Deep Learning and Quantum Computing; Is there anything else worth working on?

Yale N. Patt

The University of Texas at Austin

Severo Ochoa Memorial Lecture

UPC, June 12, 2018

Random Concerns and Annoyances, and some thoughts on some problems

Yale N. Patt

The University of Texas at Austin

Severo Ochoa Memorial Lecture

UPC, June 12, 2018

What I want to do today

- ***Comment on Machine Learning***
- ***Comment on Quantum Computing***
- ***Comment on Parallel Processing (revisited)***
- ***Comment on Education***
- ***Some problems worth looking at***
  - ***Virtual Page Size***
  - ***Memory Bank Conflicts***
  - ***Merge Point Prediction***
- ***The future microprocessor***

Machine Learning

- ***The number of papers are soaring***
- ***Lots of results: Wow! It works!***
  - ***But cars smash into the back of trucks***
  - ***Woman on a bicycle gets run over***
- ***How many understand root causes***
  - ***“Learning” or “Adaptive Pattern Recognition”***
  - ***How and where are the weights adjusted***
  - ***The pattern space not covered: So what?***
- ***What about dynamic training?***
  - ***Ideal paradigm for branch prediction***
  - ***What else?***

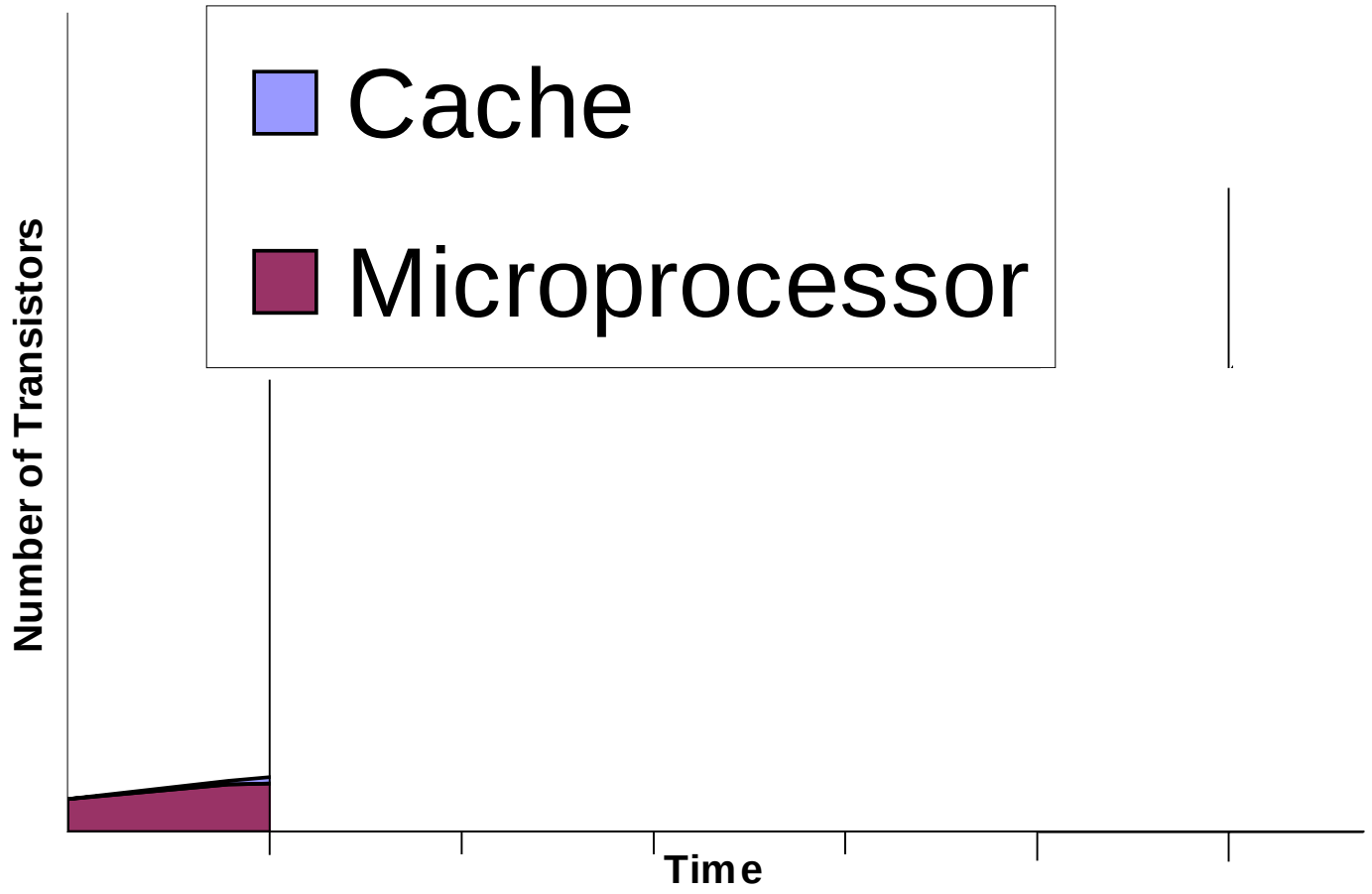
Quantum Computing

- ***Quantum: Computer or Accelerator***
- ***Too much glib hype by know-nothings***
- ***I used to ridicule it***
  - ***Then I talked to Burton Smith***
  - ***And Doug Carmean***
  - ***And the physics gurus at Microsoft who get it***
- ***So now I think there is a future in it***
  - ***But I would like to see more reality, less hype***

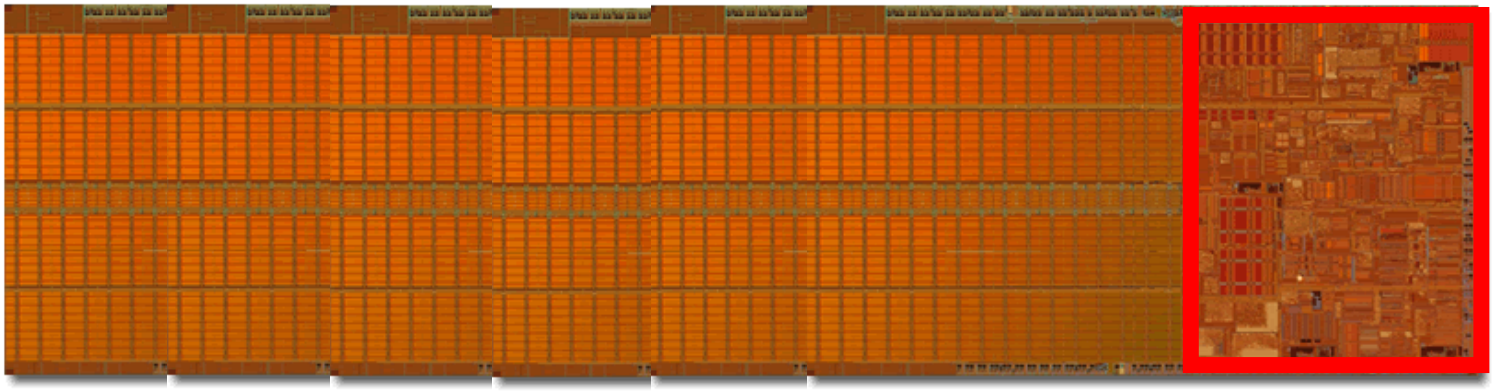
Parallel Processing

- ***Multi-core began Meganonsense***
  - ***Multi-core was a solution to a performance problem***
  - ***Hardware works sequentially***
  - ***Make the hardware simple – thousands of cores***
  - ***Do in parallel at a slower clock and save power***
  - ***ILP is dead***
  - ***Examine what is (rather than what can be)***
  - ***Communication: off-chip hard, on-chip easy***
  - ***Abstraction is a pure good***
  - ***Programmers are all dumb and need to be protected***
  - ***Thinking in parallel is hard***

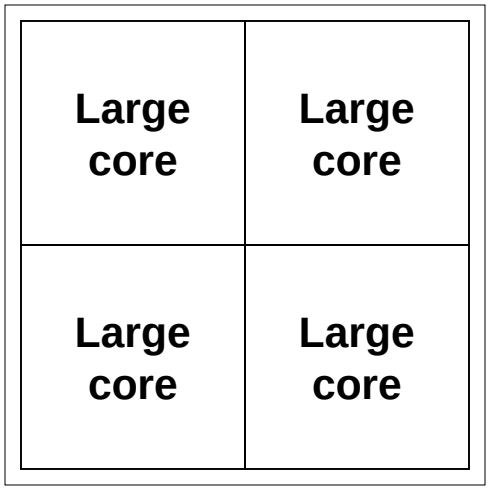
A Solution to a Performance Problem?



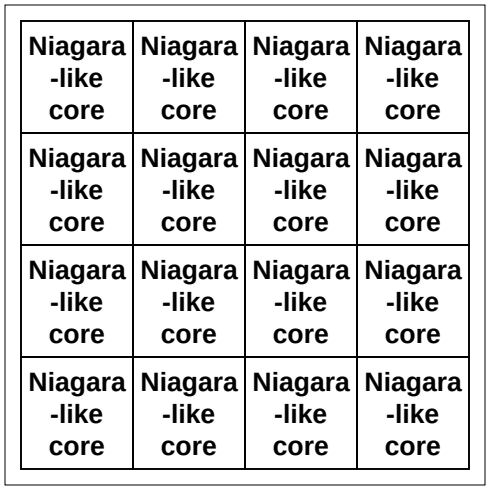
Intel Pentium M



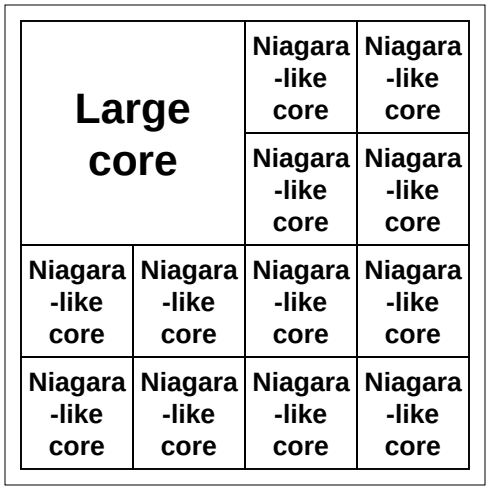
The Asymmetric Chip Multiprocessor (ACMP)



"Tile-Large" Approach

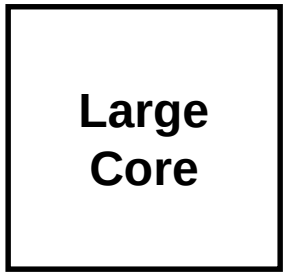


"Niagara" Approach



ACMP Approach

Large core vs. Small Core



- ***Out-of-order***
- ***Wide fetch e.g. 4-wide***
- ***Deeper pipeline***
- ***Aggressive branch predictor (e.g. hybrid)***
- ***Many functional units***
- ***Trace cache***
- ***Memory dependence speculation***

In-order

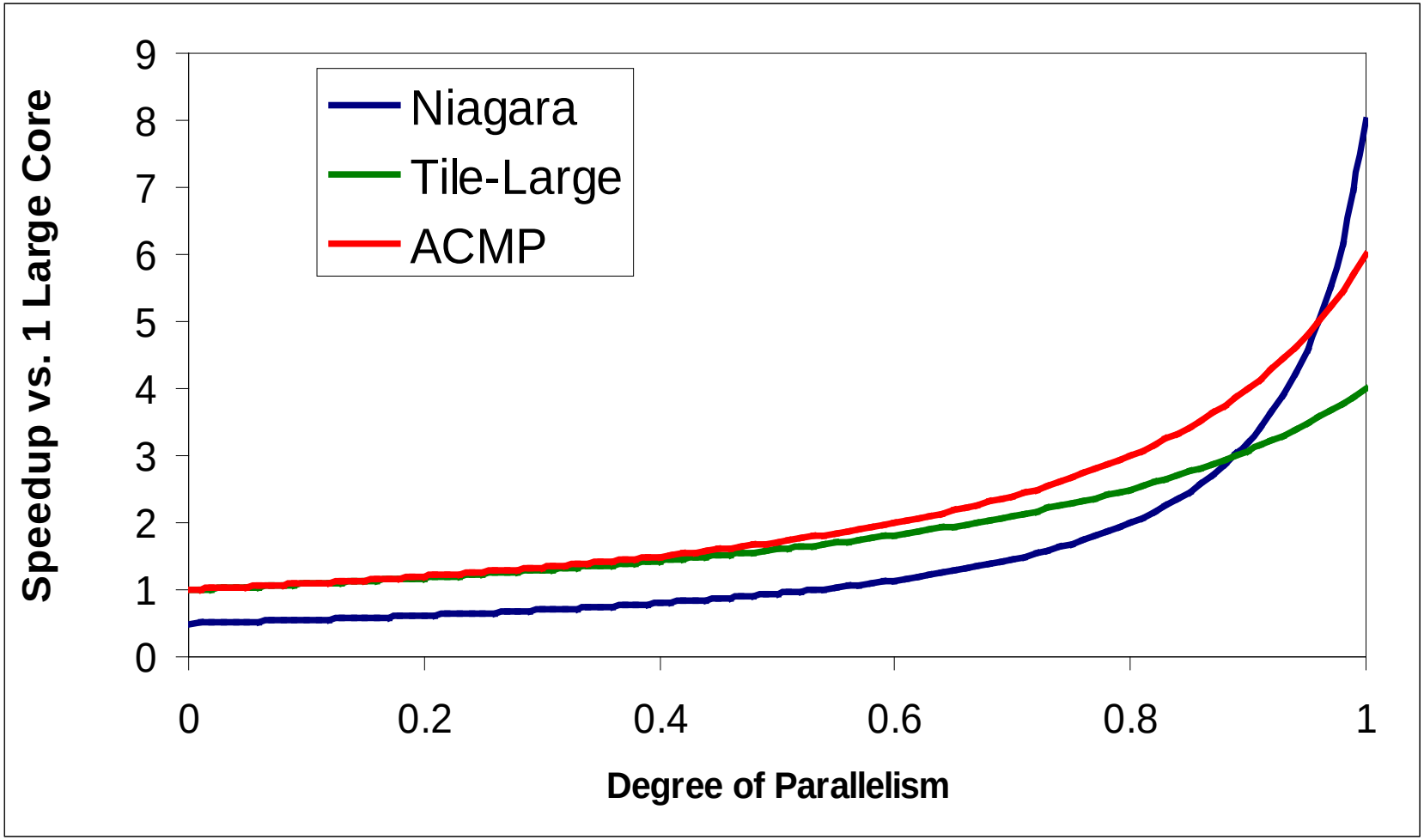
Narrow Fetch e.g. 2-wide

Shallow pipeline

Simple branch predictor (e.g. Gshare)

Few functional units

Throughput vs. Serial Performance



Thinking in Parallel is Hard

- ***Perhaps: Thinking is Hard***

Education

- ***My bottom up approach: 300 schools***
- ***JAVA or somesuch: 5000 schools***
- ***Why is that a problem?***
  - ***Hennessy's comment for software people in his Turing talk***
  - ***The needs of the future microprocessor***

Some Important Problems and our humble forays into them

- ***Virtual Memory***
  - ***Because 4K is too small,***
  - ***and we should have more flexibility***
- ***Bank Conflicts***
  - ***Because memory latency is still a major problem***
- ***Merge Prediction***
  - ***Because branch predictions are not always correct***

Higher Performance demands:

- ***Instruction Supply***
  - ***No ICACHE Misses***
  - ***No Packet Breaks***
  - ***100 Prediction Accuracy***
- ***Data Supply***
  - ***Huge Memory***
  - ***Single Cycle Access***
- ***Instruction Execution***
  - ***Lots of functional units***
  - ***Data Flow processing***
  - ***Low latency interconnect***

Virtual Memory

- ***First: All sizes 2^k, k> 11***
  - ***Normal PTE with extra bits providing size info***
  - ***Seamless access of the physical page frame***
  - ***Removes essentially all L1 TLB misses***

- ***Next step: Smallest size: 2^18***
  - ***L1 cache design: more index bits***
  - ***Prefetching: can't cross page boundary***
  - ***Memory dependence checking***
  - ***Reduced page walk latency***
  - ***Page table size is much smaller***

Bank Conflicts

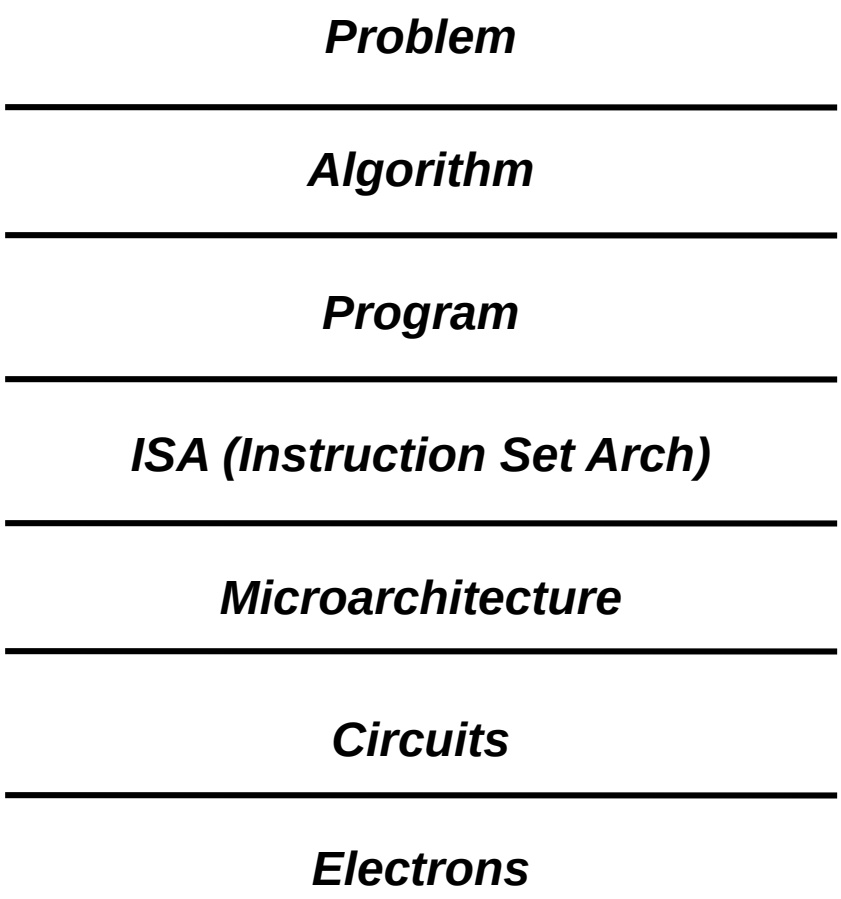
- ***Reserve a small part of each DRAM***
  - ***Bank conflict stores a duplicate***
- ***Duplicon***
  - ***Part of the memory controller***
  - ***Reroutes one of the accesses to the duplicate***

Merge Prediction

- ***What to do about unpredictable branches***
- ***Predict means likely misprediction penalty***
- ***Fetch instead from the merge point***
  - ***Provides necessary uop supply***
  - ***No wasted energy***

The Future Microprocessor

- ***The White Paper from Leiserson et. al.***
  - ***Plenty of room at the top***
  - ***i.e., my transformation hierarchy revisited***
- ***Accelerators***
- ***The vonNeumann Machine***


Problem
Algorithm
Program
ISA (Instruction Set Arch)
Microarchitecture
Circuits
Electrons

Accelerators

- ***Many implementation mechanisms***
  - ***ASICs***
  - ***FPGAs***
  - ***EMT instruction (with writeable control store)***
- ***Examples (Quantum computing, Machine learning)***
- ***Requires the attention of***
  - ***The person writing the algorithm***
  - ***The programmer***
  - ***The compiler writer***
  - ***The microarchitect***

…which requires a fresh approach to Education!

Accelerators will need CPUs to feed them

- ***Which means Von Neumann will continue to thrive!***

The VonNeumann Paradigm

- ***A straightforward model of executing programs***
  - ***Fetch, Decode, Evaluate address, Fetch Data, …***
- ***Many have suggested its demise***
  - ***I insist it will remain***
  - ***Our best mechanism for maintaining order, not chaos***
- ***Augmented with accelerators, of course***

When I describe this model,

- ***I get enormous pushback.***
- ***What about portability?***
- ***Without portability, the economics isn't there;***
- ***Industry won't buy into it.***
- ***My response: Economics Be Damned!***

Economics Be Damned!

- ***Some things are too important***
  - ***Curing cancer***
  - ***Predicting tsunamis***
  - ***Cyber defense***
- ***We already have an industry point solution***
  - ***Google's Tensor Processing Unit***
  - ***Perhaps next: a deep learning training unit?***

Thank you!