ETH Zurich Arnold Test Chip & CORE-V MCU Project Proposal

Brian Faith faith@quicklogic.com
Tim Saxe saxe@quicklogic.com
T @QuickLogic_Corp
www.quicklogic.com
Open Source FPGA Tooling: Our Journey from Resistance to Adoption

Brian Faith
CEO

Sept 2020
QuickLogic at a Glance

Snapshot

- **Founded:** 1989, public since 1999
- **Ticker:** QUIK (NASDAQ)
- **Headquarters:** San Jose, CA

What We Do

- QuickLogic is a platform company that enables our customers to quickly and easily create intelligent ultra-low power endpoints to build a smarter, more connected world
- We develop ultra-low power, multi-core semiconductor platforms and hardware- and software-based IP for AI, voice and sensor processing applications

End-to-End Solutions

Target Markets

- Hearables and wearables
- Consumer and industrial IoT
- Smartphones and tablets
- Consumer electronics
- AI-enabled devices
- Aerospace and defense devices
Resistance

- Over the past 30+ years, 60+ Programmable Logic companies have come and gone
- With so many innovative architectures and great products, why did so many fail?
- Almost all pointed back to challenges with software

And, for those of us who have had robust software…
A Somewhat Random Path
A Somewhat Random Path
A Somewhat Random Path
A Somewhat Random Path

Risk breaking up my walled garden?

No, thank you.
A Somewhat Random Path

RISC-V MCU with 22FDX eFPGA
A Somewhat Random Path

RISC-V MCU with 22FDX eFPGA
A Somewhat Random Path

RISC-V MCU with 22FDX eFPGA
A Somewhat Random Path

Hmm. Maybe there's something here.
A Somewhat Random Path

RISC-V MCU with 22FDX eFPGA
Adoption
Disrupting the Programmable Logic Status Quo

- First Programmable Logic company to **actively contribute to** a fully open source suite of development tools for its FPGA devices and embedded FPGA (eFPGA) technology
- Full RTL-to-bitstream support with complete architecture and accurate timing support
- Changing the paradigm of using existing hardware devices or reference designs that are inflexible and forces developers to adapt rather than having devices adapt to developers needs
- Developed in collaboration with open source industry influencers:
Disrupting the Programmable Logic Status Quo

- First Programmable Logic company to *actively collaborate to create*
  
  - Fully open source suite of development tools for our FPGA devices and embedded FPGA (eFPGA) technology

- Full RTL-to-bitstream support with complete architecture and accurate timing support
Changing the Paradigm…

- Let the community create and adapt hardware to developers’ needs
  - Instead of forcing developers to adapt to existing hardware devices or reference designs

- Work in collaboration with open source industry influencers:

  ![Google Logo](image)
  ![Antmicro Logo](image)
Fully Committed to Open Source FPGA Tools

RISC-V MCU with 22FDX eFPGA
Fully Committed to Open Source FPGA Tools

RISC-V MCU with 22FDX eFPGA
Fully Committed to Open Source FPGA Tools

RISC-V MCU with 22FDX eFPGA
Fully Committed to Open Source FPGA Tools

RISC-V MCU with 22FDX eFPGA
Our Journey

- The Open Source Community’s persistence and passion is remarkable
- Open Source FPGA Tooling is ready for prime time
- There will be a time in the not-so-distance future where they are the norm vs exception
ETH Zurich Arnold Test Chip & CORE-V MCU Project Proposal

Brian Faith faith@quicklogic.com
Tim Saxe saxe@quicklogic.com
T @QuickLogic_Corp
www.quicklogic.com
• Initial contribution of open source RISC-V cores from ETH Zurich PULP Platform
  • Very popular, industry adopted cores
• OpenHW Group becomes the official committer for these repositories

<table>
<thead>
<tr>
<th>Core</th>
<th>Bits/Stages</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE4</td>
<td>Embedded 32 bit 4-stage</td>
<td>An Embedded class 4-stage core that implements, the RV32IMFCXpulp, has an optional 32-bit FPU supporting the F extension and instruction set extensions for DSP operations, including hardware loops, SIMD extensions, bit manipulation and post-increment instructions.</td>
</tr>
<tr>
<td>CVA6</td>
<td>Application 32 &amp; 64 bit 6-stage</td>
<td>An Application class 6-stage, single issue, in-order CPU implementing RV64GC extensions with three privilege levels M, S, U to fully support a Unix-like (Linux, BSD, etc.) operating system. It has configurable size, separate TLBs, a hardware PTW and branch-prediction (branch target buffer, branch history table and a return address stack).</td>
</tr>
</tbody>
</table>
OpenHW Group
BHAG
BIG HAIRY AUDACIOUS GOAL

CORE-V™ APU & MCU SoCs

- Production Ready,
- Using CORE-V CVA6 & CVE4 Cores,
- Deep Sub-Micron SoCs,
- On eval boards,
- Running Linux / Zephyr
- Tapeout CORE-V MCU ~Q1’2021
Case Study: ETH Zurich “Arnold” Test Chip Platform

- 425MHz, 3mmx3mm die size on GF 22FDX
- RISC-V with QuickLogic ArcticPro 2 eFPGA
Arnold – Heterogenous, Energy-Efficient Architecture

• Features
  • RISC-V General Purpose Processor
  • 512 KB Onboard Memory
  • Broad set of peripheral I/O with memory access via µDMA
  • Tightly coupled eFPGA that supports
    • Direct connection to I/O
    • Shared memory accelerator interface
    • I/O filtering functions
    • Config and control interface to/from system

• Benefits
  • Energy efficient architecture enables flexibility to implement hardware partitioning of software requirements
  • Lower unit cost than vs discrete MCU / discrete FPGA implementations
  • OTA hardware upgrades
  • Lower NRE cost vs ‘spinning an ASIC’ for each derivative

System Partitioning in Low Power Computing

- Software running on a processor is the most flexible method to implement any function, however it may:
  - Not fit in the processor’s available compute capability
  - Not fit in the processor’s available memory footprint
  - Not be able keep up with required AI inference rate within power budget
  - Not be able to implement strict I/O timing requirements

- In the above cases, onboard eFPGA is the ideal implementation vehicle
• Project announced at Open Source Developer Forum Sept 2020

• Real Time Operating System (e.g. Zephyr) capable ~600+MHz CV32 MCU host CPU

• Embedded FPGA fabric with hardware accelerators from QuickLogic

• Multiple low power peripheral interfaces (SPI, GPIO, I2C, HyperRAM, CAMIF, etc) for interfacing with sensors, displays, and connectivity modules

• Built in 22FDX with GF
Backup
eFPGA Use Case – HW Offloaded/Accelerated DSP

- DSP functions (e.g. filters) are an integral part of many edge applications such as audio signal processing, and voice recognition

- In lieu of running all of those DSP functions as software on a general purpose processor, certain functions can be offloaded or accelerated to an eFPGA with tightly coupled DSP blocks

- The benefit is a more platform with more computational capability and flexibility to change implementation based on requirements
eFPGA Use Case – I/O Expansion

• Choosing the right specification and combination of peripheral I/O can be challenging.

• With integrated eFPGA, new & evolving I/O standards can be implemented post-tape out to extend the life of a mask set.
eFPGA Use Case – “Real Time” I/O Control

• Some system components need precisely controlled I/O timing to operate correctly

• Implementing this ‘hard real time’ with software can be challenging since processors are a shared resource

• With integrated eFPGA, precise I/O timing can be offloaded from the processor so that CPU loading is decoupled from I/O timing
eFPGA Use Case – Hardware offloaded/accelerated AI

- AI Inferencing that use neural networks tend to benefit from heterogeneous architectures that have parallel processing capability, particularly ones that can process millions of Multiply-Accumulate operations per second.
- eFPGA with tightly coupled DSP blocks, and a direct path to an integrated general purpose processor are very efficient at implementing this architecture.
- The benefit is a more platform with more computational capability (in terms of MACs) and flexibility to change the neural network implementation based on requirements.