Future Computer & Programming Trends

Axel Huebl
Lawrence Berkeley National Laboratory, U.S.

Snowmass Community Planning Meeting
Session 64: Computing Needs of the Accelerator Frontier – Oct 6th, 2020
48 Years of Microprocessor Trend Data

A100 GPU: 54 billion transistors

Transistors (thousands)

Single-Thread Performance (SpecINT x 10^3)

Frequency (MHz)

Typical Power (Watts)

Number of Logical Cores

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/
Power Consumption: HPCG Benchmark

Growing Divide: Data & Compute

John Backus, 1977 ACM Turing Award: „much of that traffic concerns not significant data itself, but where to find it“
Emerging Memory to Interconnect

L1 cache reference ......................... 0.5 ns
Branch mispredict ............................ 5 ns
L2 cache reference ........................... 7 ns
Mutex lock/unlock ........................... 25 ns
Main memory reference ...................... 100 ns
Compress 1K bytes with Zippy ............. 3,000 ns = 3 µs
Send 2K bytes over 1 Gbps network ...... 20,000 ns = 20 µs
SSD random read .............................. 150,000 ns = 150 µs
Read 1 MB sequentially from memory .... 250,000 ns = 250 µs
Round trip within same datacenter ..... 500,000 ns = 0.5 ms
Read 1 MB sequentially from SSD* ...... 1,000,000 ns = 1 ms
Disk seek .................................. 10,000,000 ns = 10 ms
Read 1 MB sequentially from disk .... 20,000,000 ns = 20 ms
Send packet CA->Netherlands->CA .... 150,000,000 ns = 150 ms

IMEC: https://semiengineering.com/a-new-memory-contender/
Today and Near-Term: ~5 years

- **Parallelism:** nodes $\rightarrow$ devices $\rightarrow$ rings/SMs $\rightarrow$ cores $\rightarrow$ (hyper)threads $\rightarrow$ SIMD-steps... *on* local $\rightarrow$ shared $\rightarrow$ cached $\rightarrow$ global $\rightarrow$ remote memory

- **Hardware Specialization**
  - SIMD: vector to matrix-processing units (tensor cores)
  - whole device:
    - RISC:
      - GPUs: massive parallelism
      - ARM / RISC-V / NEC
    - FPGAs, DSPs, ...
    - ASICs; ANTON2 (2008 $\rightarrow$ 2014)

- **Algorithmic Specialization**
  - multi-level parallelism; in situ algorithms
Mid- to Long term: >5-10 years

- Further Specialization
  - Programmable FPGAs from *high-level languages* ("HLS")
  - on-*socket* integration of “<5 year” hardware
  - *workload-specific* memory & system *designs*

- Programming Models
  - *Parallelism* will only rise: width and depth
  - C++23 et al.: unification of many of today’s capabilities
  - Emerging *new paradigms* likely – for *abstract compute*

- Potentially non-von Neumann *components*
  - First signs: FPGAs, DSPs, memory-driven algorithms, neuromorphic chips, ...
Beyond von-Neumann Architecture

Characteristics, e.g.:
- w/o sequential flow of control
- w/o the concept of a named storage variable

Programming examples (non-procedural):
- declarative (properties)
- data-driven (DSP, analog, quantum gates)

Potential Routes for Engagement

- **Programming Models**: Need continued community engagement
  - Describe and publish our algorithms and codes
  - Re-design and adopt to industry trends
  - Propose, influence and refine with scientific use-cases

- **Algorithms**: how could a Poisson-solve, PIC-push, advection-diffusion, beam-transport, QED processes be modeled with “X”?

- **Leave comfort zones**: efforts across natural sciences & engineering

- **Adopt**: codes, languages, mental models, unexpected abstractions, ...
References

- Future Technologies Group (OLCF)

- Supercomputing Conference Panels: Beyond Von Neumann, Neuromorphic Systems and Architectures


- Intel oneAPI: FPGA; SPCL (ETH Zuerich): FPGA High-Level Synthesis
  DARPA: IDEA/POSH Universal Hardware Compiler

- The Networking & Information Technology Research & Development Program (NITRD), nitrd.gov

- TOP500.org  hpcg-benchmark.org

- Blogs & online publishing: karlrupp.net  plasma.ninja/blog
  hpcwire.com  thenextplatform.com
Backup
Today and Near-Term Programming Models

- Dominating HPC / Industry Programming Models
  - trend towards mixed-functional programming
    - functional
    - declarative over prescriptive
    - array-oriented/declarative: unclear future
  - C++-based: zero-overhead abstractions, standardization
  - C/Fortran: slow adoption, few compilers
  - Gap bridged with compiler directives
  - Some continue to be adopted into JIT (Python, Julia, …)
    - scalability (latencies; duplication or broadcasts)
  - Emerging HW: low-level (VERILOG, Assembly, …) or special-purpose DSLs
“It is possible to **abstract** and simplify much of the complexity so that programmers can utilize simpler models of a system and increase their productivity. Useful **software abstractions** improve developer productivity and reduce execution risk by muting the cognitive noise produced by complexity.” ≠ **traditional levels of abstraction**

“**A New Golden Age for Computer Architecture**”
ACM Turing Award laureates John Hennessy and David Patterson:

“**The next decade will see a Cambrian explosion of novel computer architectures, meaning exciting times for computer architects in academia and industry.**”

https://cacm.acm.org/magazines/2019/2/234352-a-new-golden-age-for-computer-architecture/fulltext
NVIDIA:

“Looking forward, the practical realization of slowing technology scaling will likely require a range of approaches including: (1) architectures that incorporate increasingly specialized accelerator hardware; (2) packaging, signaling, and interconnect technologies that enable greater scaling at both the “node” and “system” level; (3) novel devices (e.g., carbon nanotube FETs) that can provide smaller digital devices at lower power; and (4) novel computing technologies such as analog, quantum, and neuromorphic that may require fundamental changes to algorithms.”
# Semiconductor Manufacturing

## Number of Semiconductor Manufacturers with a Cutting Edge Logic Fab

<table>
<thead>
<tr>
<th>Company</th>
<th>Number of Manufacturers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiTerra</td>
<td>10</td>
</tr>
<tr>
<td>X-FAB</td>
<td>10</td>
</tr>
<tr>
<td>Dongbu HiTek</td>
<td>10</td>
</tr>
<tr>
<td>ADI</td>
<td>10</td>
</tr>
<tr>
<td>Atmel</td>
<td>10</td>
</tr>
<tr>
<td>Rohm</td>
<td>10</td>
</tr>
<tr>
<td>Sanyo</td>
<td>10</td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>10</td>
</tr>
<tr>
<td>ON</td>
<td>10</td>
</tr>
<tr>
<td>Hitachi</td>
<td>10</td>
</tr>
<tr>
<td>Cypress</td>
<td>10</td>
</tr>
<tr>
<td>Sony</td>
<td>10</td>
</tr>
<tr>
<td>Infineon</td>
<td>10</td>
</tr>
<tr>
<td>Sharp</td>
<td>10</td>
</tr>
<tr>
<td>Freescale</td>
<td>10</td>
</tr>
<tr>
<td>Renesas (NEC)</td>
<td>10</td>
</tr>
<tr>
<td>Toshiba</td>
<td>10</td>
</tr>
<tr>
<td>Fujitsu</td>
<td>10</td>
</tr>
<tr>
<td>TI</td>
<td>10</td>
</tr>
<tr>
<td>Panasonic</td>
<td>10</td>
</tr>
<tr>
<td>STMicroelectronics</td>
<td>10</td>
</tr>
<tr>
<td>HLGIC</td>
<td>10</td>
</tr>
<tr>
<td>UMC</td>
<td>10</td>
</tr>
<tr>
<td>IBM</td>
<td>10</td>
</tr>
<tr>
<td>SMIC</td>
<td>10</td>
</tr>
<tr>
<td>AMD</td>
<td>10</td>
</tr>
<tr>
<td>Samsung</td>
<td>10</td>
</tr>
<tr>
<td>TSMC</td>
<td>10</td>
</tr>
<tr>
<td>Intel</td>
<td>10</td>
</tr>
</tbody>
</table>

### Technology Nodes

- 180 nm
- 130 nm
- 90 nm
- 65 nm
- 45 nm/40 nm
- 32 nm/28 nm
- 22 nm/20 nm
- 16 nm/14 nm
- 10 nm
- 7 nm
- 5 nm

[https://en.wikichip.org/wiki/technology_node](https://en.wikichip.org/wiki/technology_node)