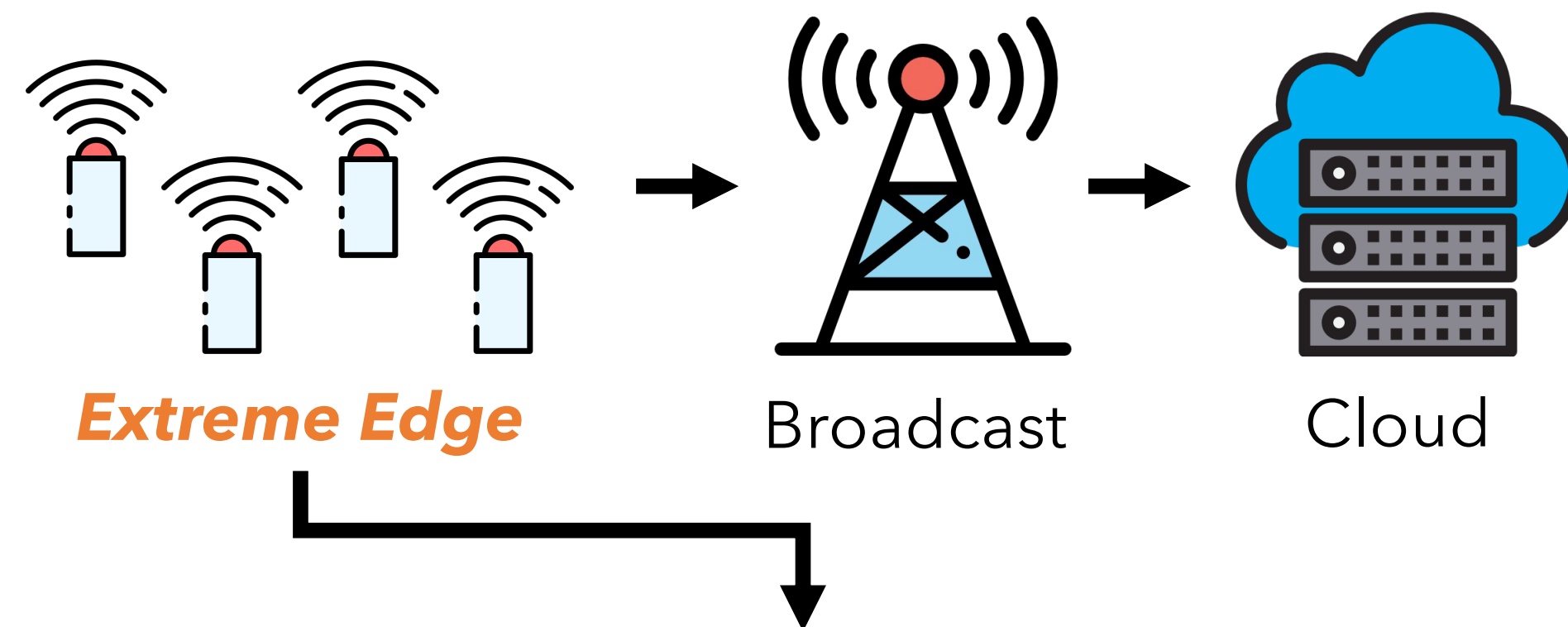


RipTide: A Programmable, Energy-Minimal Dataflow Compiler and Architecture

Graham Gobieski, **Souradip Ghosh**, Tony Nowatzki*, Todd C. Mowry, Nathan Beckmann, Brandon Lucia
Carnegie Mellon University, *UCLA

Smart devices at the extreme edge are rapidly emerging with huge industrial impact



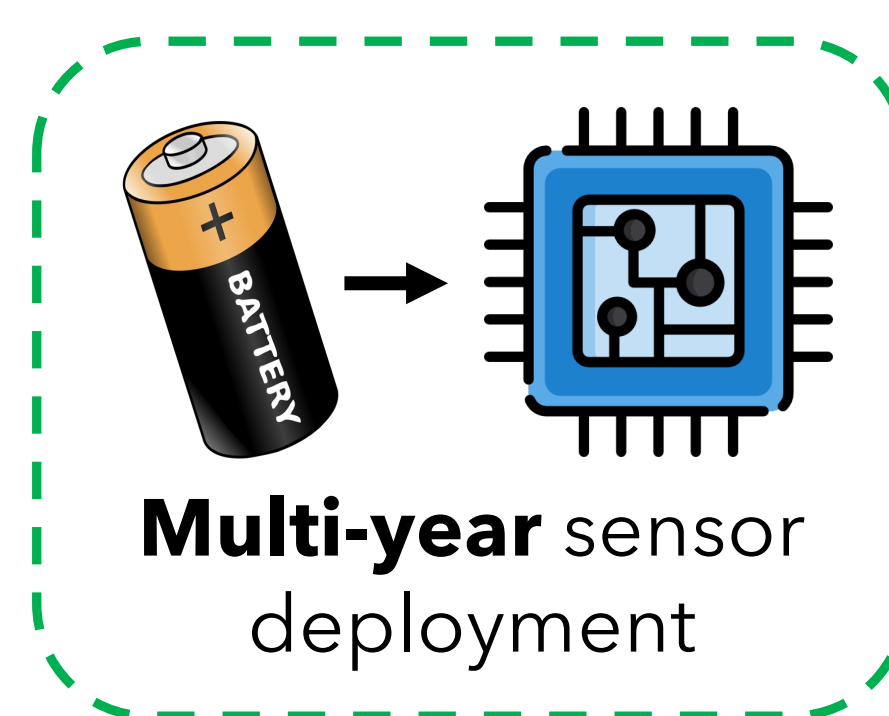
Tiny, smart sensor devices that enable advanced processing or inference



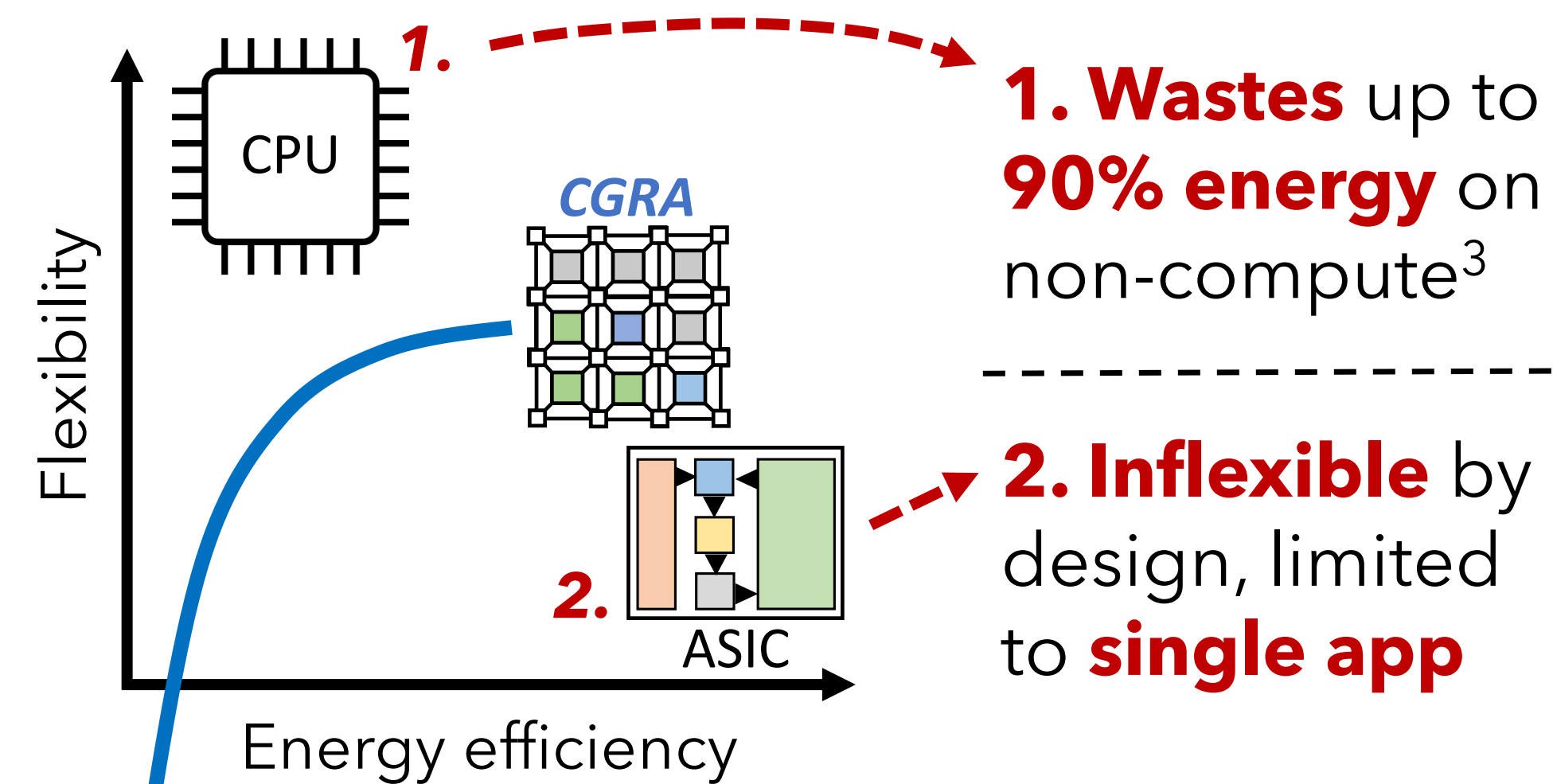
Trillions of devices coming!¹

Must sustainably & efficiently compute at the edge. How?

1. Run variety of apps on ultra-low power (ULP), μ Ws
2. More compute on-device, less communication²



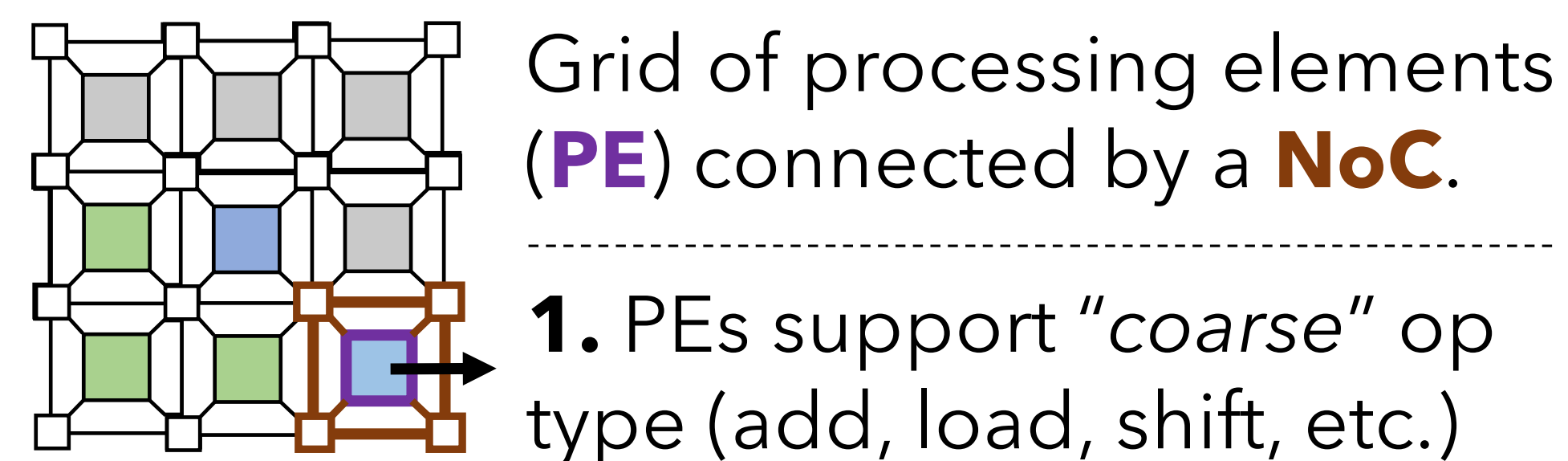
Goal: develop a highly flexible and energy efficient compute



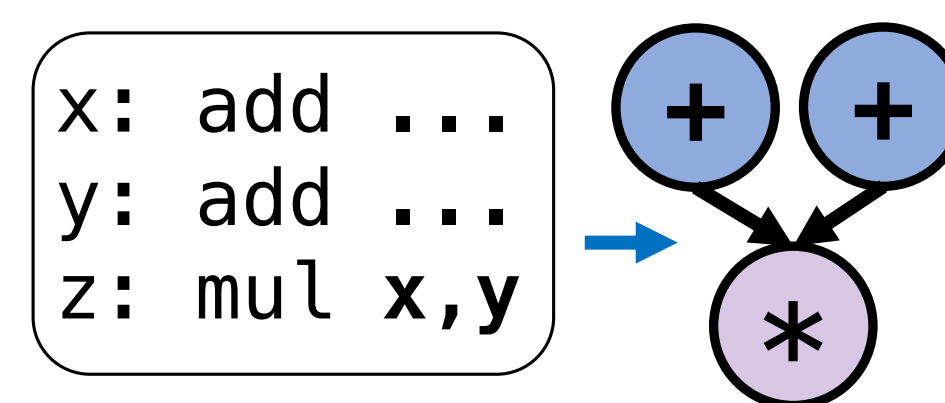
1. Wastes up to 90% energy on non-compute³
2. Inflexible by design, limited to single app

CGRAs are flexible & efficient!

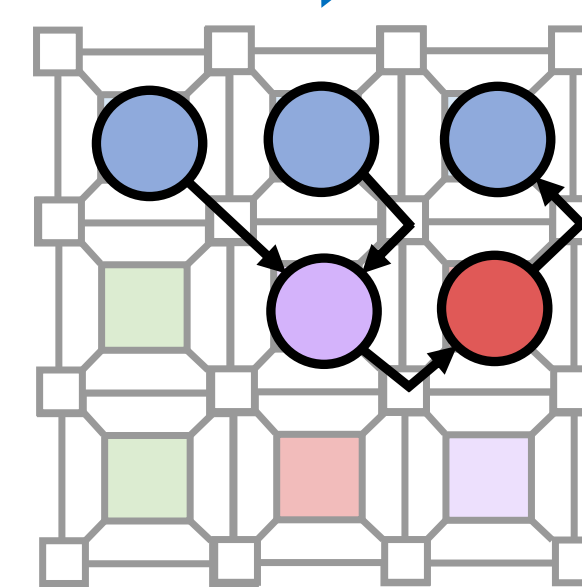
What is a coarse-grained reconfigurable array (CGRA)?



2. Compilers extract code into a dataflow graph to map to a CGRA (often small loops).



3. CGRA execution can be statically scheduled or use "dataflow firing": a PE "fires" once its inputs arrive via the NoC (no fetch/decode).



Prior ULP CGRAs are limited

```
1 void foo (...) {
2   for (i = 0..n) {
3     vlh v1, a + i
4     vadd v3, v1, v2
5     vsh b + i, v3
6   }
7 }
8 }
```

Runs **only affine inner loops**. No irregular control-flow or memory. CGRA code in assembly⁴

Insight: To achieve efficiency, CGRAs need to run entire apps and support common PL idioms

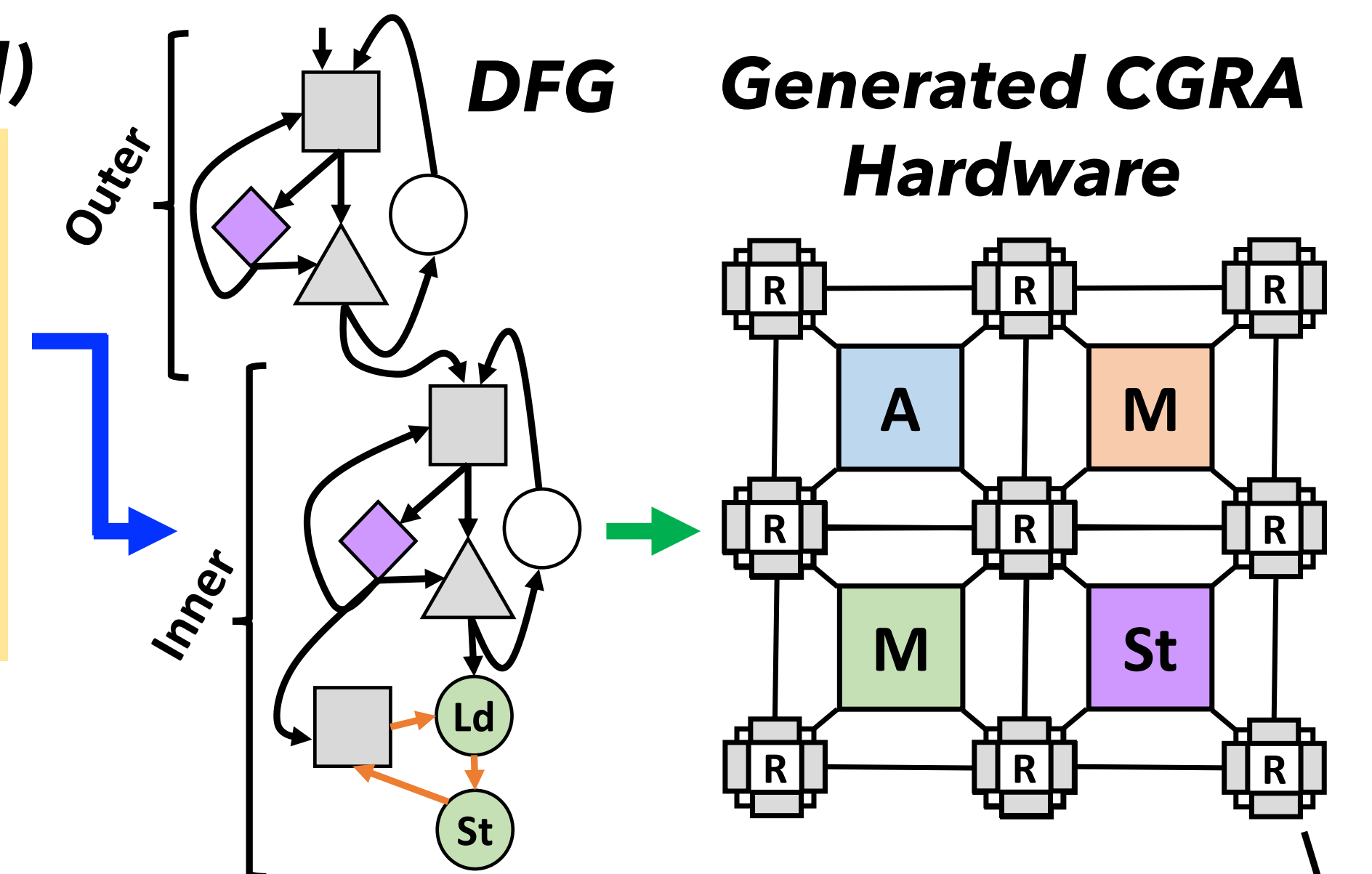
RipTide is a new ULP CGRA compiler & arch.

C code (lightly annotated)

```
1 #riptide void foo
2 (int* restrict a, b) {
3   while (!q.empty()) {
4     n = q.pop()
5     for (i in 0..n)
6       if (b[a[i]]) ...
7   }
8 }
```

- Handles arbitrary code via
- 1) Complex **control-flow**
 - 2) **Irregular mem.** accesses
 - 3) Enforced **mem. ordering**

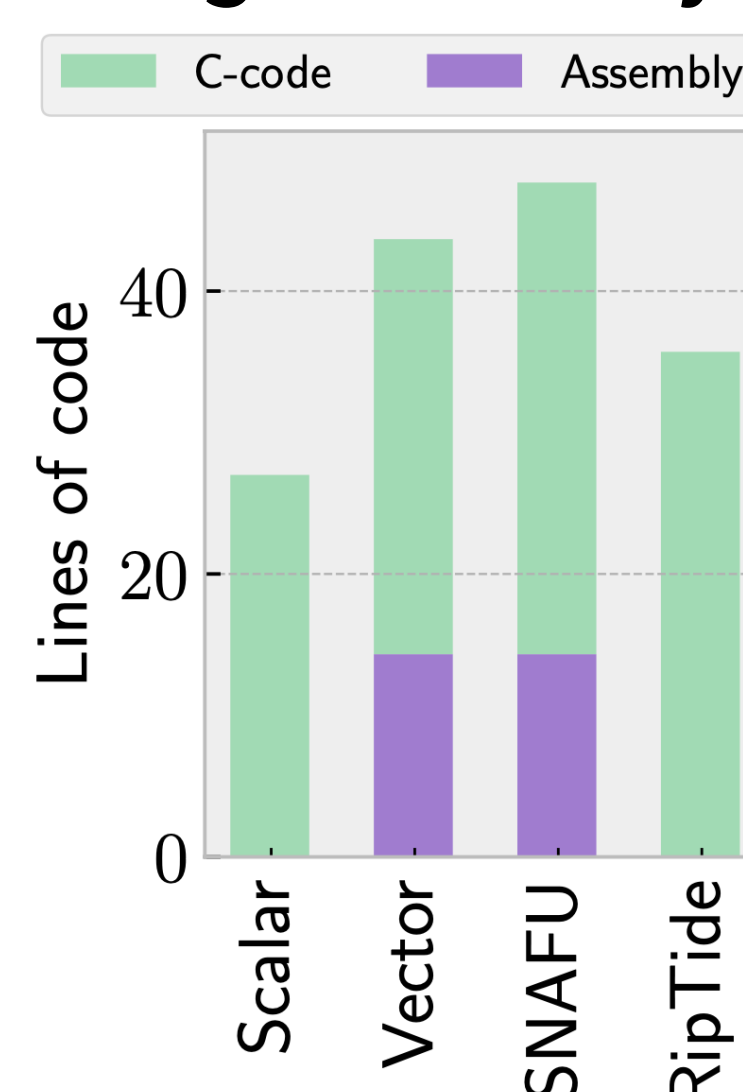
→ = Full compiler in LLVM
→ = SAT/ILP mapper (to CGRA)



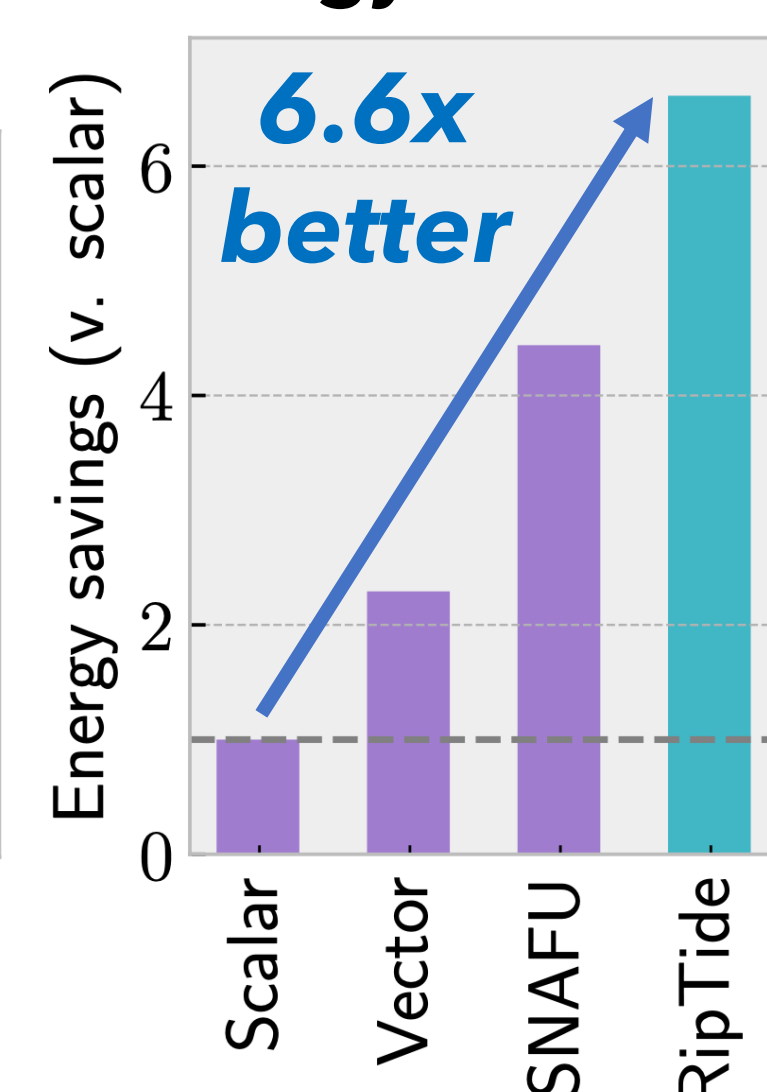
Optimizes away ops, reduces op subgraphs

Runs larger apps by **reusing routers to execute control-flow**. Frees PEs for more ops.

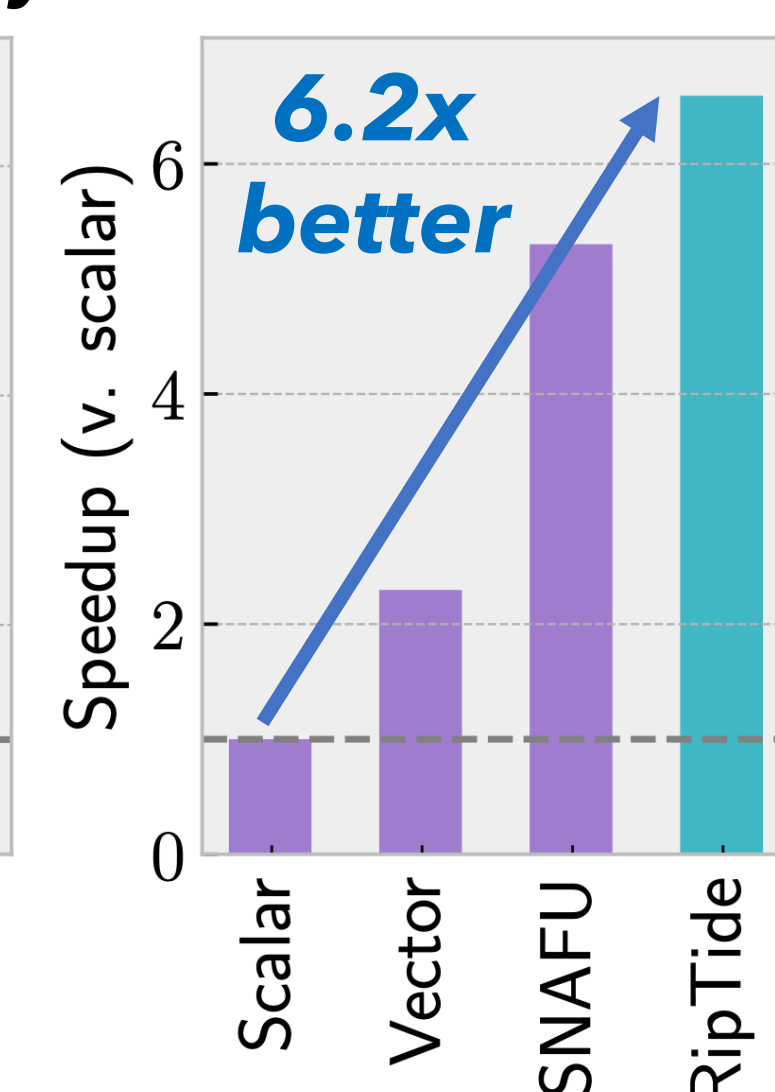
Programmability



Energy-Efficiency



Performance



10 apps from linalg, graph processing, and signal processing
+ Full RTL design and synthesized
+ Ran an entire DNN on RipTide!

¹Arm, "How to build a trillion connected things."
²Gobieski et al., "Intelligence Beyond the Edge: Inference on Intermittent Embedded Systems." (ASPLoS '19).

³Horowitz, "Computing's energy problem (and what we can do about it)." (ISSCC '14).

⁴Gobieski et al., "SNAFU: An Ultra-Low-Power, Energy-Minimal CGRA-Generation Framework and Architecture." (ISCA '21).