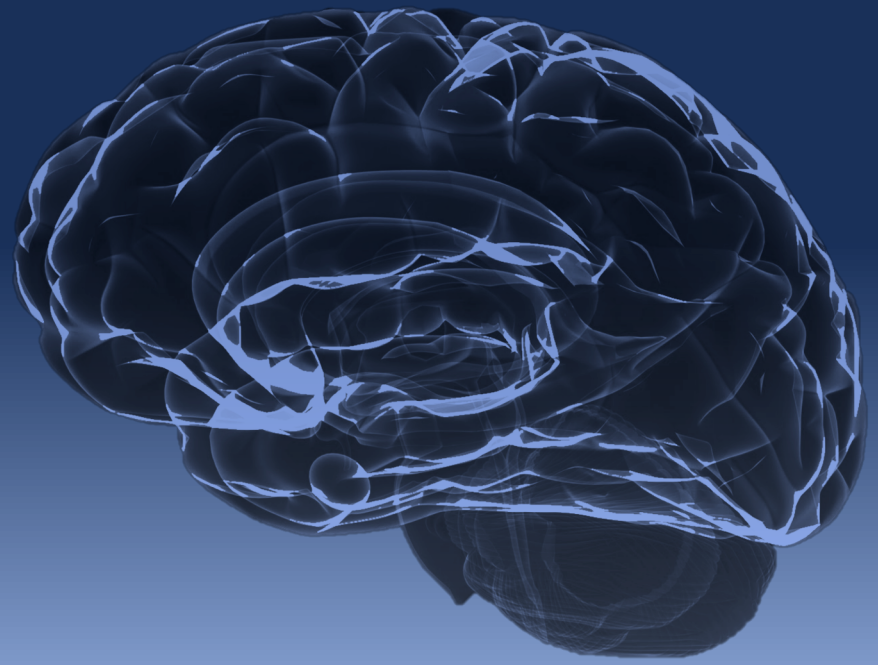


MasterMind

Many-Accelerator SoC Architecture for Real-Time BCI

ICCD 2021

Paper ID: 142

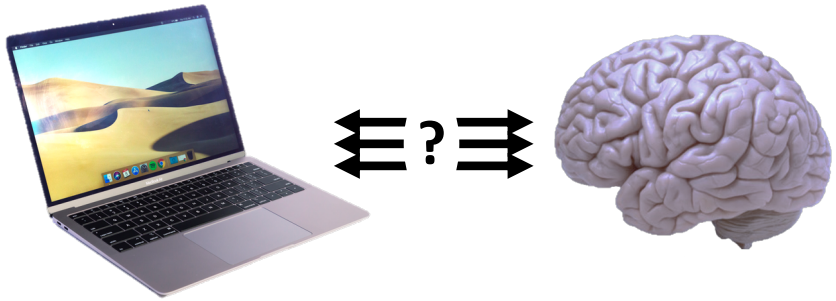
Guy Eichler, Luca Piccolboni, Davide Giri, Luca Carloni

Columbia University, New York, USA

COLUMBIA UNIVERSITY

CS @CU COMPUTER SCIENCE

Introduction



- Brain-Computer Interfaces (BCI) enable the bidirectional interaction between brains and computers
- We want BCI systems/devices that can work in a body area network (BAN):
  (1**) Real-time** goal imposed by the 0.18 sec reaction time of the brain
  (2) **Ultra-low power** constraint of 200mW on an average workload (for BCI wearables)

NEURALINK

BLACKROCK MICROSYSTEMS CTRL-Labs

- We used a system-on-chip (SoC) design platform called ESP
- MasterMind is a configurable accelerator-based SoC architecture designated for the execution of BCI algorithms (HiWA)
- We contributed our broad design-space exploration (DSE) including efficient implementations of the hardware accelerators and the SoC
- We released MasterMind into the public domain

ESP - https://esp.cs.columbia.edu/
MasterMInd - https://github.com/GuyEichler/esp/tree/mastermind

Background

- Hierarchical Wasserstien Alignment (HiWA) was presented at NeurIPS 2019

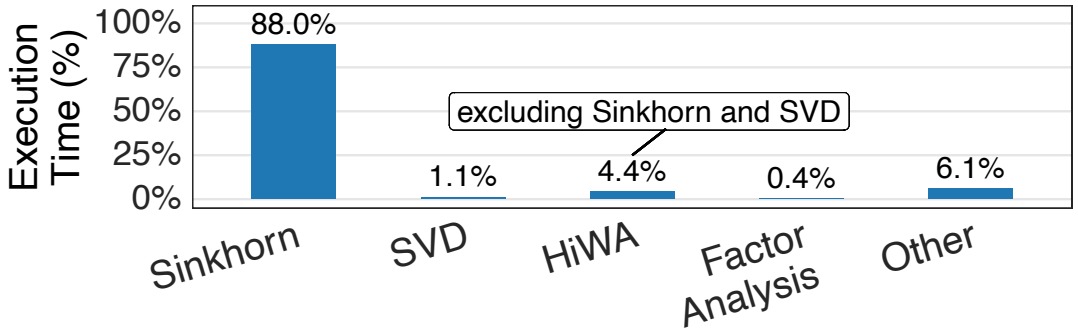
- Machine learning algorithm
  → Successfully matches neural activity from the brain to body movements

1: **global variables:** $p, q, m, \gamma, max\_iter$

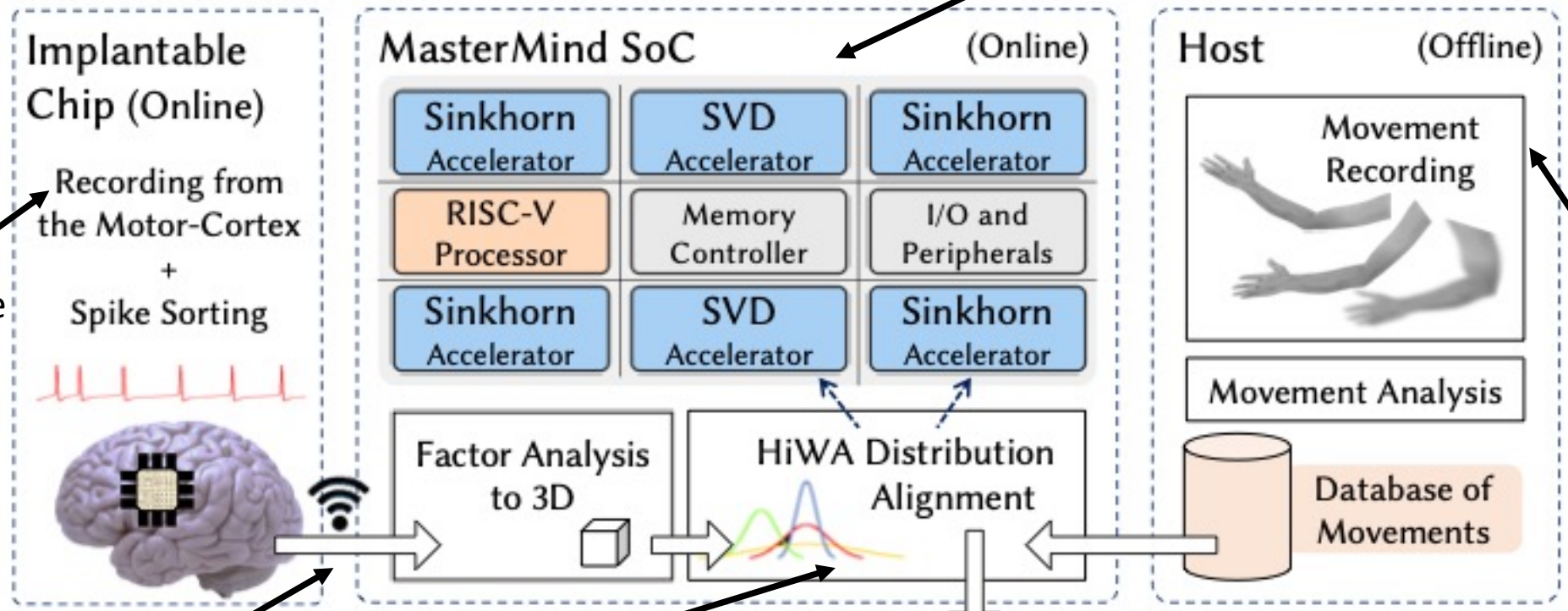
2: **function** HiWA_CORE($X_{p\times m}, Y_{q\times m}, P_{1\times 1}, T_{m\times m}$)
3: Initialize $Q_{p\times q} = \mathbb{1}_{p\times q}/(pq)$
4: Initialize $R_{m\times m} = \mathbb{1}_{m\times m}$
5: **while** $||R - R_{prev}|| > 0.01$ **do**
6: $R, XR, Y^T \leftarrow$ SVD(X, Y, T, P, Q)
7: $Q, sum(C * Q) \leftarrow$ SINKHORN(XR, Y^T)
8: **return** $R, sum(C * Q)$

- We profiled the original implementation of HiWA

- The loop that invokes **Sinkhorn** and **SVD** contains most of the computation of the algorithm and can be isolated
  → Offload into hardware



The Target System (SoC)



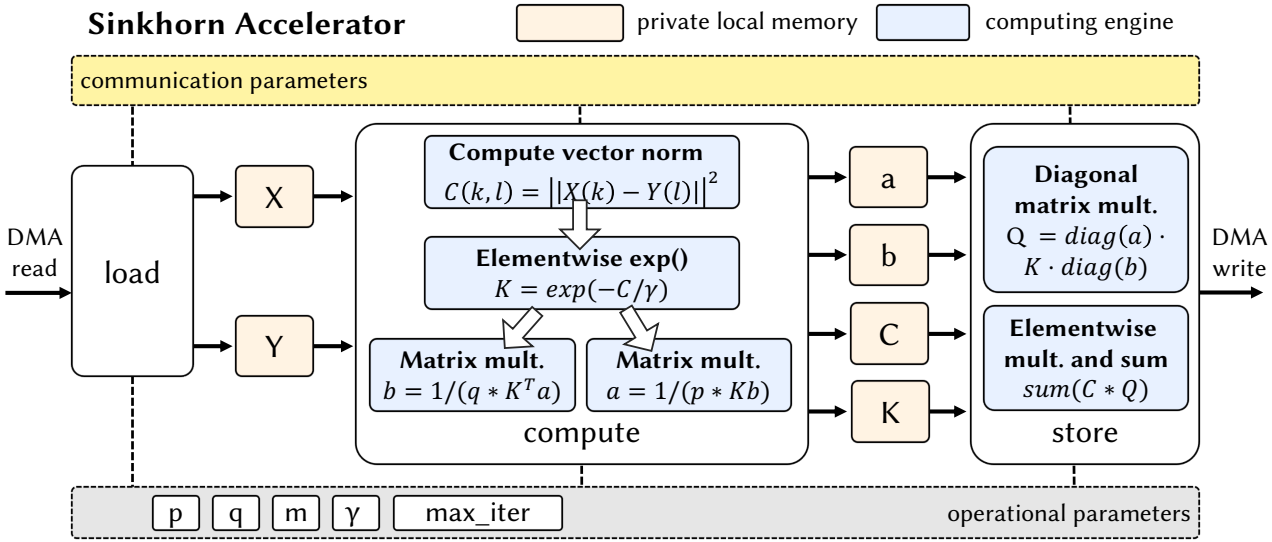
Spike Sorting – M.S. Lewicki, "Network: Computation in Neural Systems", 1998

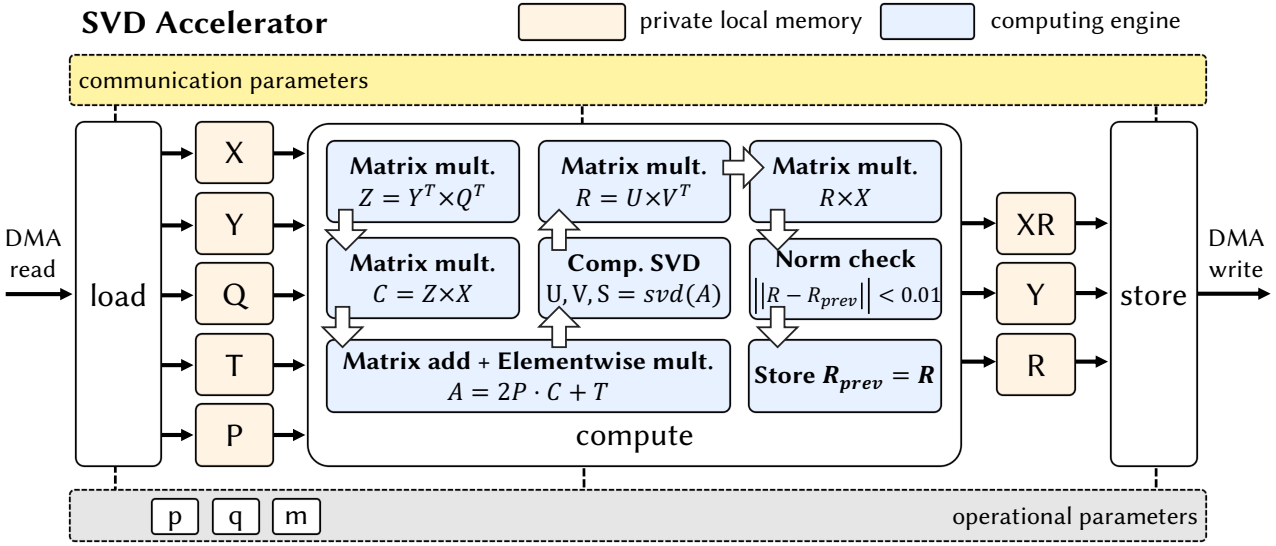
Accelerators Design and Data-Flow

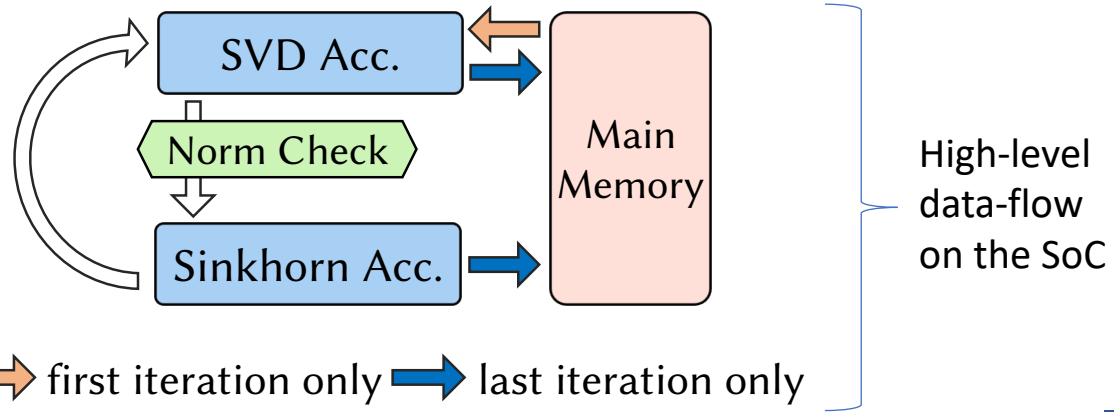
- 3-modules structure: *load, compute, store*
- **SVD** → C/C++, Vivado HLS
  Sinkhorn → SystemC, Stratus HLS

- P2P communication between accelerators
  → Minimal interaction with main memory

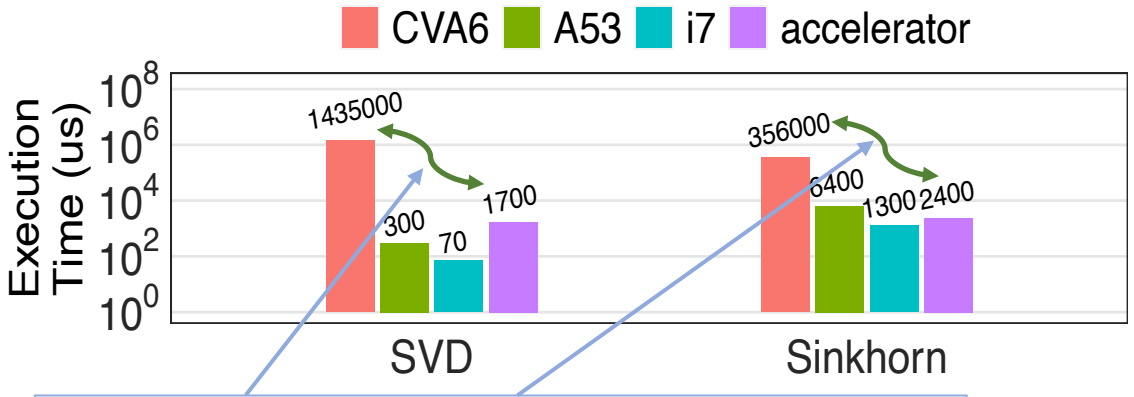
- **DSE** at the accelerator level and at the SoC level







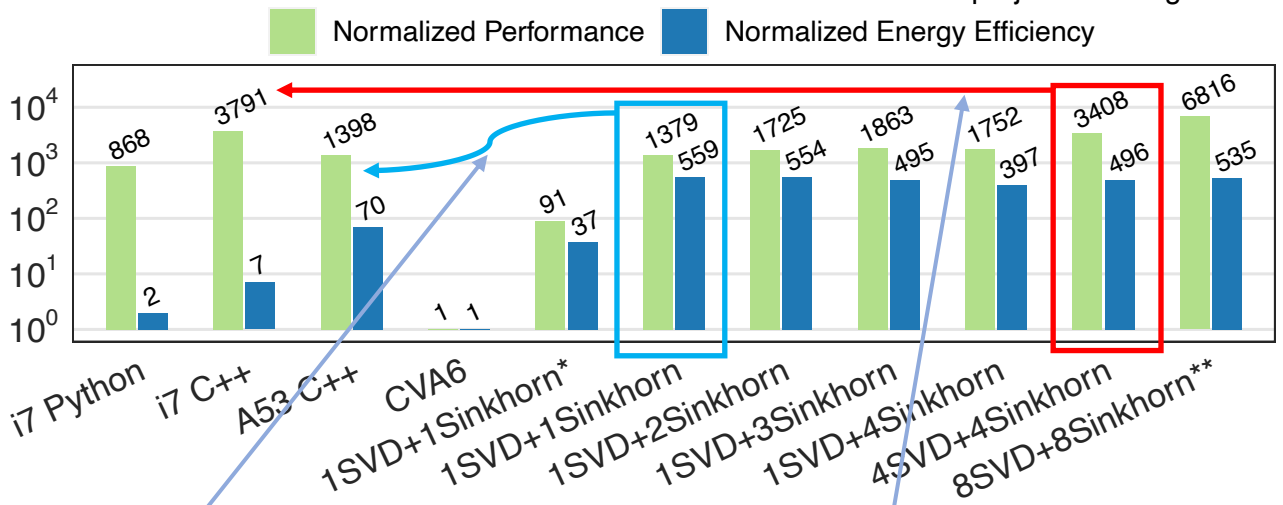
Evaluation – SoC Configurations



Significant speedup compared to software on CVA6

Software on i7 and ARM is still faster

* is for configurations without p2p
** is for projected configurations



Similar performance to ARM with 8x in energy efficiency

Similar performance to i7 with 71x in energy efficiency

- Evaluated against 3 platforms: **Intel i7, ARM Cortex-A53, and RISC-V CVA6**
- Clock frequencies: FPGA @ 78MHz, ARM @ 1.2 GHz, Intel @ 3.7GHz
- **Optimized multi-threaded software applications in C++** running on the general-purpose processors
- P2P provides **over 90% memory accesses savings** → **Minimizes off-chip energy consumption**
- An ASIC projection with a clock frequency of 1GHz
  → **13x speedup and 79x better energy efficiency over the FPGA prototype**

→ **Meets the thresholds for real-time energy-efficient BCI**

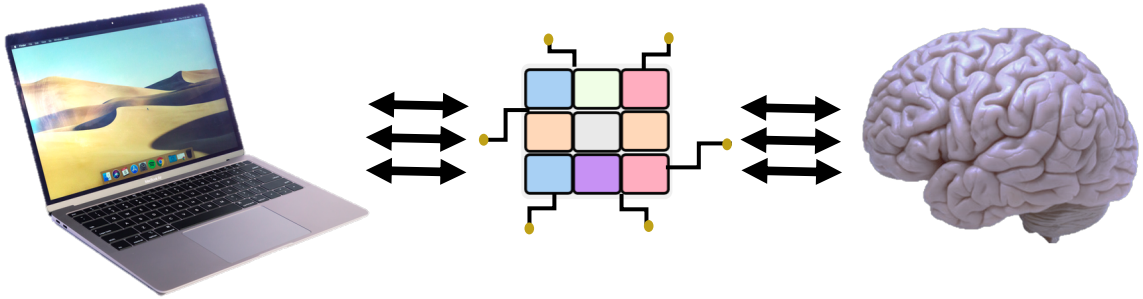
CVA6 RISC-V CPU - https://github.com/openhwgroup/cva6

Conclusions

- We designed MasterMind with the goal of advancing the system-level design research in SoC architectures for brain-computer interfaces (BCI), a field of computer engineering that is growing remarkably in importance
- MasterMind is inherently flexible, as it can seamlessly accommodate the integration of many other accelerators
- MasterMind is scalable, as it supports efficient point-to-point communication schemes among accelerators that improve performance and energy efficiency by reducing memory accesses
- We released the contributions of this work to the public domain: https://github.com/GuyEichler/esp/tree/mastermind

Thank you!

Questions?



MasterMind

Many-Accelerator SoC Architecture for Real-Time BCI

Speaker: Guy Eichler

Columbia university, NY

COLUMBIA UNIVERSITY

CS@CU COMPUTER SCIENCE