Michael B. Taylor
Professor
UC San Diego

Pulkit Bhatnagar
Scott Davidson
Anuj Rao
Luis Vega
Shaolin Xie
Chun Zhao

Bespoke Silicon Group (BSG)
Startup Software Stacks Today

Innovation Layer: Company’s Value Add
Enormous leverage
Instagram: $500K & 13 people → $1B startup

Open Source
Python
Django
Memcached
Postgres/SQL
Redis
Apache
Linux
GNU *
GCC
ASICs: Where is the Open Source?

Closed Source ($$$)
- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

Open Source

Innovation Layer

Software
Building the Fully Open Source ASIC Stack

I have a dream: git->make->gds

The HW Stack
- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

Replace with RISC-V Ecosystem
Building the Full Open Source ASIC Stack

The HW Stack
ARM A57, A7, M4, M0...
ARM Interconnect
IO Pads
Standard Cells
PLL
High-Speed I/O
Design Compiler
IC Compiler
Spice
Formality
Calibre DRC/LVS
BGA Package
PCB Design
Firmware

Replace with RISC-V Ecosystem

For 65nm and above, generally free
But open source would still be better ...
NDAs create large barriers to open sourcing
of semi-custom blocks
Building the Full Open Source ASIC Stack

The HW Stack
- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

- Replace with RISC-V Ecosystem
- For 65nm and above, generally free
- 50K

Without this, it’s hard to get a > 200 MHz clock into the chip
Building the Full Open Source ASIC Stack

**The HW Stack**

- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

- Replace with RISC-V Ecosystem
- For 65nm and above, generally free
- 50K
- 100K-$1M

*Most high-performance ASICs need high-performance I/O*
Building the Full Open Source ASIC Stack

The HW Stack
- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

Replace with RISC-V Ecosystem

For 65nm and above, generally free
- 50K
- 100K-$1M

Free-ish for academia, $250K-$1M for others

Even for academics, serious problem because it prevents sharing of flows and prevent reuse between organizations
Building the Full Open Source ASIC Stack

The HW Stack
ARM A57, A7, M4, M0...
ARM Interconnect
IO Pads
Standard Cells
PLL
High-Speed I/O
Design Compiler
IC Compiler
Spice
Formality
Calibre DRC/LVS
BGA Package
PCB Design
Firmware

Replace with RISC-V Ecosystem
For 65nm and above, generally free
50K
100K-$1M
Free-ish for academia, $250K-$1M for others
50K
Required to get high-performance I/O off chip & good VDD/VSS
Typical packages are highly specific to a given piece of silicon
requires a rethink of how packages are designed;
we need both open source AND reusability.
Building the Full Open Source ASIC Stack

The HW Stack
- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

Replace with RISC-V Ecosystem
For 65nm and above, generally free
50K
100K-$1M
Free-ish for academia, $250K-$1M for others
50K
50K-300K+ (labor)

Typical PCB/firmware is specific to a given piece of silicon requires a rethink of PCBs.
We need open source AND reusability.
Building the Full Open Source ASIC Stack

The HW Stack
- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

Replace with RISC-V Ecosystem
- For 65nm and above, generally free
- 50K
- 100K-$1M

Free-ish for academia, $250K-$1M for others
- 50K
- 50K-300K+ (labor)

Required to get high-performance I/O off chip.
Typical PCB/firmware is specific to a given piece of silicon requires a rethink of PCBs.
We need open source AND reusability.

How about some standardization of sockets and motherboards for RISC-V HW?
Dare to dream: How about a std RISC-V chipset for IPs that are only commercially available?
### Our project: 100% open source (or free*)

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The HW Stack</strong></td>
<td></td>
</tr>
<tr>
<td>ARM A57, A7, M4, M0…</td>
<td>Replace with RISC-V Ecosystem</td>
</tr>
<tr>
<td>ARM Interconnect</td>
<td></td>
</tr>
<tr>
<td>IO Pads</td>
<td>For 65nm and above, generally free</td>
</tr>
<tr>
<td>Standard Cells</td>
<td></td>
</tr>
<tr>
<td>PLL</td>
<td>50K</td>
</tr>
<tr>
<td>High-Speed I/O</td>
<td>100K-$1M</td>
</tr>
<tr>
<td>Design Compiler</td>
<td></td>
</tr>
<tr>
<td>IC Compiler</td>
<td></td>
</tr>
<tr>
<td>Spice</td>
<td>Free-ish for academia, $250K-$1M for others</td>
</tr>
<tr>
<td>Formality</td>
<td></td>
</tr>
<tr>
<td>Calibre DRC/LVS</td>
<td>50K</td>
</tr>
<tr>
<td><strong>BGA Package</strong></td>
<td>50K</td>
</tr>
<tr>
<td><strong>PCB Design</strong></td>
<td>50K-300K+ (labor)</td>
</tr>
<tr>
<td><strong>Firmware</strong></td>
<td></td>
</tr>
</tbody>
</table>
Basejump: A “Base Class” for Cheap HW Development

Basejump “Socket”

http://bjump.org

Innovation Layer

Taylor BSG
Basejump: A “Base Class” for Cheap HW Development

Basejump “Socket”

bsg_ip_cores
standard library components
for SystemVerilog; raising the level of abstraction

Innovation Layer

http://bjump.org

Taylor BSG
Basejump: A “Base Class” for Cheap HW Development

**Basejump “Socket”**

- **bsg_ip_cores**
  - standard library components for SystemVerilog

- Innovation Layer

- CAD Flow + Fab

[http://bjump.org](http://bjump.org)
Basejump: A “Base Class” for Cheap HW Development

**Basejump “Socket”**

**bsg_ip_cores**
standard library components for SystemVerilog

**Basejump BGA Package**

CAD Flow + Fab

Innovation Layer

http://bjump.org
Basejump: A “Base Class” for Cheap HW Development

Basejump “Socket”

bsg_ip_cores
standard library components for SystemVerilog

Innovation Layer

CAD Flow + Fab

Basejump BGA Package

Basejump Motherboard

http://bjump.org
Basejump: A “Base Class” for Cheap HW Development

Basejump “Socket”

bsg_ip_cores
standard library components
for SystemVerilog

CAD Flow + Fab

Basejump Motherboard

Basejump BGA Package

Innovation Layer

Basejump Open FPGA Firmware

Click!!

Xilinx FPGA Zedboard

http://bjump.org
## bsg_ip_cores: Like C++ STL but for SystemVerilog

The entire SoC framework is composed out of these blocks

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsg_misc</td>
<td>popcount, flop trays, decoders, lfsr, multiplies, flexible muxes, transposers</td>
</tr>
<tr>
<td></td>
<td>crossbars, gray_to_binary, priority encoder, thermometer encoders, counters</td>
</tr>
<tr>
<td>bsg_async</td>
<td>asynchronous fifos and interfaces</td>
</tr>
<tr>
<td>bsg_clk_gen</td>
<td>synthesizable digital clock generator</td>
</tr>
<tr>
<td>bsg_comm_link</td>
<td>high-speed I/O source-synchronous interface</td>
</tr>
<tr>
<td>bsg_dataflow</td>
<td>FIFOs, stream mergers, round-robin arbitrators, serial-to-parallel converters</td>
</tr>
<tr>
<td>bsg_fsb</td>
<td>front side bus (high-speed bridge between off-chip and on-chip worlds)</td>
</tr>
<tr>
<td>bsg_mem</td>
<td>portability layer for SRAMs</td>
</tr>
<tr>
<td>bsg_mesosync</td>
<td>mesosynchronous I/O library (high_speed + low latency)</td>
</tr>
<tr>
<td>bsg_noc</td>
<td>network-on-chip building blocks</td>
</tr>
<tr>
<td>bsg_riscv</td>
<td>RISC-V interface logic</td>
</tr>
<tr>
<td>bsg_tag</td>
<td>SoC configuration interface (like SPI or JTAG)</td>
</tr>
<tr>
<td>bsg_test</td>
<td>Test bench blocks; reset generators, delay lines, clock gens</td>
</tr>
</tbody>
</table>
Basejump: Early Adopters

- PRINCETON UNIVERSITY
- ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
- Massachusetts Institute of Technology
- UNIVERSITY OF CAMBRIDGE
- UCSB
- UCSD
- UCLA
- UNIVERSITY OF VIRGINIA
- TANDON SCHOOL OF ENGINEERING
- UNIVERSITY OF MICHIGAN
- DARPA CRAFT (16nm)
- NSF SaTC Med (Security)
- NSF SaTC Large (Crypto)
BSG Ten: RISC-V 10-core multicore

100% of BSG Ten design will be released after chip bringup.

End-to-end open source:

All design files for chip, PCB, BGA Package, Firmware..
BSG Ten: RISC-V 10-core multicore

**BSG Tile**
- BSG Vanilla Core
- Mesh Router (X)
- Remote Store
- Programming Model

**BSG Vanilla**
- RV32IM Core
  - 5-stage pipeline
  - 4KB DMEM (D)
  - 4KB IMEM (I)

Higher Performance than V-scale at extremely low area and design size.

1 core = 2 x 0.6 mm (>50% SRAM)
BSG Ten: RISC-V 10-core multicore

Tile internally uses many components from bsg_ip_cores
BSG Ten: RISC-V 10-core multicore

Tile internally uses many components from bsg_ip_cores

bsg_ip_cores clock generator
BSG Ten: RISC-V 10-core multicore

Tile internally uses many components from bsg_ip_cores

bsg_ip_cores clock generator

bsg_comm_link bsg_fsb

Tile internally uses many components from bsg_ip_cores
BSG Ten vs. MIT Raw

BSG X
‘180 nm
25 mm² die
~400 MHz

MIT Raw
180 nm
331 mm² die
~400 MHz
16 cores

Tile is 10X smaller than MIT Raw in the same node!

No FP, 16X less ram, fewer networks, no caching,...
CERTUS: TSMC 16nm SoC Design
CERTUS: TSMC 16nm SoC Design

- Rocket Tile 2
- Rocket Tile 1
- RISC-V Processor
- I-Cache
- D-Cache
- L2
- L2
- Arbiter
- ROCC Accelerator Interface
- Manycore Sub-System
- Uncore
- NoC Router
- Mem
- XBAR
- Mem
- NoC Router
- Mem
- XBAR
- Mem
- XBAR
- Mem
- XBAR
- Mem
- XBAR
- Mem

Taylor BSG
CERTUS: TSMC 16nm SoC Design

- RISC-V Processor
- I-Cache
- D-Cache
- Arbiter
- AXI
- Peripherals
- Debug Config
- L2
- L2
- ROCC Accelerator Interface
- Rocket Tile 1
- Rocket Tile 2
- BNN Accelerator
- Accelerator Suite
- Northbridge
- Southbridge
- DDR-3
- Gig-E
- Video In
- Video out
- DEBUG
- FLASH
CERTUS: TSMC 16nm SoC Design

Manycore Sub-System

ROCC Accelerator Interface

Rocket Tile 2

BNN Accelerator

Accelerator Suite

Basejump IO

Uncore

AXI

Peripherals
Debug
Config

RISC-V Processor

I-Cache

D-Cache

L2

L2

Arbiter

AXI

High-speed Offchip Network Link

Basejump Motherboard

Video In

Video out

DDR-3

Northbridge

Southbridge

DEBUG

FLASH

Synthesizable DC/DCs

Synthesizable PLLs

Southbridge

Gig-E
RISC-V @ UCSD: Some docu-contributions

- *The RoCC Doc V2:*
  *An Introduction to the Rocket Custom Coprocessor Interface (Anuj Rao)*
  [http://goo.gl/dHwK4n](http://goo.gl/dHwK4n)

- Rocket / accelerator integration instructions *(Anuj Rao)*
  [http://bitbucket.org/taylor-bsg/bsg_riscv](http://bitbucket.org/taylor-bsg/bsg_riscv)
  [http://goo.gl/cv27eI](http://goo.gl/cv27eI)
The ROCC interface should be wired to the toplevel of the Rocket hierarchy.

- Accelerators stay out of the rocket hierarchy
- Allows accelerators to attach to devices at the top level

RISC-V community should target at-least-yearly end-to-end releases of entire RISC-V stacks (Linux to Verilog) to reduce the effort of RISC-V participants to find consistent versions of infrastructure.

- Mismatches raises the bar to use/modify the rocket infrastructure
  - Today, basic users of Rocket need to full-stack experts to resolve problems
    - We need to reduce the scope of knowledge needed to contribute
  - If the only good version is “top”, then everybody is fighting bugs on their own with their particular versions.
- “Perfect is the enemy of the done”
Building the Full Open Source ASIC Stack

The HW Stack
- ARM A57, A7, M4, M0...
- ARM Interconnect
- IO Pads
- Standard Cells
- PLL
- High-Speed I/O
- Design Compiler
- IC Compiler
- Spice
- Formality
- Calibre DRC/LVS
- BGA Package
- PCB Design
- Firmware

BSG Ten
- bsg_clk_gen
- bsg_comm_link

Basejump BGA
Basejump RealTrouble Motherboard