Generating the Next Wave of Custom Chips

Borivoje Nikolić

Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA
Diverse Driving Applications

- No single driving application
- Diversified set of applications and needs
  - Both clients and cloud

Borivoje Nikolić, Generating the Next Wave of Custom Chips
Cost of Developing New Products

“8,000 engineer-years to build Nvidia Xavier SoC”
= 16,000,000 hours

Source: IBS
Key Issues Driving Cost

- Dearth of re-use is the dominant problem
  - Lots of IP is out there
  - But that IP is largely “black-box”, hard to extend/modify
- Common modules are not commoditized
  - Value is in differentiation/specialization
- Approach: don’t deliver instances – capture designer methodology in generators!
  - Facilitates re-use via parameterization and incremental extension (of the generator – not the instance)
- Apply the same to verification
- Let’s generate systems!
Generators Enable Specialization

**Digital: Chisel3**
- CHISEL: Constructing Hardware In Scala Embedded Language
  - Open-source hardware construction language
  - Software library whose classes represent hardware primitives
  - Methods connect the classes together
  - So executing the software constructs a graph representing the RTL
- Compiles to FIRRTL
- Emits Verilog + collaterals
- v.3.1 is current
  J. Bachrach, *et al*, DAC 2012

**Analog: BAG2**
- Open-source Python-based framework allowing executable specification of design procedure
  J. Crossley, *et al*, DAC 2013

---

Borivoje Nikolić, Generating the Next Wave of Custom Chips

5
Generators Enable Agile Design

- Allows a small, integrated team to develop a design through a series of functional, yet incomplete prototypes
  - Design, verification and validation
  - Improve the generator, not the instance!

Y. Lee, IEEE MICRO’16
S. Bailey, JSSC 10/19
Berkeley RISC-V ISA

- www.riscv.org

- An open, license-free ISA
  - Runs GCC, LLVM, Linux distributions, ...
  - RV32, RV64, and RV128 variants for 32b, 64b, and 128b address spaces

- Base ISA only ~40 integer instructions

- Extensions provide full general-purpose ISA, including IEEE-754/2008 floating-point

- Designed for extension, customization

- Developed at UC Berkeley, maintained by RISC-V Foundation

- Most of cost of chip development is in software, so want to make sure software is reused across many chip designs

- Many research and commercial cores offered
  - Ariane/Pulpino
  - Rocket

- Rocket is an example of a generator written in Chisel
RISC-V Rocket Chip Generator

- Parametrizable SoC generator written in Chisel
- Processor core (Rocket), floating-point, cache, interconnect
- Standardized co-processor interface, ROcket Custom Co-processor (ROCC)
- [http://github.com/chipsalliance/rocket-chip](http://github.com/chipsalliance/rocket-chip)

CHIPS Alliance – A fund under Linux Foundation
Rocket Chip Customization

- Option 1: Change generator parameters

- Options:
  - Edit configs.scala
Rocket Chip Customization

- Option 1: Change generator parameters

**Options:**
- Edit `configs.scala`
- Add inclusive L2 cache
  - H. Cook, CARRV’19

Borivoje Nikolić, Generating the Next Wave of Custom Chips
Rocket Chip Customization

- Option 1: Change generator parameters

  - Options:
    - Edit configs.scala
    - Add inclusive L2 cache
      - H. Cook, CARRV’19
    - Cache size, organization
    - Number of cores
    - etc.
  - Target FPGA or ASIC
Rocket Chip Customization

- Option 2: Develop custom circuits

- Example:
  - 4KB single p-well 8T SRAM macro for low voltage operation in 28FDSOI

- Or your own technology
  - Processor-to-DRAM photonic link
Rocket Chip Customization

- Option 3: Develop a different RISC-V core

- ‘Standard’ Rocket core
  - 5-stage, in-order

- BOOMv2/v3
  - Out-of-order core

A. Gonzalez, CARRV’19
B(R)OOM Test Chip

- BOOMv2 out-of-order-core with cache Resiliency
  - 4 months to a tapeout

**Chip summary**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>2x3 mm²</td>
</tr>
<tr>
<td>Technology</td>
<td>TSMC 28nm HPM</td>
</tr>
<tr>
<td>ISA</td>
<td>RISC-V RV64IMAFD with Sv39</td>
</tr>
<tr>
<td>Fetch width</td>
<td>2 instructions</td>
</tr>
<tr>
<td>Issue width</td>
<td>4 micro-ops</td>
</tr>
<tr>
<td>Regfile</td>
<td>6R3W (int) 3R2W (fp)</td>
</tr>
<tr>
<td>Exe Unit</td>
<td>ALU, Mul, Div, FMA, Load/Store</td>
</tr>
<tr>
<td>L1 I/D Cache</td>
<td>4-way, 16KB</td>
</tr>
<tr>
<td>L2 Cache</td>
<td>8-way, 1MB</td>
</tr>
</tbody>
</table>

P.-F. Chiu, et al, SSC-L 2019
Rocket Chip Customization

- Option 4: Develop an accelerator/co-processor

- 4-lane vector co-processor
  - 4DP/8SP/16HP FP instructions
  - Dense and sparse linear algebra
  - Machine learning workloads

C. Schmidt, RISC-V Summit 12/2018

www.hwacha.org
Rocket Chip Customization

- Option 5: Develop a peripheral device
  - Attach via a TileLink2 or AXI interface
  - DSP accelerators, off-chip interfaces
  - Analog peripheral interfaces
  - Optional DMA
ChiselDSP

- Supports number-representation-agnostic generator design
  - Datatypes and associated operators can be real/complex, fixed/floating-point without rewriting any of the core generator code
Adding ChiselDSP to Rocket Chip: DspBlock

- Basic building block of DSP functionality
- Diplomatic interface
- Streaming inputs and outputs (any number)
- Optional memory interface
- Control and status registers
  - Maps to RocketChip

FireSim

- Cycle-exactly simulating large SoCs on cloud FPGAs @10s-100s of MHz
- Open-source: [https://fires.im](https://fires.im)
- Targets:
  1. Architecture evaluation
  2. Validate application on a pre-Si SoC

S. Karandikar, ISCA ’18, IEEE Micro TopPicks ’18, CARRV ’19

Example of (2): PageRank on Rocket+Hwacha

A. Amid, CARRV’19
Need Analog – BAG to the Rescue

- Design Flow Example: Top Level Overview

- Schematic/Layout generator
  - Produces schematic/layout from structural parameters

- Measurement Manager
  - Simulates and computes performance specifications of a given circuit instance

- Design Script
  - Contains algorithm used to generate instance from top level specifications
  - Or use ML (K. Hakhamaneshi, DAC’19)

E. Chang, CICC’18
Some Generators We’ve Built

Comparator

Switch-Cap DAC

R-ladder DAC

SAR ADC

Time-Interleaved SAR ADC

SerDes TX

SerDes RX
Parametrization and Process Portability

ADC Core

ST 28nm FDSOI
GF 22nm FDX
TSMC 16nm

SerDes RX Core (variable taps)

TSMC 16nm

SerDes RX Datapath

TSMC 16nm
GF 45nm RF PDSOI

Borivoje Nikolić, Generating the Next Wave of Custom Chips
Chisel+BAG-Generated RISC-V Chips

Raven, Hurricane: ST 28nm FDSOI, SWERVE: TSMC 28nm EOS: IBM 45nm SOI, CRAFT: 16nm TSMC,
Signal Analysis SoC in 16nm FFC

- Signal analysis SoC
  - 7 GS/s ADC
  - DSP chain: ADC cal, tuner, 136-tap FIR, 12-tap PFB, 128-pt FFT
  - Pattern generator, logic analyzer
  - AXI4 crossbar
  - RISC-V Rocket core
  - 4-lane vector unit
  - 8MB main memory
  - UART, serial

- Entire design in 14,000 hours (UCB, NGC, Cadence)
- Open source
  - [https://github.com/ucb-art/craft2-chip](https://github.com/ucb-art/craft2-chip)

S. Bailey, et al, A-SSC’18, JSSC 10/19
SoC Runs Applications…

Borivoj Nikolić, Generating the Next Wave of Custom Chips
...And is Process Portable

Port in 2,700 hours
(<20% of original)
Sparse FFT Chip

- Sparse FFT
  - 3 subsampling
  - 4GS/s ADCs, \( \div 25, \div 27, \div 32 \)
  - 3 FFTs:
    - 874-pt, 800-pt, 675-pt
  - Peeling decoder
  - Rocket core

3,000 hours

A. Wang, et al, ESSCIRC’18, JSSC, 07/19
SerDes Generator and Instance

- Generated a SerDes instance in TSMC 16nm @15 Gb/s
- All generated designs DRC and LVS clean without manual modifications

---

Borivoj e Nikolić, Generating the Next Wave of Custom Chips
ADC Generator Architecture

- CLK generator, SARADC array, retimer
- Laygo flow: ADC slice => array => CLK gen => retimer
- A lot of design options
  - Clock pulse-width, source follower, sampler topology, comparator topology, Asynch. clock speed/configuration, body-biasing...

![Diagram of ADC Generator Architecture](image)

Borivoje Nikolić, Generating the Next Wave of Custom Chips
Design Flow

- LayGo engine of BAG
TISARADC Instance in TSMC16FFC

- 38.2-dB SNDR at 7GS/s, 45mW
- Generated analog design can compete with state-of-the-art custom designs!
Summary and What’s Next

- Generator-based approach facilitates re-use and agile execution
  - Just like other software ecosystems, opportunities for reuse increase as available libraries (generators) expand
  - The ecosystem is expanding – RISC-V, RocketChip, Chisel, BAG, FireSim
  - We are working on generators for SoCs, multiprocessors, radars, radios, navigation, SerDes, data converters, PLLs, RF T/RX, …
  - And building the design flow generator (architecture->layout)
    - Possibly including Federation tools, H. Cook, CARRV’19

- Interested?
  - Chisel “bootcamp”: [https://github.com/freechipsproject/generator-bootcamp](https://github.com/freechipsproject/generator-bootcamp)
  - BAG “bootcamp”: [https://github.com/ucb-art/BAG2_cds_ff_mpt](https://github.com/ucb-art/BAG2_cds_ff_mpt)
  - FireSim: [https://fires.im](https://fires.im)
  - New ‘one-stop shop’ repository of all Berkeley designs coming up…
Acknowledgments

- Faculty: E. Alon, K. Asanovic, J. Bachrach, V. Stojanovic, D. Patterson
- Staff: B. Richards, C. Markley, J. Lawson, J. Dunn
- Students (current and past): Brian Zimmer, Yunsup Lee, Ben Keller, Paul Rigge, Angie Wang, Stevo Bailey, Eric Chang, Pi-Feng Chiu, Colin Schmidt, John Wright, Martin Cochet, Albert Ou, Howard Mao, Woorham Bae, Jaeduk Han, Andrew Waterman, Henry Cook, Christopher Celio, Adam Izraelevitz, Zhongkai Wang, Sean Huang, Zhaokai Liu, Sagar Karandikar, Alon Amid, Nathan Narevsky, Jaehwa Kwak, Donggyu Kim, David Biancolin, Jack Koenig,…
- DARPA CRAFT and PERFECT programs
- STMicroelectronics, TSMC chip donations, fabrication