Welcome and Overview

Future computing for particle physics workshop

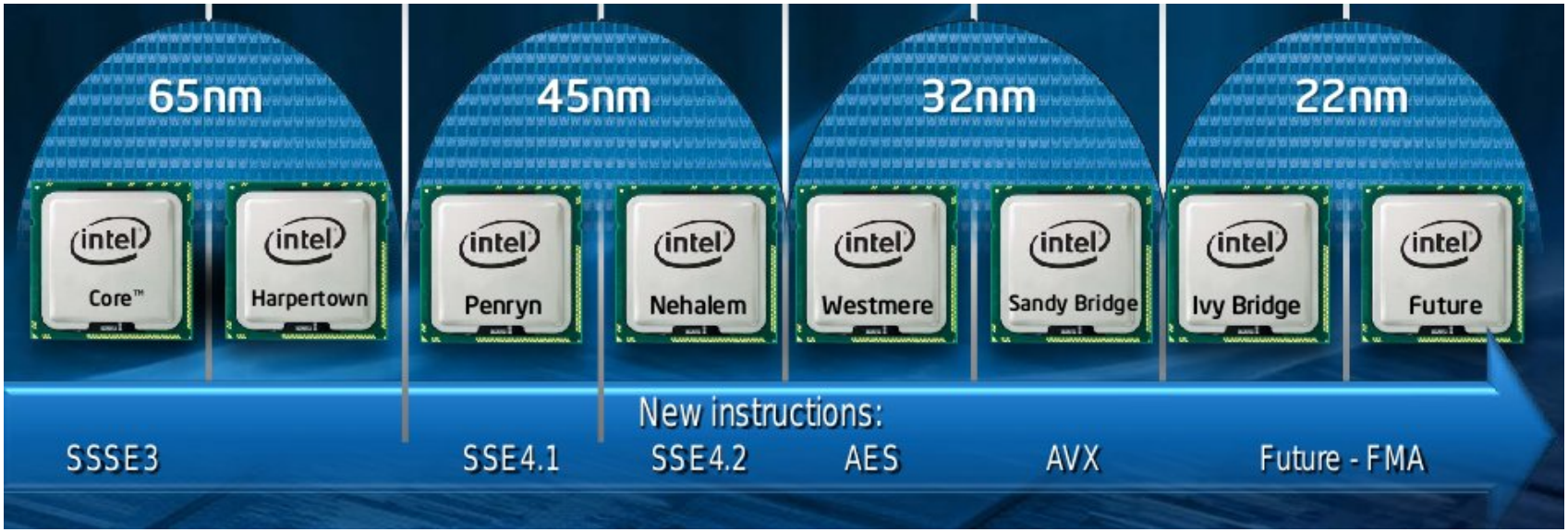
Philip J. Clark[1] and Roger Jones[2]

[1]University of Edinburgh

[2]University of Lancaster

15th June 2011

How to cope best with future devices?



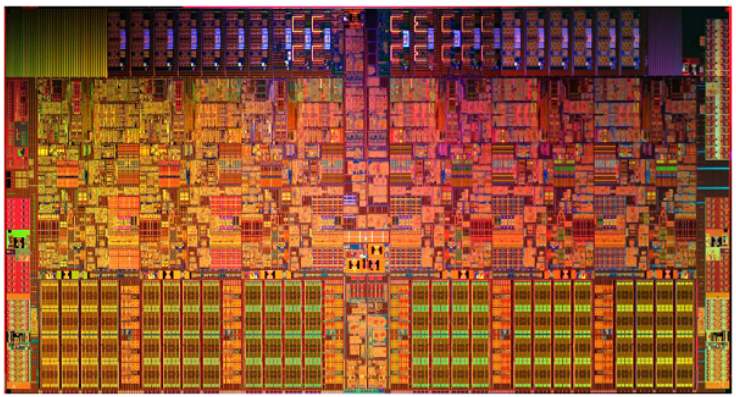


Figure: Westmere (6 cores) 32nm

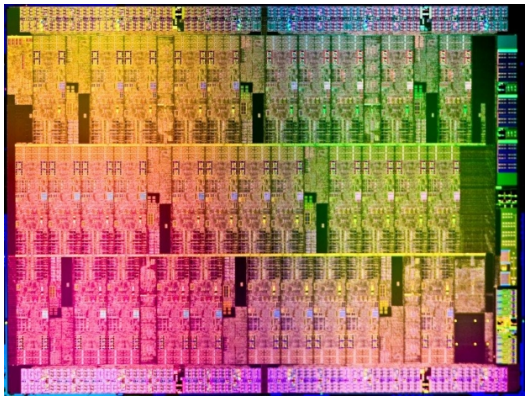
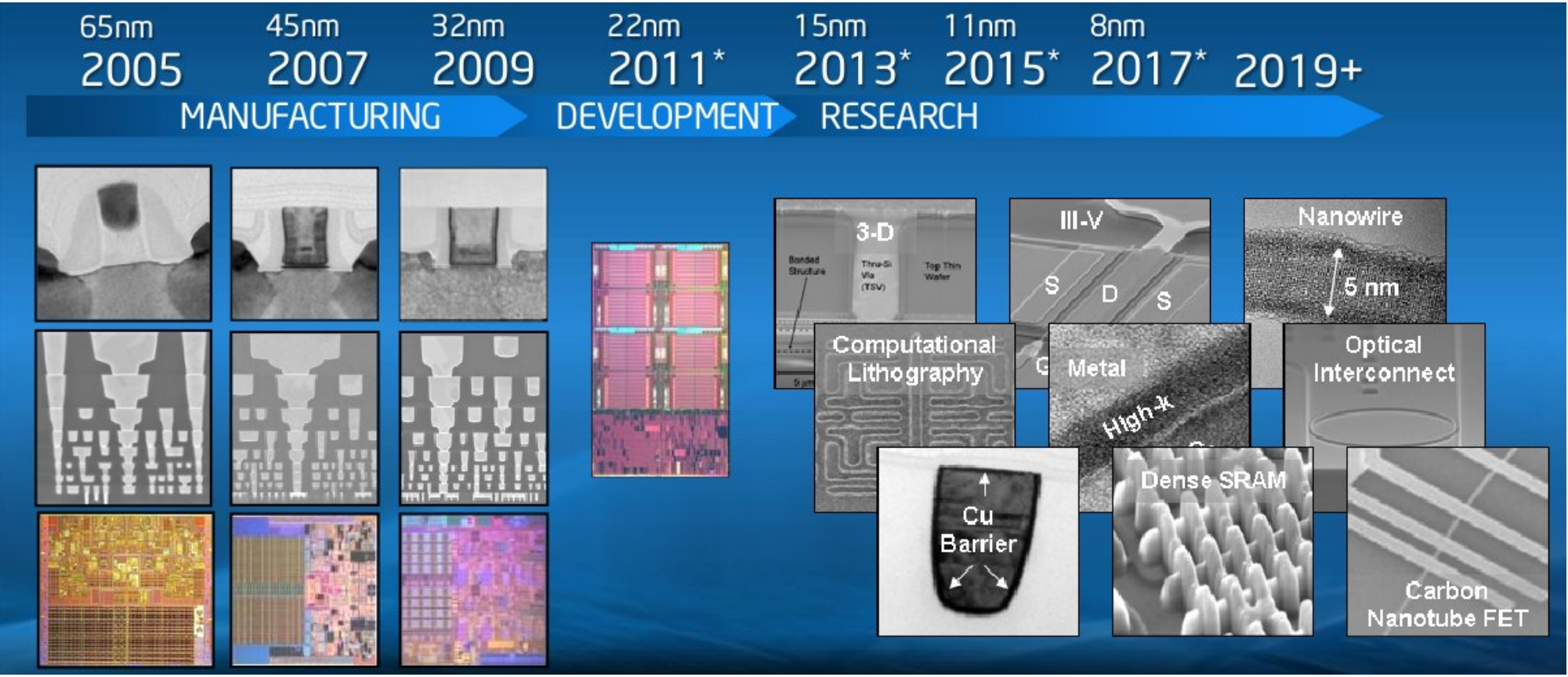


Figure: Knights Ferry (32 cores) 32nm

Will it continue? Moore's Law seems alive and well

Kirk Skaugen – Intel vice president



Will it keep continuing?

Many cores and parallel code limits

Multi-core processor

Two or more independent cores on single integrated circuit die

Many-core processor

When the number of cores is large enough that that traditional multi-processor techniques are no longer efficient (often deliberately)

Hardware trend towards many core means we must start to run more parallel algorithms to take advantage (with some theoretical limits)

Amdahl's Law

Gains limited by proportion (α) of parallel vs. seq. code (s)

$$\frac{1}{(1-\alpha)+\frac{\alpha}{s}}$$

Gustafson's Law

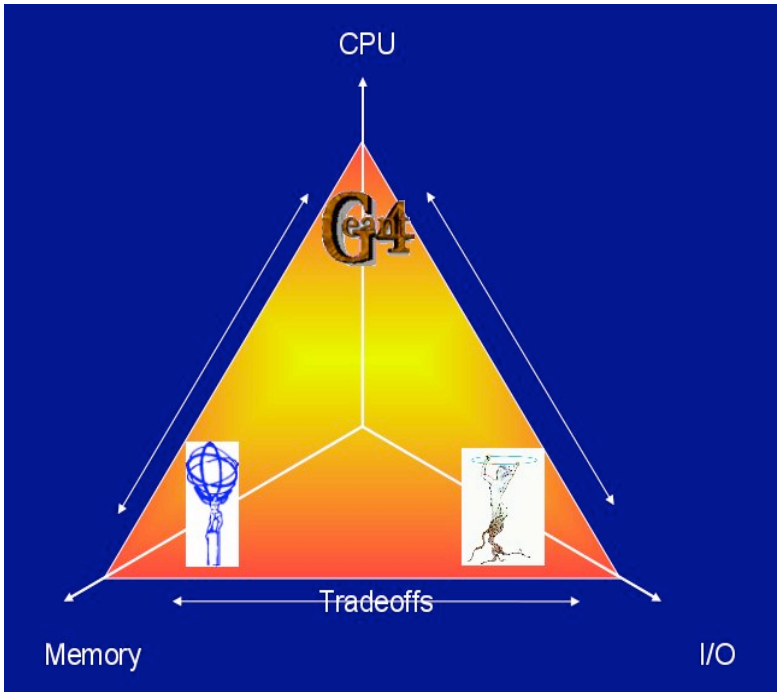
Gains limited only by no. of processors (P) i.e. problem size

$$P-\alpha(P-1)$$

Need to think big! (Gustagson's law)

What limits performance?

- The traditional limiting factors: the "Paolo triangle" (picture stolen from him) (CPU/memory/IO)
- G4 (CPU limited), Reco (Mem limited), Root (I/O limited), Digi/pileup (all?)



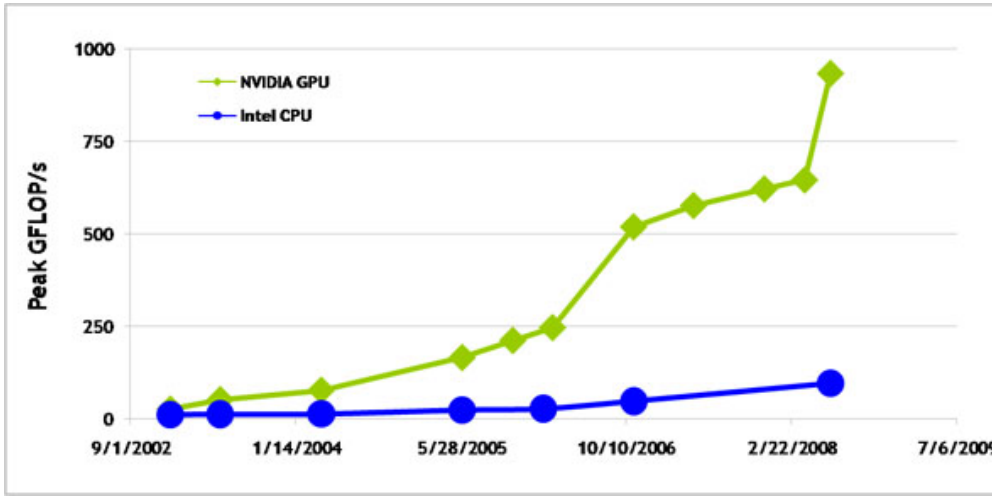
Future limitations:

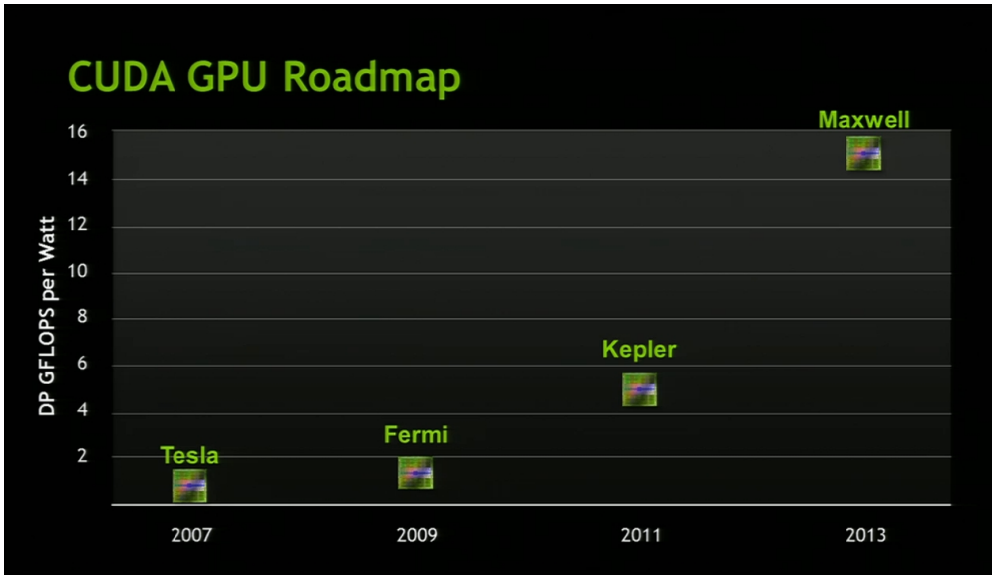
Compute limit = Power wall * Memory wall * ILP wall

where power is electricity and ILP = instruction level parallelisation.

General Purpose Graphical Processing Units

- Staggering “potential” performance
- The main strengths are:
  - Many more floating point units
  - Thus can provide many more threads in flight
  - The memory interface faster
  - No. flops increasing exponentially (doubling time is half that of CPUs)
  - Shared memory has equivalent speed to CPU cache





Possible ways forward

Options

1. Stay with simple event-level parallelisation
   - Assumes necessary memory remains affordable
   - Major I/O problem
2. Rely on forking (n processes sharing memory)
   - Use “copy-on-write” (AthenaMP idea)
   - Rely on virtualisation? (e.g. KSM - shared memory module)
   - Use NUMA (non-uniform memory access) to improve memory I/O
   - Many other ideas to improve performance
3. Move to a fully parallel (or at least multi-threaded) paradigm
   - Many cores and withalgorithm accelerators (GPUs?

Programme

Wednesday Overview and LHC Computing Challenges

Thursday

- Multicore Hardware and Applications
- Software optimisation and performance tuning
- Future I/O ideas
- GPU/Manycore Applications: motivation and projects

Friday GPU/Manycore Applications: motivation and projects cont.

- Tracking and upgrade
- High level trigger
- General GPU ideas: (openCL, openmp directives etc.)

Massive parallelisation & scheduling exercise

Whisky tasting

David Wishart (speaker), 18.00 Wednesday 15th June, 2011
Playfair Room, Royal College of Surgeons,Edinburgh

Dinner, Thurs evening, space-time coordinates (7:30pm at Agua (Apex City Hotel), Grassmarket)