NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Open-Source Architecture Code Analyzer

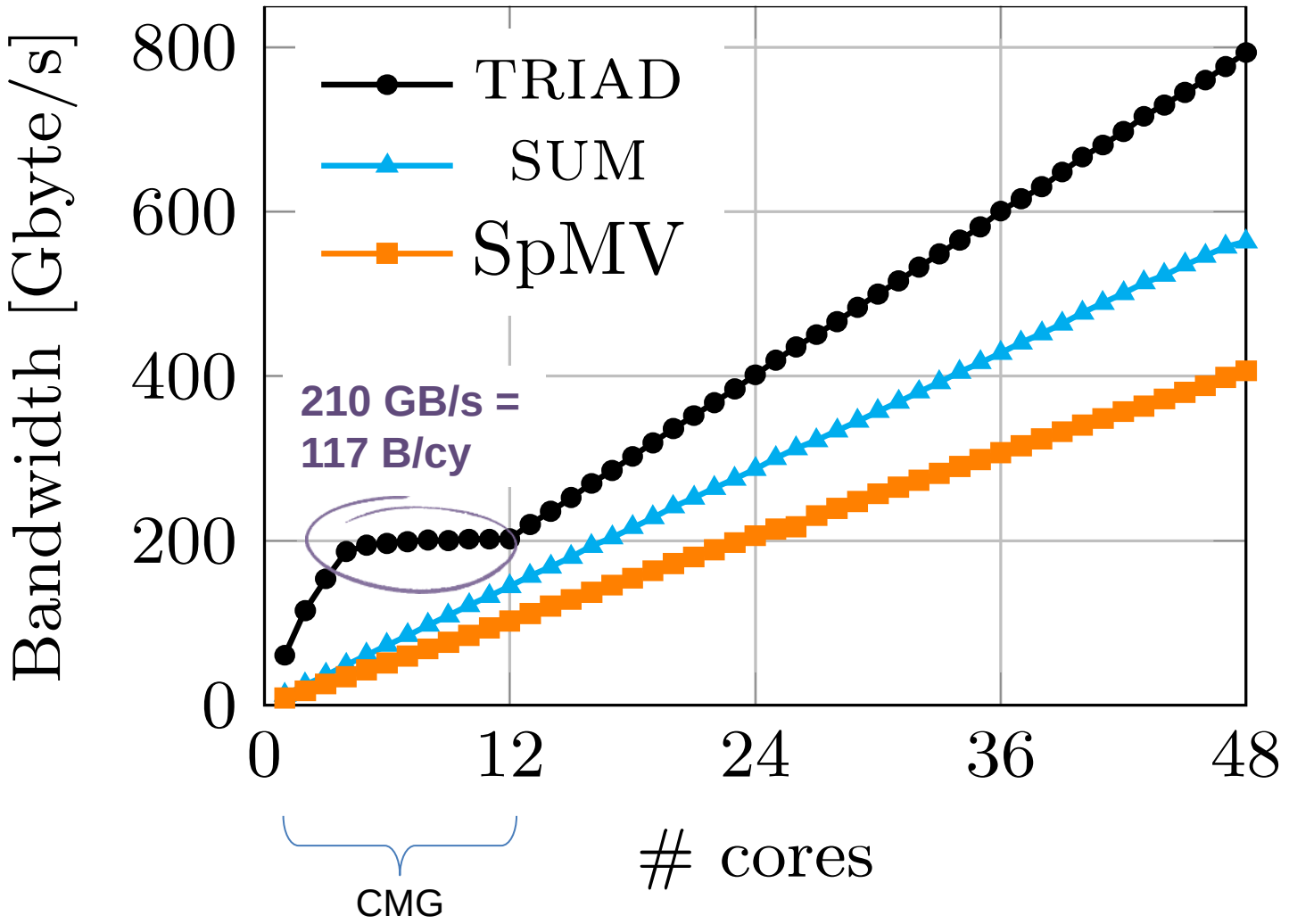
Jan Laukemann

Erlangen National High Performance Computing Center (NHR@FAU)

Friedrich-Alexander-University Erlangen-Nürnberg

Scalable Tools Workshop 2023, Lake Tahoe

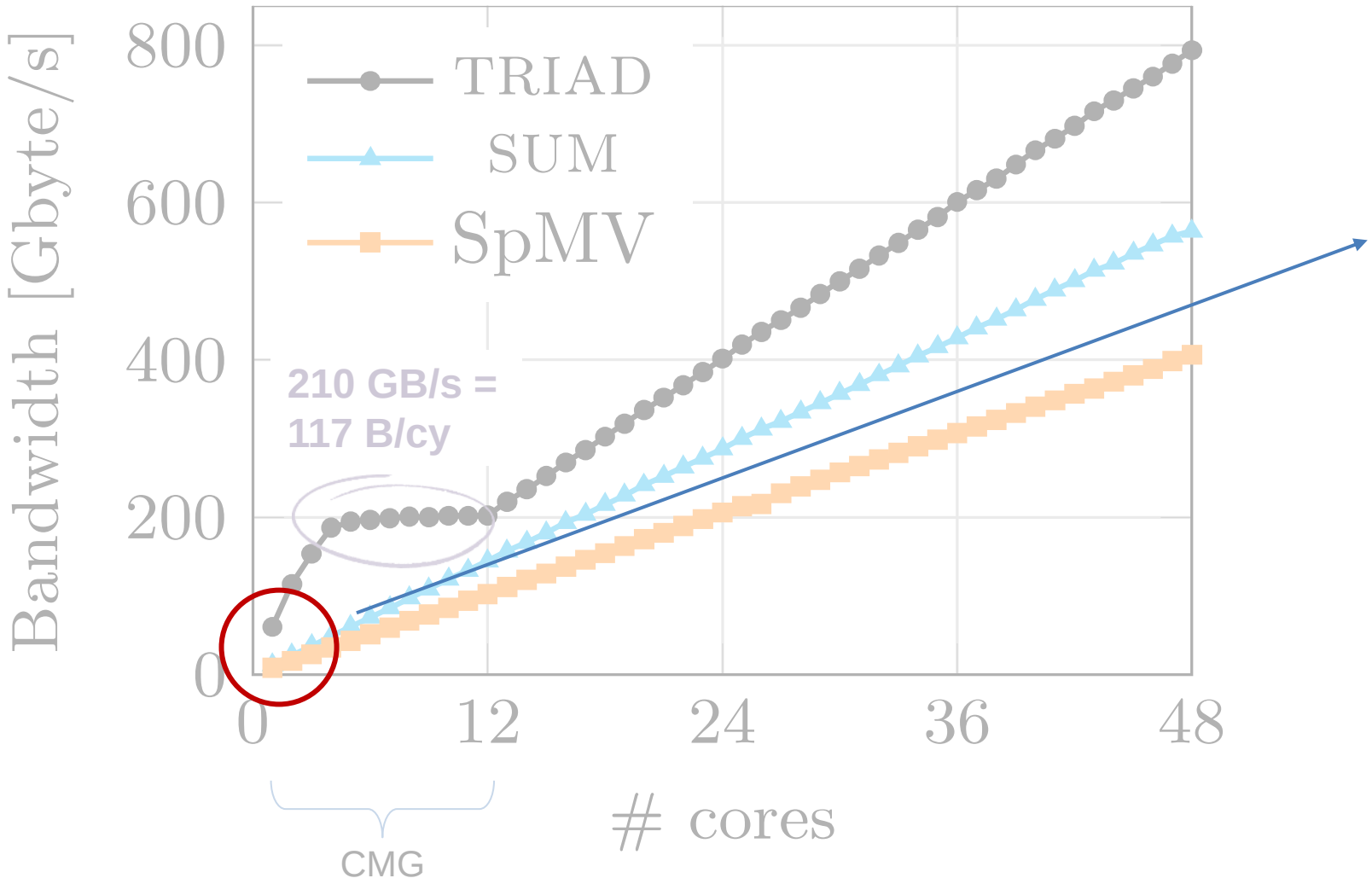
Motivation

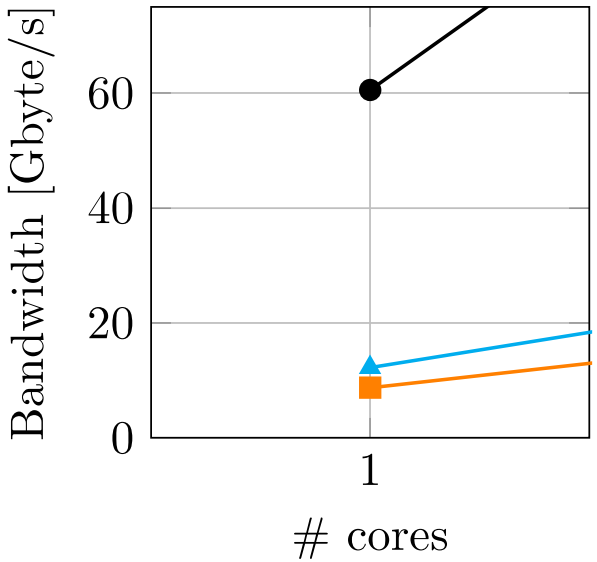


Thread pinning : Compact

C. Alappat, N. Meyer, J. Laukemann, T. Gruber, G. Hager, G. Wellein, and T. Wettig: *ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX.* Concurrency and Computation: Practice and Experience, e6512 (2021). DOI: 10.1002/cpe.6512

Motivation

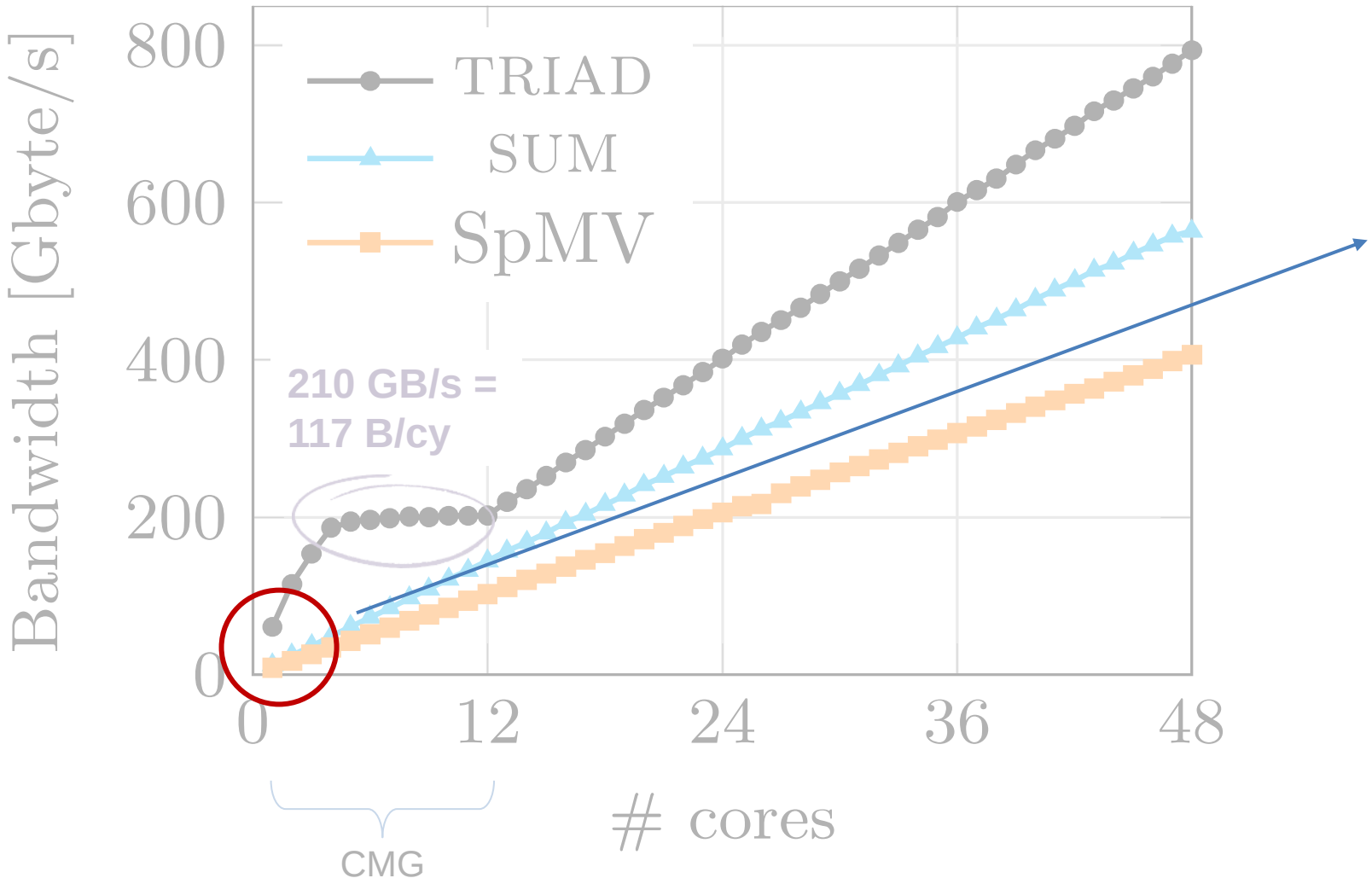




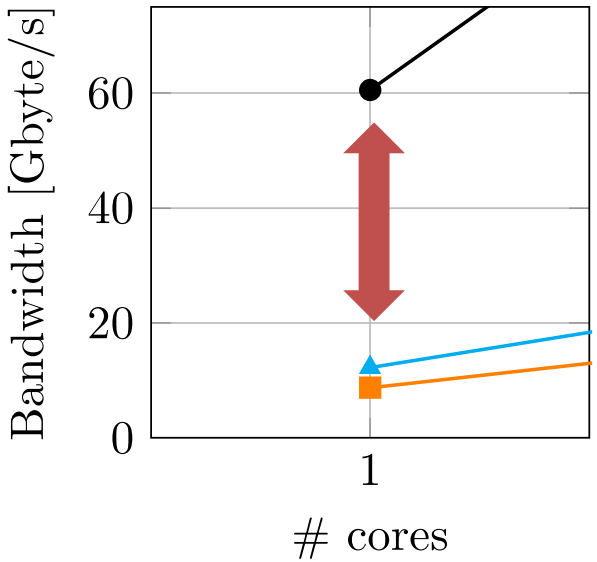
Thread pinning : Compact

C. Alappat, N. Meyer, J. Laukemann, T. Gruber, G. Hager, G. Wellein, and T. Wettig: *ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX.* Concurrency and Computation: Practice and Experience, e6512 (2021). DOI: 10.1002/cpe.6512

Motivation



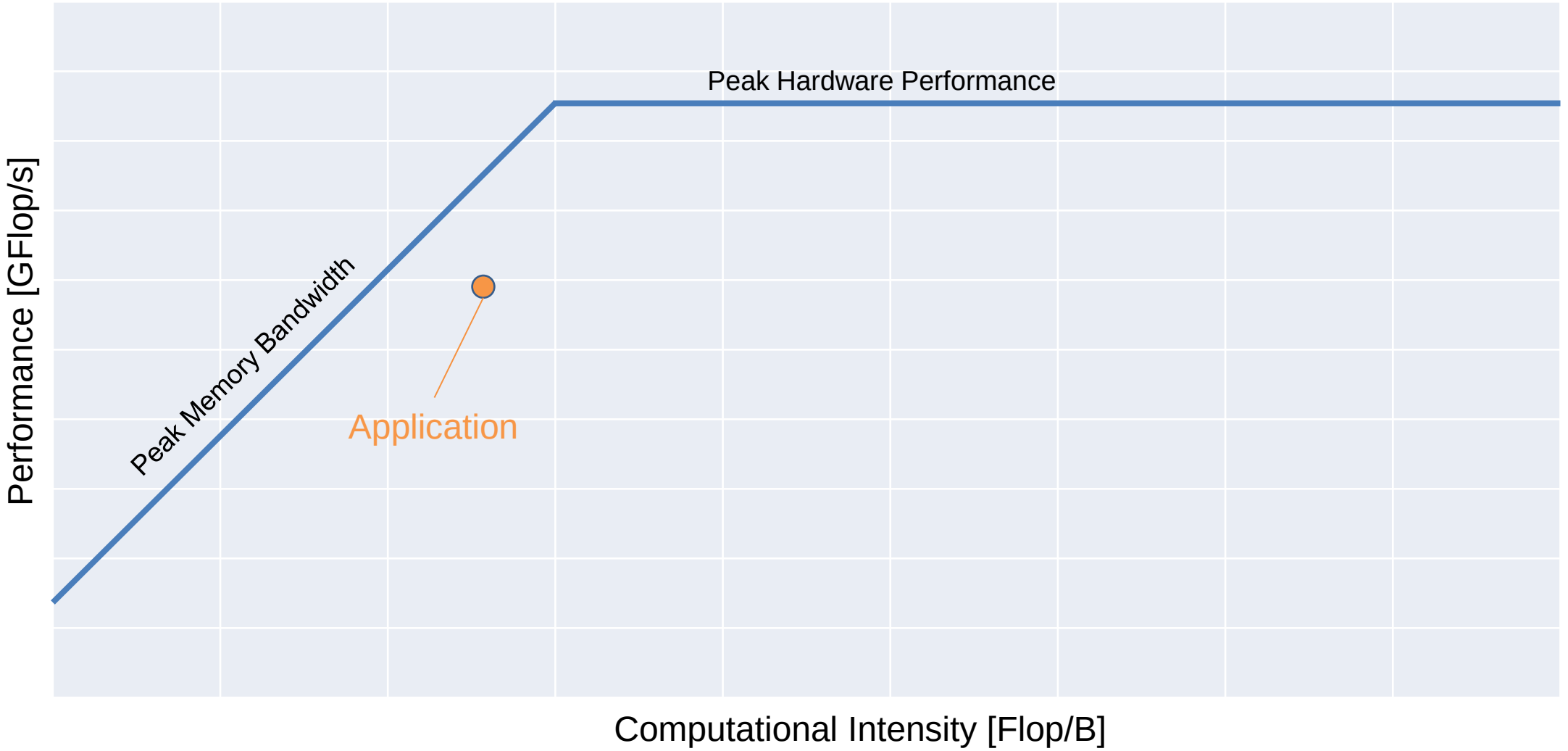
Thread pinning : Compact



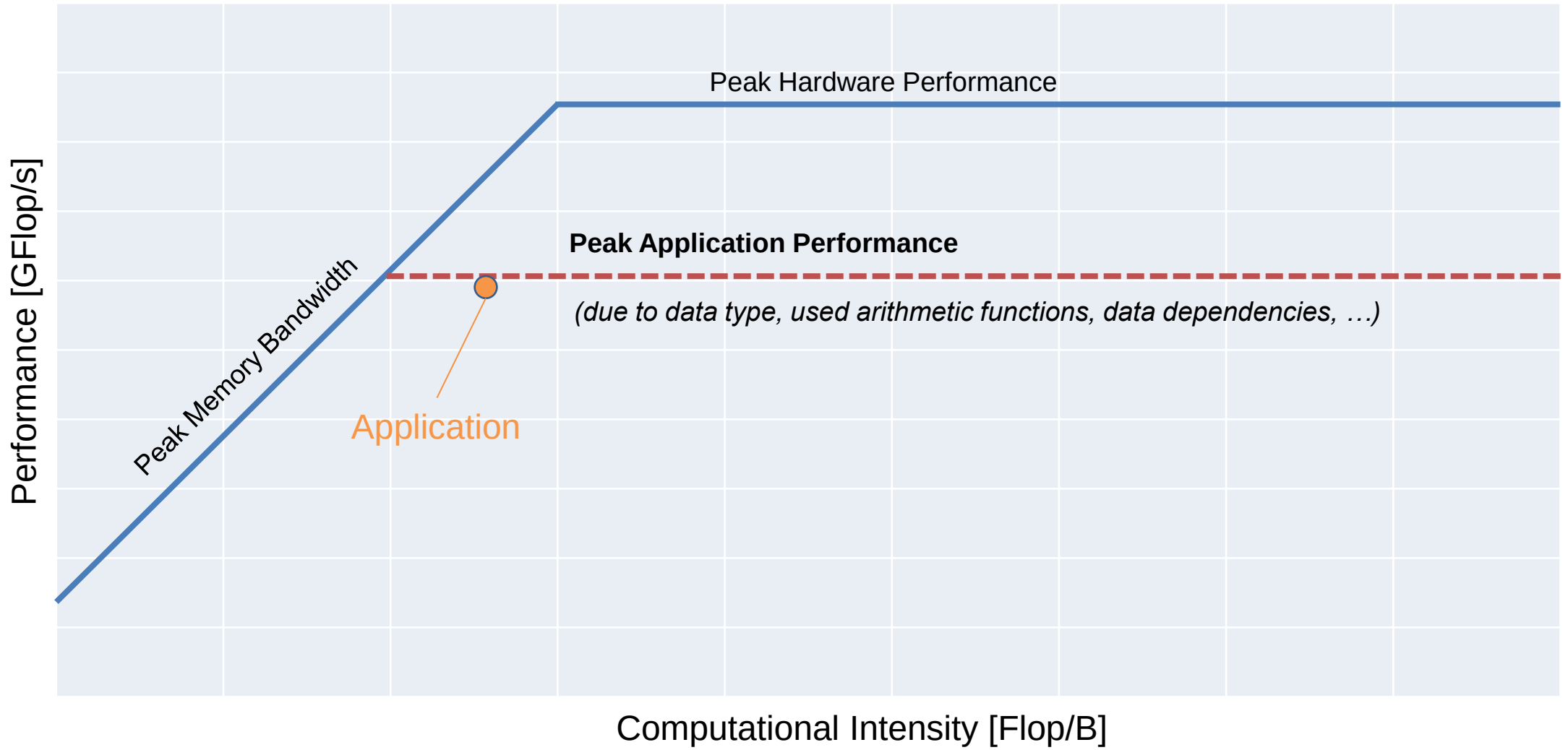
Understanding single-core performance is the key!

C. Alappat, N. Meyer, J. Laukemann, T. Gruber, G. Hager, G. Wellein, and T. Wettig: *ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX.* Concurrency and Computation: Practice and Experience, e6512 (2021). DOI: 10.1002/cpe.6512

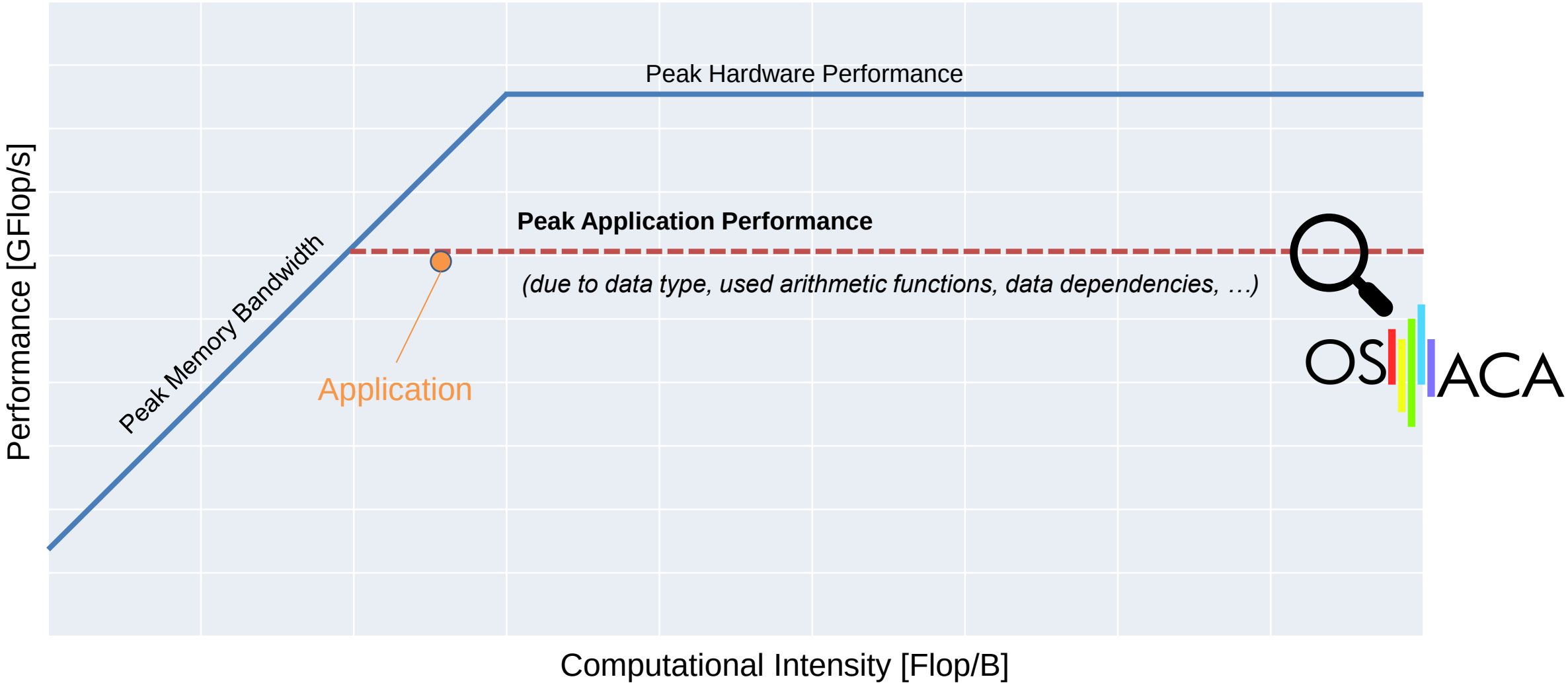
Motivation


Peak Hardware Performance
Peak Memory Bandwidth
Application
Performance [GFlop/s]
Computational Intensity [Flop/B]

Motivation

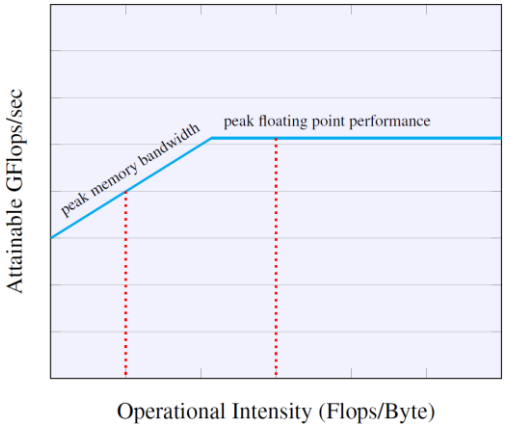

Peak Hardware Performance
Peak Application Performance
(due to data type, used arithmetic functions, data dependencies, …)
Peak Memory Bandwidth
Application
Performance [GFlop/s]
Computational Intensity [Flop/B]

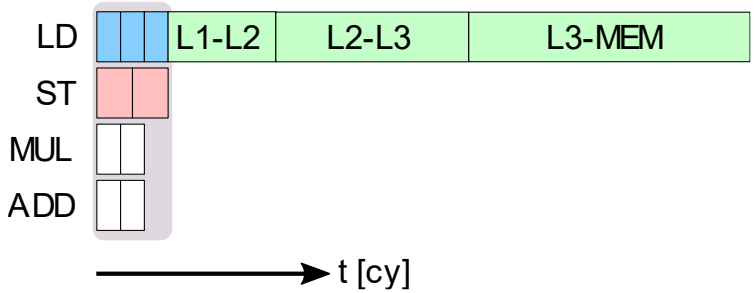
Motivation


Peak Hardware Performance
Peak Application Performance
(due to data type, used arithmetic functions, data dependencies, …)
Peak Memory Bandwidth
Application
Performance [GFlop/s]
Computational Intensity [Flop/B]
OSACA

Analytical Performance Modeling

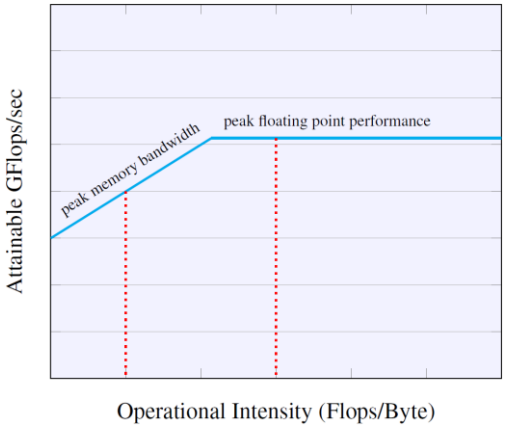
- What is the best performance my code can achieve?
- What are the relevant hardware bottlenecks?
- Apply simplified model of underlying hardware, consisting of
  - In-core execution
  - Data transfer

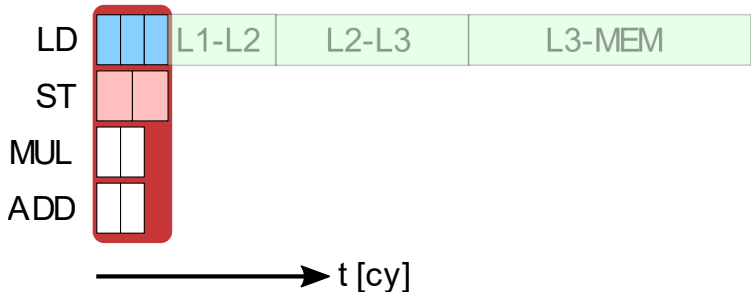




Analytical Performance Modeling

- What is the best performance my code can achieve?
- What are the relevant hardware bottlenecks?
- Apply simplified model of underlying hardware, consisting of
  - **In-core execution**
  - Data transfer





STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

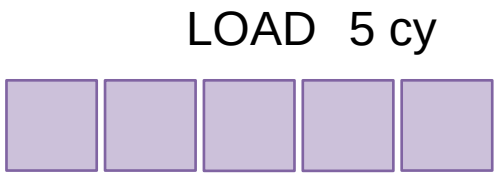
```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```



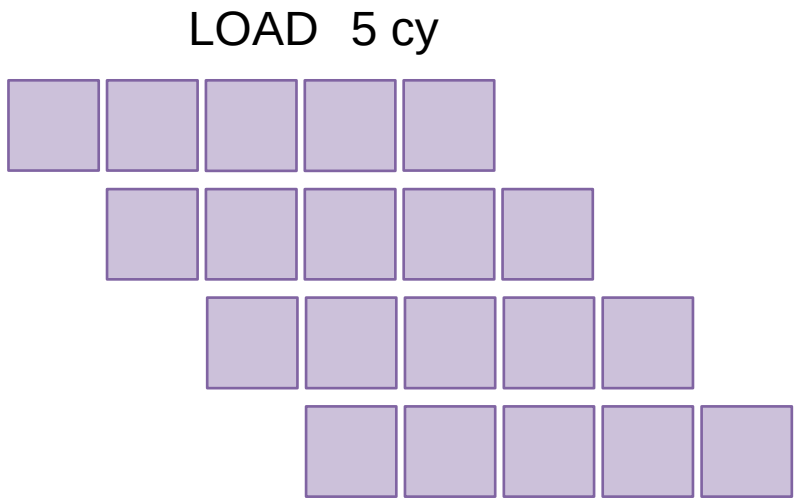

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

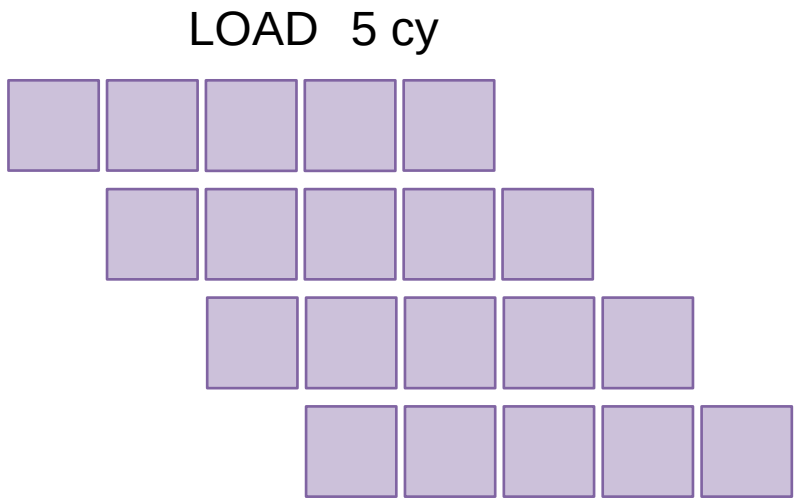


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```



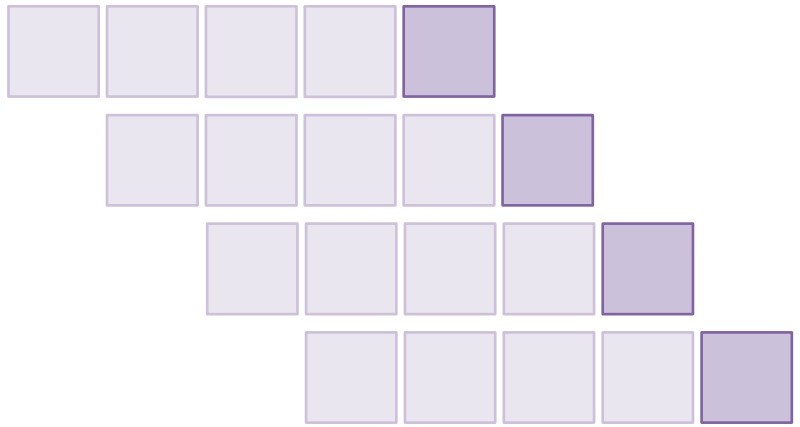
STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

LOAD ~~5 cy~~



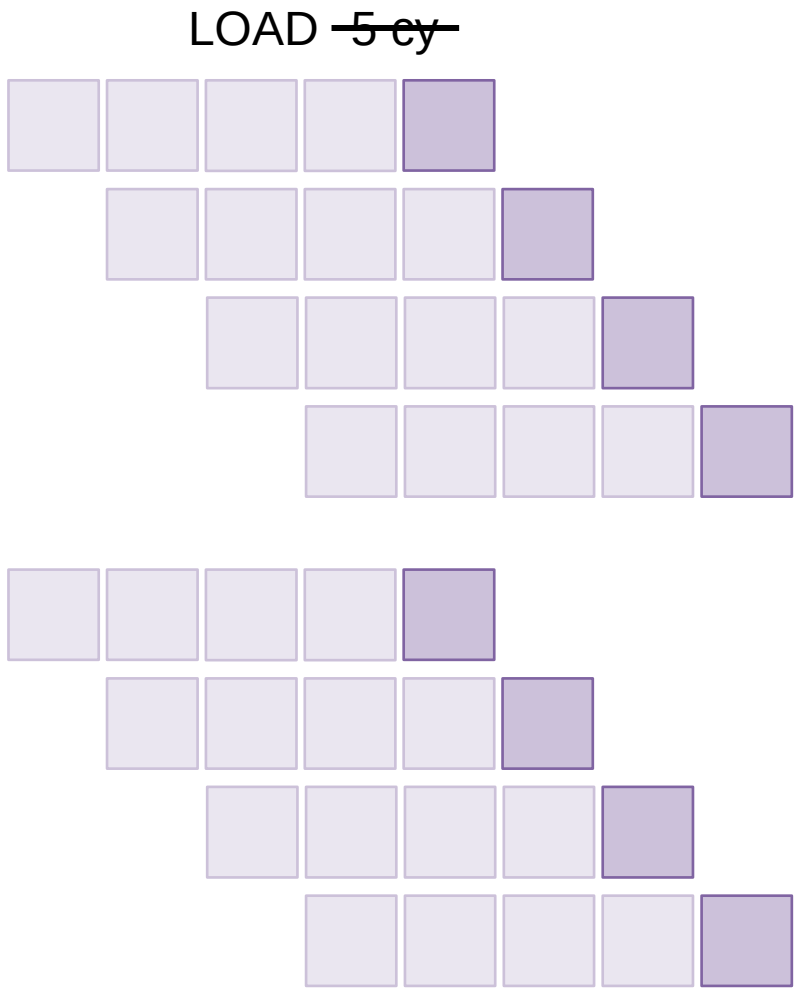
1 cy/LD

STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```



~~1 cy/LD~~

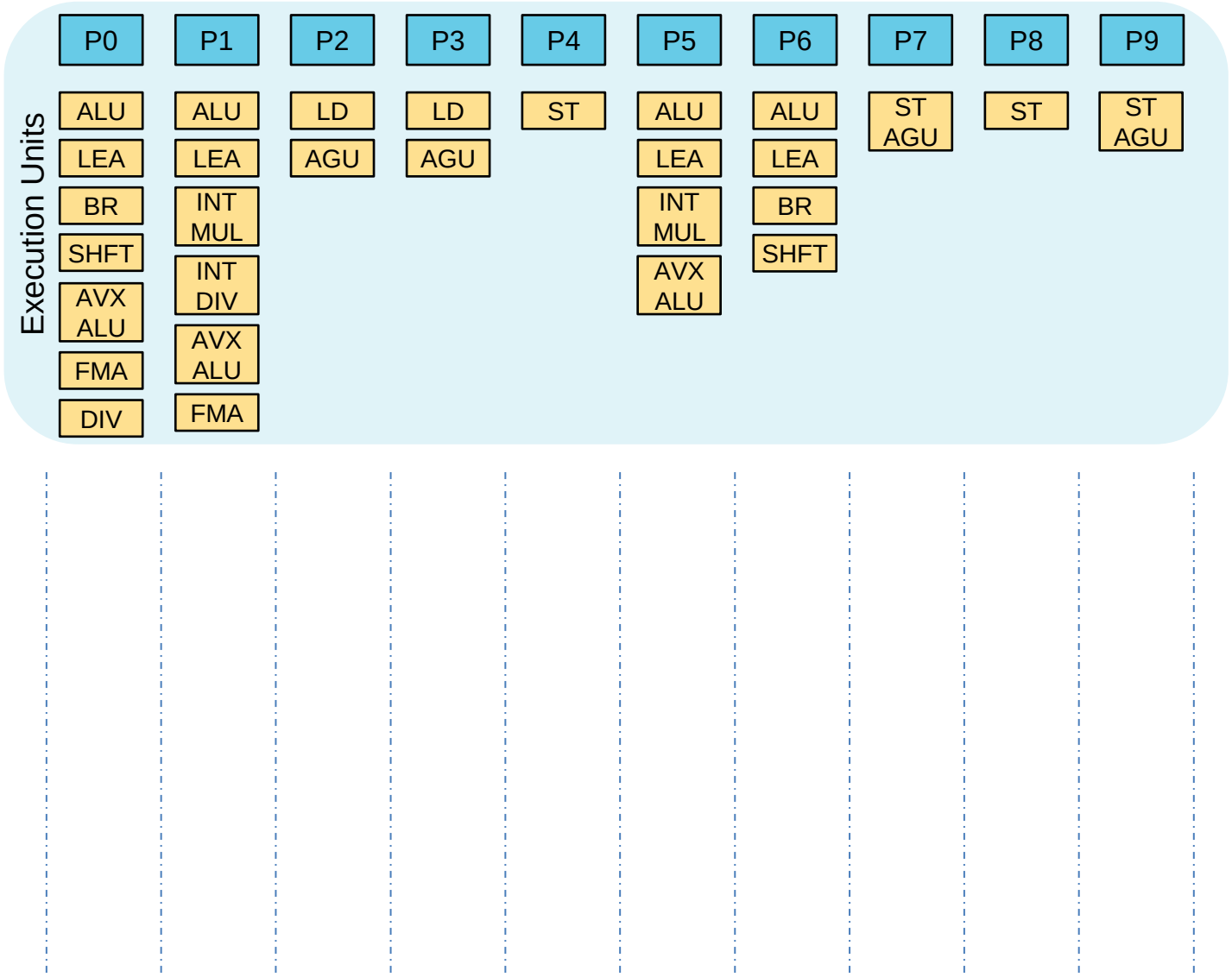
0.5 cy/LD

STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

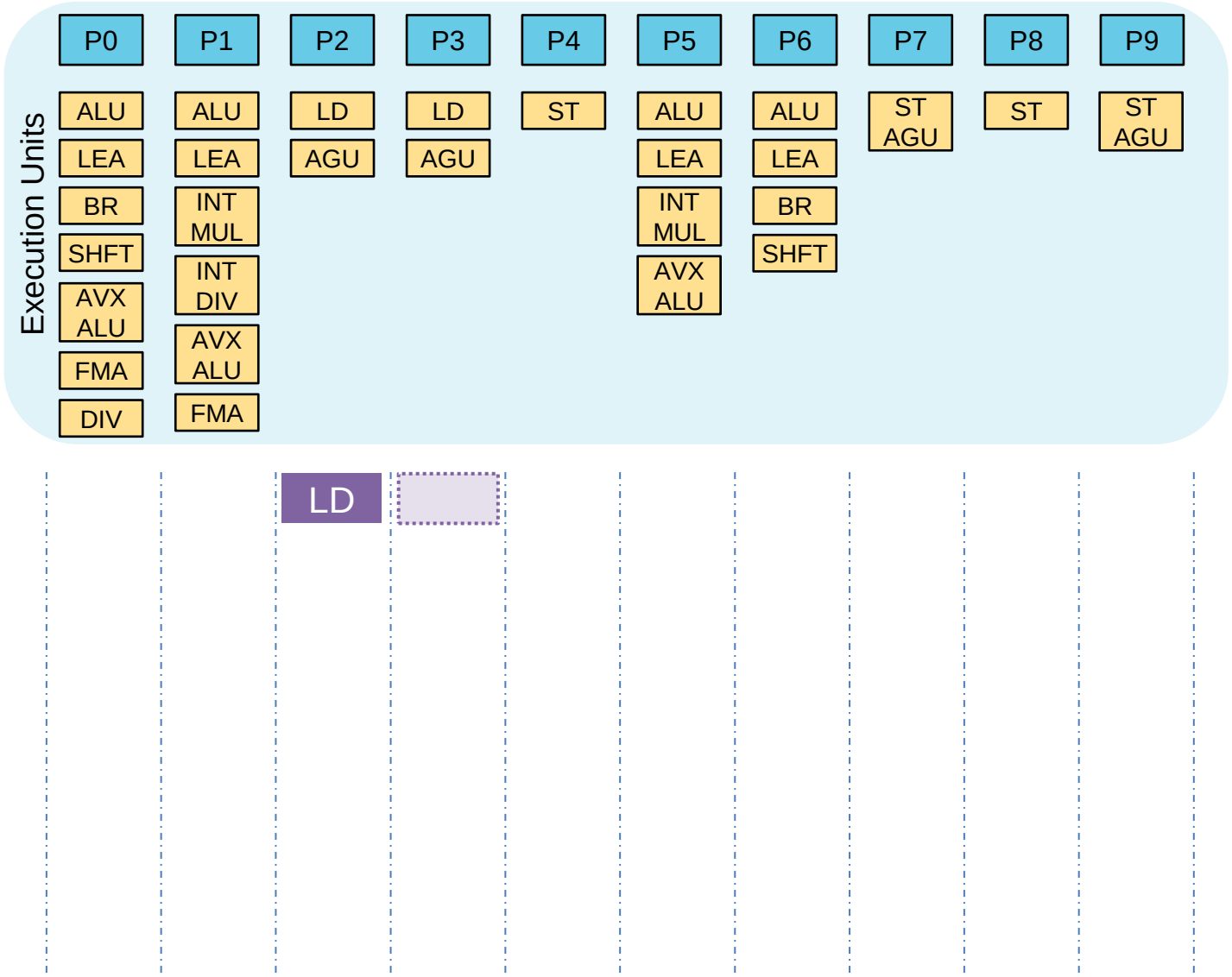


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

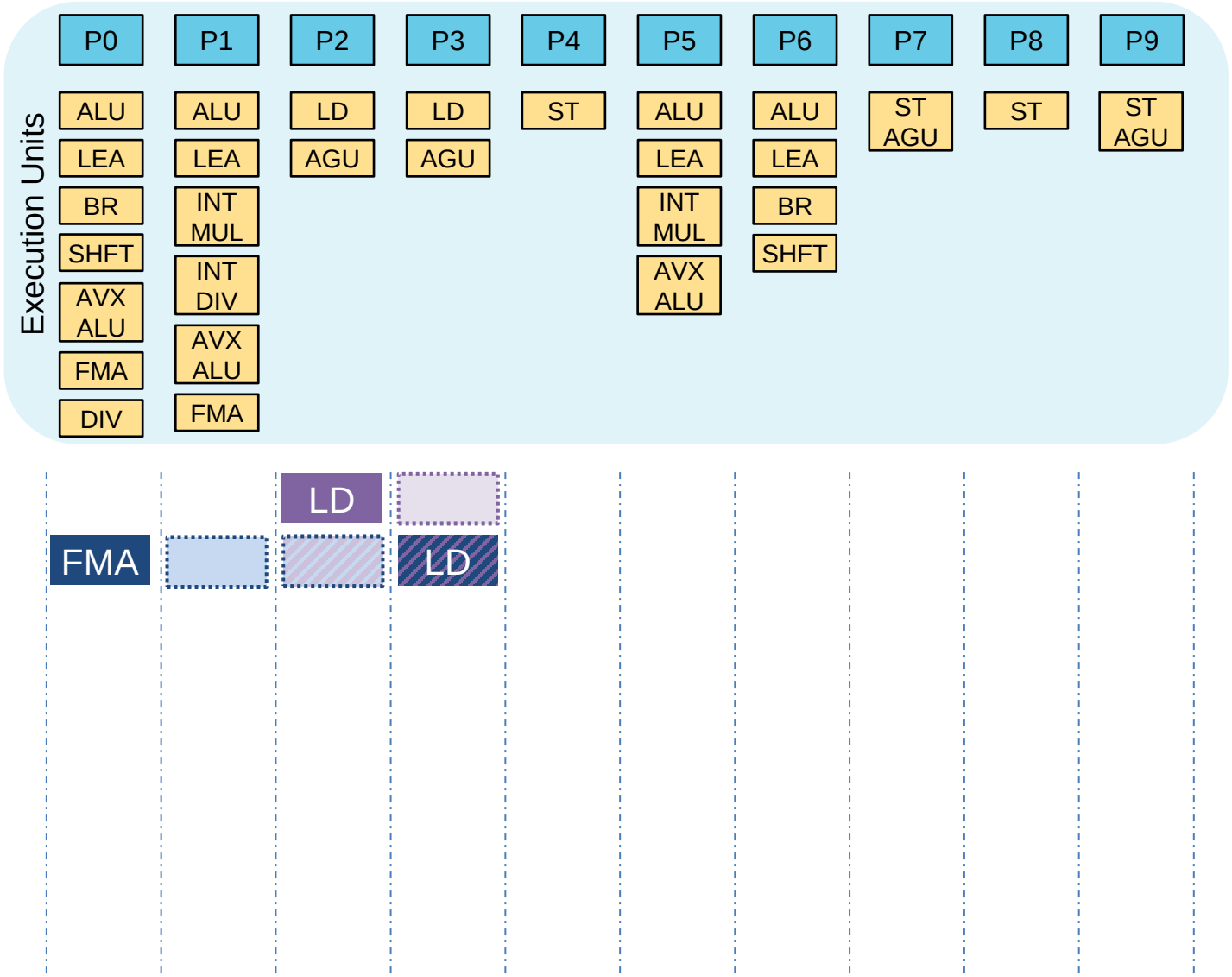


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups       (%r14,%rdx,8), %zmm1
  vfmadd213pd   (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd       %zmm1, (%r12,%rdx,8)
  addq          $8, %rdx
  cmpq          %rsi, %rdx
  jb            ..B2.42
```

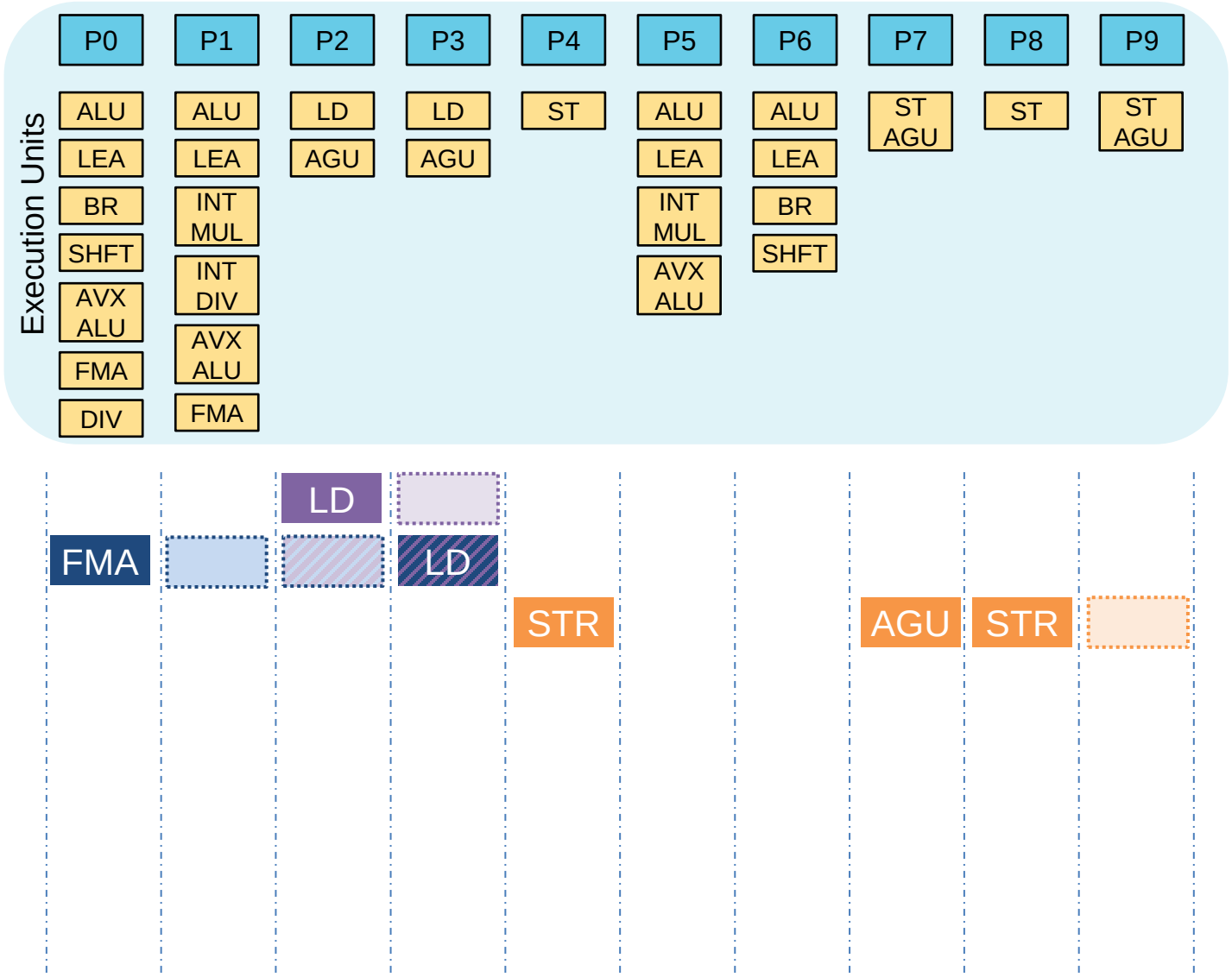


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

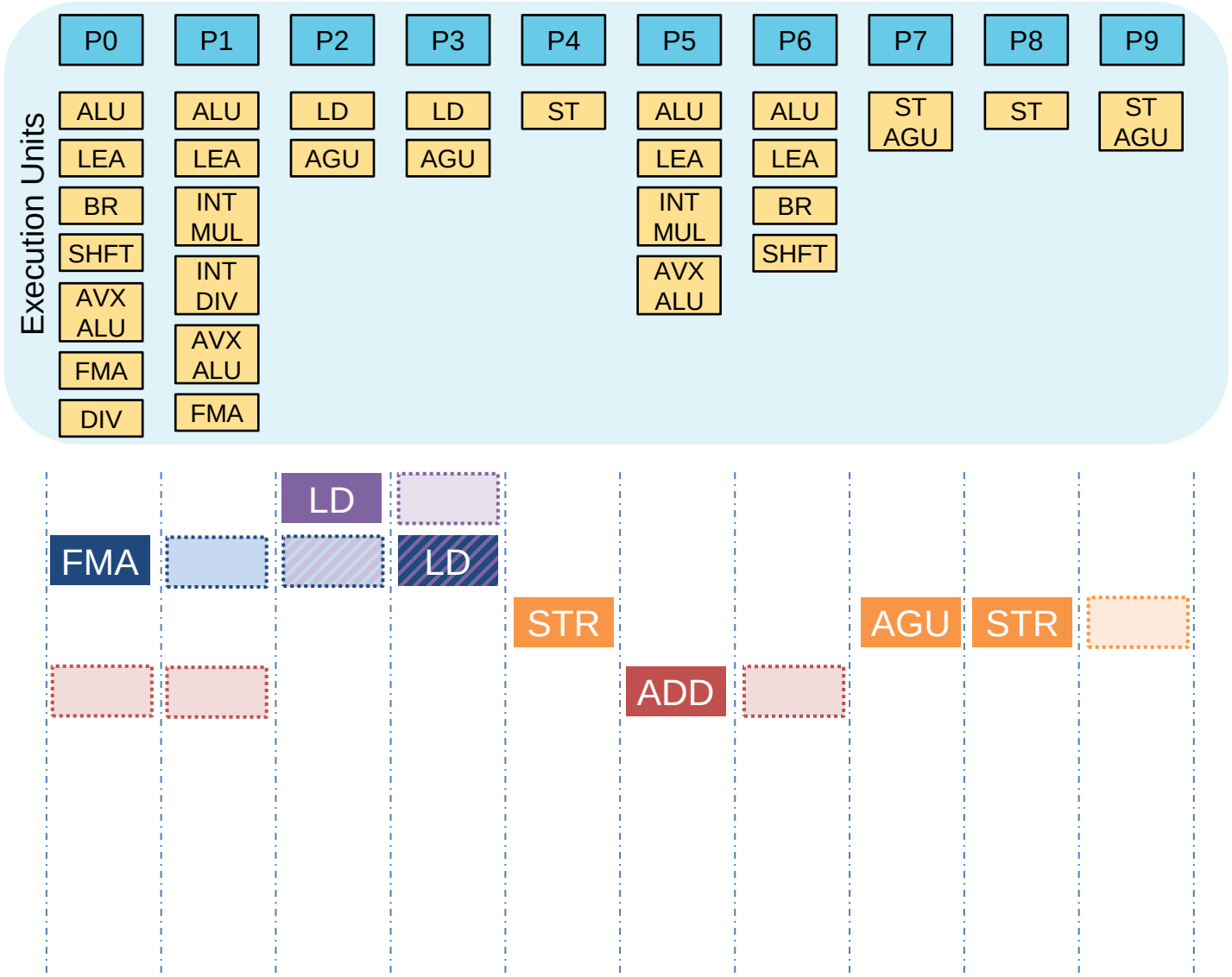


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups       (%r14,%rdx,8), %zmm1
  vfmadd213pd   (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd       %zmm1, (%r12,%rdx,8)
  addq          $8, %rdx
  cmpq          %rsi, %rdx
  jb            ..B2.42
```

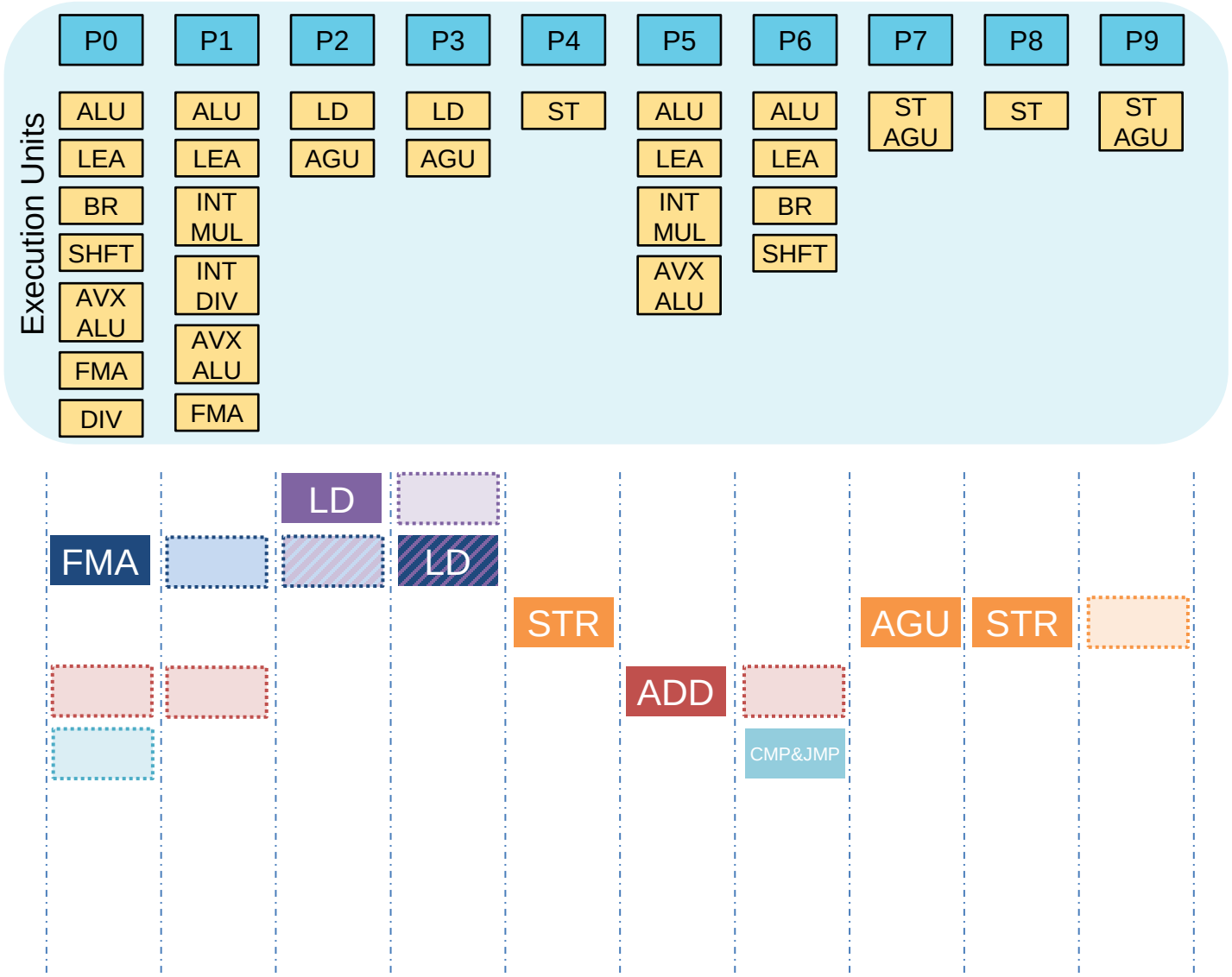


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

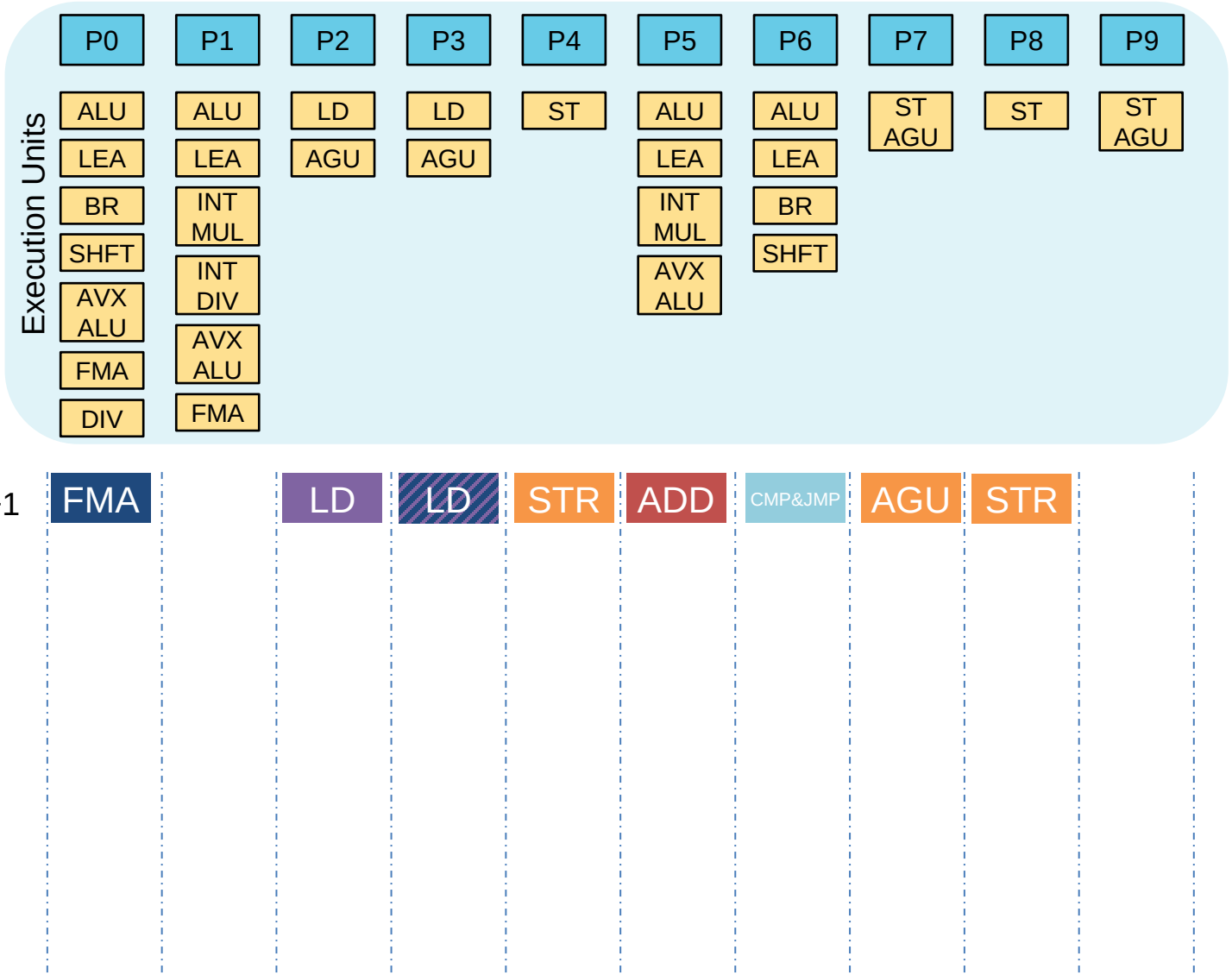


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

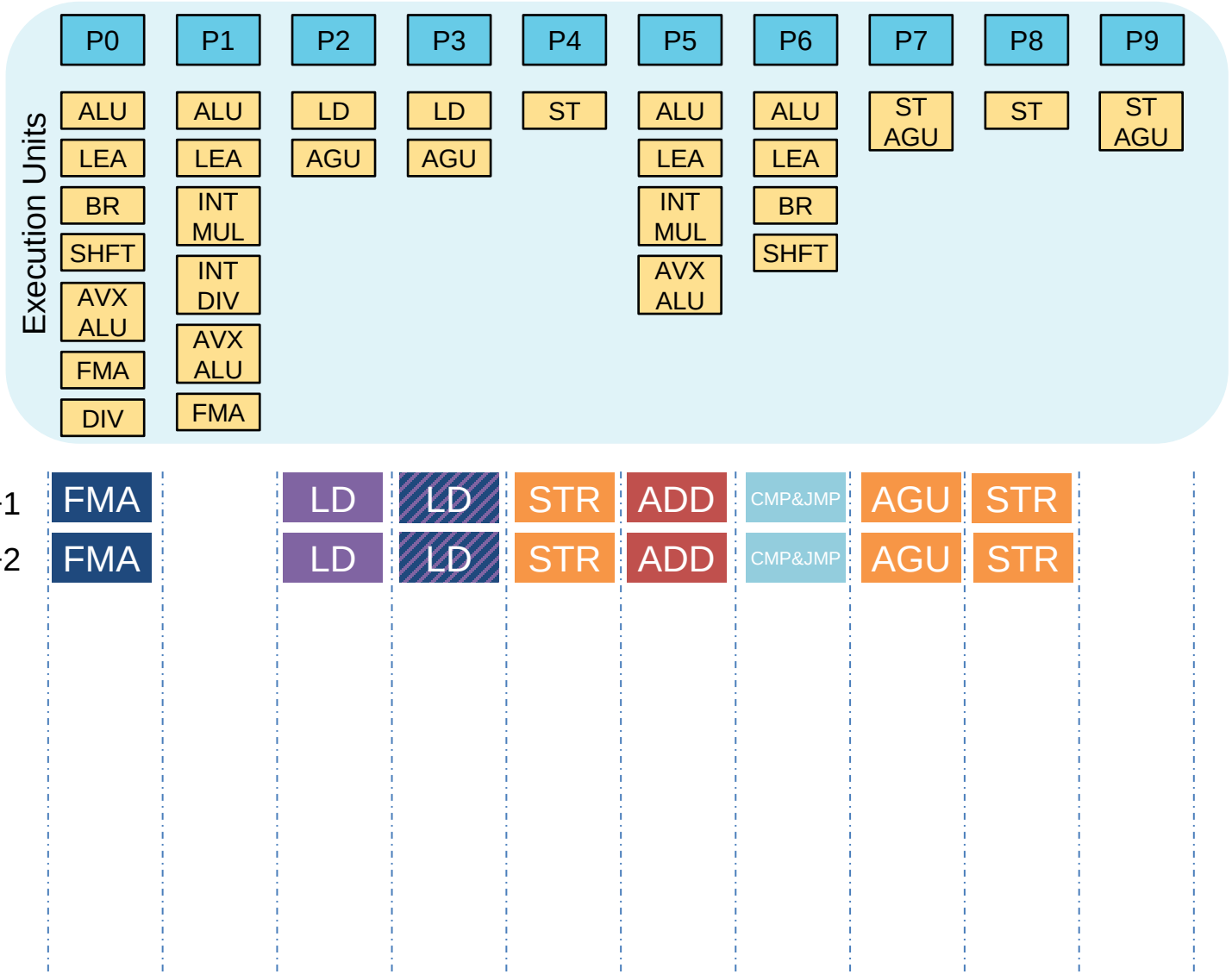


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

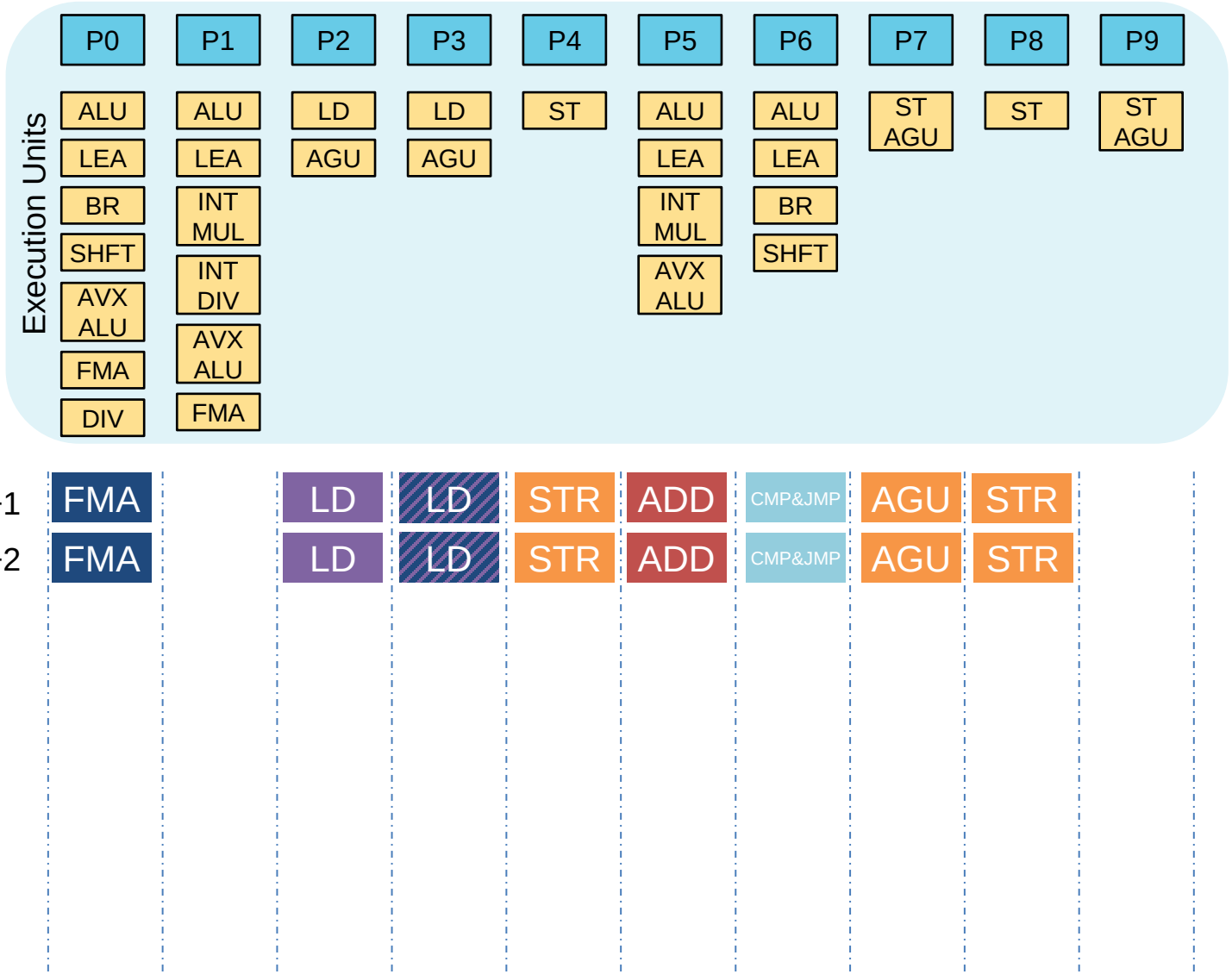


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

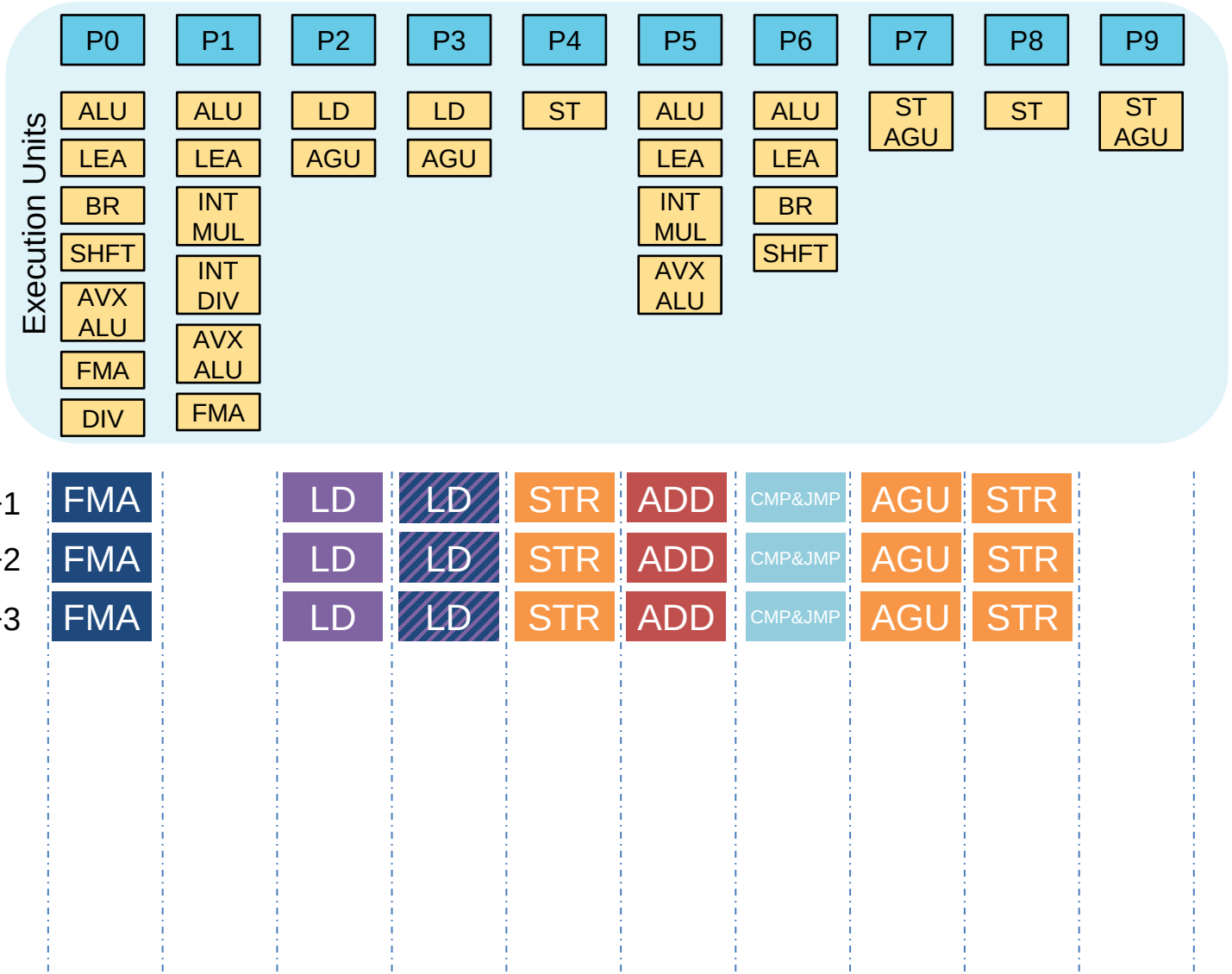


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

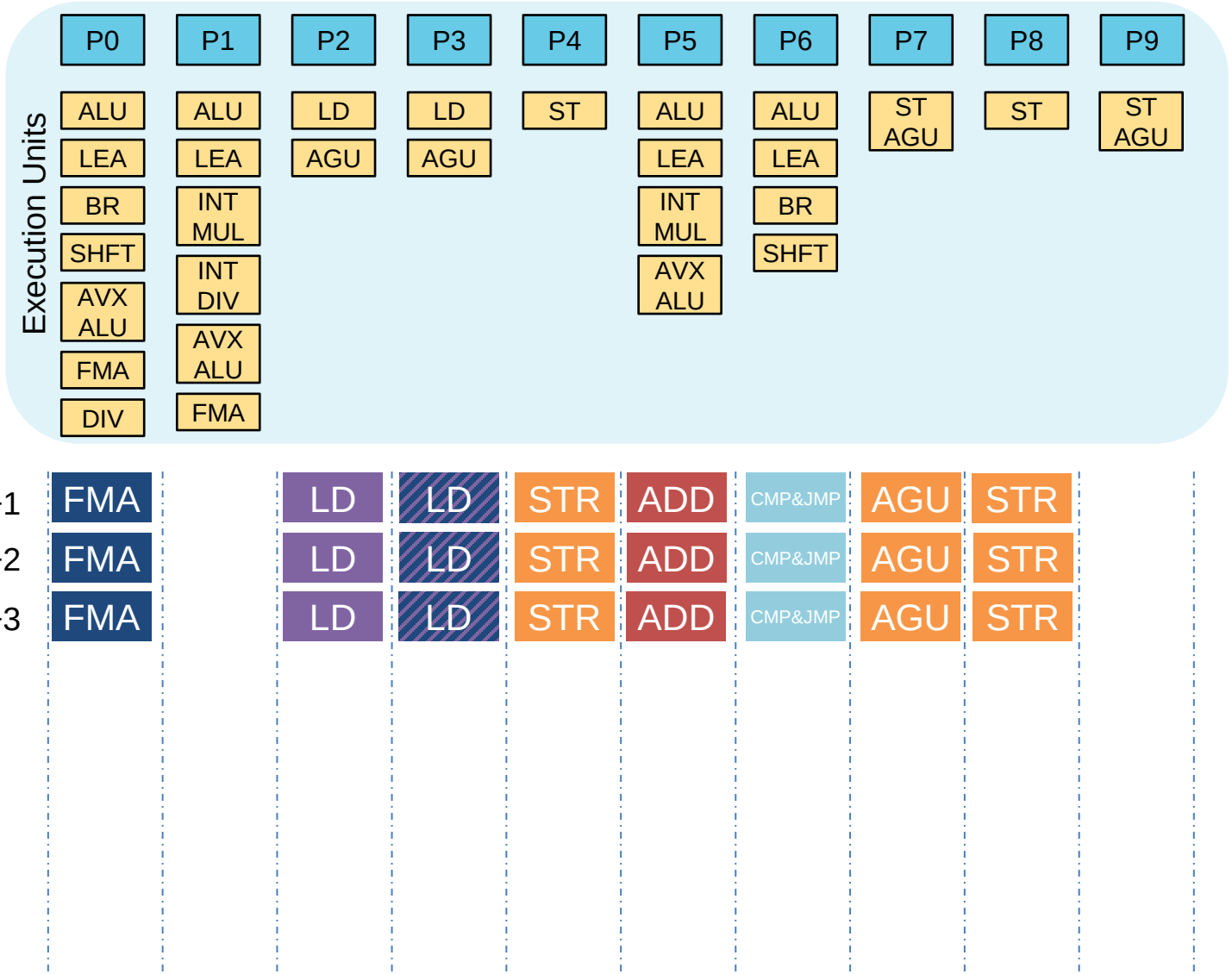


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups      (%r14,%rdx,8), %zmm1
  vfmadd213pd  (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd      %zmm1, (%r12,%rdx,8)
  addq         $8, %rdx
  cmpq         %rsi, %rdx
  jb           ..B2.42
```

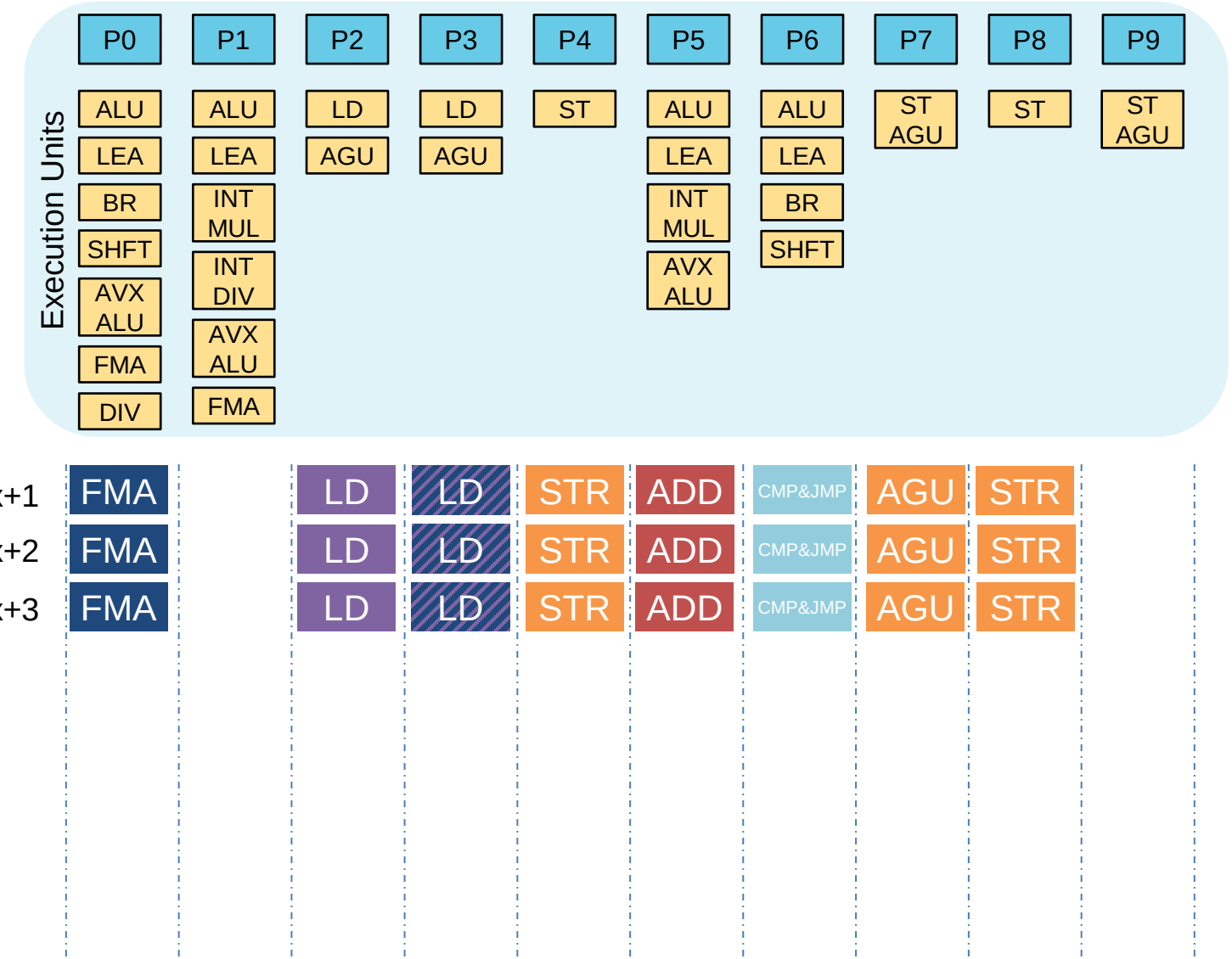


STREAM TRIAD on Intel Ice Lake

STREAM TRIAD

```
a[i] = b[i] + s * c[i]
```

```
..B2.42:
  vmovups       (%r14,%rdx,8), %zmm1
  vfmadd213pd   (%r15,%rdx,8), %zmm2, %zmm1
  vmovupd       %zmm1, (%r12,%rdx,8)
  addq          $8, %rdx
  cmpq          %rsi, %rdx
  jb            ..B2.42
```



1 cy / 8 it

OSACA

- Open Source Architecture Code Analyzer
- Static in-core code analysis for x86 and ARM µ-archs

  Assumptions

  - Steady-state execution (no warm-up/cool-down)
  - All data in L1
  - Perfect out-of-order scheduling
  - (currently) no front-end, i.e., no limit in instruction fetching, decoding, etc…
- Architecture specific model for each µ-arch

- Python module

```
$ pip install osaca
```

Workflow

Marked assembly

x86

```
# OSACA-BEGIN
..B2.42:
  vmovups (%r14,%rdx,8),%zmm1
  vfmadd213pd (%r15,%rdx,8),%zmm2,%zmm1
  vmovupd %zmm1,(%r12,%rdx,8)
  addq $8,%rdx
  cmpq %rsi,%rdx
  jne ..B2.42
# OSACA-END
```

arm

```
// OSACA-BEGIN
.L18:
  ldr q2, [x20, x0]
  ldr q1, [x21, x0]
  fmla v1.2d, v2.2d, v0.2d
  str q1, [x19, x0]
  add x0, x0, #16
  cmp x22, x0
  bne .L18
// OSACA-END
```

Machine file / database

```
load_latency: {gpr: 5, xmm: 5, ymm: 5, zmm: 5}
load_throughput_default: [[1,"23"], [1,["2D","3D"]]]
store_throughput_default: [[1,"79"], [1,"48"]]

- # ...
- name: vfmadd213pd
  operands:
  - class: register
    name: zmm
    source:      True
    destination: False
  - class: register
    name: zmm
    source:      True
    destination: False
  - class: register
    name: zmm
    source:      True
    destination: Frue
throughput: 0.5
latency: 4    #  1*p05
port_pressure: [[1,"05"]]
- # ...
```

Workflow – Creation of machine file

Workflow – Creation of machine file

ibench

```
...
loop:
  inc       i
  INSTR     zmm3, zmm0, zmm1
  INSTR     zmm4, zmm1, zmm0
  INSTR     zmm5, zmm0, zmm2
  cmp       i, N
  INSTR     zmm6, zmm2, zmm0
  INSTR     zmm7, zmm1, zmm2
  INSTR     zmm8, zmm2, zmm1
  jl        loop
...
```

Workflow – Creation of machine file

ibench

```
...
loop:
  inc       i
  INSTR     zmm3, zmm0, zmm1
  INSTR     zmm4, zmm1, zmm0
  INSTR     zmm5, zmm0, zmm2
  cmp       i, N
  INSTR     zmm6, zmm2, zmm0
  INSTR     zmm7, zmm1, zmm2
  INSTR     zmm8, zmm2, zmm1
  jl        loop
...
```

asmbench

```
asmbench -f 2.4 -t 8 -p 6 "vfmadd213pd
{src:<4 x double>:x}, {src:<4 x double>:x},
{srcdst:<4 x double>:x}"
```

Workflow – Creation of machine file

ibench

```
...
loop:
  inc       i
  INSTR     zmm3, zmm0, zmm1
  INSTR     zmm4, zmm1, zmm0
  INSTR     zmm5, zmm0, zmm2
  cmp       i, N
  INSTR     zmm6, zmm2, zmm0
  INSTR     zmm7, zmm1, zmm2
  INSTR     zmm8, zmm2, zmm1
  jl        loop
...
```

asmbench

```
asmbench -f 2.4 -t 8 -p 6 "vfmadd213pd
{src:<4 x double>:x}, {src:<4 x double>:x},
{srcdst:<4 x double>:x}"
```

Workflow – Creation of machine file

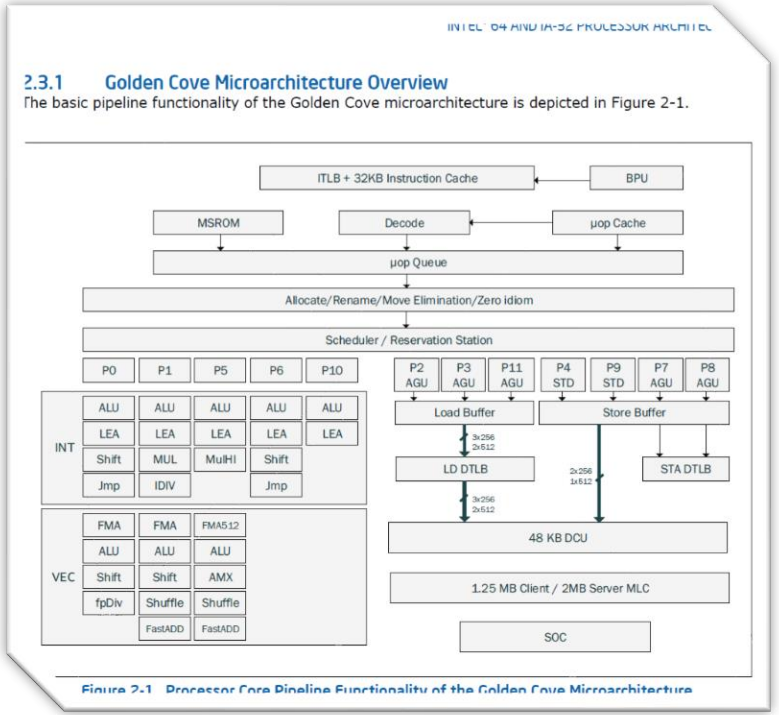
ibench

```
...
loop:
  inc       i
  INSTR     zmm3, zmm0, zmm1
  INSTR     zmm4, zmm1, zmm0
  INSTR     zmm5, zmm0, zmm2
  cmp       i, N
  INSTR     zmm6, zmm2, zmm0
  INSTR     zmm7, zmm1, zmm2
  INSTR     zmm8, zmm2, zmm1
  jl        loop
...
```

asmbench

```
asmbench -f 2.4 -t 8 -p 6 "vfmadd213pd
{src:<4 x double>:x}, {src:<4 x double>:x},
{srcdst:<4 x double>:x}"
```

vendor documentation



Workflow – Creation of machine file

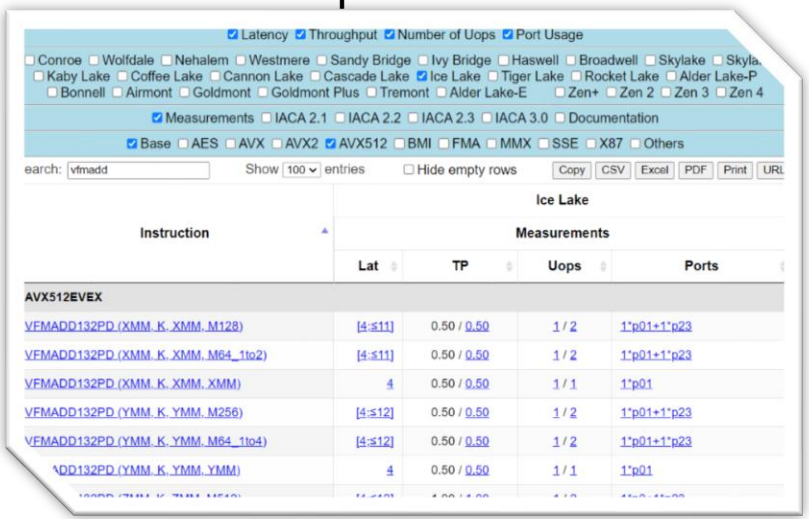
ibench

```
...
loop:
  inc       i
  INSTR     zmm3, zmm0, zmm1
  INSTR     zmm4, zmm1, zmm0
  INSTR     zmm5, zmm0, zmm2
  cmp       i, N
  INSTR     zmm6, zmm2, zmm0
  INSTR     zmm7, zmm1, zmm2
  INSTR     zmm8, zmm2, zmm1
  jl        loop
...
```

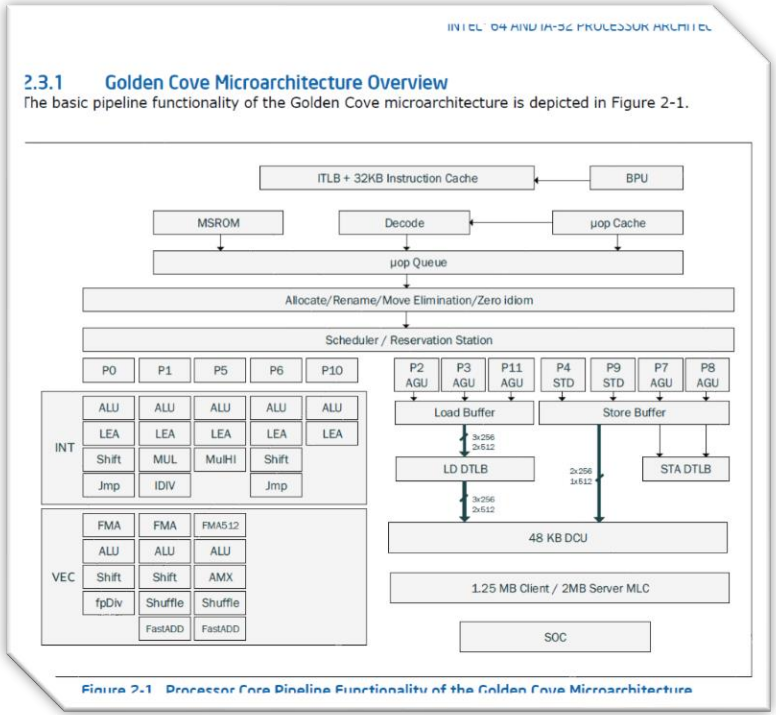
asmbench

```
asmbench -f 2.4 -t 8 -p 6 "vfmadd213pd
{src:<4 x double>:x}, {src:<4 x double>:x},
{srcdst:<4 x double>:x}"
```

uops.info

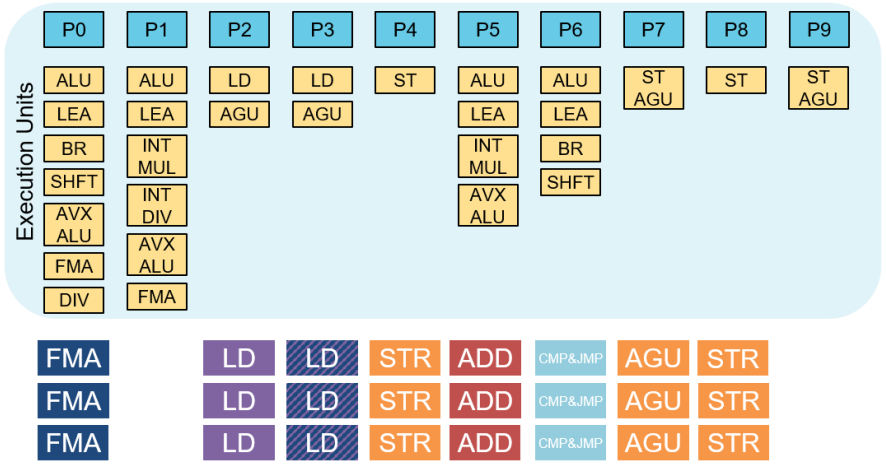


vendor documentation



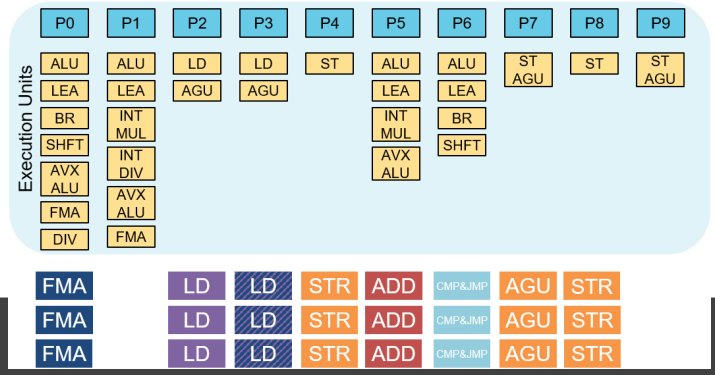
Triad on ICX with OSACA

- Recap: Manual analysis resulted in 1 cy/8 it



Triad on ICX with OSACA

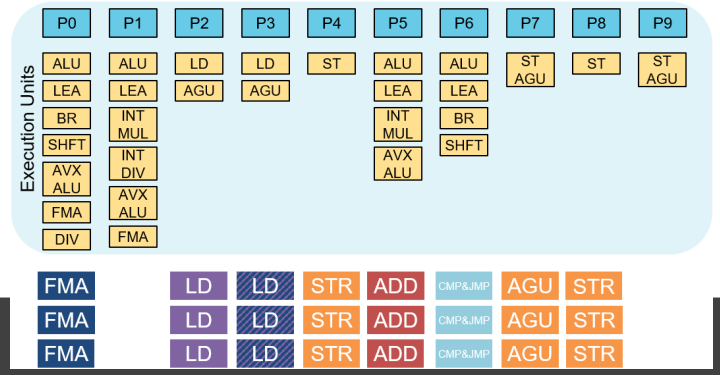
- Recap: Manual analysis resulted in 1 cy/8 it




```
$ osaca --arch ICX triad.s
Open Source Architecture Code Analyzer (OSACA) - 0.4.9
Architecture:       ICX

 * - Instruction micro-ops not bound to a port
 X - No throughput/latency information for this instruction in data file

                                        Port pressure in cycles
  | 0  - 0DV | 1  - 1DV|  2  -  2D  |  3  -  3D  |  4   |  5   |  6   |  7   |  8   |  9   ||CP |LCD|
--------------------------------------------------------------------------------------------------
1 |          |         |            |            |      |      |      |      |      |      ||   |   |   ..B2.42:
2 |          |         | 0.50  0.50 | 0.50  0.50 |      |      |      |      |      |      || 5 |   |   vmovups      (%r14,%rdx,8),%zmm1
3 | 0.50     |         | 0.50  0.50 | 0.50  0.50 |      | 0.50 |      |      |      |      || 4 |   |   vfmadd213pd (%r15,%rdx,8),%zmm2,%zmm1
4 |          |         |            |            | 1.00 |      |      | 0.50 | 1.00 | 0.50 || 0 |   |   vmovupd      %zmm1, (%r12,%rdx,8)
5 | 0.25     | 0.25    |            |            |      | 0.25 | 0.25 |      |      |      ||   | 1 |   addq         $8, %rdx
6 | 0.00     | 0.50    |            |            |      | 0.00 | 0.50 |      |      |      ||   |   |   cmpq         %rsi, %rdx
7 |          |         |            |            |      |      |      |      |      |      ||   |   | * jb           ..B2.42

    0.75       0.75      1.00  1.00   1.00  1.00   1.00   0.75   0.75   0.50   1.00   0.50    9   1


Loop-Carried Dependencies Analysis Report
-----------------------------------------
   5 |  1.0 | addq        $8, %rdx              | [5]
```

Triad on ICX with OSACA

- Recap: Manual analysis resulted in 1 cy/8 it




```
$ osaca --arch ICX triad.s
Open Source Architecture Code Analyzer (OSACA) - 0.4.9
Architecture:       ICX
```

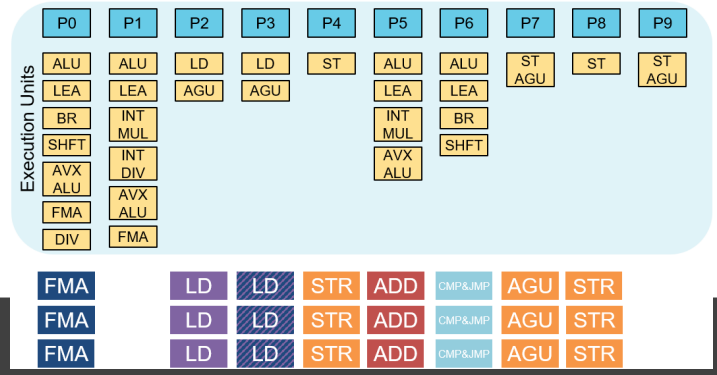
Throughput prediction

```
                                Port pressure in cycles
  | 0  - 0DV | 1  - 1DV|  2  -  2D  |  3  -  3D  |  4   |  5   |  6   |  7   |  8   |  9   |CP |LCD|
--------------------------------------------------------------------------------------------------
1 |          |         |            |            |      |      |      |      |      |      |   |   |   ..B2.42:
2 |          |         | 0.50  0.50 | 0.50  0.50 |      |      |      |      |      |      | 5 |   |   vmovups      (%r14,%rdx,8),%zmm1
3 | 0.50     |         | 0.50  0.50 | 0.50  0.50 |      | 0.50 |      |      |      |      | 4 |   |   vfmadd213pd (%r15,%rdx,8),%zmm2,%zmm1
4 |          |         |            |            | 1.00 |      |      | 0.50 | 1.00 | 0.50 | 0 |   |   vmovupd      %zmm1, (%r12,%rdx,8)
5 | 0.25     | 0.25    |            |            |      | 0.25 | 0.25 |      |      |      |   | 1 |   addq         $8, %rdx
6 | 0.00     | 0.50    |            |            |      | 0.00 | 0.50 |      |      |      |   |   |   cmpq         %rsi, %rdx
7 |          |         |            |            |      |      |      |      |      |      |   |   | * jb           ..B2.42

   0.75       0.75      1.00  1.00   1.00  1.00   1.00   0.75   0.75   0.50   1.00   0.50   9   1
```

```
Loop-Carried Dependencies Analysis Report
-----------------------------------------
   5 |  1.0 | addq        $8, %rdx              | [5]
```

Triad on ICX with OSACA

- Recap: Manual analysis resulted in 1 cy/8 it




```
$ osaca --arch ICX triad.s
Open Source Architecture Code Analyzer (OSACA) - 0.4.9
Architecture:       ICX
```

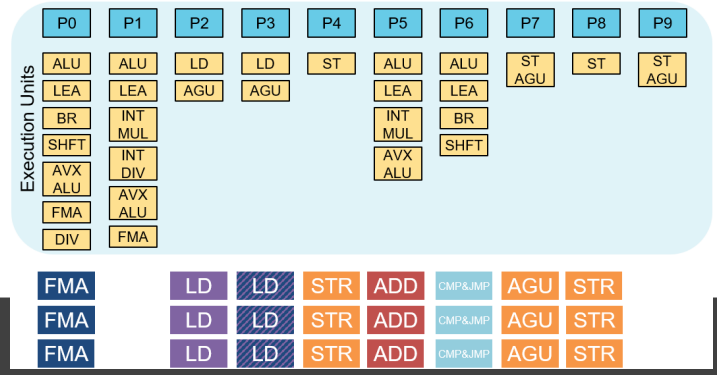
Throughput prediction

Critical Path analysis

```
                                Port pressure in cycles
  | 0  - 0DV | 1  - 1DV|  2  -  2D  |  3  -  3D  | 4    | 5    | 6    | 7    | 8    | 9    |CP |LCD|
-------------------------------------------------------------------------------------------------
1 |          |         |            |            |      |      |      |      |      |      |   |   |   ..B2.42:
2 |          |         | 0.50  0.50 | 0.50  0.50 |      |      |      |      |      |      | 5 |   |   vmovups      (%r14,%rdx,8),%zmm1
3 | 0.50     |         | 0.50  0.50 | 0.50  0.50 |      | 0.50 |      |      |      |      | 4 |   |   vfmadd213pd (%r15,%rdx,8),%zmm2,%zmm1
4 |          |         |            |            | 1.00 |      |      | 0.50 | 1.00 | 0.50 | 0 |   |   vmovupd      %zmm1, (%r12,%rdx,8)
5 | 0.25     | 0.25    |            |            |      | 0.25 | 0.25 |      |      |      |   | 1 |   addq         $8, %rdx
6 | 0.00     | 0.50    |            |            |      | 0.00 | 0.50 |      |      |      |   |   |   cmpq         %rsi, %rdx
7 |          |         |            |            |      |      |      |      |      |      |   |   | * jb           ..B2.42

    0.75       0.75      1.00  1.00   1.00  1.00   1.00   0.75   0.75   0.50   1.00   0.50   9   1

Loop-Carried Dependencies Analysis Report
-----------------------------------------
   5 |  1.0 | addq        $8, %rdx              | [5]
```

Triad on ICX with OSACA

- Recap: Manual analysis resulted in 1 cy/8 it




```
$ osaca --arch ICX triad.s
Open Source Architecture Code Analyzer (OSACA) - 0.4.9
Architecture:       ICX
```

Throughput prediction

Critical Path analysis

```
                                Port pressure in cycles
  | 0  - 0DV | 1  - 1DV|  2  -  2D  |  3  -  3D  | 4    | 5    | 6    | 7    | 8    | 9    |CP | LCD |
--------------------------------------------------------------------------------------------------------
1 |          |         |            |            |      |      |      |      |      |      |   |     |
2 |          |         | 0.50  0.50 | 0.50  0.50 |      |      |      |      |      |      | 5 |     |
3 | 0.50     |         | 0.50  0.50 | 0.50  0.50 |      | 0.50 |      |      |      |      | 4 |     |
4 |          |         |            |            | 1.00 |      |      | 0.50 | 1.00 | 0.50 | 0 |     |
5 | 0.25     | 0.25    |            |            |      | 0.25 | 0.25 |      |      |      |   | 1   |
6 | 0.00     | 0.50    |            |            |      | 0.00 | 0.50 |      |      |      |   |     |
7 |          |         |            |            |      |      |      |      |      |      |   |     |

    0.75       0.75      1.00  1.00   1.00  1.00   1.00   0.75   0.75   0.50   1.00   0.50   9   1
```

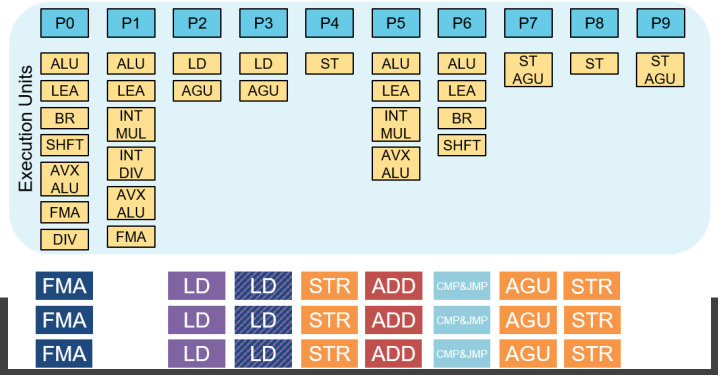
```
..B2.42:
vmovups       (%r14,%rdx,8),%zmm1
vfmadd213pd   (%r15,%rdx,8),%zmm2,%zmm1
vmovupd       %zmm1, (%r12,%rdx,8)
addq          $8, %rdx
cmpq          %rsi, %rdx
* jb          ..B2.42
```

```
Loop-Carried Dependencies Analysis Report
-----------------------------------------
   5 |  1.0 | addq        $8, %rdx              | [5]
```

Loop-carried dependency analysis

Triad on ICX with OSACA

- Recap: Manual analysis resulted in 1 cy/8 it




```
$ osaca --arch ICX triad.s
Open Source Architecture Code Analyzer (OSACA) - 0.4.9
Architecture:       ICX



                                Port pressure in cycles
  | 0   - 0DV | 1   - 1DV|  2   -  2D  |  3   -  3D  |  4   |  5   |  6   |  7   |  8   |  9   ||CP |LCD|
--------------------------------------------------------------------------------------------------------
1 |           |          |             |             |      |      |      |      |      |      ||   |   |   ..B2.42:
2 |           |          | 0.50  0.50  | 0.50  0.50  |      |      |      |      |      |      || 5 |   |   vmovups      (%r14,%rdx,8),%zmm1
3 | 0.50      |          | 0.50  0.50  | 0.50  0.50  |      | 0.50 |      |      |      |      || 4 |   |   vfmadd213pd (%r15,%rdx,8),%zmm2,%zmm1
4 |           |          |             |             | 1.00 |      |      | 0.50 | 1.00 | 0.50 || 0 |   |   vmovupd      %zmm1, (%r12,%rdx,8)
5 | 0.25      | 0.25     |             |             |      | 0.25 | 0.25 |      |      |      ||   | 1 |   addq         $8, %rdx
6 | 0.00      | 0.50     |             |             |      | 0.00 | 0.50 |      |      |      ||   |   |   cmpq         %rsi, %rdx
7 |           |          |             |             |      |      |      |      |      |      ||   |   | * jb           ..B2.42

    0.75        0.75       1.00  1.00    1.00  1.00    1.00   0.75   0.75   0.50   1.00   0.50    9   1


Loop-Carried Dependencies Analysis Report
-----------------------------------------
   5 |  1.0 | addq        $8, %rdx              | [5]
```

Useful gimmicks

- Marker insertion
- Dependency graph creation
- Compiler Explorer integration

Marking the region of interest

```
$ osaca --arch icx --insert-marker

Blocks found in assembly file:
..B1.18
        ...
..B1.21
        ...
..B1.46
        ...
..B1.49
        ...
---------------------------
Possible blocks to be marked:
..B1.18
..B1.21
..B1.46
..B1.49
Choose block to be marked [..B1.49]:
```

```
..B1.18:      # Preds ..B1.18 ..B1.17
  # Execution count [8.00e-01]: Infreq
    movq    %rcx, (%r14,%rbx,8)   #34.5
    movq    %rax, (%r12,%rbx,8)   #35.5
    incq    %rbx                  #33.3
    cmpq    %r9, %rbx             #33.3
    jb  ..B1.18     # Prob 81%    #33.3
```

```
..B1.21:  # Preds ..B1.21 ..B1.20
  # Execution count [8.00e-01]: Infreq
    vmovupd    %ymm1, (%r14,%r9,8)      #34.5
    vmovupd    %ymm0, (%r12,%r9,8)      #35.5
    vmovupd    %ymm1, 32(%r14,%r9,8)    #34.5
    vmovupd    %ymm0, 32(%r12,%r9,8)    #35.5
    vmovupd    %ymm1, 64(%r14,%r9,8)    #34.5
    vmovupd    %ymm0, 64(%r12,%r9,8)    #35.5
    vmovupd    %ymm1, 96(%r14,%r9,8)    #34.5
    vmovupd    %ymm0, 96(%r12,%r9,8)    #35.5
    addq       $16, %r9                 #33.3
    cmpq       %rax, %r9                #33.3
    jb         ..B1.21     # Prob 81%   #33.3
```

```
..B1.46:  # Preds ..B1.44 ..B1.46
   # Execution count [2.22e+01]: Infreq
  vmovsd     (%rbx,%rsi,8), %xmm3          #53.24
  vfmadd231sd (%r14,%rsi,8),%xmm3,%xmm0 #53.9
  incq       %rsi                          #52.7
  cmpq       %rdi, %rsi                    #52.7
  jb         ..B1.46       # Prob 82%      #52.7
```

```
..B1.49: # Preds ..B1.49 ..B1.48
  # Execution count [2.22e+01]: Infreq
  vmovupd    (%rbx,%rdi,8), %ymm6              #53.24
  vmovupd    32(%rbx,%rdi,8), %ymm7            #53.24
  vmovupd    64(%rbx,%rdi,8), %ymm8            #53.24
  vmovupd    96(%rbx,%rdi,8), %ymm9            #53.24
  vfmadd231pd (%r14,%rdi,8), %ymm6, %ymm2      #53.24
  vfmadd231pd 32(%r14,%rdi,8), %ymm7, %ymm5 #53.24
  vfmadd231pd 64(%r14,%rdi,8), %ymm8, %ymm4 #53.24
  vfmadd231pd 96(%r14,%rdi,8), %ymm9, %ymm3 #53.24
  addq       $16, %rdi                         #52.7
  cmpq       %rsi, %rdi                        #52.7
  jb         ..B1.49       # Prob 82%          #52.7
```

will be marked with **byte** markers, i.e.:

```
movl $111,%ebx; .byte 100,103,144;   (x86)
...
movl $222,%ebx; .byte 100,103,144;
```

Selects all inner-most loops and
suggest the one with the most SIMD instructions

Dependency graph creation

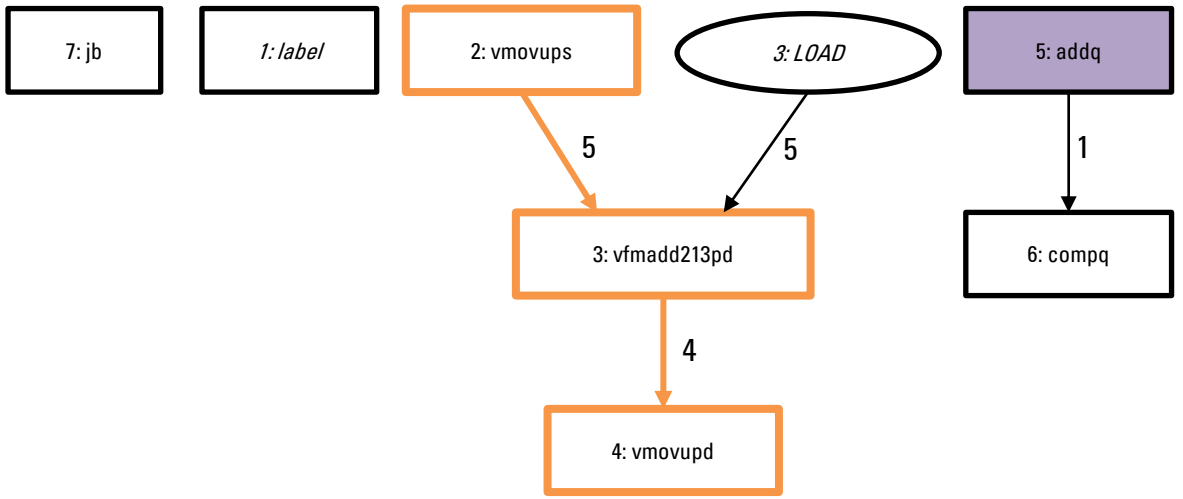
```
                                 Port pressure in cycles
  | 0  - 0DV | 1  - 1DV|  2  -  2D  |  3  -  3D  |  4   |  5   |  6   |  7   |  8   |  9   ||CP |LCD|
-------------------------------------------------------------------------------------------------
1 |          |         |            |            |      |      |      |      |      |      ||   |   |   ..B2.42:
2 |          |         | 0.50  0.50 | 0.50  0.50 |      |      |      |      |      |      || 5 |   |   vmovups     (%r14,%rdx,8),%zmm1
3 | 0.50     |         | 0.50  0.50 | 0.50  0.50 |      | 0.50 |      |      |      |      || 4 |   |   vfmadd213pd (%r15,%rdx,8),%zmm2,%zmm1
4 |          |         |            |            | 1.00 |      |      | 0.50 | 1.00 | 0.50 || 0 |   |   vmovupd     %zmm1, (%r12,%rdx,8)
5 | 0.25     | 0.25    |            |            |      | 0.25 | 0.25 |      |      |      ||   | 1 |   addq        $8, %rdx
6 | 0.00     | 0.50    |            |            |      | 0.00 | 0.50 |      |      |      ||   |   |   cmpq        %rsi, %rdx
7 |          |         |            |            |      |      |      |      |      |      ||   |   | * jb          ..B2.42

    0.75       0.75      1.00  1.00   1.00  1.00   1.00   0.75   0.75   0.50   1.00   0.50    9   1


Loop-Carried Dependencies Analysis Report
-----------------------------------------
   5 |  1.0 | addq        $8, %rdx              | [5]
```

```
$ osaca --arch icx --export-graph dependencies.dot triad.s
$ dot -Tpdf dependencies.dot -o dep_graph.pdf
```

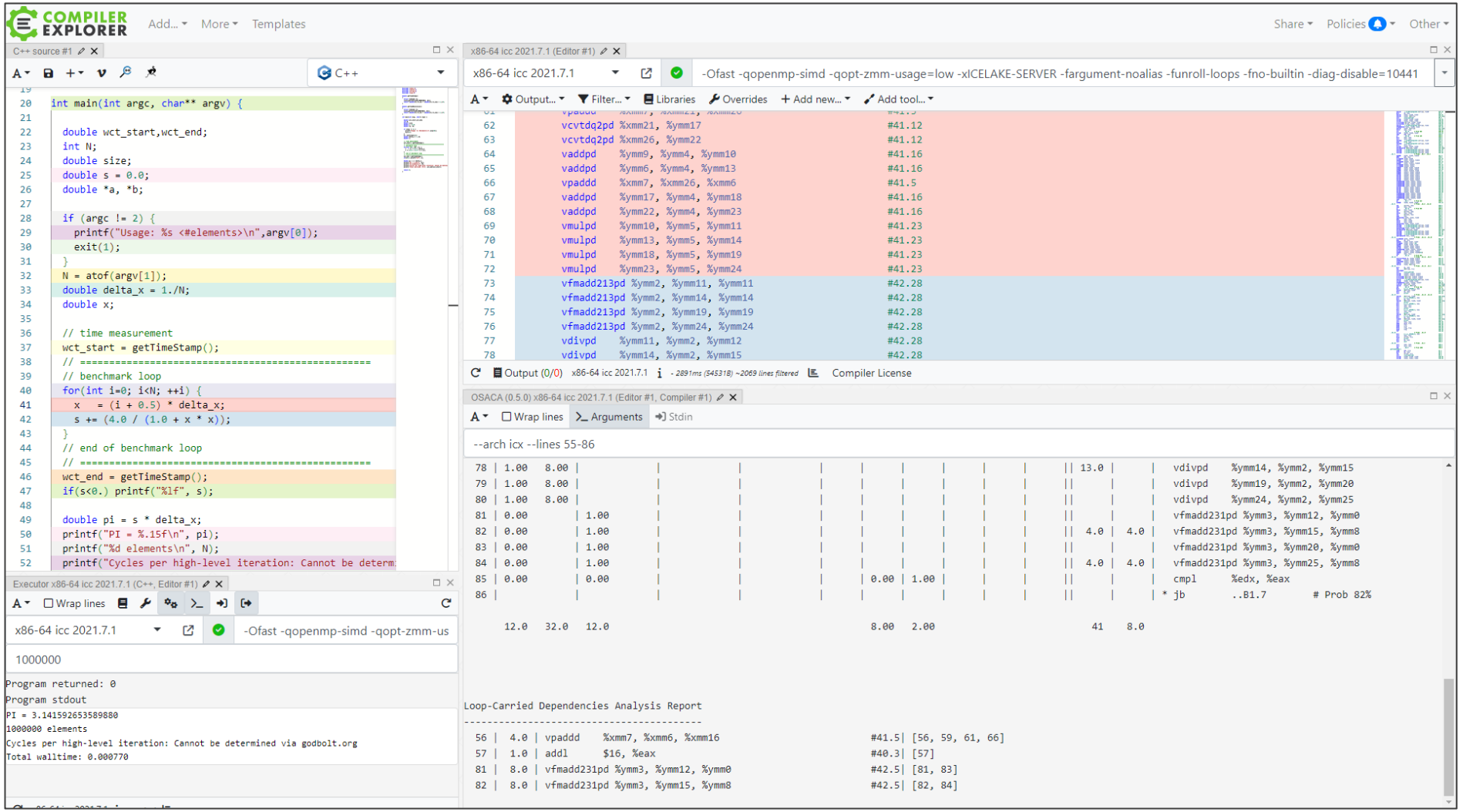
Dependency graph creation

```
                                Port pressure in cycles
  | 0  - 0DV | 1  - 1DV|  2  -  2D  |  3  -  3D  |  4   |  5   |  6   |  7   |  8   |  9   ||CP |LCD|
-------------------------------------------------------------------------------------------------
1 |          |         |            |            |      |      |      |      |      |      ||   |   |     B2.42:
2 |          |         | 0.50  0.50 | 0.50  0.50 |      |      |      |      |      |      ||  5 |   |   vmovups      (%r14,%rdx,8),%zmm1
3 | 0.50     |         | 0.50  0.50 | 0.50  0.50 |      | 0.50 |      |      |      |      ||  4 |   |   vfmadd213pd (%r15,%rdx,8),%zmm2,%zmm1
4 |          |         |            |            | 1.00 |      |      | 0.50 | 1.00 | 0.50 ||  0 |   |   vmovupd      %zmm1, (%r12,%rdx,8)
5 | 0.25     | 0.25    |            |            |      | 0.25 | 0.25 |      |      |      ||    | 1 |   addq         $8, %rdx
6 | 0.00     | 0.50    |            |            |      | 0.00 | 0.50 |      |      |      ||    |   |   cmpq         %rsi, %rdx
7 |          |         |            |            |      |      |      |      |      |      ||    |   | * jb           ..B2.42

    0.75       0.75      1.00  1.00   1.00  1.00   1.00   0.75   0.75   0.50   1.00   0.50    9   1


Loop-Carried Dependencies Analysis Report
-----------------------------------------
   5 |  1.0 | addq         $8, %rdx                | [5]
```

```
$ osaca --arch icx --export-graph dependencies.dot triad.s
$ dot -Tpdf dependencies.dot -o dep_graph.pdf
```

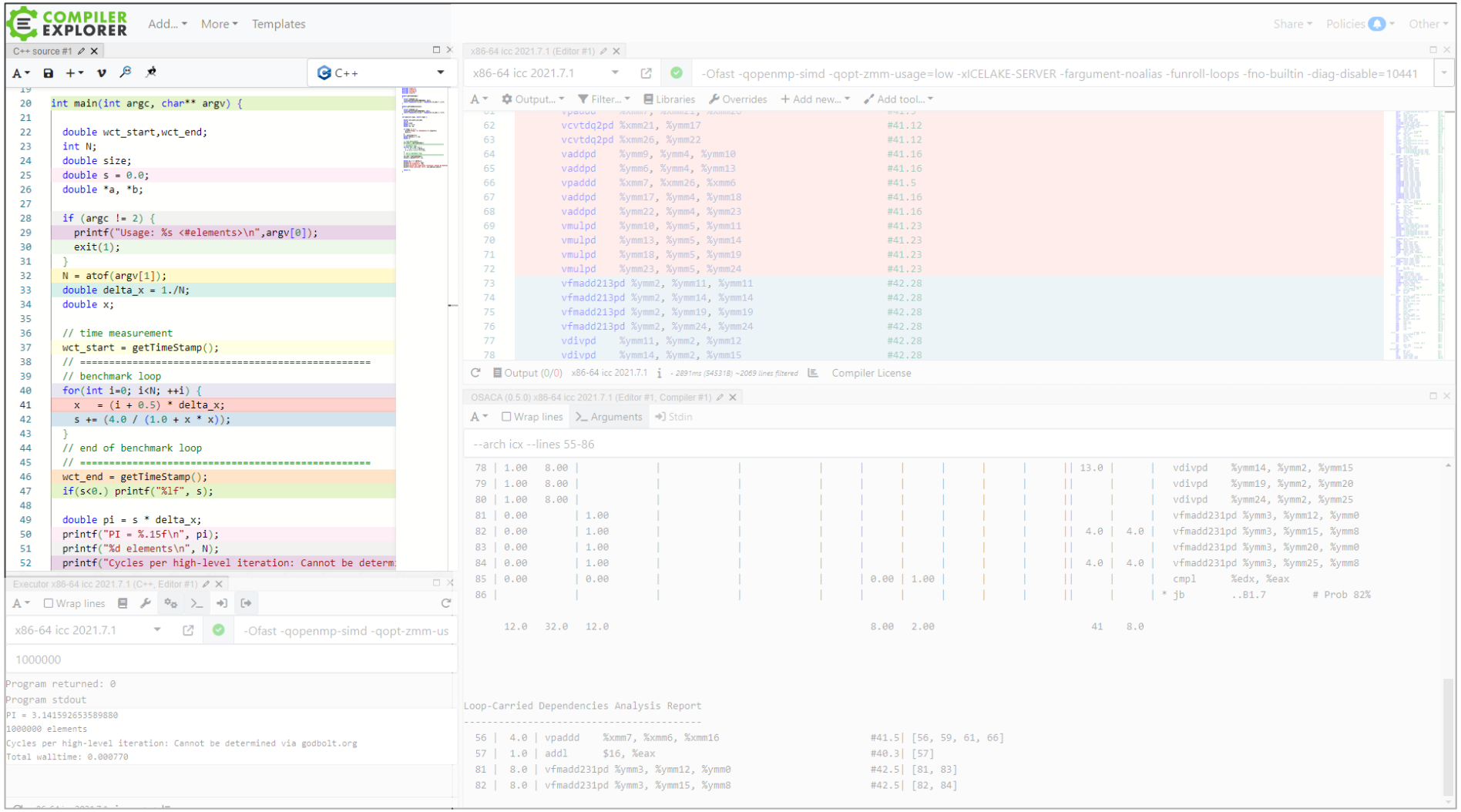


Compiler Explorer integration



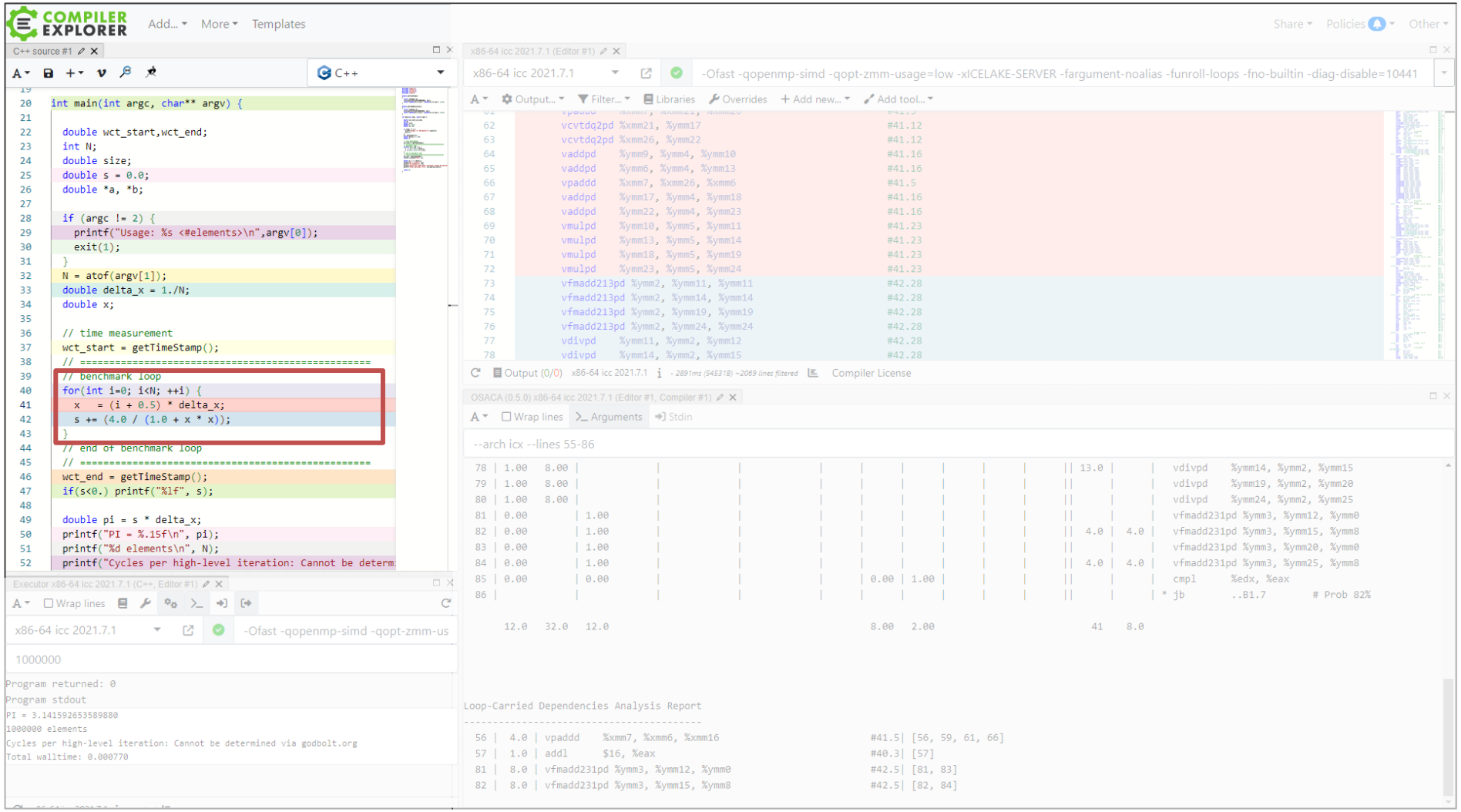
- Try it yourself: https://godbolt.org/z/GveaKe3v1

Compiler Explorer integration



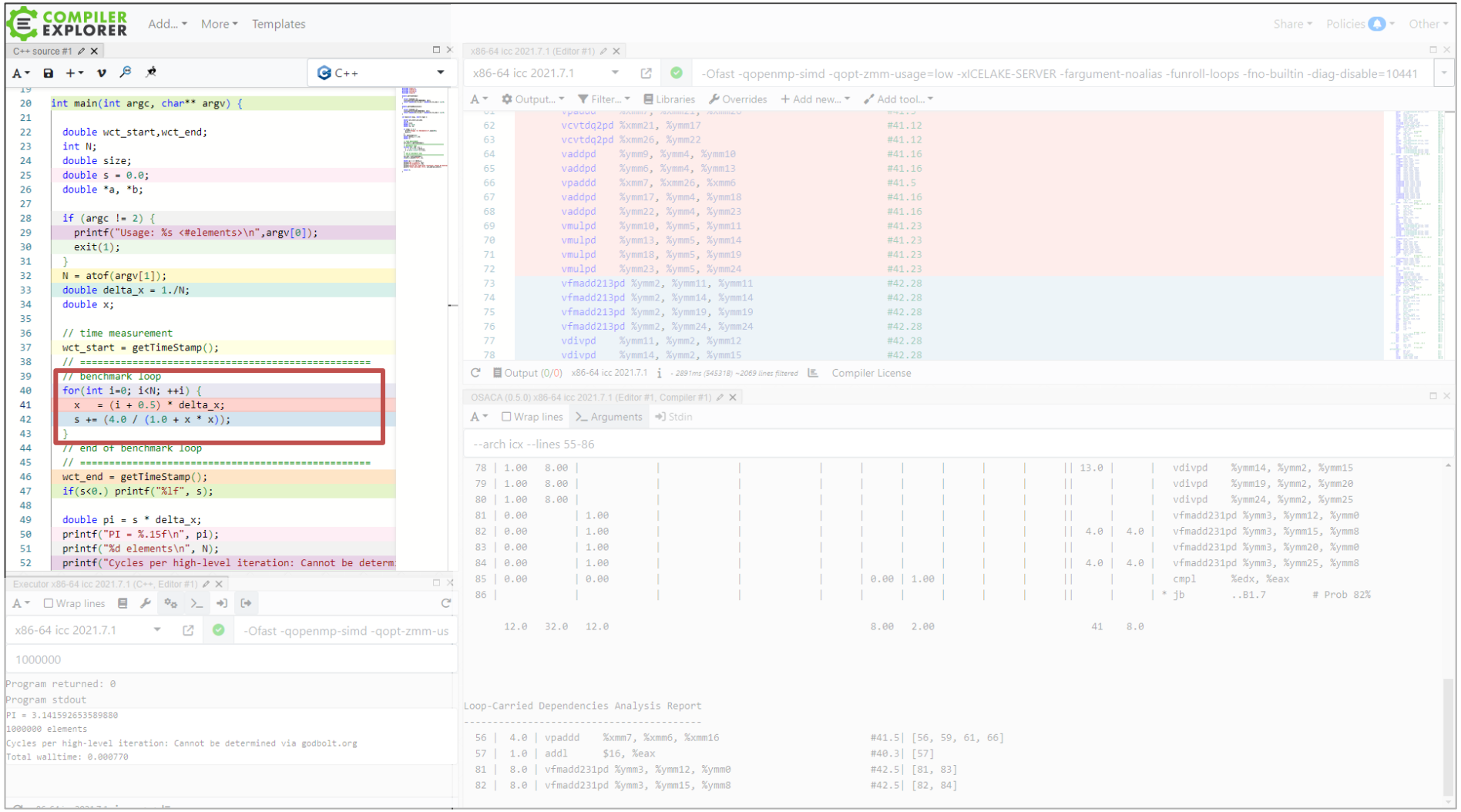
- Try it yourself: https://godbolt.org/z/GveaKe3v1

Compiler Explorer integration



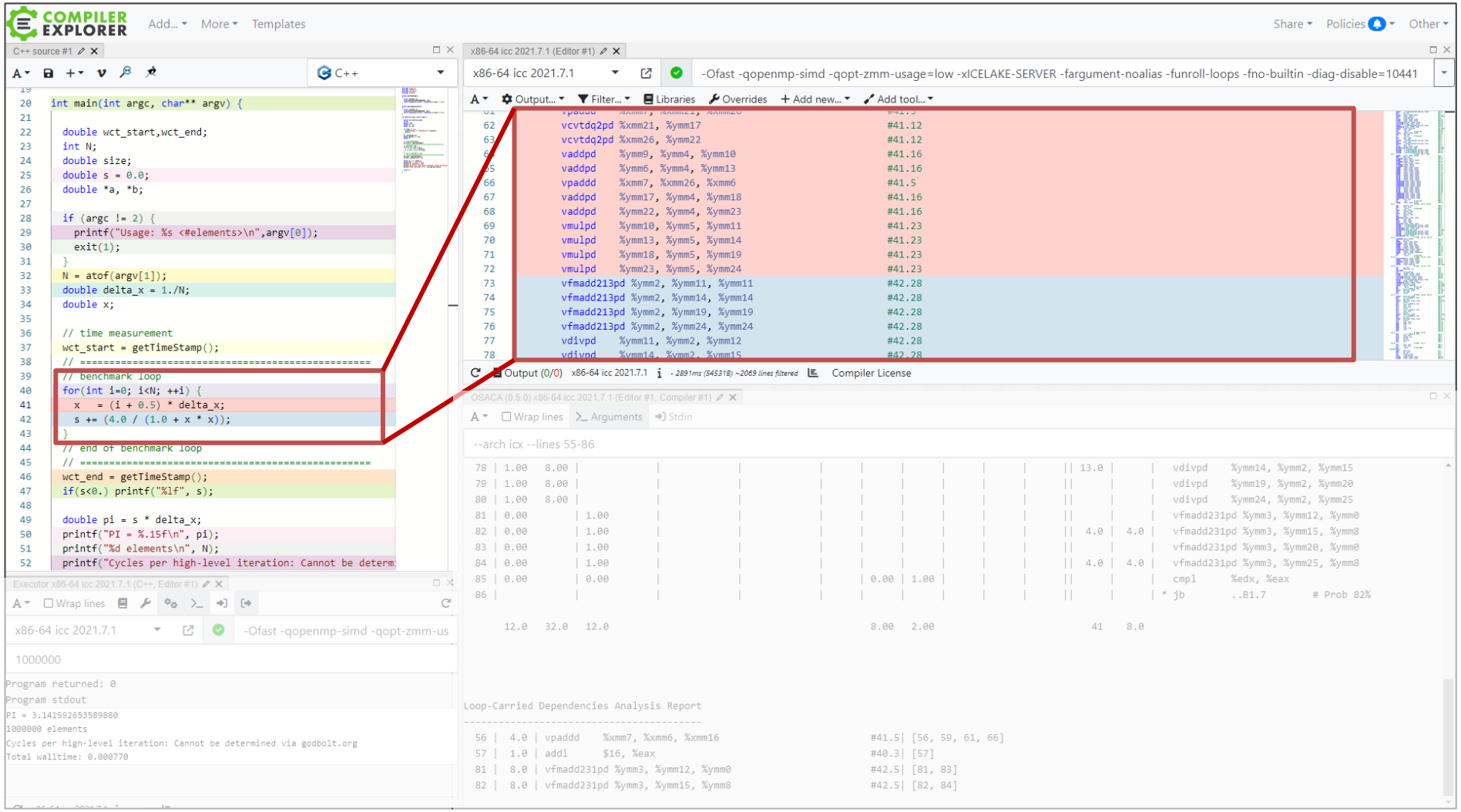
- Try it yourself: https://godbolt.org/z/GveaKe3v1

Compiler Explorer integration



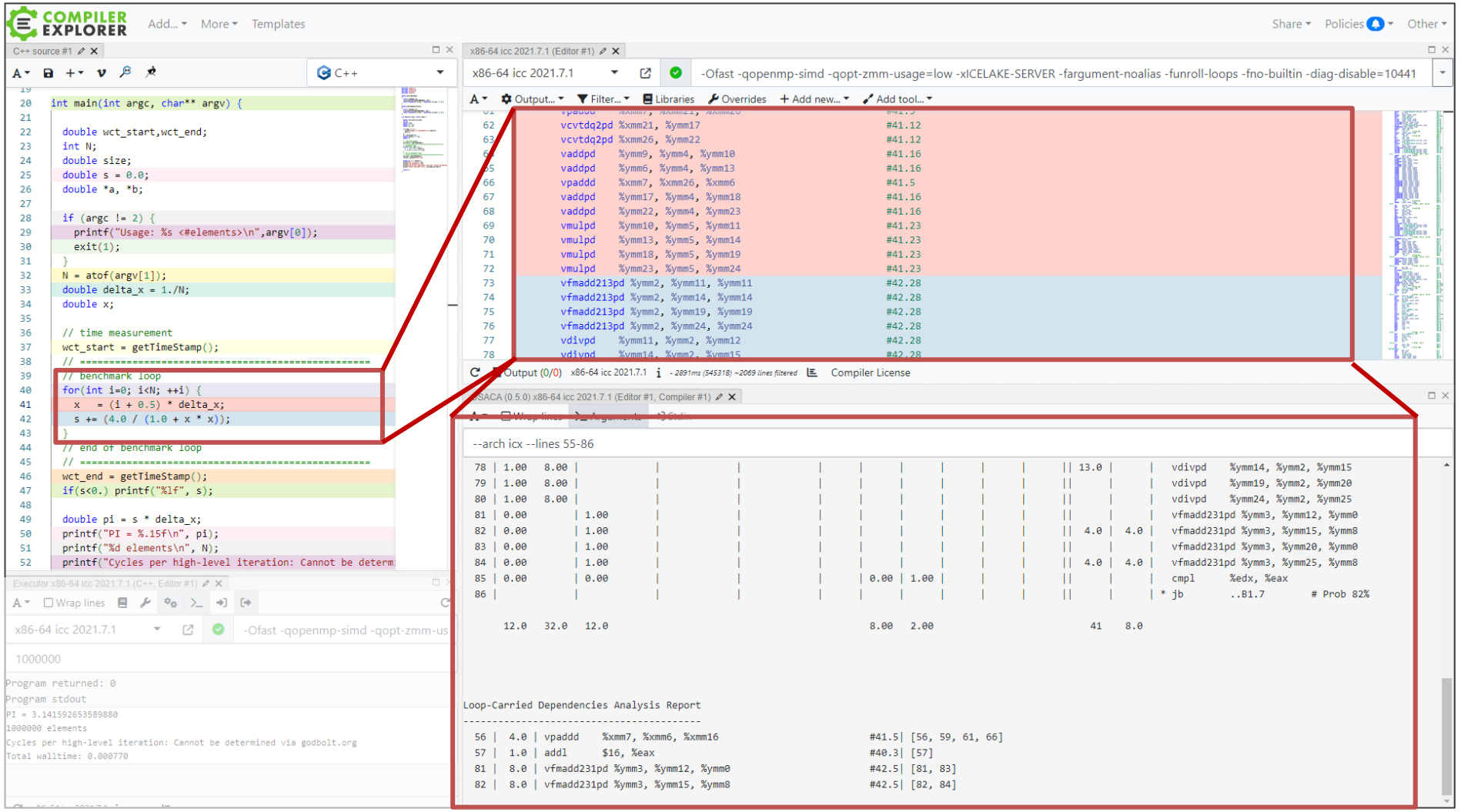
- Try it yourself: https://godbolt.org/z/GveaKe3v1

Compiler Explorer integration



- Try it yourself: https://godbolt.org/z/GveaKe3v1

Compiler Explorer integration



- Try it yourself: https://godbolt.org/z/GveaKe3v1

Compiler Explorer integration

Live demo: https://hpc-mover.rrze.uni-erlangen.de/compiler-explorer/z/va3zxe

&

Supported micro-architectures so far

- **Intel x86:**
  Sandy Bridge, Ivy Bridge, Haswell, Broadwell,
  Skylake X, Cascade Lake X, Ice Lake Server

- **AMD x86:**
  Zen 1, Zen 2, Zen 3

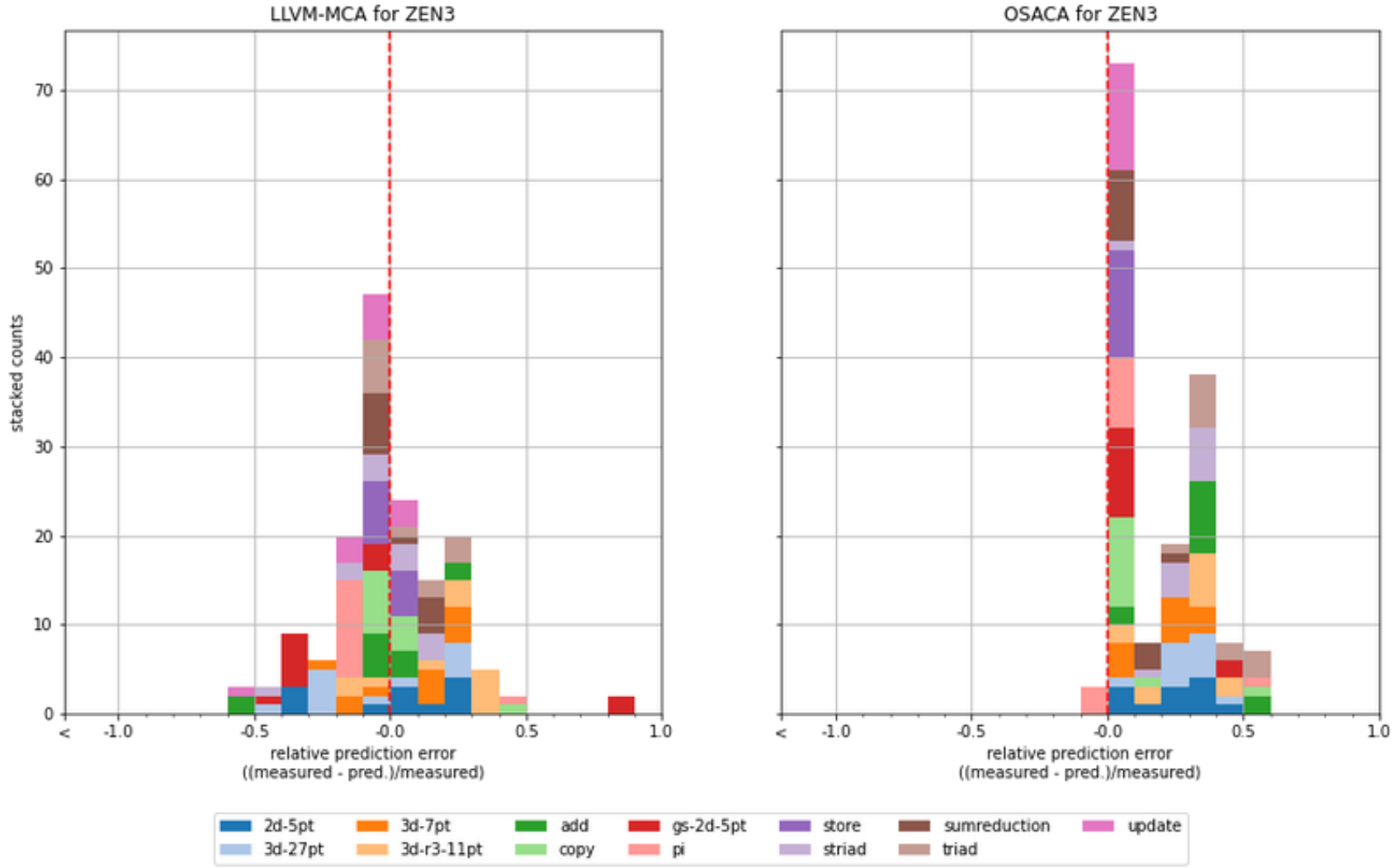
- **AArch64:**
  Marvell ThunderX2, ARM Neoverse N1, ARM Cortex A72,
  Fujitsu A64FX, HiSilicon TaiShan v110

There is not just THE one code analyzer

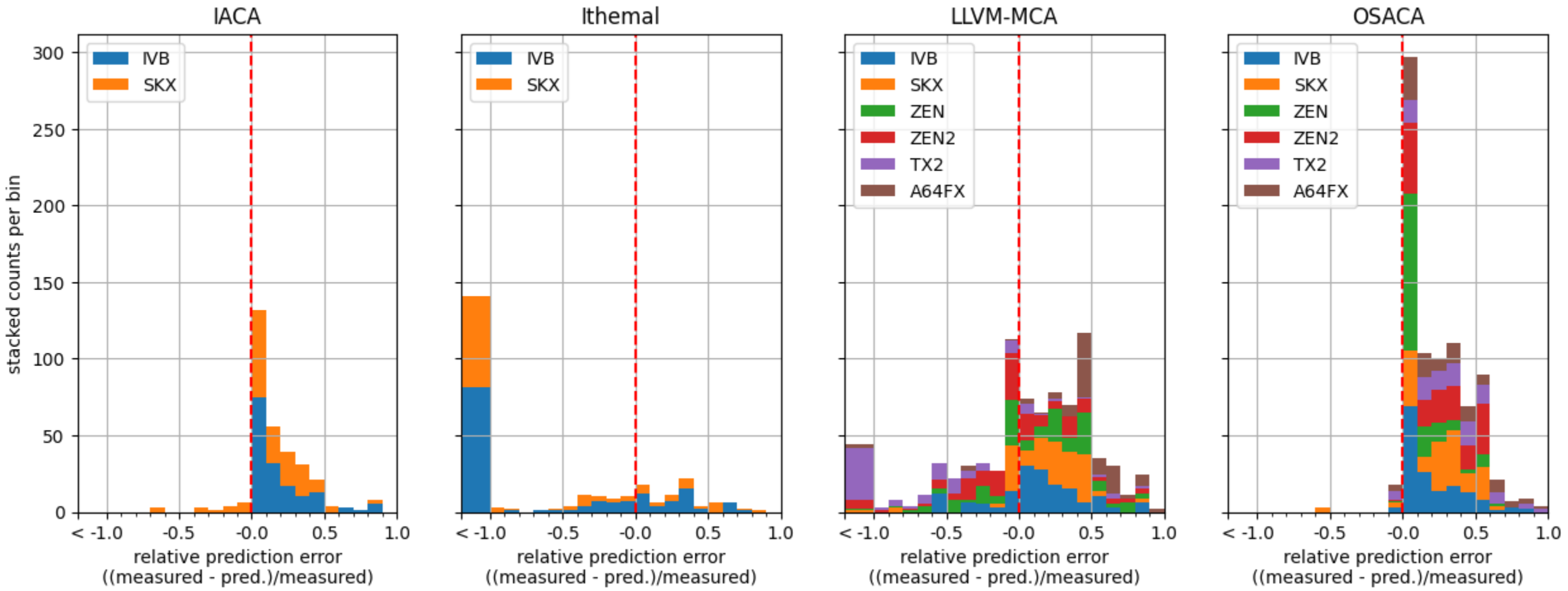
- **OSACA**: https://github.com/RRZE-HPC/OSACA
- **uiCA**: https://www.uops.info/uiCA.html
- **LLVM-MCA**: https://llvm.org/docs/CommandGuide/llvm-mca.html
- **IACA** (EoL): https://www.intel.com/content/www/us/en/developer/articles/tool/architecture-code-analyzer.html

Accuracy

- Comparison LLVM-MCA vs OSACA for Zen3



Accuracy

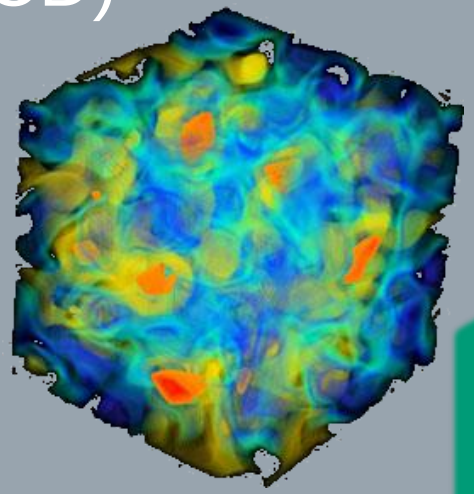

IACA
Ithemal
LLVM-MCA
OSACA
IVB
SKX
ZEN
ZEN2
TX2
A64FX
stacked counts per bin
0
50
100
150
200
250
300
< -1.0
-0.5
-0.0
0.5
1.0
relative prediction error
((measured - pred.)/measured)

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Case Study: Domain Wall (DW) Kernel

from Quantum Chromodynamics (QCD)



© Brookhaven National Lab

Based on:

C. Alappat, N. Meyer, J. Laukemann, T. Gruber, G. Hager, G. Wellein, and T. Wettig:
ECM modeling and performance tuning of SpMV and Lattice QCD on A64FX.
Concurrency and Computation: Practice and Experience, e6512 (2021).

DOI: 10.1002/cpe.6512

Context

- Lattice QCD simulates the strong interaction
- Iterative multigrid techniques on regular (4D or 5D) lattices
- Core component: Apply Dirac operator D to quark-field vector Ψ
- Domain Wall (DW) formulation: quark field lives on 4D boundary of a 5D space-time volume $V_4 \times L_s$

$$(D\psi)(n,s)_{\alpha a} =$$

$$\sum_{\mu=1}\sum_{\beta=1}\sum_{b=1}\left\{U_\mu(n)_{ab}(1+\gamma_\mu)_{\alpha\beta}\psi(n+\hat{\mu},s)_{\beta b} + U_\mu^\dagger(n-\hat{\mu})_{ab}(1-\gamma_\mu)_{\alpha\beta}\psi(n-\hat{\mu},s)_{\beta b}\right\}$$

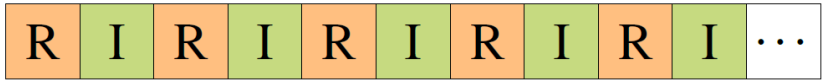
DW stencil kernel (simplified)

- “Grid” lattice QCD framework
- Uses SVE intrinsics
- Data type: double complex

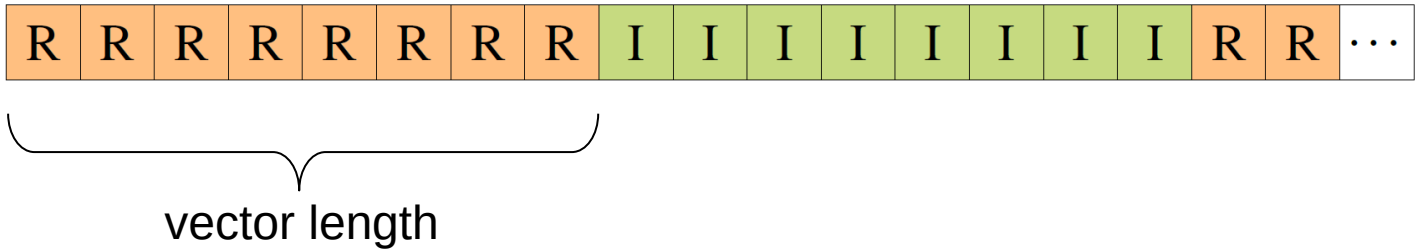
```
#define x_p 1 // x-plus direction
#define x_m 2 // x-minus direction
#define y_p 3 // y-plus direction
...
#pragma omp parallel for schedule(static)
for{t,z,y,x} = 1:{Lt-2,Lz-2,Ly-2,Lx-2} //collapsed loop over 4d space-time
{
  for(int s=0; s<Ls; ++s) //loop over fifth dimension
  {
    O[t][z][y][x][s] = R(x_p) · U[x_p][t][z][y][x] · P(x_p) · I[t][z][y][x+1][s] +
                       R(x_m) · U[x_m][t][z][y][x] · P(x_m) · I[t][z][y][x-1][s] +
                       R(y_p) · U[y_p][t][z][y][x] · P(y_p) · I[t][z][y+1][x][s] +
                       R(y_m) · U[y_m][t][z][y][x] · P(y_m) · I[t][z][y-1][x][s] +
                       R(z_p) · U[z_p][t][z][y][x] · P(z_p) · I[t][z+1][y][x][s] +
                       R(z_m) · U[z_m][t][z][y][x] · P(z_m) · I[t][z-1][y][x][s] +
                       R(t_p) · U[t_p][t][z][y][x] · P(t_p) · I[t+1][z][y][x][s] +
                       R(t_m) · U[t_m][t][z][y][x] · P(t_m) · I[t-1][z][y][x][s];
  }
}
```

Complex numbers data layout choice

AoS (standard)



AoSoA



Observed performance

- Starting point: AoS layout, ACLE intrinsics, GCC/FCC
- 1320 flops/LUP (theoretical)
- Measured code balance: 1500 byte/LUP

$$B_c \approx 1.14 \frac{\text{byte}}{\text{flop}}$$

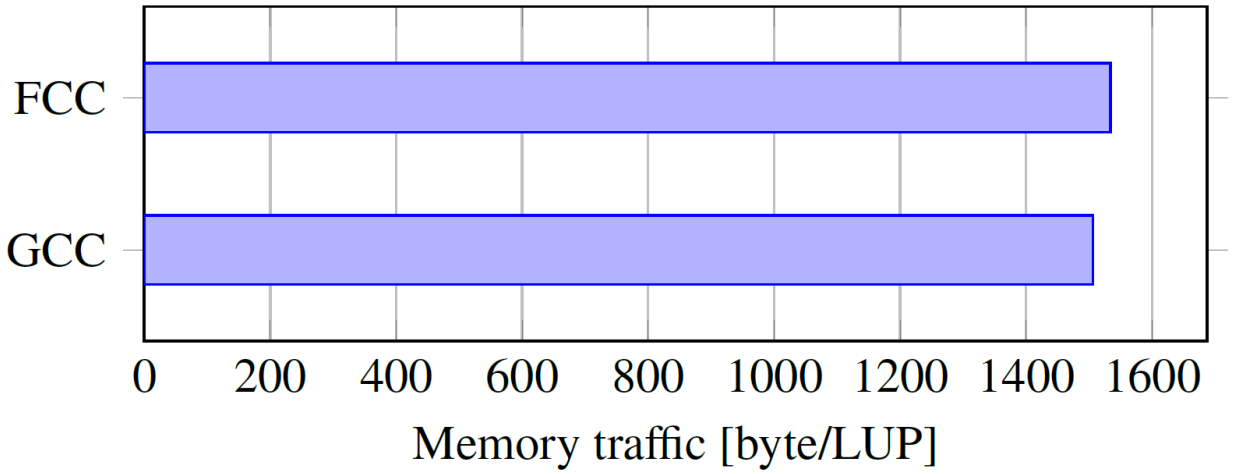
- A64FX (FX1000): $B_m = 0.25 \frac{\text{byte}}{\text{flop}}$ → expect memory bound

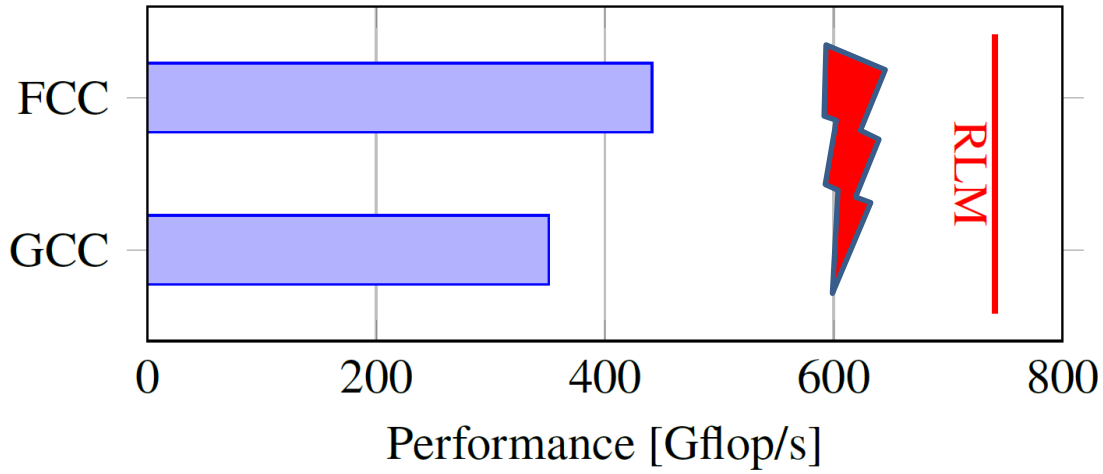
Observed performance

- Starting point: AoS layout, ACLE intrinsics, GCC/FCC
- 1320 flops/LUP (theoretical)
- Measured code balance: 1500 byte/LUP

$$B_c \approx 1.14 \frac{\text{byte}}{\text{flop}}$$

- A64FX (FX1000): $B_m = 0.25 \frac{\text{byte}}{\text{flop}}$ → expect memory bound





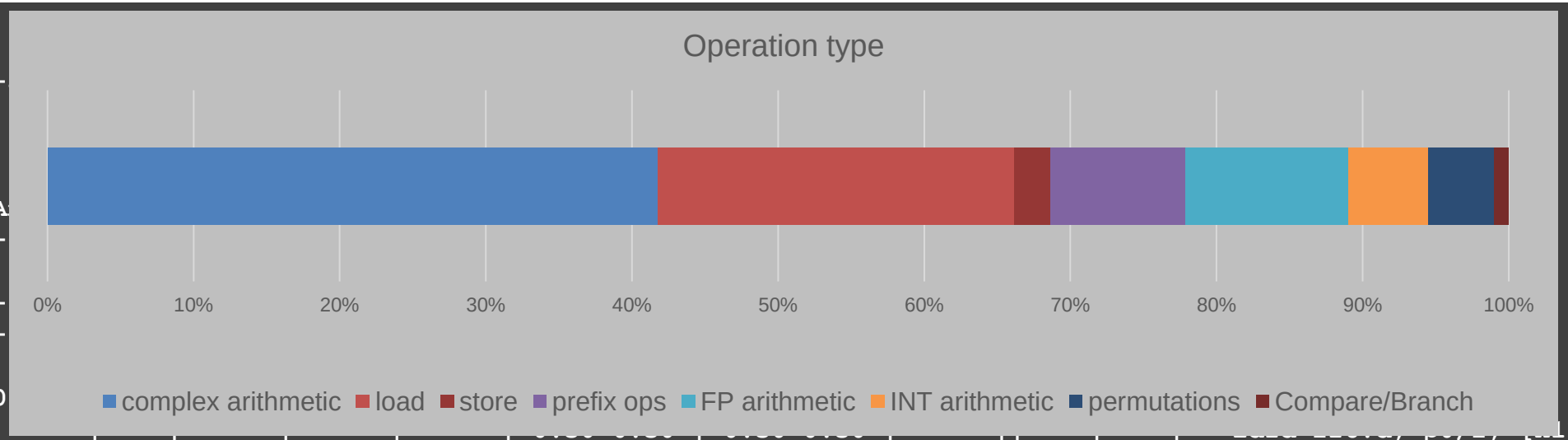
In-core analysis (complex-AoS)

```
$ osaca --arch a64fx riri-base-gcc.s
[...]


Combined Analysis Report
------------------------
                                Port pressure in cycles
    | 0  - 0DV | 1  |  2   |  3   |  4   | 5  -  5D  | 6   -  6D |  7   || CP | LCD|
-------------------------------------------------------------------------------------
560 |          |    |      |      |      |           |           |      ||    |    |   .L41:
561 | 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      ||    |    |   lsl w2, w13, 3
562 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z16.d, p0/z, [x11]
563 | 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      ||    |    |   add x18, sp, 160
564 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z18.d, p0/z, [x11, #-4, mul vl]
565 |          |    |      | 0.50 | 0.50 |           |           |      ||    |    |   sxtw x2, w2
566 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z19.d, p0/z, [x11, #-3, mul vl]
  [...]
1367| 1.00     |    |      |      |      | 1.00      | 1.00      |      ||    |    |   st1d z2.d, p0, [x0, #4, mul vl]
1368| 1.00     |    |      |      |      | 1.00      | 1.00      |      ||    |    |   st1d z13.d, p0, [x0, #5, mul vl]
1369|          |    |      | 0.00 | 1.00 |           |           |      ||    |    |   cmp w14, w13
1370|          |    |      |      |      |           |           | 1.00 ||    |    |   bne .L41
      680            500    30     30    118.5 98.5  118.5 98.5   1.0    158   1.0

Loop-Carried Dependencies Analysis Report
-----------------------------------------
1360 |  1.0 | add   w13, w13, 1                    | [1360]
```

In-core analysis (complex-AoS)

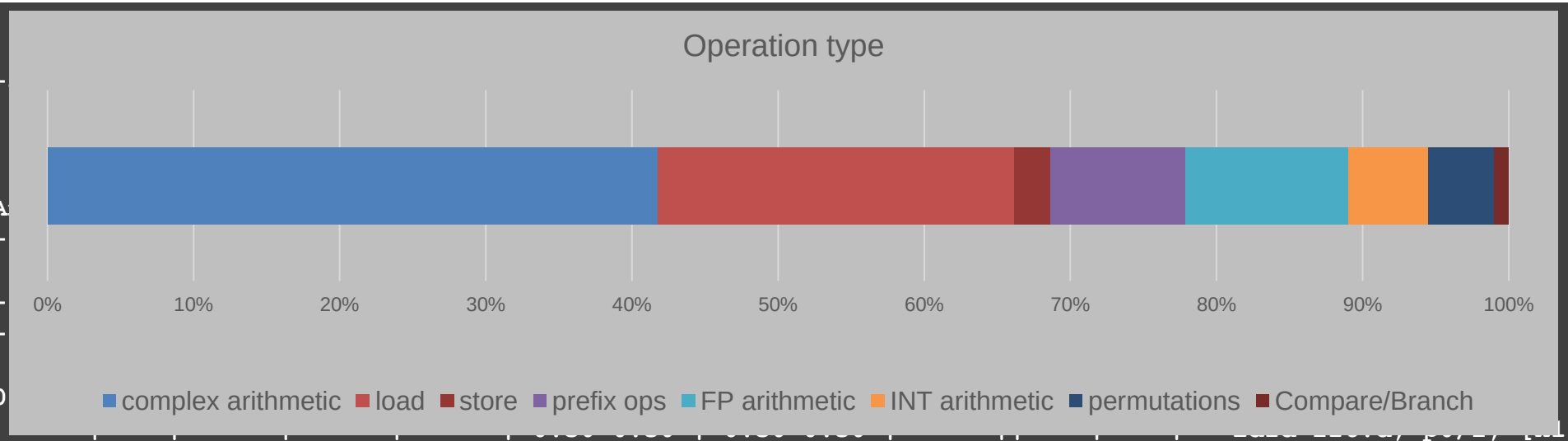
```
$ osaca --arch a64fx riri-base-gcc.s
[...]


Combined Analysis Report
------------------------
                                Port pressure in cycles
    | 0  - 0DV | 1  |  2   |  3   |  4   | 5  -  5D  | 6   -  6D |  7   || CP | LCD|
-------------------------------------------------------------------------------------
560 |          |    |      |      |      |           |           |      ||    |    |   .L41:
561 | 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      ||    |    |   lsl w2, w13, 3
562 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z16.d, p0/z, [x11]
563 | 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      ||    |    |   add x18, sp, 160
564 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z18.d, p0/z, [x11, #-4, mul vl]
565 |          |    |      | 0.50 | 0.50 |           |           |      ||    |    |   sxtw x2, w2
566 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z19.d, p0/z, [x11, #-3, mul vl]
  [...]
1367| 1.00     |    |      |      |      | 1.00      | 1.00      |      ||    |    |   st1d z2.d, p0, [x0, #4, mul vl]
1368| 1.00     |    |      |      |      | 1.00      | 1.00      |      ||    |    |   st1d z13.d, p0, [x0, #5, mul vl]
1369|          |    |      | 0.00 | 1.00 |           |           |      ||    |    |   cmp w14, w13
1370|          |    |      |      |      |           |           | 1.00 ||    |    |   bne .L41
      680            500    30     30    118.5 98.5  118.5 98.5   1.0    158   1.0

Loop-Carried Dependencies Analysis Report
-----------------------------------------
1360 |  1.0 | add   w13, w13, 1                    | [1360]
```

In-core analysis (complex-AoS)




```
$ osaca --
[...]


Combined A
----------

    | 0  -
----------
560 |
561 | 0.00
562 |                                                                    11]
563 | 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      ||    |    |   add x18, sp, 160
564 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z18.d, p0/z, [x11, #-4, mul vl]
565 |          |    |      | 0.50 | 0.50 |           |           |      ||    |    |   sxtw x2, w2
566 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      ||    |    |   ld1d z19.d, p0/z, [x11, #-3, mul vl]
  [...]
1367| 1.00     |    |      |      |      | 1.00      | 1.00      |      ||    |    |   st1d z2.d, p0, [x0, #4, mul vl]
1368| 1.00     |    |      |      |      | 1.00      | 1.00      |      ||    |    |   st1d z13.d, p0, [x0, #5, mul vl]
1369|          |    |      | 0.00 | 1.00 |           |           |      ||    |    |   cmp w14, w13
1370|          |    |      |      |      |           |           | 1.00 ||    |    |   bne .L41
      680             500    30     30    118.5 98.5  118.5 98.5   1.0    158   1.0

Loop-Carried Dependencies Analysis Report
-----------------------------------------
1360 |  1.0 | add   w13, w13, 1                      | [1360]
```

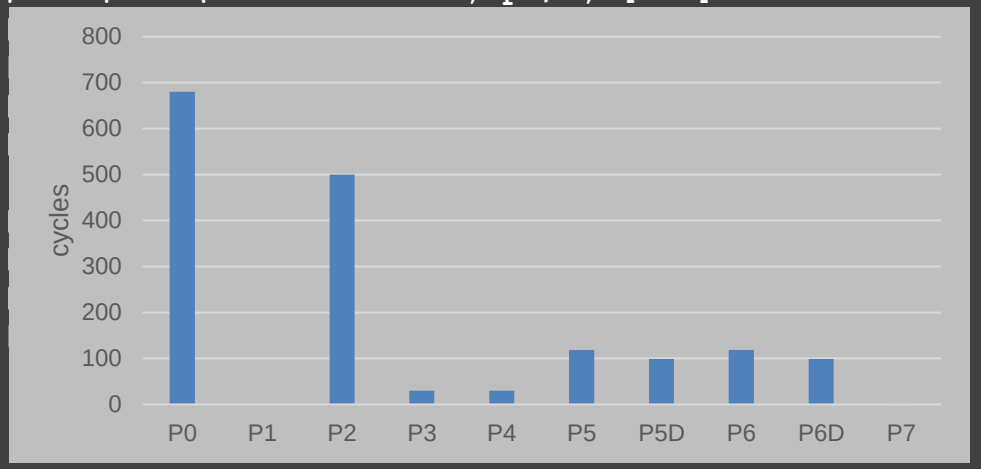
In-core analysis (complex-AoS)



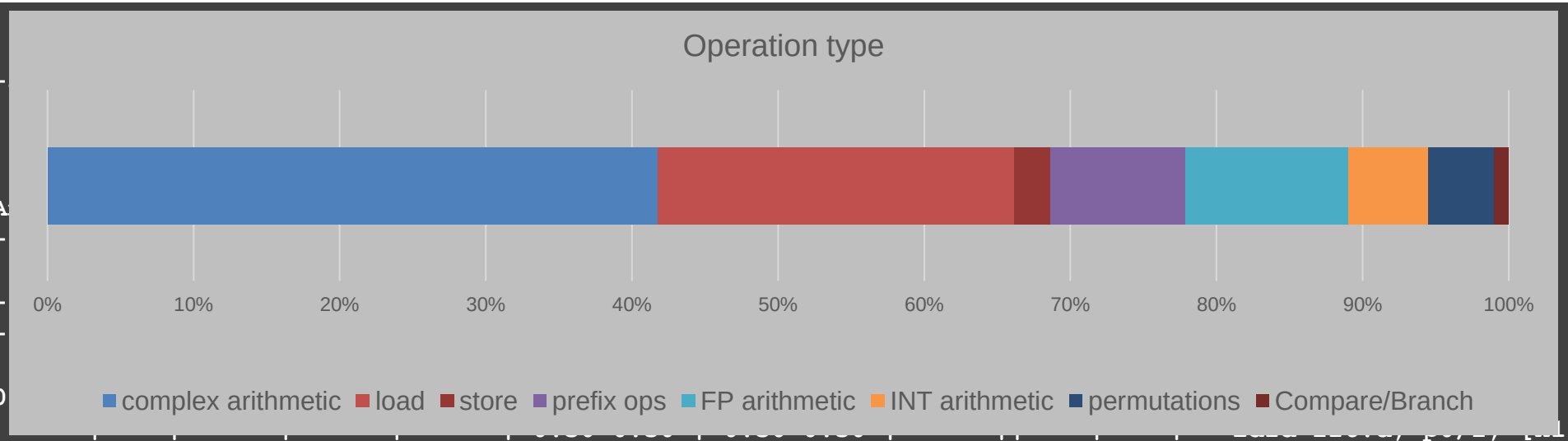

```
$ osaca --
[...]

Combined A
----------

    | 0  -
----------
560 |
561 | 0.00
562 |
563 | 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      |
564 |          |    |      |      |      | 0.50 0.50 | 0.50 0.50 |      |
565 |          |    |      | 0.50 | 0.50 |           |           |      |
5
1
1
1369|          |    |      | 0.00 | 1.00 |           |           |      |
1370|          |    |      |      |      |           |           | 1.00 |
      680              500    30     30    118.5 98.5  118.5 98.5   1.0

Loop-Carried Dependencies Analysis Report
-----------------------------------------
1360 |  1.0 | add   w13, w13, 1                    | [1360]
```

FCMLA Zd, Pg, Zn, Zm, c **2cy on P0, 1cy on P2**

FCADD Zd, Pg, Zn, Zm, c **1cy on P0, 1cy on P2**



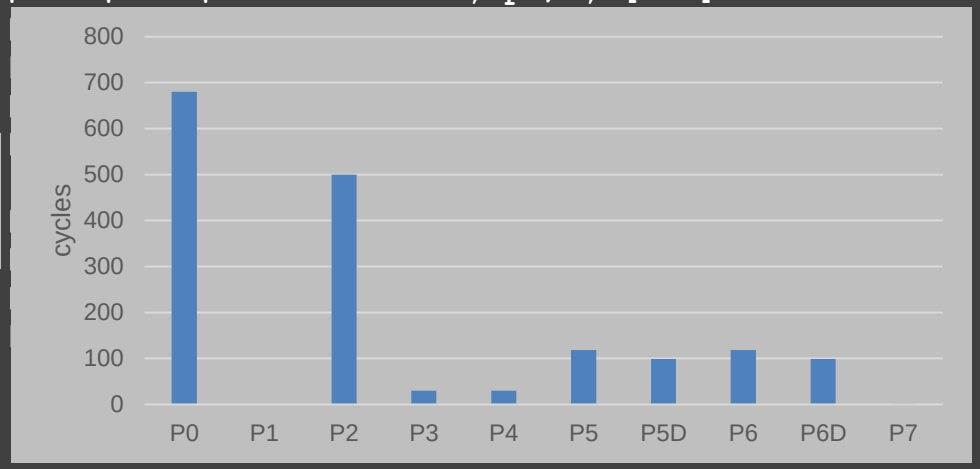
In-core analysis (complex-AoS)




```
$ osaca --
[...]

Combined A
----------

     | 0  -
----------
560 |
561 | 0.00
562 |
563 | 0.00      |     | 0.00 | 0.50 | 0.50 |           |           |      |
564 |           |     |      |      |      | 0.50 0.50 | 0.50 0.50 |      |
565 |           |     |      | 0.50 | 0.50 |           |           |      |
5
1
1
1369|           |     |      | 0.00 | 1.00 |           |           |      |
1370|           |     |      |      |      |           |           | 1.00 |
      680             500    30     30    118.5 98.5  118.5 98.5   1.0

Loop-Carried Dependencies Analysis Report
-----------------------------------------
1360 |  1.0 | add   w13, w13, 1                          | [1360]
```

FCMLA Zd, Pg, Zn, Zm, c **2cy on P0, 1cy on P2**

FCADD Zd, Pg, Zn, Zm, c **1cy on P0, 1cy on P2**



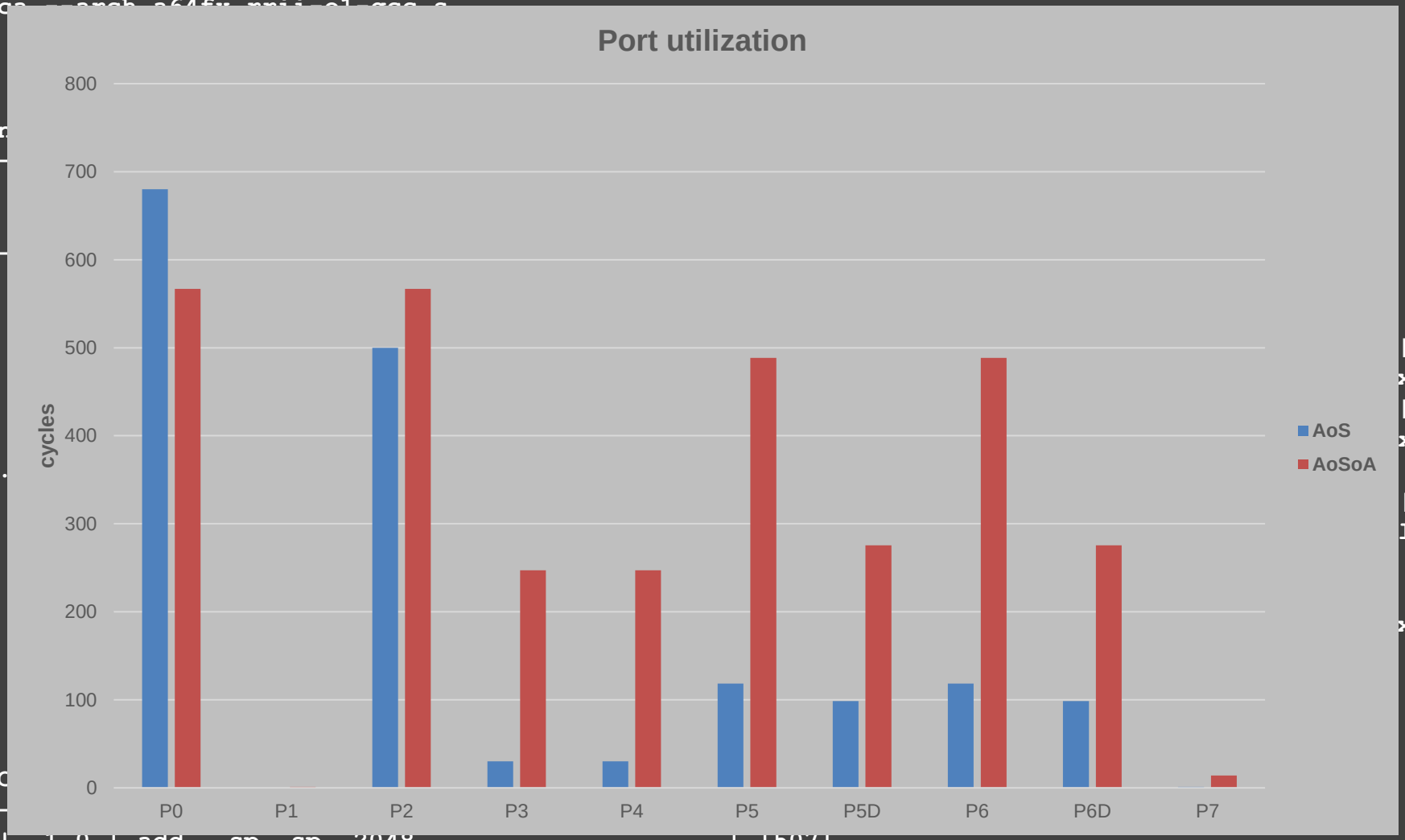
FMLA Zd, Pg, Zn, Zm **1cy on P0 OR P2**

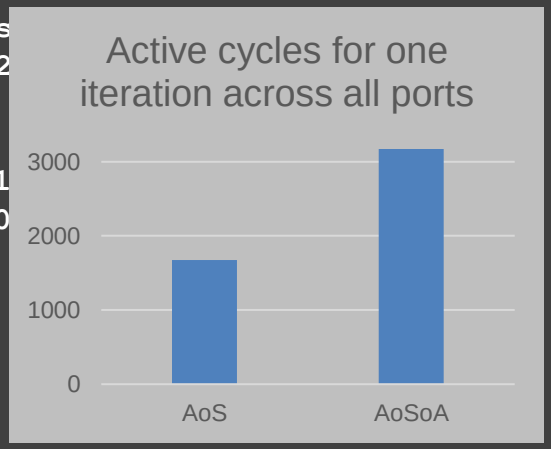
FADD Zd, Pg, Zn, Zm **1cy on P0 OR P2**

In-core analysis (complex-AoSoA)

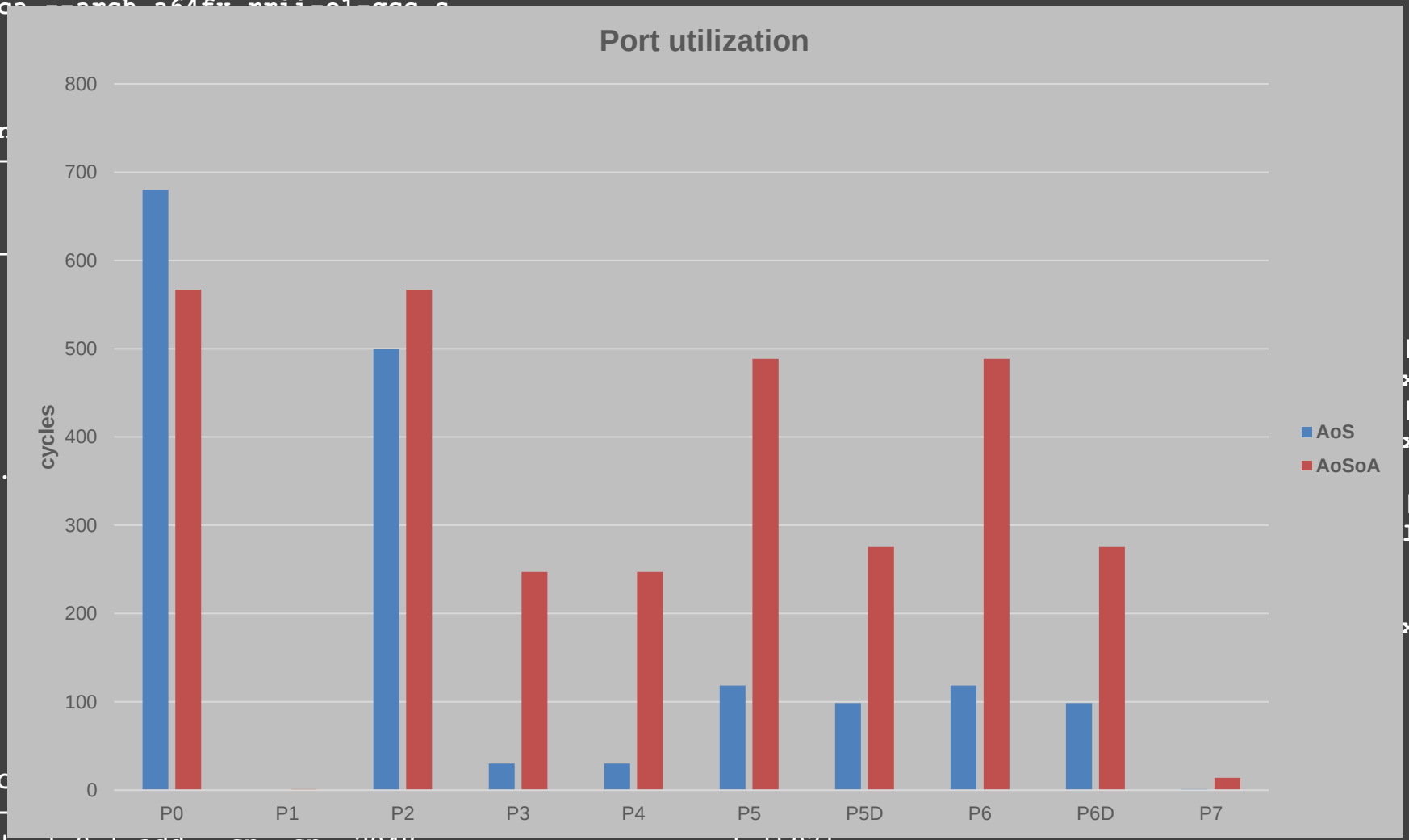
```
$ osaca --arch a64fx rrii-o1-gcc.s
[...]


Combined Analysis Report
------------------------
                                Port pressure in cycles
    | 0  - 0DV | 1  |  2   |  3   |  4   | 5  -  5D  | 6   -  6D |  7   || CP | LCD|
-------------------------------------------------------------------------------------
433 |          |    |      |      |      |           |           |      ||    |    |   .L66:
434 |          |    |      | 2.50 | 2.50 |           |           |      ||    |    |   madd x0, x1, x0, x19
435 | 1.00     |    |      |      |      | 0.50      | 0.50      |      ||    |    |   str x0, [sp, 1896]
436 | 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      ||    |    |   add x1, x1, x0
437 | 1.00     |    |      |      |      | 0.50      | 0.50      |      ||    |    |   str x1, [sp, 1936]
438 |          |    |      | 0.50 | 0.50 |           |           |      ||    |    |   cmp x0, x1
  [...]
2803|          |    |      |      |      | 0.00 0.00 | 1.00 1.00 |      ||    |    |   ldr x0, [sp, 1784]
2804|          |    |      |      |      | 0.00      | 1.00      |      ||    |    |   prfd pldl2strm, p0, [x0]
2805|          |    |      |      |      |           |           | 1.00 ||    |    |   b .L64
2806|          |    |      |      |      |           |           |      ||    |    |   .L38:
2807| 0.00     |    | 0.00 | 0.50 | 0.50 |           |           |      ||    |    |   add x1, x1, 1
2808| 0.00     |    | 0.00 | 0.00 | 1.00 |           |           |      ||    |    |   mov x19, 0
2809|          |    |      |      |      |           |           | 1.00 ||    |    |   b .L66
      567       1.0   567    247    247   488.5 275.5 488.5 275.5  14      92  1.0

Loop-Carried Dependencies Analysis Report
-----------------------------------------
 507 |  1.0 | add   sp, sp, 2048                    | [507]
```

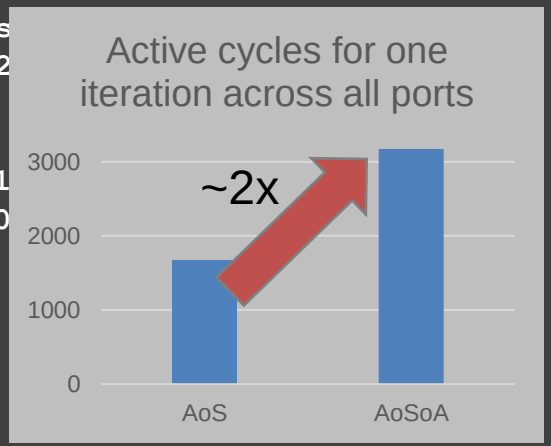
In-core analysis (complex-AoSoA)



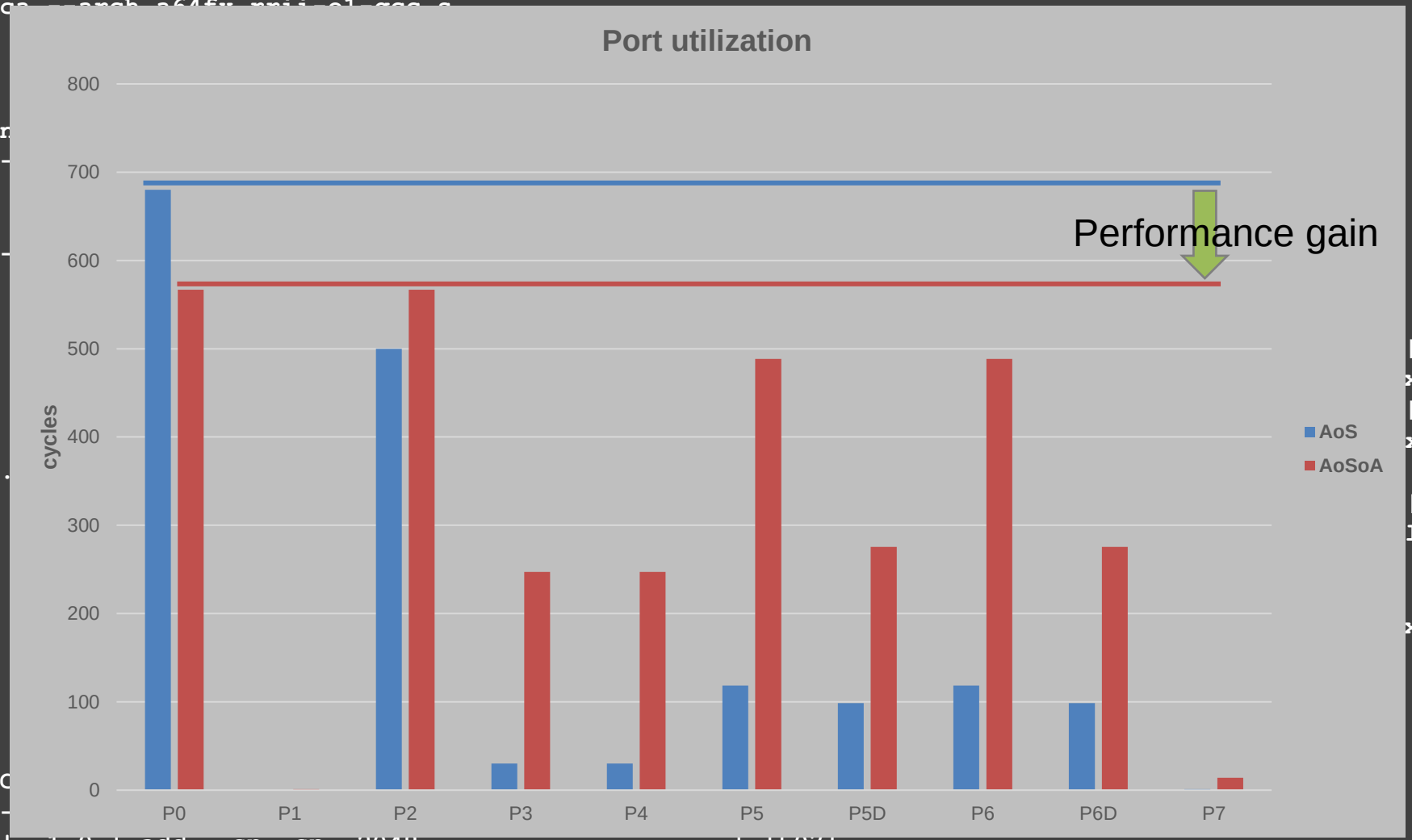


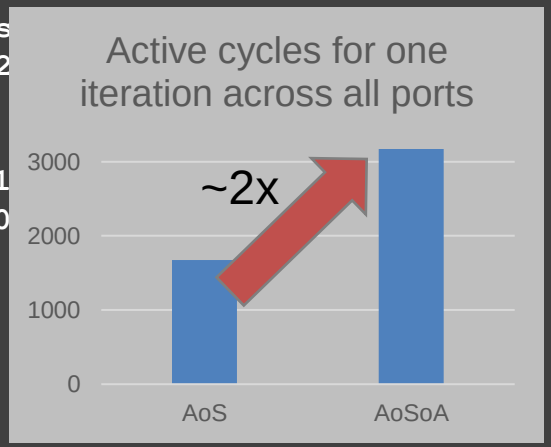
In-core analysis (complex-AoSoA)



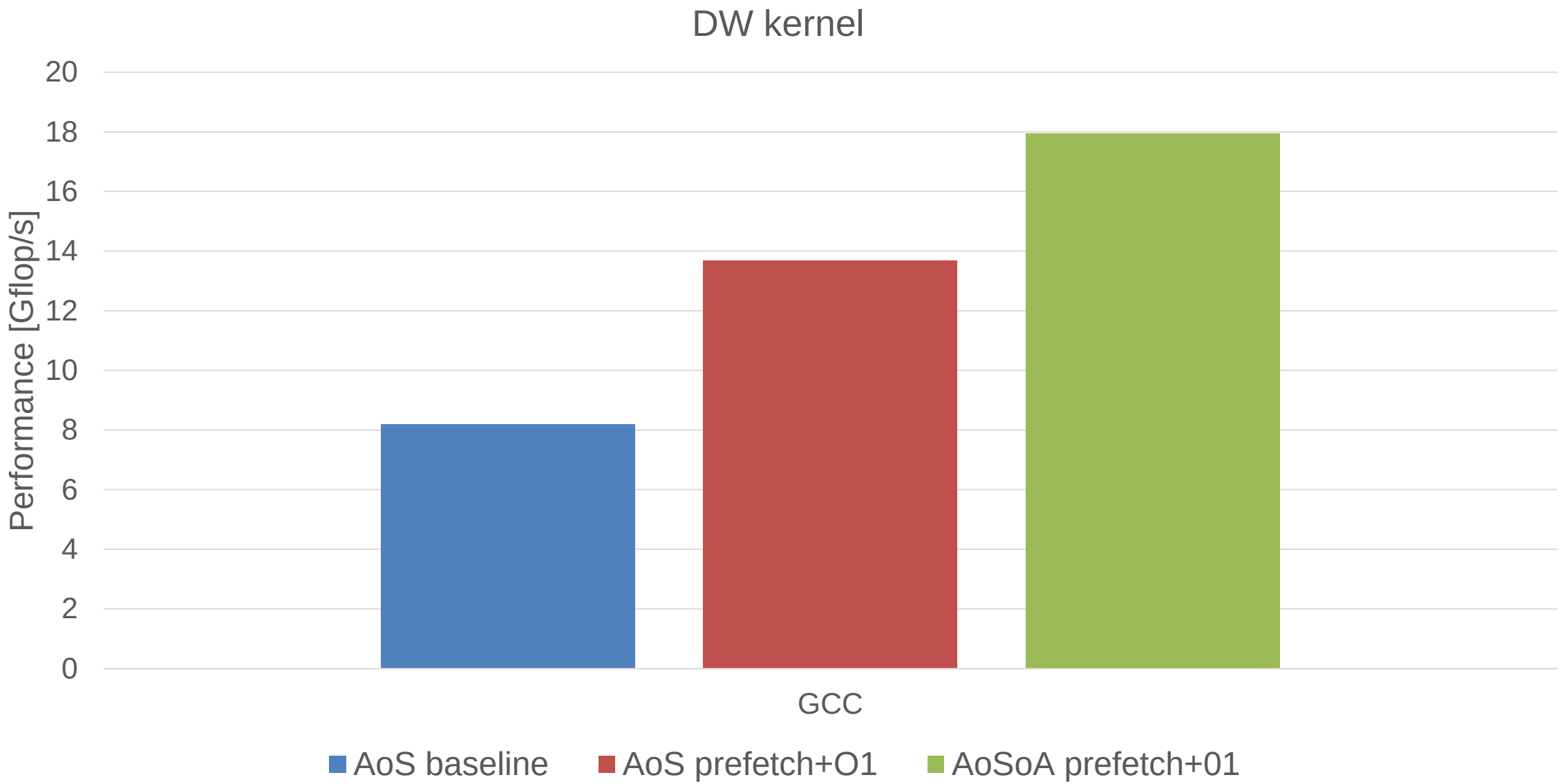


In-core analysis (complex-AoSoA)





DW kernel optimizations



Summary

- static in-core analysis of assembly basic blocks
  → no compilable asm necessary!
- runtime prediction based on throughput, latency, and loop-carried dependency
- supports various Intel, AMD, and ARM µ-architectures
- Python based and available in the Compiler Explorer

Outlook

- Continuously expanding list of supported µ-archs
  (Zen4 and SPR in the pipeline)
- Consideration of µ-arch frontends
  (decode, dispatch, retire, reg files, LSB, ...)
- Support for Intel asm syntax
- Support for IBM Power

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Thank you! Questions?

OSACA

OSACA: https://github.com/RRZE-HPC/osaca

pip: `$ pip install -u osaca`

Check out our other tools:

LIKWID TOOLS

MachineState

ClusterCockpit

KERNCRAFT

INSPECT