Hardware facilities for vector computing

2nd Annual Concurrency Forum Meeting
February 5th 2013

Andrzej Nowak, CERN openlab

CERN openlab

AVX – the philosophy

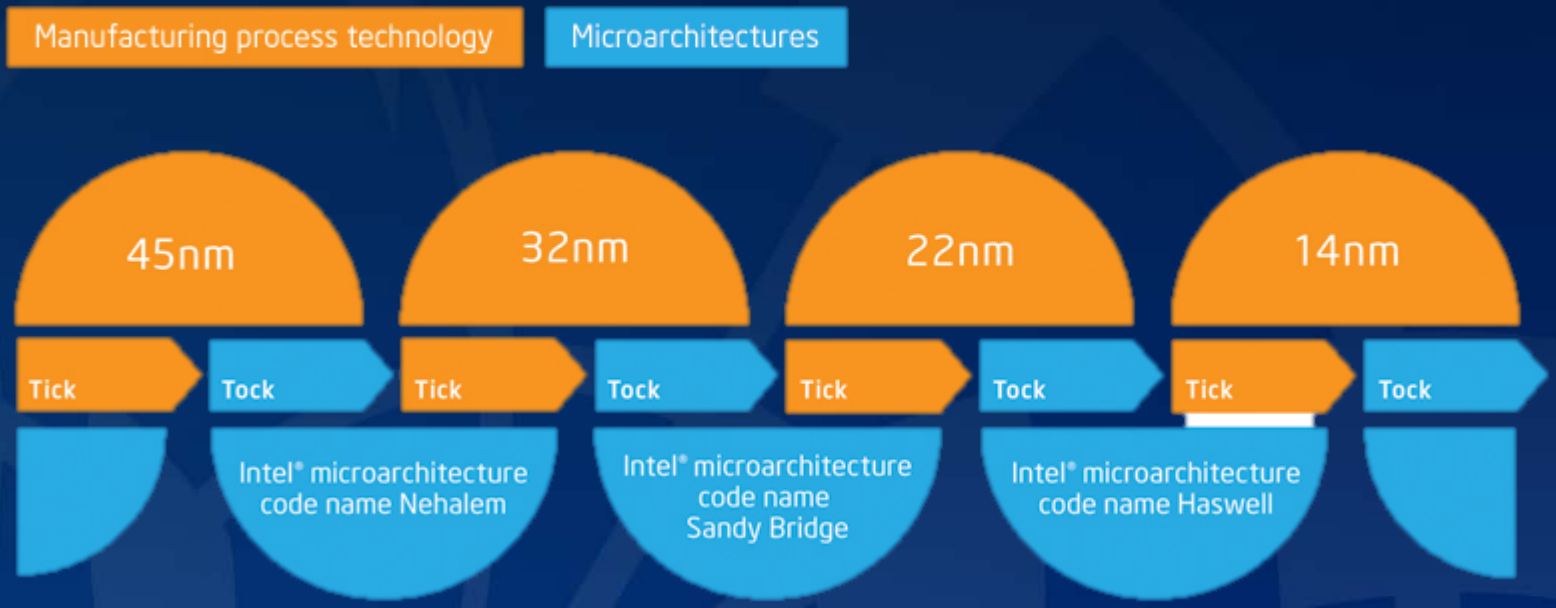
- **256-bit SIMD implementation in x86 processors, successor to SSE**
- **Support in hardware:**
  - Register size increase does not guarantee corresponding performance increases
  - Intel: partly/mostly (optimized workloads scale to 2.5x-3x; penalties for switching to SSE)
  - AMD: one shared execution unit per two cores (“module”)
- **Can scale up to 2048 bits**
  - Will it? Depends on demand
- **Consequence: need to know how well optimized code can do before approaching less optimized code**

AVX - registers

AVX – lanes


LANE DIVISION
PACKED
single
single
single
single
single
single
single
single
SINGLE PRECISION
SCALAR
single
PACKED
double
double
double
double
DOUBLE PRECISION
SCALAR
double

The tick-tock model



Source: Intel

AVX – Haswell roadmap

- **Major changes in the "Haswell" microarchitecture from Intel**
- **3op FMA support**
  - 5 cycle latency!
  - Improved cache bandwidth
  - Result: double FP throughput
- **Integer AVX**
- **Updates in broadcasts, permutes, shifts**

Open questions for AVX

- Proper gather/scatter
- Masks/predication
- Will LRBni and AVX continue independently?

SIMD on MIC

- **Imperative to vectorize for good performance**
- **Currently vector width set at 512 bits, no rumors of increase**
- **Alignment is still important (64B)**
- **Advanced operations include**
  - Predication, Masking
  - Gather, scatter - somewhat inefficient, Intel recommends to avoid
- **Compiler support important**
  - vectorizable math functions
  - inline when needed to vectorize

ARM

- **Neon extensions**
  - 128 bits
  - SSE-like character
  - Cumbersome in execution
- **Improved software support for Neon gives hope**
  - Relevant Linux distributions released recently
  - GCC support (not 100%)

Software aspect

- Vectorization is the way to 2-3x performance increases today
- ICC gives good vectorization reports and is ahead of the game in vectorization, GCC lags behind (in particular in reporting)
- If you must use GCC, consider using ICC just for the reports

THANK YOU

Q & A

Questions? Andrzej.Nowak@cern.ch