efabless’ Raven: PicoRV32 on an ASIC, Open Source, Open Silicon Design

Mohamed Kassem
Tim Edwards
efabless.com

RISC-V Workshop Zurich
June 11-13, 2019
ETH Zurich
KEY QUESTIONS

Can we design a market-relevant (with Analog) open source SoC with proprietary foundry ecosystem? *(most open source HW has no analog)*

Is there a complete open source design flow that can deliver robust SoC’s? *(maybe targeted to specific applications)*

Can we design and verify a working microprocessor SoC in < 3 months and validate first-time silicon success? *(if you don’t boil the ocean)*

Can we facilitate effective collaboration across knowledge domains?
RAVEN - 32-bit RISC-V MICROCONTROLLER

Key Features

- RISC-V CPU (PicoRV32)
- SRAM 32x1024
- 100 MHz clock rate
- Programmable clock source
- 16 channels GPIO
- 2 ADCs
- 1 DAC
- 1 Comparator
- Over-temperature alarm
- 100 kHz RC oscillator
- Programmable functions on GPIO outputs
- Programmable interrupts on GPIO inputs

http://github.com/efabless/raven-picorv32
KEY REQUIREMENTS

All software, firmware, and hardware to be open source.

Chip must demonstrate function of a set of digital and analog IP.

Test board PCB design, BOM, and USB driver to be open source.

Design is a reference design to be customized as needed.

Finished design is market-relevant *(can be commercialized)*.
RAVEN CPU CORE

The PicoRV32 RISC-V core by Clifford Wolf

Fully open source under generous license on github

Packaged with a reference SoC implementation with UART and SPI flash driver

Packaged with instructions for obtaining and installing the RISC-V gcc cross-compiler (for RV32IMC)

Packaged with example C code and testbenches
FOUNDRY CELLS & ANALOG IP

The target process: X-FAB XH018
Base MOS LP (low power) option
6 metal stack (5 standard route layers, 1 thicker top metal)

The proprietary data:
X-FAB digital standard cells
X-FAB I/O Cells(3.3V with both 3.3V and 1.8V core)
X-FAB Analog IP
X-FAB SRAM (from memory compiler)

Do we really need the process technology to be open for most designs?
Mask-geometry layout is foundry proprietary.

How can you design an entire chip and submit to the foundry for fabrication without signing an NDA, purchasing commercial tools, and installing PDKs?

All analog cells at the transistor level are abstracted views using information from the corresponding LEF files and simulation models.
ANALOG IP ABSTRACTION

Assembled LEF Data in Magic

GDSII except for SRAM
EFX3201RV
32-bit RISC-V Microcontroller for Embedded Applications

Key Features
- RISC-V CPU (PicoRV32)
- SRAM 32x1k24
- 100 MHz clock rate
- Programmable clock source
- 16 channels GPIO
- 2 ADCs
- 1 DAC
- 1 Comparator
- Over-temperature alarm
- 100 kHz RC oscillator
- Programmable GPIO outputs
- Programmable interrupts on GPIO inputs

Pin Configuration

Block Diagram
RISC-V Cores

<table>
<thead>
<tr>
<th>Name</th>
<th>Supplier</th>
<th>Links</th>
<th>Core</th>
<th>ISA</th>
<th>OS</th>
<th>Devkit</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE310-G000</td>
<td>SiFive</td>
<td>Datasheet</td>
<td>E31</td>
<td>RV32IMAC</td>
<td>RTOS</td>
<td>HiFive1</td>
</tr>
<tr>
<td>FE310-G002</td>
<td>SiFive</td>
<td>Datasheet</td>
<td>E31</td>
<td>RV32IMAC</td>
<td>RTOS</td>
<td>HiFive1 Rev B</td>
</tr>
<tr>
<td>Freedom U540</td>
<td>SiFive</td>
<td>Product page</td>
<td>U54 (4 cores), E51 (1 management core)</td>
<td>RV64GC (application cores), RV64IMAC (management core)</td>
<td>Linux</td>
<td>HiFive Unleashed development board</td>
</tr>
<tr>
<td>GAP8</td>
<td>GreenWaves Technologies</td>
<td>Product page</td>
<td>PULP / 1 + 8 RISCY</td>
<td>RV32IMC (+ Privileged and custom ISA extensions)</td>
<td>RTOS</td>
<td>GAPuino development board</td>
</tr>
<tr>
<td>K210</td>
<td>Kendryte</td>
<td>Product page, Datasheet, GitHub</td>
<td>K210</td>
<td>RV64GC</td>
<td>RTOS</td>
<td>KD233 development board</td>
</tr>
<tr>
<td>RV32M1</td>
<td>NXP</td>
<td>Reference Manual and Datasheet</td>
<td>RISCY + Zero RISCY + Arm Cortex M4F + Arm Cortex M0+</td>
<td>RV32IMC</td>
<td>RTOS</td>
<td>VEGaBoard</td>
</tr>
</tbody>
</table>

RavenRV32 efabless Datasheet, GitHub PicoRV32 RV32IMAC RTOS RavenRV32 DevKit

https://riscv.org/risc-v-cores

https://github.com/efabless/raven-picorv32

Raven: An ASIC implementation of the PicoSoC PicoRV32

Designed by efabless engineering, San Jose, CA

Overview

The purpose of this repository is to provide a completely free and open-source simulation environment for the Raven chip. The Raven chip itself can be found in the efabless IP catalog. See https://efabless.com (registration is required to view the IP catalog). Registration is free and validation is not required to view the catalog contents. However, validation is required to use the efabless design environments; e.g., to view the layout of the raven chip on the Open Galaxy design platform.)
CHIPLICITY PLATFORM

efabless.com
## OPEN SoC DESIGN FLOW

<table>
<thead>
<tr>
<th>DESIGN STEP</th>
<th>EF AT LAUNCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Design</td>
<td>CloudV</td>
</tr>
<tr>
<td>RTL Lint</td>
<td>Verilator</td>
</tr>
<tr>
<td>RTL Simulation</td>
<td>iVerilog</td>
</tr>
<tr>
<td>Logic Synthesis</td>
<td>Yosys</td>
</tr>
<tr>
<td>DFT Scan Insertion</td>
<td>QFlow (in development)</td>
</tr>
<tr>
<td>DFT ATPG</td>
<td>Efabless (in development)</td>
</tr>
<tr>
<td>Formal Verification</td>
<td>N/A**</td>
</tr>
<tr>
<td>Place and Route</td>
<td>Qflow</td>
</tr>
<tr>
<td>Clock Tree Synthesis</td>
<td>Qflow</td>
</tr>
<tr>
<td>Dynamic EMIR Drop</td>
<td>N/A**</td>
</tr>
<tr>
<td>Signal Integrity</td>
<td>N/A**</td>
</tr>
<tr>
<td>Extraction</td>
<td>Magic</td>
</tr>
<tr>
<td>Timing Analysis</td>
<td>Vesta/OpenSTA</td>
</tr>
<tr>
<td>Floor Planning</td>
<td>Magic</td>
</tr>
<tr>
<td>Top Level Placement</td>
<td>Magic</td>
</tr>
<tr>
<td>Top Level Routing</td>
<td>Magic</td>
</tr>
<tr>
<td>LVS</td>
<td>NetGen</td>
</tr>
<tr>
<td>DRC</td>
<td>Magic</td>
</tr>
</tbody>
</table>

- SoC Editor
- RTL Simulation
- Synthesis
- GL Simulation

- Schematic Capture
- SPICE Simulation
- Mixed-Mode Simulation
- Parasitic Extraction
- Physical Verification
An Open Source Hardware Framework that provides chip designers with everything needed to design, verify, and prototype Mixed Signal SoC Products.
OPEN SOURCE SoC COMMERCIALIZATION

CUSTOMER DEFINES

COMMUNITY DEVELOPS

CUSTOMERS

MANUFACTURING

Design Marketplace

COMMUNITY DEFINES

COMMUNITY DEVELOPS

DESIGN COMMUNITY

© 2019 EFABLESS CORPORATION

RISC-V Workshop - Zurich June 11 - 13, 2019
SUMMARY
KEY MESSAGES

End-to-end open source hardware is possible, although transistor level descriptions (i.e., GDS) remain elusive without open foundries.

On mature process nodes (e.g., 0.18μm), using best practices and reasonable margins, open source EDA tools are capable of making production-grade chips.

First-time silicon success is possible with open source EDA tools.

Community involvement and collaboration makes open source happen.
PATH TO OS SILICON

https://efabless.com