NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Programming Techniques for Supercomputers Tutorial

Erlangen National High Performance Computing Center

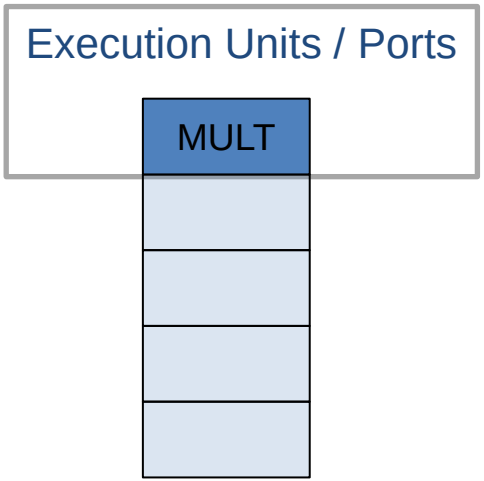
Department of Computer Science

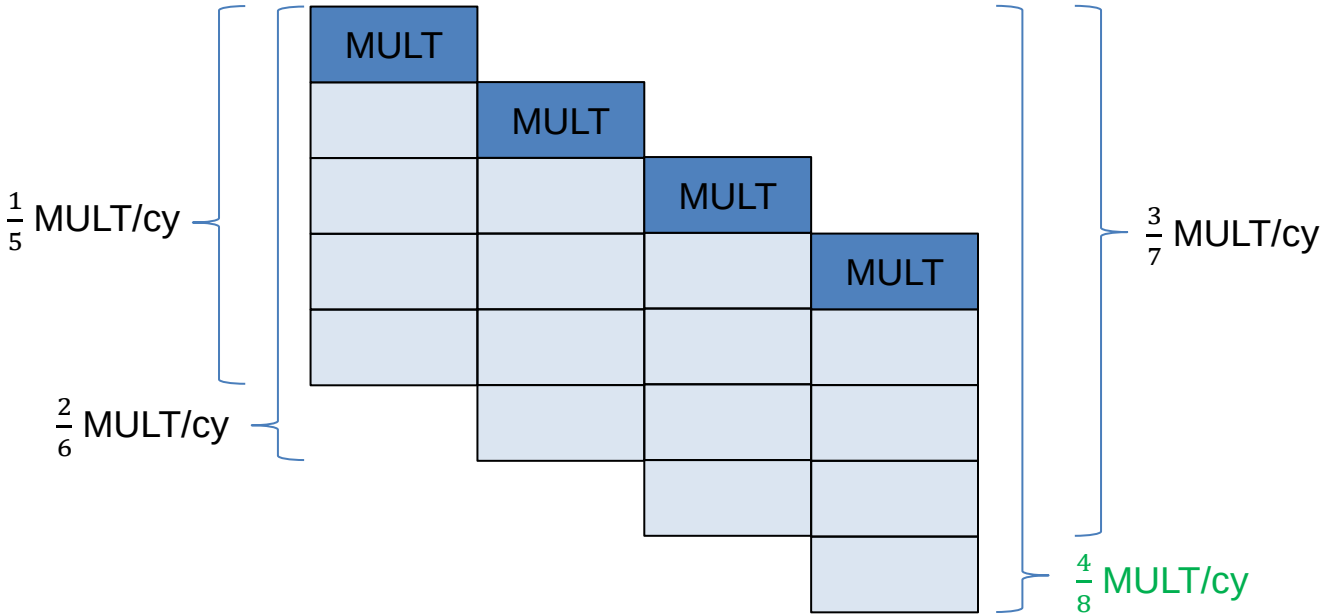
FAU Erlangen-Nürnberg

Sommersemester 2025

Assignment 1 – Task 1

Pipelines





- 5 cy latency
- 1 result per cy

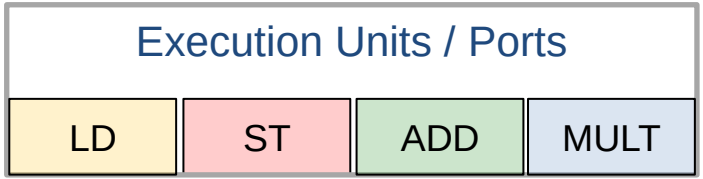
$$\boldsymbol{TP_{pipe}} = \frac{\boldsymbol{N}}{\boldsymbol{N+m-1}} \text{ with } TP_{pipe} = \frac{1}{2} \text{ inst/cy and } m = 5$$

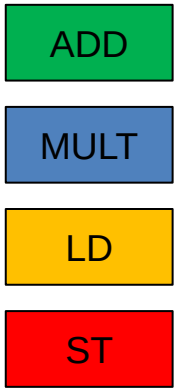
➔ $\boldsymbol{N = 4}$

Assignment 1 – Task 2

More pipelines

```
double a[...];
double s=0.1;
// a[] contains sensible data
for(int i=2; i<N; ++i) {
    a[i] += s * a[i-2];
}
```



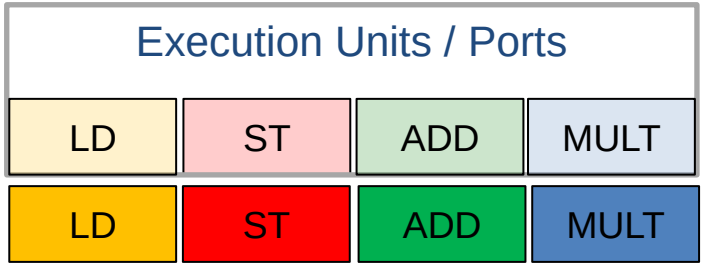


- 1 ADD, 6cy latency
- 1 MULT, 8cy latency
- 1 LOAD
- 1 STORE

Assignment 1 – Task 2

More pipelines

```
double a[...];
double s=0.1;
// a[] contains sensible data
for(int i=2; i<N; ++i) {
    a[i] += s * a[i-2];
}
```



- 1 ADD, 6cy latency
- 1 MULT, 8cy latency
- 1 LOAD
- 1 STORE

Does this mean we can do 1 iter. per cycle (i.e., 2 flops in 1 cy)?

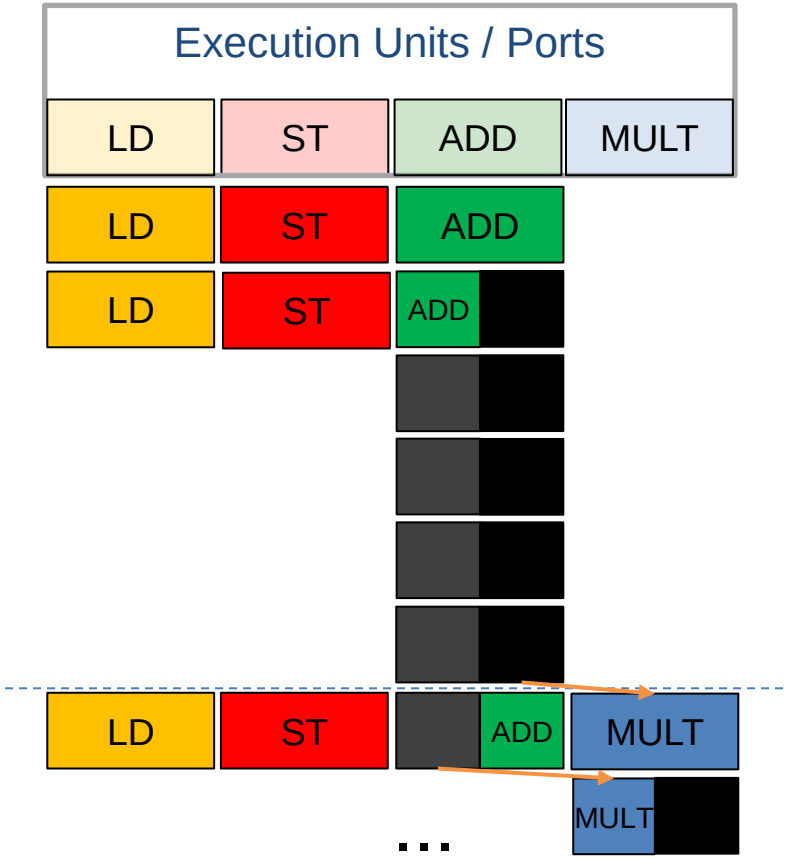
NO

Remember: Pipelining and dependencies from lecture (slide set 3a)

Assignment 1 – Task 2

More pipelines

```
double a[...];
double s=0.1;
// a[] contains sensible data
for(int i=2; i<N; ++i) {
    a[i] += s * a[i-2];
}
```



Bottleneck: ADD/MUL pipeline stall (latency)

➔ P_{max} = 2 iter/14 cy = 4 flops/ 14 cy
= 0.286 flops/cy

Assignment 1 – Task 3

Logarithm in depth

```
sum = 0.;
for(int i=0; i<N; i++) {
  x = (i + 0.5) * delta_x;
  sum = sum + 1.0 / (1.0 + x);
}
```

```
sum = 0.; sum1 = 0.; sum2 = 0.;
for(int i=0; i<N; i+=2) {
  x1 = (i + 0.5) * delta_x;
  x2 = (i+1+0.5) * delta_x;
  sum1 = sum1 + 1.0 / (1.0 + x1);
  sum2 = sum2 + 1.0 / (1.0 + x2);
}
sum = sum1 + sum2;
```

- Bottleneck: DIV with 6cy latency and 4cy reciprocal throughput

- Modulo Variable Expansion (MVE) → **different order of accumulation**
  → DIVs can overlap → hide latency by pipelining
- **partial sums must be reduced to one after the loop**

Assignment 1 – Task 4

Memory bandwidth

```
double x[], y[], a[], b[], c[], d[];

for(i=0; i<N, i++) {
  y[i] = a[i] + b[i];
  x[i] = c[i] + d[i];
}
```

- 36 core CPU
- 160 GB/s memory bandwidth
- 2.4 GHz clock frequency
- 2 ADD instructions per cycle

a) 160 GB/s per 36 cores ➔ $b_s = \frac{160\,\frac{GB}{s}}{2.4\,\frac{cy}{s} \times 36} = 1.85$ B/cy

b) 4 LOADs and 2 STOREs ➔ $6 \times 8\,\frac{B}{it} = 48$ B/it

$$P_{peak,ADD} = 2\,\frac{\text{ADD}}{cy} = 1\,\frac{it}{cy} = 48 \text{ B/cy}$$

$$P = \min(b_s, P_{peak,ADD}) = b_s = \frac{1.85}{48} \approx \frac{1}{26} P_{peak,ADD}$$