A CPU FOR EASY TECHNOLOGY MIGRATION

Dr. Richard Herveille
3rd RISC-V Workshop
Roa Logic

- Privately held and financed consultancy firm
- Specialized in custom IP and FPGA migrations
- Incorporated in 2014
- Strong industry basis
  - Founded by one of the original OpenCores members
  - Engaged in FPGA migrations since 2003.
  - Largely academic team
Freedom of Design

• Different hardware platforms are suited for different needs
  – FPGAs: prototyping, low volume, no NRE, fast TTM
  – Platform ASIC: high performance, low power, fast TTM
  – Std.cell ASICs: highest performance, lowest power, lowest unit price

• Market conditions dictate freely migrating between these technologies is highly desirable
  – E.g. price pressure, power reduction, ASIC EOL, security

• FPGA Vendor specific macros limit the migration to another vendor/technology.
  – Most notably FPGA vendor provided CPUs (e.g. Nios, Microblaze) restrict their usage in other technologies.
Why Migrate FPGAs to ASICs?

• 4P’s
  – Price
  – Performance
  – Power
  – Protection
    • IP theft by copying bitstream
    • Security breach by snooping bitstream or hijacking FPGA
    • SEU/MBU sensitivity
    • Platform availability
Alternatives Study

- Many proprietary and open source alternatives
  - ARM, MIPS, ARC
  - OpenRisc, Leon 2/3
  - OpenSparc, T1
- All have limitations on usability
  - High NREs, limiting ROI
  - Technology applicability/availability
  - Outdated ISAs
  - Potential legal issues due to (L)GPL licensing
Why RISC-V

• CPU requirements
  – Royalty free
  – Target technology independent
  – Equally well suited for FPGA and ASIC
  – Low resource requirements
  – Flexible instruction/feature set
  – Support for multiple bus interfaces

• The RISC-V ISA allows us to fulfill all of the above
RV11

• Roa Logic’s RV32/64 implementation
  – RV11 = in-order, single issue, single thread
  – RV22 = in-order, dual issue, dual thread

• ‘Folded’ optimizing, 5 stage Pipeline
  – Some classic RISC stages are folded together for performance reasons
  – ID stage decides if instruction sequence can be optimized. Improves IPC by hiding stalls

• Designed for FPGA to ASIC migration
  – Technology independent
  – Parameters allows trade offs between features, ISA extensions, and performance vs area
  – Flexible bus interface allows virtual drop-in into any existing system
Architecture

RISC-V Core
- CPU State
- Register File
- Instruction Cache

Execution Pipeline
- Fetch
- Pre-Decode
- Decode/Optimize
- Execute
- Write Back

Branch Predictor
Data Cache

Instruction Interface
Data Interface
Debug Unit
Configurable Interface
Optional Unit
Customer Case Study

- Replace NIOS-II 32-bit Control Plane CPU
  - >100DMIPS
  - No MMU, No Caches
  - AHB3 Interfaces
- Replaced NIOS in the FPGA
  - HW/SW development, test and debug
- Migration to eASIC’s Nextreme-3 Platform ASIC
Replacement Flow

- **Parallel flow**
  - Verify & Debug on FPGA
  - Implement Platform ASIC

- **Once FPGA verification completed Platform ASIC can be taped-out**
Implementation Results

<table>
<thead>
<tr>
<th></th>
<th>Logic Cells</th>
<th>Flipflops</th>
<th>bRAM</th>
<th>Fmax</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclone-V</td>
<td>1923 ALMs</td>
<td>1561</td>
<td>4</td>
<td>114MHz</td>
<td>556mW</td>
</tr>
<tr>
<td>Nextreme-3</td>
<td>7924 eCells</td>
<td>2386</td>
<td>1</td>
<td>649MHz</td>
<td>170mW</td>
</tr>
<tr>
<td>64bit</td>
<td>14721 eCells</td>
<td>4249</td>
<td>1</td>
<td>578MHz</td>
<td>221mW</td>
</tr>
</tbody>
</table>

- Cyclone-V was customer’s current FPGA
- Nextreme-3 chosen for price, performance, and power
  - 5.7x performance increase while reducing power by 70%

- ToDo:
  - Nextreme implements register file in flip flops
Summary

• Implemented RISC-V in a technology independent manner
• Successfully replaced existing FPGA CPU
• Successfully migrated FPGA to Platform ASIC thereby improving CPU performance by 5x and power by 70%

• Next steps
  – Improve resource utilization
  – Increase extensions offerings
  – Add multi-threading, multi-issue
QUESTIONS?
THANK YOU