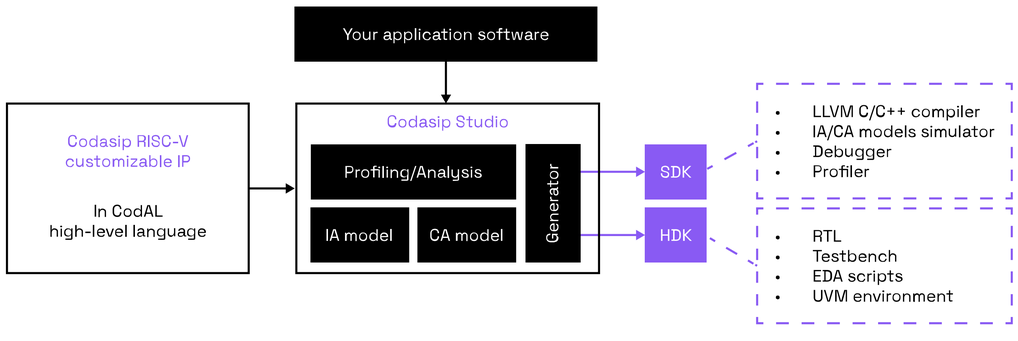
Effortless DSP extensions design for embedded RISC-V processors

Alexey Shchekin, Ettore Giliberti

Codasip

→ Codasip Studio and RISC-V Processors



Begin with a standard core

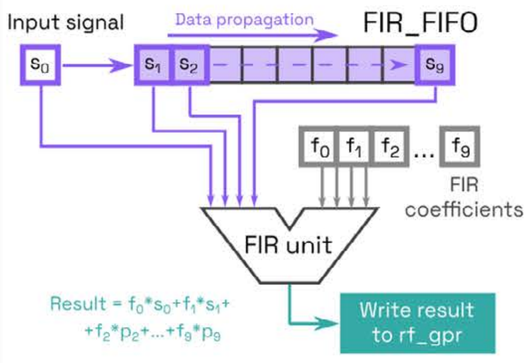
- Embedded and application cores
- High quality, production-ready
- Fully RISC-V compliant

Differentiate with Codasip Studio

- Configure / Customize
- Using CodAL architecture description language

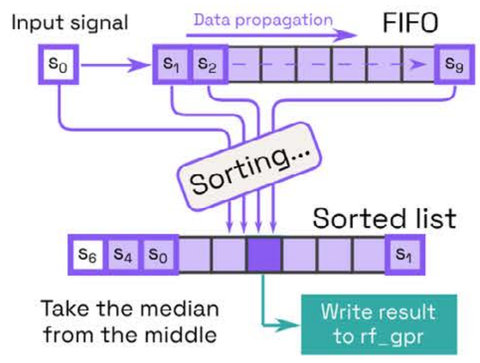
→ DSP accelerators implemented with CodAL

FIR filter accelerator



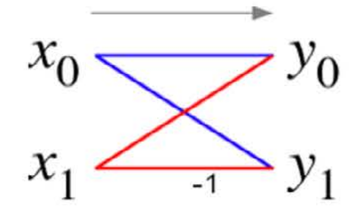
1 custom instruction call to get the FIR output value
3 cycles to get the result

Median filter accelerator



1 custom instruction call to get the median value
3 cycles to get the result

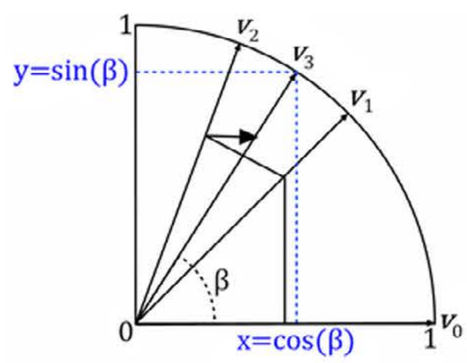
FFT accelerator



$$y_0 = x_0 + x_1 \cdot e^{-i\frac{2\pi k}{N}}$$
$$y_1 = x_0 - x_1 \cdot e^{-i\frac{2\pi k}{N}}$$

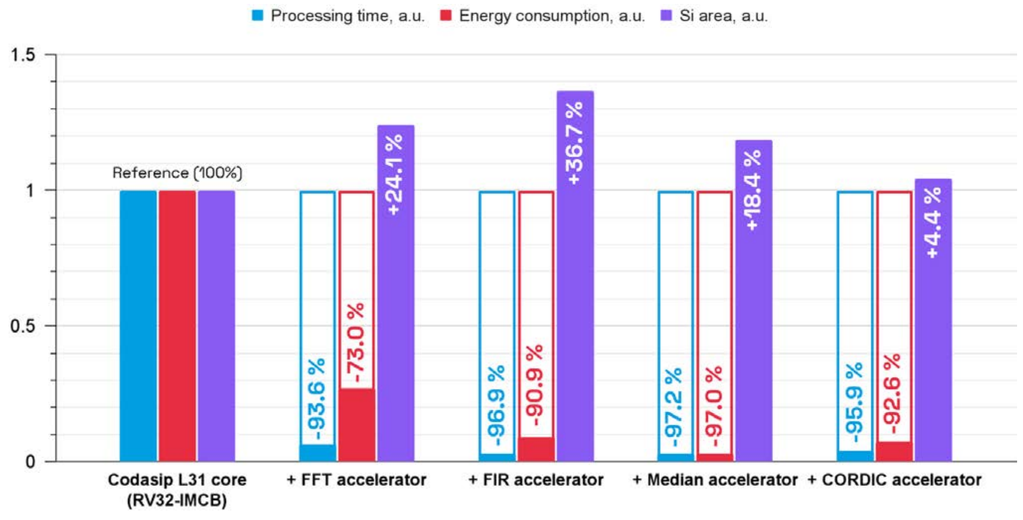
3 custom instructions call to start "butterfly" flow
42 cycles for fp32 result
20 cycles for fixed-point 32-bit

CORDIC accelerator



1 custom instruction call to start "CORDIC" flow
16 cycles to get the results

→ How DSP customization affects the PPA



DSP-specific customizations of embedded RISC-V cores that tackle several representative DSP algorithms have improved the performance and energy consumption by 14.4x & 0.27x (FFT), 14.4x & 0.1x (FIR), 30x & 0.03 (Median filtering) and 24.3x & 0.08x (Cordic), all at reasonable silicon area cost.

Customizations with CodAL and Codasip studio shorten the time to market by providing out-of-the-box SDK and HDK tools and more compact core description. The table lists the code size and the human effort required to implement the described DSP customizations.

HW module	Time to implement in CodAL	Lines of CodAL code	Lines of code in Verilog
1D FFT accelerator (256 samples)	2 person-weeks	500	2300
1D FIR filter	3 person-days	150	670
1D Median filter	3 person-days	160	1180
CORDIC module	3 person-days	210	600