NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Programming Techniques for Supercomputers Tutorial

Erlangen National High Performance Computing Center

Department of Computer Science

FAU Erlangen-Nürnberg

Sommersemester 2025

Assignment 2 – Task 1

Peak Performance

- Clock frequency of 2.0 GHz
- 52 cores
- AVX-512 instruction set (512-bit wide SIMD registers); each core is capable of retiring two full-width double-precision FMA instructions per cycle

$$P_{peak} = n_{cores} * n_{super}^{FP} * n_{FMA} * n_{SIMD} * f$$

Assignment 2 – Task 1

Peak Performance

- Clock frequency of 2.0 GHz
- 52 cores
- AVX-512 instruction set (512-bit wide SIMD registers); each core is capable of retiring two full-width double-precision FMA instructions per cycle

$$P_{peak} = n_{cores} * n_{super}^{FP} * n_{FMA} * n_{SIMD} * f$$

When in doubt, think **units**!

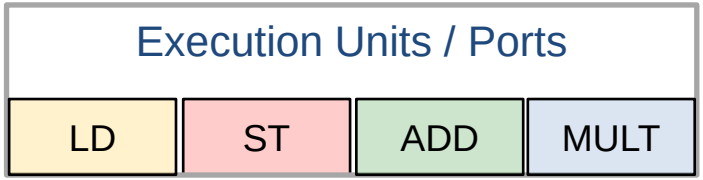
$$P_{peak} = 52 * 2 \left[\frac{instr}{cy}\right] * 2 * 8 \left[\frac{flops}{instr}\right] * 2 \left[\frac{Gcy}{s}\right]$$

$$P_{peak} = 52 * 2 \left[\frac{\cancel{instr}}{\cancel{cy}}\right] * 2 * 8 \left[\frac{flops}{\cancel{instr}}\right] * 2 \left[\frac{\cancel{Gcy}}{s}\right] \approx 3.328 \left[\frac{Tflops}{s}\right]$$

Assignment 2 – Task 2

Pipelines

```
double a[...],b[...];
double s=0.0, t=1.234;
// a[] and b[] contain sensible data
for(int i=0; i<N; ++i) {
  s += a[i]*a[i];
  b[i] *= t;
}
```

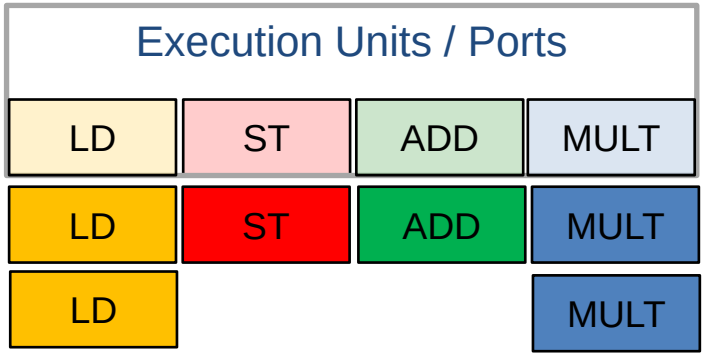


- ADD pipeline depth of 5 cy, MULT pipeline depth of 8 cy
- Capability of executing 1 ADD, 1 MULT, 1 LOAD, and 1 STORE instruction per cycle (no FMA)
- Overall instruction throughput limit of 4 instructions retired per cycle
- Register set of 16 floating-point registers and 16 integer registers
- No SIMD capability

Assignment 2 – Task 2

Pipelines a) No MVE

```
double a[...],b[...];
double s=0.0, t=1.234;
// a[] and b[] contain sensible data
for(int i=0; i<N; ++i) {
  s += a[i]*a[i];
  b[i] *= t;
}
```

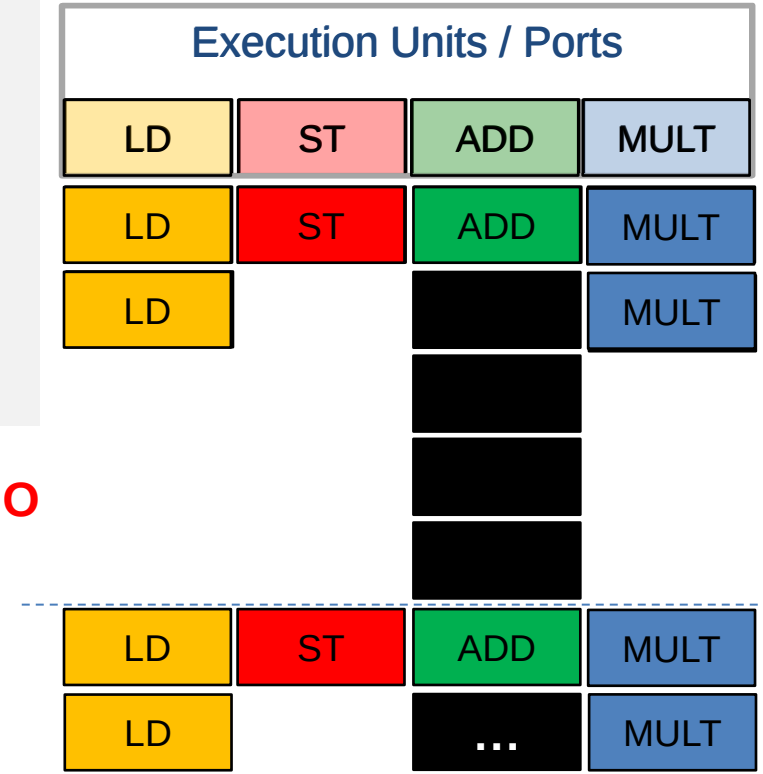


- Does this mean we can do 1 iter. per 2 cy..? **NO**

Assignment 2 – Task 2

Pipelines a) No MVE

```
double a[...],b[...];
double s=0.0, t=1.234;
// a[] and b[] contain sensible data
for(int i=0; i<N; ++i) {
  s += a[i]*a[i];
  b[i] *= t;
}
```



- Does this mean we can do 1 iter. per 2 cy..? **NO**

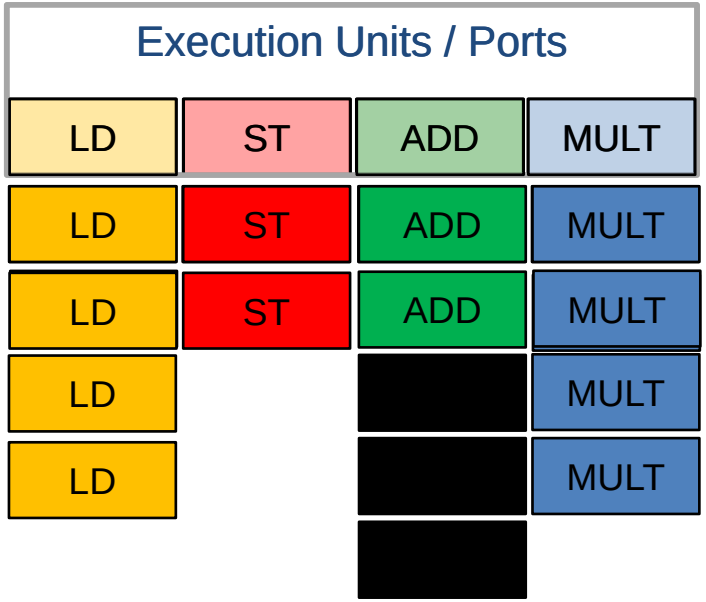
$$P_{max} = \frac{1[iter]}{5[cy]} = \frac{3[flops]}{5[cy]}$$

Assignment 2 – Task 2

Pipelines b) With MVE

```
double a[...],b[...];
double s=0.0, t=1.234;
double s0=0.0, s1=0.0;
// a[] and b[] contain sensible data
for(int i=0; i<N; i=i+2) {
  s0 += a[i]*a[i];
  s1 += a[i+1]*a[i+1];
  b[i] *= t;
  b[i+1] *= t;
}
// remainder loop
s = s0+s1;
```

2-way MVE



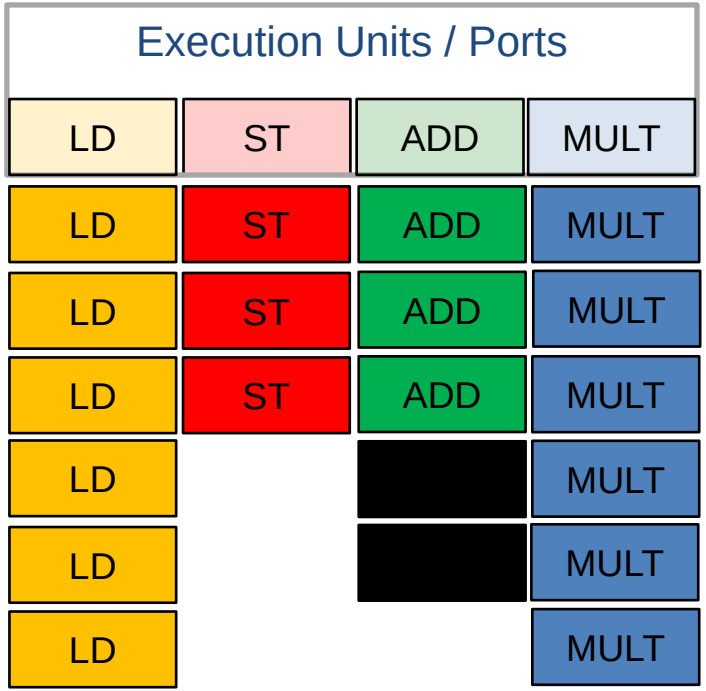
Assignment 2 – Task 2

Pipelines b) With MVE

3-way MVE

```
double a[...],b[...];
double s=0.0, t=1.234;
double s0=0.0, s1=0.0, s2=0.0;
// a[] and b[] contain sensible data
for(int i=0; i<N; i=i+3) {
  s0 += a[i]*a[i];
  s1 += a[i+1]*a[i+1];
  s2 += a[i+2]*a[i+2];
  b[i] *= t;
  b[i+1] *= t;
  b[i+2] *= t;
}
// remainder loop
s = s0+s1+s2;
```

Already sufficient, bottleneck shifts to **LD / MULT**

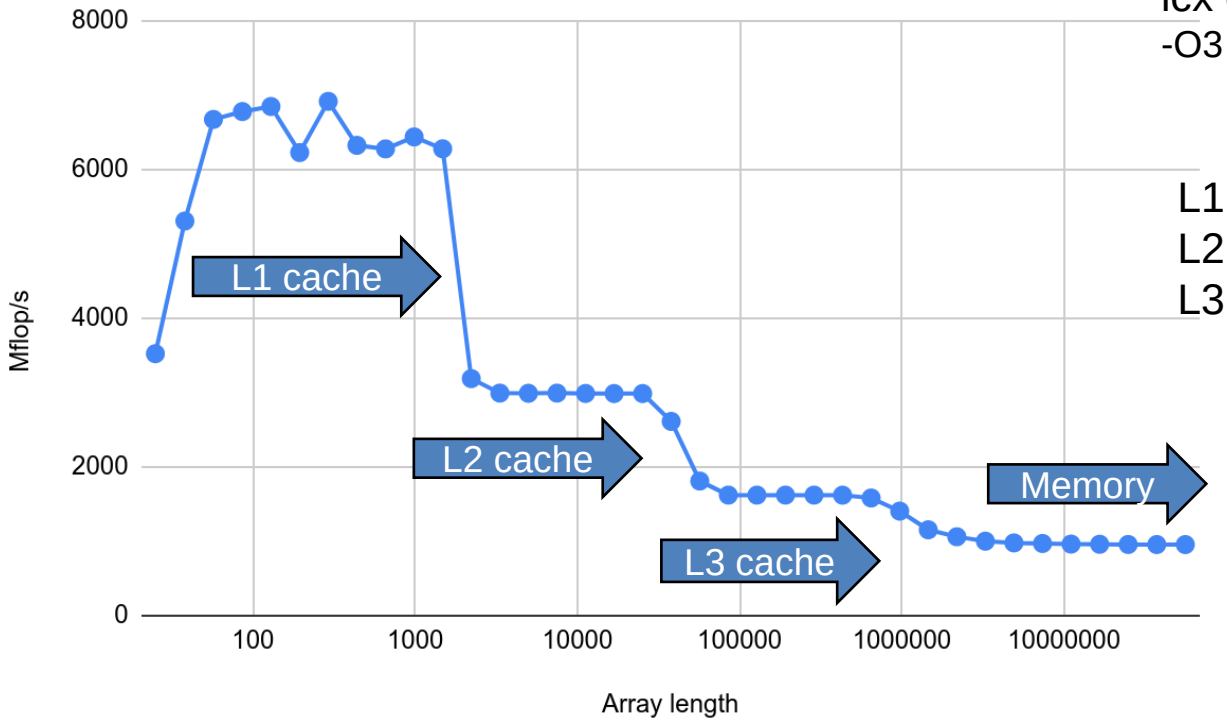


$$P_{max} = \frac{1[iter]}{2[cy]} = \frac{3[flops]}{2[cy]}$$

Assignment 2 – Task 3

Loop Kernel Benchmarking

a) Schönauer Triad: `a[i] = b[i]+c[i]*d[i]`



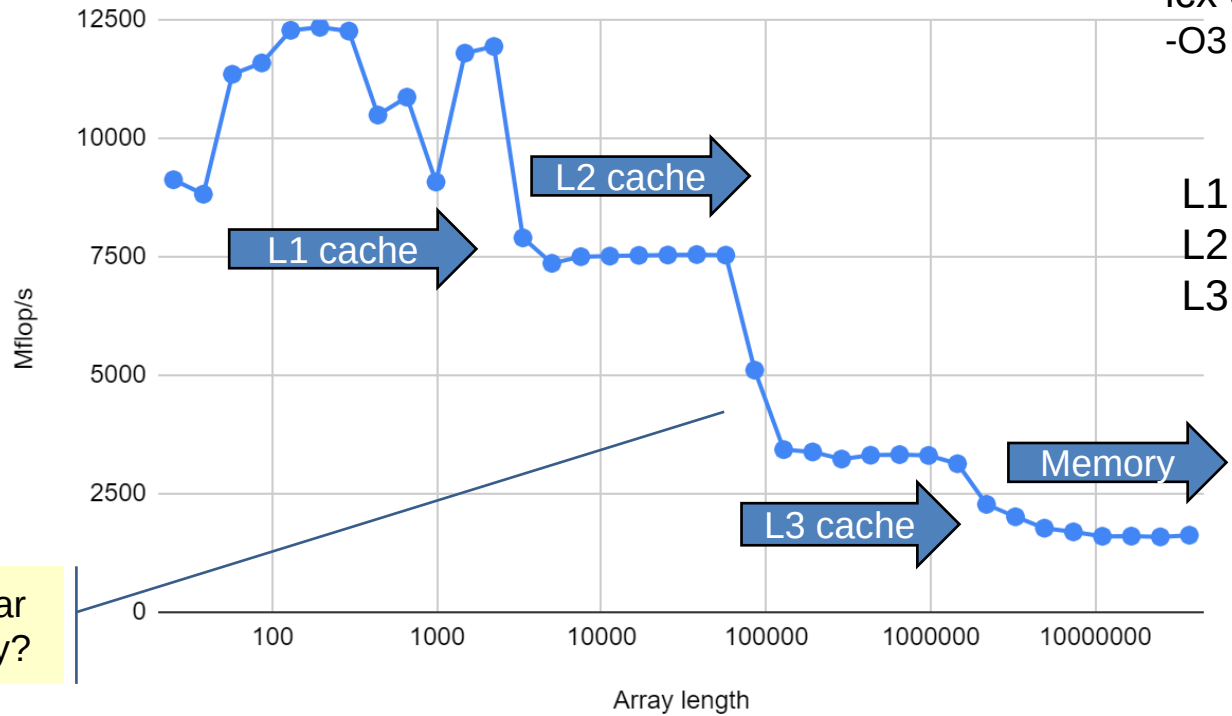
icx compiler
-O3 -xHost -fno-alias

L1 cache: 48 KiB
L2 cache: 1.25 MiB
L3 cache: 54 MiB

Assignment 2 – Task 3

Loop Kernel Benchmarking

a) DAXPY: `a[i] = s*b[i]+a[i]`



icx compiler
-O3 -xHost -fno-alias

L1 cache: 48 KiB
L2 cache: 1.25 MiB
L3 cache: 54 MiB

Drops appear 2x later. Why?