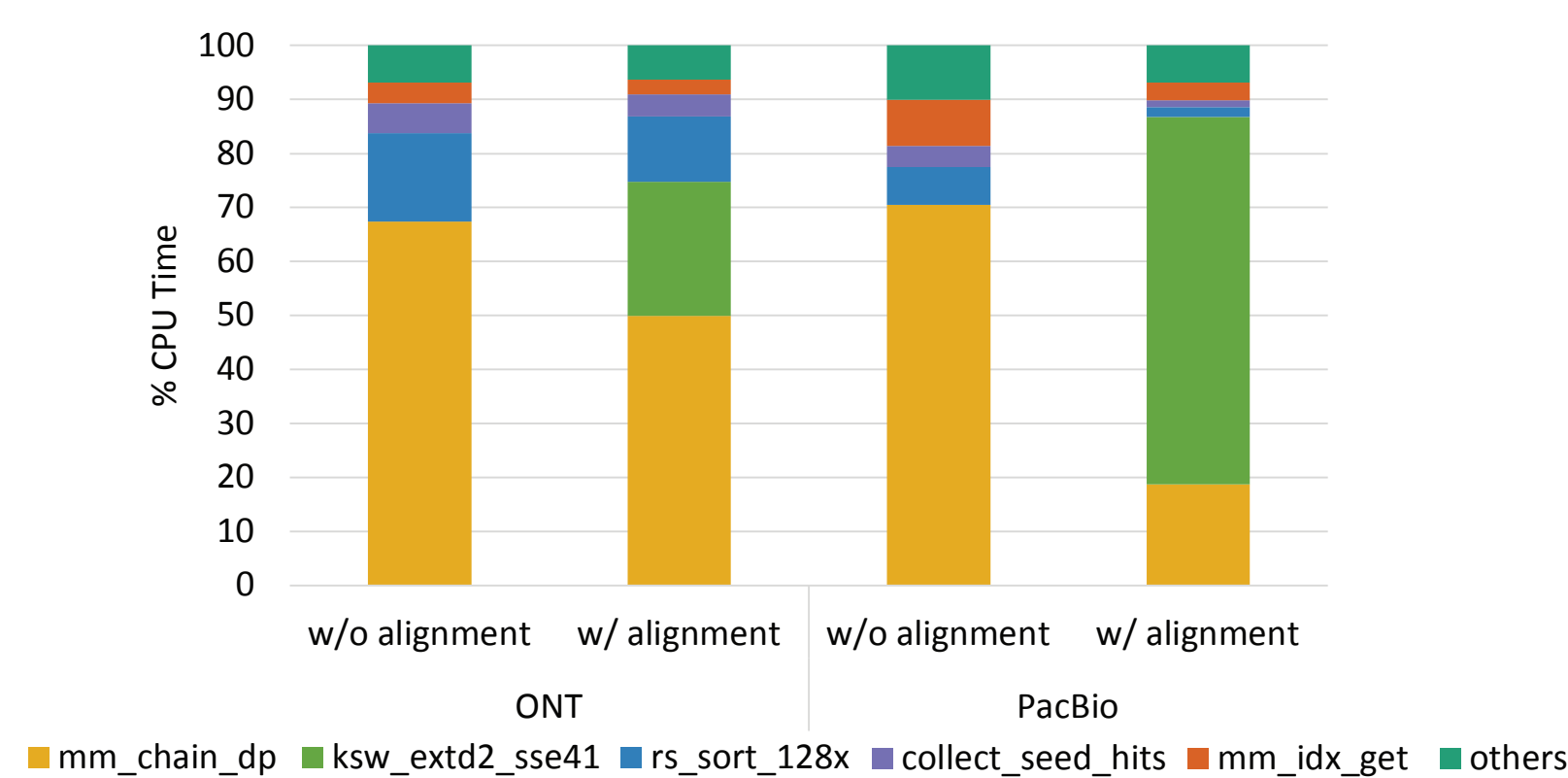


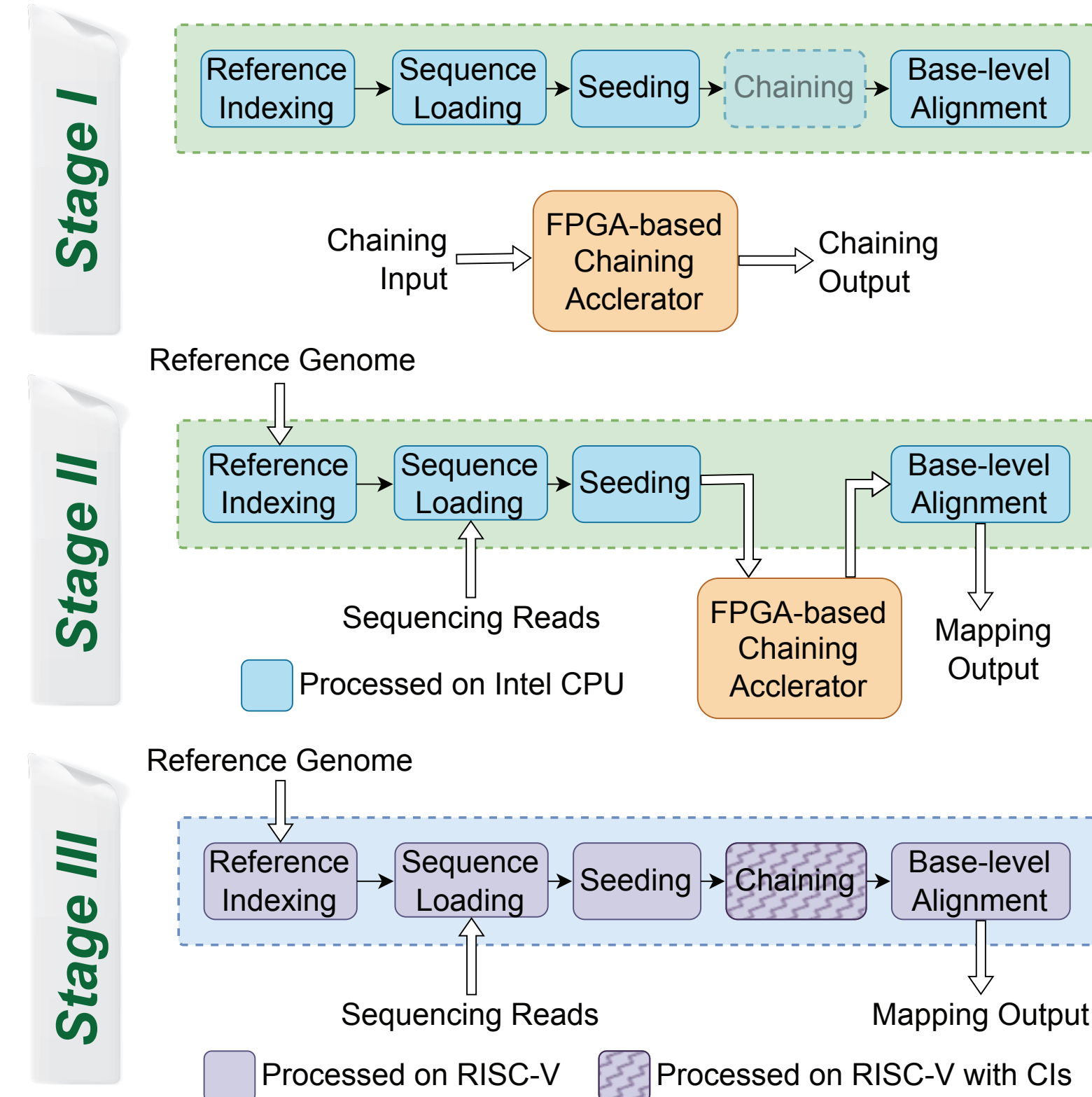
Introduction

Long-read Sequence Analysis

- Third-generation long-read sequencing is dubbed the Method of the Year 2022¹
- *minimap2*² is the gold-standard tool for third-generation sequence alignment
- While *minimap2* is fast, additional speed-up is desired, especially with large datasets
- Chaining in *minimap2* can take up-to ~70% of total run-time



Optimising *minimap2* Using HW-SW Co-Design



Contributions of this Thesis Work

Stage I:

- Isolated chaining step of *minimap2* is accelerated on FPGA-based heterogeneous system
- Achieves up to ~1.35x speed-up while consuming ~27% less energy than SIMD-based software

Stage II:

- Improved FPGA-based chaining step accelerator is integrated back into *minimap2* software
- Achieves up to ~1.8x speed-up in end-to-end time compared to original *minimap2*

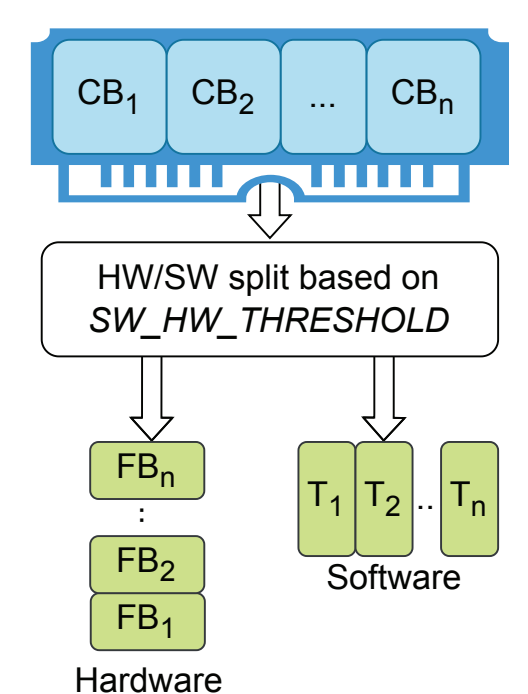
Stage III:

- Chaining step of *minimap2* is accelerated using custom instructions on RISC-V based ASIP
- Achieves up to 2.4x speed-up in chaining step

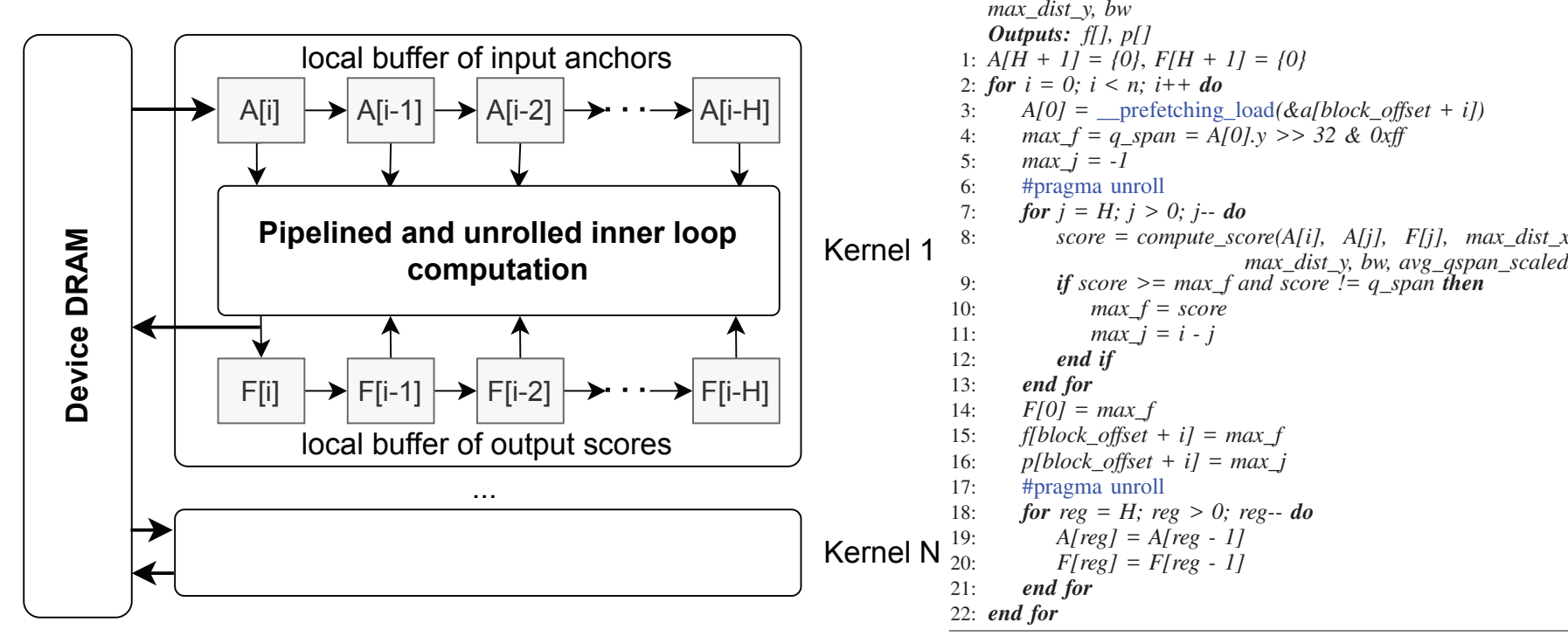
Accelerating Isolated Chaining Step with an FPGA-based Heterogeneous System

Methodology

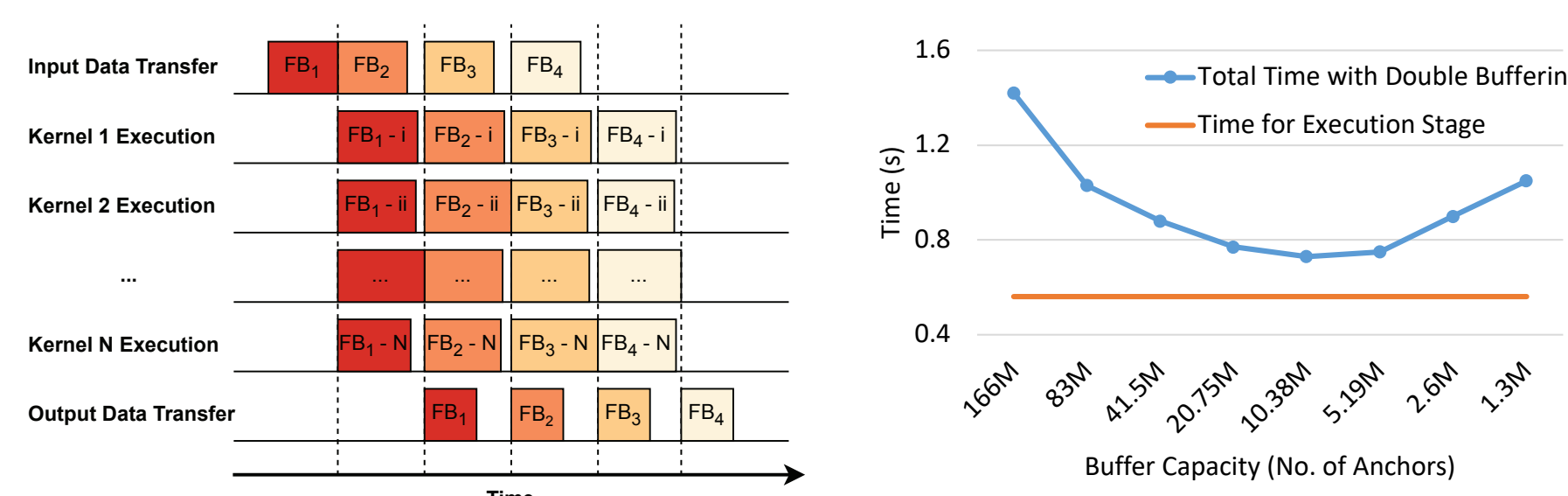
- CPU+FPGA heterogeneous system to accelerate chaining in isolation
- Heuristic algorithm to split chaining tasks between HW and SW based on compute complexity
- Multi-kernel OpenCL HLS-based custom hardware accelerator on FPGA
- Multi-threaded software execution on CPU



Hardware accelerator architecture

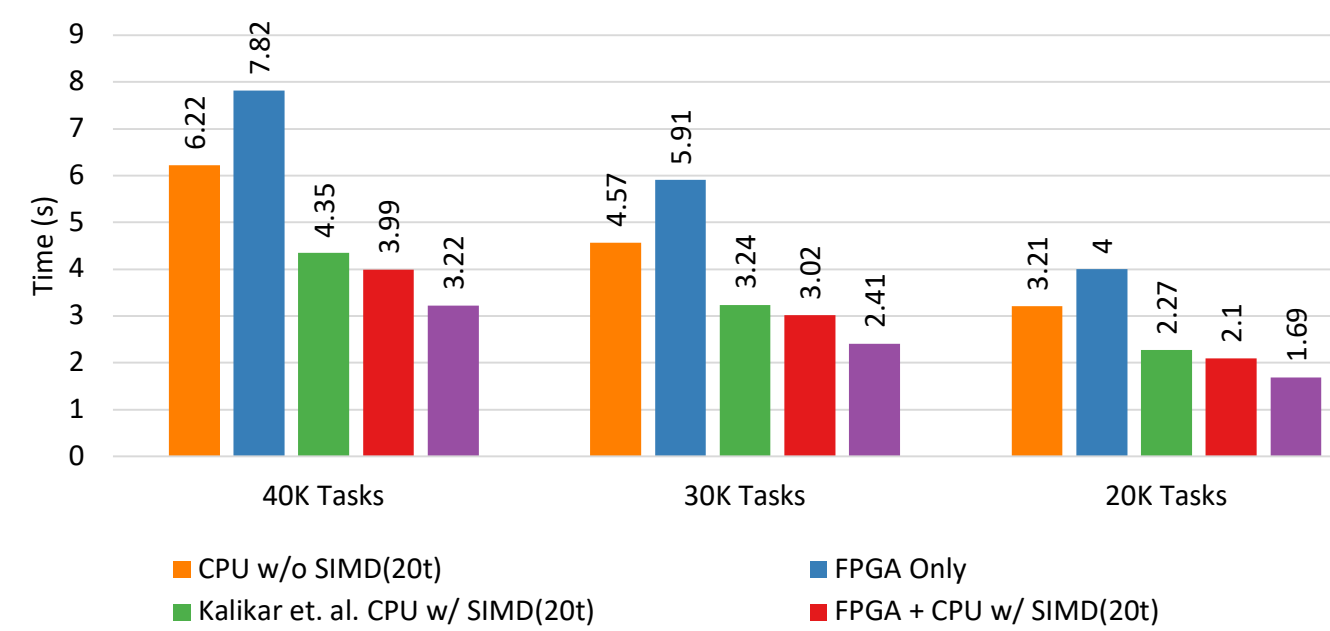


Fine-grained data batch-wise double buffering to FPGA



Results

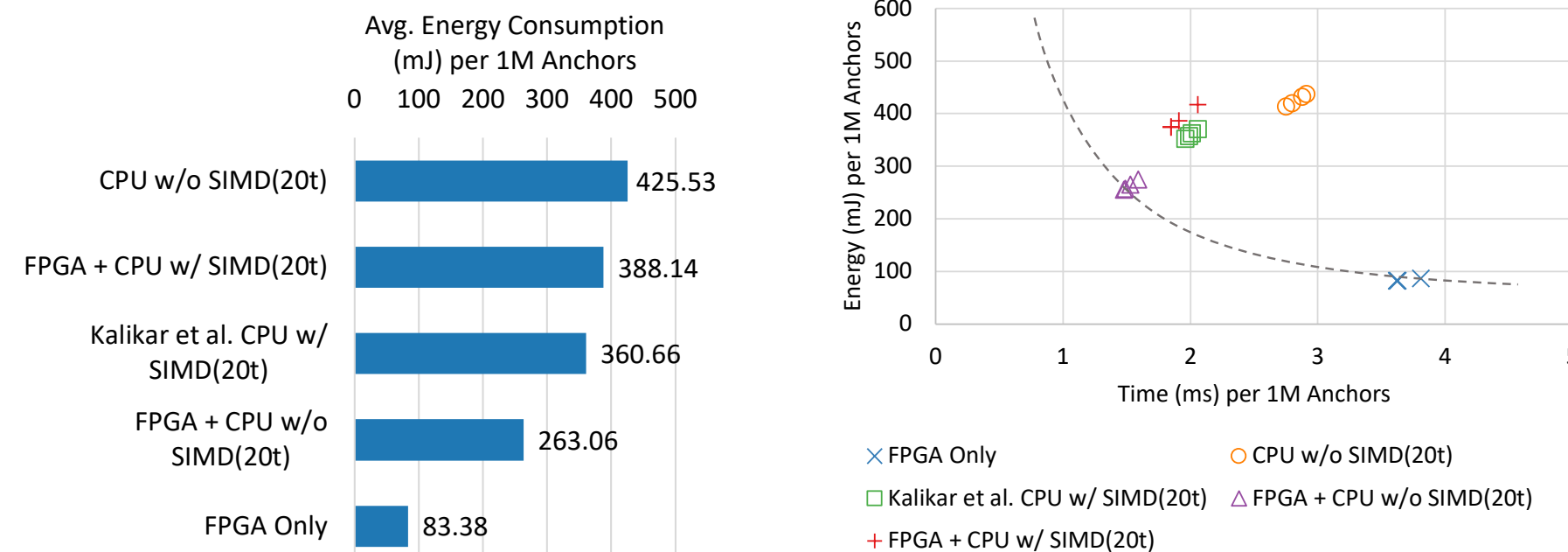
Performance comparison



More info:



Energy/energy-time comparison

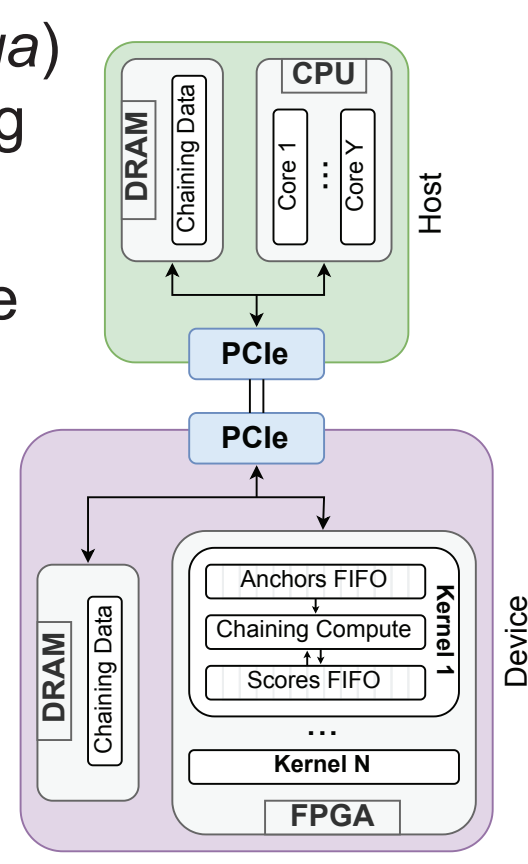


Our work (FPGA+CPU w/o SIMD) consumes ~27% less energy than CPU w/ SIMD and ~38% less energy than CPU w/o SIMD. It holds the best energy-time consumption.

End-to-end Integration of FPGA-based Chaining Step Accelerator

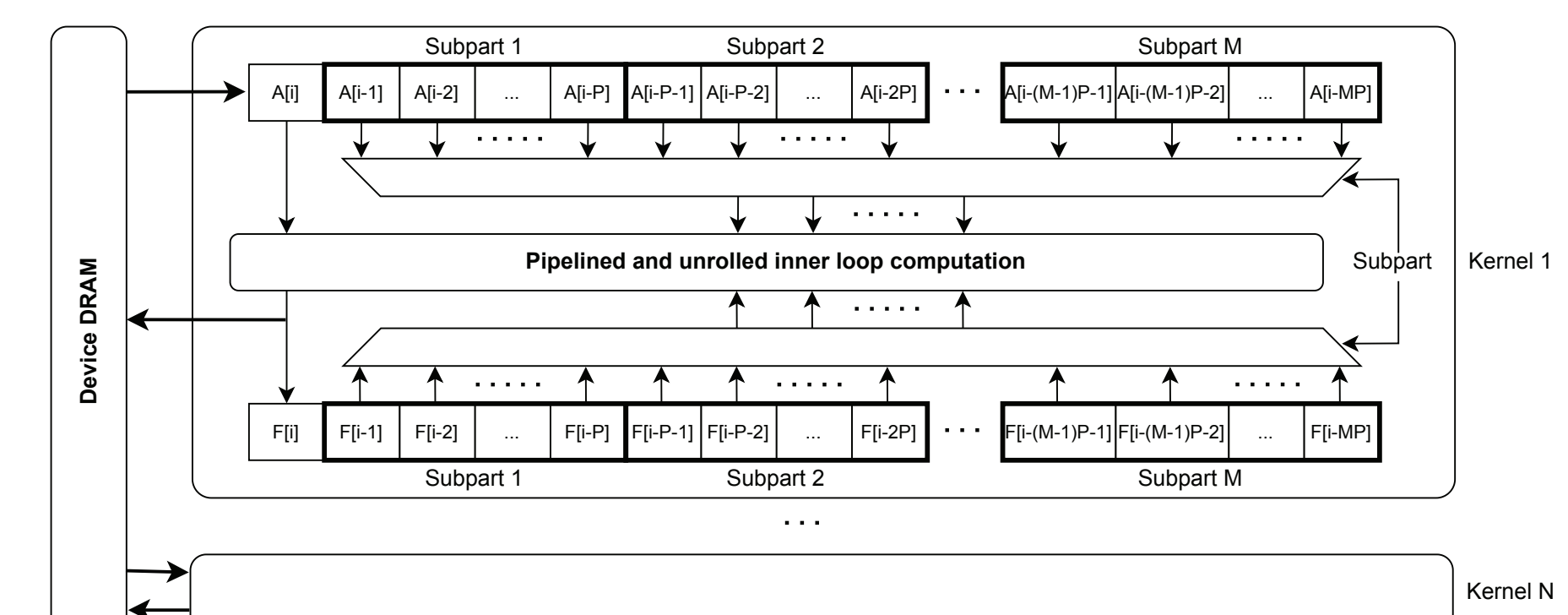
Methodology

- Fully end-to-end integrated tool (*minimap2-fpga*) that can be used in realistic sequence mapping workflows with large realistic datasets
- Host with multi-core (Y) CPU runs the software
- Device with multi-kernel (N) FPGA performs hardware-accelerated chaining
- Chaining tasks are dynamically scheduled on FPGA (hardware) or CPU (software) based on predicted execution and wait times
- Supports Intel/Xilinx, on-premise/cloud FPGA



GitHub: <https://github.com/kisarur/minimap2-fpga>

Improved hardware accelerator architecture



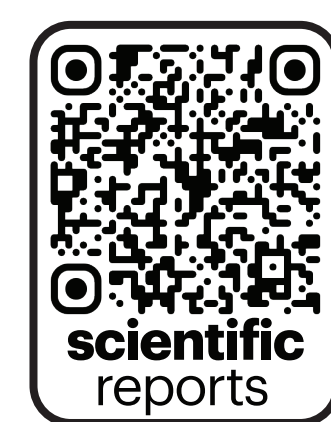
Hardware-Software chaining time prediction

$$T_{hardware} = T_{data_transfer} + T_{execution} = (K_1 \times n) + (II \times T_{clock} \times total_subparts) + C_1$$

$$T_{software} = K_2 \times \sum_{i=1}^n trip_count_i + C_2$$

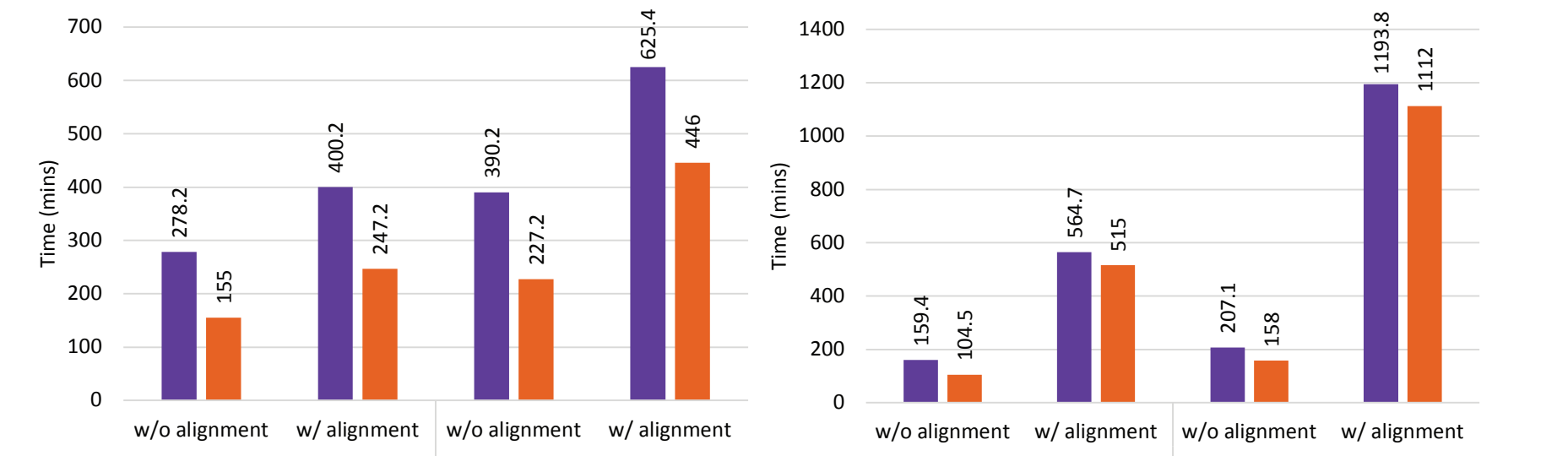
n - no. of chaining anchors K_1, K_2, C_1, C_2 - constants
 II - initiation interval of hardware pipeline
 T_{clock} - clock period of hardware

More info:

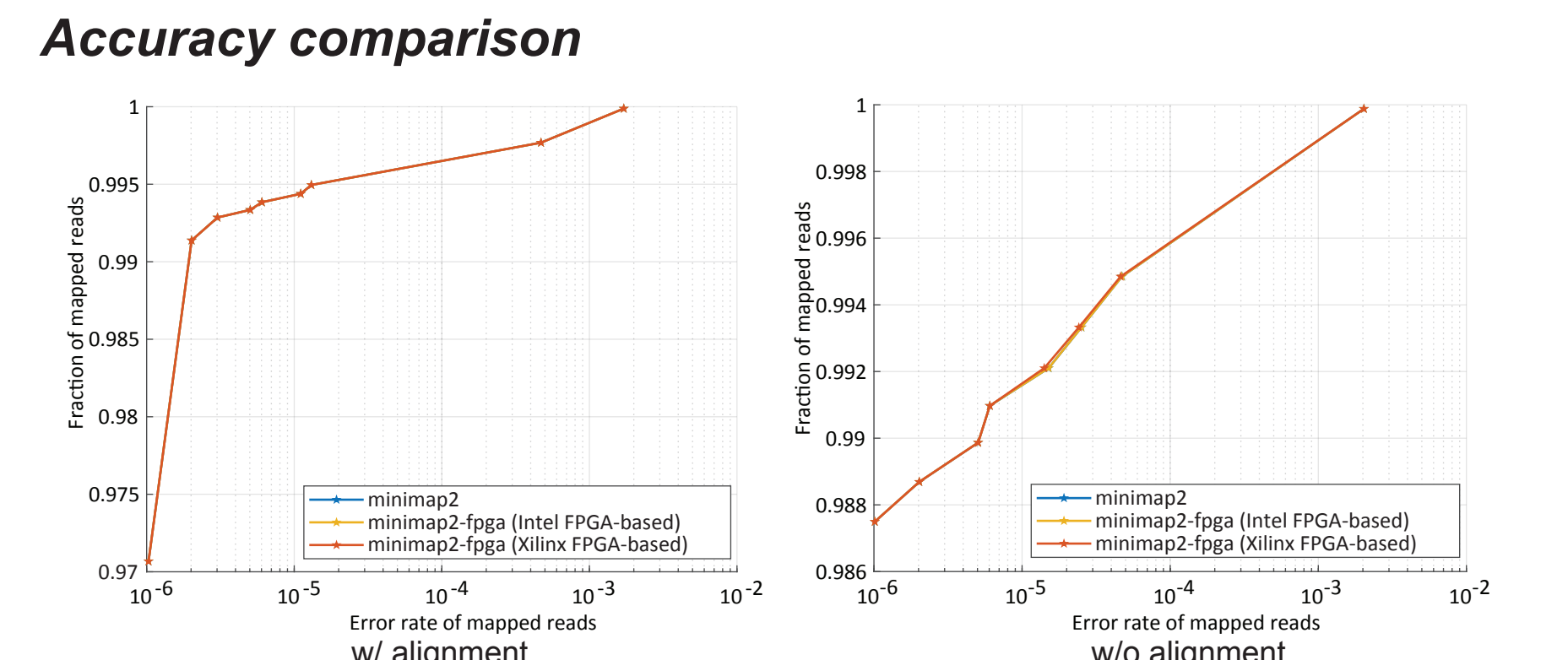


Results

Performance comparison



Accuracy comparison

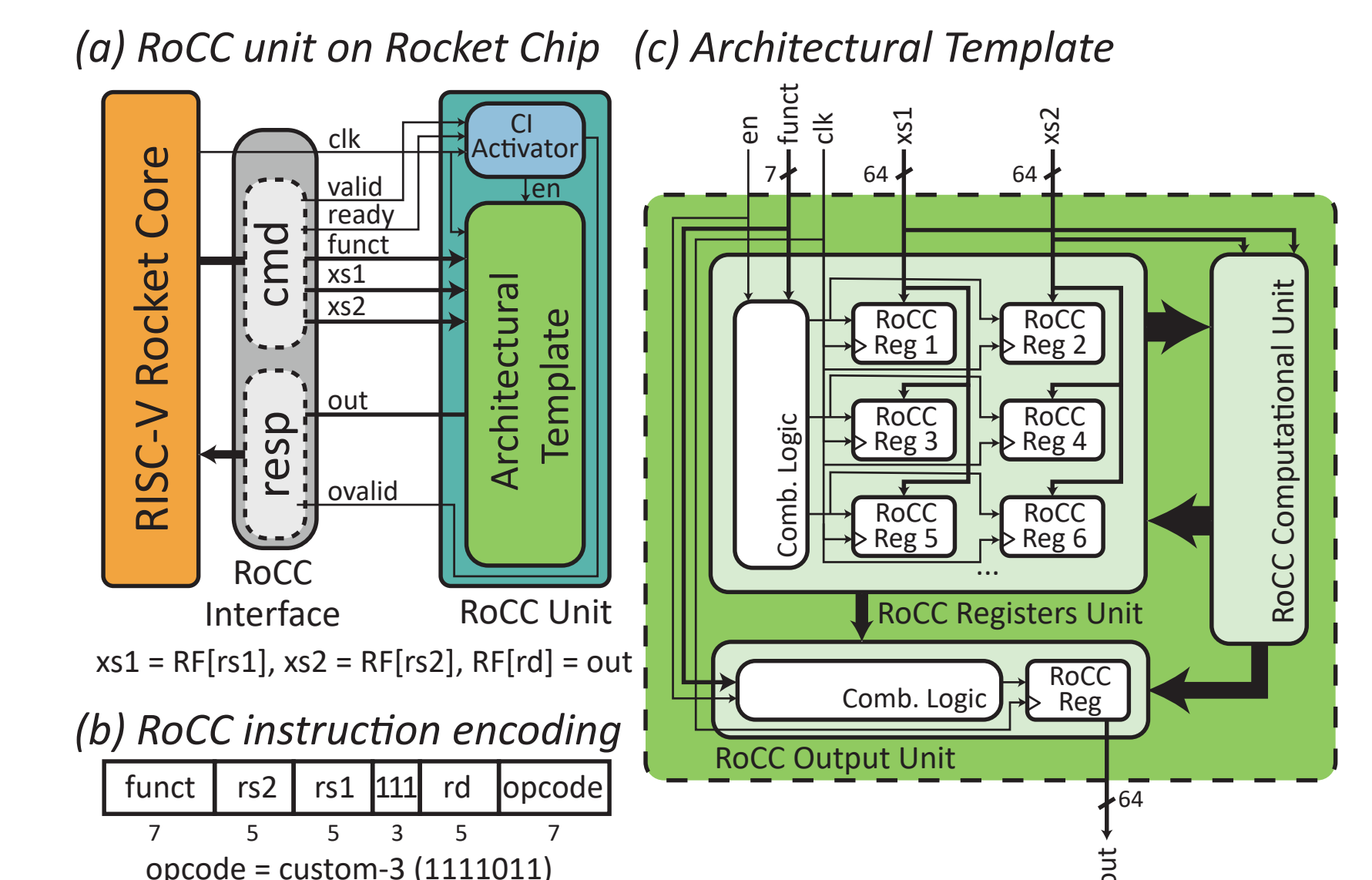


Accelerating Chaining Step Using Custom Instructions on Embedded RISC-V

Methodology

- Chaining step is accelerated using custom instructions on a RISC-V based ASIP
- A novel architectural template is introduced for designing custom instructions
- A heuristic algorithm is used to identify and map suitable code segments from C code to the architectural template
- Custom instructions are implemented for *minimap2*'s chaining step using HDL on RISC-V Rocket Chip with RoCC
- Modified Rocket Chip is configured on Xilinx FPGA and *minimap2* with custom instructions is run on it

Hardware architectural template within RoCC

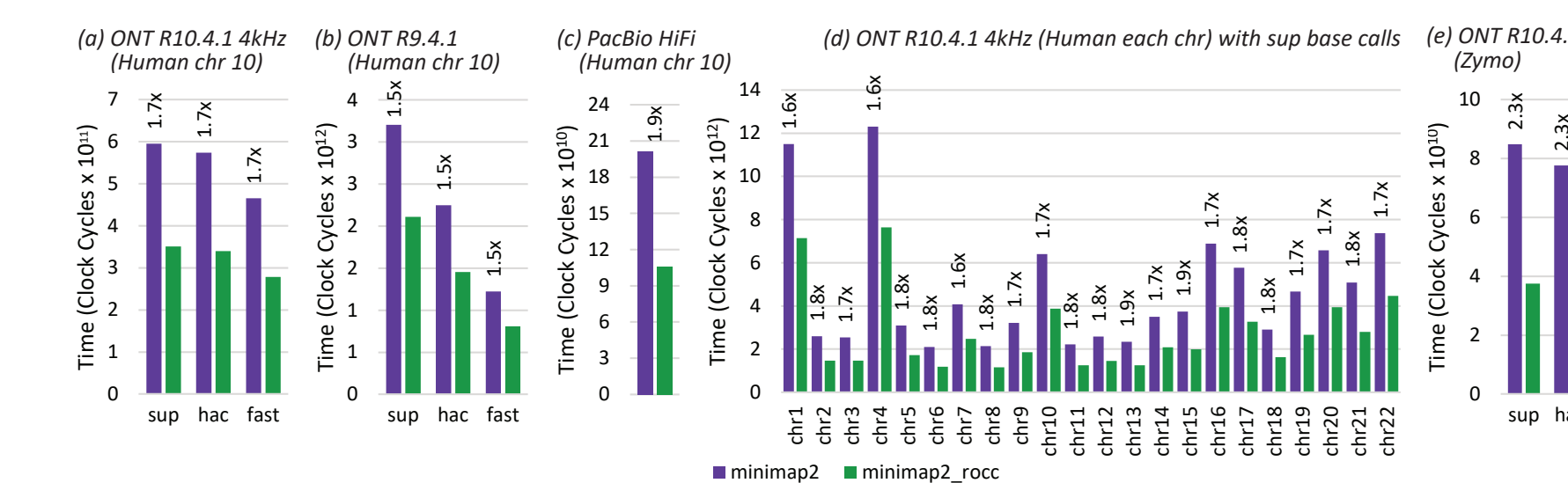


Custom instruction extraction process

1. Profile code to identify hot loops (note: 1 hot loop for chaining)
2. Extract C code blocks from hot loop targeting architectural template for conversion into CI
3. Manually convert extracted code segments into CIs
4. Enhance original code with CIs and record performance on Spike simulator
5. Iteratively unroll the hot loop and do steps 2 - 4 until performance is a maximum

Results

Performance comparison



FPGA resource utilisation of customised Rocket Chip

	LUTs	FFs	DSPs	BRAMs
Customised Rocket Chip	35832	18306	19	49
RoCC Unit	910	611	4	0

References

- [1] Marx, V. Method of the year: Long-read sequencing. *Nat. Methods* 20, 6–11 (2023)
 [2] Li, Heng. "Minimap2: pairwise alignment for nucleotide sequences." *Bioinformatics* 34.18 (2018): 3094-3100

Contact

kisarul@unsw.edu.au

