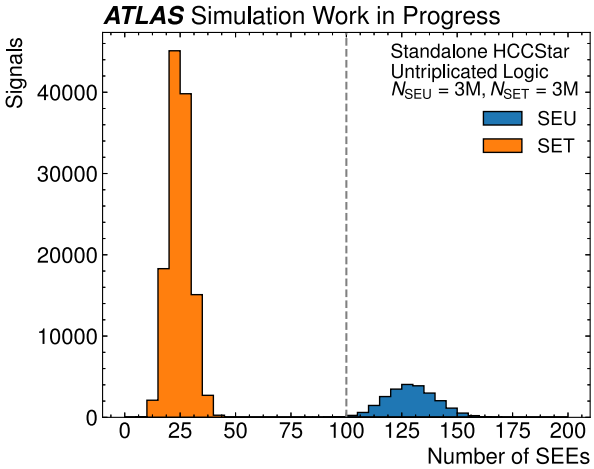
SEEs in Triplicated Logic

- SEEs on triplicated logic should **cause no errors** if TMR implemented correctly.
- Run separately over TMR, non-TMR parts of chip: confirm no errors in TMR simulation.
- Attempt to automatically detect potential triplication errors in real time:
  - Most TMR copies of flip-flops should be updated by majority voter every clock cycle.
  - Automatic check: look for redundant copies, check that all three agree after SEE.
  - This check **found problems**: after investigation, discovered some **real bugs in the chip**: signals not properly refreshed by voter could allow SEEs to accumulate over time.
  - Catches SEE vulnerabilities immediately, without waiting for data errors to be detected.



SEEs in Untriplicated Logic

- Errors expected in non-TMR logic: use simulation to estimate rates.
  - Understand different failure modes.
  - How often are resets required?
  - Hard resets require reconfiguration: ensure they are rare.
- Perform high-statistics simulations with **100 SEEs** per wire/flip-flop.
- Snapshot of current results:
  - In 3 days, reached SEU goal, performed 25% of SETs.
  - Low error rate, very few hard resets.
  - Results must be confirmed on final version before submission.



Injected 6M SEEs, found **18243** errors (0.3%), of which **3109** (17%) required a reset, **only 3** (0.016%) required hard reset.

Current Status and Conclusion

- When designing particle detector electronics, SEE mitigation very important:
  - Add as much protection as possible!
  - Consider both SETs and SEUs when adding mitigation to a chip.
- Python-based verification using cocotb is a very powerful tool for ASIC design.
- SEE simulations using cocotb testbench are essential tool in SEE mitigation:
  - Used to confirm and track down problems seen during V0 irradiations.
  - Found and fixed many triplication-related bugs, removing potential SEE vulnerabilities.
  - Helping to build confidence in V1 radiation tolerance prior to submission.
- Thank you for your attention!

Backup