UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Accelerated Calculation of Electron Repulsion Integrals on FPGAs using oneAPI

Xin Wu, Tobias Kenter, Robert Schade, Thomas Kühne, Christian Plessl

Paderborn University, Germany
Paderborn Center for Parallel Computing

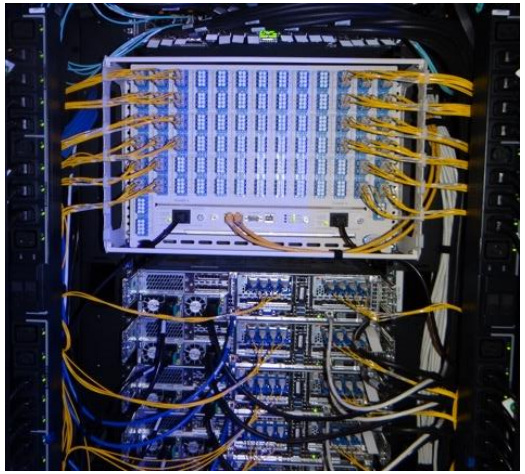
NATIONALES HOCHLEISTUNGS RECHNEN

Paderborn Center for Parallel Computing

Scientific Computing Accelerated on FPGA, Saclay, 2022-07-06

FPGAs in Noctua 2 at Paderborn Center for Parallel Computing

- Supercomputer Noctua 2
  - Atos Bull Sequana XH2000
  - 1124 nodes, each with 2x AMD Milan 7763
  - 128 NVIDIA A100 GPUs
  - **48 Xilinx Alveo U280 FPGA accelerators**
    - **16 nodes, each with 3x Xilinx Alveo U280 cards**
    - **32 GiB DDR and 8 GiB HBM2**
  - **32 BittWare 520N with Intel Stratix 10**
    - **16 nodes, each with 2x BittWare 520N cards**
    - **32 GiB DDR**
  - configurable point-to-point connections to any other FPGA
- Worldwide leading academic installation of FPGAs for HPC
- System access application https://pc2.uni-paderborn.de/go/access

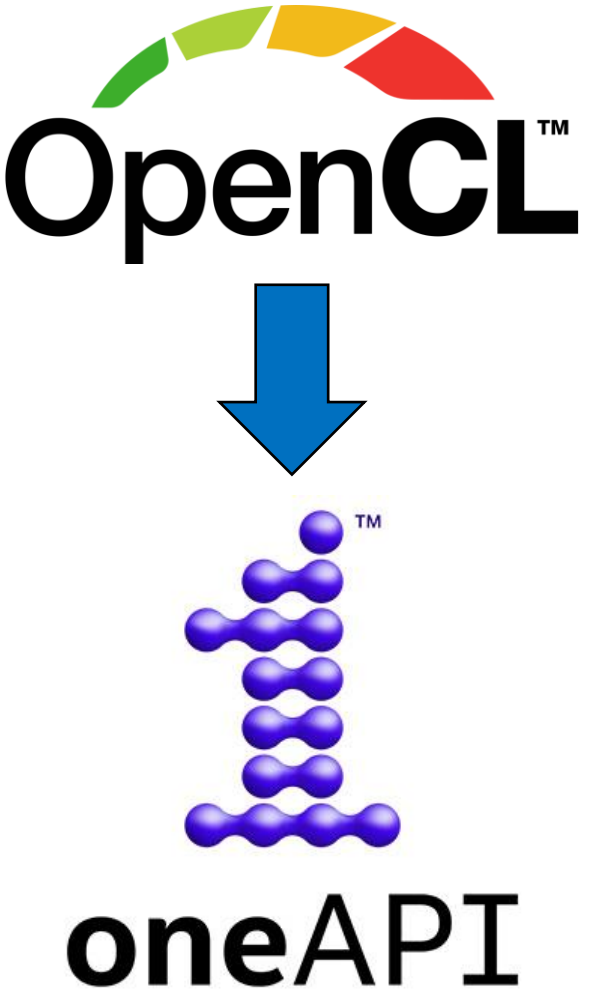


Agenda

- Introduction
- Background on Method
  - mathematical definition
  - Rys quadrature
- FPGA Design and Implementations
  - custom local memory layout
  - fully unrolled loops for recurrence relations
  - optimization for global memory stores
- Results and Discussion
  - resource consumptions
  - performance model and analysis
- Conclusion and Future Work

Introduction

- In recent years FPGAs receive increased attention as accelerator for scientific computing.
- Selected applications at Paderborn Center for Parallel Computing (PC²)
  - Intel FPGA SDK for OpenCL
    - Shallow-water simulation
    - N-body simulation method
  - Intel FPGA Add-on for oneAPI Base Toolkit
    - StencilStream Library
    - **Electron repulsion integrals (ERIs)**
      → as the nightmare of integrals by John Pople in Nobel Prize Lecture
- Benefits of FPGA using oneAPI
  - DPC++ function templates for many variants of ERIs
  - unrolled loop structures for 2-index recurrence relations

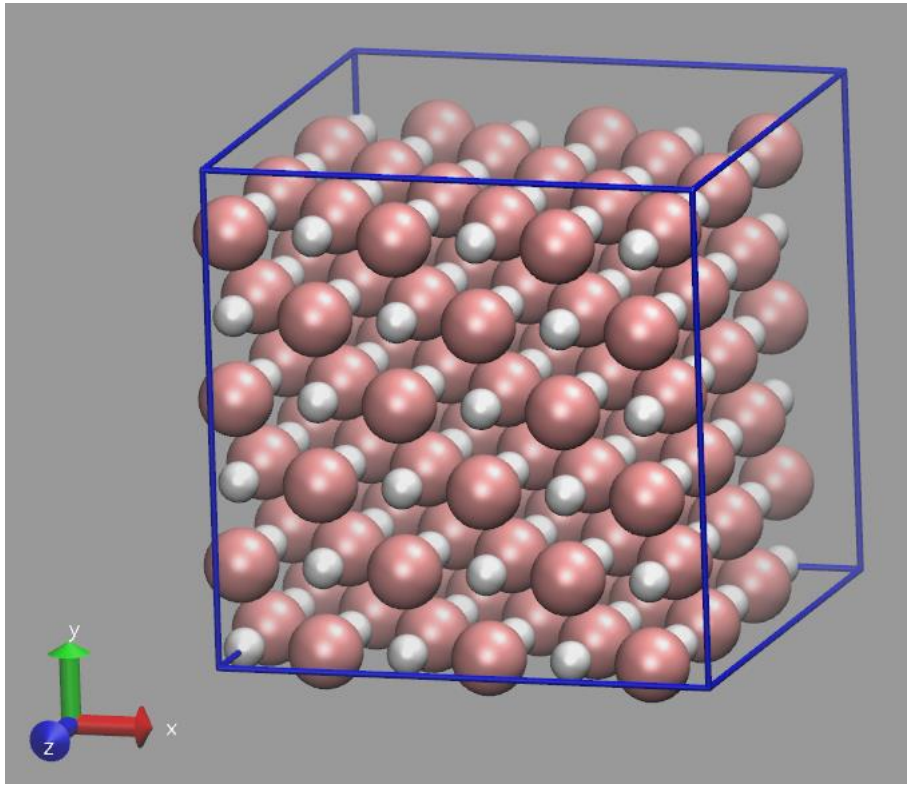


- T. Kenter et. al: *Algorithm-Hardware Co-design of a Discontinuous Galerkin Shallow-Water Model for a Dataflow Architecture on FPGA*. PASC'21.
- J. Menzel et. al: *The Strong Scaling Advantage of FPGAs in HPC for N-body Simulations*. ACM Trans. Reconfigurable Technol. Syst.'21.
- StencilStream Library: https://github.com/pc2/StencilStream

Background on Method

Background on Method

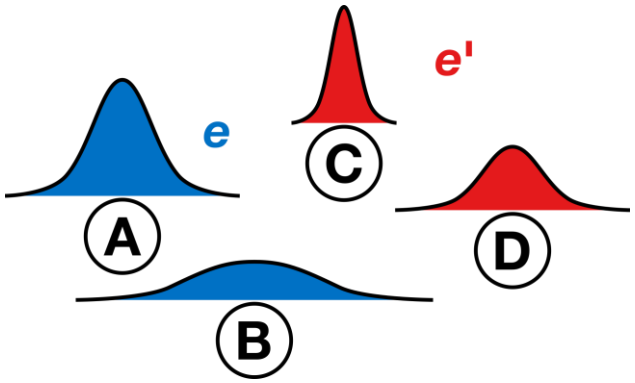
- Computation of the enormous amount of ERIs forms a major bottleneck in hybrid density functional theory calculation.
- LiH benchmark in CP2K
  - 216 atoms in a cubic box
  - 18.7 trillion ERIs (the OPT2 basis)
- LiH benchmark on Noctua 1
  - 16 compute nodes
  - each node with 2x Intel Xeon Gold Skylake 6148
    - 40 CPU cores per node
    - Hyper-Threading is disabled
  - pure MPI parallelization
    - 640 MPI processes
  - interconnect: Intel Omni Path 100 Gbps
  - **computation of ERIs → 29% elapsed walltime**



Mathematical Definition

- **Electron Repulsion Integral: 6-D integration**

$$[\mathbf{ab}|\mathbf{cd}] = \int \mathrm{d}\mathbf{r} \int \mathrm{d}\mathbf{r}' \; g_{\mathbf{A},\mathbf{a},\alpha}(\mathbf{r})\, g_{\mathbf{B},\mathbf{b},\beta}(\mathbf{r}) \frac{1}{|\mathbf{r}-\mathbf{r}'|} g_{\mathbf{C},\mathbf{c},\gamma}(\mathbf{r}')\, g_{\mathbf{D},\mathbf{d},\delta}(\mathbf{r}')$$



  - $\mathbf{r} = (x, y, z)$: coordinates of *e*
  - $\mathbf{r}' = (x', y', z')$: coordinates of *e'*

- $g_{\mathbf{A},\mathbf{a},\alpha}(\mathbf{r})$: 3-D Cartesian Gaussian function centered at atom A

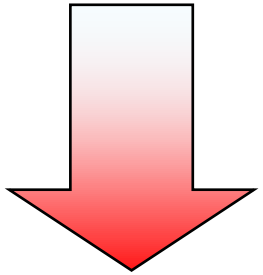
$$g_{\mathbf{A},\mathbf{a},\alpha}(\mathbf{r}) = (x - A_x)^{a_x} (y - A_y)^{a_y} (z - A_z)^{a_z} \exp\left(-\alpha |\mathbf{r} - \mathbf{A}|^2\right)$$

  - $\mathbf{A} = (A_x, A_y, A_z)$: coordinates of atom A
  - $\mathbf{a} = (a_x, a_y, a_z)$: angular momentum in x, y, z
  - α: exponent of Gaussian function

ERI Quartet

- **ERI quartet:** $[ab|cd]$
  - collection of all $[\mathbf{ab}|\mathbf{cd}]$ with the same angular momenta
  - must be computed together to maximize data re-use
  - 256 variants of ERI quartet are normally required in Physics/Chemistry applications.

ID	notation	number of integrals	FLOPs	loops
0	$[ss|ss]$	1	about tens	a few
1	$[ss|sp]$	3		
2	$[ss|sd]$	6		
3	$[ss|sf]$	10		
		… …		
255	$[ff|ff]$	10000	several hundreds of thousands	complicated nested



challenges:

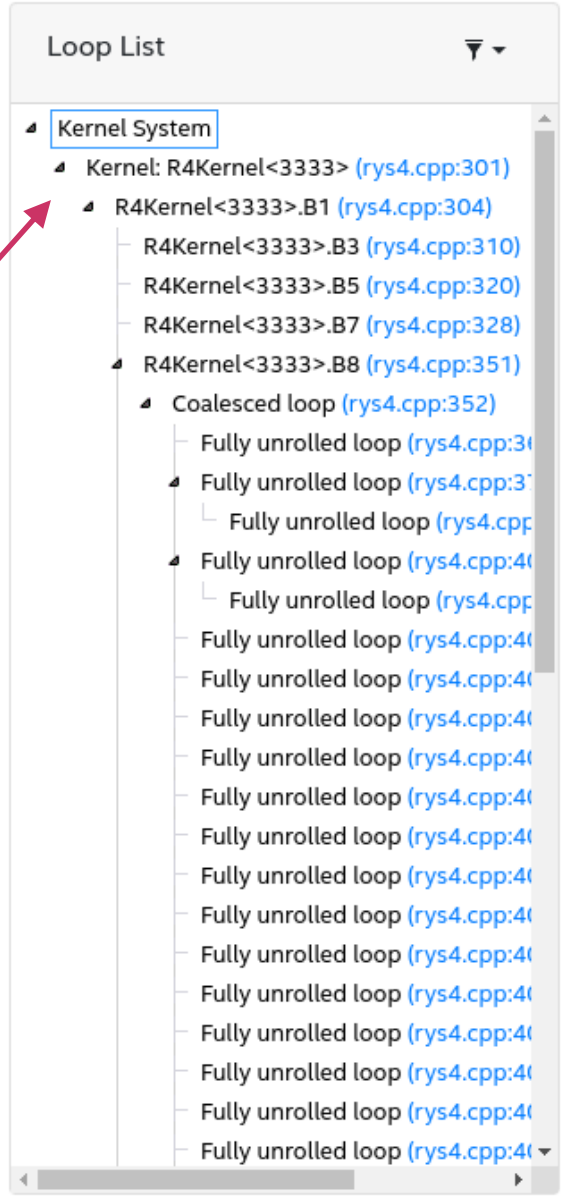
- 256 variants
- different FLOPs
- complicated loop structures

→ DPC++ function template for FPGA kernels

DPC++ Function Template for FPGA Kernels

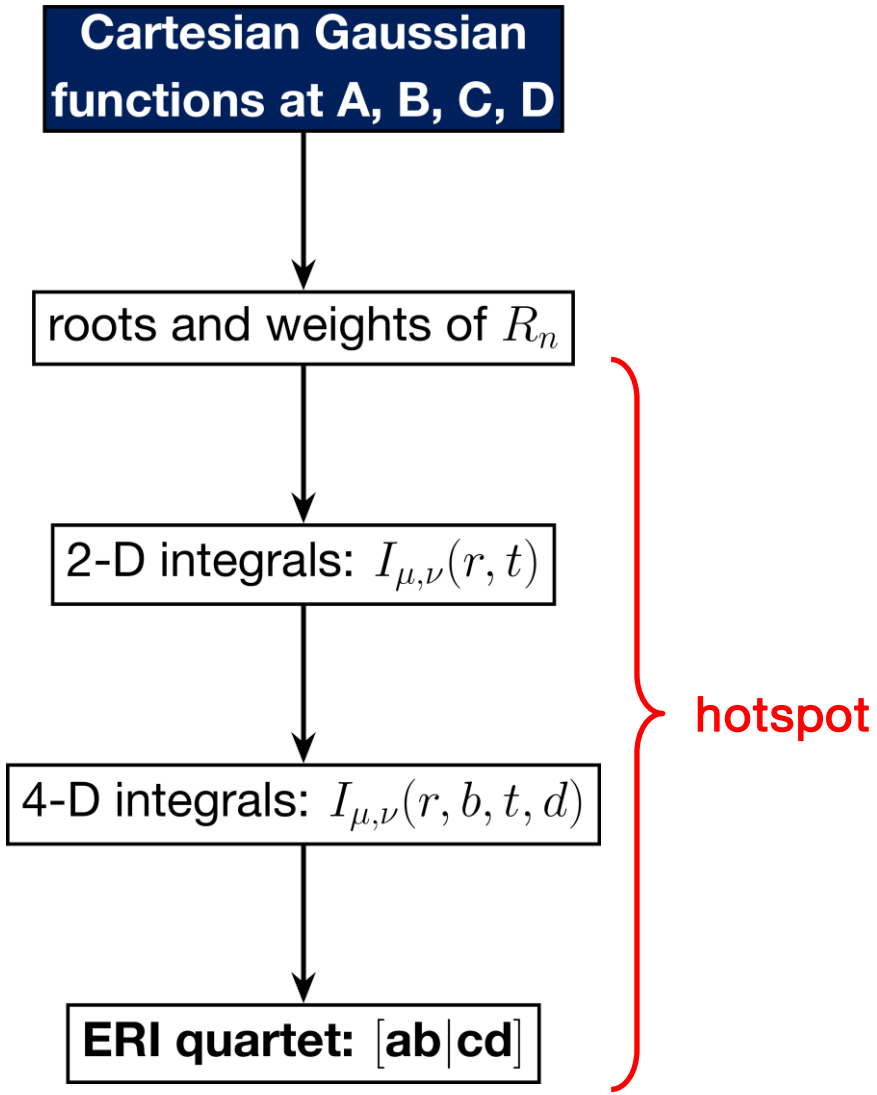
```
template<class Real, int a, int b, int c, int d>
void Rys4Compute( ... )
{
  /*
   * compile-time constants for
   * - custom FPGA local memory designs
   * - complicated unrolled loop structures
   */
  constexpr int d2a { smallest_power_of_2(a) },
                nga { (a + 1) * (a + 2) >> 1 };

  q.submit([&](sycl::handler& h)
  {
    h.single_task<R4Kernel<1000 * a + 100 * b + 10 * c + d>>([=]()
    {
      /*
       * FPGA kernel for the ERI quartet
       */
    });
  });
}
```



Rys Quadrature

- Three major algorithms
  - the McMurchie-Davidson algorithm
    → in `SHARK` of ORCA 5
  - the Head-Gordon-Pople algorithm
    → in `libint` and used in CP2K
  - the Rys quadrature algorithm
    → in `libcint`
- Rys quadrature
  - compute $[ab|cd]$ via Gaussian quadrature using a set of orthogonal Rys polynomials
  - features:
    - efficient for ERI quartet of higher angular momentum
    - low memory requirement for intermediate results
      → use of fast on-chip memories in FPGA
    - numerically very stable in computation
      → allows single-precision floating-point arithmetic



J. Rys et. al: *Computation of Electron Repulsion Integrals Using the Rys Quadrature Method*. J. Comput. Chem. 1983.

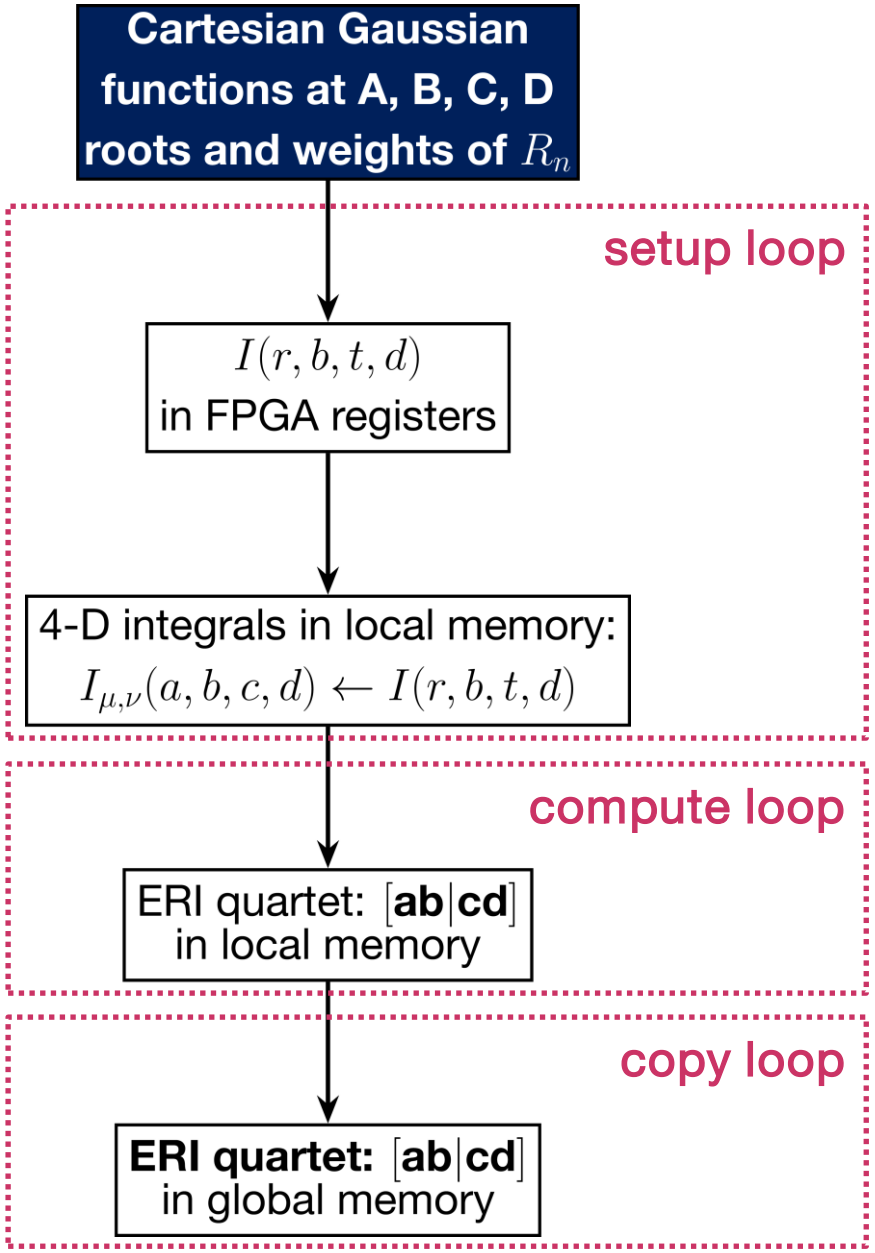
FPGA Design and Implementations

Target FPGA Device

- Intel Stratix 10 GX 2800
  - 5760 DSP blocks
    - 1 single precision FMA/cycle each
  - 11721 M20K RAM blocks (20 Kb each)
    - 229 Mbits
  - 933120 ALMs: control, addresses, all non-FP arithmetic
    - 4 registers per ALM
    - > 3.7 million registers: form pipeline stages
  - 23796 MLAB (640 bits each)
    - each is configured with 10 ALMs
    - 15 Mbits
- BittWare 520N card
  - PCIe Gen3 x8 (x16)
  - 4x DDR4 channels (8 GB each, 32 GB in total)
- Intel FPGA Add-on for oneAPI Base Toolkit

oneAPI

FPGA Design



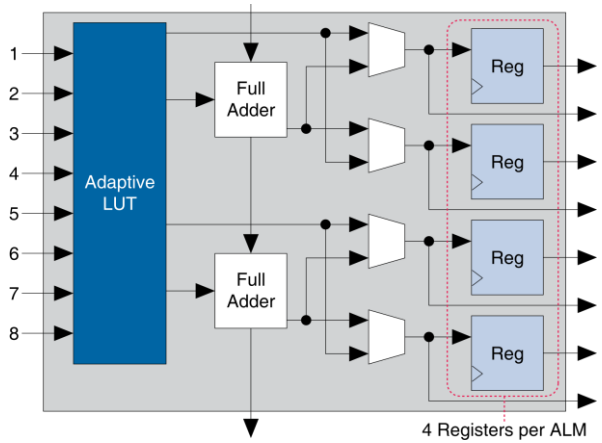
Keys to good performance:

- FPGA on-chip memories
- fully unrolled loop structures for recurrence relations
- optimized stores for global memory

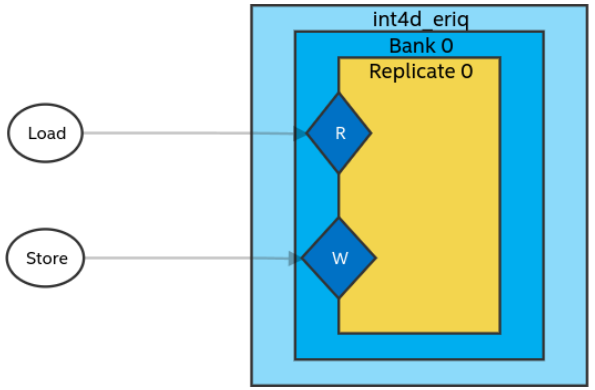
Setup Loop

Setup loop: calculation of 4-D integrals

- $I(r, b, t, d)$
  - < 3 KB, use FPGA registers in ALMs
  - All loops are fully unrolled for all 2-index recurrence relations.
  - Compiler generates deeply pipelined hardware datapath.

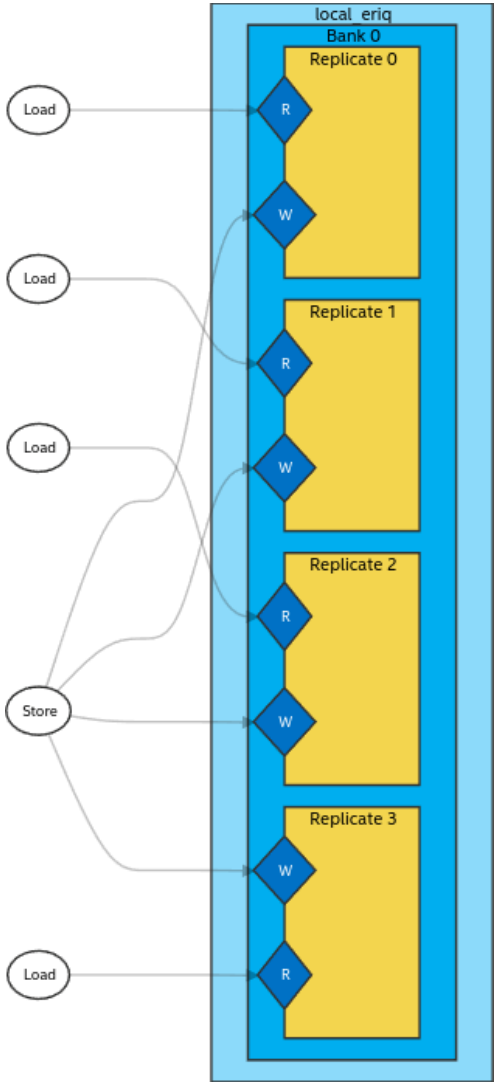


- $I_{\mu,\nu}(a, b, c, d)$
  - 6-D array with custom local memory layout
  - parallel data accesses via memory banks
  - only 1 replicate for minimized hardware resources



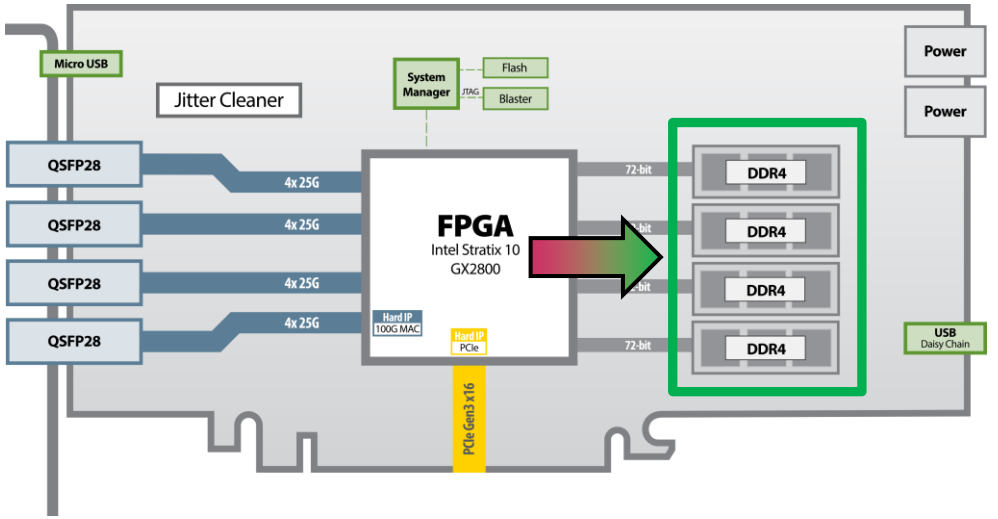
Compute loop: calculation of ERI quartet

- $[ab|cd]$
  - 2-D array with custom memory layout on FPGA is designed for this 4-D array in math.
    - $[ab|$: number of memory banks
    - $|cd]$: bank depth
  - the actual implementation of on-chip memories is determined by compiler
    - can be BRAM or MLAB
      (a few replicates may be necessary)
    - or FPGA register for small ERI quartet



Copy Loop

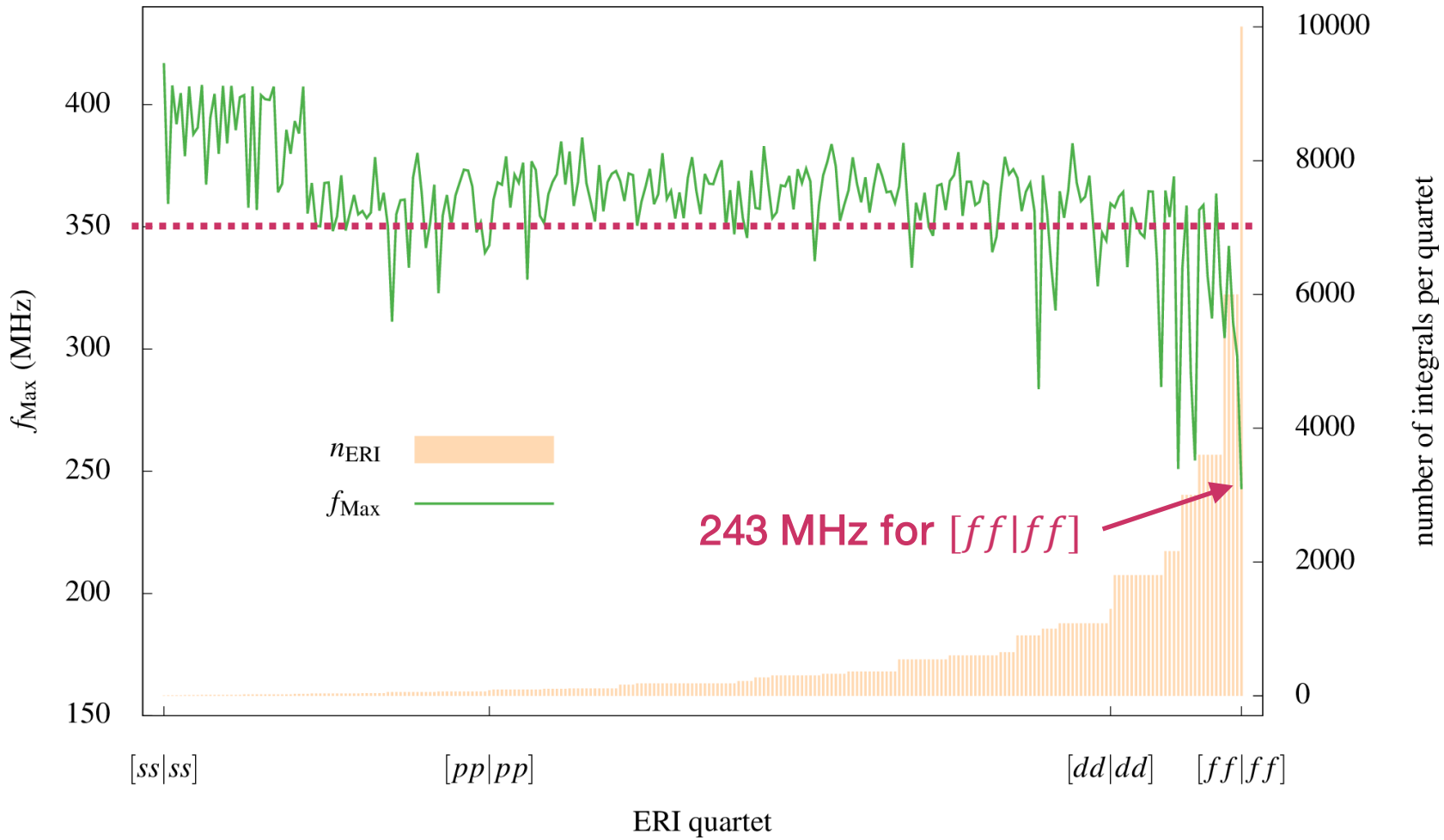
- Intel Stratix 10 GX2800 global memory
  - 4x channels
  - data width per DDR channel:
    512 bits = 64 bytes = 16 FP32
- Compute loop
  - produce $n_c \times n_d$ integers / cycle
  - however $n_c \times n_d$ may not be multiple of 16
- Copy loop
  - copy the generated $[ab|cd]$ from on-chip memories to FPGA global memory
  - all loads and stores are parallel



Results and Discussion

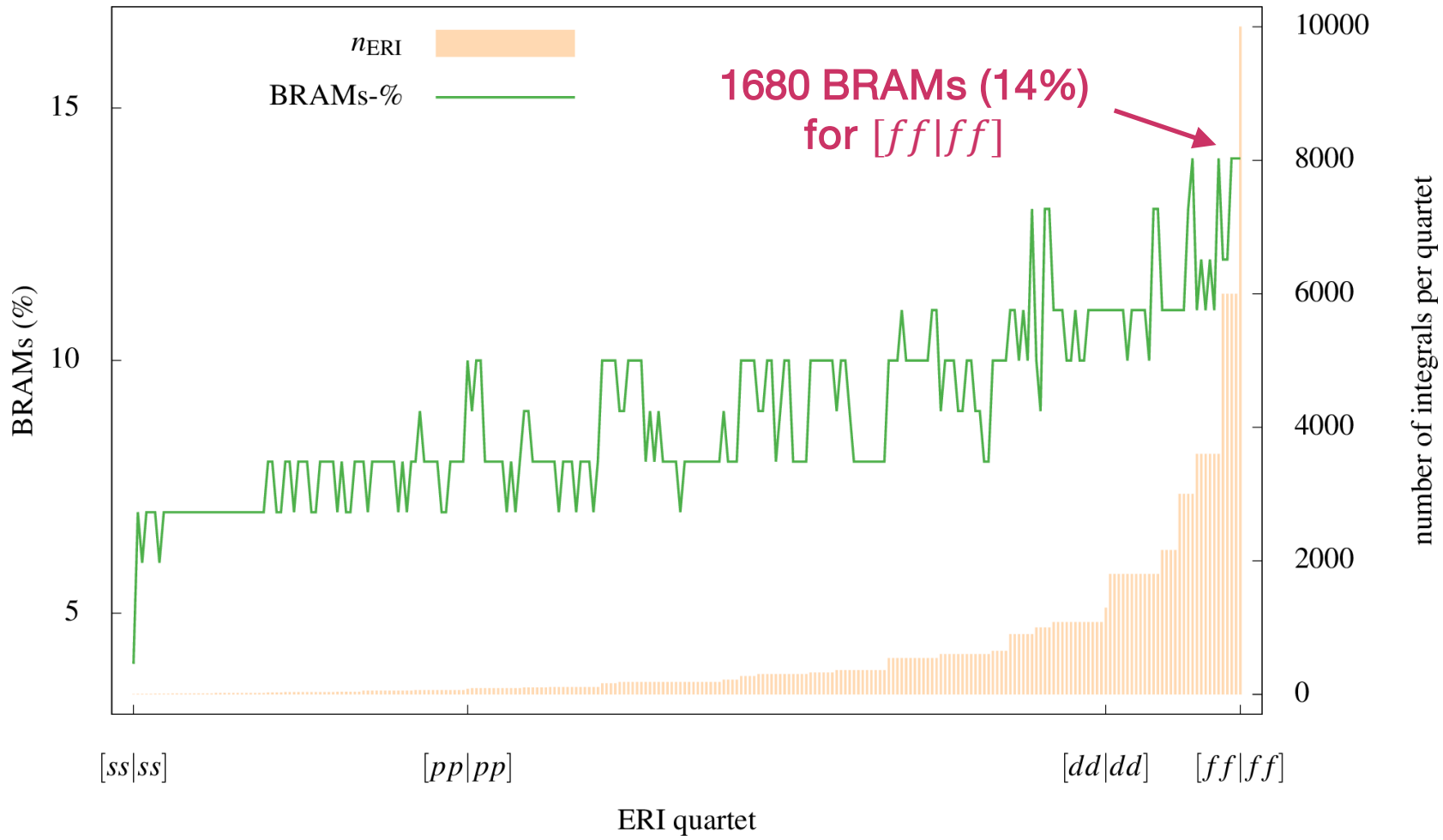
Resource Consumptions: f_{Max}

- $[ss|ss]$ to $[ff|ff]$: 256 kernel variants with DPC++ function template



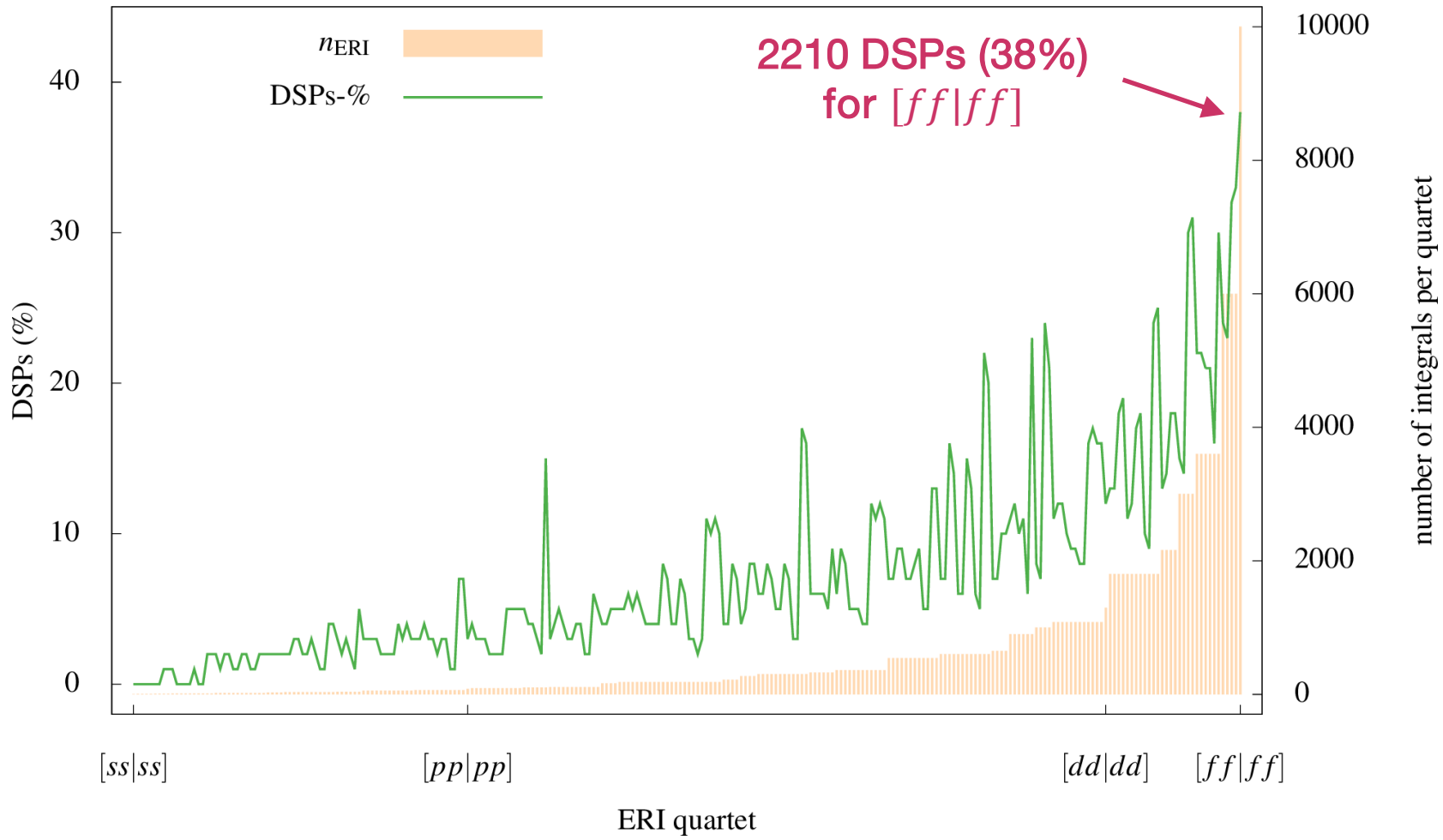
Resource Consumptions: BRAMs

- $[ss|ss]$ to $[ff|ff]$: 256 kernel variants with DPC++ function template



Resource Consumptions: DSPs

- $[ss|ss]$ to $[ff|ff]$: 256 kernel variants with DPC++ function template



Performance Model

- **FPGA performance model**

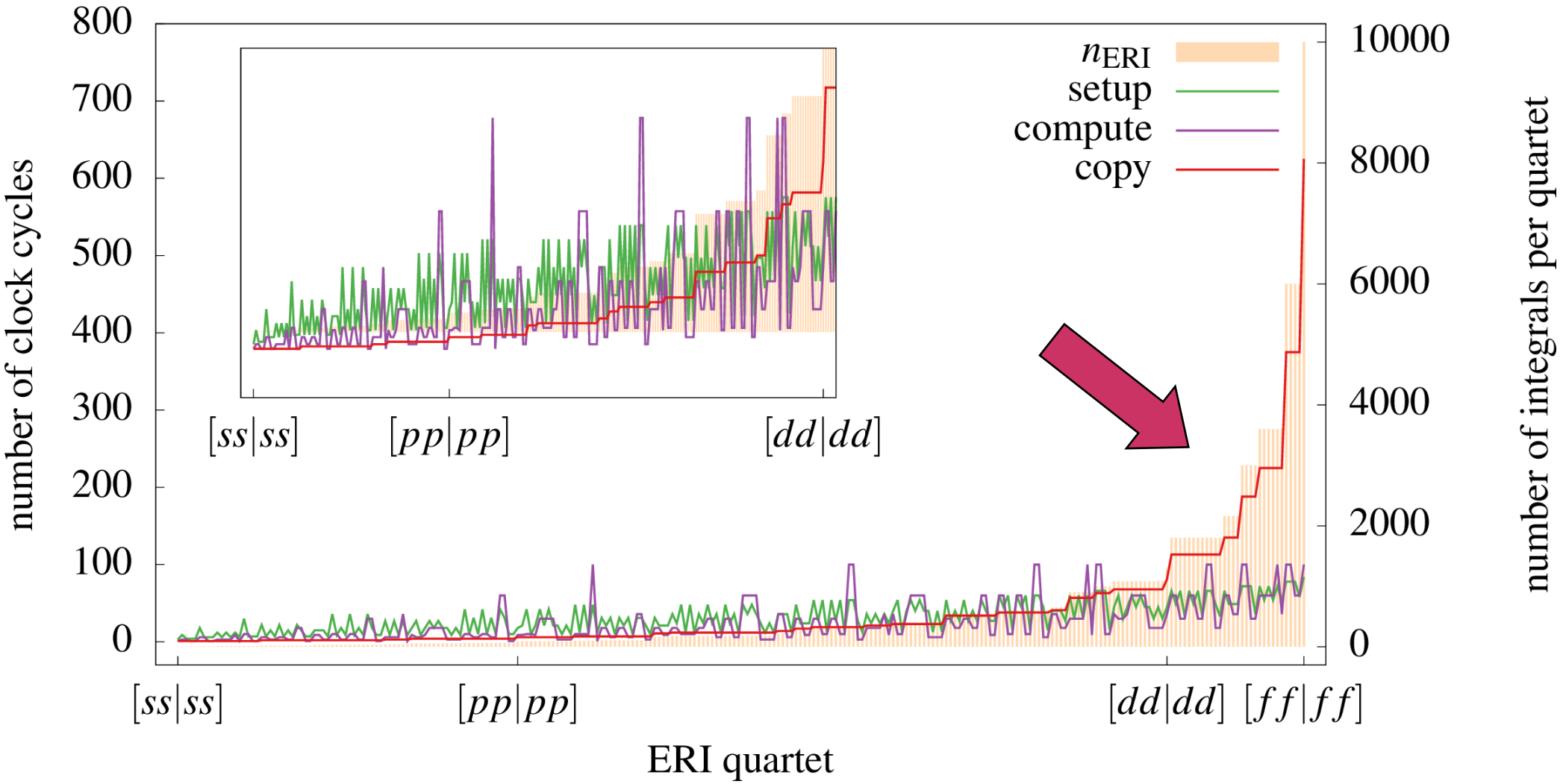
loop	number of clock cycles
setup	$3n_{\mathrm{Rys}}(d+1)$
compute	$n_c \times n_d$
copy	$\left\lceil \frac{n_{\mathrm{ERI}}}{16} \right\rceil$

- **CPI: Cycles Per Integral**

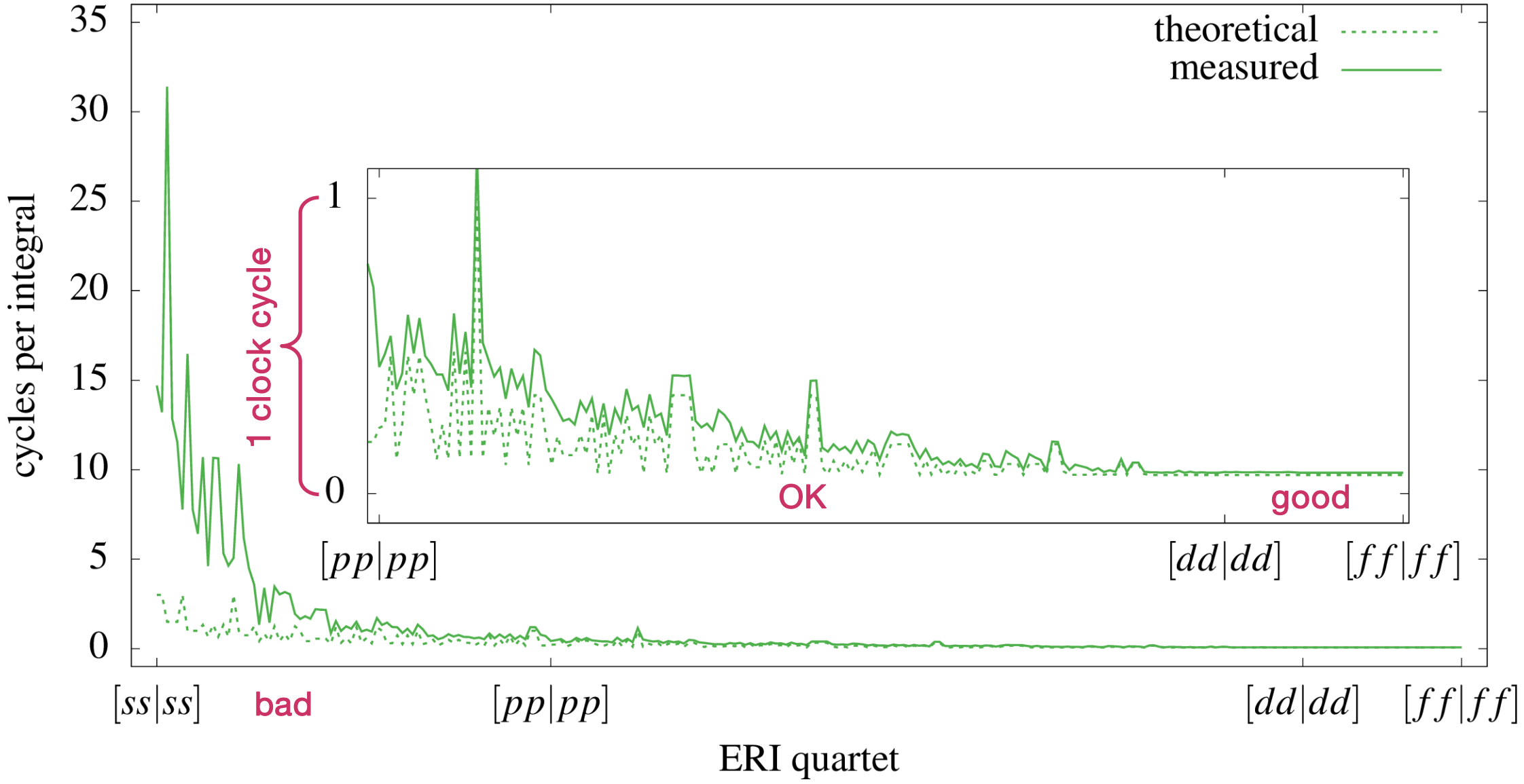
$$\mathrm{CPI} = \frac{\max(\mathrm{setup}, \mathrm{compute}, \mathrm{copy})}{n_{\mathrm{ERI}}}$$

Clock Cycles for the Loops

- The copy loop dominates for large ERI quartets.



CPI: Cycles Per Integral



Performance Analysis

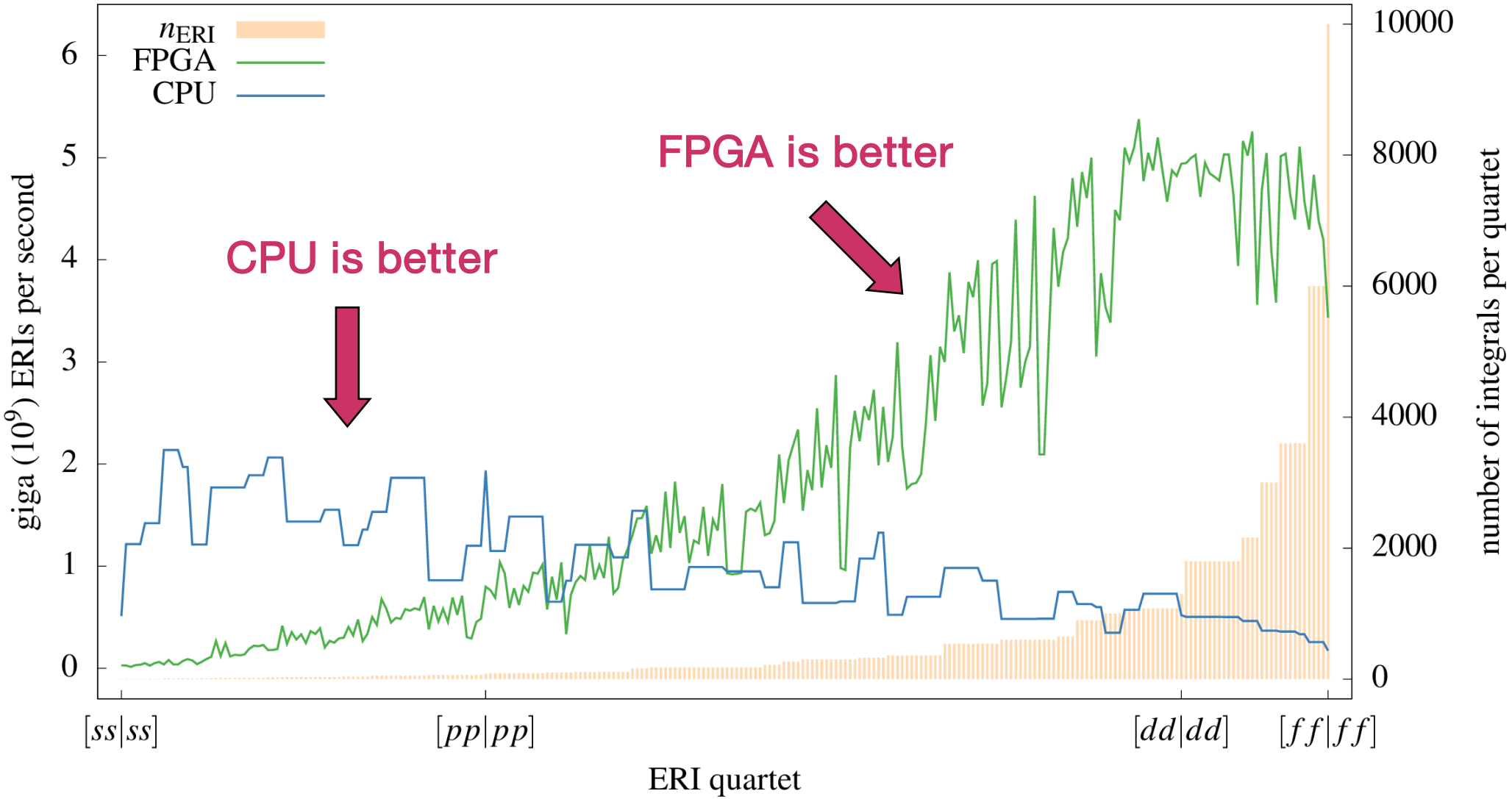
- GERIS: Giga (10^9) ERIs per Second

$$\mathrm{GERIS} = \frac{f_{\mathrm{Max}}}{\mathrm{CPI}}$$

- FPGA: 1x Intel Stratix 10 GX 2800

- CPU: 2x Intel Xeon Gold Skylake 6148
  - 40 CPU cores per node
  - libint: version 2.6.0
    - built with EasyBuild foss-2021a toolchain
    - only support double-precision floating-point arithmetic

Performance Analysis: GERIS



Conclusion and Future Work

Conclusion and Future Work

- Development of the computation of electron repulsion integrals on FPGA using oneAPI
  - DPC++ function template for FPGA kernels
  - custom FPGA local memory layouts
  - optimized stores for global memory

oneAPI

- Performance analysis
  - FPGA is good for large ERI quartets
  - CPI: good agreement with performance model
  - GERIS: 1 FPGA faster than 2x CPUs (40 cores)

- Future work
  - compression of ERIs on FPGAs
  - integration in CP2K