Soft Cores and ARM/RISC-V Processors

Ahmed Ahmed

December 1, 2021

Agenda

- Soft Cores - What are Soft Cores?
- Soft Cores versus Hard Cores
- Proprietary Soft Cores and Open Source Soft Cores
- ARM - Architecture and Soft Cores
- RISC-V - Architecture and Soft Cores

Soft Cores - What are Soft Cores?

What are Soft Cores?

- Soft Cores are **dynamically reconfigurable processing IP cores**
- Can be
  - **General Purpose Processors**
  - Digital Signal Processors
  - Coprocessors
  - Microcontrollers
- Implemented using the logic resources of the **FPGA fabric**
- Delivered as a synthesizable HDL design

What are Soft Cores?

Soft Cores are **highly flexible**, but still have some predefined characteristics which can not be changed:

- Number of instructions
- **Instruction Set Architecture** (ISA)
- some functional blocks

Other characteristics can be defined and adapted by the designer:

- Additional peripherals
- Memory map
- Configuration
- **Performance / Resource trade-off** (Altera Nios-II f/s/e)

Soft Cores can (to a certain extent) be **tailored to a specific target application.**

Soft Cores versus Hard Cores

Soft Cores versus Hard Cores

Soft Core Processor is **synthesized onto a FPGA fabric** (e.g. Xilinx PicoBlaze / MicroBlaze on Xilinx FPGA)
Hard Core Processor is a **integrated circuit on a dedicated silicon** (e.g. ARM Cortex-A9 on Zybo Z7)

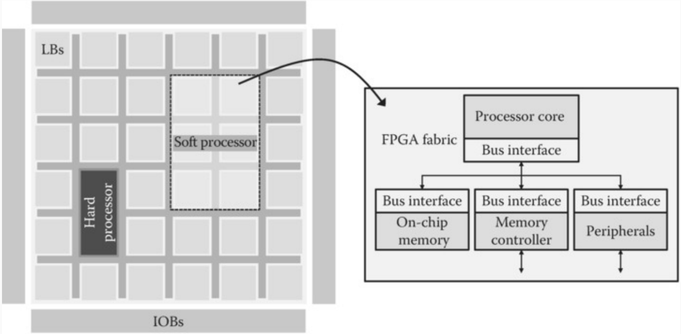


Figure 1: Soft Core

Soft Cores versus Hard Cores

Pros of Soft Core compared to Hard Core Processor:

- Adjustable area
- Flexible architecture, easy to adapt
- **Adaptable configuration, speed, characteristics**
- Optimize for particular application (e.g. add peripheral to extend functionality or remove peripheral to reduce resources)
- Combining multiple cores into a single FPGA is relatively easy, **flexible number of cores**
- Implementing more than one processor on the same FPGA
- **Scalability** (add resources to support new features, update existing features)
- **Portability** (migrate to other FPGAs)
- Lower non-recurring engineering costs
  (IC for FPGA itself already manufactured)

Soft Cores versus Hard Cores

Cons of Soft Core compared to Hard Core Processor:

- **Larger size, less area efficient**
- Higher power consumption
  (compared to same Processor as Hard Core Processor)
- **Reduced processor performance**
- Lower performance per watt
- Worse predictability
- May not work in technology not tested for

Soft Cores versus Hard Cores

When to use Soft Core Processors?

If we want or need...

- **Flexibility**
- Scalability
- Portability
- Lower engineering costs

Soft Cores versus Hard Cores

When not to use Soft Core Processors and use traditional Hard Core Processors?

If we want or need...

- **Higher performance**
- Lower power consumption
- Smaller size

Proprietary Soft Cores

Open Source Soft Cores

Proprietary Soft Cores

- Optimized for a **particular FPGA architecture**
- Better performance, better resource utilization, lower energy consumption
- **Predictive Behavior**
- Portability and the possibility of reusing the code is limited
- e.g. Xilinx PicoBlaze / MicroBlaze optimized for Xilinx FPGAs

Open Source Soft Cores

- **Technology independent**
- Low cost compared to proprietary soft cores
- Proven architectures, supported by full set of tools and operating systems
- **Better portability** but lower performance and worse resource utilization
- e.g. OpenRISC1200, LEON4

ARM - Architecture and Soft Cores

ARM - Architecture

- **ARM** - **A**dvanced **R**ISC **M**achines (originally Acorn RISC Maschines)
- Family of **Reduced Instruction Set Computing (RISC)** Architectures
- Different Versions: ARMv1 (1985) - ARMv9 (2021)
- Developed and Licensed by Arm Ltd.

Figure 2: Arm Ltd. logo

ARM - Architecture

- Known for **lower power consumption and heat generation** compared to their competitors
  (who rely on CISC architectures, e.g. AMD or Intel)
- Mostly used for portable battery-powered devices
  (smartphones, tablets, laptops) and embedded systems
- Nowadays also used in PCs and Servers
  (**fastest Supercomputer** Fujitsu Fugaku uses ARMv8.2-A)

ARM - Soft Cores

Arm Ltd. has always been known for developing hard cores
Most famous ARM cores: Arm Cortex series

- Arm Cortex-**A**: series of **A**pplication cores
- Arm Cortex-**R**: cores optimized for **R**eal-Time Applications
- Arm Cortex-**M**: smallest cores for **M**icrocontrollers

ARM - Soft Cores

In 2018, Arm released two Cortex-M **Soft Cores for FPGA integration with Xilinx products**:

- Arm Cortex-M1
- Arm Cortex-M3

Currently only these two ARM Soft Cores exist.

- Free - No license fee
- But... not **open source** :(

Arm Cortex-M1

FPGA-optimized version of the Cortex-M0 processor

- Armv6-M architecture
- Thumb/Thumb-2 subset ISA
- Three-stage pipeline
- AHB-Lite bus
- Instruction-TCM up to 1MB
- Data-TCM up to 1MB
- Up to 32 interrupts

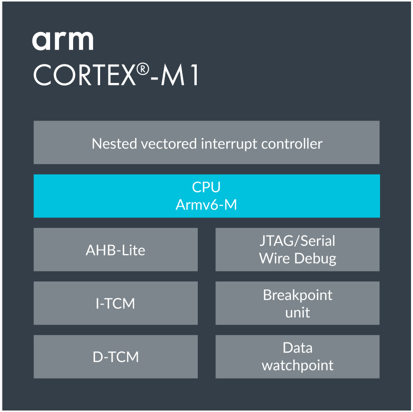


Figure 3: Cortex-M1 processor

Arm Cortex-M3

- Armv7-M architecture
- Thumb/Thumb-2 subset ISA
- Three-stage pipeline with branch predictor
- 3x AHB-Lite bus
- Instruction-TCM up to 1MB
- Data-TCM up to 1MB
- Up to 240 interrupts
- Memory Protection Unit (MPU)
- Integrated sleep modes

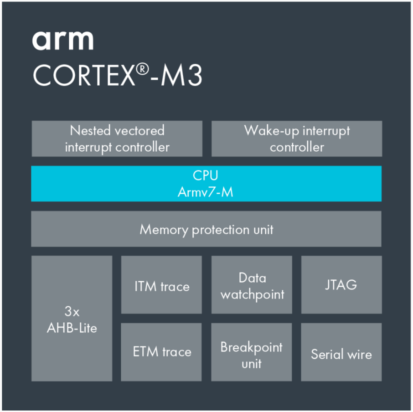


Figure 4: Cortex-M3 processor

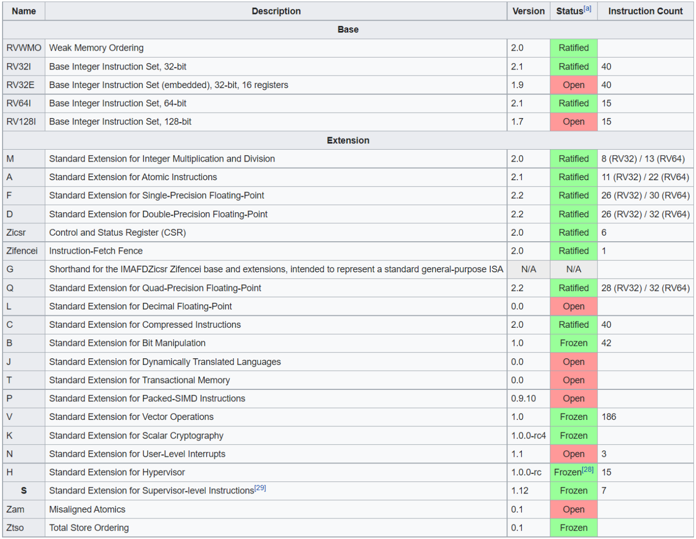
RISC-V - Architecture and Soft Cores

RISC-V - Architecture

- **RISC-V** ("risc-five") - **R**educed **I**nstruction **S**et **C**omputing **V**
- Originated 2010 at University of California, Berkeley
- "Five" because fifth generation of RISC Architecture developed at University of California, Berkeley
- **Open source** RISC Architecture
- Maintained by non-profit **RISC-V Foundation**

Figure 5: RISC-V logo

RISC-V - Architecture



Name	Description	Version	Status[a]	Instruction Count
Base				
RVWMO	Weak Memory Ordering	2.0	Ratified	
RV32I	Base Integer Instruction Set, 32-bit	2.1	Ratified	40
RV32E	Base Integer Instruction Set (embedded), 32-bit, 16 registers	1.9	Open	40
RV64I	Base Integer Instruction Set, 64-bit	2.1	Ratified	15
RV128I	Base Integer Instruction Set, 128-bit	1.7	Open	15
Extension				
M	Standard Extension for Integer Multiplication and Division	2.0	Ratified	8 (RV32) / 13 (RV64)
A	Standard Extension for Atomic Instructions	2.1	Ratified	11 (RV32) / 22 (RV64)
F	Standard Extension for Single-Precision Floating-Point	2.2	Ratified	26 (RV32) / 30 (RV64)
D	Standard Extension for Double-Precision Floating-Point	2.2	Ratified	26 (RV32) / 32 (RV64)
Zicsr	Control and Status Register (CSR)	2.0	Ratified	6
Zifencei	Instruction-Fetch Fence	2.0	Ratified	1
G	Shorthand for the IMAFDZicsr Zifencei base and extensions, intended to represent a standard general-purpose ISA	N/A	N/A	
Q	Standard Extension for Quad-Precision Floating-Point	2.2	Ratified	28 (RV32) / 32 (RV64)
L	Standard Extension for Decimal Floating-Point	0.0	Open	
C	Standard Extension for Compressed Instructions	2.0	Ratified	40
B	Standard Extension for Bit Manipulation	1.0	Frozen	42
J	Standard Extension for Dynamically Translated Languages	0.0	Open	
T	Standard Extension for Transactional Memory	0.0	Open	
P	Standard Extension for Packed-SIMD Instructions	0.9.10	Open	
V	Standard Extension for Vector Operations	1.0	Frozen	186
K	Standard Extension for Scalar Cryptography	1.0.0-rc4	Frozen	
N	Standard Extension for User-Level Interrupts	1.1	Open	3
H	Standard Extension for Hypervisor	1.0.0-rc	Frozen[28]	15
S	Standard Extension for Supervisor-level Instructions[29]	1.12	Frozen	7
Zam	Misaligned Atomics	0.1	Open	
Ztso	Total Store Ordering	0.1	Frozen	

Figure 6: RISC-V ISA base and extensions

RISC-V - Soft Cores

Since RISC-V is open source, there are **much more Soft Cores** available:

- Ibex Core
- CV32E40P
- SweRV
  - EH1 Core
  - EH2 Core
  - EL2 Core
- ... and many more

All of the above mentioned RISC-V cores are **open source**!

Ibex

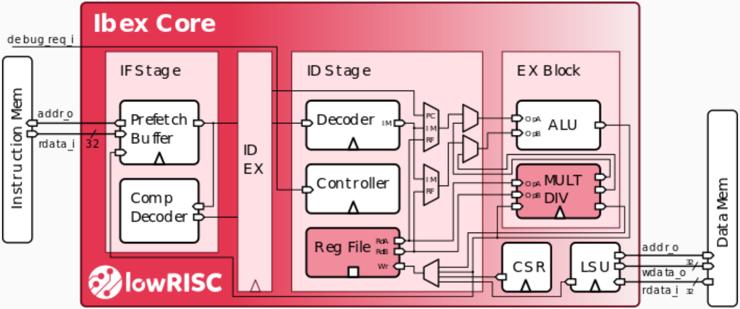


Figure 7: Ibex block diagram

- 32-bit RISC-V core
- RV32I or RV32E with [M][C][B] Extensions
- 2 stage Pipeline (IF, ID/EX)
- Implemented in SystemVerilog

CV32E40P

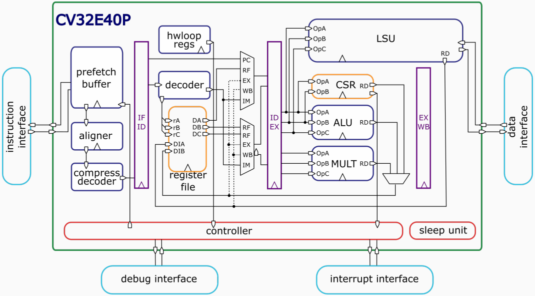


Figure 8: CV32E40P block diagram

- 32-bit RISC-V core
- RV32I with [M][F][C] Extensions
- 4 stage pipeline (IF, ID, EX, WB)
- Uses Open Bus interface
- Implemented in SystemVerilog

SweRV EH1 Core

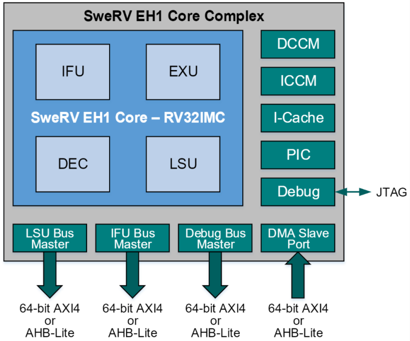


Figure 9: SweRV EH1 Core Complex

- 32-bit RISC-V core
- RV32I with [M][C][Z] Extensions
- AXI4 or AHB-Lite

SweRV EH1 Core

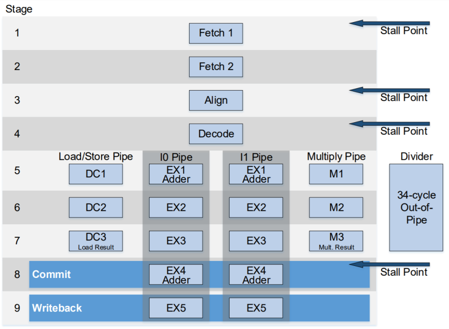


Figure 10: SweRV EH1 Core Pipeline

- Dual-issue 9-stage Pipeline supporting four ALUs (I0, I1)
- Load/Store Pipe
- Multiply Pipe
- Out-of-Pipe Divider

SweRV EH2 Core

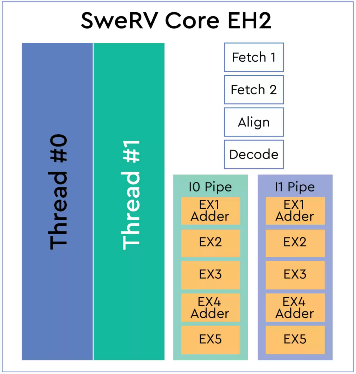


Figure 11: SweRV EH2 Core

- Built off the SweRV EH1 Core
- Dual Threaded for higher performance
- Additional ISA extensions compared to EH1 Core

SweRV EL2 Core

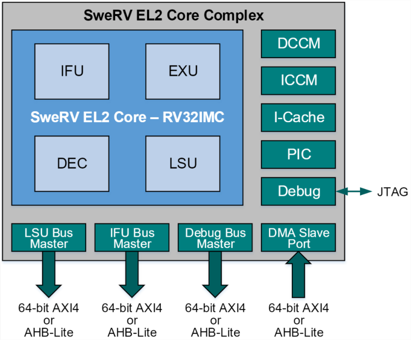


Figure 12: SweRV EL2 Core Complex

- Built off the SweRV EH1 Core
- Smaller core with less performance than EH1 Core
- Designed to replace State Machines and other logic functions in SoCs

SweRV EL2 Core

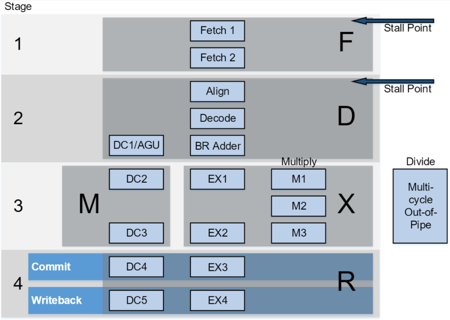


Figure 13: SweRV EL2 Core Pipeline

- Single-issue 4-stage Pipeline
- Fetch - Decode - Execute/Memory - Retire
- Pipelines: Execute, Load/Store, Multipler
- Out-of-Pipe Divider

Thank You!

Questions? Comments?

References i

[1] Vivek Jayakrishnan Vazhoth Kanhiroth, Chirag Parikh
Embedded Processors on FPGA: Soft vs Hard.
`https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1843&context=theses`. Online; accessed 28 November 2021.

[2] *Basics of core-based FPGA design: Part 1 – core types & trade-offs*. `https://www.embedded.com/basics-of-core-based-fpga-design-part-1-core-types-trad`
Online; accessed 28 November 2021.

References ii

[3] Pieter Anemaet, Thijs van As *Microprocessor Soft-Cores: An Evaluation of Design Methods and Concepts on FPGAs*. https://pretopia.net/files/paper_softcores.pdf. Online; accessed 28 November 2021.

[4] Petar Borisov Minev, Valentina Stoianova Kukenska *IMPLEMENTATION OF SOFT-CORE PROCESSORS IN FPGAs*. https://www.semanticscholar.org/paper/IMPLEMENTATION-OF-SOFT-CORE-PROCESSORS-IN-FPGAs-Minev-K
359ee91b9e750e052d0785856325860d98e66247. Online; accessed 28 November 2021.

References iii

[5] *FPGA Soft Core*. https://www.mikrocontroller.net/articles/FPGA_Soft_Core. Online; accessed 28 November 2021.

[6] Juan José Rodríguez Andina, Eduardo de la Torre Arnanz, María Dolores Valdés Peña *Embedded Processors in FPGA Architectures*. https://www.routledgehandbooks.com/doi/10.1201/9781315162133-4. Online; accessed 28 November 2021.

[7] *ARM architecture*. https://en.wikipedia.org/wiki/ARM_architecture. Online; accessed 28 November 2021.

References iv

[8] *ARM Cortex-A*. `https://en.wikipedia.org/wiki/ARM_Cortex-A`. Online; accessed 28 November 2021.

[9] *ARM Cortex-M*. `https://en.wikipedia.org/wiki/ARM_Cortex-M`. Online; accessed 28 November 2021.

[10] Phil Burr. *Arm expands design possibilities with free Cortex-M processors for Xilinx FPGAs*. `https://www.arm.com/company/news/2018/10/arm-expands-design-possibilities-with-free-cortex-m-pro` Online; accessed 28 November 2021.

References v

[11] *RISC-V*. https://en.wikipedia.org/wiki/RISC-V. Online; accessed 28 November 2021.

[12] Tom Simon. *RISC-V Business*. https://semiwiki.com/ip/sifive/7168-risc-v-business/. Online; accessed 28 November 2021.

[13] *Ibex RISC-V Core*. https://github.com/lowRISC/ibex. Online; accessed 28 November 2021.

[14] *PicoRV32 - A Size-Optimized RISC-V CPU*. https://github.com/cliffordwolf/picorv32. Online; accessed 28 November 2021.

References vi

[15] *OpenHW Group CORE-V CV32E40P RISC-V IP*. https://github.com/openhwgroup/cv32e40p. Online; accessed 28 November 2021.

[16] *OpenHW Group CV32E40P User Manual*. https://docs.openhwgroup.org/projects/cv32e40p-user-manual/. Online; accessed 28 November 2021.

[17] *VexRiscv*. https://github.com/SpinalHDL/VexRiscv. Online; accessed 28 November 2021.

[18] *RISC-V — Western Digital*. https://www.westerndigital.com/solutions/business/risc-v. Online; accessed 29 November 2021.

References vii

[19] *EH1 RISC-V SweRV Core from Western Digital*. https://github.com/chipsalliance/Cores-SweRV/blob/master/docs/RISC-V_SweRV_EH1_PRM.pdf. Online; accessed 29 November 2021.

[20] *EH2 RISC-V SweRV Core from Western Digital*. https://github.com/chipsalliance/Cores-SweRV-EH2/blob/master/docs/RISC-V_SweRV_EH2_PRM.pdf. Online; accessed 29 November 2021.

[21] *EL2 RISC-V SweRV Core from Western Digital*. https://github.com/chipsalliance/Cores-SweRV-EL2/blob/master/docs/RISC-V_SweRV_EL2_PRM.pdf. Online; accessed 29 November 2021.