# Mobile vs Desktop GPUs
# and
# GPU vs MIMD Multicore CPU
# Advanced Computer Architecture
# COMP 140
# Tuesday June 19, 2014

GPUs are found in both mobile platforms (e.g., phones), and in desktop computers

The goals for mobile and desktop (or server) GPUs are similar, but lag. E.g., in the next year or two, GPU manufactures hope to be able to produce live animation (think *Avitar*) on a desktop GPU, and in five or six years they want to be able to do the same animation on a *mobile* GPU.

# NVIDIA Tegra 4 (CPU+GPU) vs. NVIDIA GTX 780 GPU

| | Tegra 4 | GTX 780 |
|---|---|---|
| **Processor** | | |
| CPU Cores | 4 + 1 | N/A |
| CPU Architecture | ARM Cortex-A15 | N/A |
| Max Clock Speed | 1.9 GHz CPU/ 672MHz GPU | 900MHz GPU |
| **GPU** | | |
| Custom GPU Cores | 72 | 2304 |
| Computational Photography | Yes | No |
| GFlops | 74.8 | 3977 |
| **Memory** | | |
| Memory Bandwidth | 14.9 GB/s | 288.4 GB/s |
| Memory Size | 4 GB | 3GB |
| **Display** | | |
| HDMI | 4K (UltraHD) | 4K (UltraHD) |
| **Die** | | |
| Process | 28 nm | 28 nm |
| Size | 80mm | 561mm |
| Transistors | ? | 7.1 Billion |
| Power | 1.5W | 250W |
| GFLOPS/W | 50 | 16 |

Even though the Tegra 4 has much different specs, by some measures it outperforms the GTX 780!

# The GPU -vs- CPU debate

When GPUs gained compute capability (via CUDA/OpenCL, etc.) around 2008, there was a lot of press regarding the “100x speedup!” that GPUs could provide over CPUs.

Some engineers at Intel decided to do a bit of a deep-dive comparison to see whether this was really the case.

# The GPU -vs- CPU debate

**Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU**

Full Text: PDF


Authors:

| | |
|---|---|
| Victor W. Lee | Intel Corporation, Santa Clara, CA, USA |
| Changkyu Kim | Intel Corporation, Santa Clara, CA, USA |
| Jatin Chhugani | Intel Corporation, Santa Clara, CA, USA |
| Michael Deisher | Intel Corporation, Hillsboro, OR, USA |
| Daehyun Kim | Intel Corporation, Santa Clara, CA, USA |
| Anthony D. Nguyen | Intel Corporation, Santa Clara, CA, USA |
| Nadathur Satish | Intel Corporation, Santa Clara, CA, USA |
| Mikhail Smelyanskiy | Intel Corporation, Santa Clara, CA, USA |
| Srinivas Chennupaty | Intel Corporation, Hillsboro, OR, USA |
| Per Hammarlund | Intel Corporation, Hillsboro, OR, USA |
| Ronak Singhal | Intel Corporation, Hillsboro, OR, USA |
| Pradeep Dubey | Intel Corporation, Santa Clara, CA, USA |


2010 Article

Bibliometrics

- Downloads (6 Weeks): 116
- Downloads (12 Months): 913
- Downloads (cumulative): 17,993
- Citation Count: 73

## The GPU -vs- CPU debate

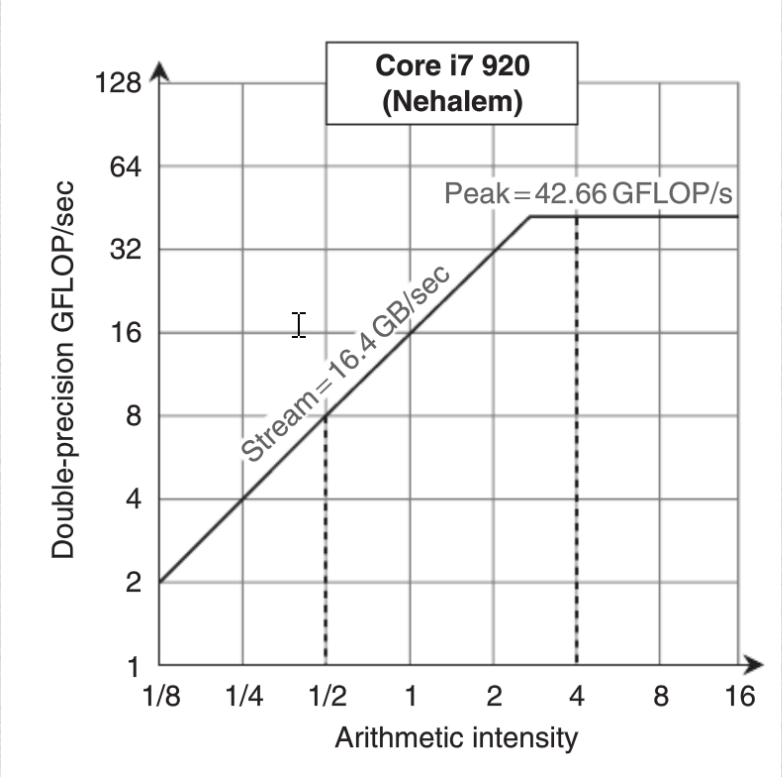
The comparison was between a quad-core Intel i7 and a previous generation (!) NVIDIA Tesla GTX 280 GPU. Both were purchased in 2009.

Although accused of being biased, the Intel paper did make waves, and argued that there was a much, much less speedup advantage (around 2x-3x) when all factors were taken into consideration.

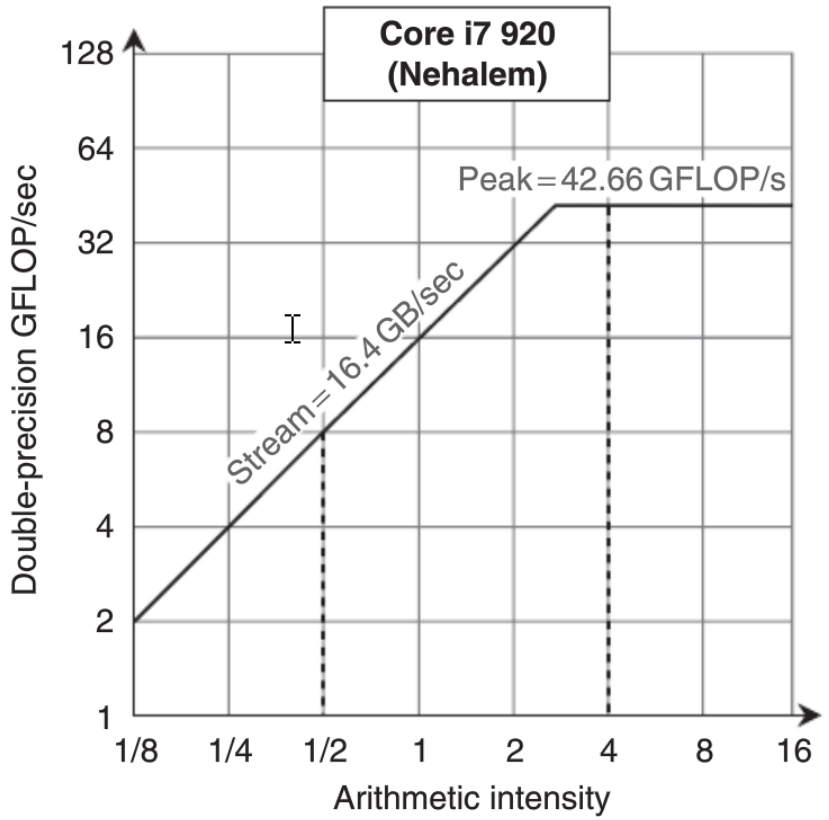
# The GPU -vs- CPU debate

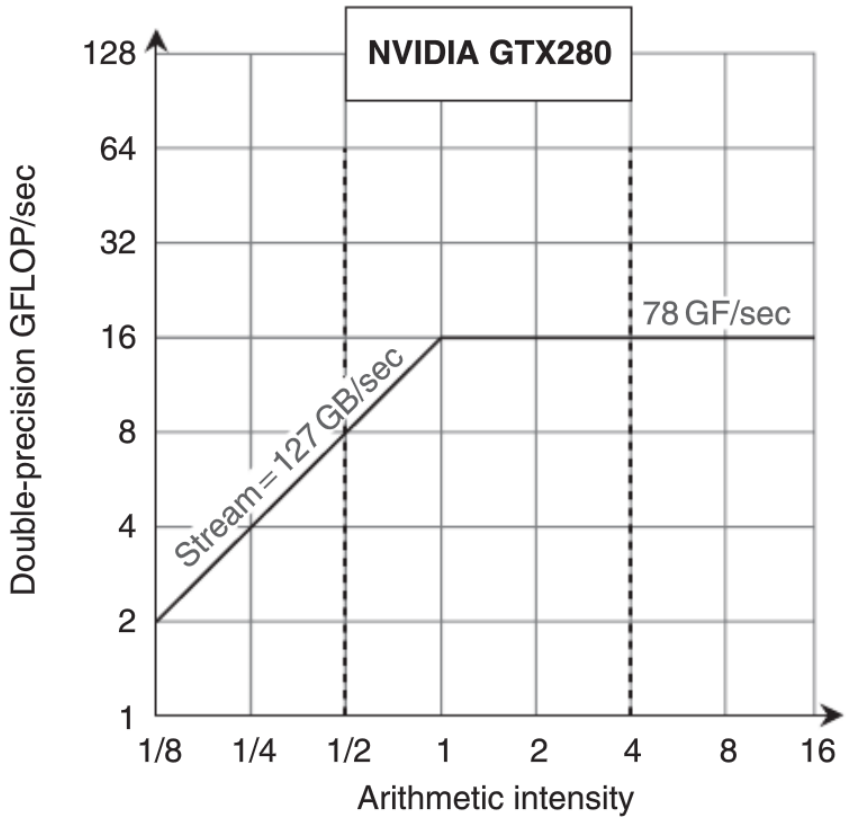
| | Core i7-960 | GTX 280 | GTX 480 | Ratio 280/i7 | Ratio 480/i7 |
|---|---|---|---|---|---|
| Number of processing elements (cores or SMs) | 4 | 30 | 15 | 7.5 | 3.8 |
| Clock frequency (GHz) | 3.2 | 1.3 | 1.4 | 0.41 | 0.44 |
| Die size | 263 | 576 | 520 | 2.2 | 2.0 |
| Technology | Intel 45 nm | TSMC 65 nm | TSMC 40 nm | 1.6 | 1.0 |
| Power (chip, not module) | 130 | 130 | 167 | 1.0 | 1.3 |
| Transistors | 700 M | 1400 M | 3030 M | 2.0 | 4.4 |
| Memory bandwidth (GBytes/sec) | 32 | 141 | 177 | 4.4 | 5.5 |
| Single-precision SIMD width | 4 | 8 | 32 | 2.0 | 8.0 |
| Double-precision SIMD width | 2 | 1 | 16 | 0.5 | 8.0 |
| Peak single-precision scalar FLOPS (GFLOP/Sec) | 26 | 117 | 63 | 4.6 | 2.5 |
| Peak single-precision SIMD FLOPS (GFLOP/Sec) | 102 | 311 to 933 | 515 or 1344 | 3.0–9.1 | 6.6–13.1 |
| (SP 1 add or multiply) | N.A. | (311) | (515) | (3.0) | (6.6) |
| (SP 1 instruction fused multiply-adds) | N.A. | (622) | (1344) | (6.1) | (13.1) |
| (Rare SP dual issue fused multiply-add and multiply) | N.A. | (933) | N.A. | (9.1) | -- |
| Peak double-precision SIMD FLOPS (GFLOP/sec) | 51 | 78 | 515 | 1.5 | 10.1 |

# The Roofline model: http://crd.lbl.gov/assets/pubs_presos/parlab08-roofline-talk.pdf

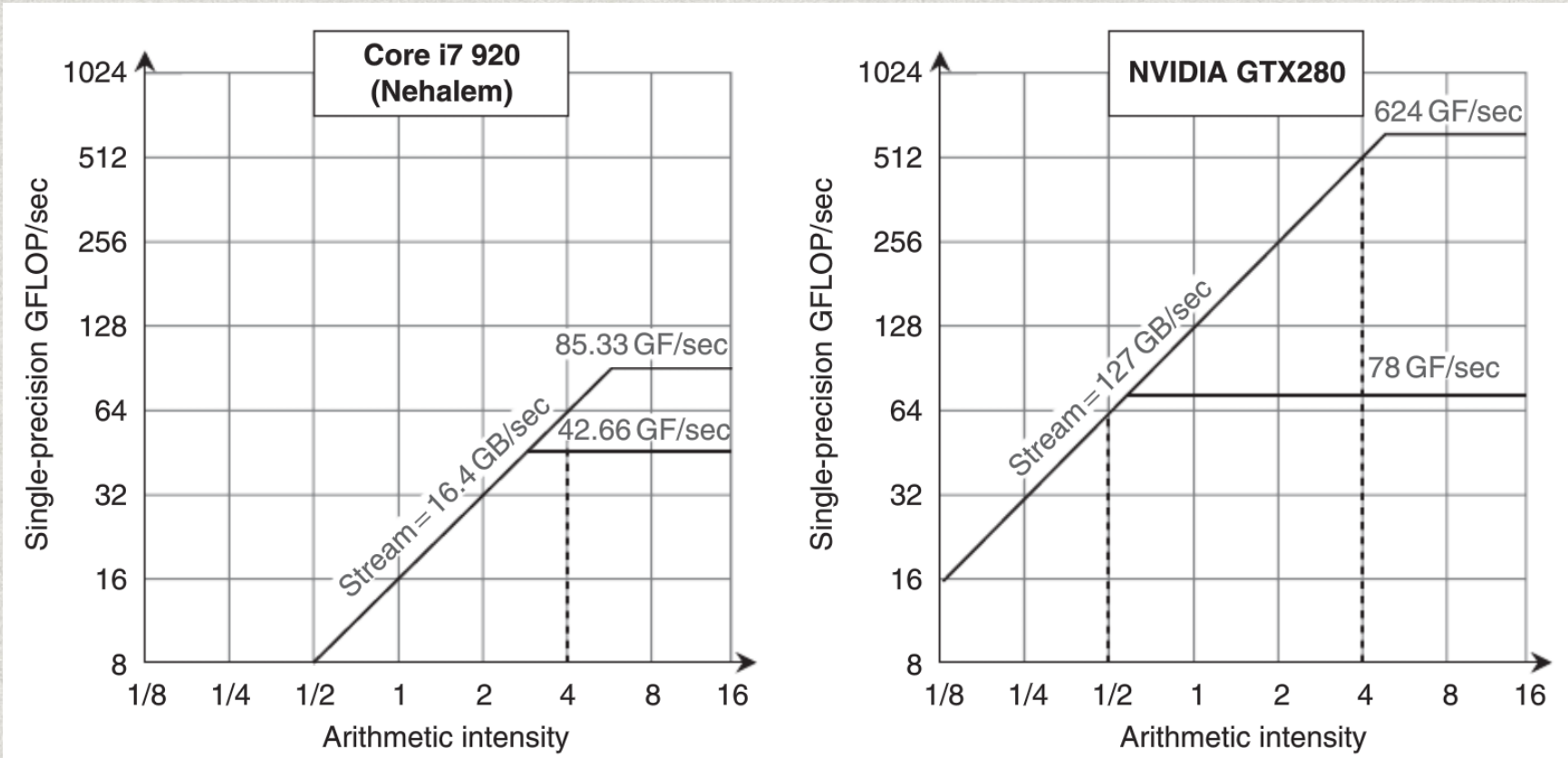


# The GPU -vs- CPU debate



# The GPU -vs- CPU debate



## Compute Bound -vs- Memory Bound

Compute-bound: there are a large number of computations per data element. I.e., memory is *not* the bottleneck.

Memory-bound: there are many data elements per computation. I.e., memory *is* the bottleneck (we would like more computation!).

Atomicity

If a number of operations all want to read from and write to the same memory location (possibly common in a GPU algorithm!) there are “data races” which can corrupt the result of your algorithm.

The algorithm needs to serialize the operations in order to ensure that it calculates the correct answer.

See: https://www.udacity.com/course/viewer#!/c-cs344/l-77202674/m-80064551

See: http://on-demand.gputechconf.com/gtc/2013/presentations/S3101-Atomic-Memory-Operations.pdf

# The GPU -vs- CPU debate

| Kernel | Application | SIMD | TLP | Characteristics |
|---|---|---|---|---|
| SGEMM (**SGEMM**) | Linear algebra | Regular | Across 2D tiles | Compute bound after tiling |
| Monte Carlo (**MC**) | Computational finance | Regular | Across paths | Compute bound |
| Convolution (**Conv**) | Image analysis | Regular | Across pixels | Compute bound; BW bound for small filters |
| FFT (**FFT**) | Signal processing | Regular | Across smaller FFTs | Compute bound or BW bound depending on size |
| SAXPY (**SAXPY**) | Dot product | Regular | Across vector | BW bound for large vectors |
| LBM (**LBM**) | Time migration | Regular | Across cells | BW bound |
| Constraint solver (**Solv**) | Rigid body physics | Gather/Scatter | Across constraints | Synchronization bound |
| SpMV (**SpMV**) | Sparse solver | Gather | Across non-zero | BW bound for typical large matrices |
| GJK (**GJK**) | Collision detection | Gather/Scatter | Across objects | Compute bound |
| Sort (**Sort**) | Database | Gather/Scatter | Across elements | Compute bound |
| Ray casting (**RC**) | Volume rendering | Gather | Across rays | 4-8 MB first level working set; over 500 MB last level working set |
| Search (**Search**) | Database | Gather/Scatter | Across queries | Compute bound for small tree, BW bound at bottom of tree for large tree |
| Histogram (**Hist**) | Image analysis | Requires conflict detection | Across pixels | Reduction/synchronization bound |

# The GPU -vs- CPU debate

| Kernel | Units | Core i7-960 | GTX 280 | GTX 280/ i7-960 |
|---|---|---|---|---|
| SGEMM | GFLOP/sec | 94 | 364 | 3.9 |
| MC | Billion paths/sec | 0.8 | 1.4 | 1.8 |
| Conv | Million pixels/sec | 1250 | 3500 | 2.8 |
| FFT | GFLOP/sec | 71.4 | 213 | 3.0 |
| SAXPY | GBytes/sec | 16.8 | 88.8 | 5.3 |
| LBM | Million lookups/sec | 85 | 426 | 5.0 |
| Solv | Frames/sec | 103 | 52 | 0.5 |
| SpMV | GFLOP/sec | 4.9 | 9.1 | 1.9 |
| GJK | Frames/sec | 67 | 1020 | 15.2 |
| Sort | Million elements/sec | 250 | 198 | 0.8 |
| RC | Frames/sec | 5 | 8.1 | 1.6 |
| Search | Million queries/sec | 50 | 90 | 1.8 |
| Hist | Million pixels/sec | 1517 | 2583 | 1.7 |
| Bilat | Million pixels/sec | 83 | 475 | 5.7 |

**Figure 4.30 Raw and relative performance measured for the two platforms.** In this study, SAXPY is just used as a measure of memory bandwidth, so the right unit is GBytes/sec and not GFLOP/sec. (Based on Table 3 in [Lee et al. 2010].)

Reasons for differences:

**Memory bandwidth**

**Compute bandwidth**

**Cache benefits**

**Gather-Scatter**

**Synchronization**

The successor to the GTX 280 (the GTX 480, or “Fermi”) actually addressed many of the issues brought up in the paper!

**Faster double-precision floating point**

**Atomic Operations**

**Cache**

Another issue brought up by critics:

**The intel engineers did not comment on how much effort they had to put in to optimize the benchmarks for the i7 vs the GTX 280. I.e., programmer effort is an important metric!**

In the end, Intel was successful in showing that claims of 100x speedup were exaggerated. But, they still admitted to speedups of, on average 2x to 3x. Still not bad for comparably priced hardware!