CRAFT

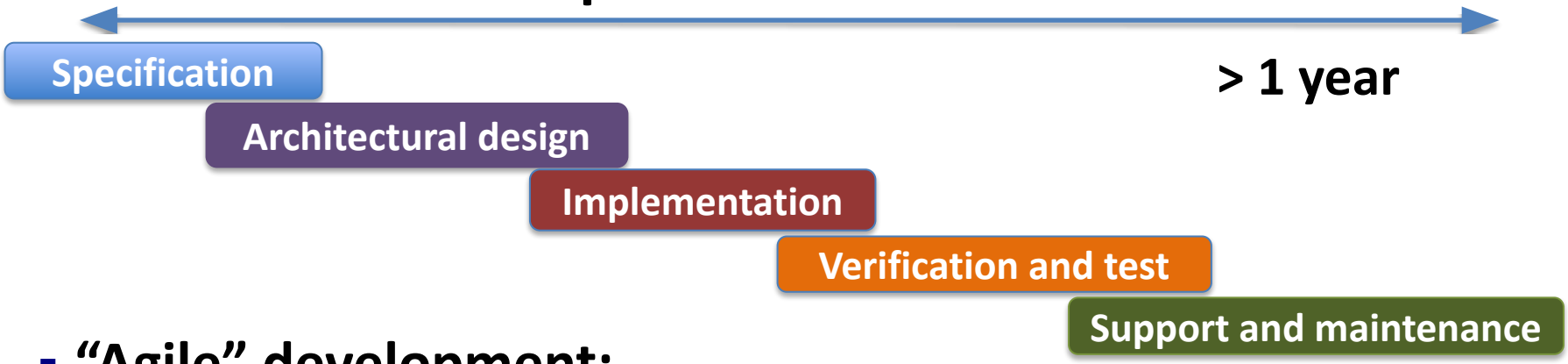
Circuit Realization At Faster Timescales

Stevo Bailey and Paul Rigge

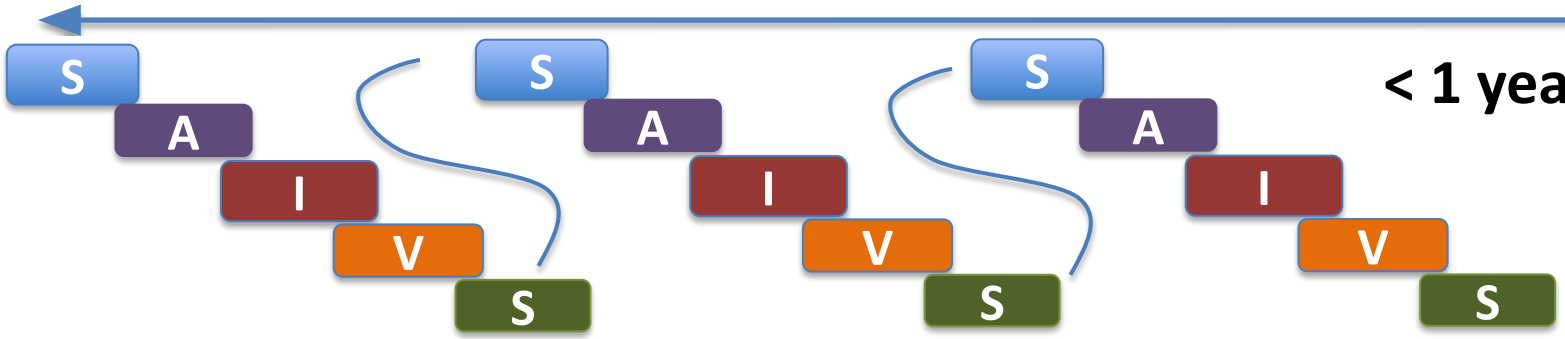
Craft Phase 1 - Agile Development of DSP Chips

Agile Software Design

- "Waterfall" development:

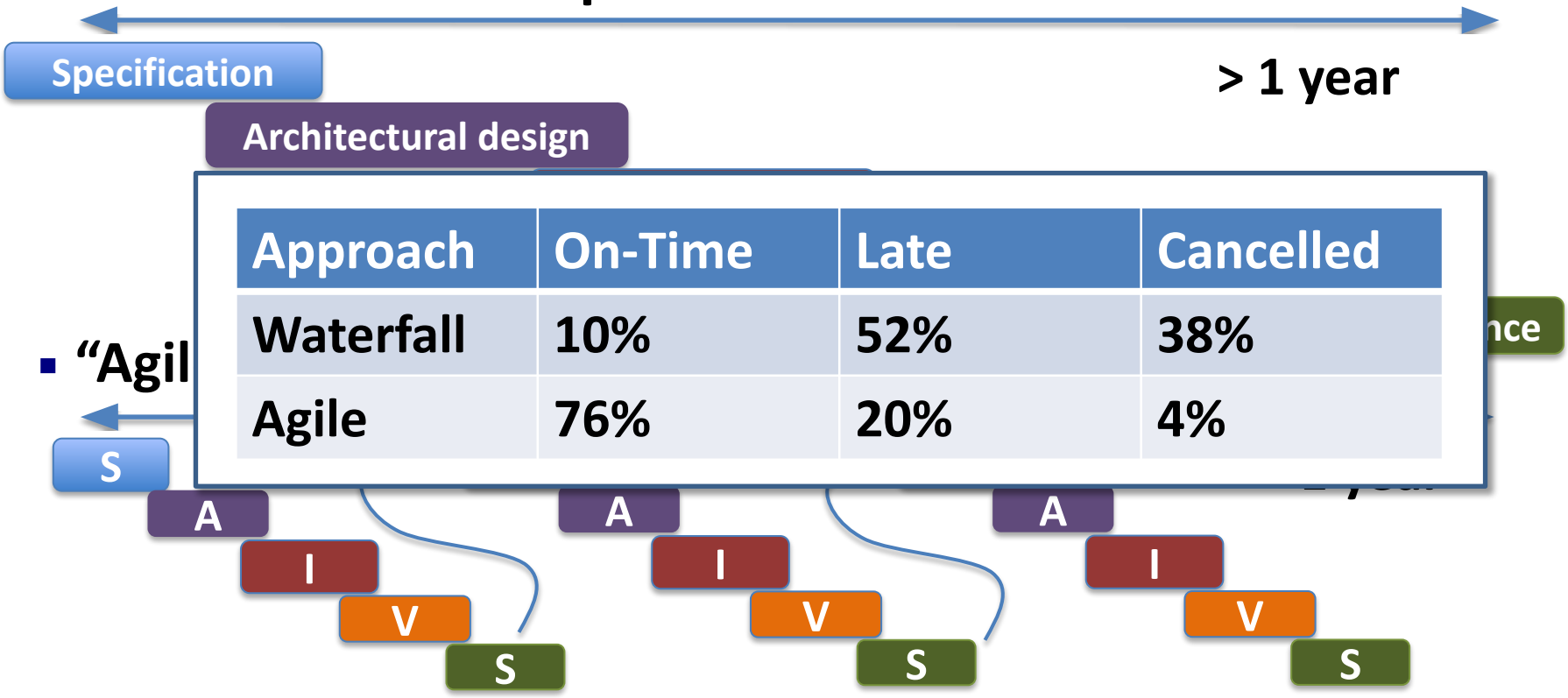


- "Agile" development:



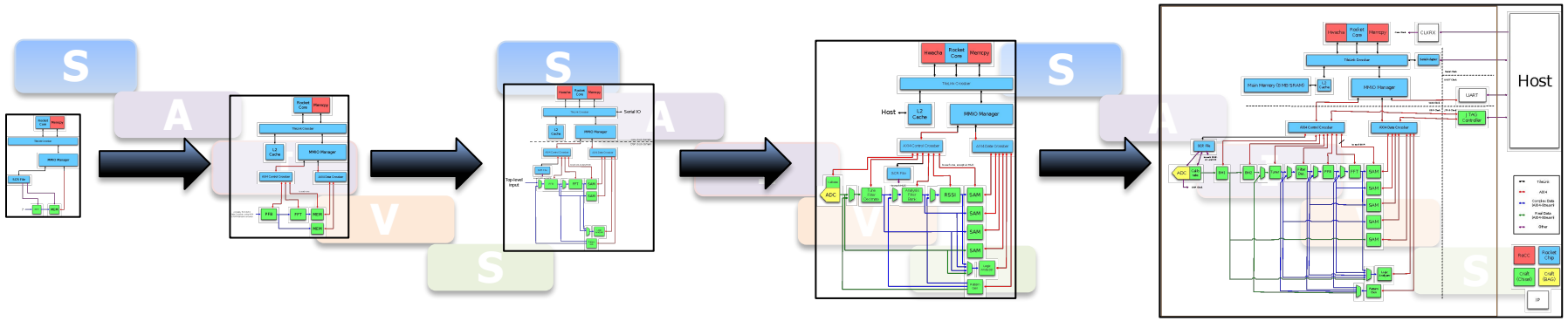
Agile Software Design

- "Waterfall" development:



Approach	On-Time	Late	Cancelled
Waterfall	10%	52%	38%
Agile	76%	20%	4%

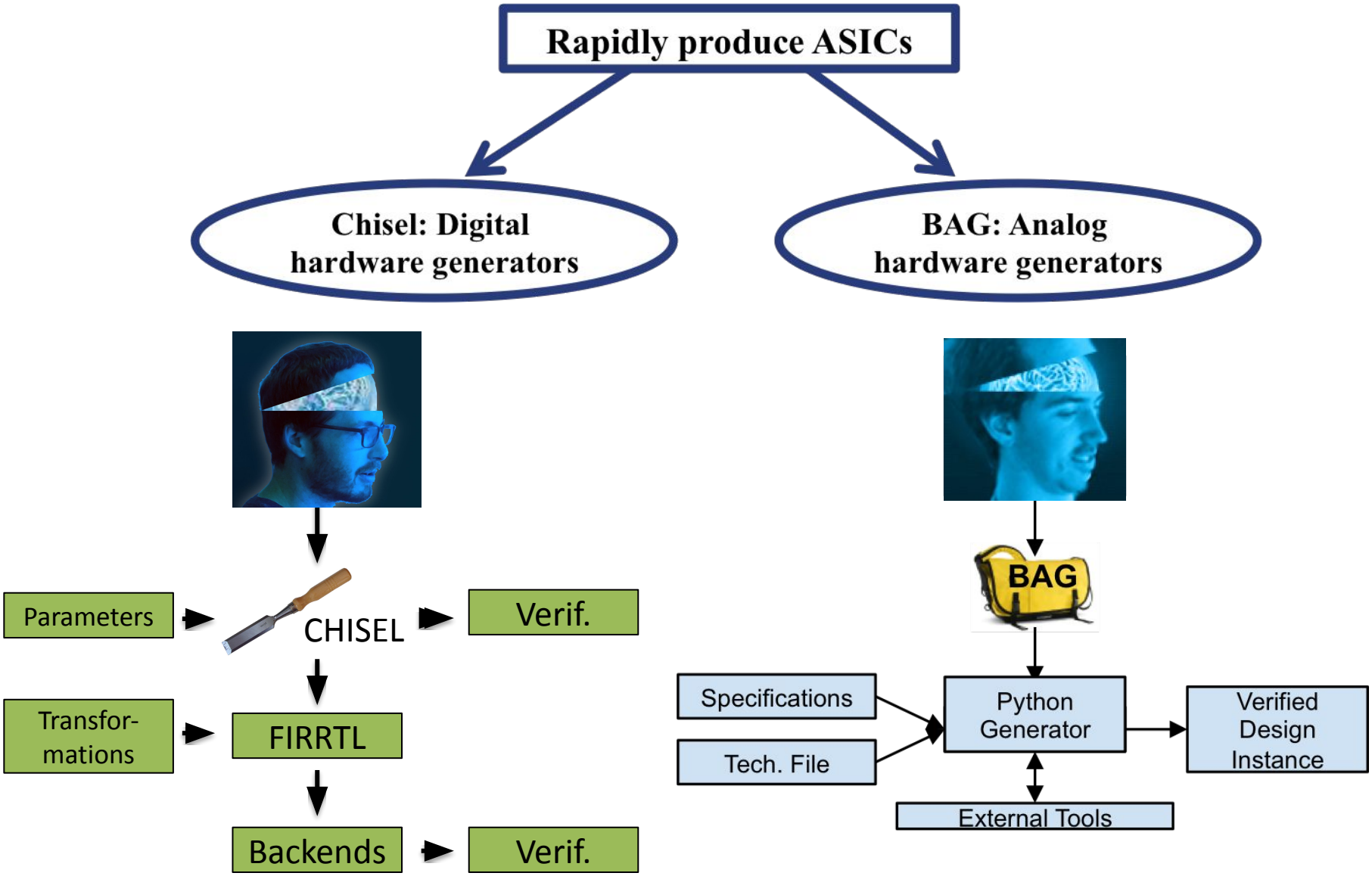
Agile Chip Design



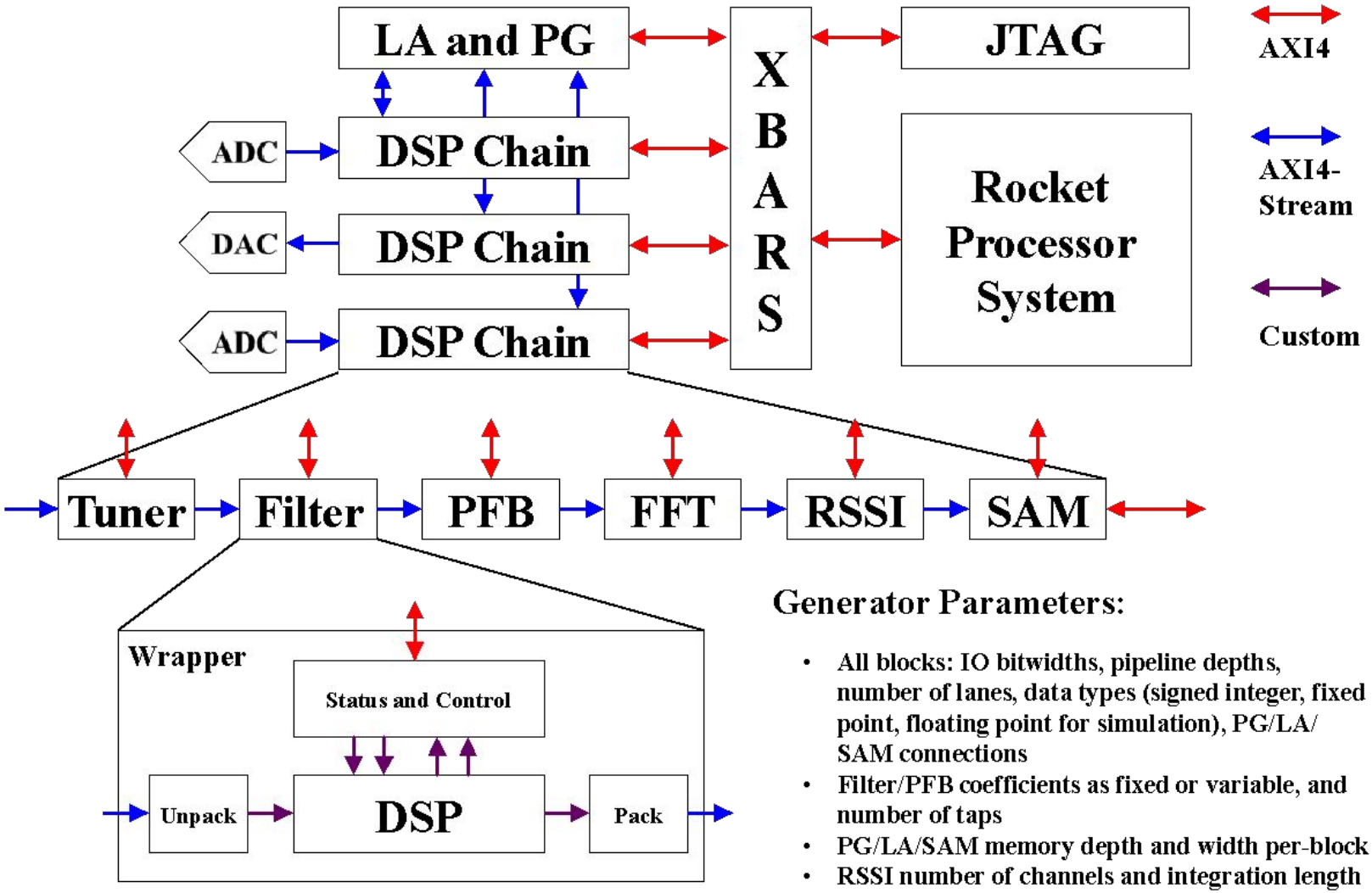
Need methodologies and flows for:

- Scalable, parameterized design generators
- Rapid design turn-around
- Aggressive re-use
- Agile verification and validation

Capturing Design Knowledge



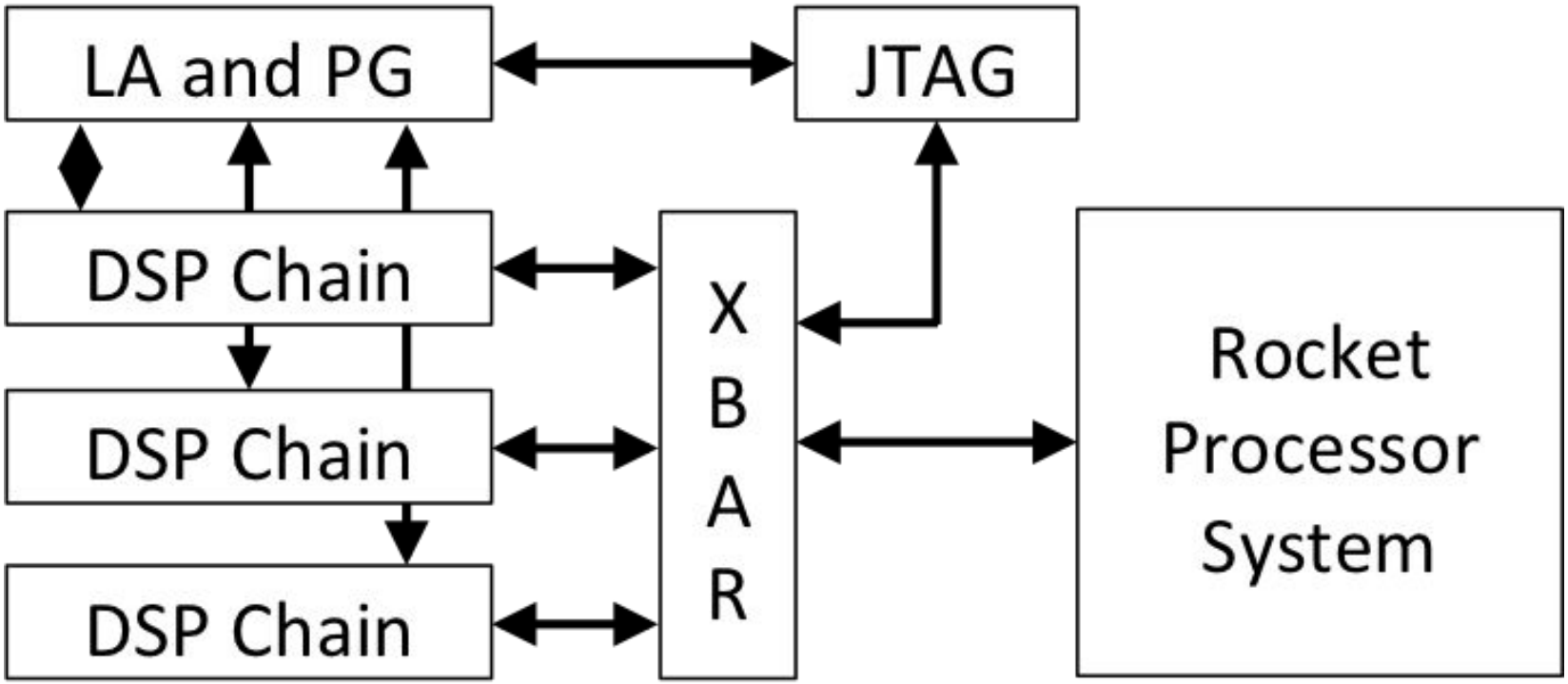
DSP Chip Methodology



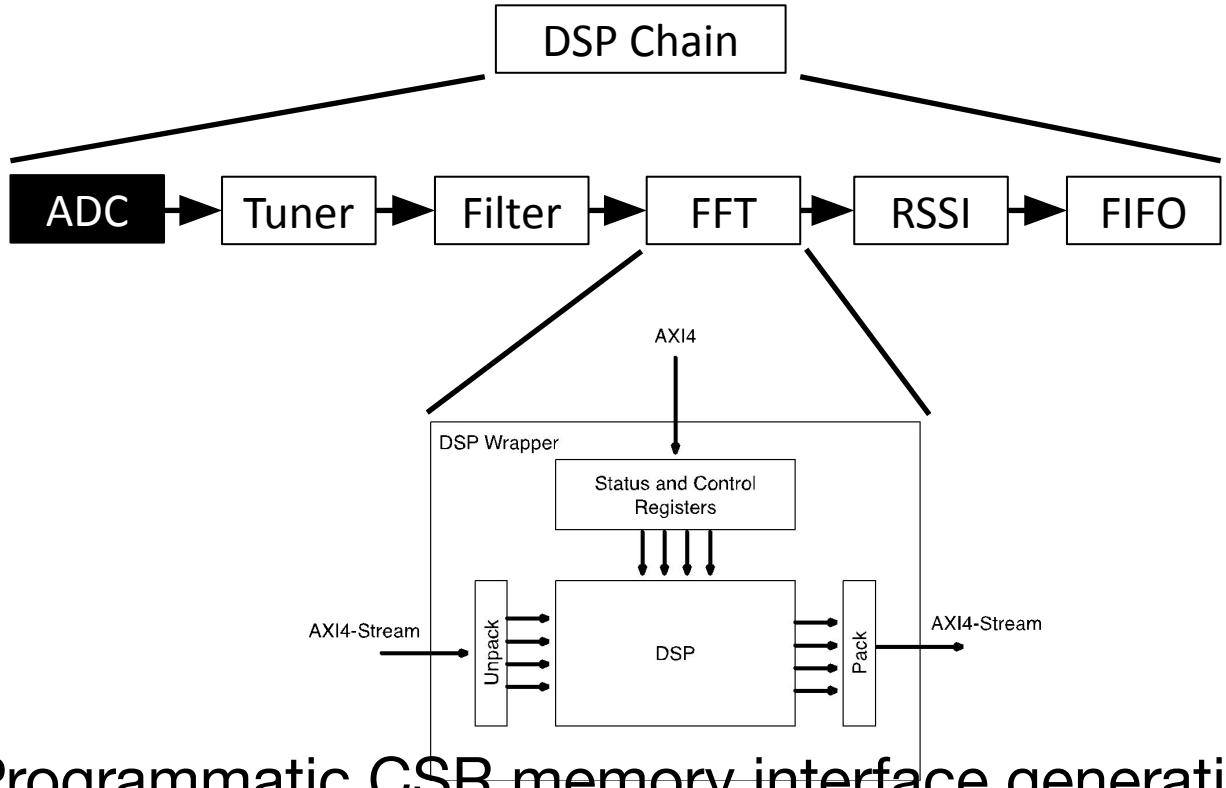
Generator Parameters:

- All blocks: IO bitwidths, pipeline depths, number of lanes, data types (signed integer, fixed point, floating point for simulation), PG/LA/SAM connections
- Filter/PFB coefficients as fixed or variable, and number of taps
- PG/LA/SAM memory depth and width per-block
- RSSI number of channels and integration length
- DSP Chain composition (automatic Xbar sizing)

System Design Methodology



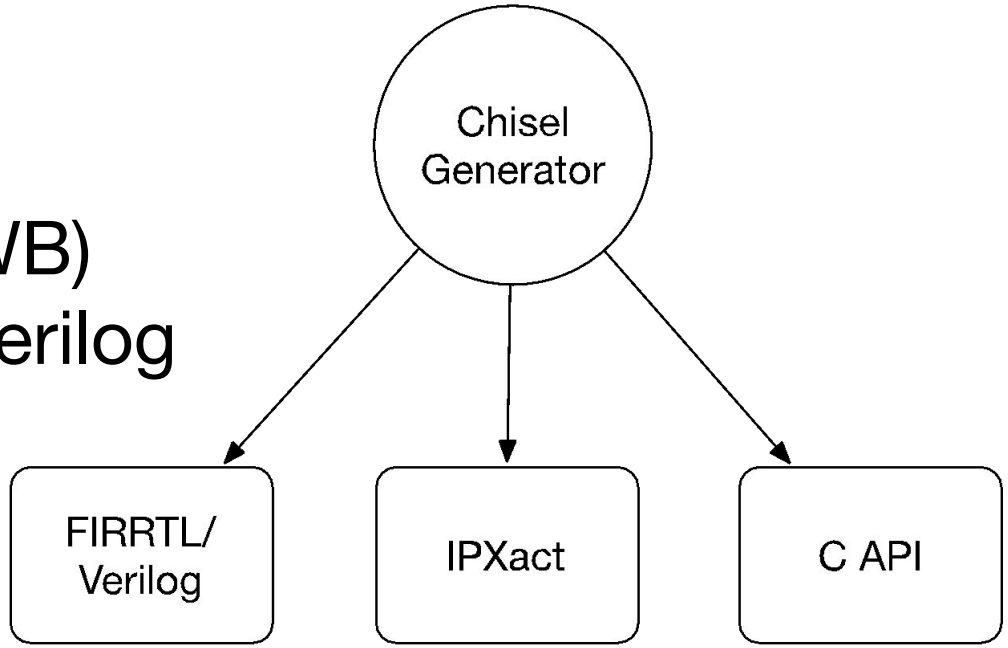
DSP Generators

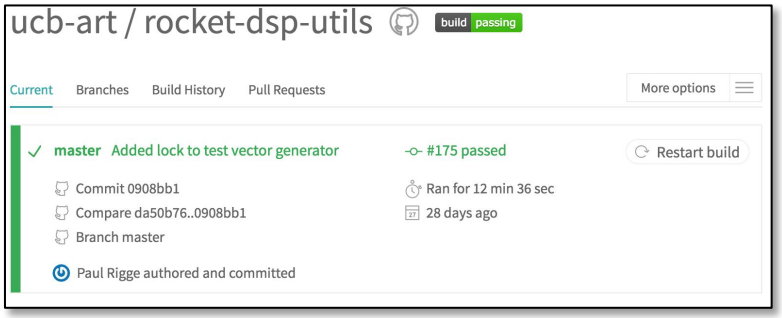


- Programmatic CSR memory interface generation
- IPXact generation for verification
- C header with memory mapping, convenience functions, etc.
- Test harnesses
  - Chisel, Verification Workbench, and C tests!

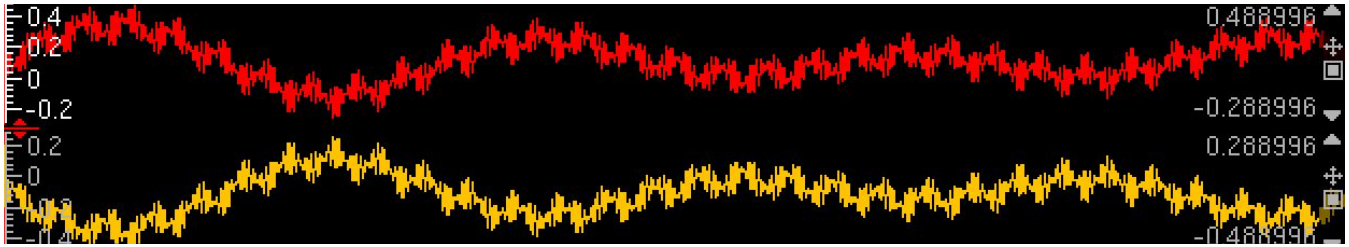
Verification

1. Chisel Unit Tests
2. Verification Workbench (VWB)
3. C mixed signal tests with Verilog

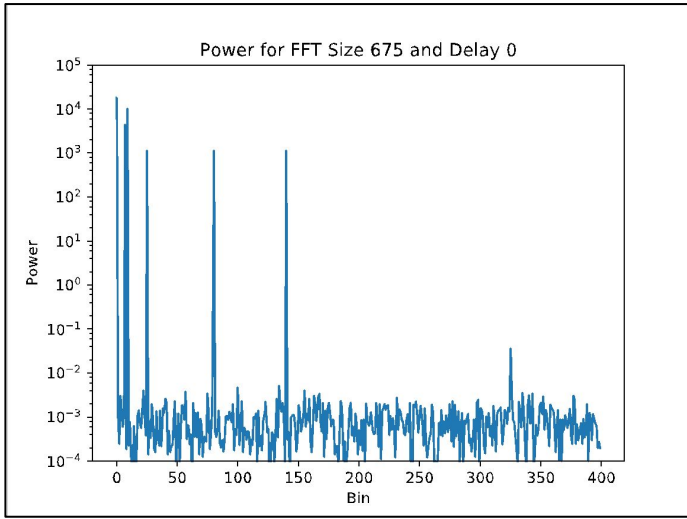




Continuous Integration

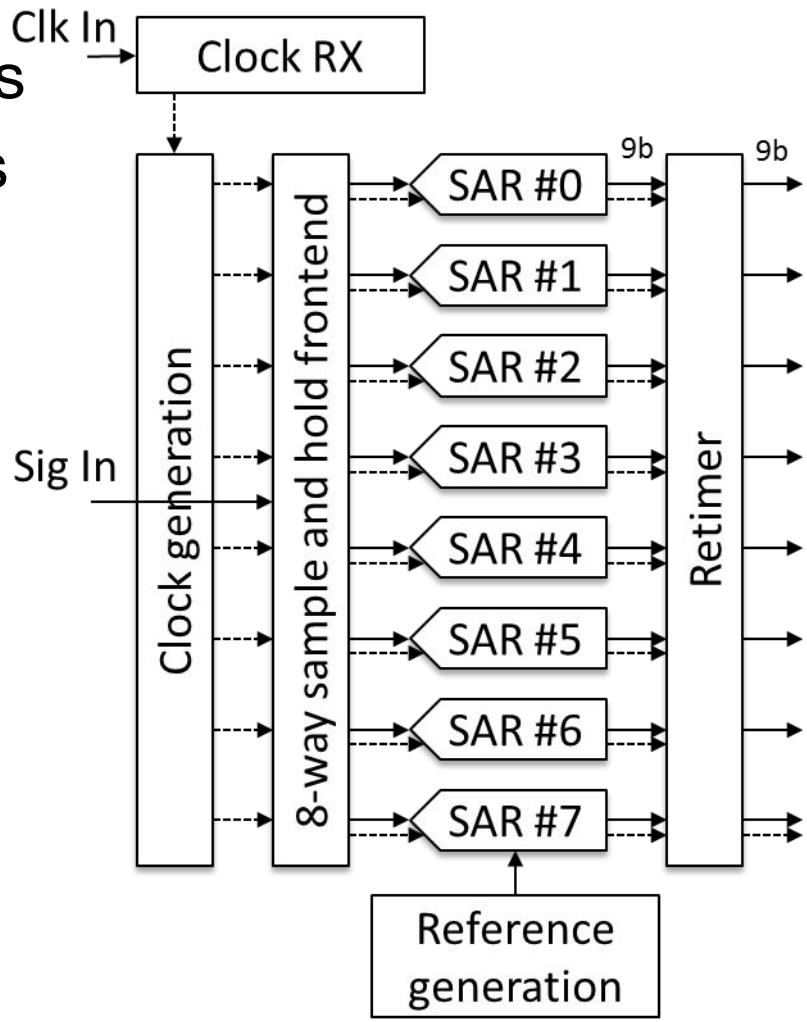


Top-level C test included SystemVerilog analog model of ADC



BAG ADC Generator

- **"ADC" generator includes:**
  - Asynchronous SAR sub-ADCs
  - Resistor-ladder DACs for bias & offset
  - High-speed clock RX
  - S/H, phase generation circuits with skew correction
  - Custom digital retimer

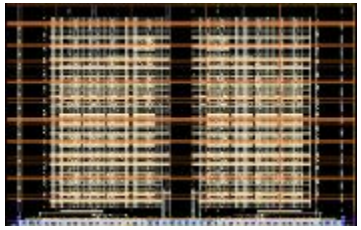


- **Generator parameters:**
  - Number of bits, redundancy
  - Interleaving factor
  - C-DAC thermal noise spec.
  - Comparator noise and sample-rate spec.

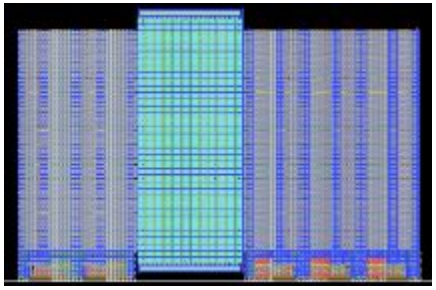
Example Layouts in 16nm

Comparator

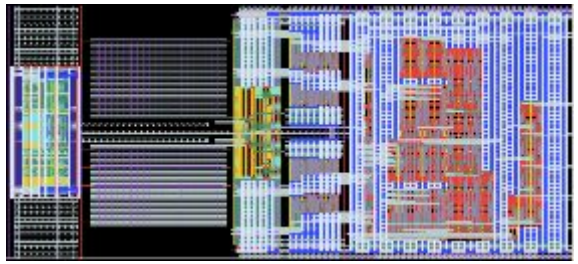
Switch-Cap DAC

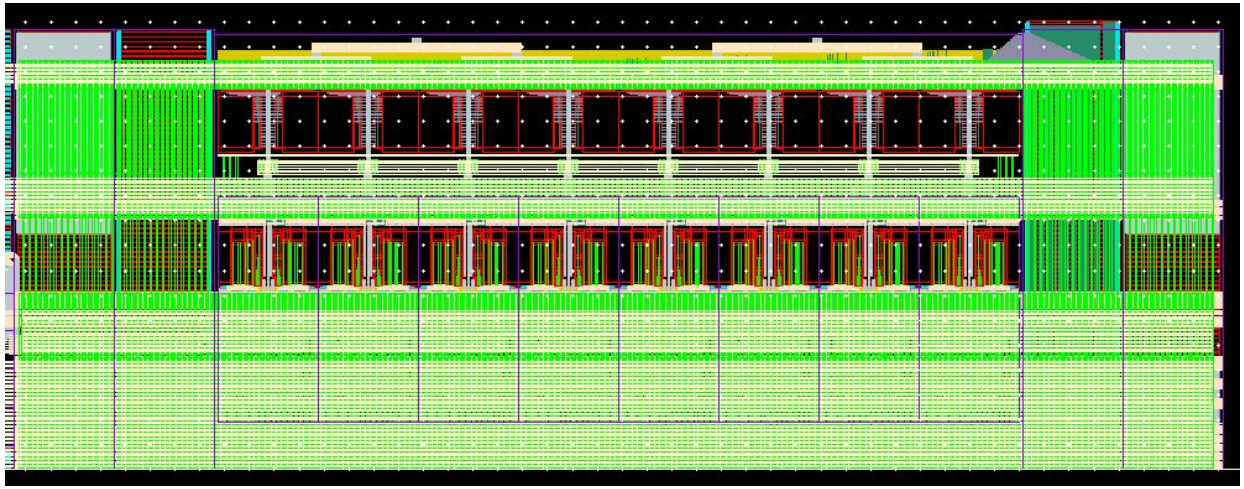


R-ladder DAC

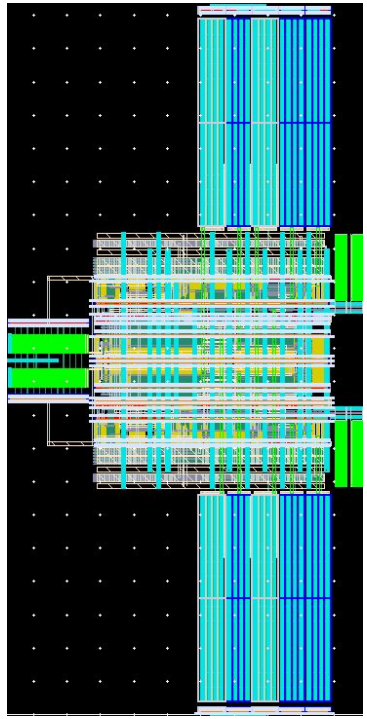


SAR ADC





Time-Interleaved SAR ADC

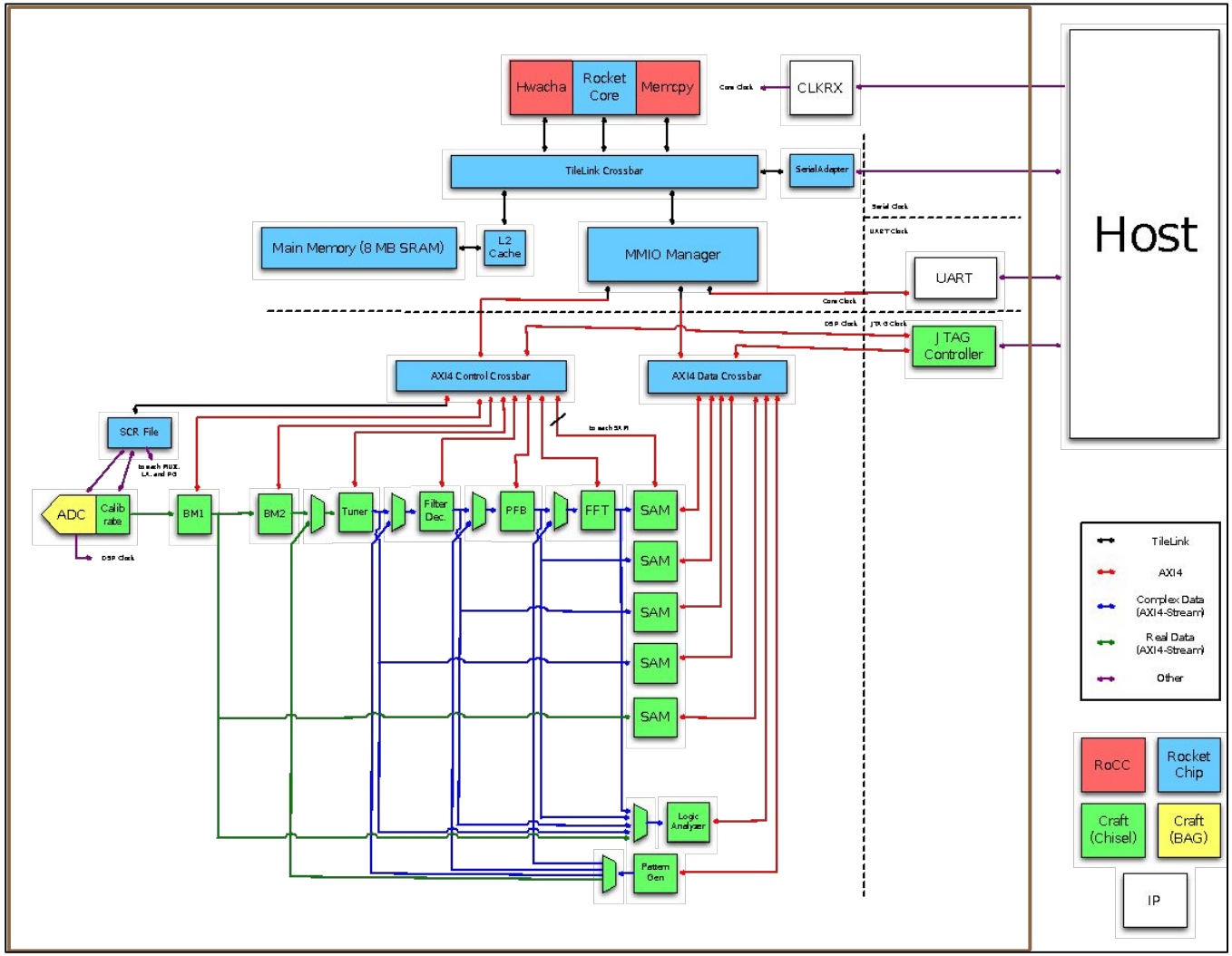


SerDes RX Front-End

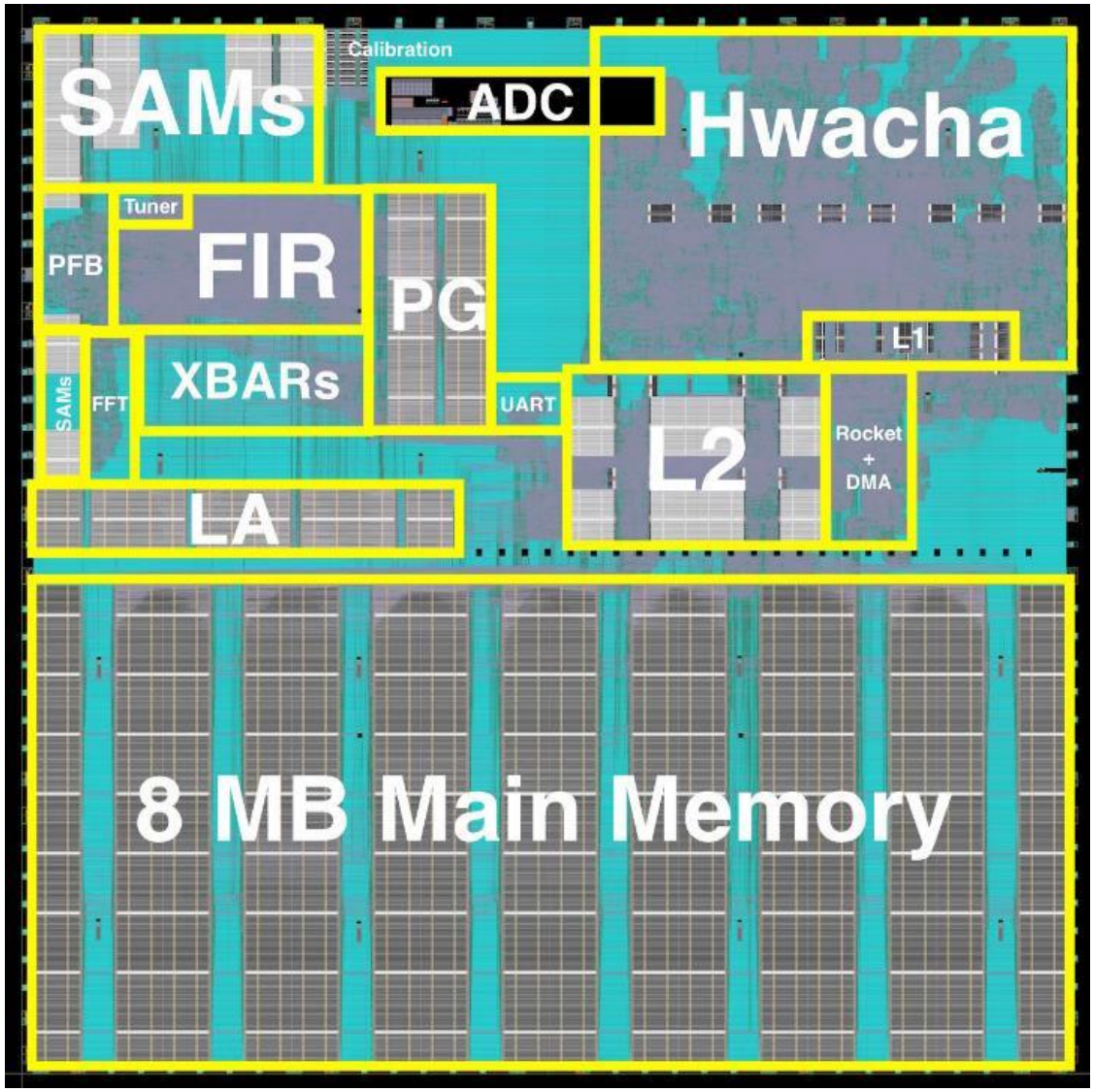
Radar Chip Specs

Chip Highlights:

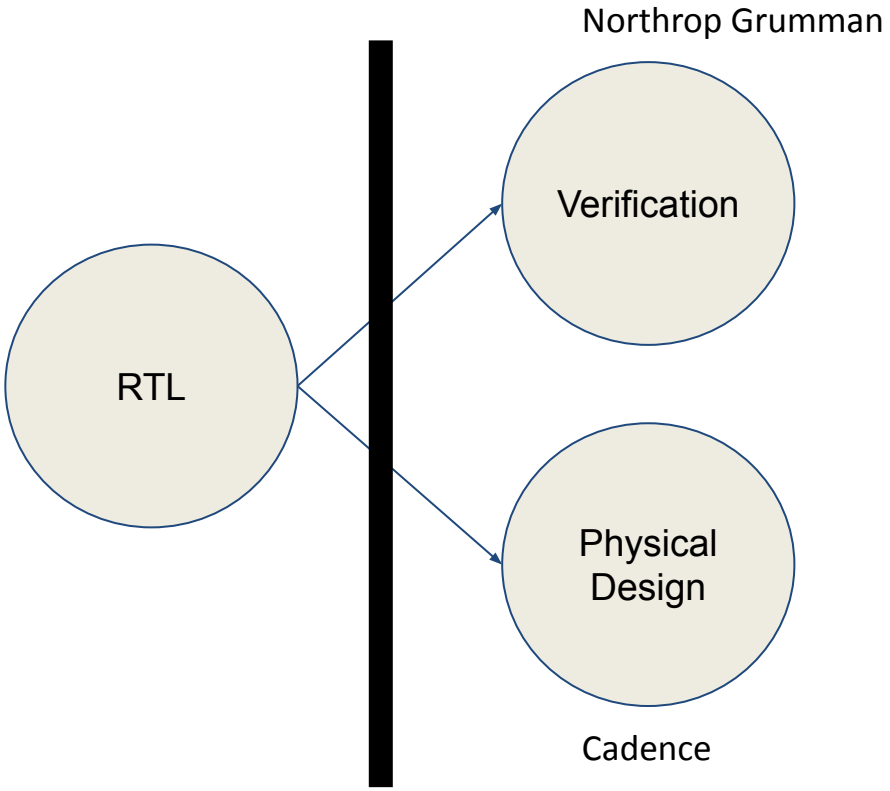
- 64-bit RISC-V processor
- 4-lane Hwacha vector unit
- DMA for DSP data movement acceleration
- 512 kB L2 cache
- 8 MB backing SRAM main memory
- 300 MHz DSP and core
- Custom IP integration (UART)
- JTAG backup controller
- Two verified AXI4 crossbars, 2x12 and 2x7
- Built-in debugging with 512 kB Logic Analyzer and 512 kB Pattern Generator
- 8-bit, 10 GHz time-interleaved SAR ADC with calibration
- Programmable digital tuner
- 136-tap programmable FIR filter
- 12-tap polyphase filter bank (PFB) with fixed coefficients
- 128-point FFT
- 5 stream-to-AXI4 memories (SAM) ranging in size from 64 kB to 256 kB
- Joint effort with Northrup Grumman Corporation (NGC) and Cadence Design Systems



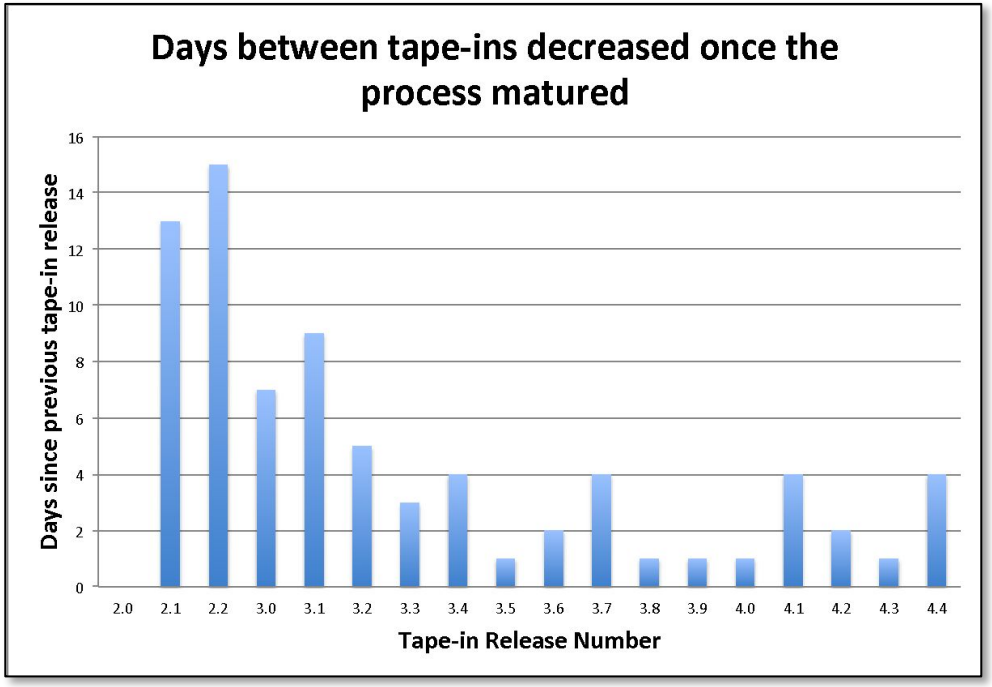
Radar Chip Layout



Agile Methodology



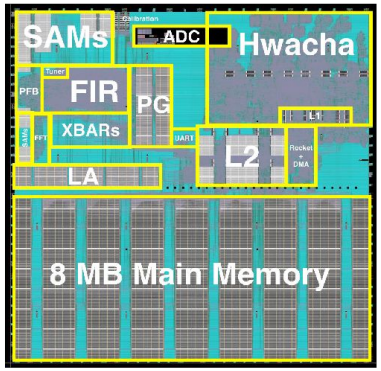
Tape-In



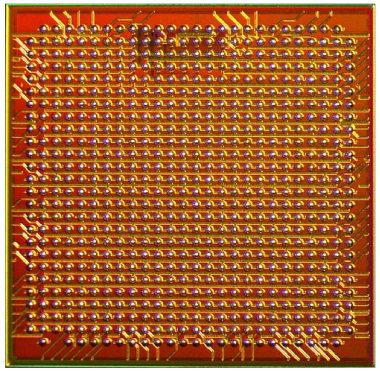
Agile Methodology

Signal Analysis SoC

~14,000 eng. hours



Annotated Layout

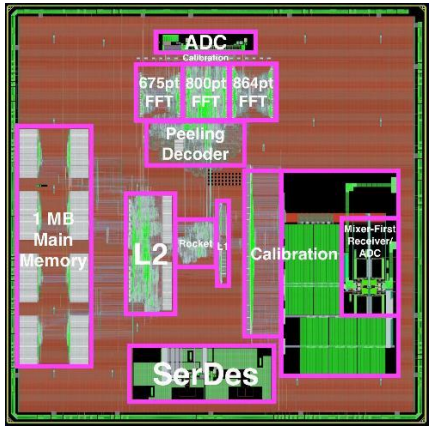


Die Photo

Technology	16nm FinFET	
Die Area	5mm x 5mm (25mm^2)	
	General-Purpose Processor	**Signal-Analysis Processor**
Area	1.1 mm^2 (gates) 9.2 mm^2 (SRAM)	1.5 mm^2 (gates) 0.8mm^2 (SRAM)
Total SRAM Size	71 Mbits	14 Mbits
Voltage	0.56 V - 0.98 V	0.56 V - 0.98 V
Max Frequency	410 MHz	417 MHz
Power	349 mW @ 0.75V, 410 MHz	210 mW @ 0.75 V, 417 MHz
Max Throughput (Mspectra/s)	0.46 (vector) 0.004 (scalar) @ 410 MHz	13 @ 417 MHz
Efficiency	23.4 GFLOPS/W (0.56V, 191MHz) (DGEMM on vector accelerator)	19.2 TOPS/W (0.56V, 192MHz) (1 op = 8-bit add ~ 17-bit mul)

Sparse FFT

~3,000 eng. hours



Summary and Lessons Learned

- Summary
  - Agile chip development is possible
  - We created a generalized DSP chip development methodology
  - We used this methodology to quickly realize multiple chips

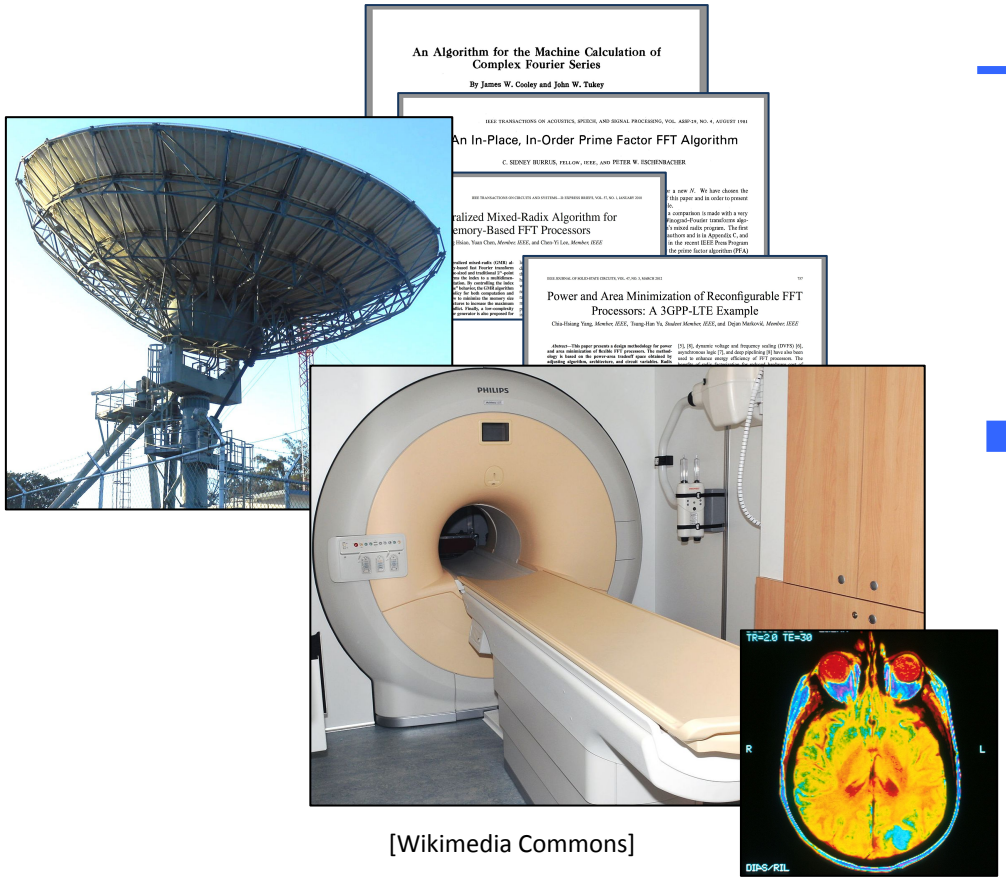
- Lessons Learned
  - Stay flexible
  - Automate and version control
  - Design for reuse
  - Physical design was the bottleneck

Angie Wang

DSPTools for Building & Verifying Hardware FFT Generators

FFTs Are Everywhere!

- Hardware FFTs implemented countless times…



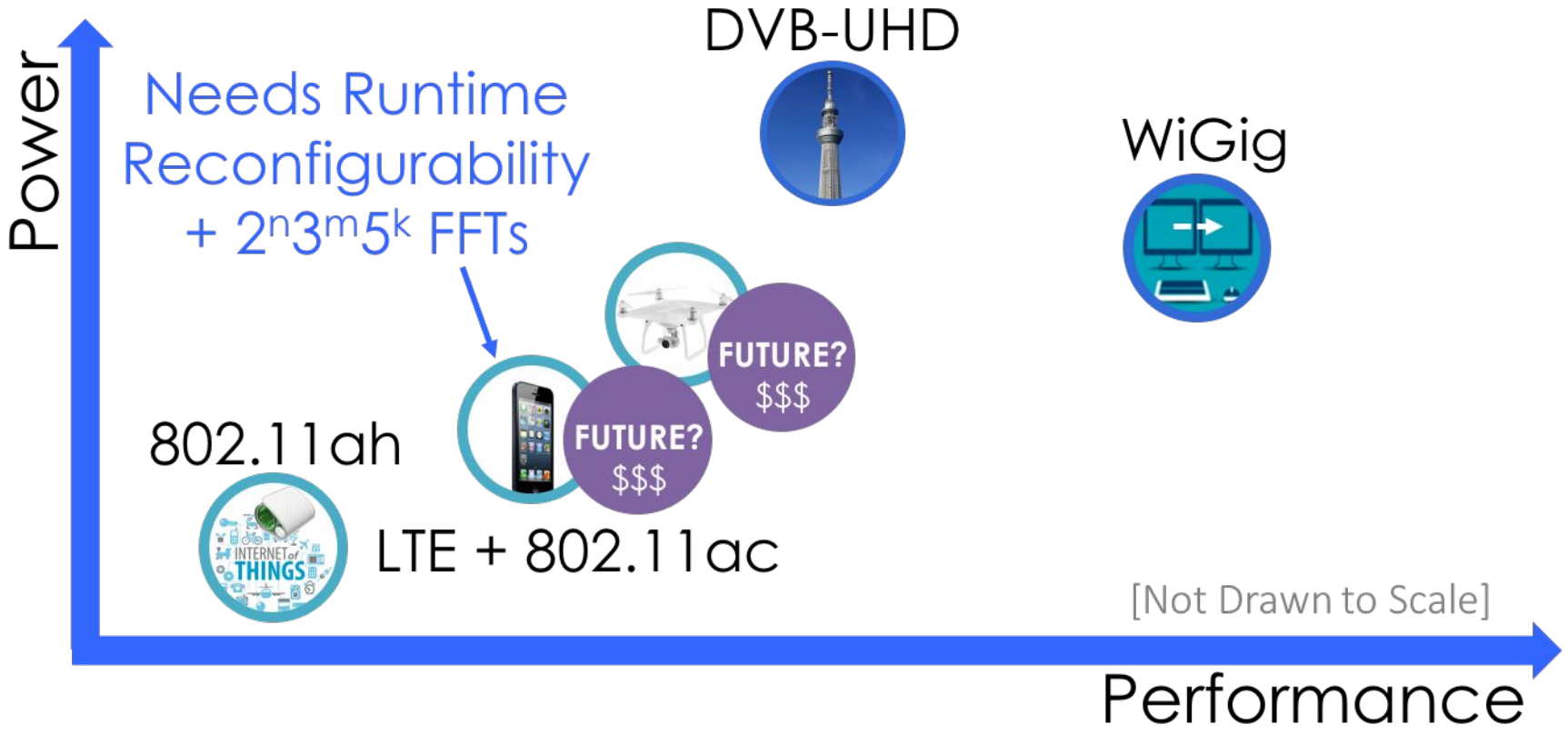
[Wikimedia Commons]

Core DSP in

- Radio Astronomy
- Audio Signal Processing
- MRI
- Cognitive Radio / SDR **5G+**
  - LTE, Wi-Fi, Future Standards
  - Dynamic Spectrum Mgmt.
- Object Detection
- *… You name it!*

Case Study: Wireless Communication

- How to generate a *hardware-optimized* building block for different standards?

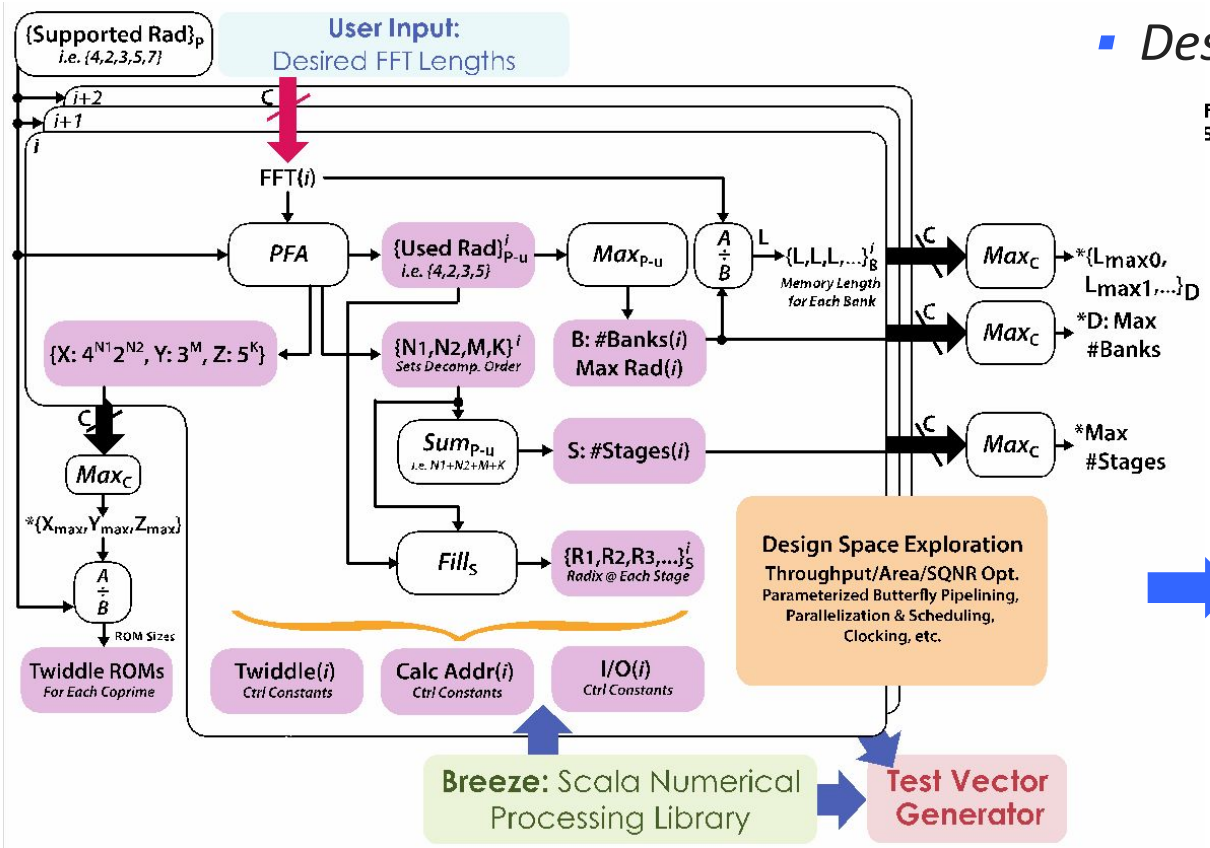


➡ One-off designs *(runtime reconfigurable or otherwise)* not resource + time-efficient!

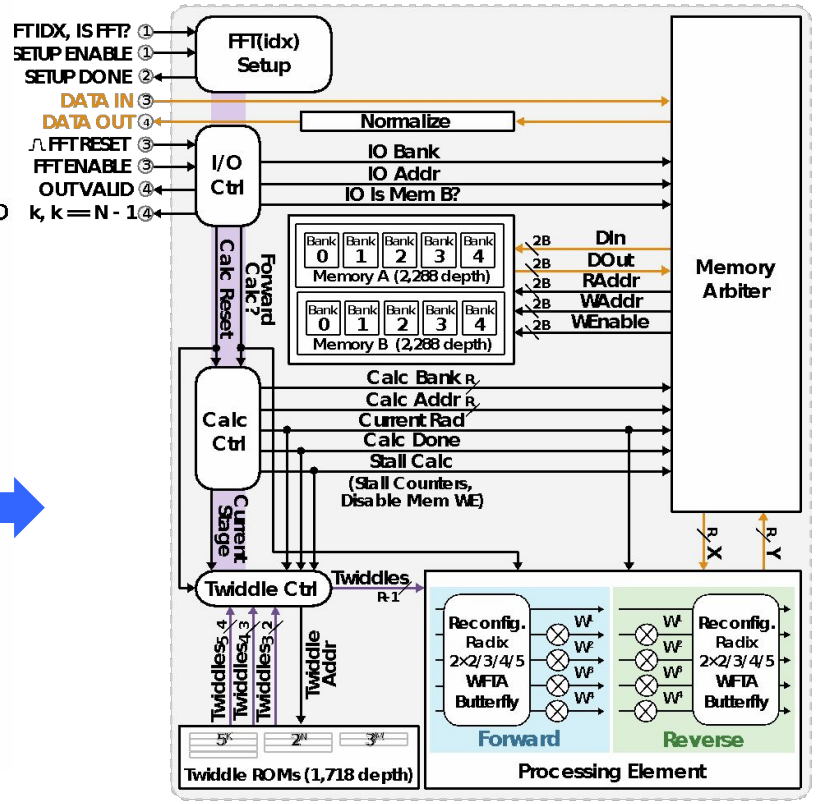
Mixed Radix, Runtime Reconfigurable FFT Generator

Scala "Firmware" ➡ **Chisel** Hardware Template

- Calculates constants for LUT generation
- Calculates optimized hardware params

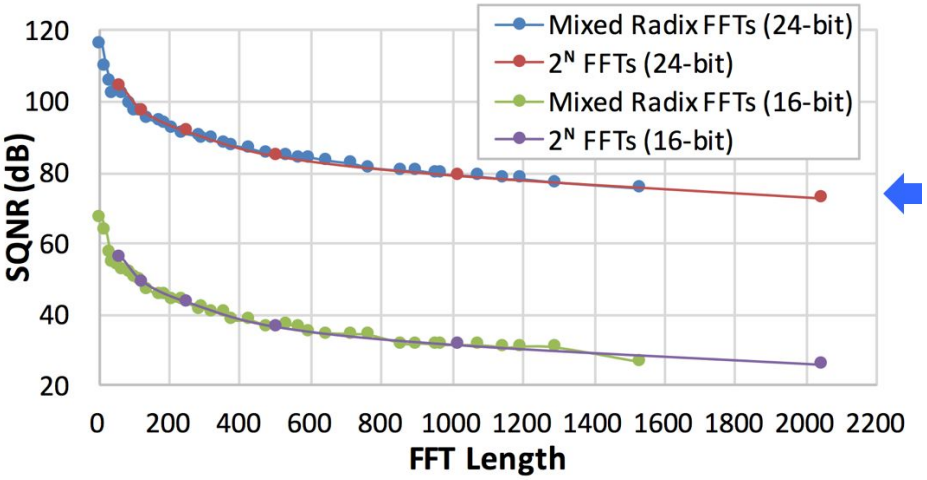


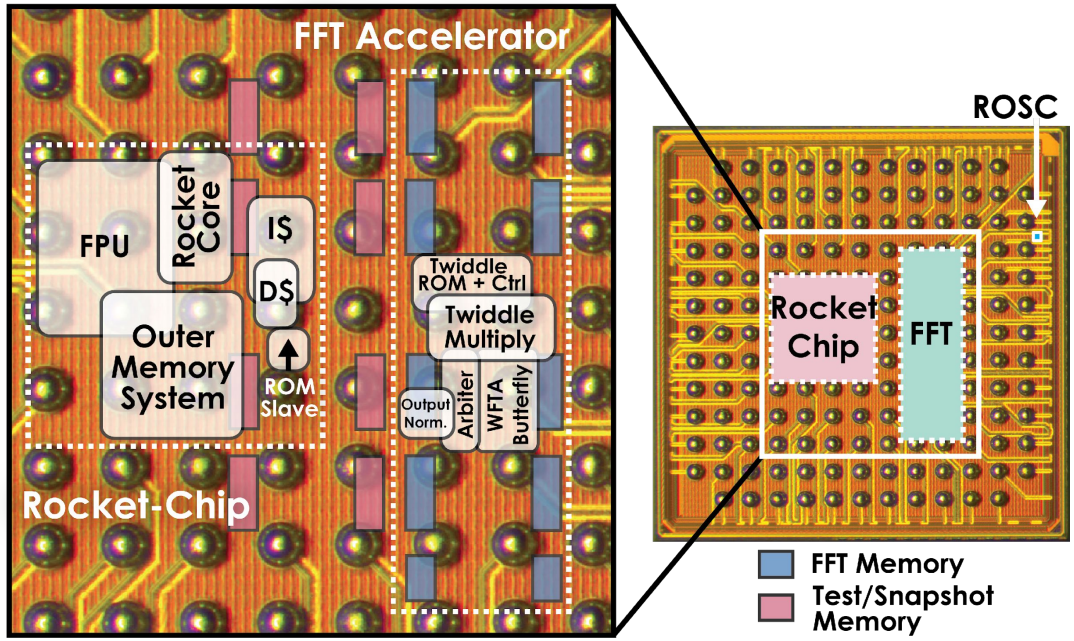
- LUTs for reconfiguration + twiddles
- Configurable blocks controlling dataflow between IO, SRAMs, & PE(s)
- *Design complexity mostly in ctrl logic*



A 0.37-mm^2 LTE/Wi-Fi Compatible FFT Accelerator Integrated with a RISC-V Core in 16-nm FinFET

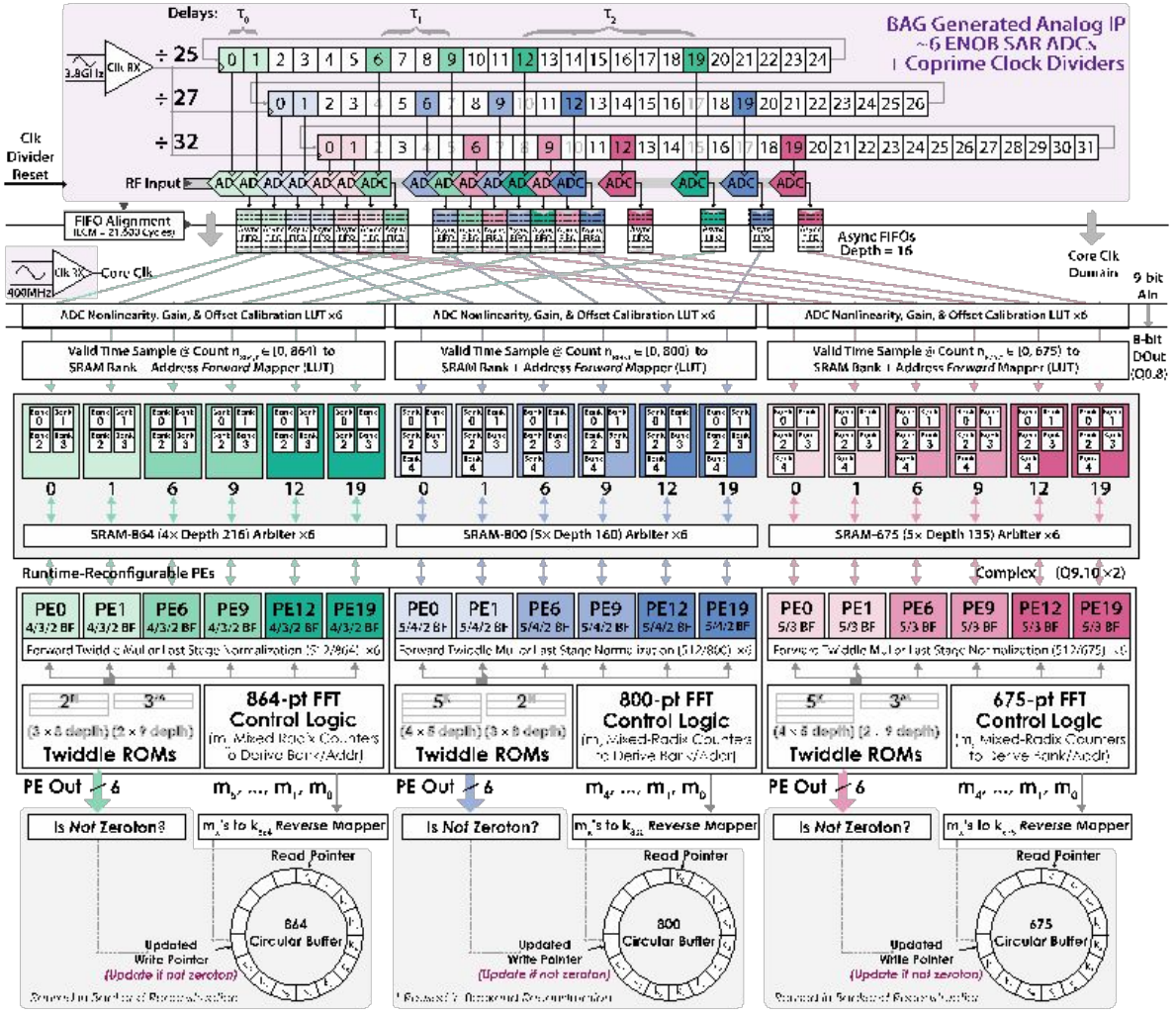
- Taped out in ~1 month from PDK delivery, with comparable performance to state-of-the-art
- Auto-generate C tests and run on Rocket-Chip

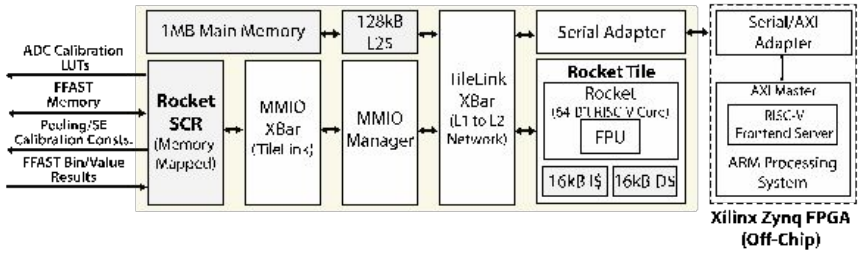


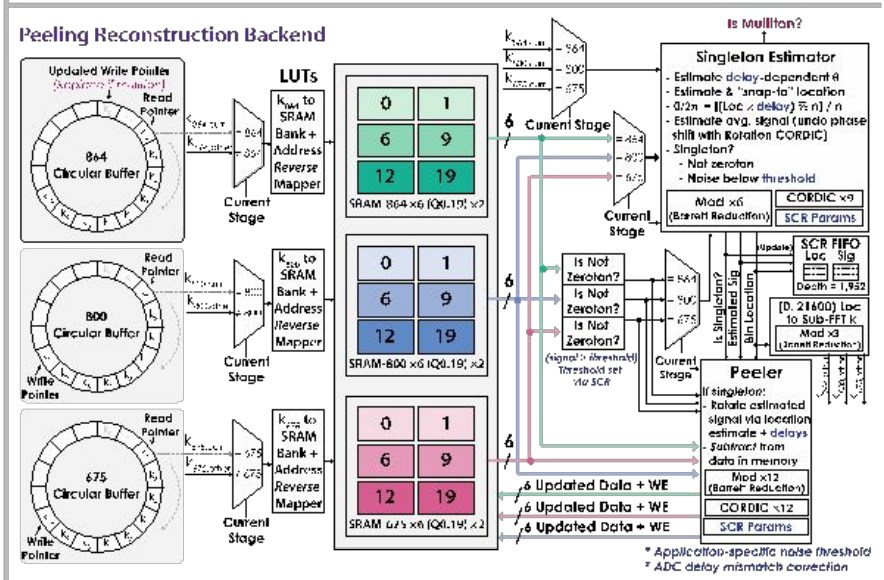


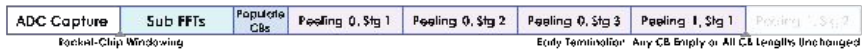
A Real-Time, 1.89-GHz Bandwidth, 175-kHz Resolution Sparse Spectral Analysis RISC-V SoC in 16-nm FinFET

- ADCs modeled in Chisel, designed with BAG
- Mixed-radix FFTs + peeling reconstruction backend designed with Chisel



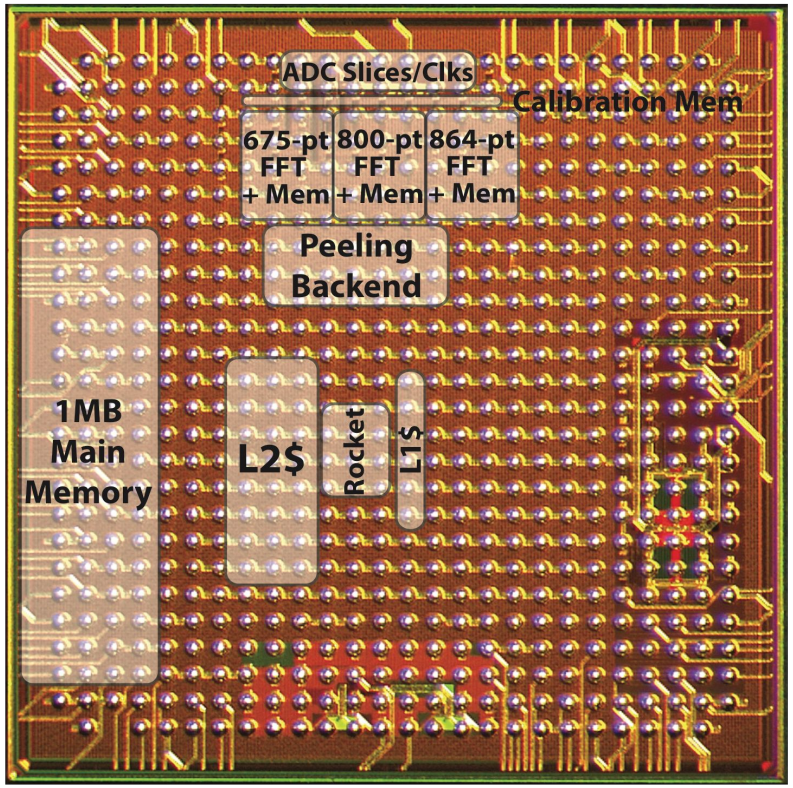


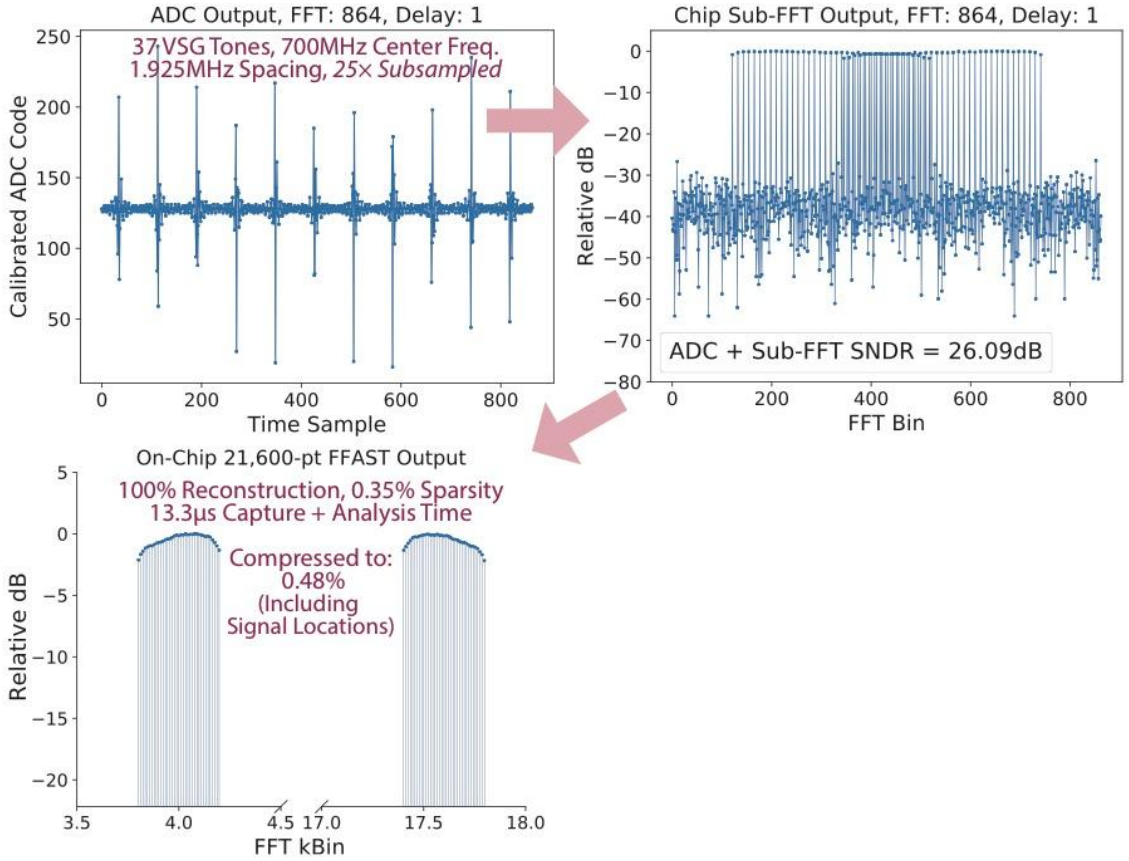




A Real-Time, 1.89-GHz Bandwidth, 175-kHz Resolution Sparse Spectral Analysis RISC-V SoC in 16-nm FinFET

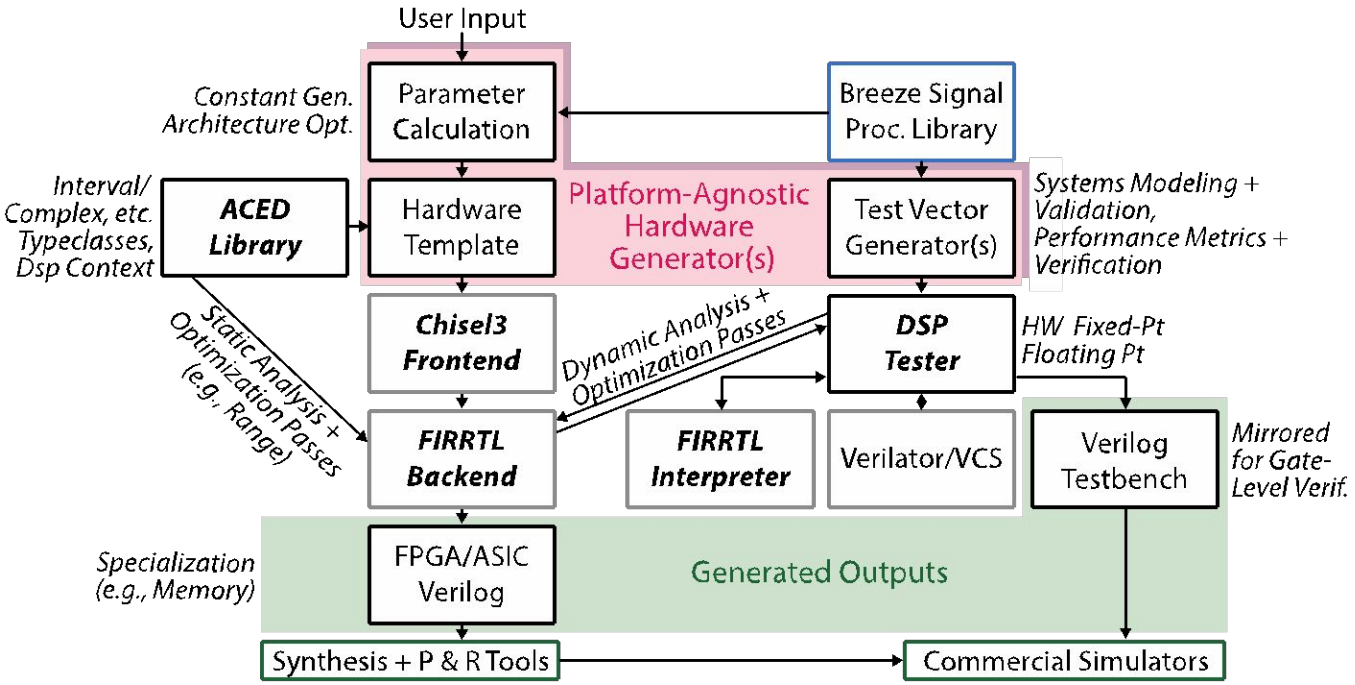
- Chisel/BAG generators enabled the first automatically generated, fully-integrated, sparse spectral analysis SoC, completed in ~2 months





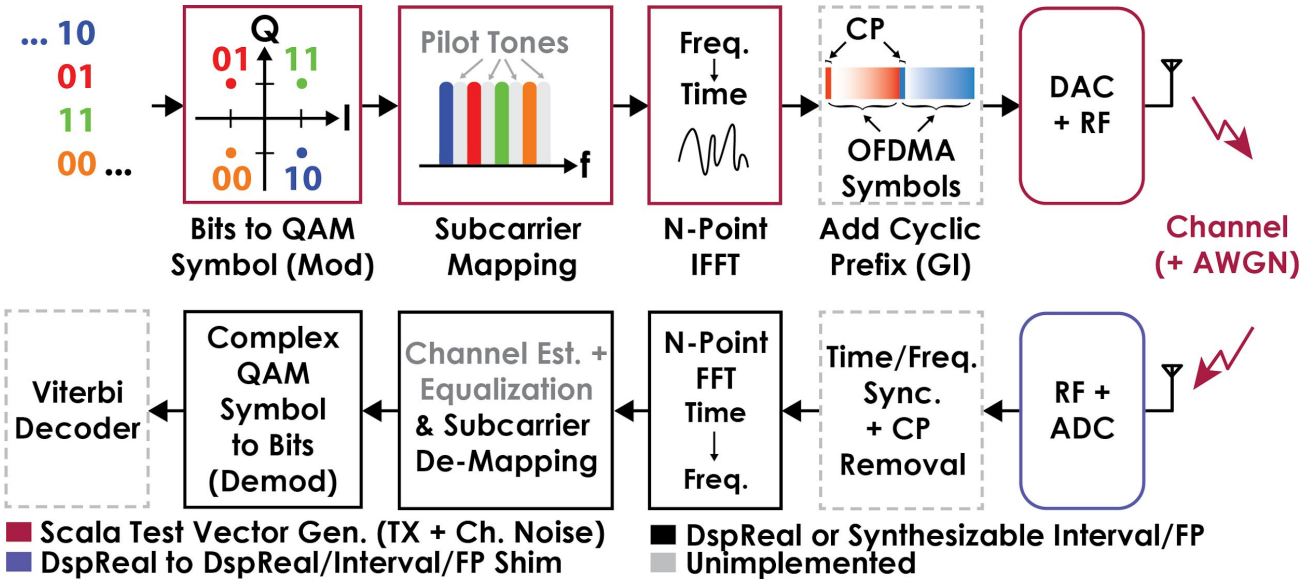
A Chisel Environment for DSP *Generator* Design

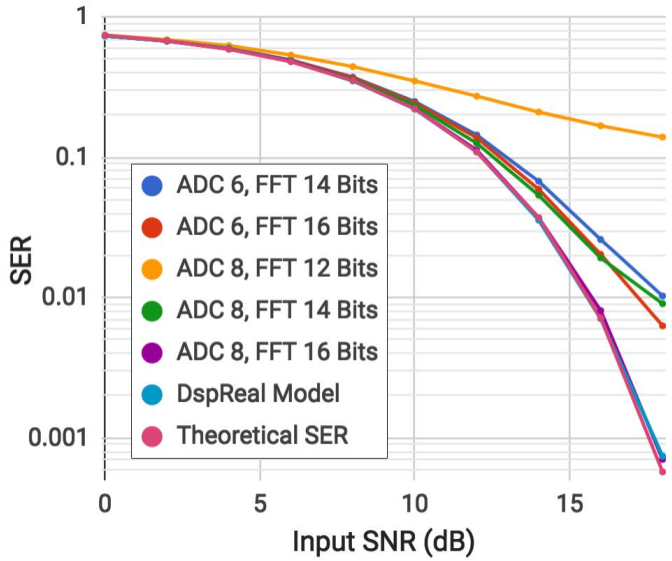
- **Capture and propagate designer intent down the hardware abstraction hierarchy without redundant specification!**
- Powerful meta-programming with zero-cost abstractions
  - Operator + data type parameterization
- Unobtrusive optimization and specialization
  - Automatic bitwidth reduction via static or simulation-based interval analysis
- Unified, yet portable modeling, validation, verification, and testing



Systems Modeling/Verification + Validation

- *DspReal* for verifying mathematical correctness/decoupling algorithmic errors from degraded performance due to quantization errors
  - Effective # of ADC bits
  - Fixed-point error propagation in the FFT, etc.
- Study system sensitivity to block performance via metrics like SER





- Use the same tests across all layers of the design stack to catch bugs

Summary

- The ACED DSP (aka DSPTools) library builds upon the Chisel ecosystem and vision.
- It operates at several *distinct* levels of abstraction.
  - High-level type-generic generators
  - Low-level bitwidth optimizations
- Tried-and-true techniques are brought immediately into the hands of the designer.
- Bitwidth optimization illustrates a powerful use case of FIRRTL's open-compiler framework in the context of writing DSP generators.
- Several chips have been fabricated using ACED