Tartan: Dataflow CMU Style
Your comments

• How do you deal with spread-out in spatial computing systems?
• Why does so much energy/perf go to the mem/interconnect?
• What about other networking topologies?
• Is this thing for real?
The latest, greatest dataflow architecture

- Map most of programs into a substrate
- One ALU/static instruction
- Hierarchical interconnect
- Dataflow execution
- Token-passing memory ordering
- They are after power efficiency (E*D)
What kind of dataflow?

- Static
  - They don’t talk about tags
  - They do claim ‘self-organized loop software-pipelining’
- Fine-grain
  - Every instruction synchronizes
  - Maybe even every slice.
- Control
  - Both Phi and steer instructions
Interconnect: 3 levels

• Local -- Inside a “page”
  • statically configured muxes steer values between ALUs

• Cluster -- between a bunch of pages
  • Statically routing a la FPGAs

• Chip-level -- to memory, the CPU, etc.
  • Dynamically-routed, packet-switched, NOC
Memory: Token-based

- A special “value” orders memory
  - in/output of every load and store
  - Parallelism is tricky.
- What’s the load/use penalty in Tartan?
Memory
Energy Delay

- Energy is not a great metric
  - Slowing down the clock decreases energy
- Power is not great either
  - It’s a rate: Energy per time, but for how long?
  - Only important for cooling.
- Energy Delay balances energy and performance
  - $E/D \sim 1 / \text{MIPS}^2/W$
  - $E/D^2$ is also popular. $(1/\text{MIPS}^3/W)$
Energy Delay

This is not a line graph!