Elastic CGRA

Yuanjie Huang, Paolo Ienne, Olivier Temam, Yunji Chen, Chengyong Wu

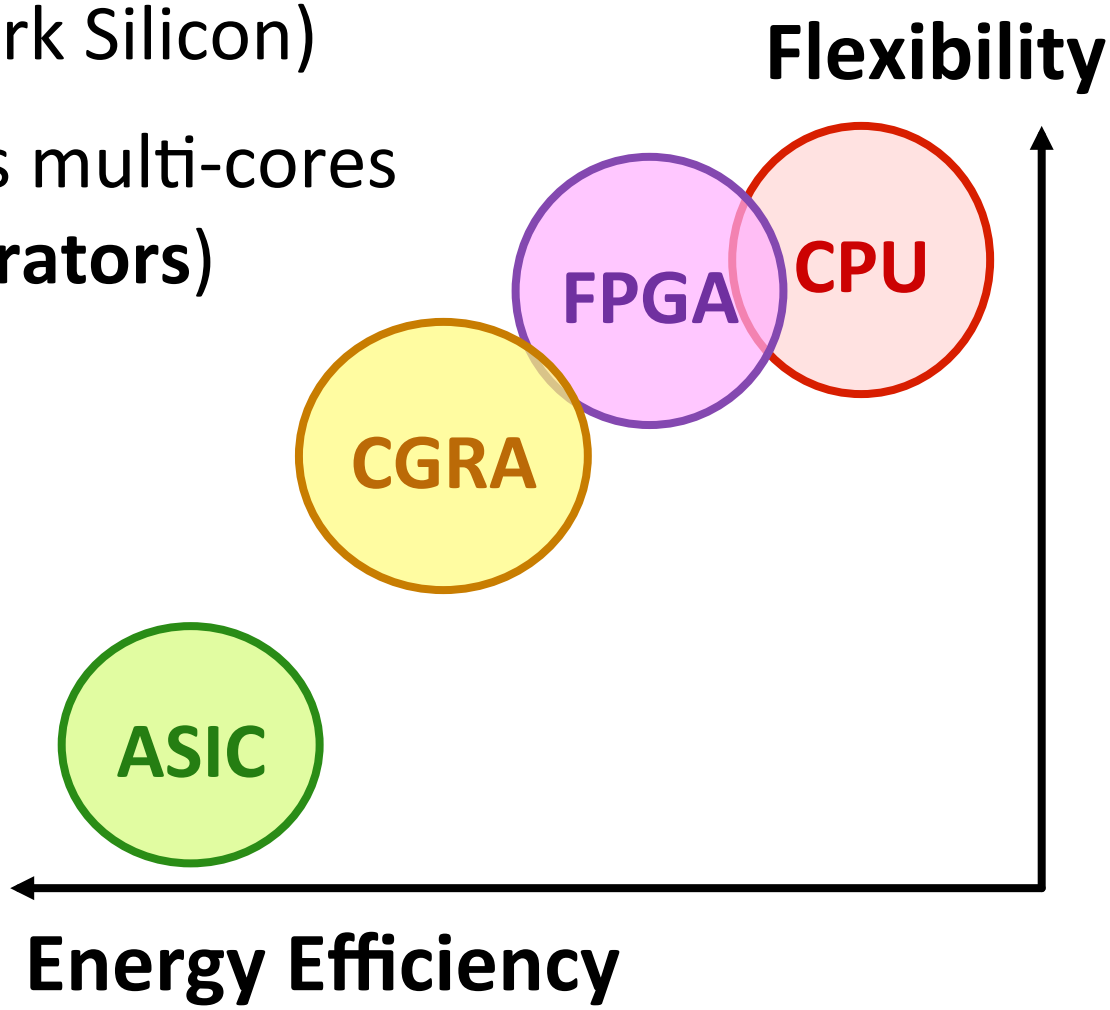
计算机体系结构国家重点实验室
State Key Laboratory of Computer Architecture, ICT, CAS

Inria
INVENTORS FOR THE DIGITAL WORLD

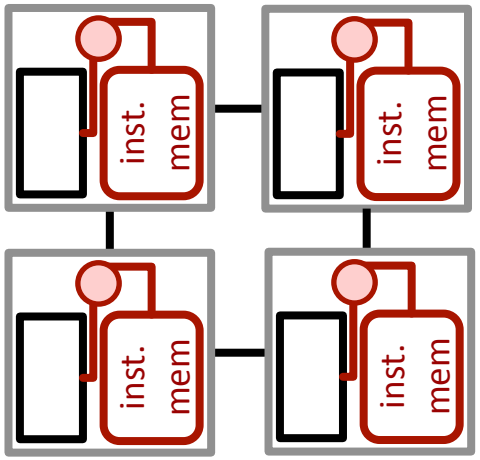
EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Why CGRA?

- Stringent processor energy constraints (Dark Silicon)
- Heterogeneous multi-cores (cores + **accelerators**)



Typical CGRAs Are Statically Scheduled

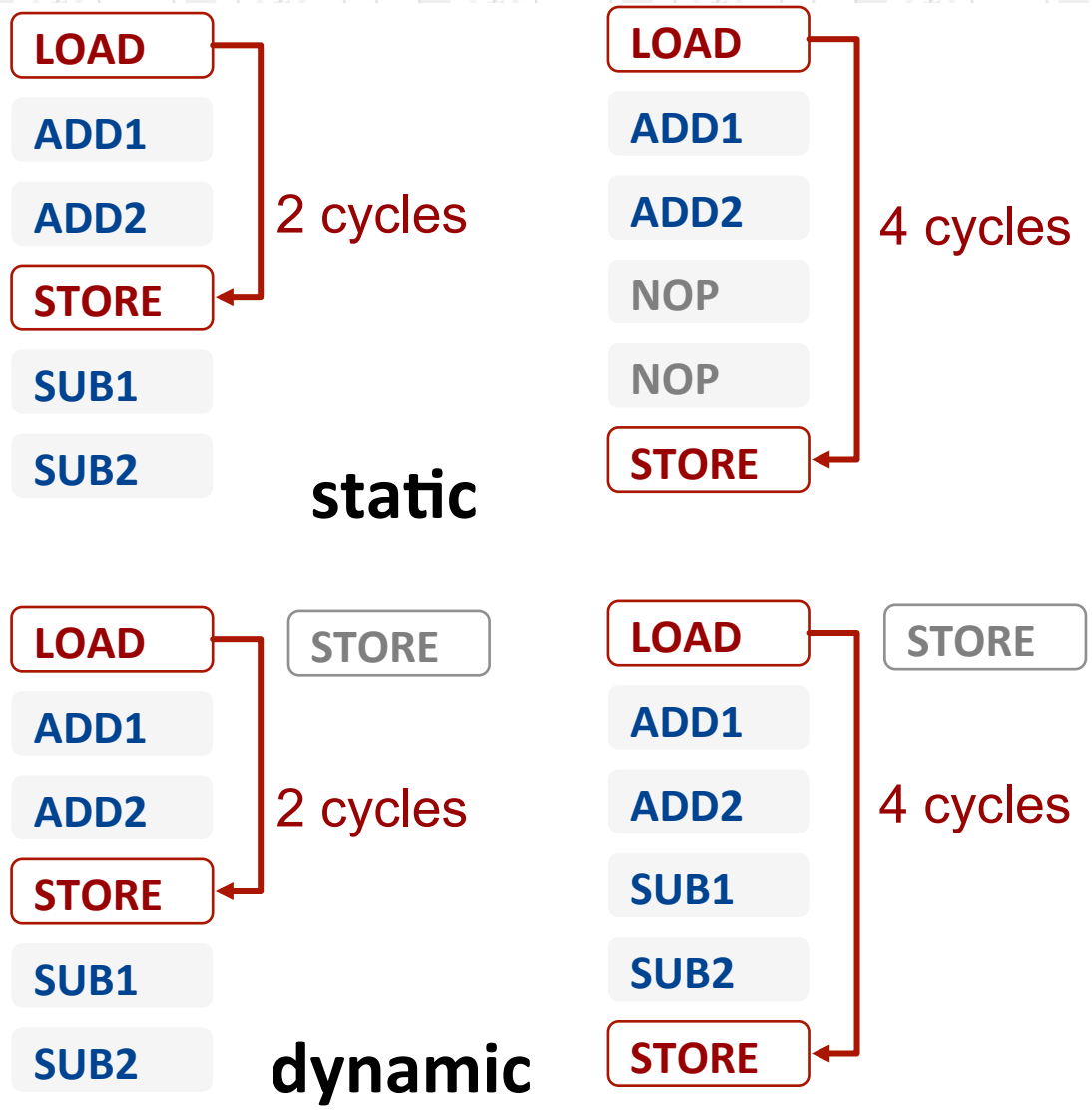


- Each operation & transfer predefined on each cycle
- Similar to VLIW
- E.g., Mei et al.'s ADRES

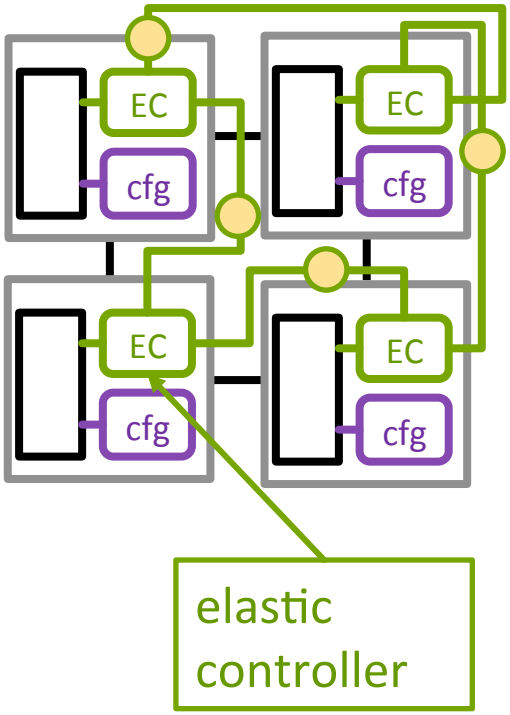
Static vs. Dynamic Scheduling

STORE	LOAD
a[i] =	a[i-1]

b = c + d	ADD1
e = f + g	ADD2
h = i - j	SUB1
k = m - n	SUB2



Dynamically Scheduled Elastic CGRAs



- Dynamic superscalar-like scheduling
- Tolerant to variable latency
- **But very low area/energy overhead**

Outline

- Elastic Circuit

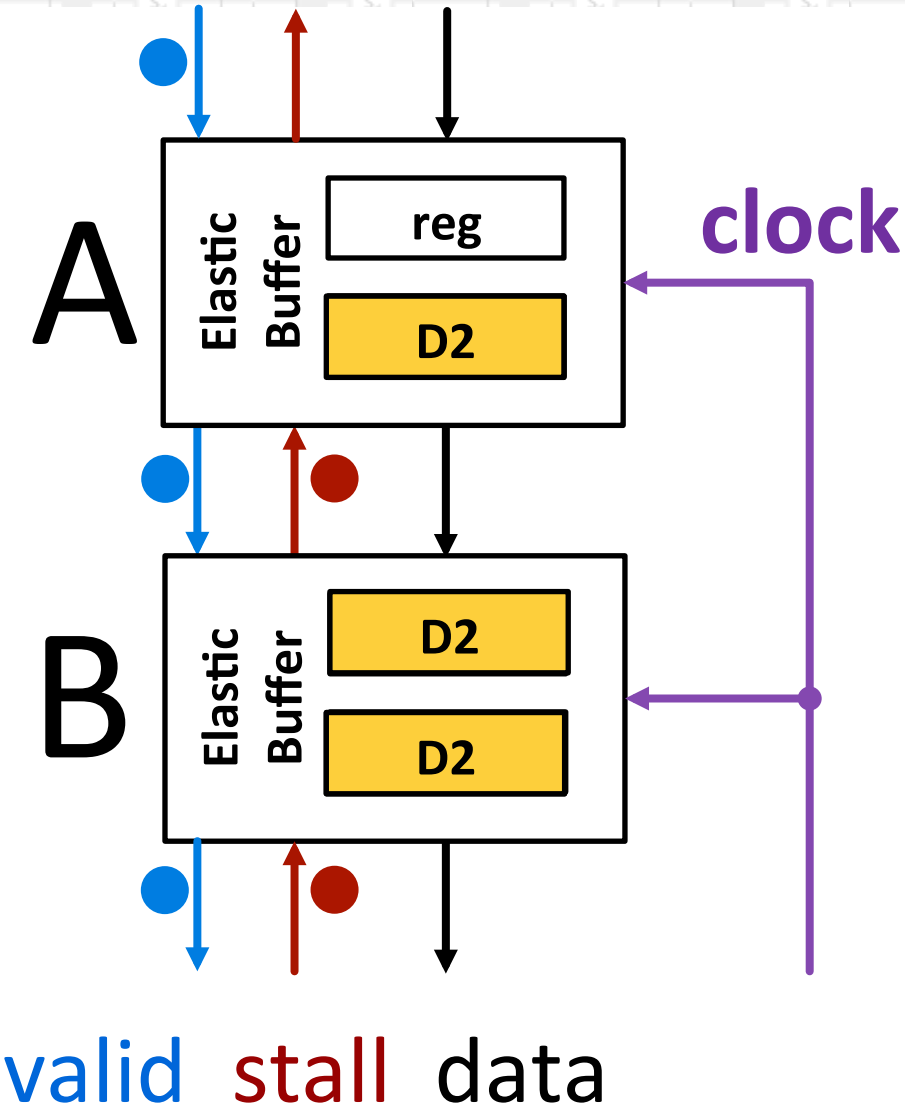
- Elastic CGRA design
- Results
- Future work and conclusion

Elastic Circuit

Jacobson et al . 2002

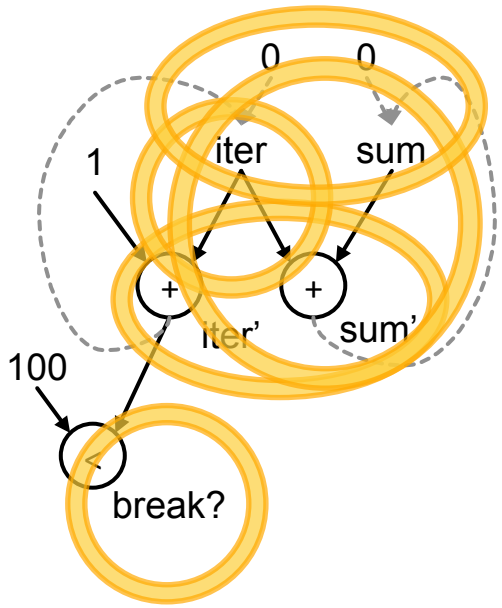
overview

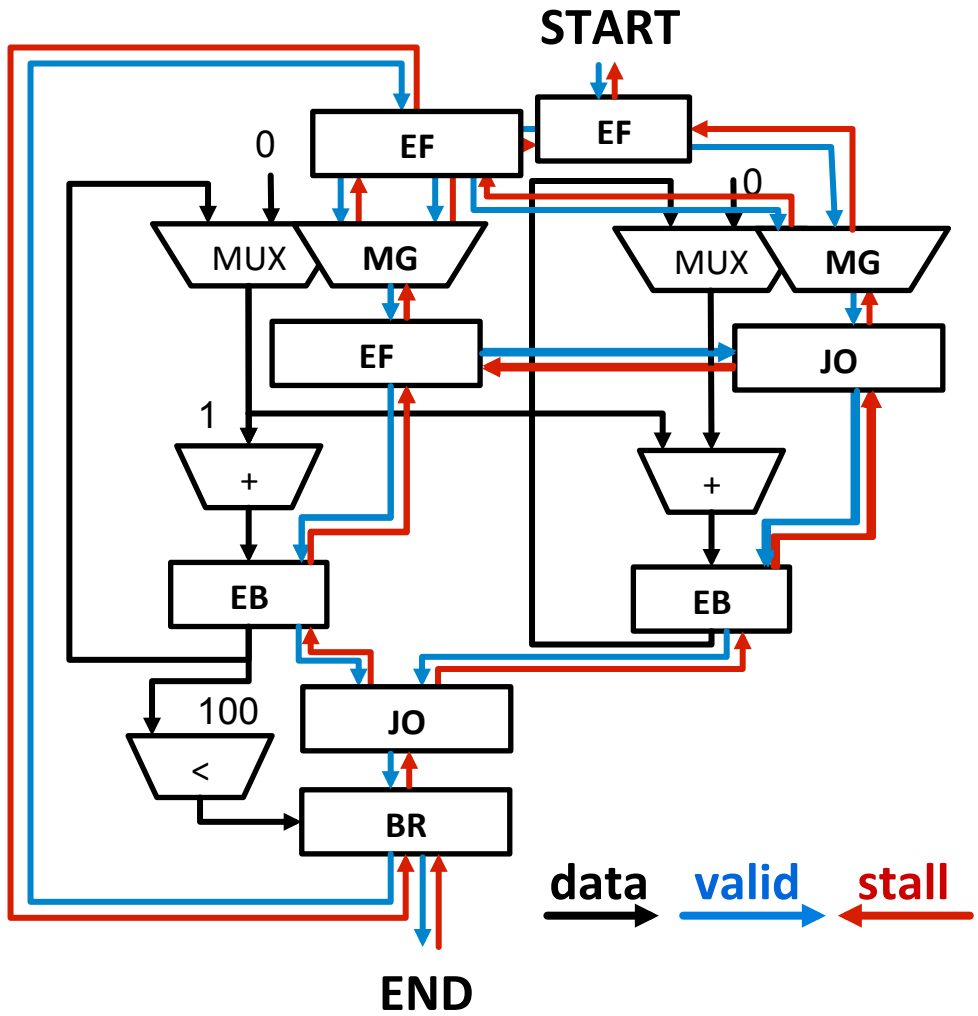
Asynchronous-like behavior using a synchronous design



Transform Code to Elastic Circuit

```
iter=0; sum=0;
do{sum+=iter++;}
while(iter<100);
```

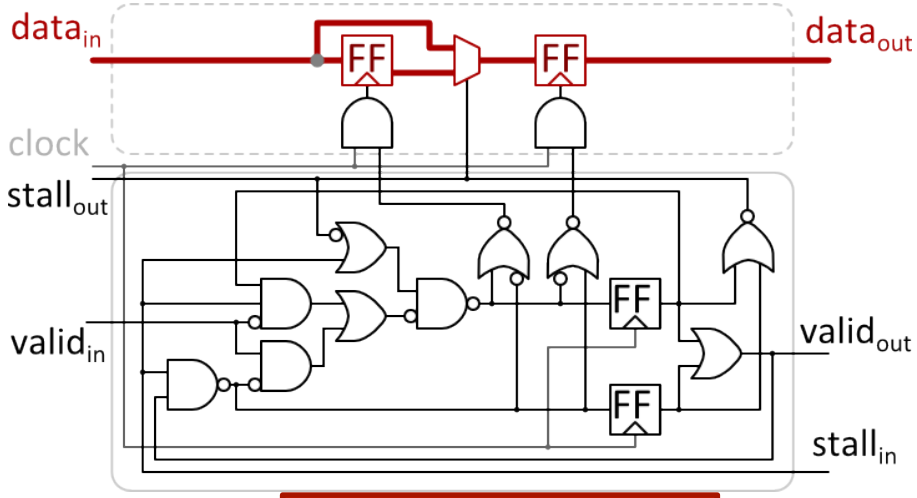




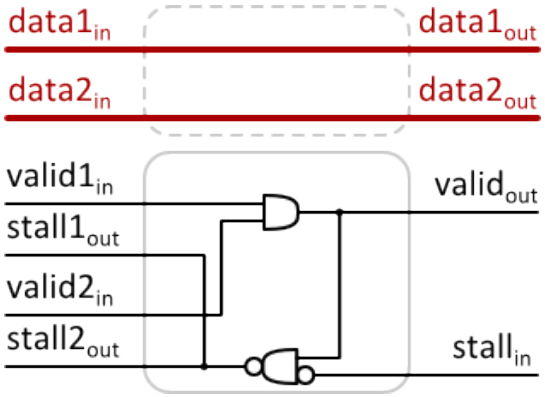
Elastic Circuit Elements

Jacobson et.al .2002
Cortadella et.al .2006

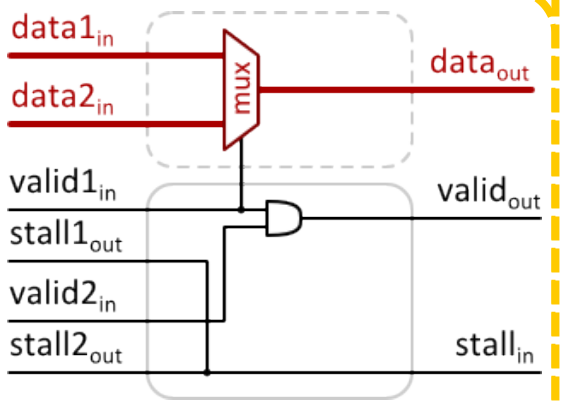
overview



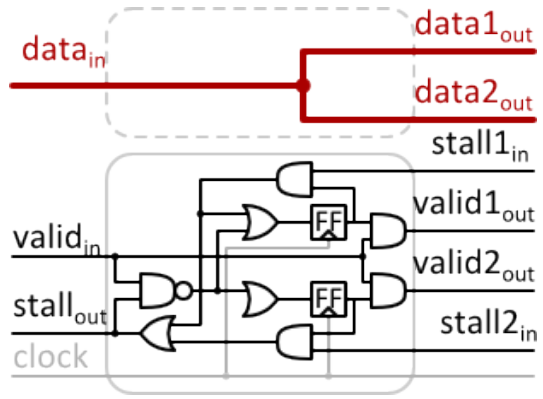
Elastic Buffer (EB)



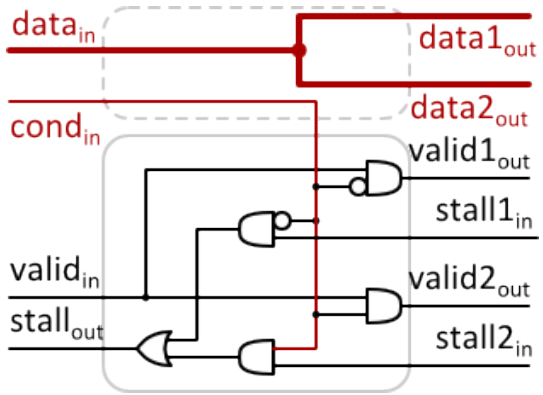
Join (JO)



Merge (MG)



Eager Fork (EF)



Branch (BR)

Outline

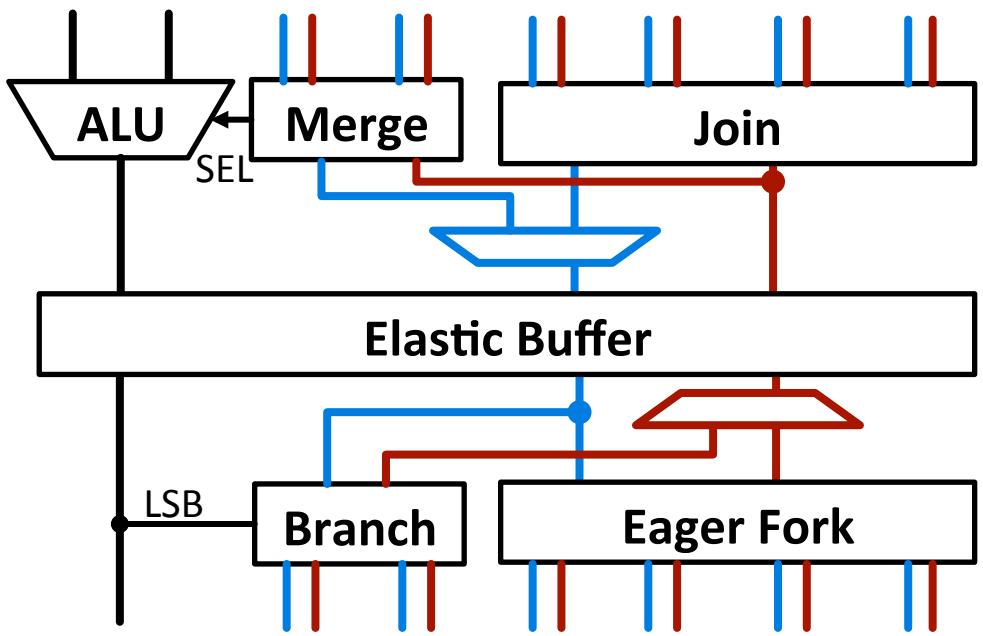
- Elastic Circuit (brief review)
- Elastic CGRA design
- Results
- Future work and conclusion

CGRA: Reconfigurable Unit (RU)

5 Merge data from different control flow

3 Join to sync inputs

1 ALU



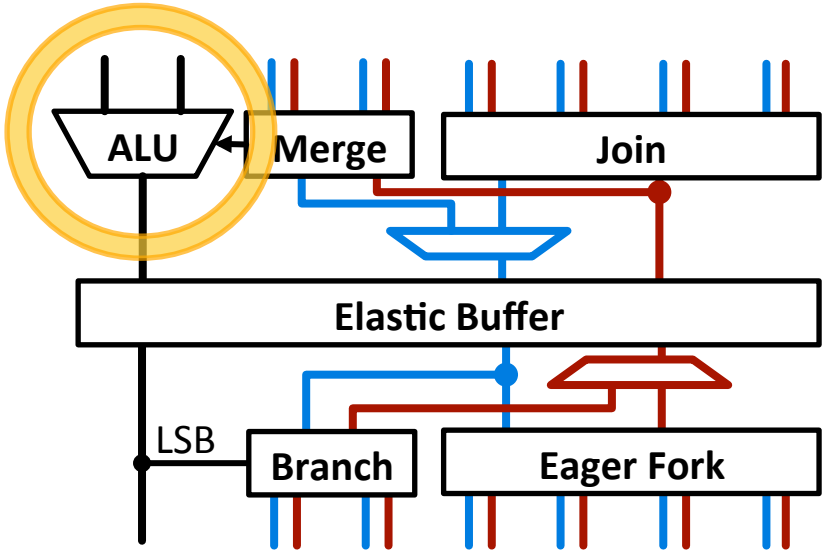
4 Eager Fork to distribute result

2 EB to store result

6 Branch to selectively send signals

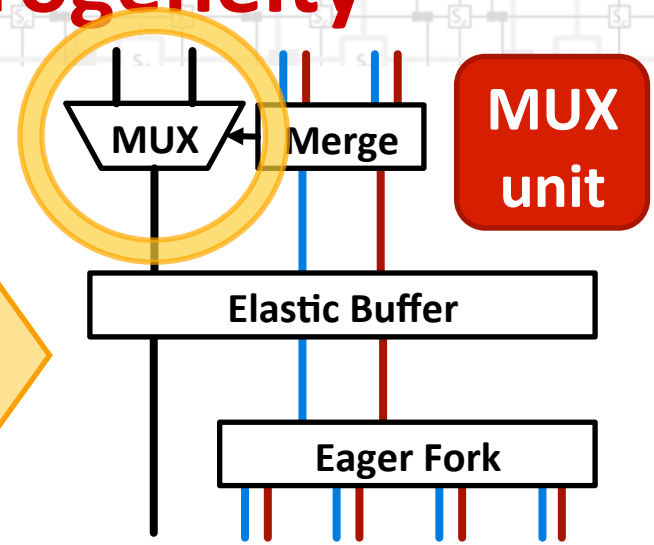
CGRA: Save Area by Heterogeneity

Full ALU is not always necessary



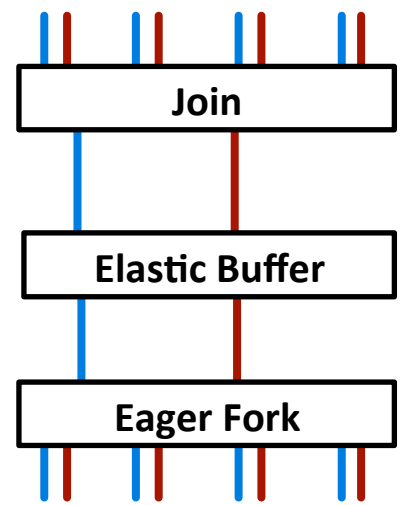
ALU unit

for data merge



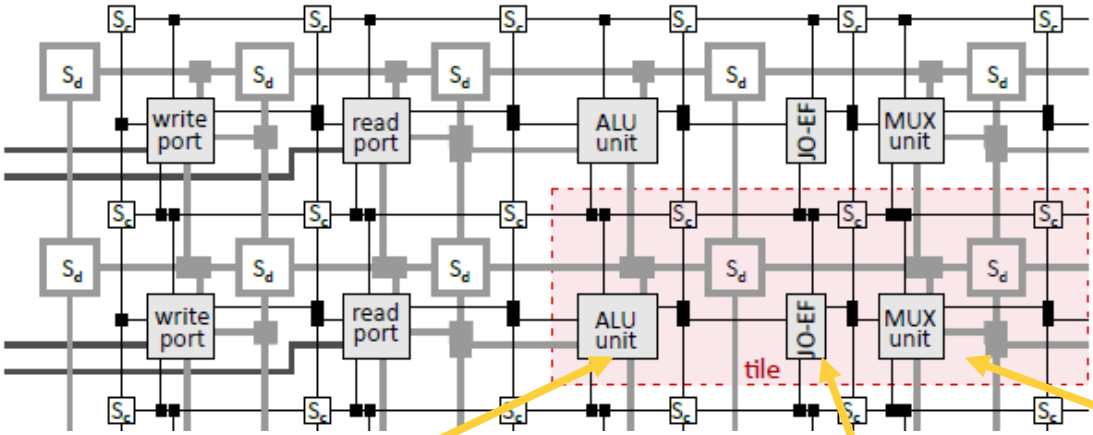
MUX unit

for control sync

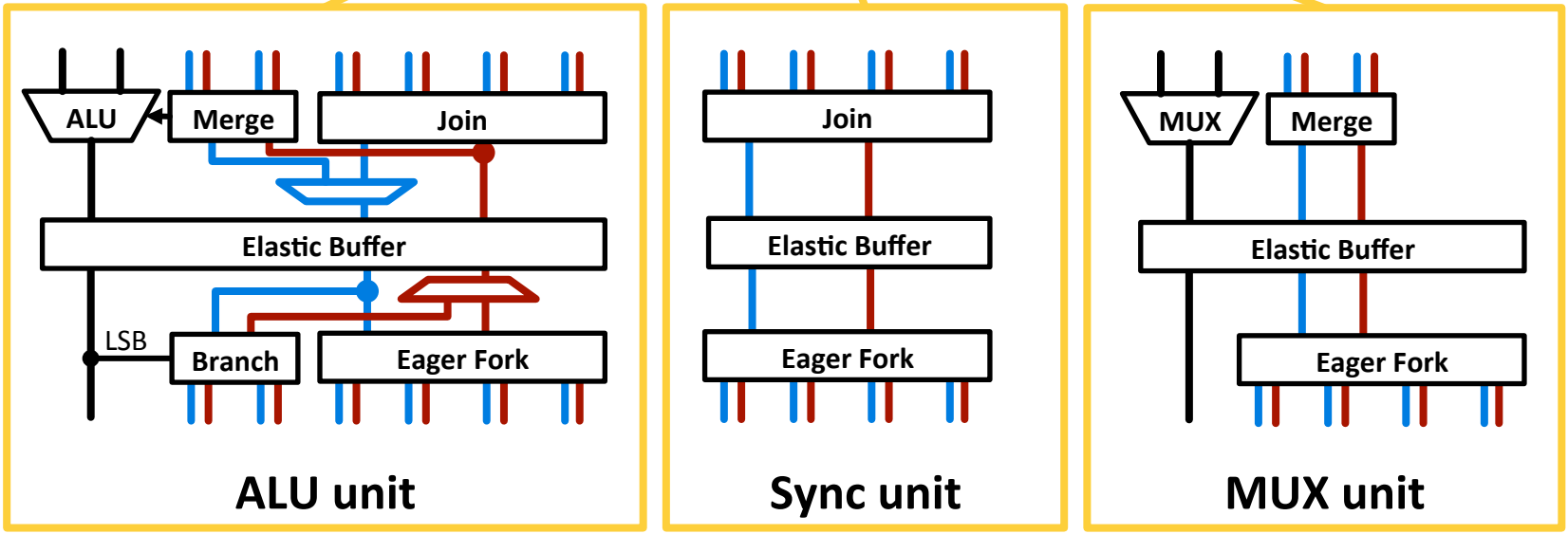


Sync unit

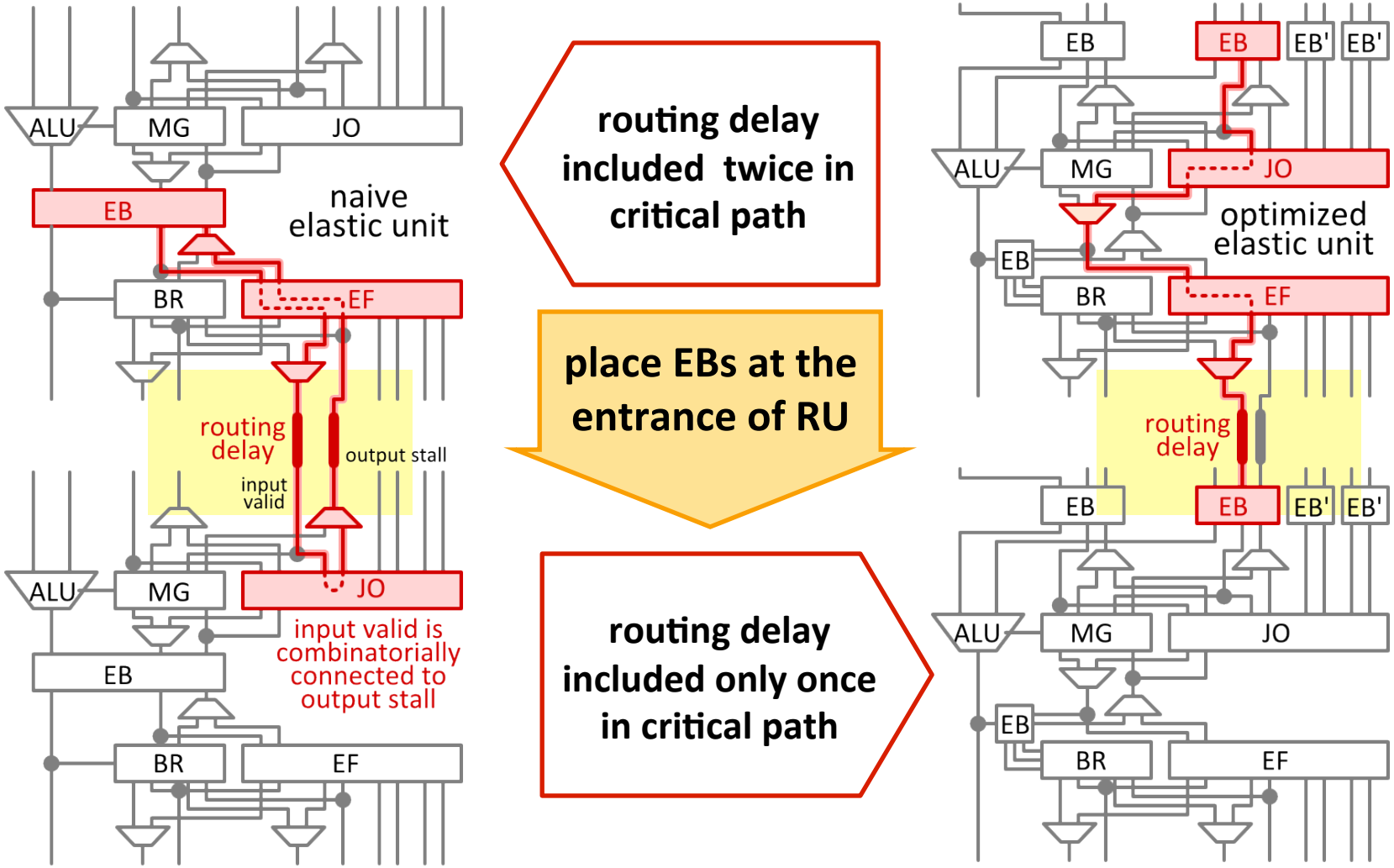
CGRA: Heterogeneous RUs



1:1:1 mix of three types of RUs



CGRA: Optimize RU for Delay



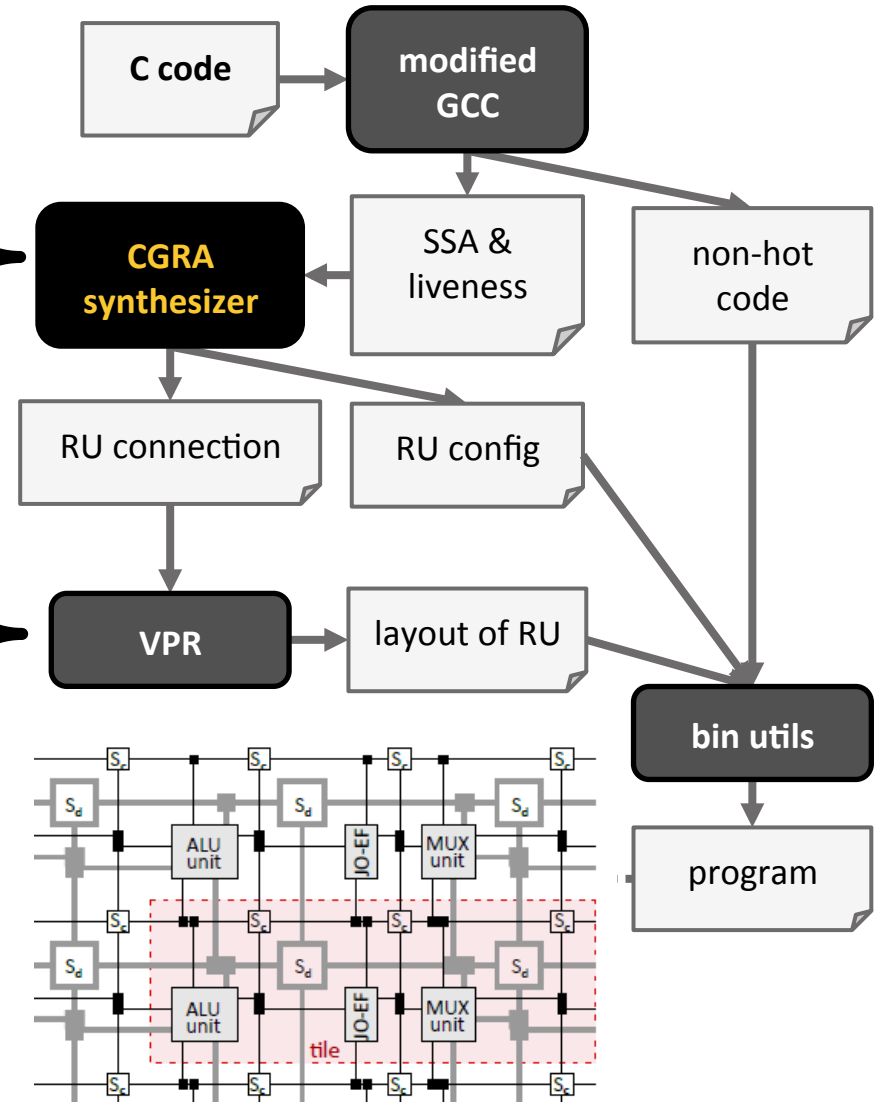
Mapping Code onto Elastic CGRA

static single

Round 1: place RUs with merged nets to a fully-connected architecture

Round 2: route real data connections

Round 3: route real control connections

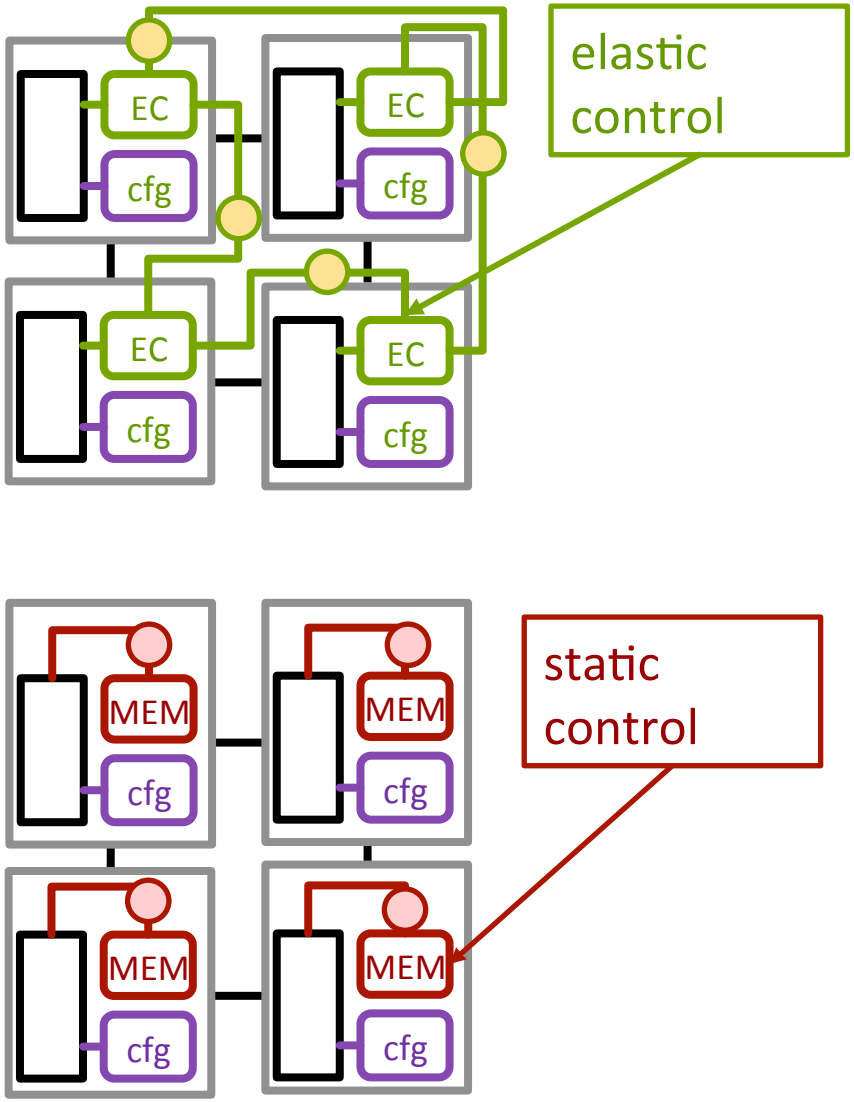


Outline

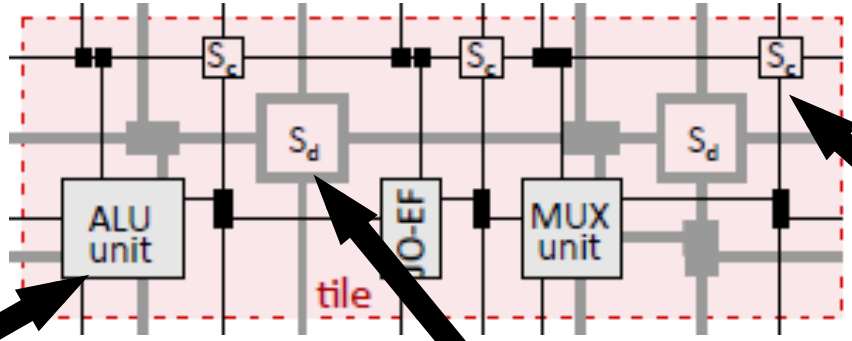
- Elastic Circuit (brief review)
- Elastic CGRA design
- Results
- Future work and conclusion

Comparison Against Static CGRA

- Baseline static CGRA controlled by instruction memory (2bit x 64 inst)
- Same RU placement and data routing
- Memory accesses always complete in one cycle



Layout of Abutting Elastic Tile

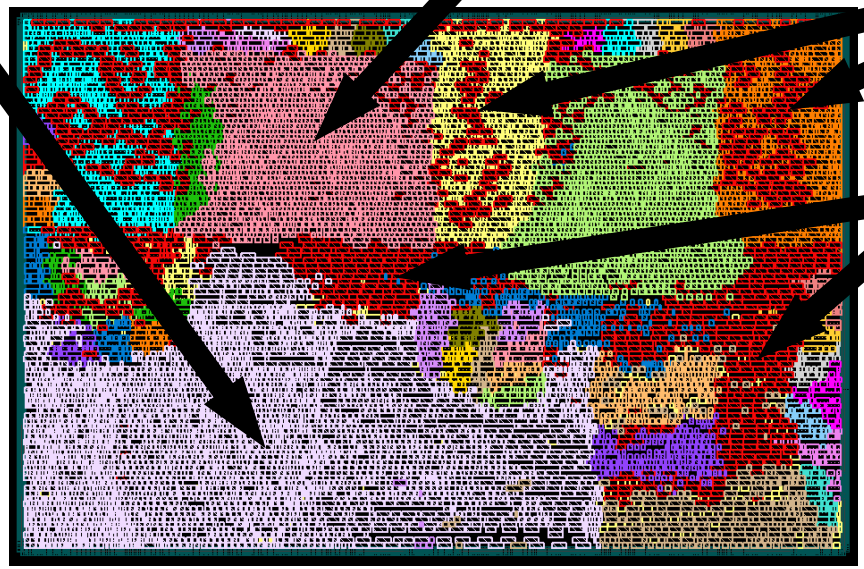


- Layout with Synopsys DC/ ICC/PT
- TSMC 65nm

ALU unit

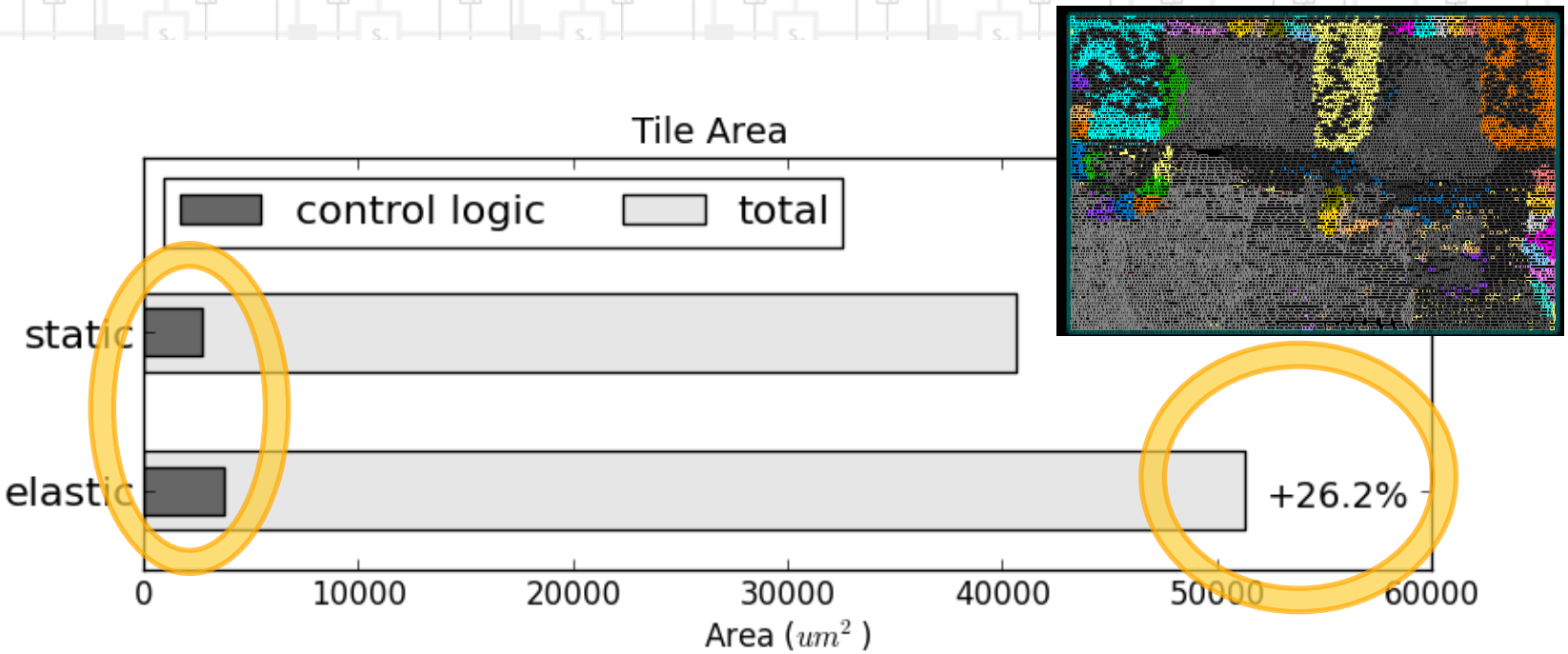
Data s-box

Ctrl s-box



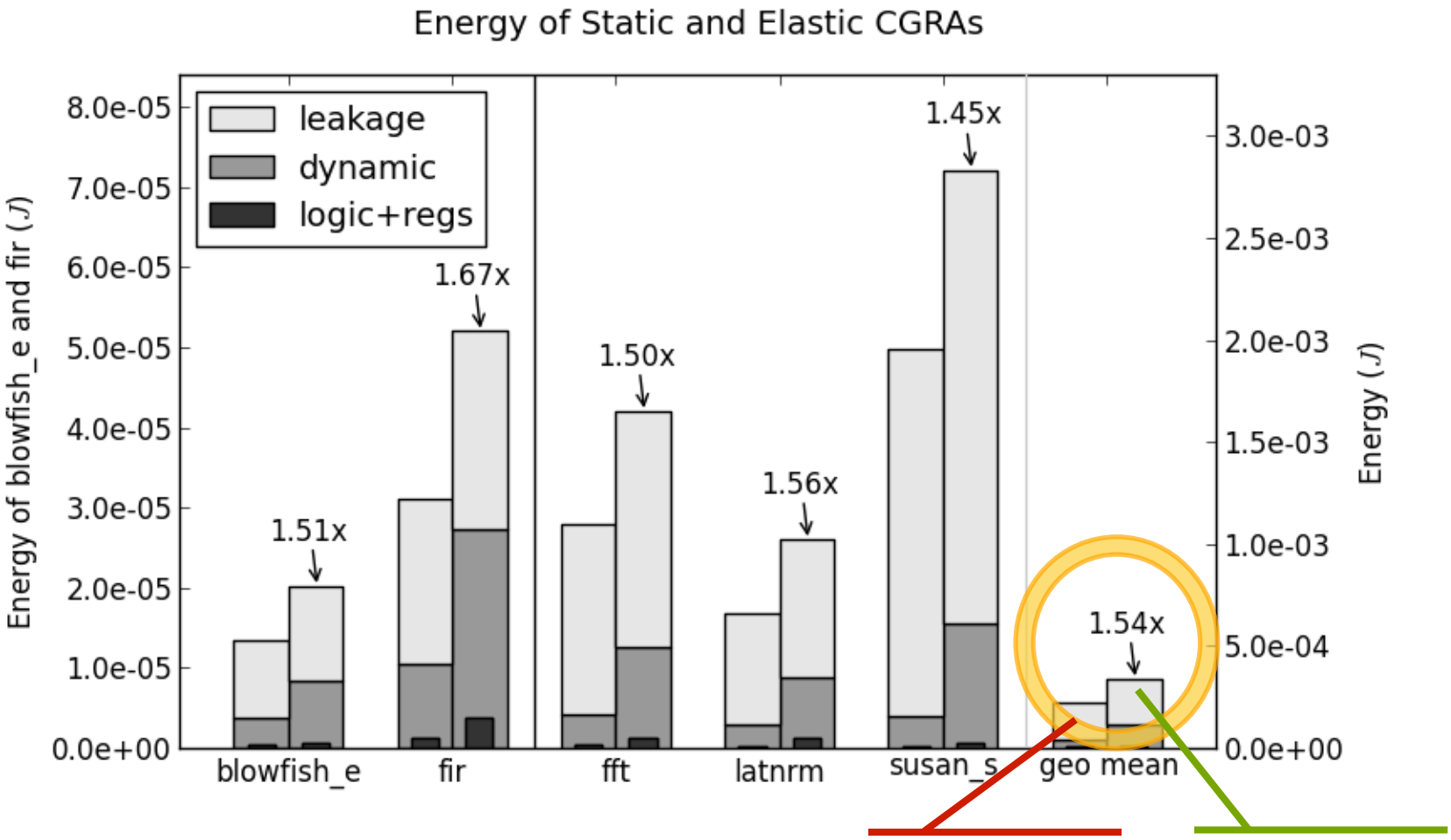
Config.

Area Results



- Negligible overhead due to elastic control logic
- Small overhead due to elastic control routing

Energy Results

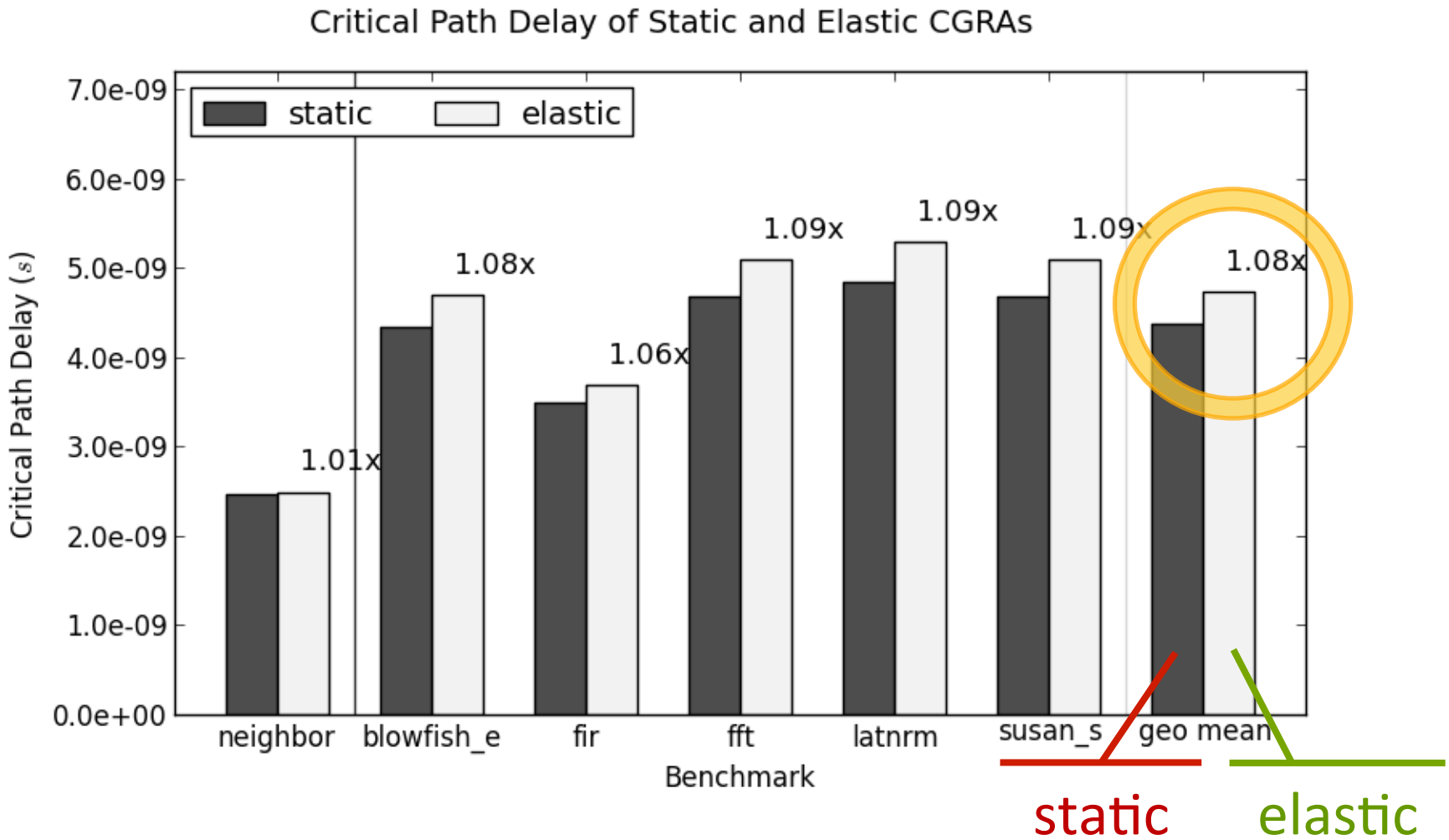


- Limited energy overhead

static

elastic

Timing Results



- Slight critical path increase due to area increase

Conclusion & Future Work

Take away:

Superscalar-like dynamically scheduled CGRAs, but with limited area/energy/time overhead

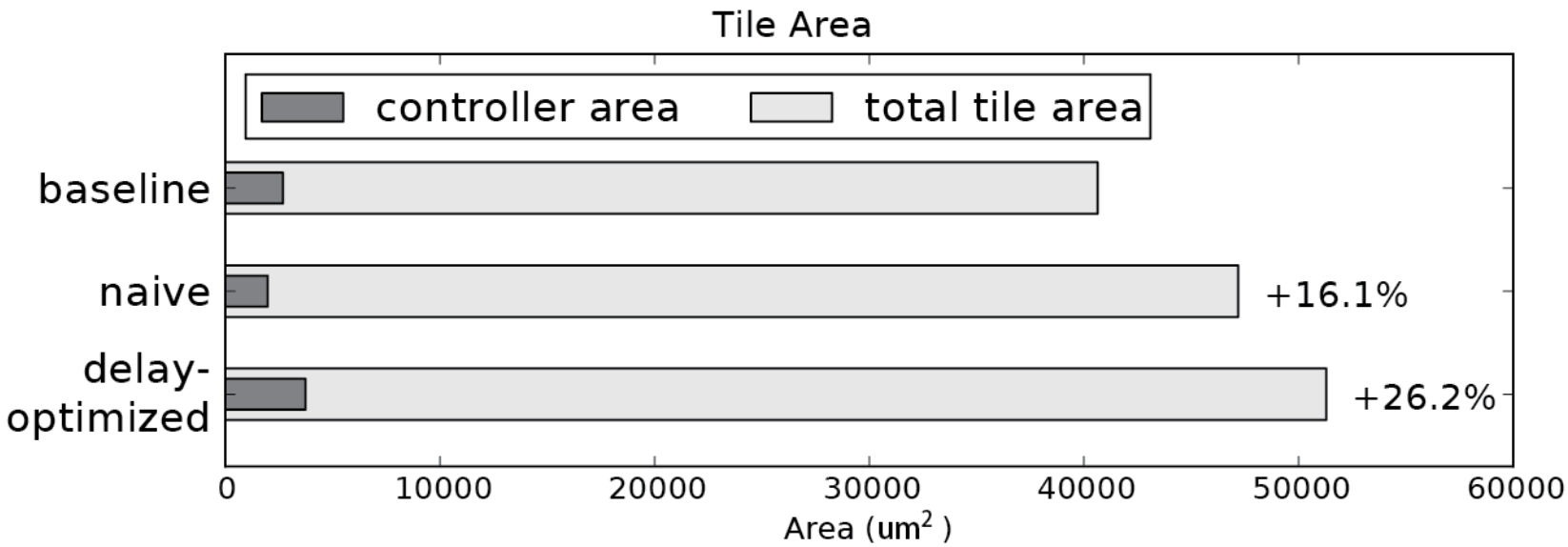
Future work:

- Boost utilization using standard compiler techniques
- Investigate inclusion of local memories
- Tape-out

Thank you!

Elastic CGRA: superscalar-like dynamic scheduling with limited overhead

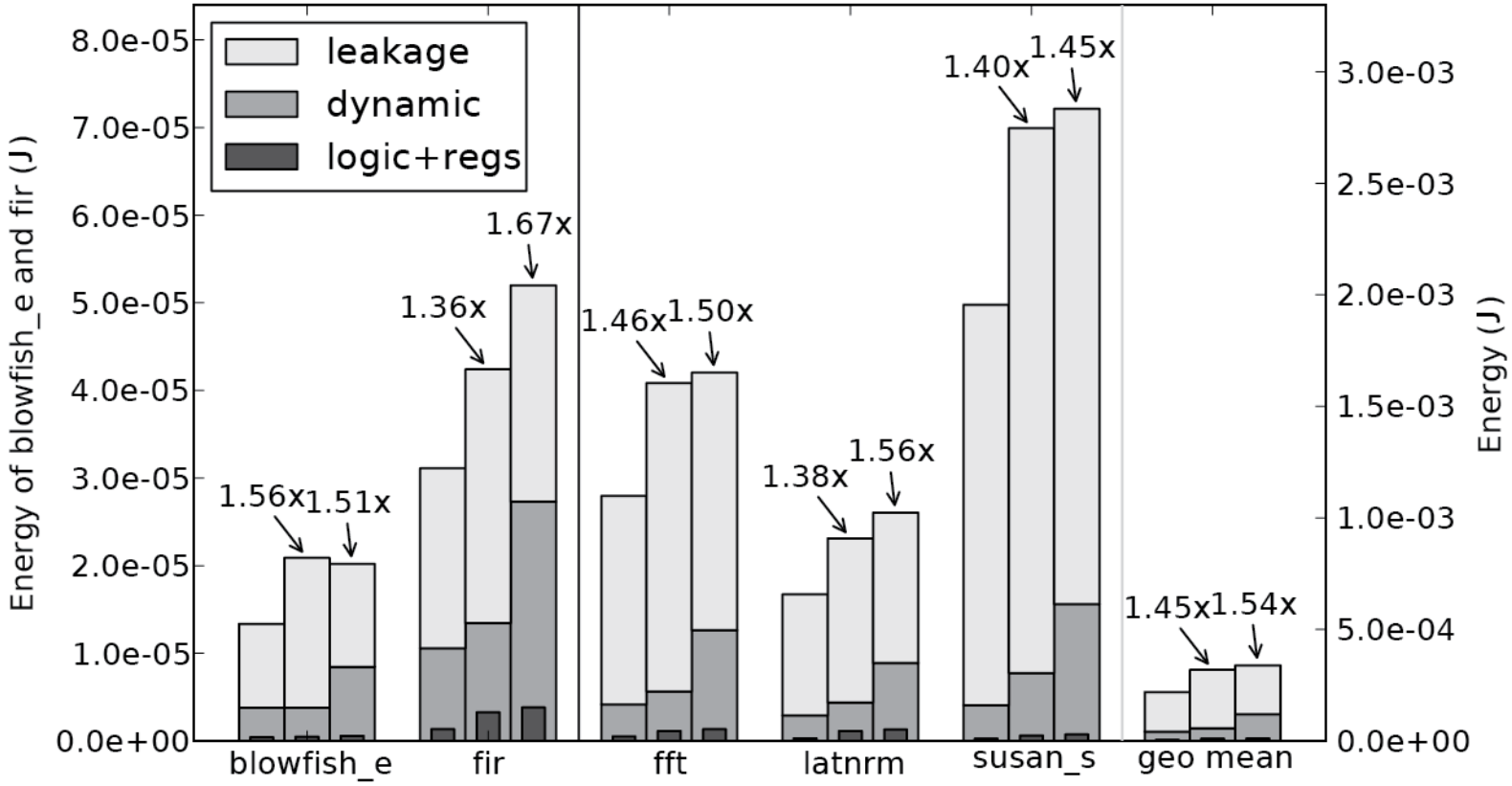
Area Results



- Negligible overhead due to elastic control logic
- Small overhead due to elastic control routing

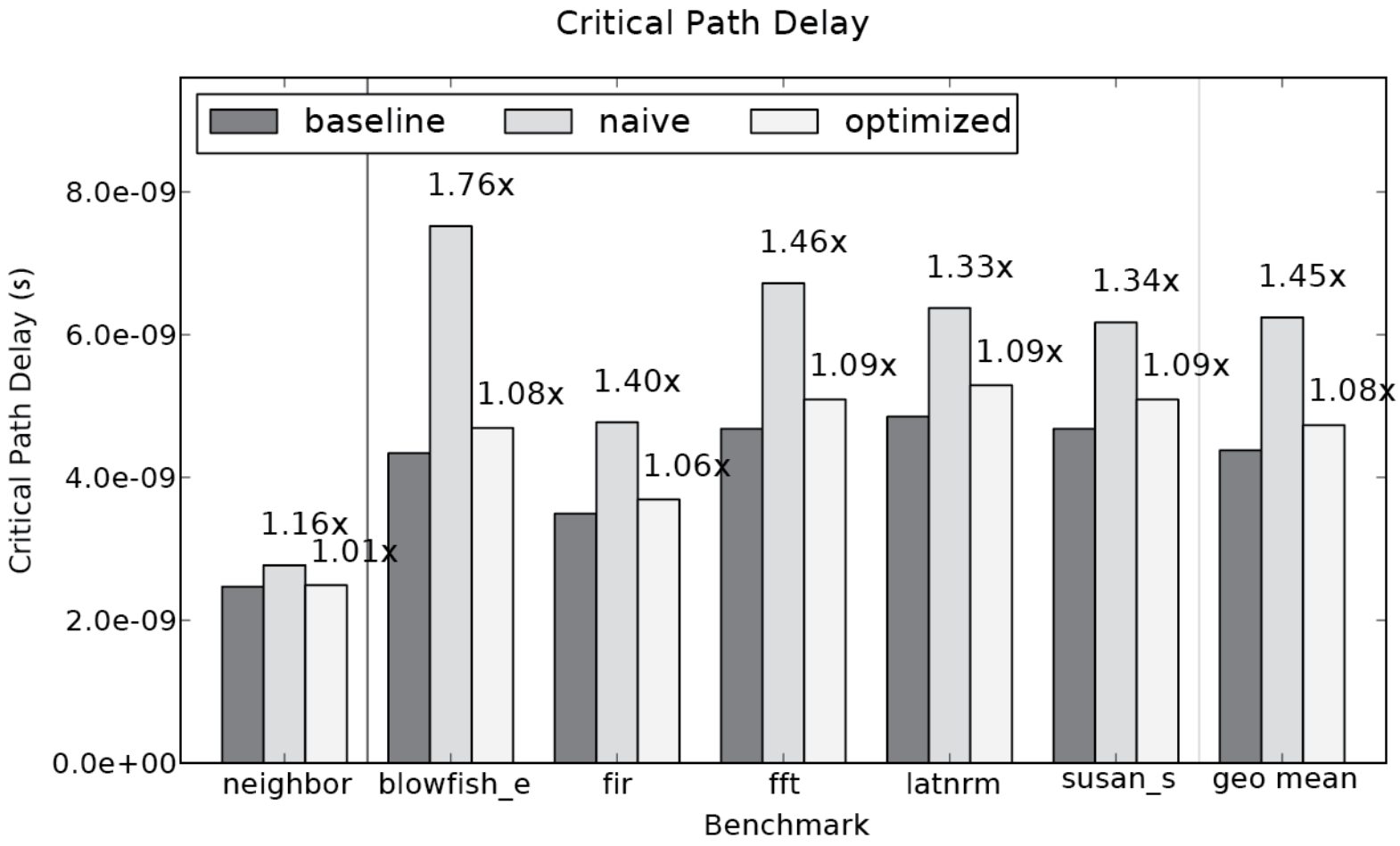
Energy Results

Energy of non-elastic baseline, naive elastic and optimized elastic CGRAs



- Limited energy overhead

Timing Results



- Slight critical path increase due to area increase