Soft Cores and ARM/RISC-V Processors

Benjamin Rohr, Moritz Waser

November 4, 2020

Overview

- Soft Cores General
- Examples of Soft Processor Cores
- Design Consideration
- RISC-V
- ARM

Soft Cores General

Soft Cores

- In the past, Field Programmable Gate Arrays (FPGAs) were primarily used for prototyping and debugging purposes
- With their increased popularity, many commercial products now incorporate FPGAs. In the late 1990s, FPGA vendors introduced System-on-chip (SoC) devices that included one or more hard-core processors and an FPGA fabric on a single integrated circuit to allow for more complex designs that involved hardware and software co-integration.

Soft Cores

Still one problem

- While this approach provides advantages of running your design at much higher speeds it does not provide the flexibility of modification to suit the application

Soft Cores

Solution

- FPGA vendors provide the solution of using soft-core processors that is configured from logic resources inside the FPGA.

Soft Cores

Furthermore

- In a context of high performance, low technology cost, and application code reusability objectives, we target FPGA devices and softcore processors to implement and to test the proposed multicore system

Soft Cores

FPGA processor cores are IP and can be categorized into the three standard IP types:

- Hard - involve the implementation of a silicon-level circuit within the FPGA fabric
- **Soft** - are design elements that can be implemented within the FPGA fabric.
- Firm - like soft, but are design elements that can be implemented within the FPGA fabric.

Soft Cores

- Processor implementations in an HDL language
- Without extensive optimization for the target

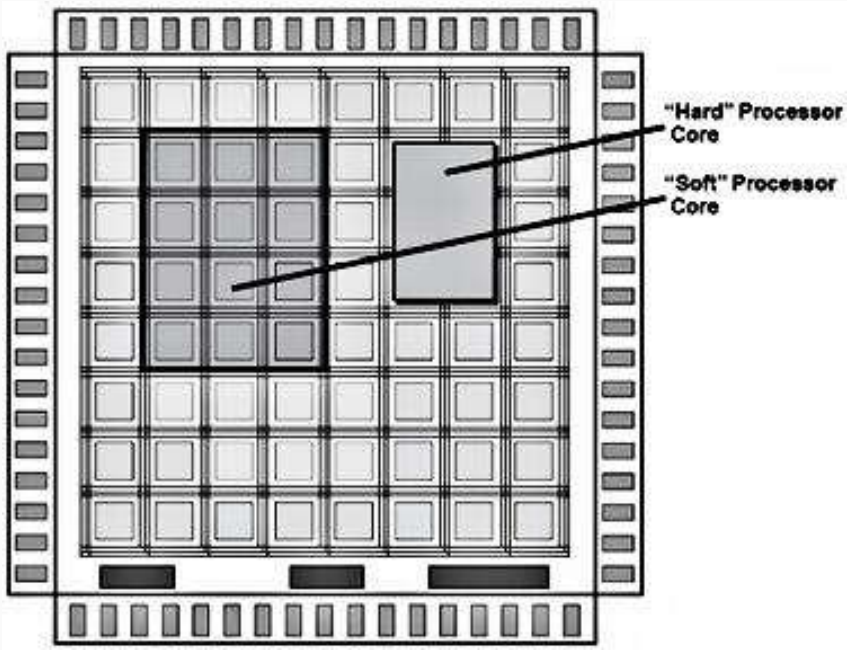

Figure 1: hard and soft processor example

Soft Cores

Advantages

- Higher portability
- Utilizing standard mass-produced and hence lower-cost FPGA parts
- Enabling a custom number of microprocessors per FPGA
- Configurable by the user

Disadvantages

- Reduced processor performance
- Higher power consumption
- Larger size

Examples of Soft Core Processor Cores

Example of Soft Core Processor Cores

- Xilinx MicroBlaze Soft Processor Core
- 32-bit Reduced Instruction Set Computer (RISC)

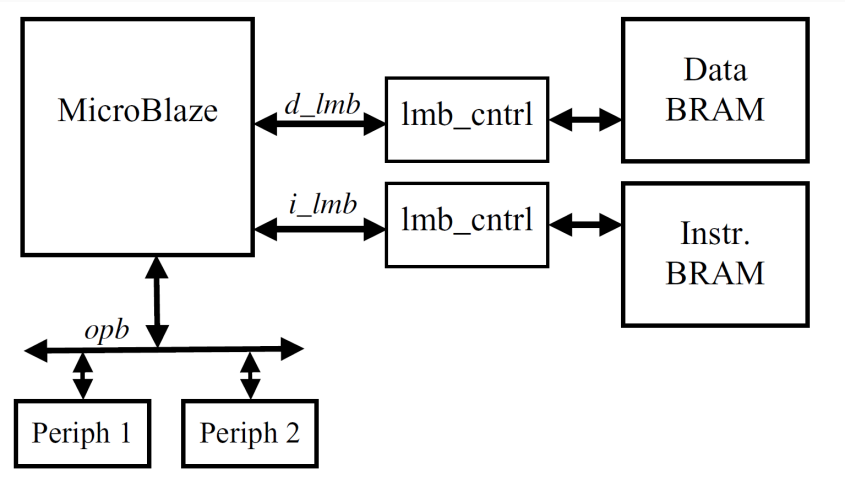

Figure 2: Simple MicroBlaze processor system.

Example of Soft Core Processor Cores

Xilinx's MicroBlaze has following characteristics:

- Harvard bus architecture
- Three-stage pipeline
- 32 general-purpose registers
- Two interrupt levels and exception handling capability
- Configurable cache
- Support for optional co-processing functionality
- Optional single precision floating-point unit (FPU) (IEEE-754 compatible)
- Standardized IBM Core Connect Bus interface

Example of Soft Core Processor Cores

- Open-Source Leon3
- 32-bit processor core conforming to the IEEE-1754 (SPARC V8) architecture
- Highly configurable
- Full source code available under the GNU GPL license

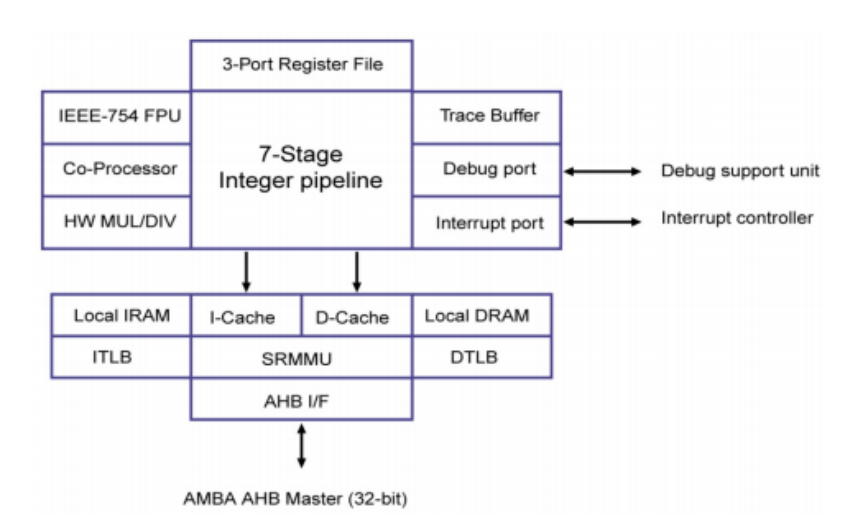

Figure 3: Leon3 core block diagram.

Design Considerations

Design Considerations

Selecting a soft processor core can be a complex task with far-reaching design effects.
Including:

- Cost of implementation
- Operational performance
- Power consumption
- Design and development tools
- Operating System considerations

Design Considerations - Cost

The cost of a delivered design is typically the most important factor for the majority of embedded design projects. The delivered product cost is based on a combination of software and hardware development costs, material and manufacturing.
Cost of the processor element may not be the most significant design cost when considering the full product life cycle including derivative designs.

Design Considerations - Performance and Power

benchmark: is the DMIPS (Dhrystone Million Instructions Per Second)

- Running an algorithm on a targeted processor core to measure its integer processing capabilities within a defined time period.

Additional performance considerations include the architecture of the soft processor core and its suitability for the targeted application.

Design Considerations - Performance and Power

Factors to evaluate include:

- Type and size of the memory and peripheral bus
- Size and model of address space
- Type and size of cache (instruction/data)
- Type of controllers like DMA and interrupt structure
- Hardware accelerator capability (co-processor functionality)
- Functional units such as the register file and execution units
- Type of pipeline and strategies to prevent stalls such as branch prediction

Design Considerations - Performance and Power

Several factors influence power consumption

- Speed of operation
- The number and type of resources required to implement the soft processor core
- The characteristics of the FPGA component including static and dynamic power consumption vs. operational speed and temperature.

One of the challenges associated with FPGA design is the difficulty of estimating power consumption. In an ideal development flow, schedule and resources will be allocated to design evaluation on a targeted development platform with an identical target FPGA component and soft processor implementation

Design Considerations - Design and Development Tools

- The tool suite (Figure 1) includes a collection of traditional software and FPGA design and development tools.
- The interaction between these two tool groups is commonly referred to as codesign or platform development tool.

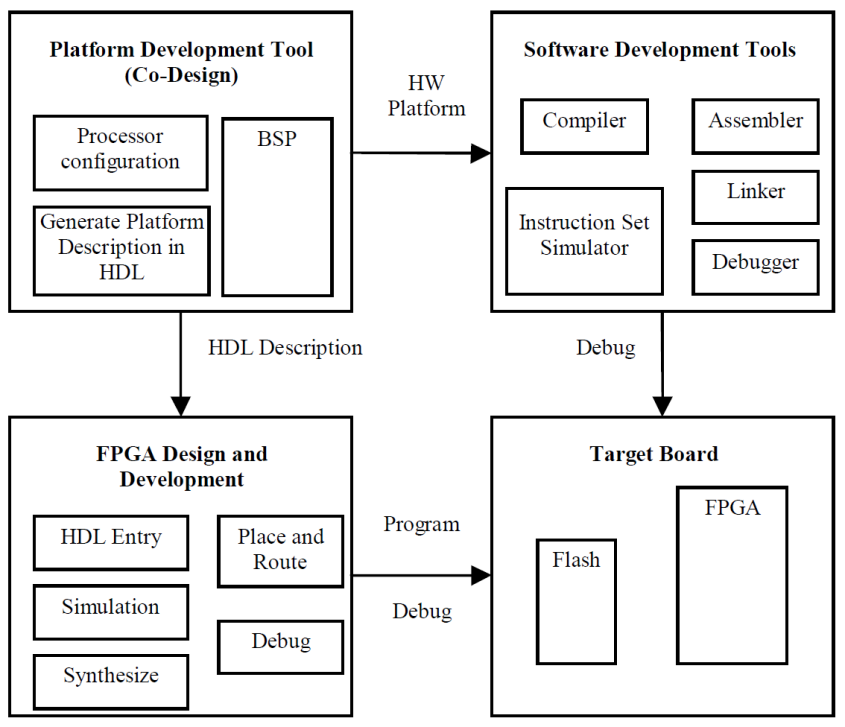

Figure 4: Tool Suite

Design Considerations - Operating System Considerations

Another important design factor is the ability to utilize popular operating systems.
Most operating systems include the OS and any lower-level software required to connect the OS to the hardware.

RISC-V

RISC-V

- RISC Instruction Set Architecture (ISA)
- First introduced in 2010 by the University of California, Berkeley
- Open Source
- Managed by RISC-V International

Figure 5: Official RISC-V logo.

Some RISC-V International Members:

Figure 6: Famous RISC-V International members.

RISC-V

Design goals:

- Modularity
- Scalability
- Academic and commercial usage

RISC-V

Unprivileged architecture:

- Modular design
- Base and extensions
- Standard modules can be combined without conflict

Base	Version	Status
RVWMO	2.0	**Ratified**
RV32I	**2.1**	**Ratified**
RV64I	**2.1**	**Ratified**
RV32E	*1.9*	*Draft*
RV128I	*1.7*	*Draft*
Extension	Version	Status
Zifencei	**2.0**	**Ratified**
Zicsr	**2.0**	**Ratified**
M	**2.0**	**Ratified**
A	*2.0*	Frozen
F	**2.2**	**Ratified**
D	**2.2**	**Ratified**
Q	**2.2**	**Ratified**
C	**2.0**	**Ratified**
Ztso	*0.1*	*Frozen*
Counters	*2.0*	*Draft*
L	*0.0*	*Draft*
B	*0.0*	*Draft*
J	*0.0*	*Draft*
T	*0.0*	*Draft*
P	*0.2*	*Draft*
V	*0.7*	*Draft*
N	*1.1*	*Draft*
Zam	*0.1*	*Draft*

Figure 7: Unprivileged ISA

Privileged architecture:

- Machine ISA
- Supervisor ISA
- Supports full virtualization

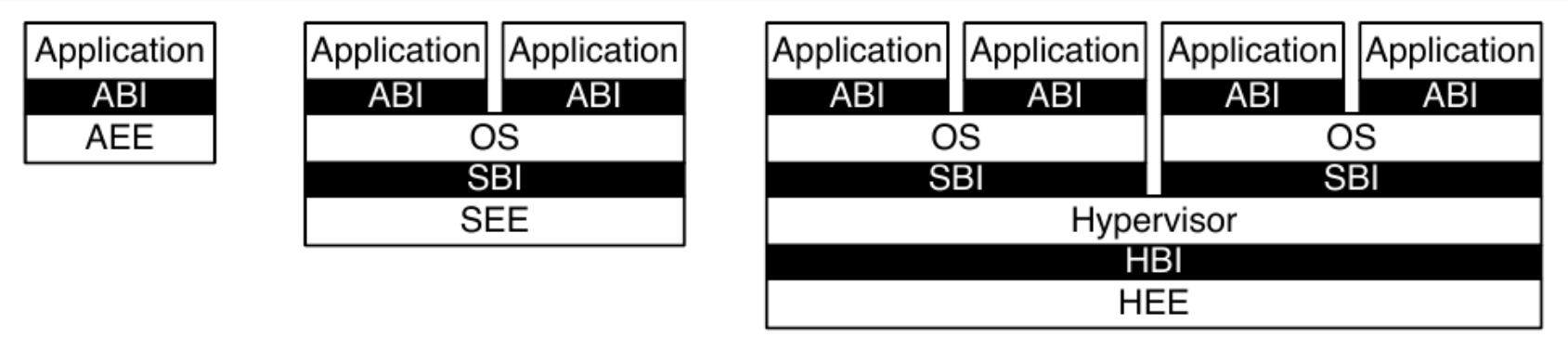

Figure 8: Privileged ISA possibilities.

Ibex by lowRISC

- Implements the RV32IMC / RV32EMC IS
- 2-stage pipeline
- Apache 2.0 license (Open Source)

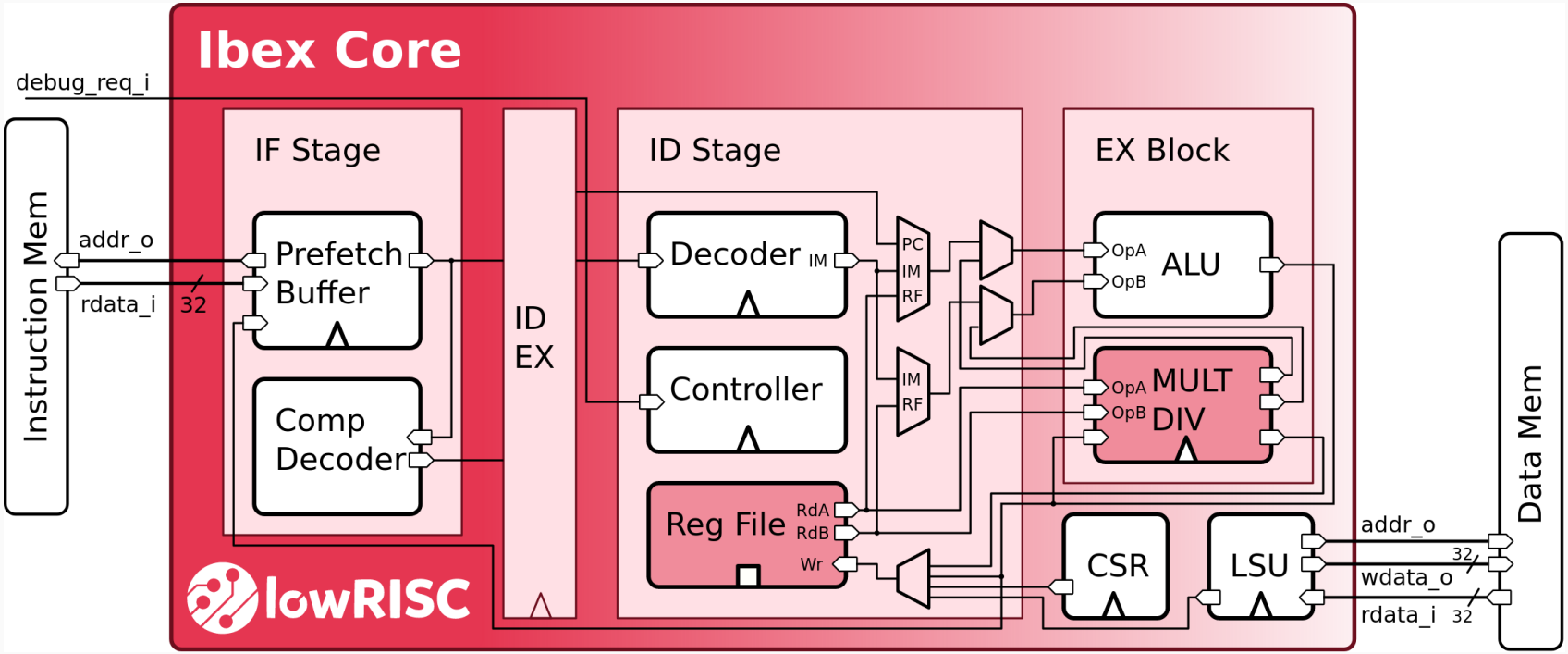

Figure 9: Ibex core overview.

RISC-V Soft Cores

VexRiscv by Charles Papon

- Won the RISC-V SoftCPU Contest in 2018
- Written in SpinalHDL (also by Charles Papon)
- Implements RV32I[M][C][A] IS
- Optionally supports MMU, caching, interrupts, privileged ISA etc.
- Compatible with Linux, Zephyr & FreeRTOS
- MIT license (Open Source)

RISC-V Soft Cores

RV32EC-P0, RV32EC-P2 and RV32IC-P5 by IQonIC Works

- Commercial
- Several cores for different performance/power/area goals
- Comes with a software toolchain for rapid prototyping/customization

Figure 10: IQonIC Works logo.

RISC-V Soft Cores

RV32EC-P2

- 2-stage pipeline
- Extendable (RV32I base, M)
- AHB/APB interface
- Interrupt support

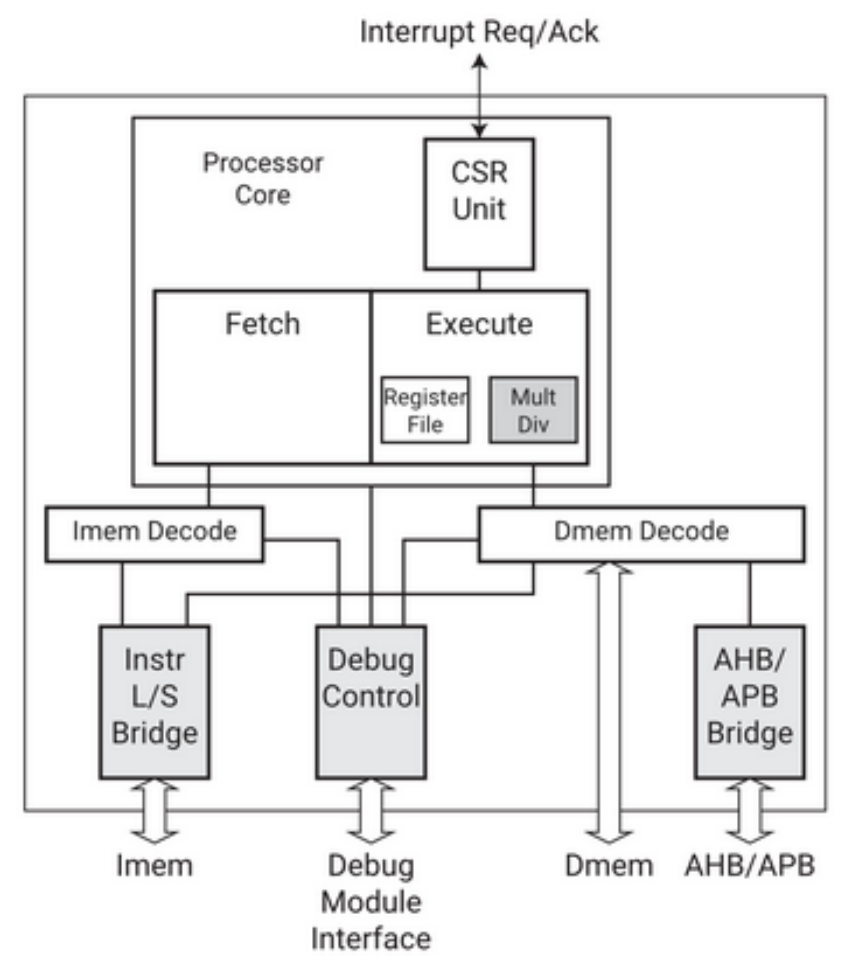

Figure 11: RV32EC-P2 overview.

RISC-V Soft Cores

RV32IC-P5

- 5-stage pipeline
- Extendable (A, N, M ...)
- Full privileged RISC-V ISA support
- Supports branch-prediction, caches etc.

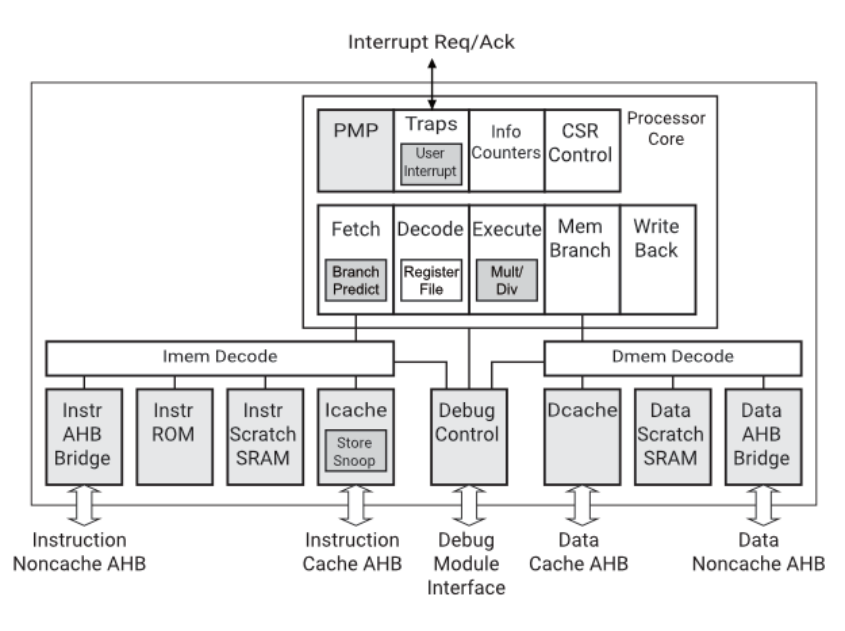

Figure 12: RV32IC-P5 overview.

ARM

ARM

- Advanced RISC Machine / Acorn RISC Machine
- Developed by Arm Holdings
- Licensing model
- Most widely used ISA

Figure 13: Arm Holdings logo.

ARM

Licensing models:

- Manufacture-ready IP cores (netlist and simulation tools)
- Adaptable IP cores (Verilog)
- Built on Cortex license (includes brand freedom)
- Architectural license
- ARM Flexible Access (per product fees)

ARM

The ARM ISA:

- 32-bit architecture until ARMv8-A
- Conditional execution (instructions like ADDGT)
- Shift/Rotate operations can be combined with data processing instructions
- ARM cores feature between 3 and 13 pipeline stages

ARM

Other features:

- Jazelle (direct Java bytecode execution)
- Thumb (16-bit instruction set to reduce code density)
- Neon (SIMD instruction set for multimedia and DSP applications)
- Helium (Vector instructions)

ARM Hard Cores

The ARM ecosystem:

- Long history of hard cores
- Used in all kinds of devices
- Different kinds of specialization
- Lots of software

ARM Hard Cores

The Zynq-7000 family:

- ARM Cortex-A9 processor(s)
- Artix-7 based FPGA
- Very flexible

Figure 14: A Zynq chip.

ARM Soft Cores

The problem with ARM soft cores:

- ARM cores are commercial products
- ARM Holdings concerned about IP theft
- Most ARM cores can only be used for SoCs/ASICs
- ARM started the *DesignStart FPGA* project in 2018
  - Makes ARM cores available as soft cores for Xilinx FPGAs
  - So far only Cortex-M1 and Cortex-M3 are available

ARM Soft Cores

Cortex-M1

- FPGA-optimized version of Cortex-M0
- ARMv6-M architecture
- Small area size
- Subset of Thumb-1/Thumb-2 IS
- Up to 32 interrupts plus NMI
- Up to 1024KiB I/D-TCM

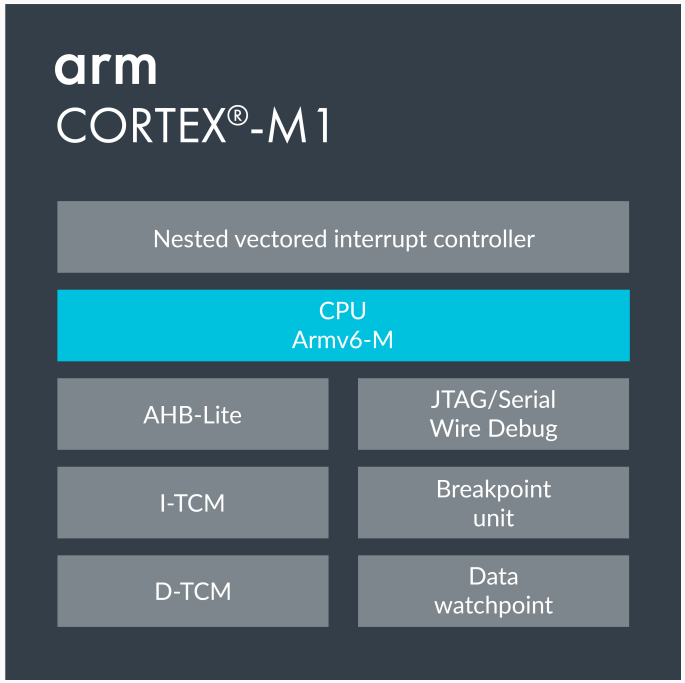

Figure 15: Cortex-M1 overview.

ARM Soft Cores

Cortex-M3

- General purpose processor
- ARMv7-M architecture
- Up to 240 interrupts plus NMI
- 8 Region memory protection unit
- Wake-up/Sleep capability

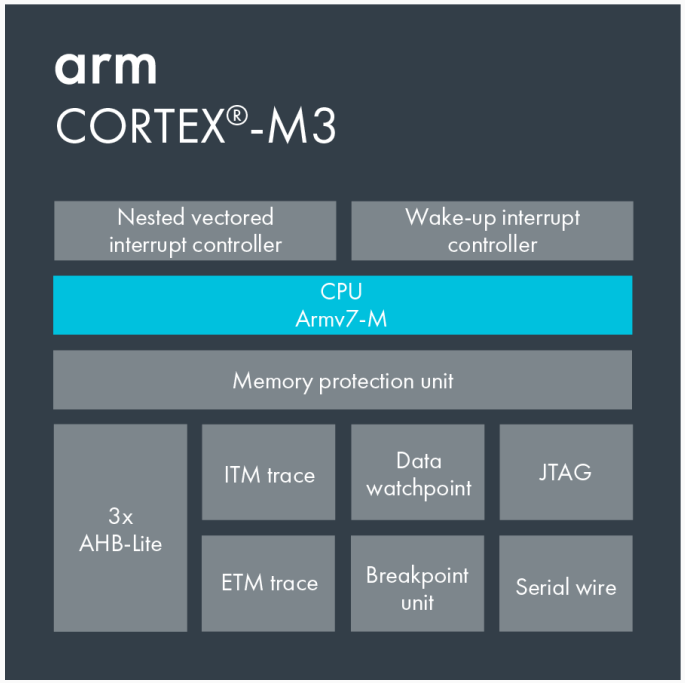

Figure 16: Cortex-M3 overview.

RISC-V vs. ARM

RISC-V

- \+ Open Source
- \+ No attatched cost
- – Requires flexible software
- – Could need a lot of tinkering

ARM

- \+ Software support
- \+ Production ready
- – Mostly closed ecosystem
- – Royalty fees

Questions?

References i

[1] Ludovic L'Hours *Generating Efficient Custom FPGA Soft-Cores for Control-Dominated Applications*, year=2005

[2] Vivek Jayakrishnan and Chirag Parikh *Embedded Processors on FPGA: Soft vs Hard*, year=2019

[3] Roman Lysecky and Frank Vahid *A Study of the Speedups and Competitiveness of FPGA Soft Processor Cores using Dynamic Hardware/Software Partitioning*, year=2005

[4] Peter Yiannacouras, J. Gregory Steffan, and Jonathan Rose *Exploration and Customization of FPGA-Based Soft Processors*, year=2007

References ii

[5] R.C. Cofer and Ben Harding. *Basics of core-based FPGA design*. https://www.embedded.com/basics-of-core-based-fpga-design-part-1-core-types-trac Online; accessed 31. October 2020.

[6] P. Minev and V. Kukenska *IMPLEMENTATION OF SOFT-CORE PROCESSORS IN FPGAs*, year=2010

[7] Mouna Baklouti and Mohamed Abid *Multi-Softcore Architecture on FPGA*, https://www.hindawi.com/journals/ijrc/2014/979327/, Online; accessed 02.11.2020

References iii

[8] R.C. Cofer and Ben Harding. *FPGA Soft Processor Design Considerations*, https://www.eetimes.com/fpga-soft-processor-design-considerations/, Online; accessed 02.11.2020

[9] Andrew Waterman, Krste Asanovic. *The RISC-V Instruction Set Manual Volume I: Unprivileged ISA*. https://github.com/riscv/riscv-isa-manual/releases/download/Ratified-IMAFDQC/riscv-spec-20191213.pdf. Online; accessed 02.11.2020

References iv

[10] Andrew Waterman, Krste Asanovic, *The RISC-V Instruction Set Manual Volume II: Privileged Architecture*. https://github.com/riscv/riscv-isa-manual/releases/download/Ratified-IMFDQC-and-Priv-v1.11/riscv-privileged-20190608.pdf Online; accessed 02.11.2020

[11] lowRISC. *Ibex RISC-V Core*. https://github.com/lowrisc/ibex. Online; accessed 02.11.2020

[12] Charles Papon. *VexRiscv*. https://github.com/SpinalHDL/VexRiscv. Online; accessed 02.11.2020

References v

[13] IQONIC Works *RISC-V IP to suit your needs....* https://iqonicworks.com/risc-v-ip. Online; accessed 02.11.2020

[14] *ARM architecture*. https://en.wikipedia.org/wiki/ARM_architecture. Online; accessed 02.11.2020

[15] *Cortex-M1*. https://developer.arm.com/ip-products/processors/cortex-m/cortex-m1. Online; accessed 02.11.2020

[16] *Cortex-M3*. https://developer.arm.com/ip-products/processors/cortex-m/cortex-m3. Online; accessed 02.11.2020