Technology ramblings from a (micro+)arch perspective

Jonathan Beard

Staff Research Engineer, Arm Inc.

MEMSYS 2017

Disclaimer

These are my ramblings, not opinions of Arm Inc.

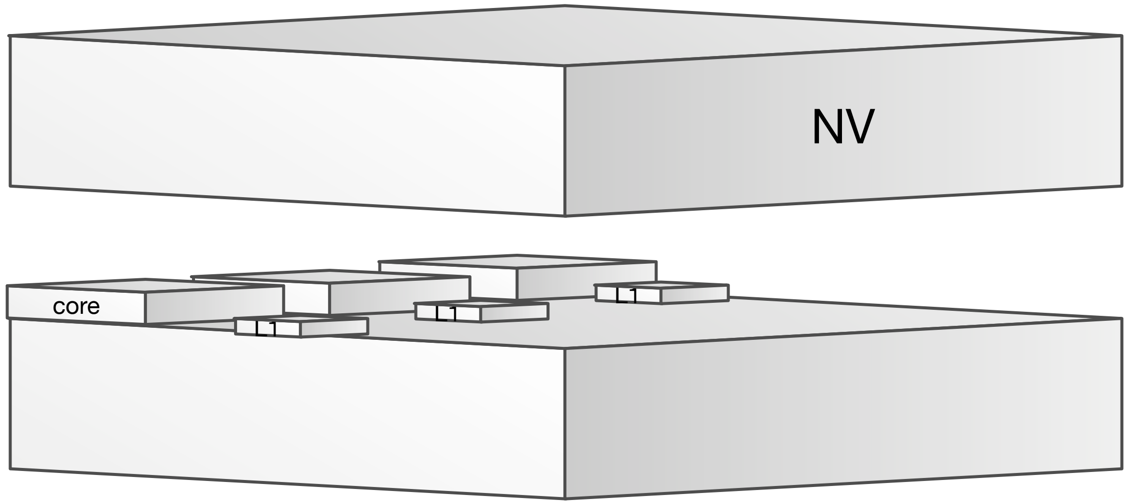
Any changes to architecture must go through the standard architecture review board process

Nothing within this presentation (or discussion) should be taken as a direction of future architecture or product plans for Arm Inc/Ltd.

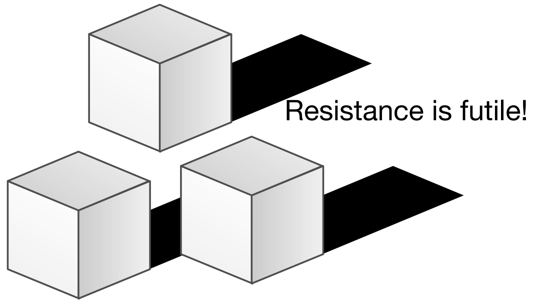
Capacity vs. bandwidth in our 3D integrated future

Lots of pins
Less memory / core
Really high bandwidth
Lower pin clock rate

Banked + Fewer pins
Less memory / core
High bandwidth
Higher pin clock rate



- Shorter wires between layers (lower latency)
- No refresh for NV (frees up core design a bit)



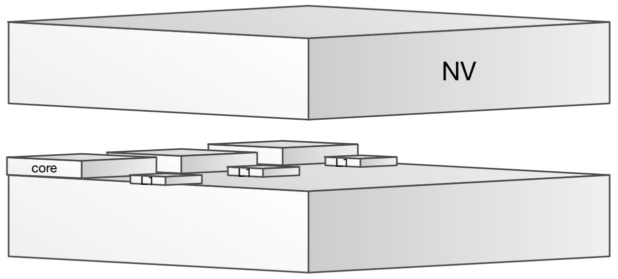
It’s not the technology, but how you use it within a system (memory tech + [micro+]arch + software)

Striking a balance

Lots of pins
Less memory / core
Really high bandwidth
Lower pin clock rate

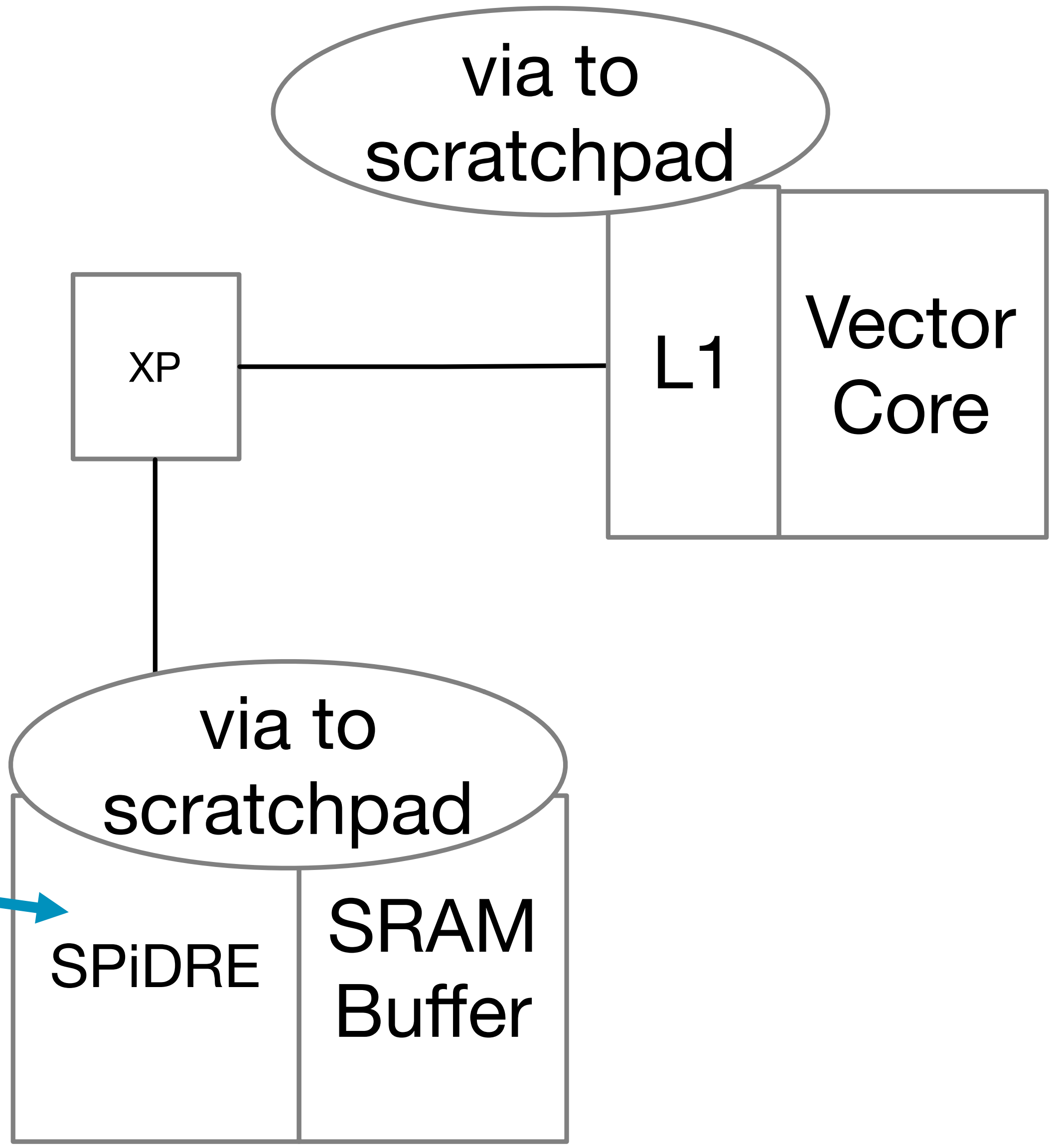
Banked + Fewer pins
Less memory / core
High bandwidth
Higher pin clock rate

- How to drive enough bandwidth to utilize?
- More pins == less capacity per core. Is this practical?
- How do you :
  - Partition data for 100K small cores?
  - Program 100K small cores?
  - Coherence….hah
  - Addressing scheme…



- More capacity, but fewer pins, how to move data from vias?
- How to drive enough bandwidth to utilize?
- Heterogeneous cores everywhere, how to offload to them? OS takes ~2000 cycles on/off, not practical to maintain bandwidth.
- Translation still a problem

Interesting idea



Shameless plug for my work

Processing In-/Near-Memory

Heterogeneous everything

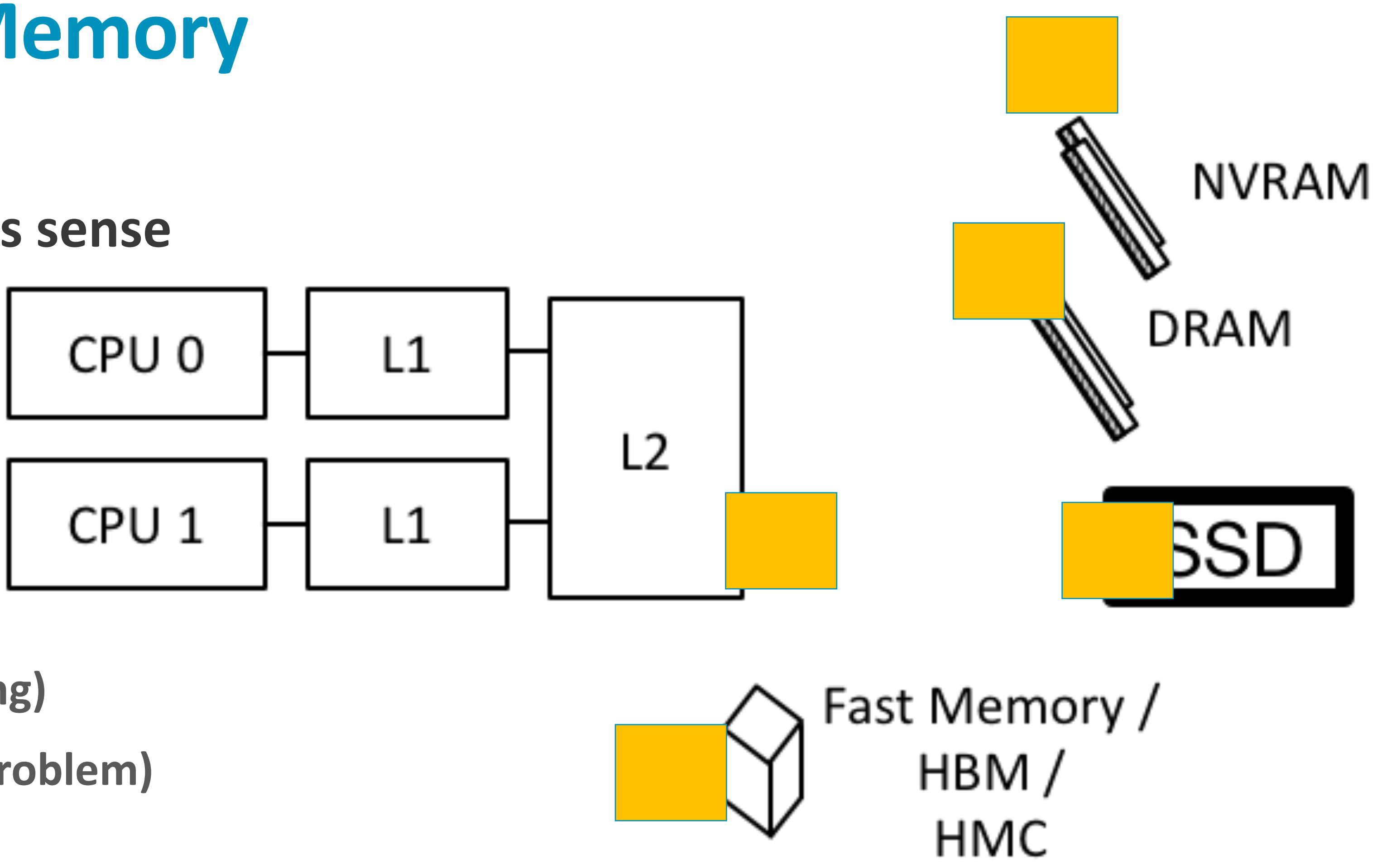
Can we compute where it makes sense

- **Right compute element**
- **Right memory technology**

Cores in/near-memory

Biggest Issues to Adoption

- **Programming Model (NP-Marketing)**
- **Translation (biggest and hardest problem)**
- **Scheduling (sounds easy right?)**



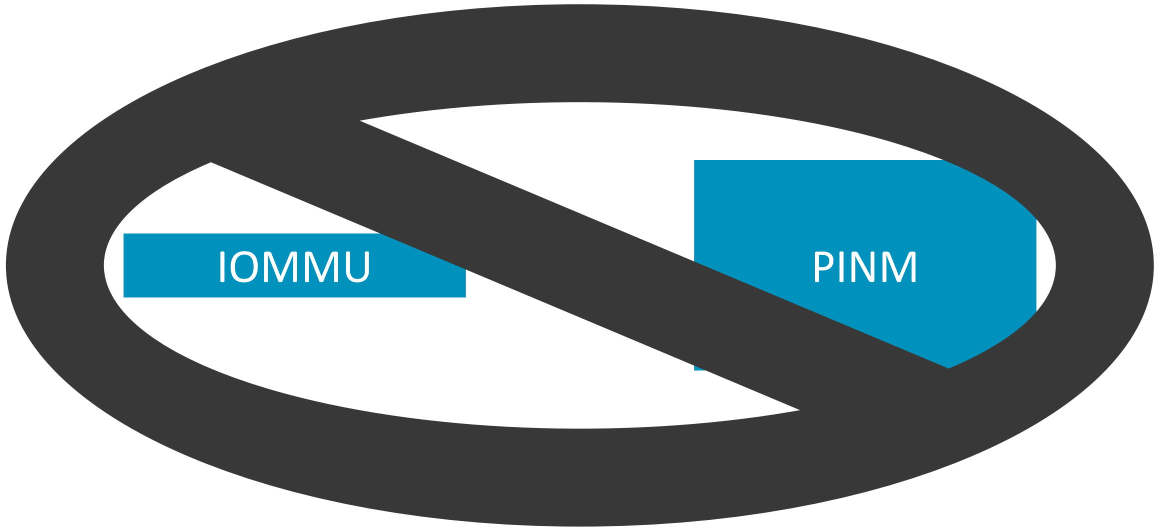
Folding (a.k.a. Manual Virtual Memory)

- Virtual memory started b/c of lack of available memory vs. storage
- Programmers wrote code that manually folded/unfolded to storage

"In ceasing to expend energy (item 1) **in a process whose main result is to make programs less fit to run on other machine configurations** (item 2), or **to run in company with other programs** (item 3), or to **run with temporarily reduced resources** (item 4), we do more than reduce costs; **we remove self-created obstacles** which today are impeding the development of needed types of systems"

- D. Sayre, IBM Yorktown Research (1969)

Evolution of Memory Separation

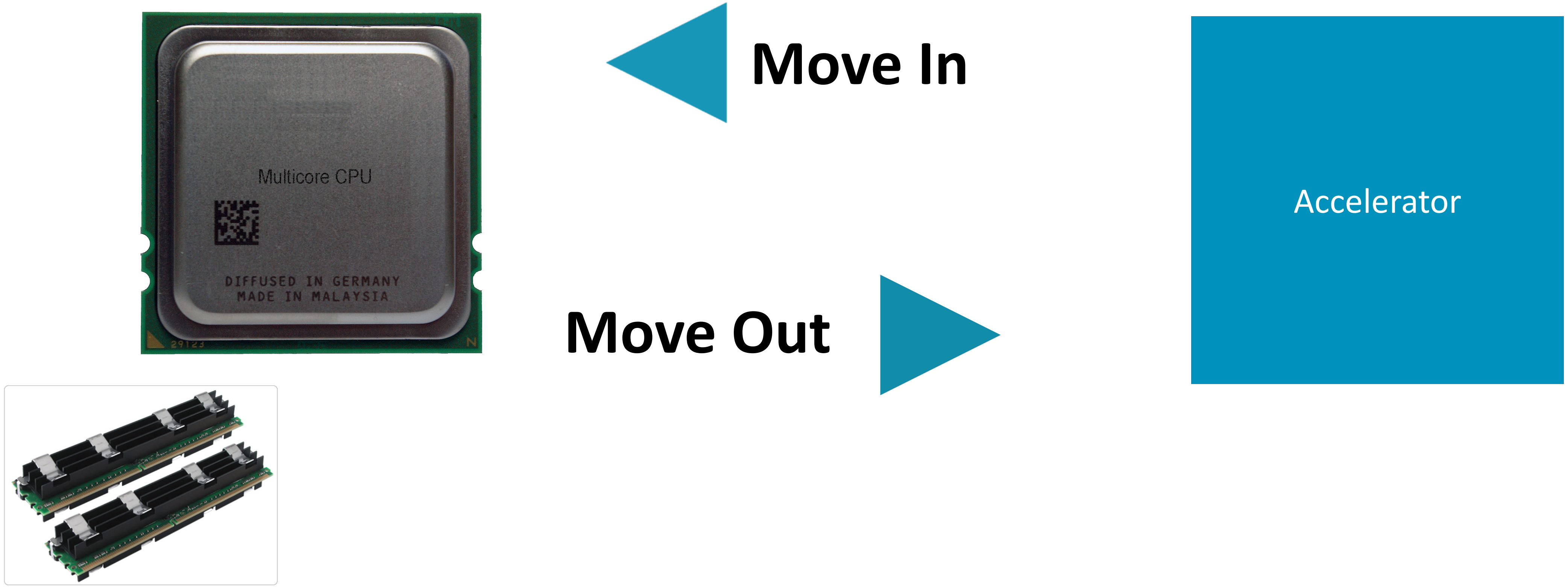


1970 - The Circle of Tech - “Self Created Obstacles”

Move In

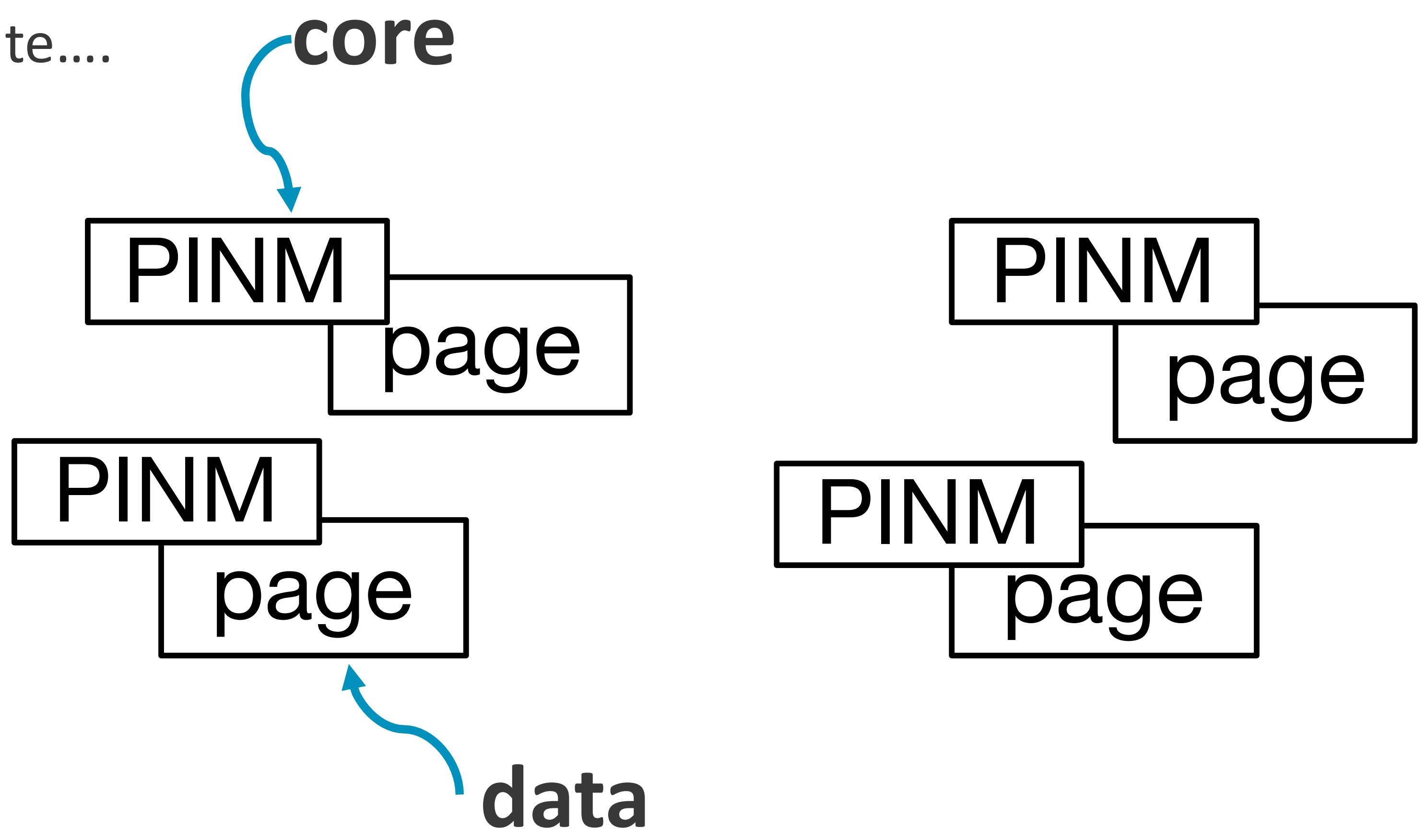
Move Out

2017 - The Circle of Tech - “Self Created Obstacles”



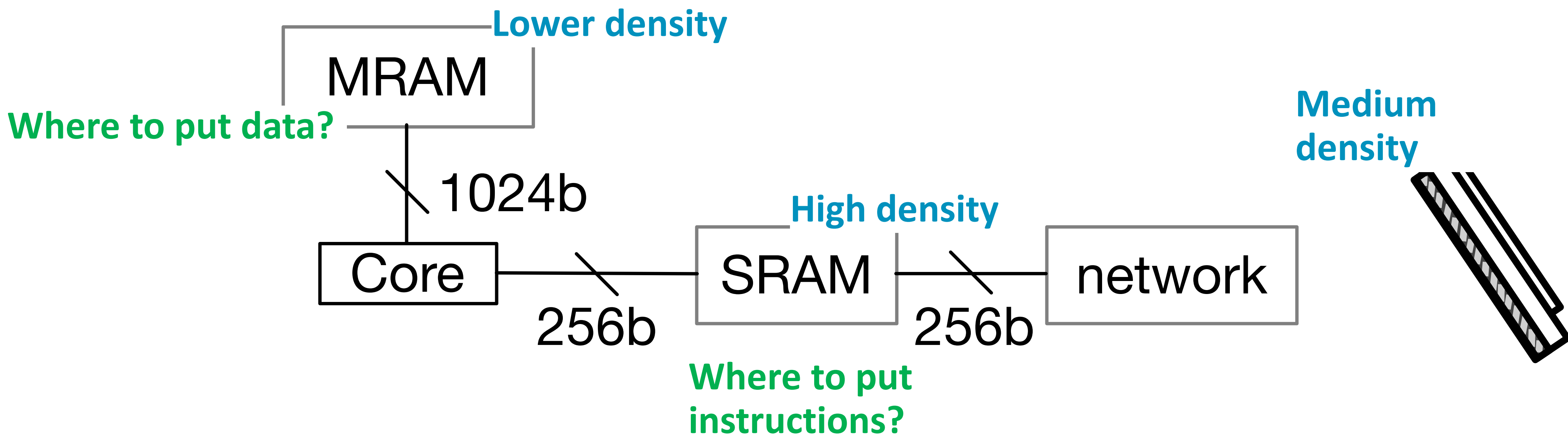
Scheduling

- Where to compute….



Scheduling

- Where to compute....
- BTW, lets make a simpler diagram



It’s the whole system, not just the technology.

arm