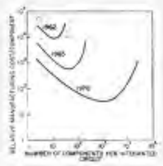
Generator-Based Hardware Design

Elad Alon — Professor Electrical Engineering and Computer Science, University of California Berkeley

Designs Thrust : Circuit Realization at Faster Timescales (CRAFT)

DESIGN COST AND AGILE DESIGN

- Development cost for a state-of-the art SoC is ~$300M
  - Need to sell 60M $5 chips for manuf. cost to equal design cost
- Software faced a similar challenge with ballooning NRE (complexity = $$$)
  - "Agile" approach developed in response, realized order-of-magnitude improvement



TRADITIONAL HARDWARE DESIGN ISN'T AGILE

- Dominant problem is dearth of re-use
  - Hardware IP of course is re-used, and expected to continue
  - But often building an SoC precisely because you want to specialize something…
- Our approach: **re-use the design process** itself, not the results of it
  - I.e., capture designers' methodologies in the form of executable **generators**
- Under CRAFT, our UC Berkeley, Cadence, Northrop-Grumman, and BAE team has been developing this approach and quantifying its benefits

PLATFORM FOR DIGITAL GENERATORS

- **Chisel** (embedded within Scala) provides same level of designer control as RTL
  - Not "HLS" - provides improved **software** abstractions for capturing methodologies

- Decouple generic vs. platform-specific optimizations to maximize re-use
  - Borrow from software again and use an intermediate representation (**FIRRTL**)
- Capture physical design approach in generators with **Hammer**

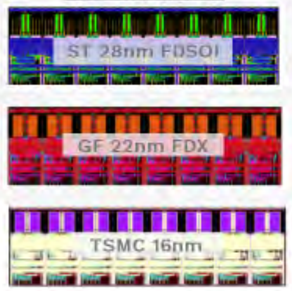
GENERATORS FOR VERIFICATION

- No one will bank their $300M SoC on "correct by construction"
  - So need to (agilely) verify any instance that you generate
- Cadence's Verification Workbench (VWB) eliminates manual labor by automatically generating automated test environment
  - Chisel passes design information to VWB via IP-XACT metadata

WHAT ABOUT ANALOG/MIXED-SIGNAL?

- Analog has traditionally been most resistant to automation
  - But requirements imposed by sub-20nm processes and analog complexity within SoCs have aligned to make generators very appealing
- Developed the **Berkeley Analog Generator (BAG)** as a Python-based process-portable framework enabling such generators

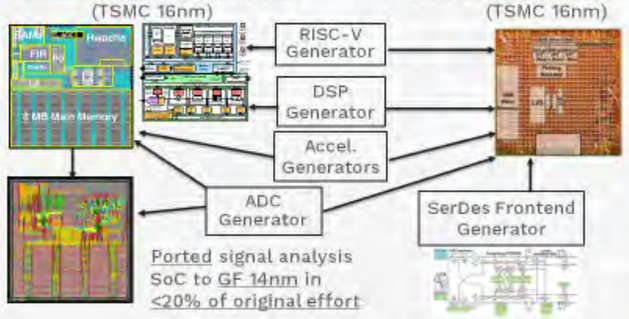
ADC Generator:



GENERATORS AND CHIPS (PHASE 1)

10.3M Gate Signal Analysis SoC (TSMC 16nm)

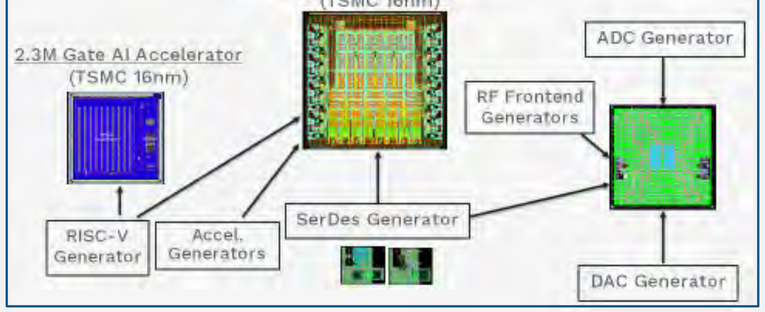
4.2M Gate Sparse Recon. SoC (TSMC 16nm)



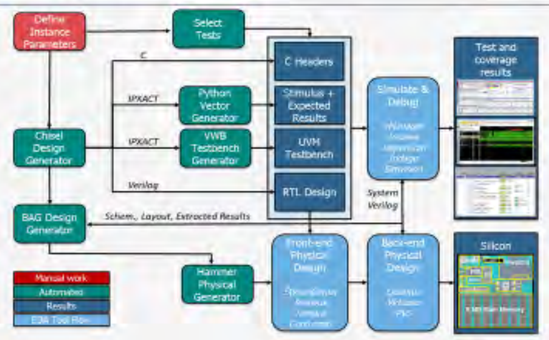
Ported signal analysis SoC to GF 14nm in <20% of original effort

GENERATORS AND CHIPS (PHASE 2)

24.6M Gate Multiprocessor SoC (TSMC 16nm)



PUTTING IT ALL TOGETHER



This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA)
The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government

THE ELECTRONICS RESURGENCE INITIATIVE