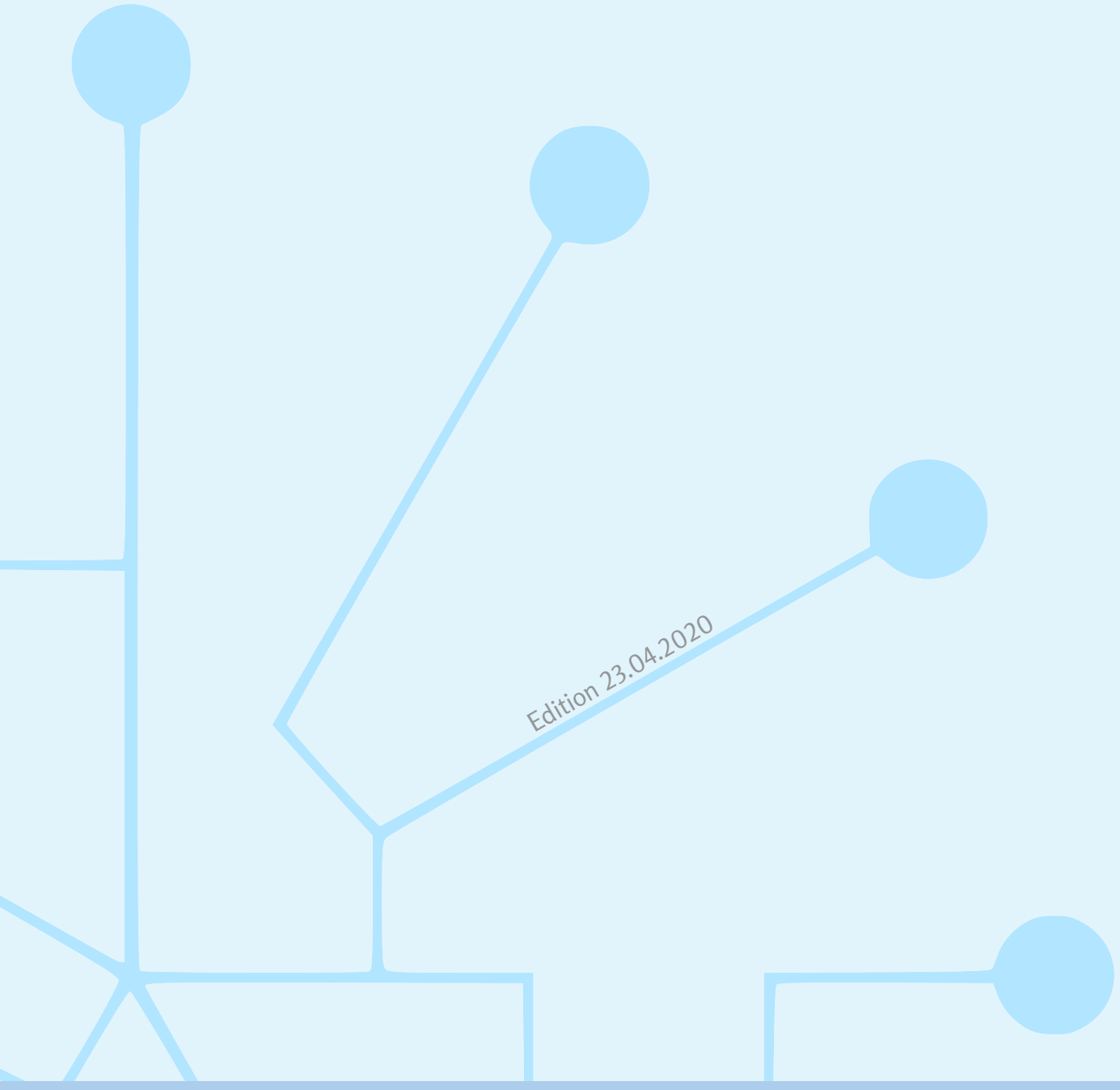
Foundation for a Centre of Excellence in
High-Performance Computing Systems



Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition)

Editors: Theo Ungerer, Paul Carpenter

Funded by the European Union Horizon 2020 Framework Programme (H2020-EU.1.2.2. - FET Proactive)

Overall Editors and Authors

Prof. Dr. Theo Ungerer, University of Augsburg
Dr. Paul Carpenter, BSC, Barcelona

Authors

Sandro Bartolini	University of Siena	Photonics, Single Source Programming Models
Luca Benini	ETH Zürich	Die Stacking and 3D-Chips
Koen Bertels	Delft University of Technology	Quantum Computing
Spyros Blanas	The Ohio State University	Data Management
Uwe Brinkschulte	Goethe University Frankfurt	Memory Hierarchy
Paul Carpenter	BSC	Overall Organization and Diverse Sections
Giovanni De Micheli	EPFL	Nanowires, Superconducting Electronics, 2D electronics
Marc Duranton	CEA LIST DACLE	Overall Comments, Reviewing
Babak Falsafi	EPFL	Data Centers, Cloud Computing, Heterogeneous Systems
Dietmar Fey	University of Erlangen-Nuremberg	Memristors, Resistive Computing
Said Hamdioui	Delft University of Technology	Near- and In-Memory Computing, Resistive Computing
Christian Hochberger	Technical University of Darmstadt	Memory Hierarchy, Nanotubes
Avi Mendelson	Technion	Hardware Impact
Dominik Meyer	Helmut-Schmidt-Universität Hamburg	Reconfigurable Computing
Ilia Polian	University of Stuttgart	Security and Privacy
Ulrich Rückert	University of Bielefeld	Neuromorphic Computing
Xavier Salazar	BSC	Overall Organization and Related Initiatives
Werner Schindler	Bundesamt für Sicherheit in der Informationstechnik (BSI)	Security and Privacy
Per Stenstrom	Chalmers University	Reviewing
Theo Ungerer	University of Augsburg	Overall Organization and Diverse Sections

Compiled by

Florian Haas	University of Augsburg

We also acknowledge the numerous people that provided valuable feedback at the roadmapping workshops at HiPEAC CSW and HPC Summit, to HiPEAC and EXDCI for hosting the workshops and Xavier Salazar for the organizational support.

Executive Summary

Radical changes in computing are foreseen for the current decade. The US IEEE society wants to "reboot computing" and the HiPEAC Visions of 2017 and 2019 see the time to "re-invent computing", both by challenging its basic assumptions. This document presents the second edition of the "EuroLab4HPC Long-Term Vision on High-Performance Computing" of January 2020[1], a road mapping effort within the EC CSA[2] EuroLab4HPC that targets potential changes in hardware, software, and applications in High-Performance Computing (HPC).

The objective of the EuroLab4HPC Vision is to provide a long-term roadmap from 2023 to 2030 for High-Performance Computing (HPC). Because of the long-term perspective and its speculative nature, the authors started with an assessment of future computing technologies that could influence HPC hardware and software. The proposal on research topics is derived from the report and discussions within the road mapping expert group. We prefer the term "vision" over "roadmap", firstly because timings are hard to predict given the long-term perspective, and secondly because EuroLab4HPC will have no direct control over the realization of its vision.

The Big Picture

High-performance computing (HPC) typically targets scientific and engineering simulations with numerical programs mostly based on floating-point computations. We expect the continued scaling of such scientific and engineering applications to continue well beyond Exascale computers.

However, three trends are changing the landscape for high-performance computing and supercomputers. The first trend is the emergence of data analytics complementing simulation in scientific discovery. While simulation still remains a major pillar for science, there are massive volumes of scientific data that are now gathered by sensors augmenting data from simulation available for analysis. High-Performance Data Analysis (HPDA) will complement simulation in future HPC applications.

The second trend is the emergence of cloud computing and warehouse-scale computers (also known as data centres). Data centres consist of low-cost volume processing, networking and storage servers, aiming at cost-effective data manipulation at unprecedented scales. The scale at which they host and manipulate (e.g., personal, business) data has led to fundamental breakthroughs in data analytics.

There are a myriad of challenges facing massive data analytics including management of highly distributed data sources, and tracking of data provenance, data validation, mitigating sampling bias and heterogeneity, data format diversity and integrity, integration, security, privacy, sharing, visualization, and massively parallel and distributed algorithms for incremental and/or real-time analysis.

Large datacentres are fundamentally different from traditional supercomputers in their design, operation and software structures. Particularly, big data applications in data centres and cloud computing centres require different algorithms and differ significantly from traditional HPC applications such that they may not require the same computer structures.

With modern HPC platforms being increasingly built using volume servers (i.e. one server = one role), there are a number of features that are shared among warehouse-scale computers and modern HPC platforms, including dynamic resource allocation and management, high utilization, parallelization and acceleration, robustness and infrastructure costs. These shared concerns will serve as incentives for the convergence of the platforms.

There are, meanwhile, a number of ways that traditional HPC systems differ from modern warehouse-scale computers: efficient virtualization, adverse network topologies and fabrics in cloud platforms, low memory and storage bandwidth in volume servers.

[1] https://www.eurolab4hpc.eu/vision/
[2] European Commission Community and Support Action

HPC customers must adapt to co-exist with cloud services; warehouse-scale computer operators must innovate technologies to support the workload and platform at the intersection of commercial and scientific computing.

It is unclear whether a convergence of HPC with big data applications will arise. Investigating hardware and software structures targeting such a convergence is of high research and commercial interest. However, some HPC applications will be executed more economically on data centres. Exascale and post-Exascale supercomputers could become a niche for HPC applications.

The third trend arises from Artificial Intelligence (AI) and Deep Neural Networks (DNN) for back propagation learning of complex patterns, which emerged as new techniques penetrating different application areas. DNN learning requires high performance and is often run on high-performance supercomputers. GPU accelerators are seen as very effective for DNN computing by their enhancements, e.g. support for 16-bit floating-point and tensor processing units. It is widely assumed that it will be applied in future autonomous cars thus opening a very large market segment for embedded HPC. DNNs will also be applied in engineering simulations traditionally running on HPC supercomputers.

Embedded high-performance computing demands are upcoming needs. It may concern smartphones but also applications like autonomous driving, requiring on-board high-performance computers. In particular the trend from current advanced ADAS (automatic driving assistant systems) to piloted driving and to fully autonomous cars will increase on-board performance requirements and may even be coupled with high-performance servers in the Cloud. The target is to develop systems that adapt more quickly to changing environments, opening the door to highly automated and autonomous transport, capable of eliminating human error in control, guidance and navigation and so leading to more safety. High-performance computing devices in cyber-physical systems will have to fulfil further non-functional requirements such as timeliness, (very) low energy consumption, security and safety. However, further applications will emerge that may be unknown today or that receive a much higher importance than expected today.

Power and thermal management is considered as highly important and will continue its preference in future. Post-Exascale computers will target more than 1 Exaflops with less than 30 MW power consumption requiring processors with a much better performance per Watt than available today. On the other side, embedded computing needs high performance with low energy consumption. The power target at the hardware level is widely the same, a high performance per Watt.

In addition to mastering the technical challenges, reducing the environmental impact of upcoming computing infrastructures is also an important matter. Reducing CO_2 emissions and overall power consumption should be pursued. A combination of hardware techniques, such as new processor cores, accelerators, memory and interconnect technologies, and software techniques for energy and power management will need to be cooperatively deployed in order to deliver energy-efficient solutions.

Because of the foreseeable end of CMOS scaling, new technologies are under development, such as, for example, 3D Chip Technologies, Non-volatile Memory (NVM) Technologies, Photonics, Resistive Computing, Neuromorphic Computing, Quantum Computing, and Nanotubes. Since it is uncertain if/when some of the technologies will mature, it is hard to predict which ones will prevail.

The particular mix of technologies that achieve commercial success will strongly impact the hardware and software architectures of future HPC systems, in particular the processor logic itself, the (deeper) memory hierarchy, and new heterogeneous accelerators.

There is a clear trend towards more complex systems, which is expected to continue over the current decade. These developments will significantly increase software complexity, demanding more and more intelligence across the programming environment, including compiler, run-time and tool intelligence driven by appropriate programming models. Manual optimization of the data layout, placement, and caching will become uneconomic and time consuming, and will, in any case, soon exceed the abilities of the best human programmers.

If accurate results are not necessarily needed, another speedup could emerge from more efficient special execution units, based on analog, or even a mix between analog and digital technologies. Such developments would benefit from more advanced ways to reason about the permissible degree of inaccuracy in calculations at run time. Furthermore, new memory

technologies like memristors may allow on-chip integration, enabling tightly-coupled communication between the memory and the processing unit. With the help of memory computing algorithms, data could be pre-processed "in-" or "near-" memory.

The adoption of neuromorphic, resistive and/or quantum computing as new accelerators may have a dramatic effect on the system software and programming models. It is currently unclear whether it will be sufficient to offload tasks, as on GPUs, or whether more dramatic changes will be needed. By 2030, disruptive technologies may have forced the introduction of new and currently unknown abstractions that are very different from today. Such new programming abstractions may include domain-specific languages that provide greater opportunities for automatic optimization. Automatic optimization requires advanced techniques in the compiler and runtime system. We also need ways to express non-functional properties of software in order to trade various metrics: performance vs. energy, or accuracy vs. cost, both of which may become more relevant with near threshold, approximate computing or accelerators.

But it is also possible that new hardware developments reduce software complexity e.g. by reducing parallelism and its burden. New materials could be used to run processors at much higher frequencies than currently possible, and with that, may even enable a significant increase in the performance of single-threaded programs.

Optical networks on die and Terahertz-based connections may eliminate the need for preserving locality since the access time to local storage may not be as significant in future as it is today. Such advancements will lead to storage-class memory, which features similar speed, addressability and cost as DRAM combined with the non-volatility of storage. In the context of HPC, such memory may reduce the cost of checkpointing or eliminate it entirely.

Nevertheless, today's abstractions will continue to evolve incrementally and will continue to be used well beyond 2030, since scientific codebases have very long lifetimes, on the order of decades.

Execution environments will increase in complexity requiring more intelligence, e.g., to manage, analyse and debug millions of parallel threads running on heterogeneous hardware with a diversity of accelerators, while dynamically adapting to failures and performance variability. Spotting anomalous behavior may be viewed as a big data problem, requiring techniques from data mining, clustering and structure detection. This requires an evolution of the incumbent standards such as OpenMP to provide higher-level abstractions. An important question is whether and to what degree these fundamental abstractions may be impacted by disruptive technologies.

The Work Needed

As new technologies require major changes across the stack, a vertical funding approach is needed, from applications and software systems through to new hardware architectures and potentially down to the enabling technologies. We see HP Lab's memory-driven computing architecture "The Machine" as an exemplary project that proposes a low-latency NVM (Non-Volatile Memory) based memory connected by photonics to processor cores. Projects could be based on multiple new technologies and similarly explore hardware and software structures and potential applications. Required research will be interdisciplinary. Stakeholders will come from academic and industrial research.

The Opportunity

The opportunity may be development of competitive new hardware/software technologies based on upcoming new technologies to advantageous position European industry for the future. Target areas could be High-Performance Computing and Embedded High-Performance devices. The drawback could be that the chosen base technology may not be prevailing but be replaced by a different technology. For this reason, efforts should be made to ensure that aspects of the developed hardware architectures, system architectures and software systems could also be applied to alternative technologies. For instance, several NVM technologies will bring up new memory devices that are several magnitudes faster than current Flash technology and the developed system structures may easily be adapted to specific technologies, even if the project has chosen a different NVM technology as basis.

EC Funding Proposals

The Eurolab4HPC vision recommends the following funding opportunities for topics beyond Horizon 2020

(ICT):

- Convergence of HPC and HPDA:
  - Data Science, Cloud computing and HPC: Big Data meets HPC
  - Inter-operability and integration
  - Limitations of clouds for HPC
  - Edge Computing: local computation for processing near sensors
- Impact of new NVMs:
  - Memory hierarchies based on new NVMs
  - Near- and in-memory processing: pre- and post-processing in (non-volatile) memory
  - HPC system software based on new memory hierarchies
  - Impact on checkpointing and reciliency
- Programmability:
  - Hide new memory layers and HW accelerators from users by abstractions
  - Managing the increasingly complex software and programming environments
  - Monitoring of a trillion threads
  - Algorithm-based fault tolerance techniques within the application as well as moving fault detection burden to the library, e.g. fault-tolerant message-passing library
- Green ICT and Energy
  - Integration of cooling and electrical subsystem
  - Supercomputer as a whole system for Green ICT

As remarked above, projects should be interdisciplinary, from applications and software systems through hardware architectures and, where relevant, enabling hardware technologies.

Contents

1 Introduction

Upcoming application trends and disruptive VLSI technologies will change the way computers will be programmed and used as well as the way computers will be designed. New application trends such as High-Performance Data Analysis (HPDA) and deep-learning will induce changes in High-Performance Computing; disruptive technologies will change the memory hierarchy, hardware accelerators and even potentially lead to new ways of computing. The HiPEAC Visions of 2017 and 2019[1] see the time to revisit the basic concepts: The US wants to "reboot computing", the HiPEAC Vision proposes to "re-invent computing" by challenging basic assumptions such as binary coding, interrupts, layers of memory, storage and computation.

This document has been funded by the EC CSA Eurolab4HPC-2 project (June 2018 - May 2020), a successor of EC CSA Eurolab4HPC (Sept. 2015 – August 2017) project. It outlines a long-term vision for excellence in European High-Performance Computing research, with a timescale beyond Exascale computers, i.e. a timespan of approximately 2023-2030. It delivers a thorough update of the Eurolab4HPC Vision of 2017[2]. An intermediate step between the Visions of 2017 and the current one of January 2020 has been reached by the Memristor Report[3] compiled by an expert group of the two German computer science associations "Gesellschaft für Informatik" and "Informationstechnische Gesellschaft" in June 2019.

1.1 Related Initiatives within the European Union

Nowadays the European effort is driven by the EuroHPC Joint Undertaking[4]. The entity started operations in November 2018, with the main objectives of developing a pan-European supercomputing infrastructure and supporting research and innovation activities related to HPC.

The Eurolab4HPC vision complements existing efforts such as the ETP4HPC Strategic Research Agenda (SRA). ETP4HPC is an industry-led initiative to build a globally competitive HPC system value chain. Development of the Eurolab4HPC vision is aligned with ETP4HPC SRA in its latest version from September 2017. SRA 2017 was targeting a roadmap towards Exascale computers that spans until approximately 2022, whereas the new SRA 2019/2020 is expected to cover 2021-2027 as it was advanced in the 'Blueprint for the new Strategic Research Agenda for High Performance Computing'[5] published in April 2019. The Eurolab4HPC visions target the speculative period beyond Exascale, so approximately beyond 2023-2030 and from a technology push point of view.

The Eurolab4HPC vision also complements the PRACE Scientific Case[6] that has the main focus in the future of HPC applications for the scientific and industrial communities. Its 3rd edition covers the timeframe 2018-2026. PRACE (Partnership for Advanced Computing in Europe) is the main public European providers of HPC infrastructure for scientific discovery. On the applications side, a set of Centers of Excellence on HPC Applications have been promoted by the European Commission to stimulate the adoption of HPC technologies among a variety of end-user communities. Those are crucial in the co-design of the future disruptive upstream technologies.

The Eurolab4HPC vision is developed in close collaboration with the "HiPEAC Vision" of HiPEAC CSA that features the broader area of "High Performance and Embedded Architecture and Compilation". The Eurolab4HPC vision complements the HiPEAC Vision 2019 document with a stronger focus on disruptive technologies and HPC.

[1] www.hipeac.net/publications/vision
[2] https://www.eurolab4hpc.eu/vision/
[3] https://fb-ti.gi.de/fileadmin/FB/TI/user_upload/Memristor_Report-2019-06-27.pdf
[4] https://eurohpc-ju.europa.eu/
[5] https://www.etp4hpc.eu/pujades/files/Blueprint%20document_20190904.pdf
[6] http://www.prace-ri.eu/third-scientific-case/

The creation and growth of an HPC ecosystem has been supported by European Commission and structured by CSA[7] instrument. Eurolab4HPC represents the most relevant HPC system experts in academia. Most of current research and development projects and innovation initiatives are led or participated by Eurolab4HPC members, who are individuals committed to strengthen the network. EXDCI's (respectively EXDCI2's) main partners are PRACE and ETP4HPC, thus representing main HPC stakeholders such as infrastructure providers and industry. On the HPC application side, FocusCoE is the initiative aiming to support the Centers of Excellence in HPC applications.

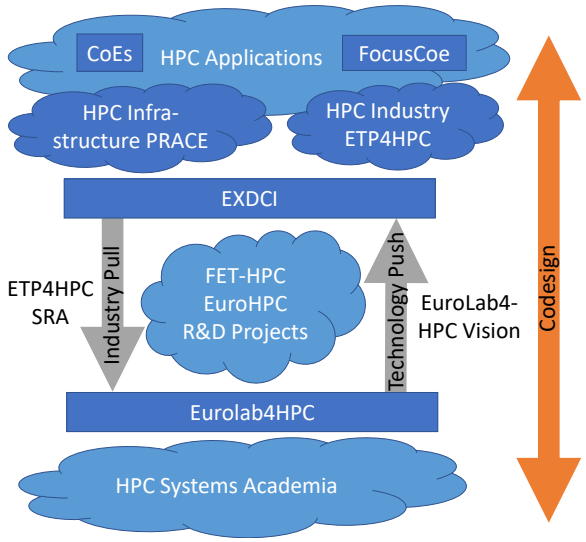


Figure 1.1: Our View on EC Initiatives

1.2 Working Towards the Revised Eurolab4HPC Vision

The second edition of the Eurolab4HPC vision has been developed as a research roadmap with a longer-term time-window. Since the beginning, it has been our target to stick to technical matters and provide an academic research perspective. Because targeting the post-Exascale era with a horizon of approximately 2023 – 2030 will be highly speculative, we proceeded as follows:

1. Select disruptive technologies that may be technologically feasible in the next decade.
2. Assess the potential hardware architectures and their characteristics.
3. Assess what that could mean for the different HPC aspects.

The vision roughly follows the structure: "*IF* technology ready *THEN* foreseeable impact could be ..."

The second edition of the Vision updates the sections of the first edition and restructures the complete document.

The new Vision was again developed by a single *expert working group*:
Sandro Bartolini, Luca Benini, Koen Bertels, Spyros Blanas, Uwe Brinkschulte, Paul Carpenter, Giovanni De Micheli, Marc Duranton, Babak Falsafi, Dietmar Fey, Said Hamdioui, Christian Hochberger, Avi Mendelson, Dominik Meyer, Ilia Polian, Ulrich Rückert, Xavier Salazar, Werner Schindler, and Theo Ungerer. Simon McIntosh–Smith and Igor Zacharov, both contributing to the first edition, were no longer available.

The working schedule for the second edition was:

- May 2018 until May 2019 disseminating the first edition and collecting input for the new Vision.
- May 2019 Kickoff Telco of expert working group
- September 23, 2019: one day Roadmap meeting of expert working group at University of Augsburg
- October until December 2019: Experts prepare inputs
- January 2020: First Vision public
- February until March 2020: Internal reviewing and final version
- January until May 2020: Vision dissemination and discussion

1.3 Document Structure

The rest of this document is structured as follows: The next section provides an overall view of HPC. After defining HPC the section covers data centres and cloud computing, eventually leading to a convergence of HPC and HPDA, as well as applications and ecosystem on open source hardware.

Section 3 focuses on Technologies, i.e. silicon-based (CMOS scaling and 3D-chips), memristor-based, and

[7] https://ec.europa.eu/research/participants/data/ref/h2020/other/wp/2018-2020/annexes/h2020-wp1820-annex-d-csa_en.pdf

non-silicon-based (Photonics, Quantum Computing and beyond CMOS) technologies. This section is followed by section 4 that summarizes the Potential Long-Term Impacts of Disruptive Technologies for HPC Hardware and Software in separate subsections.

Section 5 covers System Software and Programming Environment challenges, and finally Section 6 covers Green ICT, Resiliency and Security and Privacy as Vertical Challenges.

2 Overall View of HPC

2.1 HPC and Exascale

The Eurolab4HPC-2 Vision targets particularly technology, architecture and software of postExascale HPC computers, i.e. a period of 2023 til 2030.

The supercomputers of highest performance are listed in the Top 500 Lists[1] which is updated twice a year. The November 2019 shows two IBM-built supercomputers, Summit (148.6 petaflops) and Sierra (94.6 petaflops) in the first two positions, followed by the Chinese supercomputers Sunway TaihuLight (93.0 petaflops) and the Tianhe-2A (Milky Way-2A) (61.4 petaflops). Performance assessments are based on the 'best' performance LINPACK R_{max} as measured by the LINPACK Benchmark. A second list is provided based on the High-Performance Conjugate Gradient (HPCG) Benchmark featuring again Summit and Sierra on top and the Japanese K computer third.

All these performance data is based on pure petaflops measurements which is deemed to be too restricted for useful future supercomputers. Exascale does not merely refer to a LINPACK R_{max} of 1 exaflops. The PathForward definition of a capable Exascale system is focused on scientific problems rather than benchmarks, as well as raising the core challenges of power consumption and resiliency: "a supercomputer that can solve science problems 50X faster (or more complex) than on the 20 Petaflop systems (Titan and Sequoia) of 2016 in a power envelope of 20-30 megawatts, and is sufficiently resilient that user intervention due to hardware or system faults is on the order of a week on average" [1]. Lastly Exascale computing refers to computing systems capable of at least one exaflops, however, on real applications, not just benchmarks. Such applications comprise not only traditional supercomputer applications, but also neural network learning applications and interconnections with HPDA.

[1] https://www.top500.org/lists/

2.2 Current Proposals for Exascale Machines

USA: The U.S. push towards Exascale is led by the DoE's Exascale Computing Project[2] and its extended PathForward program landing in the 2021 – 2022 timeframe with "at least one" Exascale system. This roadmap was confirmed in June 2017 with a DoE announcement that backs six HPC companies as they create the elements for next-generation systems. The vendors on this list include Intel, Nvidia, Cray, IBM, AMD, and Hewlett Packard Enterprise (HPE) [2].

The Department of Energy's new supercomputer Aurora will be built by Intel and Cray at Argonne, and it will be the first of its kind in the United States with costs of estimated 500 million Dollar. It is scheduled to be fully operational by the end of 2021. Aurora will also be set up as a perfect platform for deep learning [3]. At the Exascale Day October 21, 2019 it was revealed that Aurora shall be based on Next-Gen-Xeon processors and Intel-Xe-GPUs. Two even faster supercomputers with 1.5 exaflops are in the line for 2021 and 2022: Frontier based on AMD Next-Gen-Epyc processors with Radeon graphic cards and El Captain with up to now unknown hardware [4].

China has a good chance of reaching exascale computing already in 2020. China's currently fastest listed supercomputer, the Sunway TaihuLight contains entirely Chinese-made processing chips. The Chinese government is funding three separate architectural paths to attain that exascale milestone. This internal competition will put the National University of Defense Technology (NUDT), the National Research Center of Parallel Computer and Sugon (formerly Dawning) against one another to come up with the country's (and perhaps the world's) first exascale supercomputer. Each vendor has developed and deployed a 512-node prototype system based on what appears to be primarily pre-exascale componentry in 2018 [5].

[2] https://www.exascaleproject.org/

The system developed at NRCPC (National Research Center of Parallel Computer Engineering and Technology), is planned as the only non-accelerated architecture, and the prototype is equipped with two ShenWei 26010 (SW26010) 260-core processors, the same chip that is powering Sunway's TaihuLight supercomputer. The Sugon prototype is a heterogenous machine comprised of nodes, each outfitted with two Hygon x86 CPU and two DCUs (accelerators built by Hygon), and hooked together by a 6D torus network. The CPU is a licensed clone of AMD's first-generation EPYC processor, while the DCU is an accelerator built by Hygon. The NUDT prototype is another heterogenous architecture, in this case using CPUs of unknown parentage, plus the Matrix-2000+, a 128-core general-purpose DSP chip [5].

The Exascale supercomputers will be able to analyse smog distribution on a national level, while current models can only handle a district. Tianhe-3 also could simulate earthquakes and epidemic outbreaks in more detail, allowing swifter and more effective government responses. The new machine also will be able to analyse gene sequence and protein structures in unprecedented scale and speed. That may lead to new discoveries and more potent medicine, he said. [6].

Japan: The successor to the K supercomputer, which is being developed under the Flagship2020 program, will use ARM-based A64FX processors and these chips will be at the heart of a new system built by Fujitsu for RIKEN (Japan's Institute of Physical and Chemical Research) that would break the Exaflops barrier [7] aiming to make the system available to its users "around 2021 or 2022." [8]

European Community: The former EC President Juncker has declared that the European Union has to be competitive in the international arena with regard to the USA, China, Japan and other stakeholders, in order to enhance and promote the European industry in the public as well as the private sector related to HPC [9].

The first step will be "Extreme-Scale Demonstrators" (EsDs) that should provide pre-Exascale platforms deployed by HPC centres and used by Centres of Excellence for their production of new and relevant applications. Such demonstrators are planned by ETP4HPC Initiative and included in the EC LEIT-ICT 2018 calls. At project end, the EsDs will have a high TRL (Technical Readiness Level) that will enable stable application production at reasonable scale. [10]

The EuroHPC Initiative is based on a Memorandum of Understanding that was signed on March 23, 2017 in Rome. Its plans for the creation of two pre-Exascale machines, followed by the delivery of two machines that are actually Exascale. There are a lot of things to consider such as the creation of the micro-processor with European technology and the integration of the micro-processor in the European Exascale machines [9]. IPCEI (Important Project of Common European Interest) is another parallel initiative, related to EuroHPC. The IPCEI for HPC at the moment involves France, Italy, Spain, and Luxembourg but it is also open to other countries in the European Union. If all goes according to plan, the first pre-Exascale machine will be released by 2022 – 2023. By 2024 – 2025, the Exascale machines will be delivered [9]. Newer roadmaps see Europe on its way to Exascale with the upcoming European pre-Exascale (2020) and European Exascale (2022) supercomputers [11]. One of the exascale computers should be based on the European processor developed by the European Processor Initiative (EPI) [3]. Some details are revealed in [12].

2.3 Convergence of HPDA and HPC

2.3.1 Convergence of HPC and Cloud Computing

High-performance computing refers to technologies that enable achieving a high-level computational capacity as compared to a general-purpose computer [13]. High-performance computing in recent decades has been widely adopted for both commercial and research applications including but not limited to high-frequency trading, genomics, weather prediction, oil exploration. Since inception of high-performance computing, these applications primarily relied on simulation as a third paradigm for scientific discovery together with empirical and theoretical science.

The technological backbone for simulation has been high-performance computing platforms (also known as supercomputers) which are specialized computing instruments to run simulation at maximum speed with lesser regards to cost. Historically these platforms were designed with specialized circuitry and

[3] https://www.european-processor-initiative.eu/

architecture with maximum performance being the only goal. While in the extreme such platforms can be domain-specific [14], supercomputers have been historically programmable to enable their use for a broad spectrum of numerically-intensive computation. To benefit from the economies of scale, supercomputers have been increasingly relying on commodity components starting from microprocessors in the eighties and nineties, to entire volume servers with only specialized interconnects [15] taking the place of fully custom-designed platforms [16].

In the past decade, there have been two trends that are changing the landscape for high-performance computing and supercomputers. The first trend is the emergence of data analytics as the fourth paradigm [17] complementing simulation in scientific discovery. The latter is often related to as High-Performance Data Analytics (HPDA).

Many fields that have been in recent decades simulation-centric (e.g., computational fluid dynamics, protein folding, brain modeling) are now transitioning into hybrid discovery paradigms where a few early iterations of simulation allows for building machine-learning models that would then lead to the final outcome much faster, with higher accuracy and dramatically less computational resources.

Moreover, while simulation still remains as a major pillar for science, there are massive volumes of scientific data that are now gathered by instruments, e.g., sensors augmenting data from simulation available for analysis. The Large Hadron Collider and the Square Kilometre Array are just two examples of scientific experiments that generate in the order of Petabytes of data a day. This recent trend has led to the emergence of data science and data analytics as a significant enabler not just for science but also for humanities.

Finally, an area that has recently emerged as phenomenally computationally intensive is natural language processing. In a recent article from MIT Technology Review, researchers at the Univeristy of Massachussetes, Amherst, quantify the carbon footprint of a single large Transformer model for learning to be five times lifetime emissions of an average American car [18]. These technologies would require custom acceleration with supercomputing capabilities.

The second trend is the emergence of cloud computing and warehouse-scale computers (also known as data centres) [19]. Today, the backbone of IT and the "clouds" are data centres that are utility-scale infrastructure. Data centres consist of mainstream volume processing, networking, and storage servers aiming at cost-effective data manipulation at unprecedented scales. Data centre owners prioritize capital and operating costs (often measured in performance per watt) over ultimate performance. Typical high-end data centres draw around 20 MW, occupy an area equivalent to 17 times a football field and incur a 3 billion Euros in investment. While data centres are primarily designed for commercial use, the scale at which they host and manipulate (e.g., personal, business) data has led to fundamental breakthroughs in both data analytics and data management. By pushing computing costs to unprecedented low limits and offering data and computing services at a massive scale, the clouds will subsume much of embarrassingly parallel scientific workloads in high-performance computing, thereby pushing custom infrastructure for the latter to a niche.

2.3.2 Massive Data Analytics

We are witnessing a second revolution in IT, at the centre of which is data. The emergence of e-commerce and massive data analytics for commercial use in search engines, social networks and online shopping and advertisement has led to wide-spread use of massive data analytics (in the order of Exabytes) for consumers. Data now also lies at the core of the supply-chain for both products and services in modern economies. Collecting user input (e.g., text search) and documents online not only has led to groundbreaking advances in language translation but is also in use by investment banks mining blogs to identify financial trends. The IBM Watson experiment is a major milestone in both natural language processing and decision making to showcase a question answering system based on advanced data analytics that won a quiz show against human players.

The scientific community has long relied on generating (through simulation) or recording massive amounts of data to be analysed through high-performance computing tools on supercomputers. Examples include meteorology, genomics, connectomics (connectomes: comprehensive maps of connections within an organism's nervous system), complex physics simulations, and biological and environmental research. The proliferation of data analytics for commercial use on the internet, however, is paving

the way for technologies to collect, manage and mine data in a distributed manner at an unprecedented scale even beyond conventional supercomputing applications.

Sophisticated analytic tools beyond indexing and rudimentary statistics (e.g., relational and semantic interpretation of underlying phenomena) over this vast repository of data will not only serve as future frontiers for knowledge discovery in the commercial world but also form a pillar for scientific discovery [20]. The latter is an area where commercial and scientific applications naturally overlap, and high-performance computing for scientific discovery will highly benefit from the momentum in e-commerce.

There are a myriad of challenges facing massive data analytics including management of highly distributed data sources, and tracking of data provenance, data validation, mitigating sampling bias and heterogeneity, data format diversity and integrity, integration, security, sharing, visualization, and massively parallel and distributed algorithms for incremental and/or real-time analysis.

With respect to algorithmic requirements and diversity, there are a number of basic operations that serve as the foundation for computational tasks in massive data analytics (often referred to as *dwarfs* [21] or *giants* [20]). They include but are not limited to: basic statistics, generalized n-body problems, graph analytics, linear algebra, generalized optimization, computing integrals and data alignment. Besides classical algorithmic complexity, these basic operations all face a number of key challenges when applied to massive data related to streaming data models, approximation and sampling, high-dimensionality in data, skew in data partitioning, and sparseness in data structures. These challenges not only must be handled at the algorithmic level, but should also be put in perspective given projections for the advancement in processing, communication and storage technologies in platforms.

Many important emerging classes of massive data analytics also have real-time requirements. In the banking/financial markets, systems process large amounts of real-time stock information in order to detect time-dependent patterns, automatically triggering operations in a very specific and tight timeframe when some pre-defined patterns occur. Automated algorithmic trading programs now buy and sell millions of dollars of shares time-sliced into orders separated by 1 ms. Reducing the latency by 1 ms can be worth up to $ 100 million a year to a leading trading house. The aim is to cut microseconds off the latency in which these systems can reach to momentary variations in share prices [22].

2.3.3 Warehouse-Scale Computers

Large-scale internet services and cloud computing are now fuelled by large data centres which are a warehouse full of computers. These facilities are fundamentally different from traditional supercomputers and server farms in their design, operation and software structures and primarily target delivering a negotiated level of internet service performance at minimal cost. Their design is also holistic because large portions of their software and hardware resources must work in tandem to support these services [19].

High-performance computing platforms are also converging with warehouse scale computers primarily due to the growth rate in cloud computing and server volume in the past decade. James Hamilton, Vice President and Distinguished Engineer at Amazon and the architect of their data centres commented on the growth of Amazon Web Services (AWS) stating in 2014 that "every day AWS adds enough new server capacity to support Amazon's global infrastructure when it was a $7B annual revenue enterprise (in 2004)". The latest large-scale datacenters are now provisioned with upwards of 250 MW of electricity and are growing in size.

Silicon technology trends such as the end of Dennard Scaling [23] and the slowdown and the projected end of density scaling [24] are pushing computing towards a new era of platform design tokened ISA: (1) technologies for tighter integration of components (from algorithms to infrastructure), (2) technologies for specialization (to accelerate critical services), and (3) technologies to enable novel computation paradigms for approximation. These trends apply to all market segments for digital platforms and reinforce the emergence and convergence of volume servers in warehouse-scale computers as the building block for high-performance computing platforms.

With modern high-performance computing platforms being increasingly built using volume servers, there are a number of salient features that are shared among warehouse-scale computers and modern high-performance computing platforms including dynamic resource allocation and management, high utilization,

parallelization and acceleration, robustness and infrastructure costs. These shared concerns will serve as incentive for the convergence of the platforms.

2.3.4 High-Performance versus Warehouse-Scale Computers

There are also a number of ways that traditional high-performance computing ecosystems differ from modern warehouse-scale computers [25]. With performance being a key criterion, there are a number of challenges facing high-performance computing on warehouse-scale computers. These include but are not limited to efficient virtualization, adverse network topologies and fabrics in cloud platforms, low memory and storage bandwidth in volume servers, multitenancy in cloud environments, and open-source deep software stacks as compared to traditional supercomputer custom stacks. As such, high-performance computing customers must adapt to co-exist with cloud services given these challenges, while warehouse-scale computer operators must innovate technologies to support the workload and platform at the intersection of commercial and scientific computing.

Network fabrics is one key area where datacenters are fundamentally different from supercomputers. Traditionally large-scale IT services have required mostly generalized communication patterns across nodes and as such have relied on fat trees and CLOS topologies. Moreover, datacenter designers primarily focus on reducing the overall cost and as such datacenters have trailed behind supercomputers in adopting the latest network fabrics, switches and interface technologies. In contrast, supercomputers have been incorporated networks with a higher bisection bandwidth, with the latest in high-bandwidth fabrics and interfaces and programmable switches available in the market irrespective of cost. Because the network fabrics are slated to improve by 20% per year in the next decade and beyond with improvements in optical interconnects, a key differentiator between datacenters and supercomputers is network performance and provisioning.

2.3.5 Cloud-Embedded HPC and Edge Computing

The emergence of data analytics for sciences and warehouse scale computing will allow much of the HPC that can run on massively parallel volume servers at low cost to be embedded in the clouds, pushing infrastructure for HPC to the niche. While the cloud vendors primarily target a commercial use of large-scale IT services and may not offer readily available tools for HPC, there are a myriad of opportunities to explore technologies that enable embedding HPC into public clouds.

Large-scale scientific experiments also will heavily rely on edge computing. The amount of data sensed and sampled is far beyond any network fabric capabilities for processing in remote sites. For example, in the Large Hadron Collider (LHC) in CERN, beam collisions occur every 25 ns, which produce up to 40 million events per second. All these events are pipelined with the objective of distinguishing between interesting and non-interesting events to reduce the number of events to be processed to a few hundreds events [26]. These endeavours will need custom solutions with proximity to sensors and data to enable information extraction and hand in hand collaboration with either HPC sites or cloud-embedded HPC services.

References

[1] N. Hemsoth. *Exascale Timeline Pushed to 2023: What's Missing in Supercomputing?* 2016. URL: https://www.nextplatform.com/2016/04/27/exascale-timeline-pushed-2023-whats-missing-supercomputing.

[2] N. Hemsoth. *American HPC Vendors Get Government Boost for Exascale R&D.* 2017. URL: https://www.nextplatform.com/2017/06/15/american-hpc-vendors-get-government-boost-exascale-rd.

[3] T. Verge. *America's first exascale supercomputer to be built by 2021.* URL: https://www.theverge.com/2019/3/18/18271328/supercomputer-build-date-exascale-intel-argonne-national-laboratory-energy.

[4] S. Bauduin. *Exascale Day: Cray spricht über kommende Supercomputer.* URL: https://www.computerbase.de/2019-10/cray-exascale-day-supercomputer/.

[5] M. Feldman. *China Fleshes Out Exascale Design for Tianhe-3 Supercomputer.* URL: https://www.nextplatform.com/2019/05/02/china-fleshes-out-exascale-design-for-tianhe-3/.

[6] Z. Zhihao. *China to jump supercomputer barrier.* 2017. URL: http://www.chinadaily.com.cn/china/2017-02/20/content_28259294.htm.

[7] T. P. Morgan. *Inside Japan's Future Exascale ARM Supercomputer.* 2016. URL: https://www.nextplatform.com/2016/06/23/inside-japans-future-exaflops-arm-supercomputer.

[8] M. Feldman. *Japan Strikes First in Exascale Supercomputing Battle.* URL: https://www.nextplatform.com/2019/04/16/japan-strikes-first-in-exascale-supercomputing-battle/.

[9] A. Emmen. *Exciting times for the European citizen: EuroHPC plans two exascale machines by 2024-2025*. 2017. URL: http://primeurmagazine.com/weekly/AE-PR-07-17-45.html.

[10] ETP4HPC European Technology Platform for High-Performance Computing. *Strategic Research Agenda 2015 Update, Section 8*. 2016. URL: http://www.etp4hpc.eu/en/news/18-strategic-research-agenda-update.html.

[11] J. Reinders. *Outlook onto European Supercomputing is amazing*. URL: https://eetimes.eu/european-processor-initiative-announces-common-platform-for-hpc/.

[12] N. Dahad. *European Processor Initiative Announces Common Platform For HPC*. URL: https://medium.com/@jamesreinders/outlook-onto-european-supercomputing-is-amazing-2589710fa2de.

[13] Wikipedia. *Supercomputer*. URL: http://en.wikipedia.org/wiki/Supercomputer.

[14] Wikipedia. *Anton (computer)*. URL: http://en.wikipedia.org/wiki/Anton_(computer).

[15] Cray Inc. *Cray: XC Series*. URL: http://www.cray.com/products/computing/xc-series.

[16] The Next Platform. *Supercomputing Strategy Shifts in a World without BlueGene*. Apr. 14, 2014. URL: https://www.nextplatform.com/2015/04/14/supercomputing-strategy-shifts-in-a-world-without-bluegene.

[17] *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Redmond, Washington, 2009. URL: https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery.

[18] K. Hao. *Training a single AI model can emit as much carbon as five cars in their lifetimes*. 2019. URL: https://www.technologyreview.com/s/613630/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/.

[19] L. A. Barroso, J. Clidaras, and U. Hölzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*. 2013. DOI: 10.2200/S00516ED2V01Y201306CAC024.

[20] National Research Council. *Frontiers in Massive Data Analysis*. Washington, DC, 2013. DOI: 10.17226/18374. URL: https://www.nap.edu/catalog/18374/frontiers-in-massive-data-analysis.

[21] K. Asanovic et al. *The Landscape of Parallel Computing Research: A View from Berkeley*. Tech. rep. UCB/EECS-2006-183. EECS Department, University of California, Berkeley, Dec. 2006. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html.

[22] R. Tieman. *Algo trading: the dog that bit its master*. 2008. URL: https://www.ft.com/content/cd68eae2-f1e0-11dc-9b45-0000779fd2ac.

[23] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki. "Toward Dark Silicon in Servers". In: *IEEE Micro* 31.4 (July 2011), pp. 6–15. DOI: 10.1109/MM.2011.77.

[24] The Economist. *After Moore's Law*. Mar. 2016. URL: http://www.economist.com/technology-quarterly/2016-03-12/after-moores-law.

[25] J. Simmons. "HPC Cloud Bad; HPC in the Cloud Good". In: 2015.

[26] M. Shapiro. *Supersymmetry, Extra Dimensions and the Origin of Mass: Exploring the Nature of the Universe Using PetaScale Data Analysis*. URL: https://www.youtube.com/watch?v=-cdbnwaW34g.

3 Technology

3.1 Digital Silicon-based Technology

3.1.1 Continuous CMOS scaling

Continuing Moore's Law and managing power and performance tradeoffs remain as the key drivers of the International Technology Roadmap For Semiconductors 2015 Edition (ITRS 2015) grand challenges. Silicon scales according to the Semiconductor Industry Association's ITRS 2.0, Executive Summary 2015 Edition [1] to 11/10 nm in 2017, 8/7 nm in 2019, 6/5 nm in 2021, 4/3 nm in 2024, 3/2.5 nm in 2027, and 2/1.5 nm in 2030 for MPUs or ASICs.

DRAM half pitch (i.e., half the distance between identical features in an array) is projected to scale down to 10 nm in 2025 and 7.7 nm in 2028 allowing up to 32 GBits per chip. However, DRAM scaling below 20 nm is very challenging. DRAM products are approaching fundamental limitations as scaling DRAM capacitors is becoming very difficult in 2D structures. It is expected that these limits will be reached by 2024 and after this year DRAM technology will saturate at the 32 Gbit level unless some major breakthrough will occur [2]. The same report foresees that by 2020 the 2D Flash topological method will reach a practical limit with respect to cost effective realization of pitch dimensions. 3D stacking is already extensively used to scale flash memories by 3D flash memories.

Process downscaling results in increasing costs below 10 nm: the cost per wafer increases from one technology node to the next [3]. The ITRS roadmap does not guarantee that silicon-based CMOS will extend that far because transistors with a gate length of 6nm or smaller are significantly affected by quantum tunneling.

CMOS scaling depends on fabs that are able to manufacture chips at the highest technology level. Only three such fabs are remaining worldwide: TSMC, Intel and Samsung.

Current State

Current (December 2019) high-performance multiprocessors and smartphone microprocessors feature 14- to 7 nm technology. 14-nm FinFET technology is available by Intel (Intel Kaby Lake) and GlobalFoundries. 10-nm manufacturing process was expected for 2nd half of 2017 or beginning of 2018 by Intel but was delayed until 2019. Samsung and TSMC applied 10 nm technology already in 2017. In 2019 only three semiconductor foundries – Samsung, Intel and TSMC – apply an 10 nm process.

First introduced between 2017-2019, the 10 nm process technology is characterized by its use of FinFET transistors with a 30-40 nm fin pitches. Those nodes typically have a gate pitch in range of 50-60 nm and a minimum metal pitch in the range of 30-40 nm [4]. The terms "7 nm" or "10 nm" are simply a commercial name for a generation of a certain size and its technology and does not represent any geometry of a transistor [4].

Samsung's first-generation 10 nm FinFET fabrication process (10LPE) allowed the company to make its chips 30% smaller compared to ICs made using its 14LPE process as well as reducing power consumption by 40% (at the same frequency and complexity) or increasing their frequency by 27% (at the same power and complexity). Samsung applied that process to the company's own Exynos 9 Octa 8895 as well as Qualcomm's Snapdragon 835 seen in the Samsung Galaxy S8 [5].

Company Roadmaps

R&D has begun already in 2017 for 5 nm by all four remaining fabs TSMC, GlobalFoundries, Intel and Samsung and also beyond towards 3 nm. Both 5 nm and 3 nm present a multitude of unknowns and challenges. Regardless, based on the roadmaps from various chipmakers, Moore's Law continues to slow as process complexities and costs escalate at each new chip generation.

Intel plans 7 nm FinFET for production in early to mid-2020, according to industry sources. Intel's 5 nm production is targeted for early 2023, sources said, meaning its traditional 2-year process cadence is extending to roughly 2.5 to 3 years [6]. WikiChip gives the Note: For the most part, foundries' 7nm process is competing against Intel's 10nm process, not their 7nm [7].

TSMC plans to ship 5 nm in 2020, which is also expected to be a FinFET. In reality, though, TSMC's 5 nm will likely be equivalent in terms of specs to Intel's 7 nm, analysts said [6].

TSMC started production of their 7 nm HK-MG FinFET process in 2017 and is already actively in development of 5 nm process technology as well. Furthermore, TSMC is also in development of 3nm process technology. Although 3 nm process technology already seems so far away, TSMC is further looking to collaborate with academics to begin developing 2 nm process technology [8].

Samsung's newest foundry process technologies and solutions introduced at the annual Samsung Foundry Forum include 8 nm, 7 nm, 6 nm, 5 nm, 4 nm in its newest process technology roadmap [9]. However, no time scale is provided.

Samsung will use EUVL for their 7nm node and thus will be the first to introduce this new technology after more than a decade of development. On May 24 2017, Samsung released a press release of their updated roadmap. Due to delays in the introduction of EUVL, Samsung will introduce a new process called 8 nm LPP, to bridge the gap between 10 nm and 7 nm [7].

GlobalFoundries: As of August 2018 GlobalFoundries has announced they will suspend further development of their 7nm, 5nm and 3nm process.

Expected to be introduced by the foundries TSMC and Samsung in 2020 (and by Intel in 2013), the 5-nanometer process technology is characterized by its use of FinFET transistors with fin pitches in the 20s of nanometer and densest metal pitches in the 30s of nanometers. Due to the small feature sizes, these processes make extensive use of EUV for the critical dimensions [10].

Not much is known about the 3 nm technology. Commercial integrated circuit manufacturing using 3 nm process is set to begin some time around 2023 [11].

Research Perspective

"It is difficult to shed a tear for Moore's Law when there are so many interesting architectural distractions on the systems horizon" [12]. However, silicon technology scaling will still continue and research in silicon-based hardware is still prevailing, in particular targeting specialized and heterogeneous processor structures and hardware accelerators.

However, each successive process shrink becomes more expensive and therefore each wafer will be more expensive to deliver. One trend to improve the density on chips will be 3D integration also of logic. Hardware structures that mix silicon-based logic with new NVM technology are upcoming and intensely investigated. A revolutionary DRAM/SRAM replacement will be needed [1].

As a result, non-silicon extensions of CMOS, using III–V materials or Carbon nanotube/nanowires, as well as non-CMOS platforms, including molecular electronics, spin-based computing, and single-electron devices, have been proposed [1].

For a higher integration density, new materials and processes will be necessary. Since there is a lack of knowledge of the fabrication process of such new materials, the reliability might be lower, which may result in the need of integrated fault-tolerance mechanisms [1].

Research in CMOS process downscaling and building fabs is driven by industry, not by academic research. Availability of such CMOS chips will be a matter of costs and not only of availability of technology.

References

[1] *Semiconductor Industry Association. ITRS 2.0, Executive Summary 2015 Edition.* URL: http://cpmt.ieee.org/images/files/Roadmap/ITRSHetComp2015.pdf.

[2] *Semiconductor Industry Association. ITRS 2.0, 2015 Edition, Executive Report on DRAM.* URL: https://www.semiconductors.org/clientuploads/Research_Technology/ITRS/2015/0_2015%20ITRS%202.0%20Executive%20Report%20(1).pdf.

[3] *HiPEAC Vision 2015.* URL: www.hipeac.net/v15.

[4] WikiChip. *7 nm lithography process.* 2019. URL: https://en.wikichip.org/wiki/10_nm_lithography_process.

[5] A. Shilov. *Samsung and TSMC Roadmaps: 8 and 6 nm Added, Looking at 22ULP and 12FFC.* URL: http://www.anandtech.com/show/11337/samsung-and-tsmc-roadmaps-12-nm-8-nm-and-6-nm-added.

[6] M. Lapedus. *Uncertainty Grows For 5 nm, 3 nm*. URL: https://semiengineering.com/uncertainty-grows-for-5nm-3nm/.

[7] WikiChip. *7 nm lithography process*. 2019. URL: https://en.wikichip.org/wiki/7_nm_lithography_process.

[8] S. Chen. *TSMC Already Working on 5 nm, 3 nm, and Planning 2 nm Process Nodes*. URL: https://www.custompcreview.com/news/tsmc-already-working-5nm-3nm-planning-2nm-process-nodes/33426/.

[9] Samsung. *Samsung Set to Lead the Future of Foundry with Comprehensive Process Roadmap Down to 4 nm*. URL: https://news.samsung.com/global/samsung-set-to-lead-the-future-of-foundry-with-comprehensive-process-roadmap-down-to-4nm.

[10] WikiChip. *5 nm lithography process*. 2019. URL: https://en.wikichip.org/wiki/5_nm_lithography_process.

[11] WikiChip. *3 nm lithography process*. 2019. URL: https://en.wikichip.org/wiki/3_nm_lithography_process.

[12] N. Hemsoth. *Neuromorphic, Quantum, Supercomputing Mesh for Deep Learning*. URL: https://www.nextplatform.com/2017/03/29/neuromorphic-quantum-supercomputing-mesh-deep-learning/.

3.1.2 Die Stacking and 3D-Chips

Die Stacking and three-dimensional chip integration denote the concept of stacking integrated circuits (e.g. processors and memories) vertically in multiple layers. 3D packaging assembles vertically stacked dies in a package, e.g., system-in-package (SIP) and package-on-package (POP).

Die stacking can be achieved by connecting separately manufactured wafers or dies vertically either via wafer-to-wafer, die-to-wafer, or even die-to-die. The mechanical and electrical contacts are realized either by wire bonding as in SIP and POP devices or microbumps. SIP is sometimes listed as a 3D stacking technology, although it is more precisely denoted as 2.5D technology.

An evolution of SIP approach which is now extremely strategic for HPC systems consists of stacking multiple dies (called chiplets) on a large interposer that provides connectivity among chiplets and to the package. The interposer can be passive or active. A passive interposer, implemented with silicon or with an organic material to reduce cost, provides multiple levels of metal interconnects and vertical vias for inter-chiplet connectivity and for redistribution of connections to the package. It also provides micropads for the connection of the chiplets on top. Active silicon interposers offer the additional possibility to include logic and circuits in the interposer itself. This more advanced and high cost integration approach is much more flexible than passive interposers, but it is also much more challenging for design, manufacturing, test and thermal management. Hence it is not yet widespread in commercial products.

The advantages of 2.5D technology based on chiplets and interposers are numerous: First, short communication distance between dies and finer pitch for wires in the interposer (compared to traditional PCBs), thus reducing communication load and then reducing communication power consumption. Second, the possibility of assembling on the same interposer dies from various heterogeneous technologies, like DRAM and non-volatile memories, or even photonic devices, in order to benefit of the best technology where it fits best. Third, an improved system yield and cost by partitioning the system in a divide-and-conquer approach: multiple dies are fabricated, tested and sorted before the final 3D assembly, instead of fabricating ultra-large dies with much reduced yield. The main challenges for 2.5D technology are manufacturing cost (setup and yield optimization) and thermal management since cooling high-performance requires complex packages, thermal coupling materials and heat spreaders, and chiplets may have different thermal densities (e.g. logic dies have typically much higher heat per unit area dissipation than memories). Passive silicon interposers connecting chiplets in heterogeneous technologies are now mainstream in HPC product: AMD EPYC processors, integrating 7nm and 14nm logic chiplets, NVIDIA TESLA GPGPUs integrating logic and DRAM chiplets (high-bandwidth memory - HBM - interface)and also Intel chips are most notable examples.

True 3D integration, where silicon dies are vertically stacked on top of each other is most advanced and commercially available in memory chips. The leading technology for 3D integration is based on Through-Silicon Vias (TSVs) and it is widely deployed in DRAMs. In fact, the DRAMs used in high-bandwidth memory (HBM) chiplets are made with multiple stacked DRAM dies connected by TSVs. Hence, TSV-based 3D technology and interposer-based 2.5D technology are indeed combined when assembling an HBM multi-chiplet system.

However, TSVs are not the densest 3D connectivity option. 3D-integrated circuits can also be achieved by stacking active layers vertically on a single wafer in a monolithic (sequential) approach. This kind of 3D

chip integration does not use micro-pads or Through-Silicon Vias (TSVs) for communication, but it uses vertical interconnects between layers, with a much finer pitch than that allowed by TSVs. The main challenge in monolithic integration is to ensure that elementary devices (transistors) have similar quality level and performance in all the silicon layers. This is a very challenging goal since the manufacturing process is not identical for all the layers (low temperature processes are needed for the layers grown on top of the bulk layer). However, monolithic 3D systems are currently in volume production, even though for very specialized structures, namely, 3D NAND flash memories. These memories have allowed flash technology to scale in density beyond the limits of 2D integration and they are now following a very aggressive roadmap towards hundreds of layers.

While TSV-based and monolithic 3D technologies are already mature and in production for memories, they are still in prototyping stage for logic, due to a number of technical challenges linked to the requirement of faster transistors, the extremely irregular connections and the much higher heat density that characterize logic processes.

Some advanced solutions for vertical die-to-die communication do not require ohmic contact in metal, i.e. capacitive and inductive coupling as well as short-range RF communication solutions that do not require a flow of electrons passing through a continuous metal connection. These approaches are usable both in die-stacked and monolithic flavors, but the transceivers and modulator/demodulator circuits do take space and vertical connectivity density is currently not better than that of TSVs, but could scale better in multi-layer stacks. These three-dimensional die-to-die connectivity options are not currently available in commercial devices, but their usability and cost is under active exploration.

Current State

2.5D technology is now extremely solid and is growing rapidly in many domains. HPC is one of the highest-penetration areas: GPUs from NVIDIA and AMD have a roadmap based on 2.5D High-Bandwidth Memory (HBM)interfaces. AMD's GPUs based on the Fiji architecture with HBM are available since 2015, and NVIDIA released the first HBM GPUs "Pascal" in 2016 [1]. Today's HBM products, based on the HBM2 spec, enable 4/8GB capacities. HBM2 features 1,024 I/Os, and pin speed is 2.4Gbps, achieving 307GB/s bandwidth. The latest HBM version is based on the HBM2E spec, which has 8/16GB capacities. It has 1,024 I/Os with 3.2Gbps transfer rates, achieving 410GB/s of bandwidth. HBM2E is sampling and it is expected to reach the market in 2020. The next version, HBM3, has 4Gbps transfer rates with 512GB/s bandwidth, and it is planned for 2020/21. HBM is also very common in specialized machine learning accelerators, such as Habana Lasb's (recently bought by Intel) Gaudi AI Training Processor. The Gaudi processor includes 32GB of HBM-2 memory.

HBM stacks DRAM dies on top of each other and connects them with TSVs. For example, Samsung's HBM2 technology consists of eight 8Gbit DRAM dies, which are stacked and connected using 5,000 TSVs. The bandwidth advantage for HBM with respect to standard DDR memories is staggering: HBM2 enables 307GB/s of data bandwidth, compared to 85.2GB/s with four DDR4 DIMMs. Recently, Samsung introduced a new HBM version that stacks 12 DRAM dies, which are connected using 60,000 TSVs. The package thickness is similar to the 8-die stack version. This HBM flavor is for data-intensive applications, like AI and HPC. It achieves 24 gigabytes of density. That's a 3x improvement over the prior generation.

HBM is probably the most advanced and well-defined 2.5D interface standard used today in HPC across multiple vendors. However, 2.5D chiplet integration technology is also heavily used by AMD in their Zen 2 EPYC server processors (codenamed Rome), to integrate up to 64 cores within a 5-chiplet package with silicon interposer. 2.5D approaches are also heavily used in high-end FPGAs from both Intel (Altera) and Xilinx, to integrate in the same package multiple FPGA dies, as well as HBM memories. Both CPU and FPGA use proprietary protocols and interfaces for their inter-chiplet connectivity, as opposed to the memory-chiplet connectivity in HBM which is standardized by JEDEC.

It is important to note that 2.D in general and HBM in particular are expensive technologies: in early 2020 the unit prices for HBM2 (16GB with 4 stack DRAM dies) is roughly $120, according to TechInsights. That does not include the cost of the package. Hence, for large DRAM capacities, the slower a lower-bandwidth DDR (4 and 5) remain the only viable option and they won't disappear in the HPC space.

As for monolithic 3D, this approach of die stacking is already used in commercial 3D Flash memories from

vendors like Samsung, Hynix, Intel and Western digital. They are used mostly for SSD storage in HPC but they are also very heavily used in mobile phones, as they allow very small form factors. Current products have up to 128 3D NAND flash cell layers although volume shipments are for 96 layers or less. By 2020 128-layer 3D NAND products will be in volume production with 192-layer 3D NAND probably sampling. By 2022 3D NAND flash with over 200 layers will probably be available. However, the manufacturing cost grows with the number of flash layers, so number of layers does not translate linearly into a storage capacity cost reduction. For this reason, NAND flash companies are also pushing multi-bit per cell (three and four bit per cell) for enterprise and client applications.

Flash memories are not the only non-volatile memories to follow a 3D-integration roadmap. Intel and Micron announced "3D XPoint" memory already in 2015 (assumed to be 10x the capacity of DRAM and 1000x faster than NAND Flash [2]). Intel/Micro 3D-Xpoint memory has been commercially available as Optane-SSDs DC P4800X-SSD as 375-Gbyte since March 2017 and stated to be 2.5 to 77 times "better" than NAND-SSDs. Even though the Optane product line has encountered several roadmap issues (technical and business-related), it is now actively marketed by Intel. Optane products are also developed as persistent-memory modules, which can only be used with Intel's Cascade Lake Xeon CPUs, available in 128 GB, 256 GB or 512 GB capacities. A second-generation Optane persistent memory module, code-named Barlow Pass and a second-generation Optane SSD, code-named Alder Stream, are planned for release by the end of 2020.

For what concerns 3D logic technology, prototypes date back to 2004 when Tezzaron released a 3D IC microcontroller [3]. Intel evaluated chip stacking for a Pentium 4 already in 2006 [4]. Early multicore designs using Tezzaron's technology include the 64 core 3D-MAPS (3D Massively Parallel processor with Stacked memory) research prototype from 2012 [5] and the Centip3De with 64 ARM Cortex-M3 Cores also from 2012 [6]. Fabs are able to handle 3D packages (e.g. [7]). In 2011 IBM announced 3D chip production process [8]. 3D-Networks-on-Chips for connecting stacks of logic dies have been demonstrated in 2011[9]. All these early prototypes, based on TSV approaches, have not reached product maturity. More recent research on 3D logic is focusing on monolithic integration, where multiple layers of active devices are fabricated together using multiple lithographic steps on the same silicon die, and on reducing the pitch of TSVs by new wafer-scale assembly processes [10].

Also the field of contactless connectivity for 3D integration is in an exploratory phase, with a number of different options being considered namely, capacitive, inductive coupling and short-range RF. These alternative approaches do not require ohmic connections between dies, and hence are potentially more flexible in terms of interconnection topologies implementable in the 3D stack. However, their maturity level is lower than that of TSVs and their cost and density need to be optimized for production [11].

Perspective

2.5D is now the to-go technology for HPC. All major silicon vendors in the HPC space (Intel, AMD, NVIDIA) have solid roadmaps based on 2.5D approaches, namely HBM and chiplet integration. In general, we will see for sure increased use of 2.5D chiplet technology, not only to tackle the memory bandwidth bottleneck (the primary goal of HBM), but also to improve yield, by integrating multiple smaller chips on a large interposer. In an alternative view, chiplets also can be used to increase the die size to 1000mm^2, which is larger than reticle size, by using a common interconnect and putting many homogeneous chiplets on a substrate to build a huge 2.5D-integrated multi-chiplet "mega-chip". It is important to note that 2.D technology is very flexible: it can be used to connect chips developed at different process nodes, depending on what makes the most sense for a particular function. This is done today in modern CPUs from AMD, which use different logic technologies in their chiplet-based processors (7nm for processor chiplets and 14nm for a IO-central-hub chiplet).

We expect to see many more variations of 2.5D integration in the next 5-10 years, with the densest logic technology (5nm will potentially be introduced in 2020/21) used for compute-intensive chiplets and differentiated, possibly less scaled technology for different functions, such as storage, IO, accelerators. It is quite easy to envision that in the 2020 decade 2.5D technology will be essential for maintaining the pace of performance and energy efficiency evolutions and compensate for the slowdown of Moore's law. A key enabler for 2.5D technology development is the definition of standard protocols for chiplet-to-chiplet communication. Currently there are several proprietary protocols, but an evolution toward a multi-vendor

standard is highly desirable: a candidate is Bunch of Wires (BoW), the new chiplet interface proposed by the OCP ODSA group, designed to address the interface void for organic substrates [12]. Significant innovation will come from interposer technology: organic substrates are aggressively developed to compete with silicon in terms of connection density and bandwidth density for passive interposers. Silicon interposers are moving toward active solutions, such as the FOVEOS approach developed by Intel, with products announced in late 2020.

3D stacked memories using TSVs (HBM DRAMs) and monolithic integration (3D NAND Flash) are now mainstream in HPC and they are here to stay, with solid and aggressive roadmaps. Various alternative non-volatile memory technologies are also heavily relying on 3D integration. The Xpoint technology, based on 3D (multi-layer) phase change memory (PCM) has already reached volume production and is available as a niche option in HPC. Other technologies are actively explored, such as magnetic random access memory (MRAM), ferroelectric RAM (FRAM), resistive RAM (RRAM). These memories will have a hard time to compete in cost as solid-state storage options against 3D Flash and the still cost-competitive traditional hard-disk drives. On the other hand, an area of growth for these new non-volatile memories is in the memory hierarchy as complement of replacement for DRAM main memory. Octane DRAM-replacing DIMMs in particular are intended for use with Intel's advanced server processors and Intel is using this technology to differentiate from competitors for the next generation of server CPUs. Other HPC manufacturers, such as Cray/HPE are using Optane memory in their storage systems as an intermediate storage element in an effort to reduce DRAM, as a write cache to achieve higher endurance NAND flash and other applications. This is because Optane memory sells for a per capacity price between NAND flash and DRAM.

We expect non-Flash 3D NV memory technologies to start competing in the HPC space in the next five years, as several semiconductor foundries are offering MRAM (as well as RRAM) as options for embedded memory to replace NOR, higher level (slower) SRAM and some DRAM. Some embedded products using MRAM for inference engine weight memory applications have appeared in 2019. Probably the most short-term entry for MRAM and RRAM technology is as embedded memories on logic SoCs, but coupling these memories in multi-layer monolithic 3D configurations, possibly as chiplets in 2.5D integrated systems, as done today for HBM DRAM, opens exciting innovation and differentiation perspectives for HPC.

All 3D solid-state memory applications will benefit from developments of interface technologies that allow utilizing their inherent higher performance with respect to HDD and traditional flashes, especially for write operation. In particular, the NVMe protocol, based upon the PCIe bus, and the use of this protocol supported over various storage fabric technologies (NVMe over fabric, or NVMe-oF), combined with software and firmware, are becoming key enablers in the development of the modern storage and memory hierarchy.

For what concerns the roadmap of three-dimensional integration for logic processes (including SRAM memories), future perspectives are blurrier. To the best of our knowledge, there are no volume commercial products using logic die stacking for high-performance computing (or computing in general) and no product announcements have been made by major players. This is mainly due to the lack of a compelling value proposition. Current production-ready TSV-based 3D integration technology does not offer enough vertical connectivity density and bandwidth density to achieve a performance boost that would justify the cost and risk to achieve production-quality 3D-stacks of logic dies. Similarly, monolithic 3D technologies have not yet been able to demonstrate sufficiently high added value, due to the performance deterioration of transistors implemented within higher layers of the chip stack.

This situation is probably going to change in the next five years, as scaled transistors (sub-5nm) are moving toward true three-dimensional structures, such as the "gate all around" devices demonstrated at large scale of integration [13]. These devices offer disruptive options for integration in the vertical dimension, creating new avenues to implement even truly monolithic three-dimensional elementary gates. Additional options are offered by "buried layer" metallization: for instance, new high-density SRAM cells can be envisioned in advanced nodes exploiting buried Vdd distribution [14].

A key challenge in establishing full-3D logic chip stacking technology is gaining control of the thermal problems that have to be overcome to realize reliably very dense 3D stacks working at high frequency. This requires the availability of appropriate design tools,

which are explicitly supporting 3D layouts. Both topics represent an important avenue for research in the next 10 years.

Impact on Hardware

Full 3D-stacking has multiple potential beneficial impacts on hardware in general and on the design of future processor-memory-architectures in particular. Wafers can be partitioned into smaller dies because comparatively long horizontally running links are relocated to the third dimension and thus enable smaller form factors, as done today for 3D memories. 3D stacking also enables heterogeneity, by integrating layers, manufactured in different processes, e.g., different memory technologies, like SRAM, DRAM, Spin-transfer-torque RAM (STT-RAM) and also memristor technologies. Due to short connection wires, reduction of power consumption is to be expected. Simultaneously, a high communication bandwidth between layers can be expected leading to particularly high processor-to-memory bandwidth, if memories can be monolithically integrated with logic gates.

The last-level caches will probably be the first to be affected by 3D stacking technologies when they will enter logic processes. 3D caches will increase bandwidth and reduce latencies by a large cache memory stacked on top of logic circuitry. In a further step it is consequent to expand 3D chip integration also to main memory in order to make a strong contribution in reducing decisively the current memory wall which is one of the strongest obstructions in getting more performance in HPC systems. Furthermore, possibly between 2025 and 2030, local memories and some arithmetic units will undergo the same changes ending up in complete 3D many-core microprocessors, which are optimized in power consumption due to reduced wire lengths and denser 3D cells. In-memory-3D processing, where computation and storage are integrated on the vertical dimension at a very fine pitch is also another long-term promising direction, with early adoption in product for specialized computation (e.g. Neural networks [15]): several startups are active in this area and have announced products (e.g. Crossbar Inc.) and some large companies (e.g. IBM) have substantial R&D effort in this field.

It is highly probable that 2.5D (chiplets) and full 3D (monolithic integration) will continue to coexist and partially merge in the 2020 decade. Most ICs will consist of multiple chiplets integrated on interposers (possibly active ones), and chiplets themselves will be true 3D integrated stacks based on high density (micrometer pitch) TSV connections as well as truly monolithic ultra-high density (Nano-meter pitch) vertical devices and wires. Niche application may be covered by non-ohmic 3D connections.

A collateral but very interesting trend is 3D stacking of sensors. Sony is market leader in imaging sensors and it uses extensively 3D stacking technology to combine image sensors directly with column-parallel analogue-digital-converters and logic circuits [16, 17]. This trend will open the opportunity for fabricating fully integrated systems that will also include sensors and their analogue-to-digital interfaces. While this integration trend won't impact directly the traditional market for HPC chips, it will probably gain traction in many high growth areas for embedded HPC.

Funding Perspectives

It is now clear that more and more hardware devices will use 3D technology and virtually all HPC machines in the future will use chips featuring some form of three-dimensional integration. Hence, circuit-level and system-level design will need to increasingly become 3D-aware. Moreover, some flavors of three-dimensional IC technology are now being commoditized, with foundries offering 2.5D integration options even for startups and R&D projects. As a consequence, three-dimensional technology won't be accessible as internal technology only to multi-billion-dollar industry players. Given the already demonstrated impact and rapidly improving accessibility and cost, definitely the EU needs to invest in research on how to develop components and systems based on 3D technology. Interconnect and interoperability standards are required. It is also clear that technology development of 3D technology is getting increasingly strategic and hence significant R&D investments are needed also in this capital-intensive area for Europe to remain competitive in HPC.

References

[1] NVIDIA. *NVIDIA Tesla P100 Whitepaper*. 2016. URL: https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

[2] Intel. *Intel® Optane™: Supersonic memory revolution to take-off in 2016*. 2016. URL: http://www.intel.eu/content/www/eu/en/it-managers/non-volatile-memory-idf.html.

[3] Tezzaron Semiconductor. *Tezzaron 3D-IC Microcontroller Prototype*. 2016. URL: http://www.tachyonsemi.com/OtherICs/3D-IC%5C_8051%5C_prototype.htm.

[4] B. Black et al. "Die Stacking (3D) Microarchitecture". In: *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO 39. Washington, DC, USA, 2006, pp. 469–479. DOI: 10.1109/MICRO.2006.18. URL: https://doi.org/10.1109/MICRO.2006.18.

[5] D. H. Kim et al. "3D-MAPS: 3D Massively parallel processor with stacked memory". In: *2012 IEEE International Solid-State Circuits Conference*. Feb. 2012, pp. 188–190. DOI: 10.1109/ISSCC.2012.6176969.

[6] D. Fick et al. "Centip3De: A 3930DMIPS/W configurable near-threshold 3D stacked system with 64 ARM Cortex-M3 cores". In: *2012 IEEE International Solid-State Circuits Conference*. Feb. 2012, pp. 190–192. DOI: 10.1109/ISSCC.2012.6176970.

[7] *3D & Stacked-Die Packaging Technology Solutions*. 2016. URL: http://www.amkor.com/go/3D-Stacked-Die-Packaging.

[8] IBM. *IBM setzt erstmals 3D-Chip-Fertigungsverfahren ein*. 2016. URL: http://www-03.ibm.com/press/de/de/pressrelease/36129.wss.

[9] G. V. der Plas et al. "Design Issues and Considerations for Low-Cost 3-D TSV IC Technology". In: *IEEE Journal of Solid-State Circuits* 46.1 (Jan. 2011), pp. 293–307.

[10] S. V. Huylenbroeck et al. "A Highly Reliable 1.4?m Pitch Via-Last TSV Module for Wafer-to-Wafer Hybrid Bonded 3D-SOC Systems". In: *IEEE 69th Electronic Components and Technology Conference (ECTC)*. 2019, pp. 1035–1040.

[11] V. F. P. I. A. Papistas and D. Velenis. "Fabrication Cost Analysis for Contactless 3-D ICs". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 66.5 (May 2019), pp. 758–762.

[12] URL: https://github.com/opencomputeproject/ODSA-BoW.

[13] A. Veloso et al. "Vertical Nanowire and Nanosheet FETs: Device Features, Novel Schemes for improved Process Control and Enhanced Mobility, Potential for Faster & More Energy-Efficient Circuits". In: *IEDM* (2019).

[14] S. M. Salahuddin et al. "SRAM With Buried Power Distribution to Improve Write Margin and Performance in Advanced Technology Nodes". In: *IEEE Electron Device Letters* 40.8 (Aug. 2019), pp. 1261–1264.

[15] M. Gao et al. "TETRIS: Scalable and Efficient Neural Network Acceleration with 3D Memory". In: *International Conference on Architectural Support for Programming Languages and Operating Systems*. Apr. 2017, pp. 751–764.

[16] T. Kondo et al. "A 3D stacked CMOS image sensor with 16Mpixel global-shutter mode and 2Mpixel 10000fps mode using 4 million interconnections". In: *2015 Symposium on VLSI Circuits (VLSI Circuits)*. June 2015, pp. C90–C91. DOI: 10.1109/VLSIC.2015.7231335.

[17] R. Muazin. *Sony Stacked Sensor Presentation (ISSCC 2013)*. 2013. URL: http://image-sensors-world-blog.blogspot.de/2013/02/isscc-2013-sony-stacked-sensor.html.

3.2 Memristor-based Technology

3.2.1 Memristor Technologies

Memristors, i.e. *resistive memories*, are an emerging class of different Non-volatile Memory (NVM) technologies. The memristor's electrical resistance is not constant but depends on the previously applied voltage and the resulting current. The device remembers its history—the so-called *Non-Volatility Property*: when the electric power supply is turned off, the memristor remembers its most recent resistance until it is turned on again [1].

Currently NAND Flash is the most common NVM technology, which finds its usages on SSDs, memory cards, and memory sticks. Flash-based SCM is currently also applied in supercomputers as so-called *Storage-Class Memory (SCM)* (see also Sect. 4.1.3), i.e., an intermediate storage layer between DRAM memory and cheap disc-based bulk storage to bridge the access times of DRAM versus disks. NAND and also NOR flash use floating-gate transistors for storing single bits. This technology is facing a big challenge, because scaling down decreases the endurance and performance significantly [2]. Hence the importance of memristors as alternative NVM technology increases. New NVM technologies will strongly influence the memory hierarchy of computer systems. Memristors will deliver non-volatile memory which can be used potentially in addition to DRAM, or possibly as a complete replacement. The latter will lead to a new Storage Memory Class (SCM) in high-performance computers that is much faster than Flash.

Memristor technology blurs the distinction between memory and storage by enabling new data access modes and protocols that serve both "memory" and "storage". Moreover, memristor technology may lead to Memristive Computing by integrating memory and compute capabilities such that in-memory computing is enabled (see Sect. 4.2). Furthermore, new neuromorphic processing is possible that utilizes analog properties of memristors (see Sect. 3.2.4). Using emerging NVM technologies in computing systems is a further step towards energy-aware measures for future computer architectures.

Memristor Defined by Leon Chua's System Theory

L. Chua [3] assumed already in 1971 that a fourth fundamental two-terminal passive circuit element exists besides the resistor, the capacitor, and the inductor. He called this element a memristor. A memristor should be able to change its resistive features non-volatile in dependence on an outer appearing electrical flux that controls the relation of the devices' inner charge. Since then such memristive features were discovered in nanoscaled devices by a research group around S. Williams at HP labs in 2008 [4].

A *memristor* is defined by Leon Chua's system theory as a memory device with a hysteresis loop that is pinched, i.e. its I–U (current–voltage) curve goes to the zero point of the coordinate system. Considered from a system theoretical view according to Chua a dynamical system is characterized by an internal state variable, x, an external excitation of the system, u, and its output y, which is characterized by a non-linear function h (see euations 3.1). The change of its internal state, $\dot{x}$, over time, t, is determined by the time-dependent non-linear function f. In general y and u can be multi-dimensional functions.

$$\begin{aligned} \vec{y} &= h(x, \vec{u}, t) \\ \dot{x} &= f(x, \vec{u}, t) \end{aligned} \tag{3.1}$$

For a memristive system it holds the special case of a dynamic system in which y and u are scalar values. According to (3.2) y is 0 when $u = 0$, which corresponds to a Lissajous figure with pinched hysteresis loop (see Fig. 3.1).

$$\begin{aligned} y &= h(x, t, u) \times u \\ \dot{x} &= f(x, u, t) \end{aligned} \tag{3.2}$$

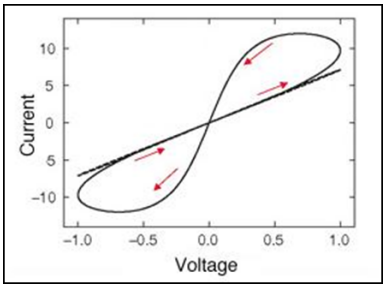


Figure 3.1: Pinched hysteresis I-U curve.

A memristor itself is a special case of a memristive system with only one state variable, x. Such a memristive system is either current-controlled (3.3), in which case the internal state variable is the charge, q, controlled by a current I, and an output voltage, V, or it is voltage-controlled (3.4), in which case the system's state variable is the flux, ϕ, controlled by the voltage V, and the output of the system is the current, I.

$$\begin{aligned} V &= R(q) \times I; \\ \dot{q} &= I \end{aligned} \tag{3.3}$$

$$\begin{aligned} I &= G(\phi) \times V; \\ \dot{\phi} &= V; \end{aligned} \tag{3.4}$$

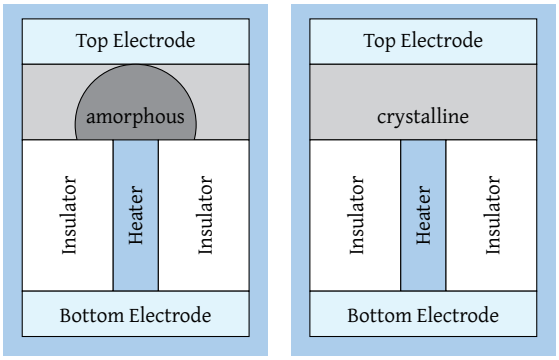
Overview of Memristor Technologies

In practice, several NVM technologies belong to Chua's memristor class:

- *PCM (Phase Change Memory)*, which switches crystalline material, e.g. chalcogenide glass, between amorphous and crystalline states by heat produced by the passage of an electric current through a heating element,
- *ReRAM* (*Resistive RAM*) with the two sub-classes
  - *CBRAM (Conductive Bridge RAM)*, which generates low resistance filament structures between two metal electrodes by ions exchange,
  - *OxRAM (Metal Oxide Resistive RAM)* consists of a bi-layer oxide structure, namely a metal-rich layer with lower resistivity (base layer) and an oxidised layer with higher resistivity. The ratio of the height of these two layers and by that the resistance of the whole structure can be changed by redistribution of oxygen vacancies,
  - *DioxRAM (Diode Metal Oxide Resistive RAM)*, in which oxygen vacancies are redistributed and trapped close to one of the two metal electrodes and lower the barrier height of corresponding metal electrode.
- *MRAM (Magnetoresistive RAM)* storing data by magnetic tunnel junctions (MTJ), which is a component consisting of two ferromagnets separated by a thin insulator,
- *STT-RAMs (Spin-Transfer Torque RAMs)* as newer technology that uses spin-aligned ("polarized") electrons to directly torque the domains, and
- *NRAM (Nano RAM)* based on Carbone-Nanotube-Technique.

The functioning of the just listed technologies are now described in more details.

PCM or also called *PRAM or PCRAM* is implemented by a material with thermally induced phase change property. The material changes its atomic structure from highly disordered, highly resistive amorphous structure to long ordered low resistive crystalline state. The structure of the PCM cell used in this work is referred to as mushroom cell as shown in Fig. 3.2.



(a) high resistance (b) low resistance

Figure 3.2: PCM cell structure [5]

In this structure a phase change material layer is sandwiched between two electrodes. When current passes through the heater it induces heat into the phase change layer and thereby eliciting the structure change. To read the data stored in the cell, a low amplitude reading pulse is applied, that is too small to induce phase change. By applying such a low reading voltage and measuring the current across the cell, its resistance and hence the binary stored value can be read out. To program the PCM cell into high resistance state, the temperature of the cell has to be higher than the melting temperature of the material, while to program the cell into the low resistance state the temperature of the cell must be well above the crystalizing temperature and below melting temperature for a duration sufficient for crystallization to take place [5].

PCM [6, 7, 8, 9, 10] can be integrated in the CMOS process and the read/write latency is only by tens of nanoseconds slower than DRAM whose latency is

roughly around 100ns. The write endurance is a hundred million or up to hundreds of millions of writes per cell at current processes. The resistivity of the memory element in PCM is more stable than Flash; at the normal working temperature of 85 °C, it is projected to retain data for 300 years. Moreover, PCM exhibits higher resistance to radiation than Flash memory. PCM is currently positioned mainly as a Flash replacement.

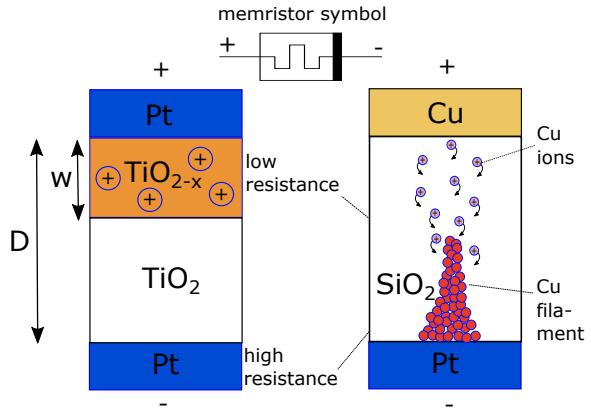


Figure 3.3: Scheme for OxRAM and CBRAM based memristive ReRAM devices.

ReRAM or also called *RRAM* offers a simple cell structure which enables reduced processing costs. Fig. 3.3 shows the technological scheme for ReRAM devices based on OxRAM or CBRAM. the different non-volatile resistance values are stored as follows.

In CBRAM [11] metal is used to construct the filaments, e.g. by applying a voltage on the top copper electrode Cu^+ ions are moving from the top electrode to the bottom negative electrode made in platinum. As result the positively charged copper ions reoxidize with electrons and a copper filament is growing that offers a lower resistance. By applying an opposite voltage this filament is removed and the increasing gap between the tip of the filament and the top electrode increases resulting in a higher resistance. In an OxRAM-based ReRAM [12, 13] oxygen ionization is exploited for the construction of layers with oxygen vacancies which have a lower resistivity. The thickness ratio in a bilayer oxide structure between the resistance switching layer with higher resistivity, e.g. TiO_{2-x}, and the base metal-rich layer with lower resistivity, e.g. TiO_2, see Fig. 3.3, is changed by redistribution of oxygen vacancies.

In bipolar OxRAM-based ReRAMs (DioxRAM), i.e. both electrodes can be connected to arbitrary voltages, oxygen vacancies are redistributed and trapped, e.g. by Ti ions in $BiFeO_3$ [14], close to one of the two metal electrodes. The accumulation of oxygen vacancies lowers the barrier height of the corresponding metal electrode [15]. If both metal electrodes have a reconfigurable barrier height, the DioxRAM works as a complementary resistance switch [16]. The resistance of the DioxRAM depends on the amplitude of writing bias and can be controlled in a fine-tuned analog manner [17]. Local ion irradiation improves the resistive switching at normal working temperature of 85 °C [18].

The endurance of ReRAM devices can be more than 50 million cycles and the switching energy is very low [19]. ReRAM can deliver 100x lower read latency and 20x faster write performance compared to NAND Flash [20]. In particular, CBRAM can be written with relatively low energy and with high speed featuring read/write latencies close to DRAM.

MRAM is a memory technology that uses the magnetism of electron spin to provide non-volatility without wear-out. MRAM stores information in magnetic material integrated with silicon circuitry to deliver the speed of SRAM with the non-volatility of Flash in a single unlimited-endurance device. Current MRAM technology from Everspin features a symmetric read-/write access of 35 ns, a data retention of more than 20 years, unlimited endurance, and a reliability that exceeds 20 years lifetime at 125 °C. It can easily be integrated with CMOS. [21]

MRAM requires only slightly more power to write than read, and no change in the voltage, eliminating the need for a charge pump. This leads to much faster operation and lower power consumption than Flash. Although MRAM is not quite as fast as SRAM, it is close enough to be interesting even in this role. Given its much higher density, a CPU designer may be inclined to use MRAM to offer a much larger but somewhat slower cache, rather than a smaller but faster one. [22]

STT (*spin-transfer torque* or *spin-transfer switching*) is a newer MRAM technology technique based on Spintronics, i.e. the technology of manipulating the spin state of electrons. STT uses spin-aligned ("polarized") electrons to directly torque the domains. Specifically, if the electrons flowing into a layer they have to change their spin, this will develop a torque that will be transferred to the nearby layer. This lowers the amount of current needed to write the cells, making it about the same as the read process.

Instead of using the electrons charge, spin states can be utilized as a substitute in logical circuits or in traditional memory technologies like SRAM. An STT-RAM [23] cell stores data in a magnetic tunnel junction (MTJ). Each MTJ is composed of two ferromagnetic layers (free and reference layers) and one tunnel barrier layer (MgO). If the magnetization direction of the magnetic fixed reference layer and the switchable free layer is anti-parallel, resp. parallel, a high, resp. a low, resistance is adjusted, representing a digital "0" or "1". Also multiple states can be stored. In [24] was reported that by adjusting intermediate magnetization angles in the free layer 16 different states can be stored in one physical cell, enabling to realize multi-cell storages in MTJ technology.

The read latency and read energy of STT-RAM is expected to be comparable to that of SRAM. The expected 3x higher density and 7x less leakage power consumption in the STT-RAM makes it suitable for replacing SRAMs to build large NVMs. However, a write operation in an STT-RAM memory consumes 8x more energy and exhibits a 6x longer latency than a SRAM. Therefore, minimizing the impact of inefficient writes is critical for successful applications of STT-RAM [25].

NRAMs, a proprietary technology of Nantero, are a very prospective NVM technology in terms of manufacturing maturity, according to their developers. The NRAMs are based on nano-electromechanical carbon nano tube switches (NEMS). In [27, 28] pinched hysteresis loops are shown for the current-voltage curve for such NEMS devices. Consequently, also NEMS and NRAMs are memristors according to Leon Chua's theory. The NRAM uses a fabric of carbon nanotubes (CNT) for saving bits. The resistive state of the CNT fabric determines, whether a one or a zero is saved in a memory cell. The resistance depends on the width of a bridge between two CNT. With the help of a small voltage, the CNTs can be brought into contact or be separated. Reading out a bit means to measure the resistance. Nantero claims that their technology features show the same read- and write latencies as DRAM, has a high endurance and reliability even in high temperature environments and is low power with essentially zero power consumption in standby mode. Furthermore NRAM is compatible with existing CMOS fabs without needing any new tools or processes, and it is scalable even to below 5 nm [29].

Fig. 3.4 gives an overview of some memristive devices' characteristics which was established by Stefan Slesazeck from NaMLab for a comparison of memristive devices with different $Hf0_2$-based ferroelectric memory technologies, FeRAM, FeFET, and ferroelectric tunnelling junction devices (FTJs), which can also be used for the realization of non-volatile memories. The table is just a snapshot of an assessment. Assessments of other authors differ widely in terms of better of worse values concerning different features. The International Technology Roadmap for Semiconductors (ITRS 2013) [30] reports an energy of operation of 6 pJ and projects 1 fJ for the year 2025 for PCMs. Jeong and Shi [31] report in 2019 an energy of operation of 80 fJ to 0.03 nJ for prototype and research PCM devices, 0.1 pJ to 10 nJ for RAM based devices, whereas the commercial OxRAM based ReRAMs from Panasonic have a write speed of 100 ns and an energy value of 50 pJ per memory cell. A record breaking energy efficiency is published in Vodenicarevic [32] for STT-MRAMs with 20 fJ/bit for a device area of 2 μm^2, compared to 3 pJ/bit and 4000 μm^2 for a state-of-the-art pure CMOS solution. The price for this perfect value is a limited speed dynamics of a few dozens MHz. However, for embedded IoT devices this can be sufficient. Despite of this distinguishing numbers it is clear that these devices offer a lot of potential and it is to expect that some of this potential can be exploited for future computer architectures.

The NVSim simulator [33] is popular in computer architecture science research to assess architectural structures based on the circuit-level performance, energy and area model of emerging non-volatile memories. It allows the investigation of architectural structures for future NVM based high-performance computers. Nevertheless, there is still a lot of work to do on the tool side. Better models for memristor technology, both physical and analytical ones, have to be integrated in the tools and besides that also the models themselves have to be fine tuned.

Multi-Level Cell Capability of Memristors

One of the most promising benefits that memristive technologies like ReRAM, PCMs, or STT-RAMs offer is their capability of storing more than two bits in one physical storage cell. MLC is necessary if memristors are used to emulate synaptic plasticity [34] (see Sect. 3.2.4. Compared to conventional SRAM or DRAM storage technology this is an additional qualitative advantage to their feature of non-volatility. In literature this benefit is often denoted as multi-level-cell

Parameter	DRAM	SRAM	NOR Flash	NAND Flash	PCM	classical ReRAM	STT-MRAM	FeFET HfO_2	FeRAM HfO_2	FTJ HfO_2
write time [ns]	10	2	10000	300000	100	20	8	20	10	10
read time [ns]	10	2	85	25000	60	10	10	10	10	1000
write voltage [V]	2,5	1,1	8,5	20	3	1,5	1	3	1,5	2,50
read voltage [V]	2,5	1,1	2,5	2,5	2	0,5	0,3	0,1	1,5	1,0
write current [uA]	2	3	10	1E-03	100	50	40	1E-03	1E-03	1E-03
power consumption	med	med	high	low	med	med	med	very low	very low	very low
write energy (pJ/cell)			850	6	30	1,5	0,32	6,0E-05	1,5E-05	2,5E-05
log10 (cycling endurance)	15	15	5	3	6	5	12	5	10	5

Figure 3.4: Snapshot of different memristive devices' characteristics and conventional Si-based memory technologies, established by S. Slesazeck // reprinted from S.Yu, P.-Y. Chen, Emerging Memory // Technologies, 2016 [26]

(MLC) or sometimes also as multi-bit capability. The different memristive technologies offer different benefits and drawbacks among each other concerning the realization of the MLC feature. Details about these benefits and drawbacks as well as the possibilities of usage of this MLC feature in future computing systems for caches, associative memories and ternary computing schemes can be found in Sect. 3.2.2.

Current State

The above mentioned memristor technologies PCM, ReRAMs, MRAM, STT-RAM, an advanced MRAM technology which uses a spin-polarized current instead of a magnetic field to store information in the electron's spin, allowing therefore higher integration densities, and NRAMs are memristor candidates, which are already commercialized or close to commercialization according to their manufacturers.

Intel and Micron already deliver the new 3D XPoint memory technology [35] as flash replacement which is based on PCM technology. Their Optane-SSDs 905P series is available on the market and offers 960 GByte for an about four times higher price than current 1 TByte SSD-NAND flash SSDs but provides 2.5 to 77 times better performance than NAND-SSDs. Intel and Micron expect that the X-Point technology could become the dominating technology as an alternative to RAM devices offering in addition NVM property in the next ten years. But the manufacturing process is complicated and currently, devices are expensive.

IBM published in 2016 achieved progress on a multi-level-cell (MLC-)PCM technology [36] replacing Flash and to use them e.g. as storage class memory (SCM) of supercomputers to fill the latency gap between DRAM main memory and the hard disk based background memory.

Adesto Technologies is offering CBRAM technology in their serial memory chips [37]. The company recently announced it will present new research showing the significant potential for Resistive RAM (RRAM) technology in high-reliability applications such as automotive. RRAM has great potential to become a widely used, low-cost and simple embedded non-volatile memory (NVM), as it utilizes simple cell structures and materials which can be integrated into existing manufacturing flows with as little as one additional mask. Adesto's RRAM technology (trademarked as CBRAM), making it a promising candidate for high-reliability applications. CBRAM consumes less power, requires fewer processing steps, and operates at lower voltages as compared to conventional embedded flash technologies [38].

MRAM is a NVM technology that is already available today, however in a niche market. MRAM chips are produced by Everspin Technologies, GlobalFoundries and Samsung [22].

Everspin delivered in 2017 samples of STT-MRAMs in perpendicular Magnetic Tunnel Junction Process (pMTJ) as 256-MBit-MRAMs und 1 GB-SSDs. Samsung is developing an MRAM technology. IBM and Samsung reported already in 2016 an MRAM device capable of scaling down to 11 nm with a switching current of 7.5 microamps at 10 ns [22]. Samsung and TSMC are producing MRAM products in 2018.

Everspin offers in August 2018 a 256Mb ST-DDR3 STT-MRAM storage device designed for enterprise-style applications like SSD buffers, RAID buffers or synchronous logging applications where performance is critical and endurance is a must. The persistence of STT-MRAM protects data and enables systems to dramatically reduce latency, by up to 90%, boosting performance and driving both efficiency and cost savings [21]. Everspin is focusing with their MRAM products

on areas where there is a need for fast, persistent memory by offering near-DRAM performance combined with non-volatility.

Right now, the price of MRAM is still rather high, but it is the most interesting emerging memory technology because its performance is close to SRAM and DRAM, and its endurance is very high. MRAM makes sense for cache buffering, and for specific applications, such as the nvNITRO NVMe storage accelerator for financial applications, where “doing a transaction quickly is important, but having a record is just as important" [39].

TSMC is also developing Embedded MRAM and Embedded ReRAM, as indicated by the TSMC roadmap in 2018 [40].

Nantero together with Fujitsu announced a Multi-GB-NRAM memory in Carbone-Nanotube-Technique in 2018. Having acquired the license to produce Nantero’s NRAM (Nano-RAM), Fujitsu targeted 2019 for NRAM Mass Production. Nantero’s CNT-based devices can be fabricated on standard CMOS production equipment, which may keep costs down. NRAM could be Flash replacement, able to match the densities of current Flash memories and, theoretically, it could be made far denser than Flash.

Nantero also announced a multi-gigabyte DDR4-compatible MRAM memory with speed comparable to DRAM at a lower price per gigabyte. Cache, based on nonvolatile technology, will remove the need for battery backup. Nantero said that this allows for a dramatic expansion of cache size, substantially speeding up the SSD or HDD. Embedded memory will eventually be able to scale to 5nm in size (the most advanced semiconductors are being produced at the 10-nm and 7-nm nodes); operate at DRAM-like speeds, and operate at very high temperature, said Nantero. The company said that the embedded memory devices will be well-suited for several IoT applications, including automotive. [41]

Perspective

It is foreseeable, that memristor technologies will supersede current Flash memory. Memristors offer orders of magnitude faster read/write accesses and also much higher endurance. They are resistive switching memory technologies, and thus rely on different physics than that of storing charge on a capacitor as is the case for SRAM, DRAM and Flash. Some memristor technologies have been considered as a feasible replacement for SRAM [42, 43, 44]. Studies suggest that replacing SRAM with STT-RAM could save 60% of LLC energy with less than 2% performance degradation [42].

Besides the potential as memories, memristors which are complementary switches offer a highly promising approach to realize memory and logic functionality in a single device, e.g. for reconfigurable logics [16], and memristors with multi-level cell capabilities enable the emulation of synaptic plasticity [34] to realize neuromorphic computing, e.g. for machine learning with memristor-based neural networks.

One of the challenges for the next decade is the provision of appropriate interfacing circuits between the SCMs, or NVM technologies in general, and the microprocessor cores. One of the related challenges in this context is the developing of efficient interface circuits in such a way that this additional overhead will not corrupt the benefits of memristor devices in integration density, energy consumption and access times compared to conventional technologies.

STT-RAM devices primarily target the replacement of DRAM, e.g., in Last-Level Caches (LLC). However, the asymmetric read/write energy and latency of NVM technologies introduces new challenges in designing memory hierarchies. Spintronic allows integration of logic and storage at lower power consumption. Also new hybrid PCM / Flash SSD chips could emerge with a processor-internal last-level cache (STT-RAM), main processor memory (ReRAM, PCRAM), and storage class memory (PCM or other NVM).

All commercially available memristive memories feature better characteristics than Flash, however, are much more expensive. It is unclear when most of the new technologies will be mature enough and which of them will prevail by a competitive price. “It’s a veritable zoo of technologies and we’ll have to wait and see which animals survive the evolutionary process," said Thomas Coughlin, founder of Coughlin Associates.

One of the most promising benefits that memristive technologies like ReRAM, PCMs, or STT-RAMs offer is their capability of storing more than two bits in one physical storage cell. Compared to conventional SRAM or DRAM storage technology this is an additional qualitative advantage to their feature of non-volatility.

3.2.2 Multi-level-cell (MLC)

The different memristive technologies offer different benefits and drawbacks among each other concerning the realization of the MLC feature. E.g., one of the main challenges in MLC-PCM systems is the read reliability degradation due to resistance drift [45]. Resistance drift means that the different phase states in the used chalcogenide storage material can overlap since each reading step changes a little bit the phase what is not a real problem in single-level cells (SLC) but in MLCs. In a recently published work the impressive number of 92 distinct resistance levels was demonstrated for a so-called bi-layer ReRAM structure [46]. In such a bi-layer structure not only one metal-oxide layer is used as storage material, like e.g. usually HfO2 or TiO2 technology, which is enclosed between a metallic top and a bottom electrode. Moreover, a sequence of metal-oxide layers separated by an isolating layer is used leading to a better separation of different resistance levels for the prize of a much more difficult manufacturing process. Memristive MLC technique based on MRAM technology without spin-polarized electrons was proposed to store up to 8 different levels [47]. In STT-MRAM technology, using spin-polarized electrons, 2-bit cells are most common and were also physically demonstrated on layout level [48].

MLC as Memory

In its general SLC form STT-MRAM is heavily discussed as a candidate memory technology for near-term realization of future last-level-caches due to its high density characteristics and comparatively fast read-/write access latencies. On academic site the next step is discussed how to profit from the MLC capability [49].

The last example, concerning MLC caches, is representative for all memristive NVM technologies and their MLC capability. It shows that the MLC feature are of interest for improving the performance of future computer or processor architectures. In this context they are closely related to future both near-memory and in-memory computing concepts for both future embedded HPC systems and embedded smart devices for IoT and CPS. For near-memory-computing architectures, e.g. as embedded memories, they can be used for a better high-performance multi-bit cache in which different tasks store their cached values in the same cache line.

Another more or less recent state-of-the-art application is their use in micro-controller units as energy-efficient, non-volatile check-pointing or normally-off/instant-on operation with near zero latency boot as it was just announced by the French start-up company eVaderis SA [50].

To this context also belongs research work on ternary content-addressable memories (TCAM) with memristive devices, in which the third state is used for the realization of the don't care state in TCAMs. In many papers, e.g. in [51], is shown that using memristive TCAMs need less energy, less area than equivalent CMOS TCAMs. However, most of the proposed memristive TCAM approaches don't exploit the MLC capability. They are using three memristors to store 1, 0, and X (don't care). In a next step this can be expanded to exploit the MLC capability of such devices for a further energy and area improvement.

Ternary Arithmetic Based on Signed-Digit (SD) Number Systems

Another promising aspect of the MLC capability of memristive devices is to use them in ternary arithmetic circuits or processors based on signed-digit (SD) number systems. In a SD number system a digit can have also a positive and a negative value, e.g. for the ternary case we have not given a bit but a trit with the values, -1, 0, and +1. It is long known that ternary or redundant number systems generally, in which more than two states per digit are mandatory, improve the effort of an addition to a complexity of O(1) compared to log(N) which can be achieved in the best case with pure binary adders. In the past conventional computer architectures did not exploit this advantage of signed-digit addition. One exception was the compute unit of the ILLIAC III [52] computer manufactured in the 1960's, at a time when the technology was not so mature than today and it was necessary to achieve high compute speeds with a superior arithmetic concept even for paying a price of doubling the memory requirements to store a ternary value in two physical memory cells. In course of the further development the technology and pipeline processing offering latency hiding, the ALUs become faster and faster and it was not acceptable for storing operands given in a redundant representation that is larger than a binary

one. This would double the number of registers, double the size of the data cache and double the necessary size of data segments in main memory. However, with the occurrence of CMOS-compatible NVM technology offering MLC capability the situation changed. This calls for a re-evaluation of these redundant computer arithmetic schemes under a detailed consideration of the technology of MLC NVM.

Perspectives and Research Challenges

Different work has already investigated the principal possibilities of ternary coding schemes using MLC memristive memories. This was carried out both for hybrid solutions, i.e. memristors are used as ternary memory cells for digital CMOS based logic circuits [53], [54], and in proposals for in-memory computing like architectures, in which the memristive memory cell was used simultaneously as storage and as logical processing element as part of a resistor network with dynamically changing resistances [55]. The goal of this work using MLC NVM technology for ternary processing is not only to save latency but also to save energy since the number of elementary compute steps is reduced compared to conventional arithmetic implemented in state-of-the-art processors. This reduced number of processing steps should also lead to reduced energy needs.

As own so far unpublished work, carried out in the group of the author of this chapter, shows that in CMOS combinatorial processing, i.e. without storing the results, the energy consumption could be reduced about 30 % using a ternary adder compared to the best parallel pre-fix binary adders for a 45 nm CMOS process. This advantage is lost if the results are stored in binary registers. To keep this advantage and exploit it in IoT and embedded devices, which are "energy-sensible" in particular, ternary storage and compute schemes based on MLC based NVMs have to be integrated in future in near- and in-memory computing schemes.

To achieve this goal, research work is necessary on following topics: (i) design tools, considering automatic integration and evaluation of NVMs in CMOS, what (ii) requires the development of appropriate physical models not only on analogue layer but also on logic and RTL level, (iii) appropriate interface circuitry for addressing NVMs, and (iv) in general the next step that has to be made is going from existing concepts and demonstrated single devices to real systems.

References

[1] Wikipedia. *Memristor*. URL: http://en.wikipedia.org/wiki/Memristor.

[2] A. L. Shimpi. *Samsung's V-NAND: Hitting the Reset Button on NAND Scaling*. AnandTech. 2013. URL: http://www.anandtech.com/show/7237/samsungs-vnand-hitting-the-reset-button-on-nand-scaling.

[3] L. Chua. "Memristor-The missing circuit element". In: *IEEE Transactions on Circuit Theory, Vol. 18, Iss. 5*. 1971.

[4] D. Strukov, G. Snider, D. Stewart, and R. Williams. "The missing memristor found". In: *Nature* 453 (2008).

[5] P. W. C. Ho, N. H. El-Hassan, T. N. Kumar, and H. A. F. Almurib. "PCM and Memristor based nanocrossbars". In: *2015 IEEE 15th International Conference on Nanotechnology (IEEE-NANO)*. IEEE. 2015. URL: https://ieeexplore.ieee.org/%20document/7388636/.

[6] B. C. Lee, P. Zhou, J. Yang, Y. Zhang, B. Zhao, E. Ipek, O. Mutlu, and D. Burger. "Phase-change Technology and the Future of Main Memory". In: *IEEE micro* 30.1 (2010).

[7] C. Lam. "Cell Design Considerations for Phase Change Memory as a Universal Memory". In: *VLSI Technology, Systems and Applications*. IEEE. 2008, pp. 132–133.

[8] B. C. Lee, E. Ipek, O. Mutlu, and D. Burger. "Architecting Phase Change Memory As a Scalable Dram Alternative". In: *Proceedings of the 36th Annual International Symposium on Computer Architecture*. ISCA '09. 2009, pp. 2–13.

[9] M. K. Qureshi, V. Srinivasan, and J. A. Rivers. "Scalable High Performance Main Memory System Using Phase-change Memory Technology". In: *Proceedings of the 36th Annual International Symposium on Computer Architecture*. ISCA '09. 2009, pp. 24–33.

[10] P. Zhou, B. Zhao, J. Yang, and Y. Zhang. "A Durable and Energy Efficient Main Memory Using Phase Change Memory Technology". In: *Proceedings of the 36th Annual International Symposium on Computer Architecture*. ISCA '09. 2009, pp. 14–23.

[11] W. Wong. *Conductive Bridging RAM*. electronic design. 2014. URL: http://electronicdesign.com/memory/conductive-bridging-ram.

[12] C. Xu, D. Niu, N. Muralimanohar, R. Balasubramonian, T. Zhang, S. Yu, and Y. Xie. "Overcoming the Challenges of Crossbar Resistive Memory Architectures". In: *21st International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2015, pp. 476–488.

[13] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie. "Design Implications of Memristor-based RRAM Cross-point Structures". In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*. IEEE. 2011, pp. 1–6.

[14] Y. Shuai et al. "Substrate effect on the resistive switching in BiFeO3 thin films". In: *Journal of Applied Physics* 111.7 (Apr. 2012), p. 07D906. DOI: 10.1063/1.3672840.

[15] T. You et al. "Bipolar Electric-Field Enhanced Trapping and Detrapping of Mobile Donors in BiFeO3 Memristors". In: *ACS Applied Materials & Interfaces* 6.22 (2014), pp. 19758–19765. DOI: 10.1021/am504871g.

[16] T. You et al. “Exploiting Memristive BiFeO3 Bilayer Structures for Compact Sequential Logics”. In: *Advanced Functional Materials* 24.22 (2014), pp. 3357–3365. DOI: 10.1002/adfm.201303365.

[17] N. Du et al. “Field-Driven Hopping Transport of Oxygen Vacancies in Memristive Oxide Switches with Interface-Mediated Resistive Switching”. In: *Phys. Rev. Applied* 10 (5 Nov. 2018), p. 054025. DOI: 10.1103/PhysRevApplied.10.054025.

[18] X. Ou et al. “Forming-Free Resistive Switching in Multiferroic BiFeO3 thin Films with Enhanced Nanoscale Shunts”. In: *ACS Applied Materials & Interfaces* 5.23 (2013), pp. 12764–12771. DOI: 10.1021/am404144c.

[19] B. Govoreanu et al. “10x10nm 2 Hf/HfO x crossbar resistive RAM with excellent performance, reliability and low-energy operation”. In: *International Electron Devices Meeting (IEDM)*. IEEE. 2011, pp. 31–6.

[20] Crossbar. *ReRAM Advantages*. URL: https://www.crossbar-inc.com/en/technology/reram-advantages/.

[21] E. technologies. *STT-MRAM Products*. URL: https://www.everspin.com/stt-mram-products.

[22] Wikipedia. *Magnetoresistive random-access memory*. URL: http://en.wikipedia.org/wiki/Magnetoresistive_random-access_memory.

[23] D. Apalkov et al. “Spin-transfer Torque Magnetic Random Access Memory (STT-MRAM)”. In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 9.2 (2013), p. 13.

[24] *Spintronic Devices for Memristor Applications*. Talk at Meeting of EU COST ACTION MemoCis IC1401, "Memristors: at the crossroad of Devices and Applications", Milano, 28th March 2016. Mar. 2016.

[25] H. Noguchi et al. “A 250-MHz 256b-I/O 1-Mb STT-MRAM with Advanced Perpendicular MTJ Based Dual Cell for Nonvolatile Magnetic Caches to Reduce Active Power of Processors”. In: *VLSI Technology (VLSIT), 2013 Symposium on*. IEEE. 2013, pp. 108–109.

[26] S.Yu and P.-Y. Chen. “Emerging Memory Technologies”. In: *SPRING 2016 IEEE Solid-sate circuits magazine*. 2016.

[27] Z. F. ; X. F. ; A. L. ; L. Dong. “Resistive switching in copper oxide nanowire-based memristor”. In: *12th IEEE International Conference on Nanotechnology (IEEE-NANO)*. 2012.

[28] Q. L. ; S.-M. K. ; C. A. R. ; M. D. E. ; J. E. Bon. “Precise Alignment of Single Nanowires and Fabrication of Nanoelectromechanical Switch and Other Test Structures”. In: *IEEE Transactions on Nanotechnology, Vol. 6 , Iss. 2*. 2007.

[29] Nantero. *Nantero NRAM Advances in Nanotechnology*. URL: http://nantero.com/technology/.

[30] *International Technology Roadmap for Semiconductors (ITRS), Emerging research devices, 2013*. URL: http://www.itrs2.net/itrs-reports.html.

[31] H. Jeong and L. Shi. “Memristor devices for neural networks”. In: *Journal of Physics D: Applied Physics* 52.2 (Jan. 2019), p. 023003. DOI: 10.1088/1361-6463/aae223.

[32] D. Vodenicarevic et al. “Low-Energy Truly Random Number Generation with Superparamagnetic Tunnel Junctions for Unconventional Computing”. In: *Physical Review Applied* 8.5 (). DOI: 10.1103/PhysRevApplied.8.054045. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.8.054045.

[33] X. Dong, C. Xu, N. Jouppi, and Y. Xie. “NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31.7 (July 2012), pp. 994–1007. DOI: 10.1109/TCAD.2012.2185930.

[34] N. Du, M. Kiani, C. G. Mayr, T. You, D. Bürger, I. Skorupa, O. G. Schmidt, and H. Schmidt. “Single pairing spike-timing dependent plasticity in BiFeO3 memristors with a time window of 25 ms to 125 ms”. In: *Frontiers in Neuroscience* 9 (June 2015), p. 227. DOI: 10.3389/fnins.2015.00227.

[35] J. Evangelho. *Intel And Micron Jointly Unveil Disruptive, Game-Changing 3D XPoint Memory, 1000x Faster Than NAND*. Hot Hardware. 2015. URL: https://hothardware.com/news/intel-and-micron-jointly-drop-disruptive-game-changing-3d-xpoint-cross-point-memory-1000x-faster-than-nand.

[36] N. Papandreou, H. Pozidis, T. Mittelholzer, G. Close, M. Breitwisch, C. Lam, and E. Eleftheriou. “Drift-tolerant Multilevel Phase-Change Memory”. In: *3rd IEEE International Memory Workshop (IMW)*. 2011.

[37] Adesto Technologies. *Mayriq Products*. URL: http://www.adestotech.com/products/mavriq/.

[38] P. Release. *Adesto Demonstrates Resistive RAM Technology Targeting High-Reliability Applications such as Automotive*. URL: https://www.adestotech.com/news-detail/adesto-demonstrates-resistive-ram-technology-targeting-high-reliability-applications-such-as-automotive/.

[39] G. Hilson. *Everspin Targets Niches for MRAM*. URL: https://www.eetimes.com/document.asp?doc_id=1332871.

[40] TSMC. *eFlash*. URL: http://www.tsmc.com/english/dedicatedFoundry/technology/eflash.htm.

[41] B. Santo. *NRAM products to come in 2019*. URL: https://www.electronicproducts.com/Digital_ICs/Memory/NRAM_products_to_come_in_2019.aspx.

[42] H. Noguchi, K. Ikegami, N. Shimomura, T. Tetsufumi, J. Ito, and S. Fujita. “Highly Reliable and Low-power Nonvolatile Cache Memory with Advanced Perpendicular STT-MRAM for High-performance CPU”. In: *Symposium on VLSI Circuits Digest of Technical Papers*. IEEE. June 2014, pp. 1–2.

[43] H. Noguchi et al. “A 3.3ns-access-time 71.2 uW/MHz 1Mb Embedded STT-MRAM Using Physically Eliminated Read-disturb Scheme and Normally-off Memory Architecture”. In: *International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2015, pp. 1–3.

[44] J. Ahn, S. Yoo, and K. Choi. “DASCA: Dead Write Prediction Assisted STT-RAM Cache Architecture”. In: *International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2014, pp. 25–36.

[45] W. Zhang and T. Li. “Helmet: A resistance drift resilient architecture for multi-level cell phase change memory system”. In: *IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*. 2011, pp. 197–208.

[46] S. Stathopoulos et al. “Multibit memory operation of metal-oxide bi-layer memristors”. In: *Scientific Reports* 7 (2017), p. 17532.

[47] H. Cramman, D. S. Eastwood, J. A. King, and D. Atkinson. “Multilevel 3 Bit-per-cell Magnetic Random Access Memory Concepts and Their Associated Control Circuit Architectures”. In: *IEEE Transactions on Nanotechnology* 11 (2012), pp. 63–70.

[48] L. Xue et al. “An Adaptive 3T-3MTJ Memory Cell Design for STT-MRAM-Based LLCs”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 99 (2018), pp. 1–12.

[49] L. Liu, P. Chi, S. Li, Y. Cheng, and Y. Xie. “Building energy-efficient multi-level cell STT-RAM caches with data compression”. In: *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. Jan. 2017, pp. 751–756. DOI: 10.1109/ASPDAC.2017.7858414.

[50] *Startup tapes out MRAM-based MCU*. 2018. URL: http://www.eenewsanalog.com/news/startup-tapes-out-mram-based-mcu.

[51] Q. Guo et al. “A Resistive TCAM Accelerator for Data-Intensive Computing”. In: *MICRO’11*.

[52] D. E. Atkins. “Design of the Arithmetic Units of ILLIAC III: Use of Redundancy and Higher Radix Methods”. In: *IEEE Transactions on Computers* C-19 (1970), pp. 720–722.

[53] D. Fey, M. Reichenbach, C. Söll, M. Biglari, J. Röber, and R. Weigel. “Using Memristor Technology for Multi-value Registers in Signed-digit Arithmetic Circuits”. In: *MEMSYS 2016*. 2016, pp. 442–454.

[54] D. Wust, D. Fey, and J. Knödtel. “A programmable ternary CPU using hybrid CMOS memristor circuits”. In: *Int. JPDES* (2018). DOI: https://doi.org/10.1080/17445760.2017.1422251.

[55] A. A. El-Slehdar, A. H. Fouad, and A. G. Radwan. “Memristor-based redundant binary adder”. In: *International Conference on Engineering and Technology (ICET)*. 2014, pp. 1–5.

3.2.3 Memristive Computing

In this section, memristive (also called resistive) computing is discussed in which logic circuits are built by memristors [1].

Overview of Memristive Computing

Memristive computing is one of the emerging and promising computing paradigms [1, 2, 3]. It takes the data-centric computing concept much further by interweaving the processing units and the memory in the same physical location using non-volatile technology, therefore significantly reducing not only the power consumption but also the memory bottleneck. Resistive devices such as memristors have been shown to be able to perform both storage and logic functions [1, 4, 5, 6].

Memristive gates have a lower leakage power, but switching is slower than in CMOS gates [7]. However, the integration of memory into logic allows to reprogram the logic, providing low power reconfigurable components [8], and can reduce energy and area constraints in principle due to the possibility of computing and storing in the same device (computing in memory). Memristors can also be arranged in parallel networks to enable massively parallel computing [9].

Memristive computing provides a huge potential as compared with the current state-of the art:

- It significantly reduces the memory bottleneck as it interweaves the storage, computing units and the communication [1, 2, 3].
- It features low leakage power [7].
- It enables maximum parallelism [3, 9] by in-memory computing.
- It allows full configurability and flexibility [8].
- It provides order of magnitude improvements for the energy-delay product per operations, the computation efficiency, and performance per area [3].

Serial and parallel connections of memristors were proposed for the realization of Boolean logic gates with memristors by the so-called *memristor ratio logic*. In such circuits the ratio of the stored resistances in memristor devices is exploited for the set-up of Boolean logic. Memristive circuits realizing AND gates, OR gates, and the implication function were presented in [10, 11, 12].

Hybrid memristive computing circuits consist of memristors and CMOS gates. The research of Singh [13], Xia et.al. [14], Rothenbuhler et.al. [12], and Guckert and Swartzlaender [15] are representative for numerous proposals of hybrid memristive circuits, in which most of the Boolean logic operators are handled in the memristors and the CMOS transistors are mainly used for level restoration to retain defined digital signals.

Figure 3.5 summarizes the activities on memristive computing. We have the large block of hardware support with memristive elements for neural networks, neuromorphic processing, and STDP (spike-timing-dependent plasticity)(see Sect. 3.2.4). Concerning the published papers a probably much smaller branch of digital memristive computing with several sub branches, like ratioed logic, imply logic or CMOS-like equivalent memristor circuits in which Boolean logic is directly mapped onto crossbar topologies with memristors. These solutions refer to pure in-memory computing concepts. Besides that, proposals for hybrid solutions exist in which the memristors are used as memory for CMOS circuits in new arithmetic circuits exploiting the MLC capability of memristive devices.

Current State of Memristive Computing

A couple of start-up companies appeared in 2015 on the market who offer memristor technology as BEOL (Back-end of line) service in which memristive elements are post-processed in CMOS chips directly on top of the last metal layers. Also some European institutes reported just recently at a workshop meeting "Memristors: at the crossroad of Devices and Applications" of the EU cost action 1401 MemoCiS[1] the possibility of BEOL integration of their memristive technology to allow experiments with such technologies [16]. This offers new perspectives in form of hybrid CMOS/memristor logic which uses memristor networks for high-dense resistive logic circuits and CMOS inverters for signal restoration to compensate the loss of full voltage levels in memristive networks.

Multi-level cell capability of memristive elements can be used to face the challenge to handle the expected huge amount of Zettabytes produced annually in a

[1] www.cost.eu/COST_Actions/ict/IC1401

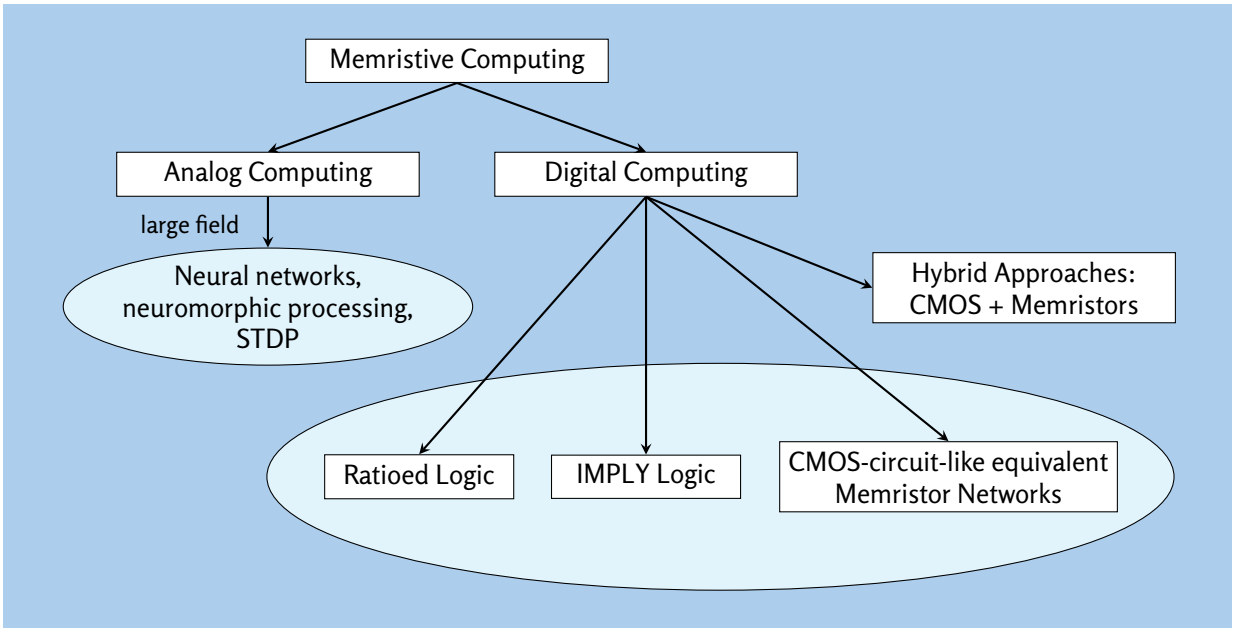


Figure 3.5: Summary of activities on memristive computing.

couple of years. Besides, proposals exist to exploit the multi-level cell storing property for ternary carry-free arithmetic [17, 18] or for both compact storing of keys and matching operations in future associative memories realized with memristors [19], so-called ternary content-addressable memories.

Impact on Hardware

Using NVM technologies for resistive computing is a further step towards energy-aware measures for future HPC architectures. In addition, there exist technology facilities at the IHP in Frankfurt/O which at least for small feature sizes allow to integrate memristors and CMOS logic in an integrated chip without a separate BEOL process step. It supports the realization of both near-memory and in-memory computing concepts which are both an important brick for the realization of more energy-saving HPC systems. Near-memory could be based on 3D stacking of a logic layer with DRAMs, e.g. extending Intel's High Bandwidth Memory (HBM) with NVM devices and stacked logic circuitry in future. In-memory computing could be based on memristive devices using either ReRAM, PCM, or STT-RAM technology for simple logic and arithmetic pre-processing operations.

A further way to save energy, e.g. in near-memory computing schemes, is to use non-volatile register cells as flip-flops or in memory cell arrays. During the last decade, the basic principle in the design of non-volatile FlipFlop (nvFF) has been to compose them from a standard CMOS Flip-Flop (FF) and a *non-volatile* memory cell, either as part of a flip-flop memristor register pair or as pair of a complete SRAM cell array and a subsequent attached memristor cell array (hybrid NVMs). At predefined time steps or on power loss, this *non-volatile* memory cell backups the contents of the standard FF. At power recovery, this content is restored in the FF and the Non-Volatile-Processor (NVP) can continue at the exact same state. nvFFs following this approach require a centralized controller to initiate a backup or a restore operation. This centralized controller has to issue the backup signal as fast as possible after a no-power standby, otherwise data and processing progress may be lost.

Four different implementation categories of nvFFs using hybrid retention architectures are available today:

- **Ferroelectric nvFF:** This category uses a ferroelectric capacitor to store one bit. Masui et al. [21] introduced this kind of nvFFs, but different approaches are also available.
- **Magnetic RAM (MRAM) nvFF:** This approach uses the spin direction of Magnetic Tunnel Junctions to store a bit. [22]
- **CAAC-OS nvFF:** CAAC-OS transistors have an extremely low off-state current. By combining

nvFF	FeRAM	MRAM	ReRAM	CAAC-OS
Technology	130nm	90nm	180nm	1um
Store Time	320ns	4ns	10ns	40ns
Store Energy(pJ/bit)	2.2	6	0.84	1.6
Recall Time	384ns	5ns	3.2ns	8ns
Recall Energy(pJ/bit)	0.66	0.3	N/A	17.4

Table 3.1: Performance comparison of nvFF types[20]

them with small capacitors a nvFF can be created[23]. The access times of these nvFFs are very low.

- **Resistive RAM (ReRAM) nvFF:** ReRAMs are a special implementation of NVM using memristor technology. They do not consume any power in their off-state. nvFFs implementations using ReRAM are currently evaluated [24, 25].

These approaches can also be applied to larger hybrid NVMs, where data, which has to be processed, is stored in conventional faster SRAM/DRAM devices. By using pipeline schemes, e.g. under control of the OS, part of the data is shifted from NVM to SRAM/-DRAM before it is accessed in the fast memory. Then, the latency for the data transfer from NVM to DRAM can be hidden by a timely overlapping of data transfer with simultaneous processing of other parts of the DRAM. The same latency hiding principle can happen in the opposite direction. Data that is newly computed and that is not needed in the next computing steps can be saved in NVMs.

Table 3.1 displays performance parameters of these nvFFs. According to overall access time and energy requirements the MRAM and the ReRAM approach are the most promising ones. But the ReRAM approach has more room for improvements because the fabrication technology is still very large compared to the current standard of seven nanometer. Table 3.1 also shows the impact memristor technology can have on NVPs. At the moment memristor-based nvFFs are only produced for research at a very large fabrication process of 180 nm. Still they can compete with nvFFs produced at a much smaller size, using a different technology.

Research papers propose an integrated FF design either by using a single memristor combined with pass transistors and a high-valued resistor [26] or by using a sense amplifier reading the differential state of two memristors, which are controlled by two transmission gates [27]. The latter approach seems to be beneficial in terms of performance, power consumption, and robustness and shows a large potential to be used for no-power standby devices which can be activated instantaneously upon an input event.

Perspective

Memristive computing, if successful, will be able to significantly reduce the power consumption and enable massive parallelism, hence, increase computing energy and area efficiency by orders of magnitudes. This will transform computer systems into new highly parallel architectures and associated technologies, and enable the computation of currently infeasible big data and data-intensive applications, fuelling important societal changes.

Research on resistive computing is still in its infancy stage, and the challenges are substantial at all levels, including material and technology, circuit and architecture, tools and compilers, and algorithms. As of today most of the work is based on simulations and small circuit designs. It is still unclear when the technology will be mature and available. Nevertheless, some start-ups on memristor technologies are emerging such as KNOWM[2], BioInspired[3], and Crossbar[4].

[2] www.knowm.org
[3] http://www.bioinspired.net/
[4] https://www.crossbar-inc.com/en/

References

[1] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams. "Memristive switches enable stateful logic operations via material implication". In: *Nature* 464.7290 (Apr. 2010), pp. 873–876. DOI: 10.1038/nature08940. URL: https://www.nature.com/nature/journal/v464/n7290/full/nature08940.html.

[2] M. Di Ventra and Y. V. Pershin. "Memcomputing: a computing paradigm to store and process information on the same physical platform". In: *Nature Physics* 9.4 (Apr. 2013), pp. 200–202. DOI: 10.1038/nphys2566. URL: http://arxiv.org/abs/1211.4487.

[3] S. Hamdioui et al. "Memristor based computation-in-memory architecture for data-intensive applications". In: *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*. May 2015, pp. 1718–1725.

[4] G. Snider. "Computing with hysteretic resistor crossbars". In: *Applied Physics A* 80.6 (Mar. 2005), pp. 1165–1172. DOI: 10.1007/s00339-004-3149-1. URL: https://link.springer.com/article/10.1007/s00339-004-3149-1.

[5] L. Gao, F. Alibart, and D. B. Strukov. "Programmable CMOS/Memristor Threshold Logic". In: *IEEE Transactions on Nanotechnology* 12.2 (Mar. 2013), pp. 115–119. DOI: 10.1109/TNANO.2013.2241075.

[6] L. Xie, H. A. D. Nguyen, M. Taouil, S. Hamdioui, and K. Bertels. "Fast boolean logic mapped on memristor crossbar". In: *2015 33rd IEEE International Conference on Computer Design (ICCD)*. Oct. 2015, pp. 335–342. DOI: 10.1109/ICCD.2015.7357122.

[7] Y. V. Pershin and M. D. Ventra. "Neuromorphic, Digital, and Quantum Computation With Memory Circuit Elements". In: *Proceedings of the IEEE* 100.6 (June 2012), pp. 2071–2080. DOI: 10.1109/JPROC.2011.2166369.

[8] J. Borghetti, Z. Li, J. Straznicky, X. Li, D. A. A. Ohlberg, W. Wu, D. R. Stewart, and R. S. Williams. "A hybrid nanomemristor/transistor logic circuit capable of self-programming". In: *Proceedings of the National Academy of Sciences* 106.6 (Feb. 2009), pp. 1699–1703. DOI: 10.1073/pnas.0806642106. URL: http://www.pnas.org/content/106/6/1699.

[9] Y. V. Pershin and M. Di Ventra. "Solving mazes with memristors: a massively-parallel approach". In: *Physical Review E* 84.4 (Oct. 2011). DOI: 10.1103/PhysRevE.84.046703. URL: http://arxiv.org/abs/1103.0021.

[10] J. J. Yang, D. B. Strukov, and D. R. Stewart. "Memristive devices for computing". In: *Nature Nanotechnology* 8.1 (Jan. 2013), pp. 13–24. DOI: 10.1038/nnano.2012.240. URL: http://www.nature.com/nnano/journal/v8/n1/full/nnano.2012.240.html.

[11] S. Kvatinsky, A. Kolodny, U. C. Weiser, and E. G. Friedman. "Memristor-based IMPLY Logic Design Procedure". In: *Proceedings of the 2011 IEEE 29th International Conference on Computer Design*. ICCD '11. Washington, DC, USA, 2011, pp. 142–147. DOI: 10.1109/ICCD.2011.6081389. URL: http://dx.doi.org/10.1109/ICCD.2011.6081389.

[12] T. Tran, A. Rothenbuhler, E. H. B. Smith, V. Saxena, and K. A. Campbell. "Reconfigurable Threshold Logic Gates using memristive devices". In: *2012 IEEE Subthreshold Microelectronics Conference (SubVT)*. Oct. 2012, pp. 1–3. DOI: 10.1109/SubVT.2012.6404301.

[13] T. Singh. "Hybrid Memristor-CMOS (MeMOS) based Logic Gates and Adder Circuits". In: *arXiv:1506.06735 [cs]* (June 2015). URL: http://arxiv.org/abs/1506.06735.

[14] Q. Xia et al. "Memristor?CMOS Hybrid Integrated Circuits for Reconfigurable Logic". In: *Nano Letters* 9.10 (Oct. 2009), pp. 3640–3645. DOI: 10.1021/nl901874j. URL: http://dx.doi.org/10.1021/nl901874j.

[15] L. Guckert and E. E. Swartzlander. "Dadda Multiplier designs using memristors". In: *2017 IEEE International Conference on IC Design and Technology (ICICDT)*. 2017.

[16] J. Sandrini, M. Thammasack, T. Demirci, P.-E. Gaillardon, D. Sacchetto, G. De Micheli, and Y. Leblebici. "Heterogeneous integration of ReRAM crossbars in 180 nm CMOS BEoL process". In: 145 (Sept. 2015).

[17] A. A. El-Slehdar, A. H. Fouad, and A. G. Radwan. "Memristor based N-bits redundant binary adder". In: *Microelectronics Journal* 46.3 (Mar. 2015), pp. 207–213. DOI: 10.1016/j.mejo.2014.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0026269214003541.

[18] D. Fey. "Using the multi-bit feature of memristors for register files in signed-digit arithmetic units". In: *Semiconductor Science and Technology* 29.10 (2014), p. 104008. DOI: 10.1088/0268-1242/29/10/104008. URL: http://stacks.iop.org/0268-1242/29/i=10/a=104008.

[19] P. Junsangsri, F. Lombardi, and J. Han. "A memristor-based TCAM (Ternary Content Addressable Memory) cell". In: *2014 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. July 2014, pp. 1–6. DOI: 10.1109/NANOARCH.2014.6880478.

[20] F. Su, Z. Wang, J. Li, M. F. Chang, and Y. Liu. "Design of nonvolatile processors and applications". In: *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. Sept. 2016, pp. 1–6. DOI: 10.1109/VLSI-SoC.2016.7753543.

[21] S. Masui, W. Yokozeki, M. Oura, T. Ninomiya, K. Mukaida, Y. Takayama, and T. Teramoto. "Design and applications of ferroelectric nonvolatile SRAM and flip-flop with unlimited read/program cycles and stable recall". In: *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, 2003.* Sept. 2003, pp. 403–406. DOI: 10.1109/CICC.2003.1249428.

[22] W. Zhao, E. Belhaire, and C. Chappert. "Spin-MTJ based Non-volatile Flip-Flop". In: *2007 7th IEEE Conference on Nanotechnology (IEEE NANO)*. Aug. 2007, pp. 399–402. DOI: 10.1109/NANO.2007.4601218.

[23] T. Aoki et al. "30.9 Normally-off computing with crystalline InGaZnO-based FPGA". In: *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. Feb. 2014, pp. 502–503. DOI: 10.1109/ISSCC.2014.6757531.

[24] I. Kazi, P. Meinerzhagen, P. E. Gaillardon, D. Sacchetto, A. Burg, and G. D. Micheli. "A ReRAM-based non-volatile flip-flop with sub-VT read and CMOS voltage-compatible write". In: *2013 IEEE 11th International New Circuits and Systems Conference (NEWCAS)*. June 2013, pp. 1–4. DOI: 10.1109/NEWCAS.2013.6573586.

[25] A. Lee et al. "A ReRAM-Based Nonvolatile Flip-Flop With Self-Write-Termination Scheme for Frequent-OFF Fast-Wake-Up Nonvolatile Processors". In: *IEEE Journal of Solid-State Circuits* 52.8 (Aug. 2017), pp. 2194–2207. DOI: 10.1109/JSSC.2017.2700788.

[26] J. Zheng, Z. Zeng, and Y. Zhu. "Memristor-based nonvolatile synchronous flip-flop circuits". In: *2017 Seventh International Conference on Information Science and Technology (ICIST)*. Apr. 2017, pp. 504–508. DOI: 10.1109/ICIST.2017.7926812.

[27] S. Pal, V. Gupta, and A. Islam. "Variation resilient low-power memristor-based synchronous flip-flops: design and analysis". In: *Microsystem Technologies* (July 2018). DOI: 10.1007/s00542-018-4044-6. URL: https://doi.org/10.1007/s00542-018-4044-6.

3.2.4 Neuromorphic and Neuro-Inspired Computing

Neuromorphic and neuro-inspired approaches mimic the functioning of human brain (or our understanding of its functioning) to efficiently perform computations that are difficult or impractical for conventional computer architectures [1, 2].

Neuromorphic Computing (NMC), as developed by Carver Mead in the late 1980s, describes the use of large-scale adaptive analog systems to mimic organizational principles used by the nervous system. Originally, the main approach was to use elementary physical phenomena of integrated electronic devices (transistors, capacitors, …) as computational primitives [1]. In recent times, the term neuromorphic has also been used to describe analog, digital, and mixed-mode analog/digital hardware and software systems that transfer aspects of structure and function from biological substrates to electronic circuits (for perception, motor control, or multisensory integration). Today, the majority of NMC implementations is based on CMOS technology. Interesting alternatives are, for example, oxide-based memristors, spintronics, or nanotubes [3, 4, 5]. Such kind of research is still in its early stage.

The basic idea of NMC is to exploit the massive parallelism of such circuits and to create low-power and fault-tolerant information-processing systems. Aiming at overcoming the big challenges of deep-submicron CMOS technology (power wall, reliability, and design complexity), bio-inspiration offers alternative ways to (embedded) artificial intelligence. The challenge is to understand, design, build, and use new architectures for nanoelectronic systems, which unify the best of brain-inspired information processing concepts and of nanotechnology hardware, including both algorithms and architectures [6]. A key focus area in further scaling and improving of cognitive systems is decreasing the power density and power consumption, and overcoming the CPU/memory bottleneck of conventional computational architectures [7].

Current State of CMOS-Based Neuromorphic Approaches

Large scale neuromorphic chips exist based on CMOS technology, replacing or complementing conventional computer architectures by brain-inspired architectures. Mapping brain-like structures and processes into electronic substrates has recently seen a revival with the availability of deep-submicron CMOS technology.

Advances in technology have successively increased our ability to emulate artificial neural networks (ANNs) with speed and accuracy. At the same time, our understanding of neurons in the brain has increased substantially, with imaging and microprobes contributing significantly to our understanding of neural physiology. These advances in both technology and neuroscience stimulated international research projects with the ultimate goal to emulate entire (human) brains. Large programs on brain research through advanced neurotechnologies have been launched worldwide, e.g. the U.S. BRAIN initiative (launched in 2013 [8]), the EC flagship Human Brain Project (launched in 2013 [9]), the China Brain Project (launched in 2016 [10]), or the Japanese government initiated Brain/MINDS project (launched in 2016 [11]). Besides basic brain research these programs aim at developing electronic neuromorphic machine technology that scales to biological levels. More simply stated it is an attempt to build a new kind of computer with similar form and function to the mammalian brain. Such artificial brains would be used to build robots whose intelligence matches that of mice and cats. The ultimate aim is to build technical systems that match a mammalian brain in function, size, and power consumption. It should recreate 10 billion neurons, 100 trillion synapses, consume one kilowatt (same as a small electric heater), and occupy less than two litres of space [8].

The majority of larger more bio-realistic simulations of brain areas are still done on High Performance Supercomputer (HPS). For example, the Blue Brain Project [12] at EPFL in Switzerland deploys just from the beginning HPSs for digital reconstruction and simulations of the mammalian brain. The goal of the Blue Brain Project (EPFL and IBM, launched in 2005): "… is to build biologically detailed digital reconstructions and simulations of the rodent, and ultimately the human brain. The supercomputer-based reconstructions and simulations built by the project offer a radically new approach for understanding the multilevel structure and function of the brain." The project uses an IBM Blue Gene supercomputer (100 TFLOPS, 10TB) with currently 8,000 CPUs to simulate ANNs (at ion-channel level) in software [12]. The time needed to simulate brain areas is at least two orders of mag-

nitude larger than biological time scales. Based on a simpler (point) neuron model, the simulation could have delivered orders of magnitude higher performance. Dedicated brain simulation machines (Neurocomputer) based on application specific architectures offer faster emulations on such simpler neuron models.

Closely related to the Blue Brain Project is the Human Brain Project (HBP), a European Commission Future and Emerging Technologies Flagship [9]. The HBP aims to put in place a cutting-edge, ICT-based scientific research infrastructure that will allow scientific and industrial researchers to advance our knowledge in the fields of neuroscience, computing and brain-related medicine. The project promotes collaboration across the globe, and is committed to driving forward European industry. Within the HBP the subproject SP9 designs, implements and operate a Neuromorphic Computing Platform with configurable Neuromorphic Computing Systems (NCS). The platform provides NCS based on physical (analogue or mixed-signal) emulations of brain models, running in accelerated mode (NM-PM1, wafer-scale implementation of 384 chips with about 200.000 analog neurons on a wafer in 180nm CMOS, 20 wafer in the full system), numerical models running in real time on a digital multicore architecture (NM-MC1 with 18 ARM cores per chip in 130nm CMOS, 48 chips per board, and 1200 boards for the full system), and the software tools necessary to design, configure and measure the performance of these systems. The platform will be tightly integrated with the High Performance Analytics and Computing Platform, which will provide essential services for mapping and routing circuits to neuromorphic substrates, benchmarking and simulation-based verification of hardware specifications [9]. For both neuromorphic hardware systems new chip versions are under development within HBP. NM-PM2: wafer-scale integration based on a new mixed-signal chip in 65nm CMOS integrating a custom SIMD processor (32-bit, 128-bit wide vectors) for learning (6-bit SRAM synapse-circuits), an analog network core with better precision per neuron- (10 bit resolution), and an improved communication system [13]; NM-MC2: 144 ARM M4F cores per chip in 22nm CMOS technology with floating point support, 128 KByte local SRAM, and improved power management. Furthermore, the chip provides a dedicated pseudo random number generator, an exponential function accelerator and a Multiply-Accumulate (MAC) array (16x4 8Bit multiplier) with DMA for rate based ANN computation [14].

The number of neuromorphic systems is constantly increasing, but not as fast as hardware accelerators for non-spiking ANNs. Most of them are research prototypes (e.g. the trainable neuromorphic processor for fast pattern classification from the Seoul National University (Korea) [15], or the Tianjic chip from Beijing's Tsinghua University Center for Brain Inspired Computing Research [16]. Examples from industry are the TrueNorth chip from IBM [17] and the Loihi chip from INTEL [18]. The IBM TrueNorth chip integrates a two-dimensional on-chip network of 4096 digital application-specific digital cores (64 x 64) and over 400 Mio. bits of local on-chip memory to store individually programmable synapses. One million individually programmable neurons can be simulated time-multiplexed per chip. The chip with about 5.4 billion transistors is fabricated in a 28nm CMOS process (4.3 cm^2 die size, 240µm x 390 µm per core) and by device count the largest IBM chip ever fabricated. The INTEL self-learning neuromorphic Loihi chip integrates 2.07 billion transistors in a 60 mm^2 die fabricated in Intel's 14 nm CMOS FinFET process. The first iteration of the Loihi houses 128 clusters of 1,024 artificial neurons each for a total of 131,072 simulated neurons, up to 128 million (1-bit) synapses (16 MB), three Lakefield (Intel Quark) CPU cores, and an off-chip communication network. An asynchronous NoC manages the communication of packetized messages between clusters. Loihi is not a product, but available for research purposes among academic research groups organized in the INTEL Neuromorphic Research Community (INRC).

Artificial Neural Networks (ANNs)

All above mentioned projects have in common that they model spiking neurons, the basic information processing element in biological nervous systems. A more abstract implementation of biological neural systems are Artificial Neural Networks (ANNs). Popular representatives are Deep Neural Networks (DNNs) as they have propelled an evolution in the machine learning field. DNNs share some architectural features of the nervous systems, some of which are loosely inspired by biological vision systems [19]. DNNs are dominating computer vision today and observe a strong growing interest for solving all kinds of classification, function approximation, interpolation, or forecasting problems. Training DNNs is com-

putationally intense. For example, Baidu Research[5] estimated that training one DNN for speech recognition can require up to 20 Exaflops (10^{18} floating point operations per second); whereas Summit, the world's largest supercomputers in June 2019, deliver about 148 Petaflops. Increasing the available computational resources enables more accurate models as well as newer models for high-value problems such as autonomous driving and to experiment with more-advanced uses of artificial intelligence (AI) for digital transformation. Corporate investment in artificial intelligence will rapidly increase, becoming a $100 billion market by 2025 [20].

Hence, a variety of hardware and software solutions have emerged to slake the industry's thirst for performance. The currently most well-known commercial machines targeting deep learning are the TPUs of Google and the Nvidia Volta V100 and Turing GPUs. A tensor processing unit (TPU) is an ASIC developed by Google specifically for machine learning. The chip has been specifically designed for Google's TensorFlow framework. The first generation of TPUs applied 8-bit integer MAC (multiply accumulate) operations. It is deployed in data centres since 2015 to accelerate the inference phase of DNNs. An in-depth analysis was published by Jouppi et al. [21]. The second generation TPU of Google, announced in May 2017, are rated at 45 TFLOPS and arranged into 4-chip 180 TFLOPS modules. These modules are then assembled into 256 chip pods with 11.5 PFLOPS of performance [22]. The new TPUs are optimized for both training and making inferences.

Nvidia's Tesla V100 GPU contains 640 Tensor Cores delivering up to 120 Tensor TFLOPS for training and inference applications. Tensor Cores and their associated data paths are custom-designed to dramatically increase floating-point compute throughput with high energy efficiency. For deep learning inference, V100 Tensor Cores provide up to 6x higher peak TFLOPS compared to standard FP16 operations on Nvidia Pascal P100, which already features 16-bit FP operations [23].

Matrix-Matrix multiplication operations are at the core of DNN training and inferencing, and are used to multiply large matrices of input data and weights in the connected layers of the network. Each Tensor Core operates on a 4×4 matrix and performs the following operation: $D = A \times B + C$, where A, B, C, and D are 4×4 matrices. Tensor Cores operate on FP16 input data with FP32 accumulation. The FP16 multiply results in a full precision product that is then accumulated using FP32 addition with the other intermediate products for a $4 \times 4 \times 4$ matrix multiply [23]. The Nvidia DGX-1 system based on the Volta V100 GPUs was delivered in the third quarter of 2017 [24] as at that time the world's first purpose built system optimized for deep learning, with fully integrated hardware and software. Further Nvidia systems, currently DGX-2, emerge yearly.

Many more options for DNN hardware acceleration are showing up [25]. AMD's Vega GPU should offer 13 TFLOPS of single precision, 25 TFLOPS of half-precision performance, whereas the machine-learning accelerators in the GPU-based Tesla V100 can offer 15 TFLOPS single precision (FP32) and 120 Tensor TFLOPS (FP16) for deep learning workloads. Microsoft has been using Altera FPGAs for similar workloads, though a performance comparison is tricky; the company has performed demonstrations of more than 1 Exa-operations per second [26]. Intel offers the Xeon Phi 7200 family and IBMs TrueNorth tackles deep learning as well [27]. Other chip and IP (Intellectual Property) vendors—including Cadence, Ceva and Synopsys—are touting DSPs for learning algorithms. Although these hardware designs are better than CPUs, none was originally developed for DNNs. Ceva's new XM6 DSP core[6] enables deep learning in embedded computer vision (CV) processors. The synthesizable intellectual property (IP) targets self-driving cars, augmented and virtual reality, surveillance cameras, drones, and robotics. The normalization, pooling, and other layers that constitute a convolutional-neural-network model run on the XM6's 512-bit vector processing units (VPUs). The new design increases the number of VPUs from two to three, all of which share 128 single-cycle (16×16)-bit MACs, bringing the XM6's total MAC count to 640. The core also includes four 32-bit scalar processing units.

Examples for start-ups are Nervana Systems[7], Knupath[8], Wave Computing[9], and Cerebas[10]. The Nervana Engine will combine a custom 28 nm chip with 32 GB of high bandwidth memory and replacing caches with software-managed memory. Kupath second generation DSP Hermosa is positioned for deep learning

[5] www.baidu.com

[6] www.ceva-dsp.com
[7] www.nervanasys.com
[8] www.knupath.com
[9] www.wavecomp.com
[10] www.graphcore.ai

as well as signal processing. The 32 nm chip contains 256 tiny DSP cores operation at 1 GHz along with 64 DMA engines and burns 34 W. The dataflow processing unit from Wave Computing implements "tens of thousands" of processing nodes and "massive amounts" of memory bandwidth to support TensorFlow and similar machine-learning frameworks. The design uses self-timed logic that reaches speeds of up to 10 GHz. The 16 nm chip contains 16 thousand independent processing elements that generate a total of 180 Tera 8-bit integer operations per second. The Graphcore wafer-scale approach from Cerebas is another start-up example at the extreme end of the large spectrum of approaches. The company claim to have built the largest chip ever with 1.2 trillion transistors on a 46,225 mm^2 silicon (TSMC 16nm process). It contains 400,000 optimized cores, 18 GB on-chip memory, and 9PetaByte/s memory bandwidth. The programmable cores with local memory are optimized for machine learning primitives and connected with high-bandwidth and low latency connections [28].

Impact on Hardware for Neuromorphic and Neuro-Inspired Computing

Creating the architectural design for NMC requires an integrative, interdisciplinary approach between computer scientists, engineers, physicists, and materials scientists. NMC would be efficient in energy and space and applicable as embedded hardware accelerator in mobile systems. The building blocks for ICs and for the Brain are the same at nanoscale level: electrons, atoms, and molecules, but their evolutions have been radically different. The fact that reliability, low-power, reconfigurability, as well as asynchronicity have been brought up so many times in recent conferences and articles, makes it compelling that the Brain should be an inspiration at many different levels, suggesting that future nano-architectures could be neural-inspired. The fascination associated with an electronic replication of the human brain has grown with the persistent exponential progress of chip technology. The decade 2010–2020 has also made the electronic implementation more feasible, because electronic circuits now perform synaptic operations such as multiplication and signal communication at energy levels of 10 fJ, comparable to biological synapses. Nevertheless, an all-out assembly of 10^{14} synapses will remain a matter of a few exploratory systems for the next two decades because of several challenges [6].

Neuromorphic hardware development is progressing fast with a steady stream of new architectures coming up. Because network models and learning algorithms are still developing, there is little agreement on what a learning chip should actually look like. The companies withheld details on the internal architecture of their learning accelerators. Most of the designs appear to focus on high throughput for low-precision data, backed by high bandwidth memory subsystems. The effect of low-precision on the learning result has not been analysed in detail yet. Recent work on low-precision implementations of backprop-based neural nets [29] suggests that between 8 and 16 bits of precision can suffice for using or training DNNs with backpropagation. What is clear is that more precision is required during training than at inference time, and that some forms of dynamic fixed point representation of numbers can be used to reduce how many bits are required per number. Using fixed point rather than floating point representations and using less bits per number reduces the hardware surface area, power requirements, and computing time needed for performing multiplications, and multiplications are the most demanding of the operations needed to use or train a modern deep network with backpropagation. A first standardization effort is the specification of the Brain Floating Point (BFLOAT16) half-precision data format for DNN learning [30]. Its dynamic range is the same as that of FP32, conversion between both straightforward, and training results are almost the same as with FP32. Industry-wide adoption of BFLOAT is expected.

Memristors in Neuromorphic and Neuro-Inspired Computing

In the long run also the memristor technology is heavily discussed in literature for future neuromorphic computing. The idea, e.g. in so-called spike-time-dependent plasticity (STDP) networks [31, 32], is to mimic directly the functional behaviour of a neuron. In STDP networks the strength of a link to a cell is determined by the time correlation of incoming signals to a neuron along that link and the output spikes. The shorter the input pulses are compared to the output spike, the stronger the input links to the neuron are weighted. In contrast, the longer the input signals lay behind the output spike, the weaker the link is adjusted. This process of strengthening or weakening the weight shall be directly mapped onto memristors

by increasing or decreasing their resistance depending which voltage polarity is applied to the poles of a two-terminal memristive device. This direct mapping of an STDN network to an analogue equivalent of the biological cells to artificial memristor-based neuron cells shall emerge new extreme low-energy neuromorphic circuits. Besides this memristor-based STDP networks there are lots of proposals for neural networks to be realised with memristor-based crossbar and mesh architectures for cognitive detection and vision applications, e.g. [33].

One extremely useful property of memristors in the context of neuromorphic is their biorealism, i.e., the ability to mimic behavior of elements found in human brain [34] and vision system [35]. Some of the early neuromorphic systems used capacitors to represent weights in the analog domain [1], and memristance can assume its role [34]. Well-known learning concepts, including spike-timing-dependent plasticity (STDP), can be mapped to memristive components in a natural way [36]. A recent example of a biorealistic hardware model is [37], which reports the manufacturing of a larger-scale network of artificial memristive neurons and synapses capable of learning. The memristive functionality is achieved by precisely controlling silver nanoparticles in a dielectric film such that their electrical properties closely matches ion channels in a biological neuron.

Biorealistic models are not the only application of memristors in neuromorphic or neuro-inspired architectures. Such architectures realize neural networks (NNs) with a vast amount of weights, which are determined, or learned, during the training phase, and then used without modification for an extended period of time, during the inference phase. After some time, when the relevant conditions have changed, it may become necessary to re-train the NN and replace its weights by new values. Memristive NVMs are an attractive, lightweight and low-power option for storing these weights. The circuit, once trained, can be activated and deactivated flexibly while retaining its learned knowledge. A number of neuromorphic accelerators based on memristive NVMs have been proposed in the last few years. For example, IBM developed a neuromorphic core with a 64-K-PCM-cell as "synaptic array" with 256 axons × 256 dendrite to implement spiking neural networks [38].

Perspectives on Neuromorphic and Neuro-Inspired Computing

Brain-inspired hardware computing architectures have the potential to perform AI tasks better than conventional architecture by means of better performance, lower energy consumption, and higher resilience to defects. Neuromorphic Computing and Deep Neural Networks represent two approaches for taking inspiration from biological brains. Software implementations on HPC-clusters, multi-cores (OpenCV), and GPGPUs (NVidia cuDNN) are already commercially used. FPGA acceleration of neural networks is available as well. From a short term perspective these software implemented ANNs may be accelerated by commercial transistor-based neuromorphic chips or accelerators. Future emerging hardware technologies, like memcomputing and 3D stacking [39] may bring neuromorphic computing to a new level and overcome some of the restriction of Von-Neumann-based systems in terms of scalability, power consumption, or performance.

Particularly attractive is the application of ANNs in those domains where, at present, humans outperform any currently available high-performance computer, e.g., in areas like vision, auditory perception, or sensory motor control. Neural information processing is expected to have a wide applicability in areas that require a high degree of flexibility and the ability to operate in uncertain environments where information usually is partial, fuzzy, or even contradictory. This technology is not only offering potential for large scale neuroscience applications, but also for embedded ones: robotics, automotive, smartphones, IoT, surveillance, and other areas [6]. Neuromorphic computing appears as key technology on several emerging technology lists. Hence, Neuromorphic technology developments are considered as a powerful solution for future advanced computing systems [40]. Neuromorphic technology is in early stages, despite quite a number of applications appearing.

To gain leadership in this domain there are still many important open questions that need urgent investigation (e.g. scalable resource-efficient implementations, online learning, and interpretability). There is a need to continue to mature the NMC system and at the same time to demonstrate the usefulness of the systems in applications, for industry and also for the society: more usability and demonstrated applications.

More focus on technology access might be needed in

Europe. Regarding difficulties for NMC in EC framework programmes, integrated projects were well fitting the needs of NMC in FP7, but are missing in H2020. For further research on neuromorphic technology the FET-OPEN scheme could be a good path as it requires several disciplines (computer scientists, material science, engineers in addition to neuroscience, modelling). One also needs support for many small-scale interdisciplinary exploratory projects to take advantage of newly coming out developments, and allow funding new generation developers having new ideas.

References

[1] C. Mead. "Neuromorphic Electronic Systems". In: *Proceedings of the IEEE* 78.10 (Oct. 1990), pp. 1629–1636. DOI: 10.1109/5.58356.

[2] W. Wen, C. Wu, X. Hu, B. Liu, T. Ho, X. Li, and Y. Chen. "An EDA framework for large scale hybrid neuromorphic computing systems". In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. June 2015, pp. 1–6.

[3] Y. V. Pershin and M. D. Ventra. "Neuromorphic, Digital, and Quantum Computation With Memory Circuit Elements". In: *Proceedings of the IEEE* 100.6 (June 2012), pp. 2071–2080. DOI: 10.1109/JPROC.2011.2166369.

[4] M. D. Pickett, G. Medeiros-Ribeiro, and R. S. Williams. "A scalable neuristor built with Mott memristors". In: *Nature Materials* 12 (Feb. 2013), pp. 114–117. DOI: 10.1038/nmat3510.

[5] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu. "Nanoscale Memristor Device as Synapse in Neuromorphic Systems". In: *Nano Letters* 10.4 (2010), pp. 1297–1301. DOI: 10.1021/nl904092h.

[6] U. Rueckert. "Brain-Inspired Architectures for Nanoelectronics". In: *CHIPS 2020 VOL. 2: New Vistas in Nanoelectronics*. 2016, pp. 249–274. DOI: 10.1007/978-3-319-22093-2_18.

[7] E. Eleftheriou. "Future Non-Volatile Memories: Technology, Trends, and Applications". 2015.

[8] *U.S. Brain Initiative*. 2019. URL: http://braininitiative.nih.gov.

[9] *Human Brain Project*. 2017. URL: http://www.humanbrainproject.eu.

[10] M. Poo et al. "China Brain Project: Basic Neuroscience, Brain Diseases, and Brain-Inspired Computing". In: *Neuron* 92 (Nov. 2016), pp. 591–596.

[11] *Japanese Brain/MIND Project*. 2019. URL: http://brainminds.jp/en/.

[12] *The Blue Brain Project - A Swiss Brain Initiative*. 2017. URL: http://bluebrain.epfl.ch/page-56882-en.html.

[13] S. Aamir et al. "An Accelerated LIF Neuronal Network Array for a Large Scale Mixed-Signa-Neuromorphic Architecture". In: *arXiv:1804.01906v3* (2018).

[14] Y. Yan et al. "Efficient Reward-Based Structural Plasticity on a SpiNNaker 2 Prototype". In: *IEEE Trans. on Biomedical Circuits and Systems* 13.3 (2019), pp. 579–591.

[15] J. Park et al. "A 65nm 236.5nJ/Classification Neuromorphic Processor with 7.5% Energy Overhead On-Chip Learning Using Direct Spike-Only Feedback". In: *IEEE Int. Solid-State Circuits Conference*. 2019, pp. 140–141.

[16] J. Pei et al. "Towards artificial general intelligence with hybrid Tianjic chip architecture". In: *Nature* 572 (2019), p. 106.

[17] P. A. Merolla et al. "A million spiking-neuron integrated circuit with a scalable communication network and interface". In: *Science* 345.6197 (2014), pp. 668–673. DOI: 10.1126/science.1254642.

[18] M. Davies et al. "A Neuromorphic Manycore Processor with On-Chip Learning". In: *IEEE Micro* 1 (2018), pp. 82–99.

[19] Y. LeCun and Y. Bengio. In: *The Handbook of Brain Theory and Neural Networks*. 1998. Chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258.

[20] D. Wellers, T. Elliott, and M. Noga. *8 Ways Machine Learning Is Improving Companies' Work Processes*. Harvard Business Review. 2017. URL: https://hbr.org/2017/05/8-ways-machine-learning-is-improving-companies-work-processes.

[21] N. P. Jouppi et al. "In-Datacenter Performance Analysis of a Tensor Processing Unit". In: *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)*. ISCA '17. 2017, pp. 1–12. DOI: 10.1145/3079856.3080246.

[22] Wikipedia. *Tensor Processing Unit*. 2017. URL: https://en.wikipedia.org/wiki/Tensor_processing_unit.

[23] Nvidia Corporation. *Nvidia Tesla V100 GPU Architecture*. Version WP-08608-001_v01. 2017. URL: https://images.nvidia.com/content/volta-architecture/pdf/Volta-Architecture-Whitepaper-v1.0.pdf.

[24] T. P. Morgan. *Big Bang for the Buck Jump with Volta DGX-1*. 2017. URL: https://www.nextplatform.com/2017/05/19/big-bang-buck-jump-new-dgx-1/.

[25] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam. "DianNao: A Small-footprint High-throughput Accelerator for Ubiquitous Machine-learning". In: *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. ASPLOS '14. 2014, pp. 269–284. DOI: 10.1145/2541940.2541967.

[26] P. Bright. *Google brings 45 Teraflops Tensor Flow Processors to its Compute Cloud*. 2017. URL: https://arstechnica.com/information-technology/2017/05/google-brings-45-teraflops-tensor-flow-processors-to-its-compute-cloud/.

[27] L. Gwennap. "Learning Chips Hit The Market: New Architectures Designed for Neural Processing". In: *Microprocessor Report*. MPR 6/27/16 (2016).

[28] M. Demler. "CEREBRAS BREAKS THE RETICLE BARRIER: Wafer-Scale Engineering Enables Integration of 1.2 Trillion Transistors". In: *MPR* (2019).

[29] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan. "Deep Learning with Limited Numerical Precision". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*. ICML'15. 2015, pp. 1737–1746.

[30] D. Kalamkar et al. "A Study of BFLOAT16 for Deep Learning Training". In: *arXiv:1905.12322v3* (2019).

[31] G. S. Snider. "Spike-timing-dependent learning in memristive nanodevices". In: *International Symposium on Nanoscale Architectures*. 2008, pp. 85–92. DOI: 10.1109/NANOARCH.2008.4585796.

[32] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco. "STDP and STDP variations with memristors for spiking neuromorphic learning systems". In: *Frontiers in Neuroscience* 7 (2013), p. 15. DOI: 10.3389/fnins.2013.00002.

[33] C. K. K. Lim, A. Gelencser, and T. Prodromakis. "Computing Image and Motion with 3-D Memristive Grids". In: *Memristor Networks*. 2014, pp. 553–583. DOI: 10.1007/978-3-319-02630-5_25.

[34] I. E. Ebong and P. Mazumder. "CMOS and Memristor-Based Neural Network Design for Position Detection". In: *Proceedings of the IEEE* 100.6 (2012), pp. 2050–2060.

[35] C. K. K. Lim, A. Gelencser, and T. Prodromakis. "Computing Image and Motion with 3-D Memristive Grids". In: *Memristor Networks*. 2014, pp. 553–583. DOI: 10.1007/978-3-319-02630-5_25.

[36] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco. "STDP and STDP variations with memristors for spiking neuromorphic learning systems". In: *Frontiers in Neuroscience* 7 (2013).

[37] X. Wang, S. Joshi, S. Savelév, et al. "Fully memristive neural networks for pattern classification with unsupervised learning". In: *Nature Electronics* 1.2 (2018), pp. 137–145.

[38] G. Hilson. *IBM Tackles Phase-Change Memory Drift, Resistance*. EETimes. 2015. URL: http://www.eetimes.com/document.asp?doc_id=1326477.

[39] B. Belhadj, A. Valentian, P. Vivet, M. Duranton, L. He, and O. Temam. "The improbable but highly appropriate marriage of 3D stacking and neuromorphic accelerators". In: *International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*. 2014, pp. 1–9. DOI: 10.1145/2656106.2656130.

[40] FET. *Workshop on the Exploitation of Neuromorphic Computing Technologies*. 2017. URL: https://ec.europa.eu/digital-single-market/en/news/workshop-exploitation-neuromorphic-computing-technologies.

3.3 Applying Memristor Technology in Reconfigurable Hardware

Reconfigurable computing combines the advantages of programmability of software with the performance of hardware. Industry and research exploit this ability for fast prototyping of hardware, update hardware in the field or to reduce costs in environments where a company only requires a small volume of chips. Even in High Performance Computing (HPC), reconfigurable hardware plays an important role by accelerating time consuming functions. Reconfigurable hardware is well-integrated in modern computational environments, like in System on Chips (SoCs) or additional accelerator cards. The most common chip types used for reconfigurable hardware are Field Programmable Gate Arrays (FPGAs). Their importance has increased in the last years because FPGA vendors like Xilinx and Intel switched to much smaller chip fabrication processes and could double the size of the available reconfigurable hardware per chip.

At the moment reconfigurable hardware is produced in a standard CMOS fabrication process. Configuration memory, Block-RAM, and Look-Up-Tables (LUTs) are implemented using Static-Random-Access-Memory (SRAM) cells or flash-based memory. Crossbar switches consisting of multiple transistors provide routing and communication infrastructure.

CMOS compatibility and a small area and power footprint are the important features of memristor technology for reconfigurable hardware. At the moment the main challenges for reconfigurable hardware are a high static power consumption and long interconnection delays. Memristor technology, applied to important building blocks of reconfigurable hardware, can help overcoming these challenges.

The following subsections describe the impact of memristor technology to key parts of reconfigurable hardware.

Memristors in Block RAM

Block RAM is the most obvious part of reconfigurable hardware for the deployment of memristor technology. Current *Block RAM* is SRAM based and one SRAM cell consists of six CMOS transistors.

The 1T1R[11] memristor technique introduced by Tanachutiwat et al. [1] reduces the number of transistors required for one memory cell to one. Memristor based memory cells require a small encode/decode hardware, but this technique still has an area density enhancement of six times to the SRAM based approach. The memristor based cells only require power if their content changes, reducing the static power consumption of reconfigurable hardware. Because of the density improvements even more *Block RAM* can be deployed on the reconfigurable hardware than currently available. Another important improvement using memristor technology is its non-volatile feature. Even if the whole reconfigurable hardware looses power, the content of the *Block RAM* is still available after power restoration.

Memristors in Configurable Logic Blocks (CLBs)

The CLBs are another important building block of reconfigurable hardware because they implement the different hardware functions. In general this is achieved by using/ combining LUTs and/or multiplexers. Like *Block RAM*, LUTs are, at the moment, based on SRAM cells. The 1TR1 approach of Section 3.3 is also a simple approach to improve area density and power consumption within LUTs (see for example, Kumar[2]). The non-volatile feature of memristors would improve configuration management of reconfigurable hardware because the configuration of the hardware does not need to be reloaded after a power loss.

Memristors in the Interconnection Network

The interconnection network of reconfigurable hardware is responsible for 50%-90% of the total reconfigurable hardware area usage, 70%-80% of the total signal delay and 60%-85% of the total power consumption[3]. Improving the interconnection network will have a huge impact on the overall reconfigurable hardware performance. Routing resources of the interconnection network are implemented using seven CMOS transistors at the moment. Six transistors for a SRAM cell and one transistor for controlling the path.

Tanachutiwat etal. [1] extend their 1TR1 approach for *Block RAM* cells to a 2T1R and 2T2R technique for routing switches. The second is fully compatible to the current implementation because one transistor

[11] 1 Transistor Element and 1 Resistive Element

controls the path, while in the 2T1R technique a memristor does. The 2T1R and 2T2R approach is also used by Hasan etal. [4] to build complex crossbar switches. A complex routing switch built out of many 2T1R or 2R2R elements can save even more transistors by combining different programming transistors.

The memristor based improvements for the interconnection network reduce the standby power of reconfigurable hardware considerably. They also reduce the area requirements for the interconnection network, allowing a more dense placement of the logic blocks and, therefore, improving the overall signal delay. Like in the previous sections, the non-volatile nature of the memristors prevents configuration loss on power disconnect.

Conclusion and Research Perspective

Memristor technology will have a high impact on reconfigurable hardware development and research. This Section presented improvements through memristor technology on important building blocks of reconfigurable hardware. These improvements target power consumption and area reduction, both important challenges of modern reconfigurable hardware development.

At the moment, the non-volatile nature of the memristor technology is not the focus of research. But this aspect can be a game changer for certain application areas and even open up new application areas for reconfigurable computing. For example, a reconfigurable hardware system would not require any external configuration memory and the initialization time of a system could be reduced multiple times. Deep sleep states are easily implemented, reducing the power consumption even more. These improvements are important for application areas like building automation, wearables and safety critical applications.

Further research areas include the evaluation of memristor technology in the logic building block of reconfigurable hardware, more research in the optimization of routing and interconnection resources with memristors, and the evaluation of the non-volatile aspects of memristors for reconfigurable hardware applications.

References

[1] S. Tanachutiwat, M. Liu, and W. Wang. "FPGA Based on Integration of CMOS and RRAM". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 19.11 (2011), pp. 2023–2032.

[2] T. Nandha Kumar. "An Overview on Memristor-Based Nonvolatile LUT of an FPGA". In: *Frontiers in Electronic Technologies: Trends and Challenges*. Singapore, 2017, pp. 117–132. DOI: 10.1007/978-981-10-4235-5_8. URL: https://doi.org/10.1007/978-981-10-4235-5_8.

[3] J. Cong and B. Xiao. "FPGA-RPI: A Novel FPGA Architecture With RRAM-Based Programmable Interconnects". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22.4 (Apr. 2014), pp. 864–877. DOI: 10.1109/TVLSI.2013.2259512.

[4] R. Hasan and T. M. Taha. "Memristor Crossbar Based Programmable Interconnects". In: *2014 IEEE Computer Society Annual Symposium on VLSI*. July 2014, pp. 94–99. DOI: 10.1109/ISVLSI.2014.100.

3.4 Non-Silicon-based Technology

3.4.1 Photonics

The general idea of using photonics in computing systems is to replace electrons with photons in intra-chip, inter-chip, processor-to-memory connections and maybe even logic.

Introduction to Photonics and Integrated Photonics

An optical transmission link is composed by some key modules: laser light source, a modulator that converts electronic signals into optical ones, waveguides and other passive modules (e.g. couplers, photonic switching elements, splitters) along the link, a possible drop filter to steer light towards the destination and a photodetector to revert the signal into the electronic domain. The term *integrated photonics* refers to a photonic interconnection where at least some of the involved modules are integrated into silicon [1]. Also directly modulated integrated laser sources have been developed and are improving at a steady pace [2, 3, 4, 5]. *Active* components (lasers, modulators and photodetectors) cannot be trivially implemented in CMOS process as they require the presence of materials (e.g., III-V semiconductors) different from silicon and, typically, not naturally compatible with it in the production process. However, great improvements have been done in the last years on this subject.

Optical communication nowadays features about 10-50 GHz modulation frequency and can support wavelength-division-multiplexing (WDM) up to 100+ *colors* in fiber and 10+ (and more are expected in near future) in silicon. Modulations are investigated too, as to boost bandwidth per laser color in the 100s GBps zone, like in [6]. Propagation loss is relatively small in silicon and polymer materials so that optical communication can be regarded as substantially insensitive to chip- and board-level distances. Where fiber can be employed (e.g. rack- and data centre levels) attenuation is no problem. Optical communication can rely on extremely fast signal propagation speed (head-flit latency): around 15 ps/mm in silicon and about 5.2 ps/mm in polymer waveguides that is traversing a 2 cm x 2 cm chip corner-to-corner in 0.6 and 0.2 ns, respectively. However, conversions to/from the optical domain cost energy and can erode some of this intrinsic low-latency, as it is the case for network-level protocols and shared resource management.

Manufacturing of *passive* optical modules (e.g. waveguides, splitters, crossings, microrings) is relatively compatible with CMOS process and the typical cross-section of a waveguide (about 500 nm) is not critical, unless for the smoothness of the waveguide walls as to keep light scattering small. Turns with curvature of a few µm and exposing limited insertion loss are possible, as well as grating couplers to introduce/emit light from/into a fiber outside of the chip. Even various 5x5 optical switches [7] can be manufactured out of basic photonic switching elements relying on tunable micro-ring resonators. Combining these optical modules, various optical interconnection topologies and schemes can be devised: from all-to-all contentionless networks up to arbitrated ones which share optical resources among different possible paths.

In practice, WDM requires precision in microring manufacturing, runtime tuning (e.g. thermal), alignment (multiple microrings with the same resonant frequency) and make more complex both the management of multi-wavelength light from generation, distribution, modulation, steering up to photo-detection. The more complex a topology, the more modules can be found along the possible paths between source and destination, on- and off-chip, and more laser power is needed to compensate their attenuation and meet the sensitivity of the detector. For these reasons, relatively simple topologies can be preferable as to limit power consumption and, spatial division multiplexing (using multiple parallel waveguides) can allow to trade WDM for space occupation.

Optical inter-chip signals are then expected to be conveyed also on different mediums to facilitate integrability with CMOS process, e.g., polycarbonate as in some IBM research prototypes and commercial solutions.

Current Status and Roadmaps

Currently, optical communication is mainly used in HPC systems in the form of *optical cables* which have progressively substituted shorter and shorter electronic links. From 10+ meters inter-rack communication down to 1+ meter intra-rack and sub meter intra-blade links.

A number of industrial and research roadmaps are projecting and expecting this trend to arrive within boards and then to have optical technology that crosses the chip boundary, connects chips within

Table 3.2: Expected evolution of optical interconnection [8].

Time Frame	~2000	~2005	~2010	~2015	~2020	~2025
Interconnect	Rack	Chassis	Backplane	Board	Module	Chip
Reach	20 – 100 m	2 – 4 m	1 – 2 m	0.1 – 1 m	1 – 10 cm	0.1 – 3 cm
Bandw. (Gb/s, Tb/s)	40 – 200 G	20 – 100 G	100 – 400 G	0.3 – 1 T	1 – 4 T	2 – 20 T
Bandw. Density ($GB/s/cm^2$)	~100	~100 – 400	~400	~1250	> 10000	> 40000
Energy (pJ/bit)	1000 → 200	400 → 50	100 → 25	25 → 5	1 → 0.1	0.1 → 0.01

silicon- and then in optical-interposers and eventually arriving to a complete integration of optics on a different layer of traditional chips. For this reason, also the evolution of 2.5 - 3D stacking technologies is expected to enable and sustain this roadmap up to seamless integration of optical layers along with logic ones, and the dawn of disaggregated architectures enabled by the low-latency features of optics [9]. The expected rated performance/consumption/density metrics are shown in the 2016 Integrated Photonic Systems Roadmap [8] (see Table 3.2).

IBM, HPM, Intel, STM, CEA–LETI, Imec and Petra, to cite a few, essentially share a similar view on this roadmap and on the steps to increase bandwidth density, power consumption and cost effectiveness of the interconnections needed in the Exascale, and post-Exascale HPC systems. For instance, Petra labs demonstrated the first optical silicon interposer prototype [10] in 2013 featuring 30 $TB/s/cm^2$ bandwidth density and in 2016 they improved consumption and high-temperature operation of the optical modules [11]. HP has announced the Machine system which relies on the optical X1 photonic module capable of 1.5 Tbps over 50m and 0.25 Tbps over 50km. Intel has announced the Omni-Path Interconnect Architecture that will provide a migration path between Cu and Fiber for future HPC/Data Centre interconnections. Optical thunderbolt and optical PCI Express by Intel are other examples of optical cable solutions. IBM is shipping polymer + micro-pod optical interconnection within HPC blades since 2012 and it is moving towards module-to-module integration.

The main indications from current roadmaps and trends can be summarized as follows. Optical-cables (AOC - Active Optical Cables) are evolving in capability (bandwidth, integration and consumption) and are getting closer to the chips, leveraging more and more photonics in an integrated form. Packaging problem of photonics remains a major issue, especially where optical signals need to traverse the chip package. Also for these reasons, interposers (silicon and optical) appear to be the reasonable first steps towards optically integrated chips. Then, full 3D processing and hybrid material integration are expected from the process point of view.

Figure 3.6 underlines the expected adoption roadmap of the different levels of adoption of optical technologies, published in the 2017 Integrated Photonic Systems Roadmap. In particular, from the interconnect, packaging and photonic integration standpoints. The expected evolution of laser sources over time is confirmed as well as interposer-based solutions will pave the way to full integrated ones.

Conversion from photons to electrons is costly and for this reason there are currently strong efforts in improving the crucial physical modules of an integrated optical channel (e.g. modulators, photodetectors and thermally stable and efficiently integrated laser sources).

Alternate and Emerging Technologies Around Photonics

Photonics is in considerable evolution, driven by innovations in existing components (e.g. lasers, modulators and photodetectors) in order to push their features and applicability (e.g. high-temperature lasers). Consequently, its expected potential is a moving target based on the progress in the rated features of the various modules. At the same time, some additional variations, techniques and approaches at the physical

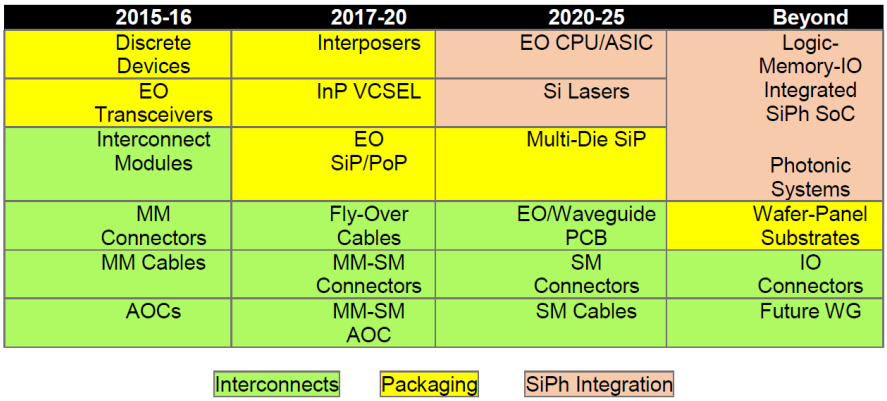


Figure 3.6: Integrated Photonic Systems Roadmap, 2017. Adoption expectations of the different optical technologies.

level of the photonic domain are being investigated and could potentially create further discontinuities and opportunities in the adoption of photonics in computing systems. For instance, we cite here a few:

- Mode division multiplexing [12]: where light propagates within a group of waveguides in parallel. This poses some criticalities but could allow to scale parallelism more easily than WDM and/or be an orthogonal source of optical bandwidth;
- Free-air propagation: there are proposals to exploit light propagation within the chip package without waveguides to efficiently support some interesting communication pattern (e.g. fast signaling) [13];
- Plasmonics: interconnect utilize surface plasmon polaritons (SPPs) for faster communication than photonics and far lower consumption over relatively short distances at the moment (below 1mm) [14, 15];
- Optical domain buffering: recent results [16] indicate the possibility to temporarily store light and delay its transmission. This could enable the evolution of additional network topologies and schemes, otherwise impossible, for instance avoiding the reconversion to the electronic domain;
- Photonic non-volatile memory [17]. This could reduce latencies of memory accesses by eliminating costly optoelectronic conversions while dramatically reducing the differences in speed between CPU and main memory in fully optical chips.
- Optics computing: Optalysys project[12] for computing in the optical domain mapping information onto light properties and elaborating the latter directly in optics in an extremely energy efficient way compared to traditional computers [18]. This approach cannot suit every application but a number of algorithms, like linear and convolution-like computations (e.g. FFT, derivatives and correlation pattern matching), are naturally compatible [19]. Furthermore, also bioinformatics sequence alignment algorithms have been recently demonstrated feasible. Optalysis has recently announced a commercial processor programmable either through a specific API or via TensorFlow interface to implement convolutional neural networks (CNN) [20].

Optical Communication Close to the Cores and Perspectives

As we highlighted, the current trend is to have optics closer and closer to the cores, from board-to-board, to chip-to-chip and up to within chips. The more optical links get close to the cores, the more the managed traffic becomes processor-specific. Patterns due to the micro-architectural behaviour of the processing cores become visible and crucial to manage, along with cache-coherence and memory consistency effects. This kind of traffic poses specific requirements to the interconnection sub-system which can be quite different from the ones induced by traffic at a larger scale. In fact, at rack or inter-rack level, the aggregate, more application-driven, traffic tends to smooth out

[12] www.optalysys.com

individual core needs so that "average" behaviours emerge.

For instance, inter-socket or intra-processor coherence and synchronizations have been designed and tuned in decades for the electronic technology and, maybe, need to be optimized, or re-though, to take the maximum advantage from the emerging photonic technology.

Research groups and companies are progressing towards inter-chip interposer solutions and completely optical chips. In this direction researchers have already identified the *crucial importance of a vertical cross-layer design* of a computer system endowed with integrated photonics. A number of studies have already proposed various kinds of on-chip and inter-chip optical networks designed around the specific traffic patterns of the cores and processing chips [21, 22, 23, 24, 25, 26, 27].

These studies suggest also that further challenges will arise from inter-layer design interference, i.e. lower-layer design choices (e.g. WDM, physical topology, access strategies, sharing of resources) can have a significant impact in higher layers of the design (e.g. NoC-wise and up to memory coherence and programming model implications) and vice versa. This is mainly due to the scarce experience in using photonics technology for serving computing needs (close to processing cores requirements) and, most of all, due to the intrinsic end-to-end nature of an efficient optical channel, which is conceptually opposed to the well-established and mature know-how of "store-and-forward" electronic communication paradigm. Furthermore, the quick evolution of optical modules and the arrival of discontinuities in their development hamper the consolidation of layered design practices.

Lastly, intrinsic low-latency properties of optical interconnection (on-chip and inter-chip) could imply a re-definition of what is local in a future computing system, at various scales, and specifically in a perspective HPC system, as it has already partially happened within the HP *Machine*. These revised locality features will then require modifications in the programming paradigms as to enable them to take advantage of the different organization of future HPC machines. On this point, also resource disaggregation is regarded as another dimension that could be soon added to the design of future systems and, in particular, HPC systems [9, 15]. Then, from another perspective, if other emerging technologies (e.g. NVM, in-memory computation, approximate, quantum computing, etc.) will appear in future HPC designs as it is expected to be in order to meet performance/watt objectives, it is highly likely that for the reasons above, photonic interconnections will require to be co-designed in integration also with the whole heterogeneous HPC architecture.

Funding Opportunities

Photonic technology at the physical and module level has been quite well funded in H2020 program [28] as it has been regarded as strategic by the EU since years. For instance Photonics21 [29] initiative gather groups and researchers from a number of enabling disciplines for the wider adoption of photonics in general and specifically also integrated photonics. Very recently, September 2019, Photonics21 has announced a request to EU for a doubling of the budget from 100 million€ to 200 million€per year, or 1.4 billion€ over the course of the next research funding initiative. Typically, funding instruments and calls focus on basic technologies and specific modules and in some cases towards point-to-point links as a final objective (e.g. optical cables).

Conversely, as photonics is coming close to the processing cores, which expose quite different traffic behaviour and communication requirements compared to larger scale interconnections (e.g. inter-rack or wide-area), it is highly advisable to promote also a separate funding program for investigating the specific issues and solutions for the adoption of integrated photonics at the inter-chip and intra-chip scale in order to expose photonic technologies with the constraints coming from the actual traffic generated by the processing cores and other on-chip architectural modules. In fact, the market is getting close to the cores *from the outside* with an *optical cable* model that will be less and less suitable to serve the traffic as the communication distance decreases. Therefore, now could be just the right time to invest into chip-to-chip and intra-chip optical network research in order to be prepared to apply it effectively when current roadmaps expect optics to arrive there.

References

[1] IBM. *Silicon Integrated Nanophotonics*. 2017. URL: http://researcher.watson.ibm.com/researcher/view%5C_group.php?id=2757.

[2] Y. Kurosaka, K. Hirose, T. Sugiyama, Y. Takiguchi, and Y. Nomoto. "Phase-modulating lasers toward on-chip integration". In: *Nature - scientific reports* 26.30138 (July 2016). DOI: 10.1038/srep30138.

[3] H. Nishi et al. "Monolithic Integration of an 8-channel Directly Modulated Membrane-laser Array and a SiN AWG Filter on Si". In: *Optical Fiber Communication Conference*. 2018, Th3B.2. DOI: 10.1364/OFC.2018.Th3B.2. URL: http://www.osapublishing.org/abstract.cfm?URI=OFC-2018-Th3B.2.

[4] Z. Li, D. Lu, Y. He, F. Meng, X. Zhou, and J. Pan. "InP-based directly modulated monolithic integrated few-mode transmitter". In: *Photon. Res.* 6.5 (May 2018), pp. 463–467. DOI: 10.1364/PRJ.6.000463. URL: http://www.osapublishing.org/prj/abstract.cfm?URI=prj-6-5-463.

[5] S. Matsuo and K. Takeda. "λ-Scale Embedded Active Region Photonic Crystal (LEAP) Lasers for Optical Interconnects". In: *Photonics* 6 (July 2019), p. 82. DOI: 10.3390/photonics6030082.

[6] C. Prodaniuc, N. Stojanovic, C. Xie, Z. Liang, J. Wei, and R. Llorente. "3-Dimensional PAM-8 modulation for 200 Gbps/lambda optical systems". In: *Optics Communications* 435 (2019), pp. 1–4. DOI: https://doi.org/10.1016/j.optcom.2018.10.046. URL: http://www.sciencedirect.com/science/article/pii/S0030401818309210.

[7] H. Gu, K. H. Mo, J. Xu, and W. Zhang. "A Low-power Low-cost Optical Router for Optical Networks-on-Chip in Multiprocessor Systems-on-Chip". In: *IEEE Computer Society Annual Symposium on VLSI*. May 2009, pp. 19–24. DOI: 10.1109/ISVLSI.2009.19.

[8] *2016 Integrated Photonic Systems Roadmap*. 2017. URL: http://photonicsmanufacturing.org/2016-integrated-photonic-systems-roadmap.

[9] K. Bergman. "Flexibly Scalable High Performance Architectures with Embedded Photonics - Keynote". In: *Platform for Advanced Scientific Computing (PASC) Conference*. 2019.

[10] Y. Urino, T. Usuki, J. Fujikata, M. Ishizaka, K. Yamada, T. Horikawa, T. Nakamura, and Y. Arakawa. "Fully Integrated Silicon Optical Interposers with High Bandwidth Density". In: *Advanced Photonics for Communications*. 2014. DOI: 10.1364/IPRSN.2014.IM2A.5.

[11] Y. Urino, T. Nakamura, and Y. Arakawa. "Silicon Optical Interposers for High-Density Optical Interconnects". In: *Silicon Photonics III: Systems and Applications*. 2016, pp. 1–39. DOI: 10.1007/978-3-642-10503-6_1.

[12] H. Huang et al. "Mode Division Multiplexing Using an Orbital Angular Momentum Mode Sorter and MIMO-DSP Over a Graded-Index Few-Mode Optical Fibre". In: *Scientific Reports* 5 (2015), pp. 2045–2322. DOI: 10.1038/srep14931.

[13] A. Malik and P. Singh. "Free Space Optics: Current Applications and Future Challenges". In: *International Journal of Optics* 2015 (2015).

[14] L. Gao, Y. Huo, K. Zang, S. Paik, Y. Chen, J. S. Harris, and Z. Zhou. "On-chip plasmonic waveguide optical waveplate". In: *Scientific reports* 5 (2015).

[15] N. Pleros. "Silicon Photonics and Plasmonics towards Network-on-Chip Functionalities for Disaggregated Computing". In: *Optical Fiber Communication Conference*. 2018, Tu3F.4. DOI: 10.1364/OFC.2018.Tu3F.4. URL: http://www.osapublishing.org/abstract.cfm?URI=OFC-2018-Tu3F.4.

[16] K. L. Tsakmakidis et al. "Breaking Lorentz Reciprocity to Overcome the Time-Bandwidth Limit in Physics and Engineering". In: *Science* 356.6344 (2017), pp. 1260–1264. DOI: 10.1126/science.aam6662.

[17] C. Ríos, M. Stegmaier, P. Hosseini, D. Wang, T. Scherer, C. D. Wright, H. Bhaskaran, and W. H. P. Pernice. "Integrated all-photonic non-volatile multi-level memory". In: *Nature Photonics* 9 (11 2015), pp. 725–732. DOI: 10.1038/nphoton.2015.182.

[18] Wikipedia. *Optical Computing*. 2017. URL: http://en.wikipedia.org/wiki/Optical%5C_computing.

[19] Optalysys. *Optalysys Prototype Proves Optical Processing Technology Will Revolutionise Big Data Analysis and computational Fluid Dynamics*. 2017. URL: http://www.optalysys.com/optalysys-prototype-proves-optical-processing-technology-will-revolutionise-big-data-analysis-computational-fluid-dynamics-cfd.

[20] *Optalysys Rolls Commercial Optical Processor*. HPC Wire. Mar. 2019. URL: https://www.hpcwire.com/2019/03/07/optalysys-rolls-commercial-optical-processor/.

[21] Y. Pan, J. Kim, and G. Memik. "FlexiShare: Channel sharing for an energy-efficient nanophotonic crossbar". In: *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*. 2010, pp. 1–12. DOI: 10.1109/HPCA.2010.5416626.

[22] D. Vantrease et al. "Corona: System Implications of Emerging Nanophotonic Technology". In: *SIGARCH Comput. Archit. News* 36.3 (June 2008), pp. 153–164. DOI: 10.1145/1394608.1382135.

[23] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary. "Firefly: Illuminating Future Network-on-chip with Nanophotonics". In: *SIGARCH Comput. Archit. News* 37.3 (June 2009), pp. 429–440. DOI: 10.1145/1555815.1555808.

[24] M. Petracca, B. G. Lee, K. Bergman, and L. P. Carloni. "Design Exploration of Optical Interconnection Networks for Chip Multiprocessors". In: *Proceedings of the Symposium on High Performance Interconnects (HOTI)*. 2008, pp. 31–40. DOI: 10.1109/HOTI.2008.20.

[25] P. Grani and S. Bartolini. "Design Options for Optical Ring Interconnect in Future Client Devices". In: *Journal on Emerging Technologies in Computing Systems (JETC)* 10.4 (June 2014), 30:1–30:25. DOI: 10.1145/2602155.

[26] I. O'Connor, D. Van Thourhout, and A. Scandurra. "Wavelength Division Multiplexed Photonic Layer on CMOS". In: *Proceedings of the 2012 Interconnection Network Architecture: On-Chip, Multi-Chip Workshop (INA-OCMC)*. 2012, pp. 33–36. DOI: 10.1145/2107763.2107772.

[27] P. Grani, R. Hendry, S. Bartolini, and K. Bergman. "Boosting multi-socket cache-coherency with low-latency silicon photonic interconnects". In: *2015 International Conference on Computing, Networking and Communications (ICNC)*. Feb. 2015, pp. 830–836. DOI: 10.1109/ICCNC.2015.7069453.

[28] Horizon 2020 EU Framework Programme. *Photonics*. 2017. URL: https://ec.europa.eu/programmes/horizon2020/en/h2020-section/photonics.

[29] Photonics21. *The European Technology Platform Photonics21*. 2017. URL: http://www.photonics21.org.

3.4.2 Quantum Computing

Overall View

In the Quantum Technologies Flagship Final Report, the following overall Quantum Computing objective is formulated: *The goal of quantum computing is to complement and outperform classical computers by solving some computational problems more quickly than the best known or the best achievable classical schemes. Current applications include factoring, machine learning, but more and more applications are being discovered. Research focuses both on quantum hardware and quantum software – on building and investigating universal quantum computers, and on operating them, once scaled up, in a fault-tolerant way.* The defined quantum milestones are:

- In 3 years, fault tolerant routes will be demonstrated for making quantum processors with eventually more than 50 qubits
- In 6 years, quantum processor fitted with quantum error correction or robust qubits will be realized, outperforming physical qubits;
- In 10 years, quantum algorithms demonstrating quantum speed-up and outperforming classical computers will be operated.

And finally as far as the quantum computing community is concerned, the enabling tools consist of engineering and control such as further development of optimal control schemes and suitable hardware, materials, cryogenics, lasers, electronics including FPGAs and ASICs, microwave sources, detectors, low-level software.

What we want to achieve is to build a real, scalable quantum computer in a 10 year time frame.[13] When building any computational device, including a Quantum Computer, it is absolutely necessary to include computer engineering as a scientific effort to define the overall system view as well as to provide the low level details.

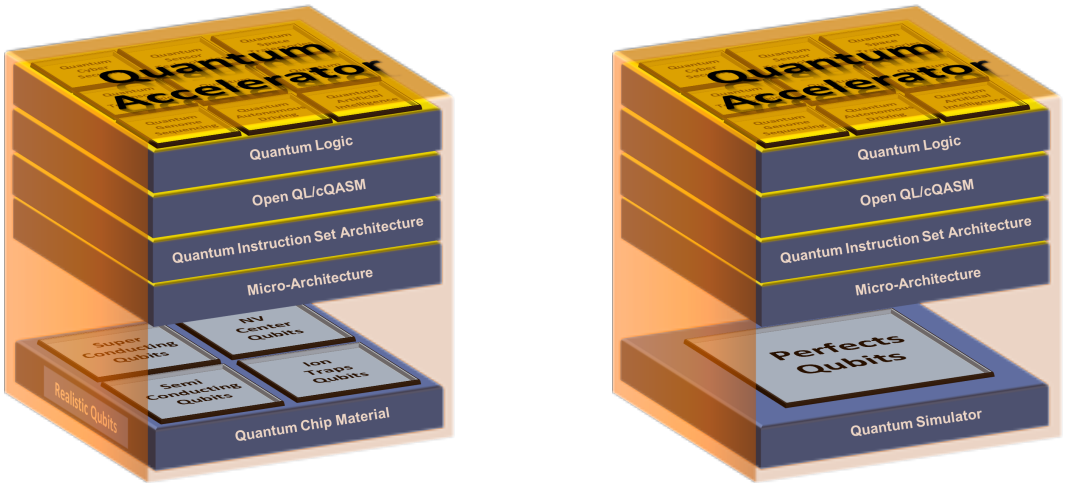
This immediately made the computer engineers in Delft to formulate a different definition and explain the colleagues that a quantum architecture is much more than just the 2D chip just like a computer architecture is much more than assuming it is enough to build a processor connected to some memory. What has been achieved is the definition of the cubic schematic as given in Fig. 3.7 which is one of the first if not the only structured view on what the different components and system layers are of a quantum computer. This particular view is still the main driver of e.g. the collaboration that QuTech in Delft has with the US company Intel.

More recently, the semiconductor industry has expressed interest in the development of the qubit processor. The underlying quantum technology has matured such that it is now reaching the phase where large and established industrial players are becoming increasingly interested and ambitious to be among the first to deliver a usable and realistic quantum platform.

Current State of Quantum Computing

If we compare the evolution of quantum computers with classical computers, quantum computers are in the pre-transistor era. Quantum devices are unreliable and have a large size compared to the expected integration scale, different quantum devices are being developed, and the winning technology has not been decided yet. Moreover, how to scale-up quantum processors is still an open issue. But even more important, quantum computers lack the full stack of layers existing in a classical computer. If today we had quantum hardware with billions of highly reliable qubits, it would be impossible to develop applications for it in the way we do for classical computers. Therefore, in addition to advancing the state of the art on quantum devices, which is the goal of many ongoing industry and academic research projects, we need to develop a full stack to support the development of quantum applications with the same ease as we do for classical computers. Layers in the full stack must include from the definition of suitable high-level languages and the development of the corresponding compilers down to the microarchitecture. Moreover, until quantum computers scale to a much larger number of qubits, simulators are needed. Developing such full stack and quantum simulators to support its execution, as well as developing quantum applications in relevant fields to demonstrate the validity of the proposed approach. We expect to deliver a much needed set of layers so that all the ongoing and future research projects on quantum devices have a much easier way to experiment with and evaluate the physical components they

[13] This section is written from the viewpoint of the Quantum Computer Architecture Lab of Delft University of Technology, Netherlands.



(a) Experimental Full-Stack with realistic qubits.

(b) Simulated Full-Stack with Perfect Qubits.

Figure 3.7: Components and System Layers of a Quantum Computer

develop, also defining a road map for enabling quantum computation and quantum-accelerated applications in Europe and worldwide for the next decade.

Semi-conductor or related companies such as IBM, Intel and Microsoft are increasingly known for their activities in quantum computing but also players such as Google and Rigetti are becoming very present, and the Canadian company D-WAVE is a well-known and established player in this field. Also national initiatives in, e.g. China are becoming increasingly important. This has two main consequences.

First, the companies involved are racing against each other to find ways to make a realistically sized and stable quantum computer. What technology will ultimately be selected for making the quantum processor is still unknown but the set of candidates has been reduced to basing the qubit development on impurities in diamonds for Nitrogen Vacancy (NV) Centres, the use of semiconducting or superconducting qubits, the adoption of Ion traps, quantum annealing such as the D-WAVE machine, or an even more futuristic approach involving Majorana-based qubits which is best described as topological quantum computing. Each technology has advantages and disadvantages and which one will win is still an open issue.

The second major consequence is that companies, countries as well as universities see the increasing need for training quantum engineers, people capable of designing both the quantum device as well as the complete system design and related tools to build a quantum computer. Just like making a digital computer, there are different disciplines involved ranging from pure hardware oriented activities such as material science, circuit design and hardware architecture to more software fields such as operating system and compiler construction to high level programming levels and algorithm design. So educating a quantum engineer involves many disciplines and goes beyond understanding quantum physics.

Building a new kind of computer is a very multidisciplinary task that spans fields ranging from microelectronics up to computer science. Computer Engineering is the connection between computer science and microelectronics. Computer Engineering is also defined as hardware-software co-design which basically means deciding about what should be implemented in hardware and what will stay software.

In conventional CMOS technology, this means that a processor architecture is defined consisting of e.g. the instruction set and the corresponding micro-architecture. In the context of quantum computing, computer engineering then ranges from quantum physics up to computer science. When the request came in Delft to be involved in the creation of a Quantum Computer, the first literature excursions immediately taught us that there is no scientific research that is being done on what it means to build such a machine. In the first conversations with the physicists, the term 'Quantum Architecture' was very frequently used until the computer engineers asked what their definition is of that term. The answer was amazingly simple even though their research was extremely complex: a quantum architecture is a 2D layout of qubits that can be addressed and controlled individually. Therefore, we lack a definition of a system architecture for quantum

computers. Such a system architecture is an essential component for both establishing a common interface for the different projects on quantum computing and defining structured contents for training engineers in this discipline.

As shown in Figure 3.7, we focus more on the levels above the quantum physical chip, which is a necessary, but not sufficient component of building a quantum computer. This layered stack defines the research roadmap and layers that need to be developed when building a quantum computer, going from a high-level description of a quantum algorithm to the actual physical operations on the quantum processor. Quantum algorithms [1] are described by high-level quantum programming languages [2, 3, 4]. Such algorithm description is agnostic to the faulty quantum hardware and assumes that both qubits and quantum operations are perfect. In the compilation layer, quantum algorithms are converted to their fault tolerant (FT) version based on a specific quantum error correction code such as surface code [5] or color codes [6] and compiled into a series of instructions that belong to the quantum instruction set architecture (QISA). The micro-architecture layer contains components that focus on quantum execution (QEX) and parts that are required for quantum error correction (QEC) which together are responsible for the execution of quantum operations and for the detection and correction of errors [7]. We will extend the micro-architecture already developed for the 5 qubit superconducting processor [8] for supporting larger number of qubits and error correction feedback. It is in these layers, where quantum instructions are translated into the actual pulses that are sent through the classical-to-quantum interface to the quantum chip.

The consortium's guiding vision is that quantum computing will be delivered through the union of the classical computer with quantum technology through the integration of a quantum computing device as an accelerator of the general-purpose processors. All accelerators are required to define a model of computation and provide the appropriate consistency in order to support a software ecosystem. As with the semiconductor device, there are many material implementations of a qubit and as such the Q-Machine will define a micro-architecture through which different qubit devices can be integrated into the machine and deliver their value across all applications that utilize the Q-Machine Architecture.

From this discussion, one can also realize that building a complete quantum computer involves more than just building quantum devices. Whereas physicists are mostly working at the quantum chip layer trying to improve the coherence of the qubits and the fidelity of the gates, as well as to increase the number of qubits that can be controlled and entangled, computer and electronic engineers are responsible for the development of the infrastructure required for building such a quantum system. As we will expand in this proposal, the combination of the classical with the quantum logic is needed when investigating and building a full quantum computer. Following Figure 3.7, we briefly describe in the following the different abstraction layers on which it is focusing its research and contributions, starting at the application layer and going down to the micro-architectural one. The ongoing research does not focus on the quantum-to-classical layer nor on the quantum physics in making good qubits. However, explicit links with (those layers and) quantum physics and control electronics projects will be established and described later.

Important in this context is the definition and implementation of an open-standard quantum computer system design, which has the overall architecture and fault tolerance required to solve problems arising in the computational science domain, and which are of practical interest for the different industries such as aerospace and medicine. The overall design will be detailed and implemented both in an experimental way using physical quantum devices as well as large scale simulation models.

- **OBJ 1 - Full Stack System Design:** design and develop a full stack quantum system that integrates all scientific and technological results from different fields, ranging from algorithms up to the quantum processors. This will allow potential users of that device to easily express quantum algorithms using an appropriate programming language. This full-stack approach defines and implements the bridge between the qubit device and the user application-driven world.
- **OBJ 2 - Scalable Architecture:** provide an open and available simulation platform for design space exploration of the quantum computer architecture as well as an instrument to advance application development. It will allow to control large number of qubits (over 1000 qubits), and will include fault-tolerance mechanisms, mapping of quantum circuits and routing of quantum states.

- **OBJ 3 - Societal Relevant Quantum Applications:** provide relevant quantum search algorithms for DNA analysis, quantum linear solvers, and quantum-accelerated optimization algorithms with immediate use in the medicine and aerospace domains. We also provide associated benchmarking capability to evaluate quantum computers as they evolve to a level of practical industrial interest.

Any computational platform needs a system design that ties together all layers going from algorithm to physical execution. We will investigate, develop and implement an architecture for a quantum computer, the Q-Machine, that includes and integrates algorithms and applications, programming languages, compilers and run-time, instruction set architecture and micro-architecture. The Q-Machine shall be realised in two implementations: the first is a simulator that will use the QX-simulator as a back-end to perform and validate large-scale and fault-tolerant quantum computing. The second implementation is that of most likely two physical machines that will utilize one of the latest cryogenic qubit devices, superconducting qubits or silicon spin qubits, to verify the results of this research (up to TRL7). These machines will each support the new quantum architecture and corresponding processors on which applications can be run. These devices adopt and enhance the software development stack developed by Delft University of Technology, as well as the existing experimental quantum demonstrator.

The development and realisation of a new computing paradigm, such as the Q-Machine, requires participation, expertise and the capabilities from a broad consortium of different disciplines. It is necessary to assemble experts ranging from the end customer, e.g. represented by the aerospace and genome industry and HPC, the computer science and mathematics community as well as the classical semiconductor industry with computer and systems engineering experts.

Short term vision - Quantum Computer Engineering is a very new field which is only in the very first phase of its existence. Computer architecture research on developing a gate-based quantum computer is basically not existing and QuTech is one of the few places on earth where this line of research is actively pursued. The work planned for this project is heavily based on the ongoing QuTech research on these topics. The Delft team has already demonstrated the use of a micro-architecture for both the superconducting as well as the semiconducting qubits. They have also developed OpenQL and are instrumental in standardising the Quantum Assembly Language which allows to express quantum logic in the quantum instructions which can be executed. The notion of micro-code generation has also been introduced as part of the micro-architecture.

Quantum Genome Sequencing – a Quantum Accelerator Application

Since one of the first papers about quantum computing by R. Feynman in 1982 [9], research on quantum computing has focused on the development of low-level quantum hardware components like superconducting qubits, ion trap qubits or spin-qubits. The design of proof-of-concept quantum algorithms and their analysis with respect to their theoretical complexity improvements over classical algorithms has also received some attention. A true quantum killer application that demonstrates the exponential performance increase of quantum over conventional computers *in practice* is, however, still missing but is urgently needed to convince quantum sceptics about the usefulness of quantum computing and to make it a mainstream technology within the coming 10 years.

Genomics concerns the application of DNA sequencing methods and bioinformatics algorithms to understand the structure and function of the genome of an organism. This discipline has revealed insights with scientific and clinical significance, such as the causes that drive cancer progression, as well as the intragenomic processes which greatly influence evolution. Other practical benefits include enhancing food quality and quantity from plants and animals. An exciting prospect is personalised medicine, in which accurate diagnostic testing can identify patients who can benefit from targeted therapies [10].

Such rapid progress in genomics is based on exponential advances in the capability of sequencing technology, as shown in Figure 3.8. However, to keep up with these advances, which outpace Moore's Law, we need to address new computational challenges of efficiently analysing and storing the vast quantities of genomics data. Despite the continual development of tools to process genomic data, current approaches are still yet to meet the requirements for large-scale clinical genomics [11]. In this case, patient turnaround time, ease-of-use, resilient operation and running costs are critical.

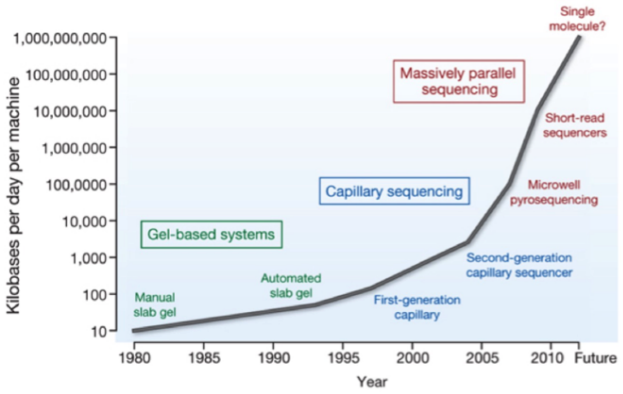


Figure 3.8: Advances in Sequencing Technology [12]

Quantum computing promises to become a computational game changer, allowing the calculation of various algorithms much faster (in some cases exponentially faster) than their classical counterparts. One of the most suitable types of algorithms quantum computers can accelerate are those that have abundance of input data parallelism. With the availability of enough qubit capacity, the entire parallelizable input dataset can be encoded simultaneously as a superposition of a single wave function. This makes it possible to perform the computation of the entire dataset in parallel. This kind of computational acceleration provides a promising approach to address the computational challenges of DNA analysis algorithms.

Quantum Programming Languages and Compilers

The quantum algorithms and applications presented in the previous section, can be described using a high-level programming language such as Q#, Scaffold, Quipper and OpenQL [13, 3, 2], and compiled into a series of instructions that belong to the (quantum) instruction set architecture.

As shown in Figure 3.9, the compiler infrastructure for such a heterogeneous system will consist of the classical or host compiler combined with a quantum compiler. The host compiler compiles for the classical logic and the quantum compiler will produce the quantum circuits (we adopt the circuit model as a computational model) and perform reversible circuit design, quantum gate decomposition and circuit mapping that includes scheduling of operations and placement of qubits. The output of the compiler will be a series of instructions, expressed in a quantum Assembly language QASM, that belong to the defined instruction set architecture. Note that the architectural heterogeneity where classical processors are combined with different accelerators such as the quantum accelerator, imposes a specific compiler structure where the different instruction sets are targeted and ultimately combined in one binary file which will be executed.

A key pass in the quantum compiler is the generation of fault tolerant (FT) quantum circuits. The main handicap of quantum technology is its fragility. First, the coherence time of qubits is extremely short. For example, superconducting qubits may lose its information in tens of microseconds [14, 15]. Second, quantum operations are unreliable with error rates around 0.1% [16]. It is therefore inconceivable to think about building a quantum computer without error correction. Quantum Error Correction (QEC) is more challenging than classical error correction because unknown quantum states cannot be copied (no-cloning theorem), quantum errors are continuous and any measurement will destroy the information stored in qubits. The basic idea of modern QEC techniques is to use several physical imperfect qubits to compose more reliable units called *logical qubits* based on a specific quantum error correction code [17, 18, 19, 20, 21, 22], e.g., surface code [5] and continuously monitoring the quantum system to detect and recover from possible errors.

We will also investigate classical reliability mechanisms throughout the stack from the application and the compiler layer all the way into the micro-architecture and circuit layers to achieve resilience in a co-designed manner. This will require overhauling classical fault-tolerance schemes such as checkpointing and replicated execution and adapt them to the resilience requirements of a quantum computing system. Error reporting will be propagated up the system stack to facilitate the application of holistic fault-tolerance.

After the discussion of the quantum algorithm layer and the needed quantum compiler, we focus in the next sections on the quantum instruction set architecture and the micro-architecture that we intend to investigate and build, which should be as independent of the underlying quantum technology as possible.

QISA and Micro-Architecture

As we already mentioned, a quantum computer will not be an standalone machine but an heterogeneous

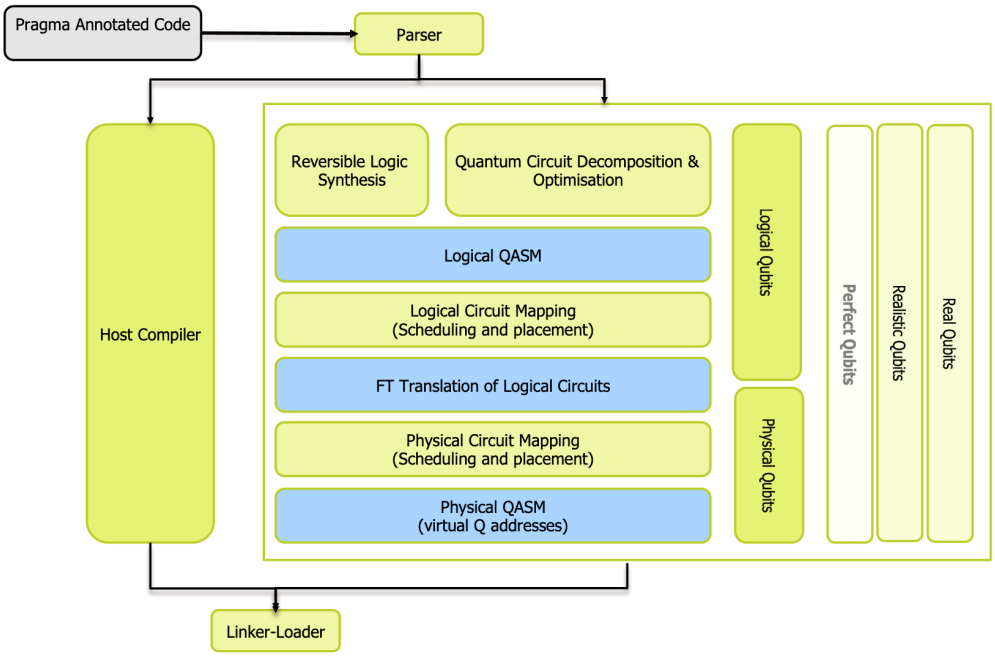


Figure 3.9: Compiler Infrastructure

system, in which classical parts will have to interact with the quantum accelerator or coprocessor. In Figure 3.10, we show what is currently understood as the main system design where accelerators extend the classical server architecture. What is important to emphasise is that such a heterogeneous multi-core architecture is basically a multi-instruction set architecture where e.g. the FPGA, the GPU and now also the future quantum accelerator have their own instruction set which will be targeted by their respective compilers. Currently, GPUs and FPGAs combined with many classical processors are a big component of modern server architectures to provide the necessary performance speedup. However, the real performance breakthrough may come from adding a fully operational quantum processor as an accelerator. In the case of a quantum accelerator, any application will have a lot of classical logic but also calls to quantum libraries which will be called from time to time.

Based on Figure 3.10, we now describe the layers Quantum Instruction Set Architecture (QISA) and the corresponding micro-architecture at a high level such that we have basic design of a quantum computer for supporting the execution of quantum instructions and error correction mechanisms. The instruction set architecture (ISA) is the interface between hardware and software and is essential in a fully programmable classical computer. So is QISA in a programmable quantum computer. Existing instruction set architecture definitions for quantum computing mostly focus on the usage of the description and optimization of quantum applications without considering the low-level constraints of the interface to the quantum processor. It is challenging to design an instruction set that suffices to represent the semantics of quantum applications and to incorporate the quantum execution requirements, e.g., timing constraints.

It is a prevailing idea that quantum compilers generate technology-dependent instructions [23, 3, 24]. However, not all technology-dependent information can be determined at compile time because some information can only be generated at runtime due to hardware limitations. An example is the presence of defects on a quantum processor affecting the layout of qubits used in the algorithm. In addition, the following observations hold: (1) quantum technology is rapidly evolving, and more optimized ways of implementing the quantum gates are continuously explored and proposed; a way to easily introduce those changes, without impacting the rest of the architecture, is important; (2) depending on the qubit technology, the kind, number and sequence of the pulses can vary. Hence, it forms another challenge to micro-architecturally support a set of quantum instructions which is as independent as possible of a particular technology and its current state-of-the-art.

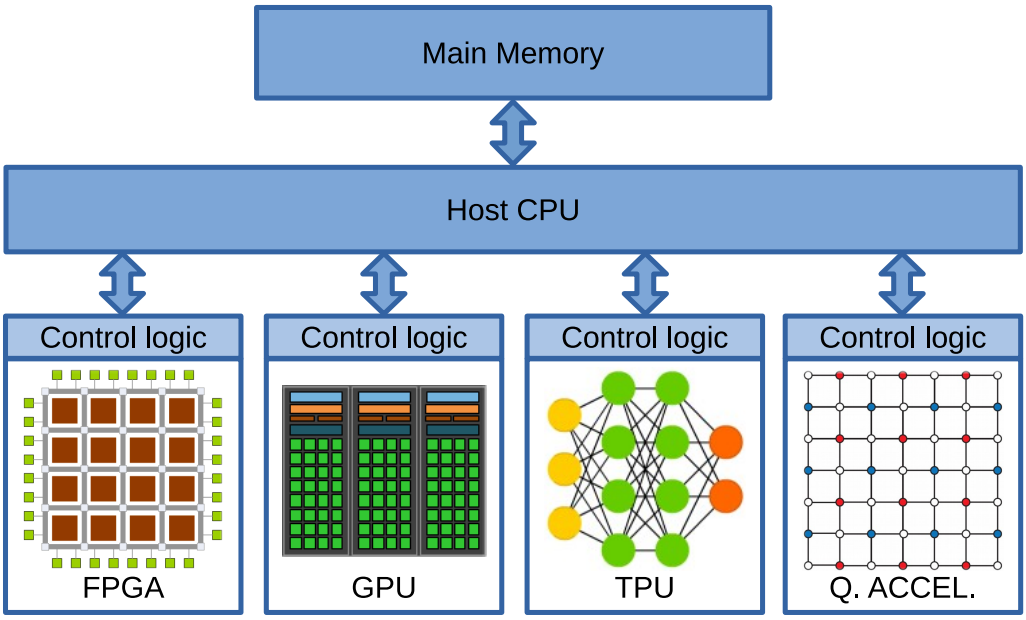


Figure 3.10: Heterogeneous System Design with Different Kinds of Accelerators.

Example of Quantum Computer Micro-Architecture

The overall micro-architecture, as shown in Figure 3.11 is a heterogeneous architecture which includes a classical CPU as a host and a quantum coprocessor as an accelerator. The input of the micro-architecture is a binary file generated by a compiler infrastructure where classical code and quantum code are combined. As we mentioned previously, the classical code is produced by a conventional compiler such as GCC and executed by the classical host CPU. Quantum code is generated by a quantum compiler and executed by the quantum coprocessor. As shown in Figure 3.11, based on the opcode of the instruction, the arbiter sends the instruction either to the host CPU or to the quantum accelerator. In the remainder sections, we focus on the architectural support for the execution of quantum instructions and not on the execution of instructions on the classical CPU. The goal of this research part is to define a Quantum Hardware Abstraction Layer (QHAL) such that quantum accelerators can be easily integrated with classical processors. The QHAL which will be defined and implemented such that it is used in the full stack simulation that we intend in this project as the final demonstrator of the research.

In the quantum accelerator, executed instructions in general flow through modules from left to right. The topmost block of the figure represents the quantum chip and the other blocks represent the classical logic needed to control it. The blue parts (classical-quantum interface) are underlying technology dependent wave control modules. Digital-to-Analogue Converters (DAC) are used to generate analogue waveforms to drive the quantum chip and Analogue-to-Digital Converters (ADC) to read the measurement analogue waveform. They receive or send signals to the Flux and Wave Control Unit and the Measurement Discrimination Unit. In the following paragraphs, we discuss the functional blocks that are needed to execute instructions in QISA and to support quantum error correction. These blocks are based on the control logic developed for the Transmon-based processor as described in [14].

The Quantum Control Unit, called QCU, which is one implementation of the QHAL, decodes the instructions belonging to the QISA and performs the required quantum operations, feedback control and QEC. The QCU can also communicate with the host CPU where classical computation is carried through the eXchange Register File (XRF). The QCU, includes the following blocks:

- **Quantum Instruction Cache, Qubit Address Translation and Q Symbol Table:** Instructions from the Quantum Instruction Cache are first address-translated by the Qubit Address Translation module. This means that the compiler-generated, virtual qubit addresses are translated into physical ones. This is based on the information contained in the Qubit Symbol Table which provides the overview of the exact physical loca-

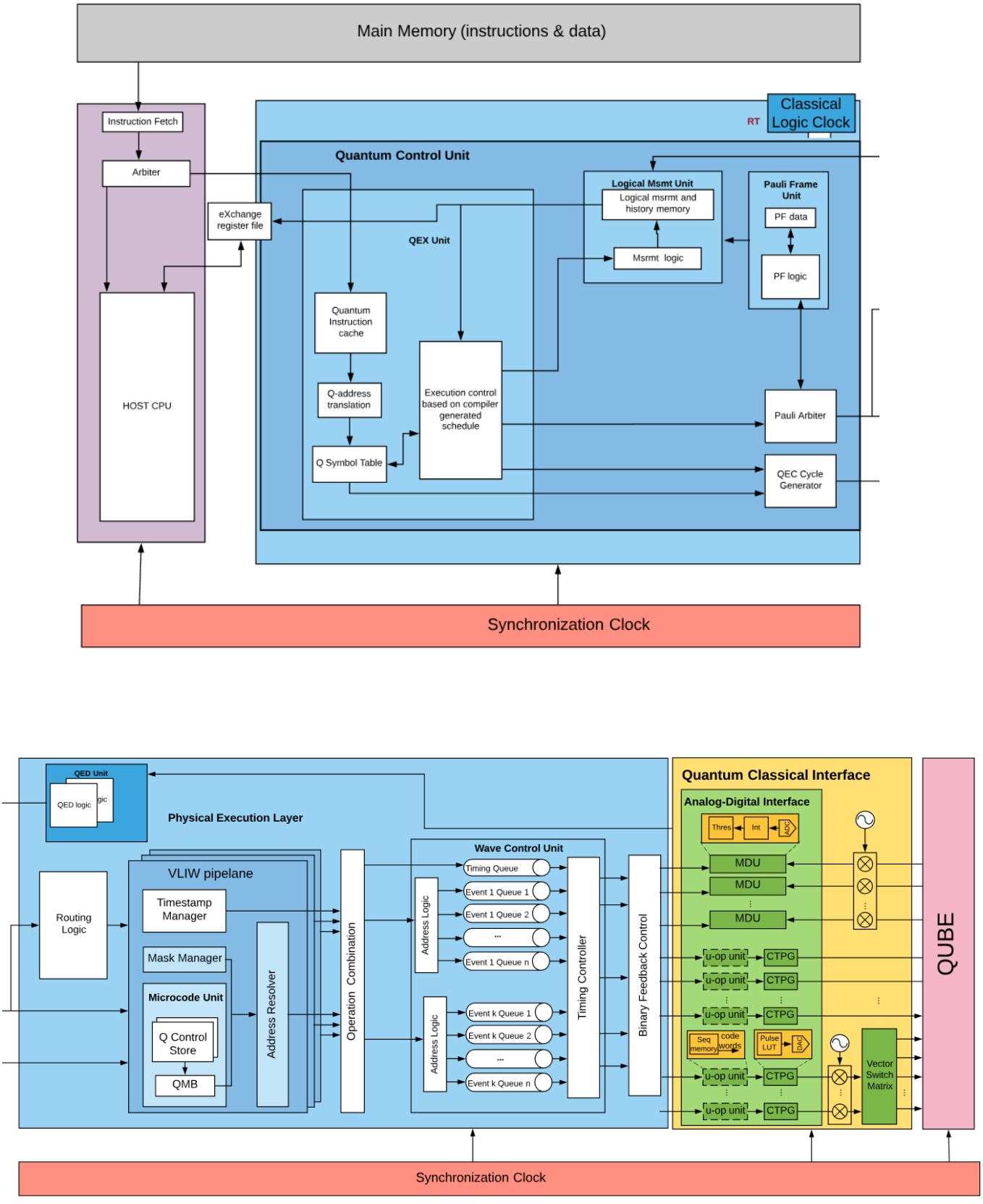


Figure 3.11: Example of Quantum Computer Micro-Architecture

tion of the logical qubits on the chip and provides information on what logical qubits are still alive. This information needs to be updated when quantum states or qubits are moved (routing).

- **Execution Controller:** The Execution Controller can be seen as the brain of the Quantum Control Unit and it asks the Quantum Instruction Cache and the Qubit Address Translation to perform the necessary instruction fetch and decode. The Execution Controller then will make sure the necessary steps in the instruction execution are taken such as sending them to the Pauli Arbiter for further processing. It will also be responsible for keeping the Q Symbol Table up to date.

- **Cycle Generator:** As far as error correction is concerned, the necessary ESM instructions for the entire qubit plane are added at run-time by the QED Cycle Generator, based on the information stored in the Q Symbol Table. It also reduces substantially the datapath for the execution of these instructions.

- **QED Unit:** The responsibility of the QED Unit is to detect errors based on error syndrome measurement results. These measurements are decoded to identify what kind of error was produced and on which qubit. The decoder will use decoding algorithms such as Blossom algorithm.

- **The Pauli Frame Unit and Pauli Arbiter:** The Pauli Frame mechanism [25] allows us to classically track Pauli errors without physically correcting them. The Pauli Frame Unit manages the Pauli records for every data qubit. The Pauli Arbiter receives instructions from the Execution Controller and the QED Unit. It skips all operations on ancilla qubits and sends them directly to the PEL, regardless of the operation type.

- **Logical Measurement Unit:** The function of the Logical Measurement Unit is to combine the data qubit measurement results into a logical measurement result for a logical qubit. The Logical Measurement Unit sends the logical measurement result to the ERF, where it can be used in Binary Control by the Execution Controller, or picked up by the host processor and used e.g. in branch decisions.

Conclusion

This section provides an overview of what quantum computing involves and where we are in the current years. Compared to classical computers, it is clear that quantum computing is in the pre-transistor phase. This is mainly due to the fact that multiple technologies are competing against each other to be the dominant qubit-transistor technology. The second great problem that still needs to be solved is the computational behaviour of the qubits which is many orders of magnitude lower than any computation performed by a CMOS-based transistor. A final challenge is that the quantum bits are analogue and need to be controlled by a digital micro-architecture. One can think of developing an analogue computer again but that technology is far from evident in the next 10 years. We need to focus very intensively on quantum computing but we have to realise that it takes at least 10 to 15 years before the first quantum accelerators will be available.

References

[1] J. Stephen. "Quantum Algorithm Zoo". In: *list available at http://math.nist.gov/quantum/zoo* (2011).

[2] A. Green et al. "An introduction to quantum programming in quipper". In: *Reversible Computation*. Springer. 2013, pp. 110–124.

[3] A. J. Abhari et al. *Scaffold: Quantum programming language*. Tech. rep. DTIC Document, 2012.

[4] D. Wecker and K. M. Svore. "LIQUi|>: A software design architecture and domain-specific language for quantum computing". In: *arXiv preprint arXiv:1402.4467* (2014).

[5] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. "Surface codes: Towards practical large-scale quantum computation". In: *Physical Review A* 86.3 (2012), p. 032324.

[6] H. Bombín. "Dimensional jump in quantum error correction". In: *New Journal of Physics* 18.4 (2016), p. 043038.

[7] X. Fu et al. "A heterogeneous quantum computer architecture". In: *Proceedings of the ACM International Conference on Computing Frontiers*. ACM. 2016, pp. 323–330.

[8] D. Risté, S. Poletto, M.-Z. Huang, A. Bruno, V. Vesterinen, O.-P. Saira, and L. DiCarlo. "Detecting bit-flip errors in a logical qubit using stabilizer measurements". In: *Nature communications* 6 (2015).

[9] R. P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6-7 (1982), 467–488. DOI: {10.1007/BF02650179}.

[10] M. Hamburg and F. Collins. "The path to personalized medicine". In: *New England Journal of Medicine* (2003), 363:301–304.

[11] R. Gullapalli et al. “Next generation sequencing in clinical medicine: Challenges and lessons for pathology and biomedical informatics”. In: *J. Pathol. Inform.* (2012), 3(1):40.

[12] M. Stratton et al. “The cancer gene”. In: *Nature* (2009), 458:719–724.

[13] “The Q# Programming Language”. In: *https://docs.microsoft.com/en-us/quantum/quantum-qr-intro?view=qsharp-preview* (2017).

[14] D. Riste, S. Poletto, M. .-.-Z. Huang, et al. “Detecting bit-flip errors in a logical qubit using stabilizer measurements”. In: *Nat Commun* 6 (Apr. 2015). URL: http://dx.doi.org/10.1038/ncomms7983.

[15] A. Córcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow. “Demonstration of a quantum error detection code using a square lattice of four superconducting qubits”. In: *Nature communications* 6 (2015).

[16] J. Kelly et al. “State preservation by repetitive error detection in a superconducting quantum circuit”. In: *Nature* 519.7541 (2015), pp. 66–69.

[17] B. T. Lidar D. *Quantum Error Correction*. 2013.

[18] P. W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Physical review A* 52.4 (1995), R2493.

[19] A. Steane. “Multiple-particle interference and quantum error correction”. In: *Proceedings of the Royal Society of London A: Math., Phys. and Eng. Sciences*. 1996.

[20] A. R. Calderbank and P. W. Shor. “Good quantum error-correcting codes exist”. In: *Phys. Rev. A* 54.2 (1996), p. 1098.

[21] D. Gottesman. “Class of quantum error-correcting codes saturating the quantum Hamming bound”. In: *Phys. Rev. A* 54.3 (1996), p. 1862.

[22] H. Bombin and M. A. Martin-Delgado. “Topological quantum distillation”. In: *Phys. Rev. Lett.* 97.18 (2006), p. 180501.

[23] K. M. Svore, A. V. Aho, A. W. Cross, I. Chuang, and I. L. Markov. “A layered software architecture for quantum computing design tools”. In: *Computer* 1 (2006), pp. 74–83.

[24] T. Häner, D. S. Steiger, K. Svore, and M. Troyer. “A software methodology for compiling quantum programs”. In: *arXiv preprint arXiv:1604.01401* (2016).

[25] E. Knill. “Quantum computing with realistically noisy devices”. In: *Nature* 434.7029 (Mar. 2005), pp. 39–44. URL: http://dx.doi.org/10.1038/nature03350.

3.4.3 Beyond CMOS Technologies

Nano structures like Carbon Nanotubes (CNT) or Silicon Nanowires (SiNW) expose a number of special properties which make them attractive to build logic circuits or memory cells.

Carbon Nanotubes

Carbon Nanotubes (CNTs) are tubular structures of carbon atoms. These tubes can be single-walled (SWNT) or multi-walled nanotubes (MWNT). Their diameter is in the range of a few nanometers. Their electrical characteristics vary, depending on their molecular structure, between metallic and semiconducting [1]. CNTs can also be doped by e.g. nitrogen or boron casting n-type or p-type CNTs.

A CNTFET consists of two metal contacts which are connected via a CNT. These contacts are the drain and source of the transistor. The gate is located next to or around the CNT and separated via a layer of silicon oxide [2]. Also, crossed lines of appropriately selected CNTs can form a tunnel diode. This requires the right spacing between the crossed lines. The spacing can be changed by applying appropriate voltage to the crossing CNTs.

CNTs can be used to build nano-crossbars, which logically are similar to a PLA (programmable logic array). The crosspoints act as diodes if the CNTs have the proper structure in both directions They offer wired-AND conjunctions of the input signal. Together with inversion/buffering facilities, they can create freely programmable logic structures. The density of active elements is much higher as with individually formed CNTFETs.

Current State of CNT In September 2013, Max Shulaker from Stanford University published a computer with digital circuits based on CNTFETs. It contained a 1 bit processor, consisting of 178 transistors and runs with a frequency of 1 kHz [3]. The current state on CNTFETs was demonstrated by Shulaker (now at MIT) by implementing a RISC V based processor with 32 bit instructions and a 16 bit datapath running at 10kHz and consisting of 14000 transistors[4].

While the RISC V implementation is an impressive achievement, it should be noted that the size of the CNTFETS currently is in the range of 1 μm^2 which is orders of magnitudes higher than in conventional silicon technology. Also, the switching speed is by far not comparable to regular CMOS technology. The main purpose of this demonstration chip is to prove that the difficult manufacturing process and the inherent uncertainties of using CNTs can be managed even at large scale.

Nanotube-based RAM is a proprietary memory technology for non-volatile random access memory developed by Nantero (this company also refers to this memory as NRAM) and relies on crossing CNTs as described above. An NRAM "cell" consists of a non-woven fabric matrix of CNTs located between two electrodes. The resistance state of the fabric is high (representing *off* or 0 state) when (most of) the CNTs are not in contact and is low (representing *on* or 1 state) vice versa. Switching the NRAM is done by adjusting the space between the layers of CNTs. In theory NRAM can reach the density of DRAM while providing performance similar to SRAM [5]. NRAMs are supposed to be produced in volume in 2020.

Currently there is little to no activity to develop CNT based programmabe logic devices in crossbar architecture.

Impact on Hardware Apart from active functionality, CNTs are also excellent thermal conductors. As such, they could significantly improve conducting heat away from CPU chips [6].

The impact of CNTFETs is not yet clear. Their major benefit of CNTFETs is their superior energy efficiency compared to CMOS transistors. With additional

The use of CNTs in RAMs is at a comercial threshold and will have a similar impact as other memristor technologies.

CNT based NanoPLAs promise to deliver very high logic density for programmable logic. This would enable the inclusion of very large programmable devices into processors and could thus be the basis for large scale application specific accelerators.

Silicon Nanowires or Nanosheets

Silicon Nanowires are the natural extension of FinFet technology. Silicon NanoWires are silicon filaments with circular cross sections surrounded by a gate all-around. This structure gives an excellent electrostatic control of the charge under the gate area, and thus

enhances conduction in the ON region and reduces leakage in the OFF region. Silicon Nanowires can be horizontally stacked, thus providing designers with a small stack of transistors in parallel [7]. Intel has shown working Silicon Nanowires based transistors and their extension to Silicon Nanosheets, that are similar to stacks of horizontal nanowires but with an oval-elongated cross section.

Silicon Nanowires can be used to design controlled-polarity transistors when interfaced to nickel source/drain contacts [8]. Such transistors have two independent gates. A polarity gate dopes electrostatically (i.e., through a radial field) the underlying Schottky junctions at the contact interfaces. As a result, the polarization of this gate can block the flow of either electrons or holes in this natively ambipolar device, thus generating a *p* or an *n* transistor respectively. The control gate is then used to turn the transistor on or off. Also these types of transistors can be fabricated as horizontal stacks. Their advantage is their lack of implanted regions, and thus the avoidance of the related dopant variability problems.

Vertical nanowires have also been studied, and they can potentially yield very dense computational structures [9]. Nevertheless, they have a much higher fabrication complexity and their readiness for being used in computational circuits is still far.

SiNWs can be formed in a bottom up self-assembly process. This might lead to substantially smaller structures as those that can be formed by lithographic processes. Additionally, SiNWs can be doped and thus, crossed lines of appropriately doped SiNW lines can form diodes.

Current State of SiNW Nano crossbars have been created from SiNWs [10]. Similar to CNT based crossbars, the fundamental problem is the high defect density of the resulting circuits. Under normal semiconductor classifications, these devices would be considered broken. In fact, usage of these devices is only possible, if the individual defects of the devices can be respected during the logic mapping stage of the HW synthesis [11].

While there is not such an impressive demonstration chip for SiNWs as for CNTFETS, the area supremacy of SiNW circuits has already been shown in 2005 by DeHon[12]. He demonstrated that such devices can reach a logic density which is two orders of magnitude higher than traditional FPGAs built with CMOS technology.

Currently, less research on nanowires is active than in the early 2000s. Nevertheless, some groups are pushing the usage of nanowires for the creation of logic circuits. At the same time, more research is going on to deal with the high defect density.

Impact on Hardware SiNW devices are currently only used in NanoPLAs. The major reason for this restriction is their manufacturing process. They can only efficiently be manufactured if identical structures are created in large quantity. This perfectly fits NanoPLAs but not irregular device structures such as processors. Thus, SiNW will most likely only be relevant for the realization of programmable logic, as used e.g. in application specific accelerators.

Two-dimensional (2D) Electronics

Groundbreaking research on graphene has paved the way to explore various two-dimensional (2D) electronic materials. The applications of graphene in nanoelectronics are limited, because graphene does not have a bandgap and thus it is harder (though not impossible) to fabricate efficient transistors. This decade has shown a surge in research on materials with a structure like graphene, where a mono layer (or few adjacent layers) can provide the means for fabricating transistors. Despite the diversity in conduction properties and atomic composition, all 2-D materials consist of covalently-bonded in-plane layers that are held together by weak van der Waals interactions to form a 3-D crystal. In general, a 2D materials are transition metal dichalcogenides (TMDC) composed by a transition metal sandwiched between two chalcogen atoms.

Radisavljevic and co-workers (under the supervision of A. Kis [13]) designed and fabricated the first transistor in Molybdenum DiSulfide (MOS2), which constituted the entry point of 2D materials into nanoelectronics. Wachter [14] designed in 2017 the first processor, though simple, in this material, with about 120 transistors. A major limitation of MOS2 is the inability of realizing complementary devices and circuits, thus requiring the use of depletion loads like in NMOS that contribute to static power consumption. Other 2D materials, like Tungsten DiSelenide (WSe2) have

been used to realize both *n* and *p* transistors, thus enabling complementary logic in 2D. Resta [15] designed and fabricated controlled-polarity WSe2 transistors, and with these elements he fabricated and validated a simple cell library including ANDs, ORs, XORs and MAJority gates. This field is evolving rapidly, with few materials being studied for their properties and attitude towards supporting high-performance, low-power computation. A recent survey is [16].

Superconducting Electronics

Superconducting electronics (SCE) is a branch of engineering that leverages computation at few degrees Kelvin (typically 4K) where resistive effects can be neglected and where switching is achieved by Josephson junctions (JJ). Current difficulties in downscaling CMOS have made superconducting electronics quite attractive for the following reasons. First, the technology can match and extend current performance requirements at lower energy cost. ALU prototypes have been shown to run at 20-50GHz clock rates and with increasingly higher power efficiency. Whereas experimental data vary according to circuit family, size and year of production, it is possible to measure a power consumption that is two orders of magnitude lower as compared to standard CMOS [17], while considering a cryocooling efficacy of 0.1 %. Performance of single-precision operations is around 1 TFLOPS/Watt [18]. Second, today current superconductor circuits are designed in a 250 nm technology, much easier to realize in integrated fashion (as compared to 5 nm CMOS) and with a horizon of a 10-50X possible downscaling, thus projecting one or two decades of further improvement. Cryocooling efficacy is expected to improve as well. Moreover, superconducting interconnect wires allow the ballistic transfer of picosecond waveforms. Therefore, SCE is a strong candidate for high-performance large system design in the coming decade.

IBM led a strong effort in SCE in the 70s with the objective of building computers that would outperform the currently-available technology. The circuits utilized Josephson junctions exhibiting hysteresis in their resistive states (i.e., resistive and superconductive). The JJ acts as a switch that can be set and reset by applying a current. A logic TRUE is associated with the JJ in its resistive state, and a logic FALSE with its superconductive state. This effort faded in the mid 80s, because of various drawbacks, including the choice of materials and the latching operation of logic [19].

Likharev [19] brought back strong interest in SCE by proposing rapid single flux quantum (RSFQ) circuits. In these circuits, the logic values (TRUE, FALSE) are represented by the presence or absence of single flux quantum pulses called fluxons. Junctions are DC biased and when a pulse is applied to the junction, the small associated current pulse can be sufficient to drive the current level over its threshold and to generate a pulse that can be propagated through the circuit. This type of behavior is often called Josephson transmission line (JTL) and it is the basic operational principle of RSFQ circuits that conditionally propagate flux pulses. A specific feature of RSFQ circuits is that logic gates are clocked, and that the overall circuit is pipelined. The RSFQ technology evolved in many directions, e.g., energy-efficient SFQ (eSFQ) [20], reciprocal quantum logic (RQL) [17] and low-voltage RSFQ (LV-RSFQ) [21]. Various realizations of ALUs have been reported, with deep-pipelined, wave-pipelined and asynchronous operation [22].

This technology (and its variations) has several peculiarities. In particular, it is worth mentioning the following. Pulse splitters are used to handle multiple fanouts. Registers are simpler to implement (as compared to CMOS). Conversely, logic gates are more complex (in terms of elementary components). Logic gates can be realized by combining JJs and inductors with different topologies. A fundamental logic gate in RSFQ is a majority gate, that can be simplified to realize the AND and OR functions. Whereas in the past the interest in this technology was related to the realization of arithmetic units (e.g., adders and multipliers) that exploit widely the majority function, today majority logic is widely applicable to general digital design.

Recent research work has addressed technologies that target low-energy consumption. This can be achieved by using AC power (i.e., alternating current supply). In RQL, power is carried by transmission lines. Two signals in quadrature are magnetically coupled to generate a 4-phase trigger. A TRUE logic signal is represented by sending a positive pulse followed by a negative pulse, while a FALSE logic signal is just the absence of the pulsed signals. An alternative technology is adiabatic quantum flux parametron (AQFP) where the circuits are also biased by AC power. (A parametron is a resonant circuit with a nonlinear reactive element.) As an example, Takeuchi [23] used a 3-phase bias/excitation as both multi-clock signal and power supply. In general, signal propagation in AQFP circuits requires overlapping clock signals from

neighboring phases [24]. In AQFP, inductor loop pairs are used to store logic information in terms of flux quanta depending on the direction of an input current and to the magnetic coupling to other inductors. A corresponding output current represents the output of a logic gate. It was shown [25] that the ¤parallel combination¤ of three AQFP buffers yields a majority gate. Recent publications [24] have also advocated the design and use of majority logic primitives in AQFP design. Simple cell libraries have been designed for AQFP as well as some simple synthesis flow from an HDL description to a cell-based physical design.

References

[1] Wikipedia. *carbon nanotubes*. URL: https://en.wikipedia.org/wiki/Carbon%5C_nanotube.

[2] L. Rispal. "Large scale fabrication of field-effect devices based on in situ grown carbon nanotubes". PhD thesis. Darmstädter Dissertationen, 2009.

[3] M. M. Shulaker, G. Hills, N. Patil, H. Wei, H.-Y. Chen, H.-S. P. Wong, and S. Mitra. "Carbon nanotube computer". In: *Nature* 501.7468 (2013), pp. 526–530.

[4] G. Hills, C. Lau, A. Wright, et al. "Modern microprocessor built from complementary carbon nanotube transistors". In: *Nature* 572 (Aug. 2019), pp. 595–602. DOI: 10.1038/s41586-019-1493-8.

[5] Wikipedia. *Nano-RAM*. URL: https://en.wikipedia.org/wiki/Nano-RAM.

[6] J. Hruska. *This carbon nanotube heatsink is six times more thermally conductive, could trigger a revolution in CPU clock speeds*. URL: http://www.extremetech.com/extreme/175457-this-carbon-nanotube-heatsink-is-six-times-more-thermally-conductive-could-trigger-a-revolution-in-cpu-clock-speeds.

[7] M. Zervas, D. Sacchetto, G. De Micheli and Y. Leblebici. "Top-down fabrication of very-high density vertically stacked silicon nanowire arrays with low temperature budget". In: *Microelectronic Engineering* 88.10 (2011), pp. 3127–3132.

[8] P.-E. Gaillardon, L. G. Amaru', S. K. Bobba, M. De Marchi, D. Sacchetto, and G. De Micheli. "Nanowire systems: technology and design". In: *Philosophical Transactions of the Royal Society of London A* 372.2012 (2014).

[9] Y. Guerfi, and GH. Larrieu. "Vertical Silicon Nanowire Field Effect Transistors with Nanoscale Gate-All-Around". In: *Nanoscale Research Letters* (2016).

[10] S. Devisree, A. Kumar, and R. Raj. "Nanoscale FPGA with reconfigurability". In: *Electrotechnical Conference (MELECON), 2016 18th Mediterranean*. IEEE. 2016, pp. 1–5.

[11] M. Zamani and M. B. Tahoori. "Self-timed nano-PLA". In: *2011 IEEE/ACM International Symposium on Nanoscale Architectures*. June 2011, pp. 78–85. DOI: 10.1109/NANOARCH.2011.5941487.

[12] A. Dehon. "Nanowire-Based Programmable Architectures". In: *J. Emerg. Technol. Comput. Syst.* 1.2 (July 2005), pp. 109–162. DOI: 10.1145/1084748.1084750. URL: https://doi.org/10.1145/1084748.1084750.

[13] B. Radisavljevic, A. Radenovic, J. Brivio, and A. Kis. "Single-layer MoS_2 transistors". In: *Nature Nanotech* 6 (2011).

[14] S. Wachter, D. Polyushkin, O. Bethge, and T. Mueller. "A microprocessor based on a two-dimensional semiconductor". In: *Nature Communications* 8 (2017), p. 14948.

[15] G.V. Resta, Y. Balaji, D. Lin, I.P. Radu, F. Catthoor, P.-E. Gaillardon, G. De Micheli. "Doping-Free Complementary Logic Gates Enabled by Two-Dimensional Polarity-Controllable Transistors". In: *ACS Nano* 12.7 (2018), pp. 7039–7047.

[16] G. Resta, A. Leondhart, Y. Balaji, S. De Gendt, P.-E. Gaillardon G. De Micheli. "Devices and Circuits using Novel 2-Dimensional Materials: a Perspective for Future VLSI Systems". In: *IEEE Transaction on Very Large Scale Integration Systems* 27.7 (July 2019).

[17] Q. Herr, A. Herr, O. Oberg and A. Ioannidis. "Ultra-Low Power Superconducting Logic". In: *Journal of Applied Physics* 109 (2011).

[18] M. Dorojevets, Z. Chen, C. Ayala and A. Kasperek. "Towards 32-bit Energy-Efficient Superconductor RQL Processors: The Cell-Level Design and Analysis of Key Processing and On-Chip Storage Units". In: *IEEE Transactions on Applied Superconductivity* 25.3 (June 20115).

[19] K. Likharev and V. Semenov. "RSFQ Logic/Memory Family: A New Josephson-Junction Technology for Sub-terahertz Clock-Frequency Digital Circuits". In: *IEEE Transactions on Applied Superconductivity* 1.3 (Mar. 1991), pp. 3–28.

[20] O.Mukhanov. "Energy-Efficient Single Flux Quantum Technology". In: *IEEE Transactions on Applied Superconductivity* 21.3 (June 2011), pp. 760–769.

[21] M. Tanaka, A. Kitayama, T. Koketsu, M. Ito and A. Fujimaki. "Low-Energy Consumption RSFQ Circuits Driven by Low Voltages". In: *IEEE Transactions on Applied Superconductivity* 23.3 (2013).

[22] T. Filippov, A. Sahu, A. Kirchenko, I. Vernik, M. Dorojevets, C. Ayala and O. Mukhanov. "20 GHz operation of an Asynchronous Wave-Pipelined RSFQ Arithmetic-Logic Unit". In: *Elsevier SciVerse Science Direct, Physics Procedia* 36 (2012).

[23] N. Takeuchi, D. Ozawa, Y. Yamanashi and N. Yoshikawa. "An Adiabatic Quantum Flux Parametron as an Ultra-Low Power Logic Device". In: *Superconductor Science and Technology* 26.3 (2013).

[24] R. Cai, O. Chen, A. Ren, N. Liu, C. Ding, N. Yoshikawa and Y. Wang. "A Majority Logic Synthesis Framework for Adiabatic Quantum-Flux-Parametron Superconducting Circuits". In: *Proceedings of GVLSI*. 2019.

[25] C. Ayala, N. Takeuki, Y. Yamanashi and N. Yoshikawa. "Majority-Logic-Optimizaed Parallel Prefix Carry Look-Ahead Adder Families Using Adiabatic Quantum-Flux-Parametron Logic". In: *IEEE Transactions on Applied Superconductivity* 27.4 (June 2017).

4 HPC Hardware Architectures

Potential long-term impacts of disruptive technologies could concern the processor logic, the processor-memory interface, the memory hierarchy, and future hardware accelerators. We start with potential future memory hierarchies (see Sect. 4.1) including memristor technologies, and the inherent security and privacy issues. Next we look at the processor-memory interface, in particular near- and in-memory computing (see Sect. 4.2). We conclude with future hardware accelerators (see Sect. 4.3) and speculate on future processor logic and new ways of computing (see Sect. 4.4).

4.1 HPC Memory Hierarchies in Systems with NV Memories

4.1.1 Introduction

The Von Neumann architecture assumes the use of central execution units that interface with memory hierarchies of several layers. This model serves as the execution model for more than five decades. Locality of references is a central assumption of the way we design systems. The consequence of this assumption is the need of hierarchically arranged memories.

The memory hierarchy of HPC systems typically consists of thousands of nodes that communicate by message passing. Each node consists of a multi-core processor extended by hardware accelerators (GPUs or FPGAs). The memory hierarchy of a node features three disjoint technological layer types. Closest to the processor cores, we find the cache hierarchy layers that are based on SRAM cells located on the processor chip. Below the cache layer, we find the main memory layer that usually consists of DRAM cells. Eventually, the last layer of the memory hierarchy represents the non-volatile mass storage. Traditionally, this layer was realized using magnetic disk drives. In recent years these drives have been replaced by solid state drives which use Flash memory to store the data. Memory is accessed by linear adresses in chunks of word or cache-line size, mass storage as files.

But this model of a memory hierarchy is not effective in terms of performance for a given power envelop. The main source of inefficiency in the meantime became data movement: the energy cost of fetching a word of data from off-chip DRAM is up to 6400 times higher than operating on it [1]. The current consequence is to move the RAM memory closer to the processor by providing High-Bandwidth Memories.

4.1.2 High-Bandwidth Memory (HBM)

A state-of-the-art memory hierarchy for server-class of computers contains *High-Bandwidth Memory (HBM)* [2] (see Fig. 4.1), which provides higher memory-bandwidths to the cores. Memory is connected in a HBM system via an interposer in the same package with the processor. Memory chips are vertically stacked and connected by TSVs (Through-Silicon Via) with an access logic chip that serves the memory requests of the processor.

HBM provides a tight 3D integration of DRAM memory modules to reduce latency and to increase bandwidth by reducing the energy costs for the data transfer simultaneously.

HBM is based on Die Stacking (see Sect. 3.1.2), which denotes the concept of stacking integrated circuits (e.g. processors and memories) vertically in multiple layers. Die stacking diminishes wire length between memory and logic chips and is applied to three-dimensional DRAM memories, where the bottom layer is active and hosts the physical interface of the memory to the external system. NVIDIA, AMD and Intel apply HBM to exploit the high-bandwidth and low latencies given by 3D stacked memories for a high-dense memory architecture.

3D stacking also enables heterogeneity, by integrating layers, manufactured in different processes, e.g., memristor technologies, which would be incompatible among each other in monolithic circuits. Power

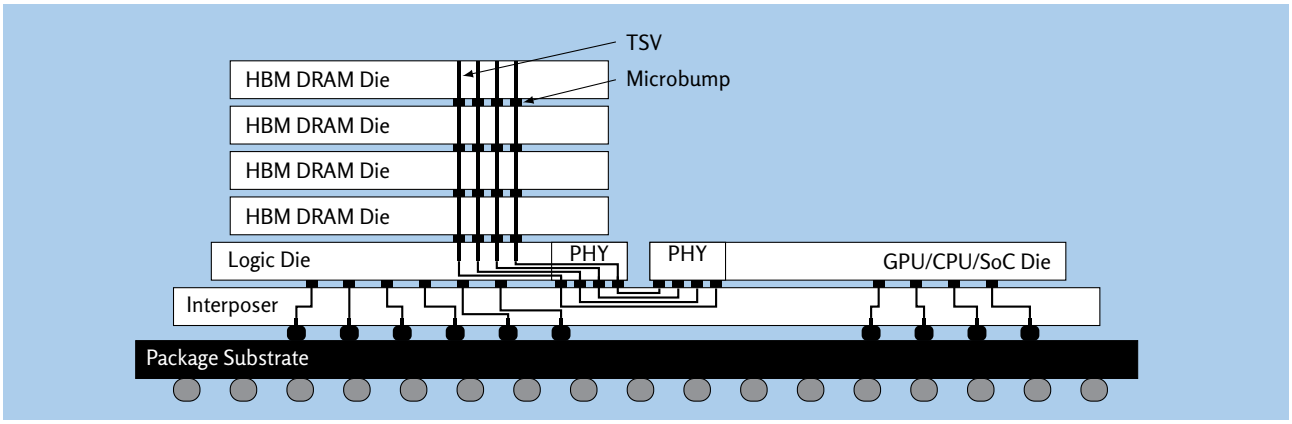


Figure 4.1: High Bandwidth Memory utilizing an active silicon Interposer [2]

consumption is reduced because of the short wire lengths of TSVs and interposers. Simultaneously, a high communication bandwidth between layers can be expected leading to particularly high processor-to-memory bandwidth.

4.1.3 Storage-Class Memory (SCM)

Storage-Class Memory (SCM) currently fills the latency gap between fast and volatile RAM-based memory and slow, but non-volatile disk storage in supercomputers. The gap is currently filled by Flash storage, but could in future be extended by memristive NVM with access times, which are much closer to RAM access times than Flash technology.

In that case memristive NVM based SCM could blur the distinction between memory and storage and require new data access modes and protocols that serve both "memory" and "storage". These new SCM types of non-volatile memory could even be integrated on-chip with the microprocessor cores as they use CMOS-compatible sets of materials and require different device fabrication techniques from Flash. In a VLSI post-processing step they can be integrated on top of the last metal layer, which is often denoted as a back-end of line (BEOL) step (see Sect. 3.2.3).

4.1.4 Potential Memory Hierarchy of Future Supercomputers

Deep Memory Hierarchy

Low-speed non-volatile memories might lead to additional levels in the memory hierarchy to efficiently close the gap between mass-storage and memory as demonstrated by Fig. 4.2 for a potential memory hierarchy of a future supercomputer. Memristors as new types of NV memories can be used in different layers of the memory hierarchy not only in supercomputers but in all kinds of computing devices. Depending on which memory technologies mature, this can have different impacts. Fast non-volatile memories (e.g. STT-RAM) offer the opportunity of merging cache and memory levels. Mid-speed NV memories (e.g. PCM) could be used to merge memory and storage levels.

Shallow Memory Hierarchy

On the other hand, the memory hierarchy might become flatter by merging main memory with storage in particular for smaller systems. Such a shallow memory hierarchy might be useful for future embedded HPC systems. The cache, main memory and mass storage level might be merged to a single level, as shown in Figure 4.3. As a result, the whole system would provide an improved performance, especially in terms of real-time operation. An increased resistance against radiation effects (e.g. bit flips) would be another positive effect. Also, a shallow memory hierarchy would enable applications to use more non-uniform or highly random data access.

4.1.5 Implications

Merging main memory and mass storage allows applications to start much faster. It might be helpful for crash recovery and it can reduce energy consumption as it takes less time to activate/deactivate applications.

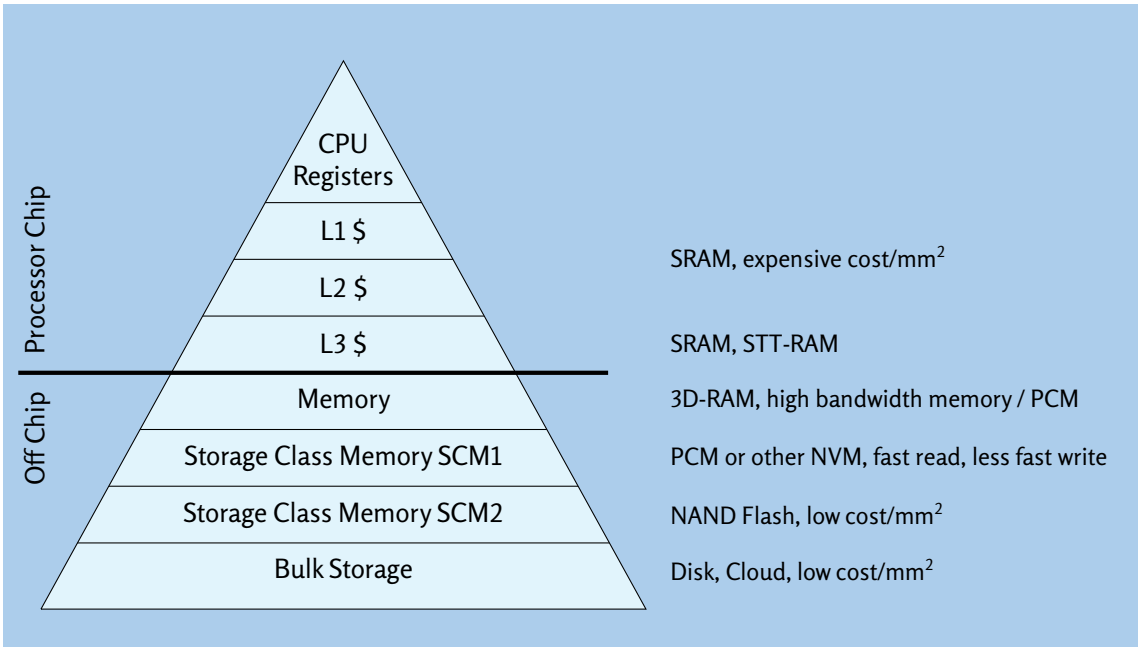


Figure 4.2: Usage of NVM in a future complex supercomputer memory hierarchy.

Programs can be run in intermittent operation (being active for short periods and then stay calm for longer periods) without large overhead. Also, the whole system might be put into standby in a very short time. E.g. if the 2nd / 3rd level of cache is built from NV memory, the processor only needs to purge the 1st or 2nd level of the cache and then the system can be shut off.

However, this also implies security issues (see section 6.3). If data in cache is not lost on power down, this could be exploited to retrieve sensible data from the system.

Also, other problems have to be considered. Realizing such merged levels by NV memory technology might increase the cost to a point where it becomes no longer economically justifiable. The tradeoff between cost and performance has to be well evaluated. The durability and reliability of NV technologies might raise additional problems.

On the other hand, fault tolerance could also be improved by new NV memory concepts. Non volatile memory significantly simplifies checkpointing. If an error is detected, a valid state saved in NM memory could be easily retrieved. Checkpointing could be done on a very fine-grain level. Using so-called in-memory checkpointing, the checkpoint replication would be done automatically for memory to memory operations.

4.1.6 Research Challenges

From the viewpoint of operating systems, modifying the memory hierarchy in one or the other way changes the design of memory maps. Code and data might be shared between processes at different hierarchy levels and might be moved in different ways between levels. Therefore, the operating systems will need to provide new models for program execution and data exchange.

With data stored in non-volatile memory, applications can be active for an arbitrary time. Thus, operating systems must provide different ways to start/stop/deactivate/reactivate and secure programs.

From the viewpoint of computer architecture, changing the memory map has also strong implications for the design of distributed and multi-/many-core systems. Hardware support for memory management in the processor architecture might have to be reconsidered. Different endurance capabilities of different memory levels might demand for new cache and virtual memory replacement strategies. Memory coherence protocols will also be affected. Overall, cache-, memory- and storage interactions on the hardware and OS level will offer research opportunities.

From the viewpoint of application programming, changes in memory hierarchy can modify application models and might improve the behavior of some application classes. If e.g. memory and storage levels

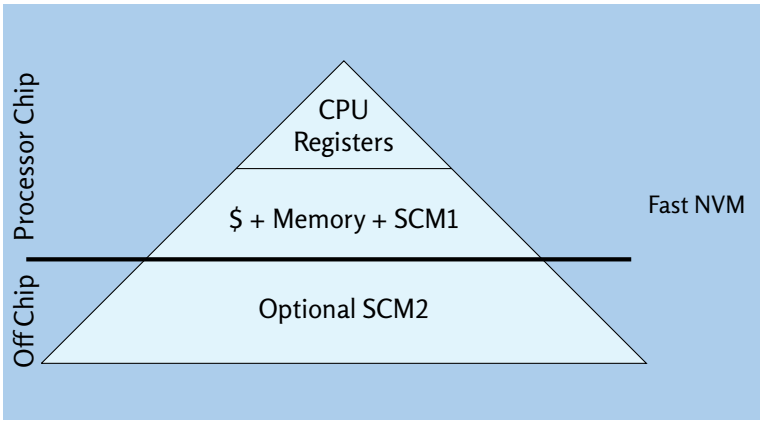


Figure 4.3: Usage of NVM in a future (embedded) HPF systems with shallow memory hierarchy.

are merged, information retrieval applications (e.g. database systems) do no longer need to think in separate categories for storing and memorizing data.. Nevertheless, this will require different structuring of data than usual today.

Security and privacy should be essential design targets for a steadily increasing number of applications, and hardware components play a crucial role in this context. The emergence and propagation of memristors, and in particular memristive NVMs in main memories and caches of computational devices may change established assumptions on their security and privacy properties. It is important to consider security and privacy already in the system conceptualization and design phase as it becomes increasingly difficult to "add security" to a system that has been created without considering such aspects.

A particular need are new possibilities for secure erasure of sensitive data when the device is in operation, which could circumvent the non-volatility of information in cases when it is undesirable. The secure erasure function should be supported on the hardware level that, e.g., overwrites the data designated for deletion (possibly several times) by random data. This problem occurs today when hard disks with sensitive data are reused, but if large parts of the system's memories and caches become non-volatile, the secure erasure would resolve many of the security vulnerabilities mentioned in this chapter. Moreover, a better understanding of the new memory technologies might be useful for the design of Random-Number-Generators (RNGs).

References

[1] M. Drumond, A. Daglis, N. Mirzadeh, D. Ustiugov, J. Picorel, B. Falsafi, B. Grot, and D. Pnevmatikatos. "The Mondrian Data Engine". In: *Proceedings of the 44th Annual International Symposium on Computer Architecture*. ACM. 2017, pp. 639–651.

[2] AMD. *High Bandwidth Memory*. 2017. URL: https://www.amd.com/Documents/High-Bandwidth-Memory-HBM.pdf.

4.2 Near- and In-Memory Computing

As today's architectures and device technologies are facing major challenges (making them incapable to meet the demands of emerging computing applications being extremely demanding in terms of energy and computational efficiency), many alternative computing architectures are being explored in the light of emerging post-CMOS device technologies. The goal is to significantly reduce the data movement, improve parallelism, and reduce the dynamic power and leakage; all of these at economically affordable cost, especially for those devices targeting edge computing (e.g., Gops per 10mW).

In-memory-computing based on memristive devices is one of the potential architectures that can elevate the existing challenges. Next the major difference between traditional architectures and in-memory computing/ near-memory computing will be highlighted. Then near-memory and in-memory computing are further discussed and elaborated; the strong dependency of the targeted application and the selection/design of such architecture will be illustrate. Finally the potential challenges of each of these computing paradigms will be covered.

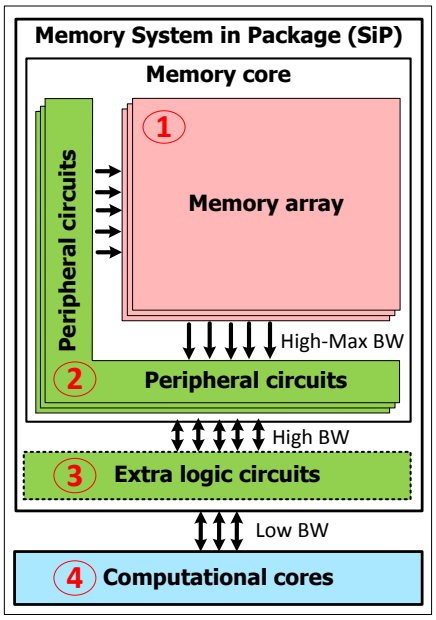


Figure 4.4: Computer Architecture classification

4.2.1 Classification of Computer Architectures

As shown in Fig. 4.4, a computer architecture (or computing system) consists of (one or more) computational units and (one or more) memory cores. A memory core typically consists of one or more cell arrays (used for storage) and memory peripheral circuits (optimised and used to access the memory cells). These memory cores can also integrated with some dedicated logic circuit in a form of System-in-Packages (SiP). Although many criteria can be used to classify computer architectures, *computation location* seems to be the best to use in order to define the different classes in unique manner. Computation location indicates ***where*** the result of the computation is *produced*. Hence, depending on *where* the result of the computation is produced, we can identify four possibilities as Fig. 4.4 shows; they are indicated with four circled numbers and can be grouped into two classes [1].

- *Computation-outside-Memory (COM):* In this class, the computing takes place outside the memory. Hence, the need of data movement. There are two flavours of this class: a) Computation-Outside-Memory Far (COM-F), and b) Computation-Outside-Memory Near (COM-N). COM-F refers to the traditional architectures where the computing takes place in the computational cores such as such as CPU (circle 4 in Fig. 4.4); the memory is seen to be *far* from the processing unit. In order to reduce the length of the communication channel and increase the bandwidth, recent architectures have included computation units with the memory core(s) to form an SiP (circle 3 in Fig. 4.4). Note that the computing is taking place also outside the memory; however, near to it. Hence, the name COM-N or NMC (near-memory computing). An example of such architectures is the Hybrid Memory Cubes (HMC) [2].

- *Computation-Inside-Memory (CIM) of In-Memory Computing (IMC):* in this class, the computing result is produced *within* the memory core, (i.e., the computing takes places within one of the memories). It consists also of two flavours: CIM-Array (CIM-A) and CIM-Periphery (CIM-P). In CIM-A, the computing result is produced within the array (circle 1 in Fig. 4.4), while in CIM-P the result is produced in the memory peripheral circuits (circle 2 in the figure). Examples of CIM-A architectures use memristive logic designs such as MAGIC and imply [3, 4], and examples of examples of CIM-P architectures containing logical bit-wise operations and vector-matrix multiplications [5, 6].

Table 4.1 shows a qualitative comparison of the four architecture sub-classes [1]. Both CIM-A and CIM-P architecture have a relatively low amount of data movement outside the memory core, as the processing occurs inside the memory core. Therefore, they have the potential to alleviate the memory bottleneck. Instead of moving data from the memory to the computational cores, in these architectures the instructions are moved and directly applied to the memory; these instructions typically operate on a large data set, hence a high level of parallelism can be obtained. Data alignment is required for all architectures. However, CIM-A and CIM-P classes perform computations directly on the data residing inside the memory, and hence, the robustness and performance are impacted more by data misalignment. Note that performing a data alignment cannot be handled by host processors in case of CIM architectures due to a far communication distance, while adding additional logic inside the memory core to handle this is also not trivial. Available bandwidth is another important metric. CIM-A architectures may exploit the maximum bandwidth, as operations happen inside the memory array. CIM-P architectures have a bandwidth range from high to max, depending on the complexity of the memory peripheral circuitry. For COM-N, the bandwidth is

	Data Movement outside memory core	Computation requirements		Available bandwith	Memory design efforts			Scalability
		Data Alignment	Complex function		Cells & array	Periphery	Controller	
CIM-A	No	Yes	High latency	Max	High	Low/med.	High	Low
CIM-P	No	Yes	High cost	High-Max	Low/med.	High	Medium	Medium
COM-N (NMC)	Yes	NR	Low cost	High	Low	Low	Low	Medium
COM-F	Yes	NR	Low cost	Low	Low	Low	Low	High

NR: Not Required

Table 4.1: Comparison among Architecture Classes

bounded by on-chip interconnections between the memory core and extra logic circuits; for example, in Hybrid Memory Cube [2] the bandwidth is limited by the number of TSVs and available registers. This bandwidth for TSV is considered high in comparison with COM-F, where the bandwidth is even lower due to off-chip interconnections [7]. Memory design efforts are required to make the computing feasible, especially for CIM. CIM-A architectures require a redesign of the cell, which needs a huge effort. CIM-P architecture require complex read-out circuits as the output value of two or more accessed cells may end up in multiple levels, resulting in large complexity which may limit the scalability. COM-N and COM-F architectures utilize the memory in a conventional way, and hence, standard memory controllers can be used. Note that CIM-A has a low scalability due to several reasons such as the lack/ complexity of interconnect network within the memory array it needs. COM-N has a medium scalability even though the logic layer of memory SiP has more processing resources than peripheral circuits; it cannot accommodate many complex logic units. COM-F has high scalability due to a mature interconnect network and large space for logic devices.

4.2.2 Near Memory Computing NMC of COM-N

Near-memory computing (Near-Memory Processing, NMP) is characterized by processing in proximity of memory to minimize data transfer costs [8]. Compute logic, e.g. small cores, is physically placed close to the memory chips in order to carry out processing steps, like e.g. stencil operations, or vector operations on bulk of data. Near-memory computing can be seen as a co-processor or hardware accelerator. Near-memory computing can be realized by replacing or enhancing the memory controller to be able to perform logic operations on the row buffer. In HBM the Logic Die (see Fig. 4.1) could be enhanced by processing capabilities, and the memory controller can be enabled to perform semantically richer operations than load and store, respectively cache line replacements.

Near-memory computation can provide two main opportunities: (1) reduction in data movement by vicinity to the main storage resulting in reduced memory access latency and energy, (2) higher bandwidth provided by Through Silicon Vias (TSVs) in comparison with the interface to the host limited by the pins [9].

Processing by near-memory computing reduces energy costs and goes along with a reduction of the amount of data to be transferred to the processor. Near-memory computing is to be considered as a near- and mid-term realizable concept.

Proposals for near-memory computing architectures currently don't rely yet on memristor technologies but on innovative memory devices which are commercially available in the meantime such as the Hybrid Memory Cube from Micron [10] [11]. It stacks multiple DRAM dies and a separate layer for a controller which is vertically linked with the DRAM dies. The Smart Memory Cube proposed by [9] is the proposal of a near-memory computing architecture enhancing

the capabilities of the logic die in the Hybrid Memory Cube. The Mondrian Data Engine [12] investigates algorithms of data analytics for near-memory computing.

Such technologies exploit similar to HBM tight 3D integration of DRAM memory modules. However, data buffers and small accelerator cores are attached for near-memory computing in the data path between memory controller and DRAM modules. This forms a near-DRAM acceleration architecture which demonstrated in conceptional investigations for known benchmark suites both speed-up increase and energy decrease compared to non near-DRAM acceleration architectures [13].

Near-memory computing can use available basic memory modules. The challenge is more focused on building new and efficient system architectures. Also the software side is affected, namely new specific instructions have to be created in the instruction sets that consider near-memory computing accelerator instructions [14].

4.2.3 In-Memory Computing (In-Memory Processing, IMP)

In-memory computing (In-Memory Processing, IMP) goes a step further such that the memory cell itself is not only a storage cell but it becomes an integral part of the processing step. This can help to further reduce the energy consumption and the area requirement in comparison to near-memory computing. However, this technology has to be improved and therefore it is considered at least as a mid-term or probably as a more long-term solution.

As already mentioned, CIM (IMC or IMP) can be divided in CIM-A and CIM-P; see Fig. 4.4. For CIM based on memristive devices, as shown in Fig. 4.5, we can further divide the CIM-A and CIM-P classes into two categories. In the first category, *all operands* of the operation are stored in the array, e.g., in the form of resistance. In the second category, *only* part of the operands is stored in the array and the other part is received via the memory port(s). Hence, the logic values of the second category are *hybrid*, e.g., resistive and voltage. If none of the operands is stored in the array, then CIM concept is not applicable as the data is not stored in the same physical location as the computation will take place. The above classification results into four sub-categories as indicated

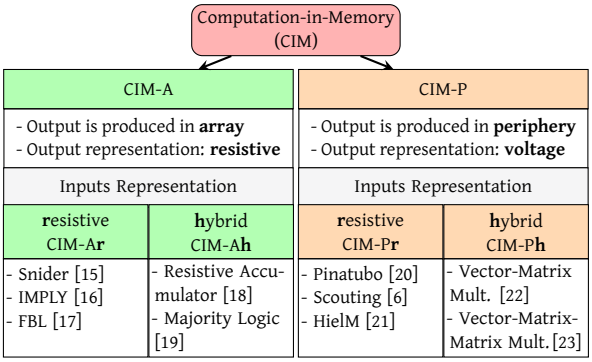


Figure 4.5: CIM circuit classification [24].

in the figure: CIM-Ar, CIM-Ah, CIM-Pr and CIM-Ph; the additional letters 'r' and 'h' indicate the nature of the inputs (operands), namely resistive and hybrid, respectively. The bottom part of the figure shows the existing work for each of the classes.

Developing and designing an appropriate CIM architecture is strongly dependent on the targeted application domain; different applications with different requirements result in different designs; i.e., the circuit design of the array and the periphery. Therefore, to design an efficient CIM circuit, two pieces of information must be extracted from the targeted application: (1) the kernel(s) to accelerate, and (2) the CIM architecture, as shown in Fig. 4.6 [25]:

- Kernel(s): the kernel is the most time/energy consuming function in the targeted application. It dictates the size of the operands, i.e., whether one-bit or multiple-bit numbers. For example, database applications require bitwise logic functions, while compressed sensing requires arithmetic vector-matrix multiplication.
- Architecture: the architecture is mainly related to the location and type of inputs and outputs of the kernel; i.e., the architecture can be CIM-Ar, CIM-Ah, CIM-Pr or CIM-Ph. For example, database applications extract information from a database (stored in the memory) using queries; hence it requires CIM-Pr (or CIM-Ar) architecture. Compressed sensing application converts a sensory signal (i.e., voltage input) using many predefined weights (i.e., the second resistive input) to another signal (i.e., a voltage output); hence, it requires e.g., CIM-Ph architecture.

After analyzing the kernel and suited architecture, the circuit design can start as shown in Fig. 4.6. A CIM circuit can be roughly divided into two parts, i.e., the memory array and the periphery. For the memory array, a suitable memristive technology such

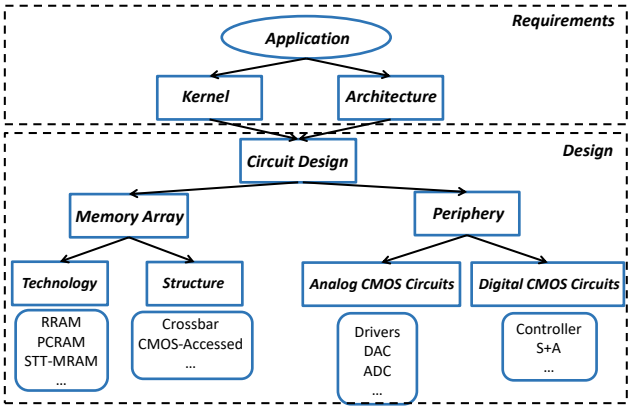


Figure 4.6: CIM design flow.

as RRAM, PCRAM, or STT-MRAM, should be selected based on the requirements of the endurance, resistance variation, etc. Thereafter, the structure of the array should be determined. It could be a crossbar containing only memristive devices or one with additional CMOS transistors that control the access, e.g., the one-transistor-one-memristor (1T1R) structure. For the periphery, the analog components including drivers, digital-analog converters (DACs) and analog-digital converters (ADCs) must be designed based on the needed functionality. In some cases, digital components such as controllers and shift-and-add (S+A) are required as well.

4.2.4 Potential and Challenges for In-memory Computing

In general, memristive device based computing, if successful, will be able to significantly reduce the power consumption and enable massive parallelism; hence, increase computing energy and area efficiency by orders of magnitudes. This may enable new (economically affordable) computing paradigms such as Neuromorphic computing, Artificial neural networks, Bio-inspired neural networks, etc. As memristive device based computing enables computing at the edge (e.g., at the sensors), a lot of application domains can strongly benefit from this computation; examples are IoT devices, wearable devices, wireless sensors, automotive, avionics, etc. Moreover, applications of in-memory computing accelerators could be data intensive applications often categorized as "Big Data" workloads. Such accelerators are especially fitting for data analytics, as they provide immense bandwidth to memory-resident data and dramatically reduce data movement, the main source of energy consumption. Analytic engines for business intelligence are increasingly memory resident to minimize query response time [12]. Graph traversal applications are fitting well due to their unpredictable memory access patterns and high ratio of memory access to computation. Such algorithms are common in social network analysis as e.g. Average Teenage Follower (ATF) that counts for each vertex the number of its teenage followers by iterating over all teenager, in Breadth-First Search (BFS), PageRank (PR), and Bellman-Ford Shortest Path (BF) [9]. In short, if successful, memristive device based computing will enable the computation of currently (economically) infeasible applications, fuelling important societal changes.

Research on memristive device based computing is still in its infancy stage, and the challenges are substantial at all levels, including material/technology, circuit and architecture, and tools and compilers.

- Materials/Technology: At these stage, there are still many open questions and aspects where the technology can help in making memristive device based computing a reality. Examples are device endurance, high resistance ratio between the off and on state of the devices, multi-level storage, precision of analog weight representation, resistance drift, inherent device-to-device and cycle-to-cycle variations, yield issues, exploring 3D chip integration, etc.

- Circuit/Architecture: Analog Computation-in-Memory comes with new challenges to the design of peripheral circuits. Examples are high precision programming of memory elements, relatively stochastic process of analog programming, complexity of signal conversion circuit (digital to analog and analog-to-digital converters), accuracy of measuring (e.g., the current as a metric of the output), scalability of the crossbars and their impact on the accuracy of computing, etc.

- Tools/Compilers: Design automation is still an open question. Profiling and simulation tools can help the user to a) identify the kernels that can be accelerated on memristive device based computing and estimate the benefit, b) perform design exploration to select appropriate device technology/ architecture/ design/ etc. Moreover, design tools can support automatic integration techniques. For example, some memristive technologies can be integrated with CMOS circuits in a so-called BEOL (back end of line) process without costly 3D stacking processes; in this case

the memristive elements are deposited inside holes on the top metal layer and an additional top electrode for the memristive element has to be realized on the top layer while the bottom electrode is realized in the layers beneath. Another approach is the direct integration of memristive behavior directly in MOSFET gate transistors as so-called MemFlash which was demonstrated for neuromorphic memristive cells [26]. For both approaches holds that current design tools do not support automatic integration techniques and simulations of both technologies, CMOS and memristive devices.

As of today, most of the work in the public domain is based on simulations and/or small circuit designs. It is not clear yet when the technology will be mature enough to start commercialization for the first killing applications. Nevertheless, some start-ups on memristor technologies and their application are already emerging; examples are Crossbar, KNOWM, BioInspired, and GrAI One.

References

[1] H.A. Du Nguyen, J. Yu, M. Abu Lebdeh, M. Taouil, S. Hamdioui, F. Catthoor. "A Classification of Memory-Centric Computing". In: *ACM Emerging Technologies in Computing (JETC)* (2020).

[2] J. T. Pawlowski. "Hybrid memory cube (HMC)". In: *Hot Chips 23 Symposium (HCS), 2011 IEEE*. IEEE. 2011, pp. 1–24.

[3] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. "MAGIC–Memristor-aided logic". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 61.11 (2014), pp. 895–899.

[4] K. Kim, S. Shin, and S.-M. Kang. "Stateful logic pipeline architecture". In: *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. IEEE. 2011, pp. 2497–2500.

[5] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie. "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories". In: *DAC*. IEEE. 2016.

[6] L. Xie, H. A. D. Nguyen, J. Yu, A. Kaichouhi, M. Taouil, M. AlFailakawi, and S. Hamdioui. "Scouting Logic: A Novel Memristor-Based Logic Design for Resistive Computing". In: *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2017, pp. 335–340.

[7] W. A. Wulf and S. A. McKee. "Hitting the memory wall: implications of the obvious". In: *ACM SIGARCH computer architecture news* 23.1 (1995), pp. 20–24.

[8] S. Khoram, Y. Zha, J. Zhang, and J. Li. "Challenges and Opportunities: From Near-memory Computing to In-memory Computing". In: *Proceedings of the 2017 ACM on International Symposium on Physical Design*. ACM. 2017, pp. 43–46.

[9] E. Azarkhish, D. Rossi, I. Loi, and L. Benini. "Design and Evaluation of a Processing-in-Memory Architecture for the Smart Memory Cube". In: *Proceedings of the Architecture of Computing Systems – ARCS 2016, Lecture Notes in Computer Science, vol 9637. Springer*. 2016.

[10] AMD. *High Bandwidth Memory*. 2017. URL: https://www.amd.com/Documents/High-Bandwidth-Memory-HBM.pdf.

[11] J. Jeddeloh and B. Keeth. "Hybrid memory cube new DRAM architecture increases density and performance". In: *VLSI Technology (VLSIT), 2012 Symposium on*. IEEE. 2012, pp. 87–88.

[12] M. Drumond, A. Daglis, N. Mirzadeh, D. Ustiugov, J. Picorel, B. Falsafi, B. Grot, and D. Pnevmatikatos. "The Mondrian Data Engine". In: *Proceedings of the 44th Annual International Symposium on Computer Architecture*. ACM. 2017, pp. 639–651.

[13] N. S. Kim, D. Chen, J. Xiong, and W.-m. W. Hwu. "Heterogeneous Computing Meets Near-Memory Acceleration and High-Level Synthesis in the Post-Moore Era". In: *IEEE Micro* 37.4 (2017), pp. 10–18. DOI: 10.1109/MM.2017.3211105. URL: http://ieeexplore.ieee.org/document/8013455/.

[14] J. Ahn, S. Yoo, O. Mutlu, and K. Choi. "PIM-enabled instructions: a low-overhead, locality-aware processing-in-memory architecture". In: 2015, pp. 336–348. DOI: 10.1145/2749469.2750385. URL: http://dl.acm.org/citation.cfm?doid=2749469.2750385.

[15] G. Snider. "Computing with hysteretic resistor crossbars". In: *Applied Physics A* 80.6 (Mar. 2005), pp. 1165–1172. DOI: 10.1007/s00339-004-3149-1. URL: https://link.springer.com/article/10.1007/s00339-004-3149-1.

[16] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. "Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22.10 (Oct. 2014), pp. 2054–2066.

[17] L. Xie, H. A. D. Nguyen, M. Taouil, S. Hamdioui, and K. Bertels. "Fast boolean logic mapped on memristor crossbar". In: *Computer Design (ICCD), 2015 33rd IEEE International Conference on*. Oct. 2015, pp. 335–342.

[18] S. R. Ovshinsky and B. Pashmakov. "Innovation providing new multiple functions in phase-change materials to achieve cognitive computing". In: *MRS Online Proceedings Library Archive* 803 (2003).

[19] P. E. Gaillardon, L. Amarú, A. Siemon, E. Linn, R. Waser, A. Chattopadhyay, and G. De Micheli. "The programmable logic-in-memory (PLiM) computer". In: *Design, Automation & Test in Europe (DATE) Conference*. 2016, pp. 427–432.

[20] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie. "Pinatubo: A Processing-in-memory Architecture for Bulk Bitwise Operations in Emerging Non-volatile Memories". In: *Proceedings of the 53rd Annual Design Automation Conference*. DAC '16. Austin, Texas, 2016, 173:1–173:6.

[21] F. Parveen, Z. He, S. Angizi, and D. Fan. "HielM: Highly flexible in-memory computing using STT MRAM". In: *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*. Jan. 2018, pp. 361–366.

[22] A. Velasquez and S. K. Jha. “Parallel boolean matrix multiplication in linear time using rectifying memristors”. In: *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2016.

[23] A. Velasquez and S. K. Jha. “Computation of boolean matrix chain products in 3D ReRAM”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 2017, pp. 1–4.

[24] M. Abu Lebdeh, U. Reinsalu†, H. A. D. Nguyen, S. Wong, and S. Hamdioui. “Memristive Device Based Circuits for Computation-in-Memory Architectures”. In: *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2019, pp. 1–5.

[25] J. Yu, M. A. Lebdeh, H. Nguyen, M. Taouil, and S. Hamdioui. “The Power of Computation-in-Memory Based on Memristive Devices”. In: *25th Asia and South Pacific Design Automation Conference ASP-DAC*. 2020.

[26] M. Ziegler, M. Oberländer, D. Schroeder, W. Krautschneider, and H. Kohlstedt. “Memristive operation mode of floating gate transistors: A two-terminal MemFlash-cell”. In: *Applied Physics Letters* 101 (Dec. 2012), p. 263504.

4.3 New Hardware Accelerators

Figure 4.7 shows a potential supercomputing node with possible accelerators (not all will be present in a concrete computer) and memories. Such supercomputing nodes could be connected by high-speed networks, potentially based on photonics.

Besides the already common GPU and FPGA accelerators also Neuromorphic Processing Units (NPUs) and Quantum Computing are promising technologies that may be suitable for new hardware accelerators. While GPUs, FPGAs and NPUs may enhance each node of the supercomputer, a single Quantum Computer may be connected to one node or by other means to enhance a supercomputer. Quantum Computing might outperform the whole supercomputer on public-key cryptography, searching, and a number of specialized computing applications.

On the memory side standard DRAM memory may be complemented by memristive NVM memory/storage, as well as near- and in-memory computing devices that combine memory with computing capabilities based on CMOS-logic or future memristive cells. Resistive computing applied in near- or in-memory devices promises a reduction in power consumption and massive parallelism. It could enforce memory-centric by avoiding data movements.

4.4 New Ways of Computing

4.4.1 New Processor Logic

Processor logic could be totally different if materials like graphene, nanotube or diamond would replace classical integrated circuits based on silicon transistors, or could integrate effectively with traditional CMOS technology to overcome its current major limitations like limited clock rates and heat dissipation.

A physical property that these materials share is the high thermal conductivity: Diamonds for instance can be used as a replacement for silicon, allowing diamond based transistors with excellent electrical characteristics. Graphene and nanotubes are highly electrically conductive and could allow a reduced amount of heat generated because of the lower dissipation power, which makes them more energy efficient. With the help of those good properties, less heat in the critical spots would be expected which allows much higher clock rates and highly integrated packages. Whether such new technologies will be suitable for computing in the next decade is very speculative.

Furthermore, Photonics, a technology that uses photons for communication, can be used to replace communication busses to enable a new form of inter- and intra-chip communication.

Current CMOS technology may presumably scale continuously in the next decade, down to 4 or 3 nm. However, scaling CMOS technology leads to steadily increasing costs per transistor, power consumption, and to less reliability. Die stacking could result in 3D many-core microprocessors with reduced intra core wire length, enabling high transfer bandwidths, lower latencies and reduced communication power consumption.

A valuable way to evaluate potential disruptive technologies is to examine their impact on the fundamental assumptions that are made when building a system using current technology, in order to determine whether future technologies have the potential to change these assumptions, and if yes what the impact of that change is.

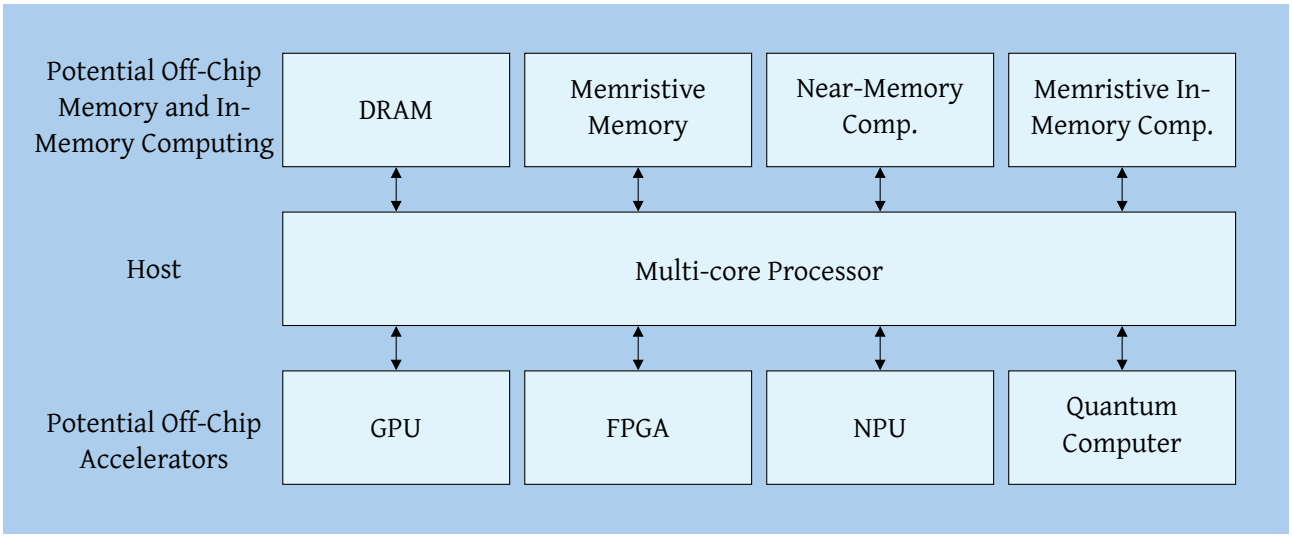


Figure 4.7: Future Architecture of a Supercomputer Node

4.4.2 Power is Most Important when Committing to New Technology

For the last decade, power and thermal management has been of high importance. The entire market focus has moved from achieving better performance through single-thread optimizations, e.g., speculative execution, towards simpler architectures that achieve better performance per watt, provided that vast parallelism exists. The problem with this approach is that it is not always easy to develop parallel programs and moreover, those parallel programs are not always performance portable, meaning that each time the architecture changes, the code may have to be rewritten.

Research on new materials, such as nanotubes and graphene as (partial) replacements for silicon can turn the tables and help to produce chips that could run at much higher frequencies and with that may even use massive speculative techniques to significantly increase the performance of single threaded programs. A change in power density vs. cost per area will have an effect on the likelihood of dark silicon.

The reasons why such technologies are not state of the art yet are their premature state of research, which is still far from fabrication, and the unknown production costs of such high performing chips. But we may assume that in 10 to 20 years the technologies may be mature enough or other such technologies will be discovered.

Going back to improved single thread performance may be very useful for many segments of the market. Reinvestment in this field is essential since it may change the way we are developing and optimizing algorithms and code.

Dark Silicon (i.e. large parts of the chip have to stay idle due to thermal reasons) may not happen when specific new technologies ripen. New software and hardware interfaces will be the key for successfully applying future disruptive technologies.

4.4.3 Locality of References

Locality of references is a central assumption of the way we design systems. The consequence of this assumption is the need of hierarchically arranged memories, 3D stacking and more.

But new technologies, including optical networks on die and Terahertz based connections, may reduce the need for preserving locality, since the differences in access time and energy costs to local memory vs. remote storage or memory may not be as significant in future as it is today.

When such new technologies find their practical use, we can expect a massive change in the way we are building hardware and software systems and are organizing software structures.

The restriction here is purely the technology, but with all the companies and universities that work on this problem, we may consider it as lifted in the future.

4.4.4 Digital and Analog Computation

The way how today's computers are built is based on the digital world. This allows the user to get accurate results, but with the drawbacks of cost of time, energy consumption and loss of performance. But accurate results are not always needed. Due to this limitation the production of more efficient execution units, based on analog or even a mix between analog and digital technologies could be possible. Such an approach can revolutionize the way of the programming and usage of future systems.

Currently the main problem is, that we have no effective way to reason at run time on the amount of inaccuracy we introduces to a system.

4.4.5 End of Von Neumann Architecture

The Von Neumann architecture assumes the use of central execution units that interface with different layers of memory hierarchies. This model, serves as the execution model for more than three decades. But this model is not effective in terms of performance for a given power.

New technologies like memristors may allow an on-chip integration of memory which in turn grants a very tightly coupled communication between memory and processing unit.

Assuming that these technologies will be mature, we could change algorithms and data structures to fit the new design and thus allow memory-heavy "in-memory" computing algorithms to achieve significantly better performance.

We may need to replace the notion of general purpose computing with clusters of specialized compute solution. Accelerators will be "application class" based, e.g. for deep learning, molecular dynamics, or other important domains. It is important to understand the usage model in order to understand future architectures/systems.

4.4.6 Summary of Potential Long-Term Impacts of Disruptive Technologies for HPC Software and Applications

New technologies will lead to new hardware structures with demands on system software and programming environment and also opportunities for new applications.

CMOS scaling will require system software to deal with higher fault rate and less reliability. Also programming environment and algorithms may be affected, e.g., leading to specifically adapted approximate computing algorithms.

The most obvious change will result from changes in memory technology. NVM will prevail independent of the specific memristor technology that will win. The envisioned Storage-Class Memory (SCM) will influence system software and programming environments in several ways:

- Memory and storage will be accessed in a uniform way.
- Computing will be memory-centric.
- Faster memory accesses by the combination of NVM and photonics could lead either to an even more complex or to a shallower memory hierarchy envisioning a flat memory where latency does not matter anymore.
- Read accesses will be faster than write accesses, though, software needs to deal with the read-/write disparity, e.g., by database algorithms that favour more reads over writes.
- NVM will allow in-memory checkpointing, i.e. checkpoint replication with memory to memory operations.
- Software and hardware needs to deal with limited endurance of NVM memory.

A lot of open research questions arise from these changes for software.

Full 3D stacking may pose further requirements to system software and programming environments:

- The higher throughput and lower memory latency when stacking memory on top of processing may require changes in programming environments and application algorithms.
- Stacking specialized (e.g. analog) hardware on top of processing and memory elements lead to new (embedded) high-performance applications.
- Stacking hardware accelerators together with processing and memory elements require programming environment and algorithmic changes.

- 3D multicores require software optimizations able to efficiently utilize the characteristics of 3rd dimension, .i.e. e.g., different latencies and throughput for vertical versus horizontal interconnects.
- 3D stacking may to new form factors that allow for new (embedded) high-performance applications.

Photonics will be used to speed up all kind of interconnects – layer to layer, chip to chip, board to board, and compartment to compartment with impacts on system software, programming environments and applications such that:

- A flatter memory hierarchy could be reached (combined with 3D stacking and NVM) requiring software changes for efficiency redefining what is local in future.
- It is mentioned that energy-efficient Fourier-based computation is possible as proposed in the Optalysys project.
- The intrinsic end-to-end nature of an efficient optical channel will favour broadcast/multicast based communication and algorithms.
- A full photonic chip will totally change software in a currently rarely investigated manner.

A number of new technologies will lead to new accelerators. We envision programming environments that allow defining accelerator parts of an algorithm independent of the accelerator itself. OpenCL and OpenACC are such languages distinguishing "general purpose" computing parts and accelerator parts of an algorithm, where the accelerator part can be compiled to GPUs, FPGAs, or many-cores like the Xeon Phi. Such programming environment techniques and compilers have to be enhanced to improve performance portability and to deal with potentially new accelerators as, e.g., neuromorphic chips, quantum computers, in-memory resistive computing devices etc. System software has to deal with these new possibilities and map computing parts to the right accelerator.

Neuromorphic Computing is particularly attractive for applying artificial neural network and deep learning algorithms in those domains where, at present, humans outperform any currently available high-performance computer, e.g., in areas like vision, auditory perception, or sensory motor-control. Neural information processing is expected to have a wide applicability in areas that require a high degree of flexibility and the ability to operate in uncertain environments where information usually is partial, fuzzy, or even contradictory. It is envisioned that neuromorphic computing could help understanding the multi-level structure and function of the brain and even reach an electronic replication of the human brain at least in some areas such as perception and vision.

Quantum Computing potentially solves problems impossible by classical computing, but posts challenges to compiler and runtime support. Moreover, quantum error correction is needed due to high error rates.

Resistive Computing may lead to massive parallel computing based on data-centric and reconfigurable computing paradigms. In memory computing algorithms may be executed on specialised resistive computing accelerators.

Quantum Computing, Resistive Computing as well as Graphene and Nanotube-based computing are still highly speculative hardware technologies.

Open Questions and Research Challenges

The discussion above leads to the following principal questions und research challenges for future HPC hardware architectures and implicitly for software and applications as well:

- Impact, if power and thermal will not be limiter anymore (frequency increase vs. many-cores)?
- Impact, if Dark Silicon can be avoided?
- Impact, if communication becomes so fast so locality will not matter?
- Impact, if data movement could be eliminated (and so data locality)?
- Impact, if memory and I/O could be unified and efficiently be managed?

5 System Software and Programming Environment

5.1 Accelerator Ecosystem Interfaces

The slowdown in silicon scaling, and the emergence of heterogeneous logic and memory technologies have led to innovation in interface technologies to connect various components of an HPC platform node together, providing unified protected abstractions to access memory as well as access to the network and storage. Example consortia in recent years that have emerged and provide a variety of connectivity among heterogeneous components and peripherals are CCIX [1], Gen-Z [2], CAPI [3] and NVLink [4]. These interface technologies vary in compatablity among each other, also by the degree of compatibility with legacy interfaces (e.g., PCIe), and whether they support hardware cache coherence and conventional memory abstractions.

A key challenge in supporting accelerator-rich environments in future platforms will be supporting higher-level software abstractions in hardware that would not only enable protected seamless sharing of memory among near-neighbor components but also allow accelerators offering services over the network which are coordinated by a host CPU but with the host CPU and OS outside the critical path of computationg and communication across nodes. Microsoft Catapult [5] placing FPGA's directly on the network to enable communication with other FPGA's across the network without the host in the way.

Another key challenge in future accelerator-rich environments is moving away from virtual memory and paging abstractions for protected access. Conventional OS abstractions for address translation and their architectural support in modern platforms date back to desktop PC's of the 80's, and are already at limits for Terabyte-scale memory nodes requiring tens of thousands of TLB entries in hierarchies per core. Many important classes of emerging accelerators are limited in efficiency and performance by data movement and require protected access to memory that can reach orders of magnitude more capacity than conventional address translation can support. Recent techniques to reduce fragmentation in address translation through segmentation [6] or coalescing [7] are promising. With emerging memory technologies, novel abstractions for isolation, protection and security are needed that lend themselves well to efficient hardware support and enable a continued scaling in memory capacity in an accelerator-rich environment.

5.2 Integration of Network and Storage

Modern HPC platforms are based on commodity server components to benefit from economies of scale and primarily differ from datacenters in that they incorporate cutting-edge network fabrics and interfaces. The canonical blade server architecture finds its roots in the desktop PC of the 80's with the CPU (e.g., x86 sockets) managing memory at hardware speed and the OS (e.g., Linux) moving data between the memory and storage/network over legacy I/O interfaces (e.g., PCIe) in software.

Unfortunatetly, the legacy OS abstractions to communicate with the network interface and storage are a bottleneck in today's system and will be a fundamental challenge in future HPC platforms. Because the network/storage controllers can not access the host memory directly, recent years have seen a plathora of technologies that integrate private memory and logic closer to network/storage controllers to add intelligence to services but result in a major fragmentation of silicon in the platform across I/O interfaces and do not fundamentally address the legacy interface bottleneck. For example the latest HPC flash array controllers or network interface cards [8] integrate 32 out-of-order ARM cores that can reach tens of GB of private memory and can directly talk to PCIe-based accelerators.

The emerging consortia for new interfaces (Section 5.1) help with a closer coordination of hardware components not just between the host CPU and accelerators but also with the network. Future interfaces will

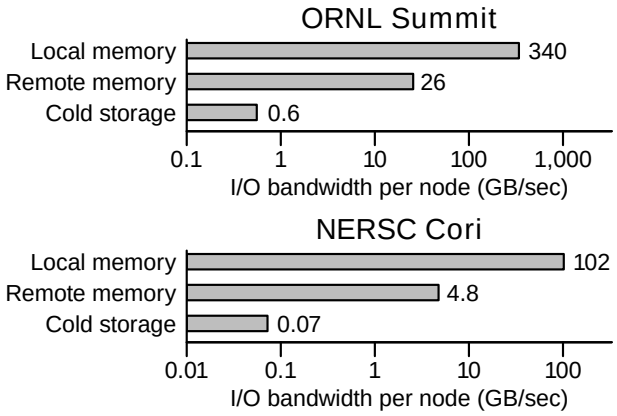


Figure 5.1: Many scientific applications are increasingly becoming I/O bound, as the bandwidth to cold storage is 100×—1000× less than the bandwidth to memory. Per-node bandwidth to local memory, remote memory and cold storage is shown for Summit and Cori, #1 and #13 computers in TOP500, respectively.

help integrate the network and storage controllers together with the host. This integration requires novel OS abstractions beyond conventional address translation and paging mechanisms to enable low-overhead protected movement of data among not just the host and accelerators but also together with the network and storage.

5.3 Data Management

More than a decade ago, the pioneering computer scientist Jim Gray envisioned a fourth paradigm of scientific discovery that uses computers to solve data-intensive scientific problems [9]. Today, scientific discovery, whether observational, in-silicon or experimental, requires sifting through and analyzing complex, large datasets. For example, plasma simulation simulates billions of particles in a single run, but analyzing the results requires sifting through a single frame (a multi-dimensional array) that is more than 50 TiB big—and that is for only one timestep of a much longer simulation [10]. Similarly, modern observation instruments also produce large datasets—a two-photon imaging of a mouse brain yields up to 100 GiB of spatiotemporal data per hour [11] and electrocorticography (ECoG) recordings yield 280 GiB per hour [12].

Although scientific computing continuously shatters records for floating point performance, the I/O capabilities of scientific computers significantly lag on-premise datacenters and cloud computing. As shown in Figure 5.1, while the bandwidth to local memory and to other nodes is many GB/sec, the bandwidth to cold storage is less than 1 GB/sec per node [13]. Making matters worse, reaching this level of I/O performance assumes large, sequential I/Os. Many modern scientific applications, however, do not follow this I/O pattern. For example, the inference step of many ML-based data processing pipelines is often I/O bound, as object classification in scientific applications is often performed on millions of KB-sized objects [14]. Given the exponential growth trend in data volumes, the bottleneck for many scientific applications is no longer floating point operations per second (FLOPS) but I/O operations per second (IOPS).

The computational side of the scientific computing is undergoing a rapid transformation to embrace such array-centric computing all the way from applications (e.g. TensorFlow, PyTorch, Theano, DSSTNE) to the hardware (e.g. TPUs, custom-designed ASICs for tensor processing). However, the POSIX I/O interface to cold data (e.g. Lustre, GPFS, or the limited Hadoop DFS) remains largely agnostic to the I/O optimization opportunities that array-centric computing presents. This simplistic view of I/O for array-centric analyses is challenged by the dramatic changes in hardware and the diverse application and analytics needs in today's large-scale computers: On the hardware front, new storage devices such as SSDs and NVMe not only are much faster than traditional devices, but also provide a different performance profile between sequential and random I/Os. On the application front, scientific applications have become much more versatile in both access patterns and requirements for data collection, analysis, inference and curation. Both call for more flexible and efficient I/O abstractions for manipulating scientific data.

The Rising Prominence of File Format Libraries

Traditionally, scientific applications were opaque when it comes to optimizing data representation, placement and access. Individual experiments often stored and processed data in custom binary data representations that were tailored to the specific problem, were non-portable across computers, and lacked any documentation. Unlike the field "big data" processing and the dominance of the Hadoop/Spark ecosystem for data processing, "big science" has not converged to a single data representation and analysis platform,

and is unlikely to do so in the future. This lack of convergence has been attributed to externalities, including the significant upfront effort to use new software, the lack of computer science expertise that often manifests as underestimating the difficulty of fundamental data management problems, and the limited ability to provide long-term support for scientific software due to the transitory nature of the team members (largely PhD students or post-docs) who are developing new scientific software [15].

As a consequence, scientists often try to emulate the sophisticated I/O optimizations that a semantics-aware runtime (such as a database system) performs automatically by following a simple "place data sequentially" mantra: scientists need to know what data are going to be accessed together to be placed adjacently, while the system ensures that accesses to sequential data are fast by using optimization techniques such as I/O batching and prefetching. This I/O optimization model is overly simplistic given the increasing heterogeneity of the storage stack—comprising of burst buffers, all-flash storage, and non-volatile memory.

The rapid adoption of new storage media, has shifted the perception on using third-party I/O libraries for data storage and access. A 2014 user survey of the NERSC scientific facility shows that the HDF5 and NetCDF file format libraries are among the most widely used libraries, and have at least as many users as popular numerical libraries, such as BLAS, ScaLAPACK, MKL and fftw [16]. In 2018, 17 out of the 22 projects in the U.S. Department of Energy Exascale Computing Project (ECP) portfolio were using HDF5 [17]. These file format libraries present a semantically richer, array-centric data access interface to scientific applications. Using such file format libraries allows applications to navigate through a hierarchical organization of datasets, explore rich metadata about each dataset, choose subsets of an array, perform strided accesses, transform data, make value-based accesses, and automatically compress sparse datasets to save space.

Many research efforts seek to bring richer data management functionality in file format libraries and largely explore three complementary avenues: (1) extending the functionality of file format libraries, (2) developing connectors and (3) automatically migrating data. Focusing on the HDF5 format, the ExaHDF5 project seeks to extend the HDF5 file format library with user-defined functionality. ExaHDF5 has developed the virtual object layer (VOL) feature which permits system builders to intercept and respond to HDF5 I/O calls [18]. ArrayBridge is a connector that allows SciDB to directly query HDF5 data without the onerous loading step and produce massive HDF5 datasets using parallel I/O from within SciDB [19]. Automatic migration between data storage targets has been investigated for HDF5 with the Data Elevator that transparently moves datasets to different storage locations, such as node-local persistent memory, burst buffers, flash, disks and tape-based archival storage [20]. Looking ahead, we envision that efforts to bring data management functionality inside file format libraries will be sustained to create a vibrant ecosystem of data analysis tools for accelerated I/O performance.

Rethinking Data Storage

Storage technology providers are quickly innovating to reduce latency and significantly improve performance for today's cutting-edge applications. I/O optimizations such as coalescing, buffering, prefetching and aggregation optimize accesses to block-based devices, but are (at best) irrelevant for next-generation storage devices or (at worst) they incur unnecessary overhead if used. New storage technologies will thus reach their full potential only if they reduce I/O stack overheads with direct user-mode access to hardware. Today applications can directly interact with new storage hardware by using libraries such as the persistent memory development kit (PMDK) and the storage performance development kit (SPDK). In the meantime, storage interfaces such as NVMe are being extended to support datacenter-scale connectivity with NVMe over Fabrics (NVMe-OF), which ensures that the network itself will not be the bottleneck for tomorrow's solid state technologies. This motivates a reconsideration of the data storage architecture of datacenter-scale computers for science.

An idea that is getting significant application traction is a transition away from block-based, POSIX-compliant file systems towards scalable, transactional object stores. The Intel Distributed Asynchronous Object Storage (DAOS) project is one such effort to reinvent the exascale storage stack [21]. DAOS is an open source software-defined object store that provides high bandwidth, low latency and high I/O operations per second. DAOS aggregates multiple storage devices

(targets) in pools, which are also the unit of redundancy. The main storage entity in DAOS is a container, which is an object address space inside a specific pool. Containers support different schemata, including a filesystem, a key/value store, a database table, an array and a graph. Each container is associated with metadata that describe the expected access pattern (read-only, read-mostly, read/write), the desired redundancy mechanism (replication or erasure code), and the desired striping policy. Proprietary object-store solutions, such as the WekaIO Matrix, are also competing for market share in the same space. In these new storage designs, POSIX is no longer the foundation of the data model. Instead, POSIX interfaces are built as library interfaces on top of the storage stack, like any other I/O middleware.

Many challenges remain in how applications interact with persistent storage. One point of significant friction has been how applications make writes durable in case of failure. Block-based interfaces rely on page-level system calls, such as `fsync` and `msync`, however flushing an entire page is too coarse-grained for byte-addressable non-volatile storage devices. Flushing at a finer granularity is possible with user-space instructions such as `CLFLUSH` that flush at a cache line granularity and avoid an expensive system call. However, cache line flushing is not a panacea, as it evicts lines from the cache. This means that accessing a location immediately after flushing it will cause a cache miss, which doubles the cost of a store. Looking ahead, a more elegant solution would be to extend the power failure protection domain to include the cache hierarchy, but implementing such a feature requires careful vertical integration between the application, the CPU and the memory subsystem.

Open Questions and Research Challenges

Future HPC architectures will need to address the following research challenges to efficiently managing large scientific datasets:

- As simulation, data processing and machine learning converge, a new benchmark needs to be developed to measure I/O performance. What are representative workloads for measuring I/O performance?
- How should I/O metadata be acquired, stored and analyzed to convey scientific intent (intensional metadata) and performance bottlenecks (performance metadata)? Instead of a test-and-see approach, can extreme-scale I/O be optimized systematically from I/O metadata?
- How can one use I/O behaviors during testing at small scale to predict and ameliorate bottlenecks in production at large scale?
- How should massive multi-dimensional datasets be chunked and placed in heterogeneous storage targets? How can stratification be leveraged to accelerate deterministic approximate query processing?
- What are the opportunities for hardware acceleration for simple data transformations, such as converting from column-major order to row-major order, that are notoriously inefficient? How can hardware acceleration be leveraged for more elaborate I/O patterns, such as bitweaving, data-dependent record skipping, co-analysis and delta encoding?
- How should smart storage devices be architected? How can they expose application-level semantics (indexed access, array subseting, object references) through new storage device interfaces such as NVMe over fabrics (NVMe-OF)?

5.4 Single-Source Programming Models for Heterogeneity

5.4.1 Introduction

In the last decade, clock frequency scaling in processors has substantially stopped and it is no more a crucial approach to seek performance. Since then, Moore's law, seen from the performance progression standpoint, has been respected as the *potential aggregate performance* of the increasing number of cores available on-chip. However, translating this *potential* into application performance passes through a hard path: various flavors of parallel programming approaches need to be mastered by the developers, and this fact substantially increases the complexity of coding [22], debugging and testing. A number of different strategies, frameworks, and libraries have been proposed over the years, and none of them can be considered a consolidated general solution for the current parallel architectures.

The majority of current top HPC systems get their aggregate performance from multi-core processors, from their SIMD extensions and, in many cases, from many-core modules like GPUs and accelerators. The latter are able to make tousands of cores available to programmers in the same physical board equipped with tens GByte of RAM. GPGPU coding is different from multi-core coding and, typically, very complex due to the extreme parallelism of computational/memory resources, their organization, distribution, and interconnection, which reduces productivity and/or achieved performance. GPU manufacturers are trying to simplify programming through libraries and powerful tools such as NVIDIA CUDA C++ [23] or AMD Brook+ [24]. These tools come from the same companies that produce the graphic cards and therefore induce code that is not portable across GPUs from different vendors.

Developing an application or implementing an algorithm for both multi-core CPUs and GPUs can be quite common in order to support the deployment on different platforms, transparent performance scaling, and efficiency (e.g. in workstations, dedicated or mobile devices, datacenters and HPC systems). On top of this, especially in HPC systems applications need to be potentially distributed on a huge number of physical nodes to reach the required aggregate performance. And this facet typically needs to be explicitly managed by the programmer (e.g., via MPI), who is exposed to details of the physical organization of the overall machine.

Focusing on a single node, programming a multi-core CPU is very different from programming a GPU and this dramatically reduces code portability and, in the meanwhile, also performance portability in current and future parallel architectures. Heterogeneous architectures, featuring both architectures, suffer from this situation in a similar way, and with even worse effects as they could benefit from a coordinated execution on its heterogeneous resources.

5.4.2 Single-Source Approaches

A number of proposals exist, namely frameworks and strategies, that allow programmers writing cross-platform code, maintain smaller codebases, and respect the (*Don't Repeat Yourself*-principle [25]). However, we still miss a definitive solution providing an effective cross-platform parallel programming approach for CPUs and GPUs, and maybe FPGAs, as well as multi-node machines, while providing a significant higher level of abstraction and expressiveness compared to the respective native programming approaches.

Cross-platform heterogeneous programming approaches can be classified according to various criterions. Here we focus on a clustering based on the induced coding macro-features and we define: CUDA-like frameworks, compiler-based approaches, and high-level wrappers and we try to do a best-match assignment even if each approach can have some hybrid traits. CUDA-like models explicitly distinguish between the host code, coordinating the execution, and the so-called kernel functions to be launched on the devices in parallel. The only example of cross-platform CUDA-like framework is OpenCL [26], which is quite low-level. The solutions in the *compiler-based* category rely on compilers to generate device parallel code. Some examples are OpenACC [27], OpenMP 4 [28], OmpSS [29], C++ AMP [30], SYCL [31], and PACXX [32].

high-level wrappers comprise high-level libraries that wrap one or more approaches lying in the first two categories. This way, some low-level details are shielded from the programmer and managed by the inner layers of the library. Some examples are ArrayFire [33], SkePU [34], SkelCL [35], Kokkos [36], Boost.Compute [37] and PHAST [38].

OpenCL [26] is a C99 (and C++ from version 2.0) extension that follows CUDA C++'s structure and expressiveness. Unlike CUDA, it offers code portability, but it has more verbose setup code and a non single-source code organization. CUDA-like approaches like this require explicit data transfers between platforms and specification of kernel launch parameters, and needs to manage many low-level details to seek performance. For this reason, they are not the best from the programmer productivity point-of-view.

Two examples of compiler-based heterogeneous framework are OpenACC [27], OpenMP 4 [28] and OmpSS [29]. They are pragma-based solutions that require the code to be properly annotated to be parallelized. They have the great value of not requiring code re-writing, but they also need their users to learn a new language to be paired with their C/C++ code: the annotation code. C++ AMP [30] is an open extension to C++11 provided with a Microsoft implementation, based on `array_view` data structures, processed via `parallel_for_each` invocations with a callable argument.

SYCL [31] is a royalty-free C++14 abstraction layer that builds on top of OpenCL concepts. It permits writing single-source programs with kernel functions expressed as lambdas. It requires programmers to express their program in terms of buffers, queues, and accessors. It also features a Parallel STL that is now at an experimental stage. PACXX [32] is a unified programming model implemented as a custom C++14 Clang-based compiler. Developers can express parallel computation on `std::vectors` via `std::async` or annotating function invocations. These compiler-based solutions, despite permitting users to write concise single-source code, also require them to adopt non-native compilers. This can be sometimes a sub-optimal choice, since native compilers produce, in general, better code and are continuously improved.

ArrayFire [33] is an array-centric high-level library that wraps CUDA and OpenCL. It defines an `array` class and functions that express computations via a math-resembling syntax. SkePU [34] and SkelCL [35] are C++ template libraries based on common algorithmic patterns called *skeletons*. SkePU has been implemented as a library on top of CUDA, OpenCL, and OpenMP, whereas SkelCL is an OpenCL wrapper. They both define a custom `Vector` class, various common algorithmic skeletons in the form of classes and support user-defined functions.

Kokkos [36] is a high-level C++ library that wraps CUDA, PThreads, and OpenMP. It defines multi-dimensional arrays accessible via views and support user-defined functors or lambdas in `parallel_for` or `parallel_reduce` algorithms. Boost.Compute [37] is a C++ wrapper around OpenCL API, based on containers, algorithms, and iterators. Programmers can express custom functions to be used in transform-like algorithms using strings à-la OpenCL or using macros. These functions are compiled at run-time, but can also be cached and loaded from file-system. PHAST [38] is a single-source data-parallel library based on multi-dimensional containers, iterators and STL-like algorithms operating in parallel on container slices, allowing to express STL-like algorithms also in the *kernel/device* code. It can target applications to both multi-cores and NVIDIA GPUs and features a task model and task-DAG support, where each task can be data-parallel and assigned, statically or dynamically, to the CPU or the GPU. PHAST typically requires the simplest code compared to OpenCL, Kokkos and SYCL according to various code complexity metrics.

oneAPI is based on the Sycl approach, using c++ templates to have a 3-way compilation, allowing for automatic host, accelerator and interface code generation. It is advancing in the direction of programmer-guided parallelization, but it is still leaving too many details to the programmer, like manually defining the interfaces between the host and the accelerator. In this way, the programmer is required to indicate the parameters that will be need to be copied to the accelerator. This is still low level programming, compared to higher level programming models like OpenMP, where the interface definition between host and accelerator code is made by the compiler, given the shared/private/firstprivate/map clauses using in the compiler directives.

High-level wrappers can take the best of both worlds: high-level interfaces and efficient inner layers of the framework.

Other approaches, and some parts of the ones already cited, aim at tackling the multi-node scenario, which is typical in HPC machines. For instance rCuda [39] which is a middleware software framework based on MPI for remote GPU virtualization, libWater [40], which addresses heterogeneous targets through MPI, and SnuCL [41], which addresses heterogeneity in clusters via OpenCL and MPI. Also some skeleton-based approaches like FastFlow [42] are able to support multi-node scenarios and heterogeneous architectures.

5.4.3 Hiding Hardware Complexity

Hiding or mitigating this increasingly complex and varied hardware requires more and more intelligence across the programming environment. Manual optimization of the data layout, placement, and caching will become uneconomic and time consuming, and will, in any case, soon exceed the abilities of the best human programmers. There needs to be a change in mentality from programming "heroism" towards trusting the compiler and runtime system (as in the move from assembler to C/Fortran). Automatic optimization requires advanced techniques in the compiler and runtime system. In the compiler, there is opportunity for both fully automated transformations and the replacement of manual refactoring by automated program transformations under the direction of human programmers (e.g. Halide [43] or DaCe [44]). Advanced runtime and system software techniques, e.g., task scheduling, load balancing, malleability, caching, energy proportionality are needed.

Increasing complexity also requires an evolution of the incumbent standards such as OpenMP, in order to provide the right programming abstractions. There is as yet no standard language for GPU-style accelerators (CUDA is controlled and only well supported by a single vendor and OpenCL provides portability). Domain-specific languages (e.g. for partial differential equations, linear algebra or stencil computations) allow programmers to describe the problem in terms much closer to the original scientific problem, and they provide greater opportunities for automatic optimization. In general there is a need to raise the level of abstraction. In some domains (e.g. embedded) prototyping is already done in a high-level environment similar to a DSL (Matlab), but the implementation still needs to be ported to a more efficient language.

There is a need for global optimization across all levels of the software stack, including OS, runtime system, application libraries, and application. Examples of global problems that span multiple levels of the software stack include a) support for resiliency (system/application-level checkpointing), b) data management transformations, such as data placement in the memory hierarchy, c) minimising energy (sleeping and controlling DVFS), d) constraining peak power consumption or thermal dissipation, and e) load balancing. Different software levels have different levels of information, and must cooperate to achieve a common objective subject to common constraints, rather than competing or becoming unstable.

5.4.4 Conclusions

Overall, the described multitude of approaches aiming to abstract the parallelism of modern architectures, both at smaller grain and nature (e.g., multicores in CPUs vs many-codes in GPUs), as well as at larger scale, considering distributed multi-node machines, highlight the importance of the topic and its critical role in enabling a seamless and simpler code portability across different architectures. Furthermore, these experiences show that many significant steps have been done and that a number of tools already solve a number of portability problems but, unfortunately, this happens in specific domains. In fact, each approach cannot be easily extended to comprise the full spectrum of applications and hardware heterogeneity available in modern and future systems, especially HPC systems. Therefore, it is crucial that intense research and development efforts are peformed in the strategic direction of easing code portability, performance portability and programmer's productivity, regarding the applications that we will be running on the next generation of heterogeneous HPC machines.

5.5 Performance Models

Models are necessary to understand how performance and energy consumption depend on the resources that are allocated to a job. Resources include the number of nodes, memory system (choice of memory devices and their capacities), and compute accelerators. Such models may be used directly by the user, e.g. to select the resources to give a job or to demonstrate that an application has sufficient scalability to merit execution on a supercomputer, or they may be employed by runtime systems, compilers or job schedulers to automatically allocate, and potentially re-allocate, resources to a job.

Both analytical models (e.g LogP) and coarse-grain simulation (e.g. Dimemas) have been employed for some time. Nevertheless, there are many challenges, relating to system size, system complexity (memory types, accelerators and heterogeneity, etc.) and software complexity (e.g. layers of abstraction, runtime and compiler intelligence, malleability, storage, I/O, workflows). Moreover, these models need to be integrated into toolchains and should be interoperable with programming models.

Analytical models typically benefit from execution speed and, since they employ measurements and performance counters on a real system, they may be more accurate when the precise behaviour is unknown or intractable to model exactly [45]. Promising directions for analytical models include PROFET [45], which predicts performance, energy and power on different memory systems (different devices and/or clock frequencies) based on hardware performance counters measured on a baseline memory system, and Extra-P [46], which extrapolates the performance of each kernel or routine in the application, and can be used to derive scalability and/or isolate unexpected scalability bottlenecks. Such approaches should be extended and integrated into compilers, runtimes and performance analysis tools.

Simulation-based approaches vary in level of abstraction, from those based on MPI-level communication

(e.g. Dimemas) to those based on detailed architectural simulation (e.g. gem5). Hierarchical approaches, e.g. MUSA [47], based on sampling with varying levels of detail are most likely to be appropriate for future exascale systems.

Another approach, taken for example in the Hi-EST project uses machine learning to derive a performance model of applications competing for shared resources [48]. Given the impressive advances in machine learning in recent years that is likely to be a fruitful direction for future research.

5.6 Complex Application Performance Analysis and Debugging

Performance analysis and debugging are particularly difficult problems beyond Exascale. The problems are two-fold. The first problem is the enormous number of concurrent threads of execution (millions), which provides a scalability challenge (particularly in performance tools, which must not unduly affect the original performance) and in any case there will be too many threads to analyse by hand. Secondly, there is an increasing gap between (anomalous) runtime behaviour and the user's changes in the source code needed to fix it, due to DSLs, libraries, intelligent runtime systems and system software, and potentially disaggregated resources, that the application programmer would know little or nothing about. Tools are needed to automatically verify programming model assumptions, via compiler, run-time checking or formal methods.

Spotting anomalous behaviour, such as the root cause of a performance problem or bug, will be a big data or machine learning problem, requiring techniques from data mining, clustering and structure detection, as well as high scalability through summarized data, sampling and filtering and special techniques like spectral analysis. As implied above, the tools need to be interoperable with programming abstractions, so that problems in a loop in a library or dynamic scheduling of tasks can be translated into terms that the programmer can understand.

There are serious difficulties with performance analysis and debugging, and existing techniques based on printf, logging and trace visualization will soon be intractable. Existing debuggers are good for small problems, but more work is needed to (graphically) track variables to find out where the output first became incorrect, especially for bugs that are difficult to reproduce. It is necessary to verify validity of a program according to the programming model (correct task dependencies, lack of data races, etc.), via compiler checks, runtime checks or formal model checking. Performance analysis tools require lightweight data collection using sampling, folding and other techniques, so as not to increase execution time or disturb application performance (leading to non-representative analysis). This is especially important given the increasing number of available hardware performance counters (>50 per core on Intel Ivy Bridge). There is a need for both superficial on-the-fly analysis and in-depth AI and deep learning analytics. As compilers and runtime systems become more complex, there will be a growing gap between runtime behaviour and the changes in the application's source code required to improve performance—although this does not yet seem to be a significant problem.

There is a concern that future systems will have worse performance stability and predictability, due to complex code transformations, dynamic adapting for energy and faults, dynamically changing clock speeds, and migrating work [49]. This is problematic when predictability is required, e.g., for real-time applications such as weather forecasting and for making proposals for access to HPC resources (since proposals need an accurate prediction of application performance scalability). Noisy performance is problematic for the many HPC applications that involve fine-grained communication, since they become bottlenecked by the performance of the slowest process.

References

[1] WikiChip. *Cache Coherent Interconnect for Accelerators (CCIX)*. 2018. URL: https://en.wikichip.org/wiki/ccix.

[2] *The Gen-Z Consortium*. URL: https://genzconsortium.org.

[3] *OpenCAPI*. URL: https://opencapi.org.

[4] WikiChip. *NVLink, Nvidia*. URL: https://en.wikichip.org/wiki/nvidia/nvlink.

[5] A. M. Caulfield et al. "A Cloud-scale Acceleration Architecture". In: *The 49th Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO-49. Taipei, Taiwan, 2016, 7:1–7:13. URL: http://dl.acm.org/citation.cfm?id=3195638.3195647.

[6] S. Haria, M. D. Hill, and M. M. Swift. "Devirtualizing Memory in Heterogeneous Systems". In: *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '18. Williamsburg, VA, USA, 2018, pp. 637–650. DOI: 10.1145/3173162.3173194. URL: http://doi.acm.org/10.1145/3173162.3173194.

[7] Z. Yan, D. Lustig, D. Nellans, and A. Bhattacharjee. "Translation Ranger: Operating System Support for Contiguity-aware TLBs". In: *Proceedings of the 46th International Symposium on Computer Architecture*. ISCA '19. Phoenix, Arizona, 2019, pp. 698–710. DOI: 10.1145/3307650.3322223. URL: http://doi.acm.org/10.1145/3307650.3322223.

[8] "Mellanox Accelerates BlueField SoC". In: *Microprocessor Report* (Aug. 2017).

[9] *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Redmond, Washington, 2009. URL: https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery.

[10] K. J. Bowers, B. Albright, L. Yin, B. Bergen, and T. Kwan. "Ultrahigh performance three-dimensional electromagnetic relativistic kinetic plasma simulation". In: *Physics of Plasmas* 15.5 (2008), p. 055703.

[11] J. Freeman et al. "Mapping brain activity at scale with cluster computing". In: *Nature methods* 11.9 (2014), pp. 941–950.

[12] K. E. Bouchard et al. "High-Performance Computing in Neuroscience for Data-Driven Discovery, Integration, and Dissemination". In: *Neuron* 92.3 (2016), pp. 628–631. DOI: https://doi.org/10.1016/j.neuron.2016.10.035. URL: http://www.sciencedirect.com/science/article/pii/S0896627316307851.

[13] S. S. Vazhkudai et al. "The Design, Deployment, and Evaluation of the CORAL Pre-exascale Systems". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*. SC '18. Dallas, Texas, 2018, 52:1–52:12. DOI: 10.1109/SC.2018.00055. URL: https://doi.org/10.1109/SC.2018.00055.

[14] D. Kang, V. Patel, A. Nair, S. Blanas, Y. Wang, and S. Parthasarathy. "Henosis: workload-driven small array consolidation and placement for HDF5 applications on heterogeneous data stores". In: *Proceedings of the ACM International Conference on Supercomputing, ICS 2019, Phoenix, AZ, USA, June 26-28, 2019*. 2019, pp. 392–402. DOI: 10.1145/3330345.3330380. URL: https://doi.org/10.1145/3330345.3330380.

[15] S. Habib et al. *High Energy Physics Forum for Computational Excellence: Working Group Reports (I. Applications Software II. Software Libraries and Tools III. Systems)*. https://arxiv.org/abs/1510.08545. 2015. arXiv: 1510.08545 [physics.comp-ph].

[16] *NERSC Benchmarking & Workload Characterization*. https://www.nersc.gov/research-and-development/benchmarking-and-workload-characterization/.

[17] *InsideHPC: Let's Talk Exascale Podcast looks at Parallel I/O with ExaHDF5*. https://insidehpc.com/2018/02/lets-talk-exascale-podcast-looks-parallel-o-exahdf5/.

[18] Q. K. Prabhat et al. "ExaHDF5: An I/O Platform for Exascale Data Models, Analysis and Performance". In: *SciDAC 2011* ().

[19] H. Xing, S. Floratos, S. Blanas, S. Byna, M. Prabhat, K. Wu, and P. Brown. "ArrayBridge: Interweaving declarative array processing in SciDB with imperative HDF5-based programs". In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE. 2018, pp. 977–988.

[20] B. Dong, S. Byna, K. Wu, H. Johansen, J. N. Johnson, N. Keen, et al. "Data elevator: Low-contention data movement in hierarchical storage system". In: *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*. IEEE. 2016, pp. 152–161.

[21] *DAOS Storage Engine*. https://github.com/daos-stack/daos.

[22] S. P. VanderWiel, D. Nathanson, and D. J. Lilja. "Complexity and performance in parallel programming languages". In: *Proceedings Second International Workshop on High-Level Parallel Programming Models and Supportive Environments*. Apr. 1997, pp. 3–12.

[23] NVIDIA. *CUDA C Programming Guide*. Sept. 2015. URL: http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf.

[24] G. Chen, G. Li, S. Pei, and B. Wu. "High Performance Computing via a GPU". In: *2009 First International Conference on Information Science and Engineering*. Dec. 2009, pp. 238–241.

[25] A. Hunt and D. Thomas. *The Pragmatic Programmer*. Boston, MA, USA, 2000, pp. 26–33.

[26] Khronos OpenCL Working Group. *The OpenCL Specification, version 2.2*. Mar. 2016. URL: https://www.khronos.org/registry/cl/specs/opencl-2.2.pdf.

[27] OpenACC. *OpenACC Programming and Best Practices Guide*. June 2015. URL: http://www.openacc.org/sites/default/files/OpenACC_Programming_Guide_0.pdf.

[28] OpenMP Architecture Review Board. *OpenMP Application Program Interface*. 2013. URL: http://www.openmp.org/wp-content/uploads/OpenMP4.0.0.pdf.

[29] A. Duran, E. Ayguadé, R. M. Badia, J. Labarta, L. Martinell, X. Martorell, and J. Planas. "Ompss: a Proposal for Programming Heterogeneous Multi-Core Architectures". In: *Parallel Processing Letters* 21 (2011), pp. 173–193.

[30] K. Gregory and A. Miller. *C++ AMP: Accelerated Massive Parallelism with Microsoft Visual C++*. Sebastopol, CA, USA, 2012.

[31] Khronos OpenCL Working Group. *SYCL Provisional Specification, version 2.2*. Feb. 2016. URL: https://www.khronos.org/registry/sycl/specs/sycl-2.2.pdf.

[32] M. Haidl and S. Gorlatch. "PACXX: Towards a Unified Programming Model for Programming Accelerators Using C++14". In: *Proceedings of the 2014 LLVM Compiler Infrastructure in HPC*. LLVM-HPC '14. New Orleans, Louisiana, 2014, pp. 1–11.

[33] P. Yalamanchili, U. Arshad, Z. Mohammed, P. Garigipati, P. Entschev, B. Kloppenborg, J. Malcolm, and J. Melonakos. *ArrayFire - A high performance software library for parallel computing with an easy-to-use API*. Atlanta, 2015. URL: https://github.com/arrayfire/arrayfire.

[34] J. Enmyren and C. W. Kessler. "SkePU: A Multi-backend Skeleton Programming Library for multi-GPU Systems". In: *Proc. of the Int. Workshop on High-level Parallel Programming and Applications*. HLPP '10. Baltimore, Maryland, USA, 2010, pp. 5–14.

[35] M. Steuwer, P. Kegel, and S. Gorlatch. “SkelCL - A Portable Skeleton Library for High-Level GPU Programming”. In: *Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*. IPDPSW ’11. Washington, DC, USA, 2011, pp. 1176–1182.

[36] H. C. Edwards and C. R. Trott. “Kokkos: Enabling Performance Portability Across Manycore Architectures”. In: *2013 Extreme Scaling Workshop (xsw 2013)*. Aug. 2013, pp. 18–24.

[37] K. Lutz. *Boost.Compute*. 2016. URL: http://www.boost.org/doc/libs/1_61_0/libs/compute/doc/html/index.html.

[38] B. Peccerillo and S. Bartolini. “PHAST - A Portable High-Level Modern C++ Programming Library for GPUs and Multi-Cores”. In: *IEEE Transactions on Parallel and Distributed Systems* 30.1 (Jan. 2019), pp. 174–189. DOI: 10.1109/TPDS.2018.2855182.

[39] B. Imbernn, J. Prades, D. Gimnez, J. M. Cecilia, and F. Silla. “Enhancing Large-scale Docking Simulation on Heterogeneous Systems”. In: *Future Gener. Comput. Syst.* 79.P1 (Feb. 2018).

[40] I. Grasso, S. Pellegrini, B. Cosenza, and T. Fahringer. “LibWater: Heterogeneous Distributed Computing Made Easy”. In: *Proc. of the Int. Conference on Supercomputing*. ICS ’13. Eugene, Oregon, USA, 2013, pp. 161–172.

[41] J. Kim, S. Seo, J. Lee, J. Nah, G. Jo, and J. Lee. “SnuCL: An OpenCL Framework for Heterogeneous CPU/GPU Clusters”. In: *Proceedings of the 26th ACM International Conference on Supercomputing*. ICS ’12. San Servolo Island, Venice, Italy, 2012, pp. 341–352.

[42] M. Aldinucci, M. Danelutto, P. K. Kilpatrick, and M. Torquati. “FastFlow: High-level and Efficient Streaming on Multi-core”. In: 2011.

[43] J. Ragan-Kelley, A. Adams, S. Paris, M. Levoy, S. Amarasinghe, and F. Durand. “Decoupling algorithms from schedules for easy optimization of image processing pipelines”. In: (2012).

[44] T. Ben-Nun, J. de Fine Licht, A. N. Ziogas, T. Schneider, and T. Hoefler. “Stateful Dataflow Multigraphs: A data-centric model for performance portability on heterogeneous architectures”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2019, pp. 1–14.

[45] M. Radulovic, R. Sánchez Verdejo, P. Carpenter, P. Radojković, B. Jacob, and E. Ayguadé. “PROFET: Modeling System Performance and Energy Without Simulating the CPU”. In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3.2 (2019), p. 34.

[46] A. Calotoiu, T. Hoefler, M. Poke, and F. Wolf. “Using automated performance modeling to find scalability bugs in complex codes”. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM. 2013, p. 45.

[47] T. Grass et al. “MUSA: a multi-level simulation approach for next-generation HPC machines”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press. 2016, p. 45.

[48] D. B. Prats, J. L. Berral, and D. Carrera. “Automatic generation of workload profiles using unsupervised learning pipelines”. In: *IEEE Transactions on Network and Service Management* 15.1 (2017), pp. 142–155.

[49] Department of Energy Advanced Scientific Computing Advisory Committee. *Exascale Computing Initiative Review*. 2015. URL: https://www.osti.gov/scitech/servlets/purl/1222712.

6 Vertical Challenges

6.1 Green ICT and Power Usage Effectiveness

The term "Green ICT" refers to the study and practice of environmentally sustainable computing. The 2010 estimates put the ICT at 3% of the overall carbon footprint, ahead of the airline industry [1]. Modern large-scale data centres are already multiple of tens of MWs, on par with estimates for Exascale HPC sites. Therefore, computing is among heavy consumers of electricity and subject of sustainability considerations with high societal impact.

For the HPC sector the key contributors to electricity consumption are the computing, communication, and storage systems and the infrastructure including the cooling and the electrical subsystems. Power usage effectiveness (PUE) is a common metric characterizing the infrastructure overhead (i.e., electricity consumed in IT equipment as a function of overall electricity). Data centre designs taking into consideration sustainability [2] have reached unprecedented low levels of PUE. Many EU projects have examined CO2 emissions in cloud-based services [3] and approaches to optimize air cooling [4].

It is expected that the (pre-)Exascale IT equipment will use direct liquid cooling without use of air for the heat transfer [5]. Cooling with temperatures of the liquid above 45° C open the possibility for "free cooling" in all European countries and avoid energy cost of water refrigeration. Liquid cooling has already been employed in HPC since the earlier Cray machines and continues to play a key role. The CMOSAIC project [6] has demonstrated two-phase liquid cooling previously shown for rack-, chassis- and board-level cooling to 3D-stacked IC as a way to increase thermal envelopes. The latter is of great interest especially for end of Moore's era where stacking is emerging as the only path forward in increasing density. Many vendors are exploring liquid immersion technologies with mineral-based oil and other material to enable higher power envelopes.

We assert that to reach Exascale performance and beyond an improvement must be achieved in driving the Total Power usage effectiveness (TUE) metric [7]. This metric highlights the energy conversion costs within the IT equipment to drive the computing elements (processor, memory, and accelerators). As a rule of thumb, in the pre-Exascale servers the power conversion circuitry consumes 25% of all power delivered to a server. Facility targeting TUE close to one will focus the power dissipation on the computing (processor, memory, and accelerators) elements. The CMOS computing elements (processor, memory, accelerators) power dissipation (and therefore also the heat generation) is characterized by the leakage current. It doubles for every 10° C increase of the temperature [8]. Therefore the coolant temperature has influence on the leakage current and may be used to balance the overall energy effectiveness of the data centre for the applications. We expect that the (pre-)Exascale pilot projects, in particular funded by the EU, will address creation and usage of the management software for global energy optimization in the facility [9].

Beyond Exascale we expect to have results from the research related to the CMOS devices cooled to low temperatures [10] (down to Liquid Nitrogen scale, 77 K). The expected effect is the decrease of the leakage current and increased conductivity of the metallic connections at lower temperatures. We suggest that an operating point on this temperature scale can be found with significantly better characteristics of the CMOS devices. Should such operating point exist, a practical way to cool such computational device must be found. This may be one possible way to overcome the CMOS technology challenges beyond the feature size limit of 10 nm [11]. We suggest that such research funded in Europe may yield significant advantage to the European HPC position beyond Horizon 2020 projects.

The electrical subsystem also plays a pivotal role in Green ICT. Google has heavily invested in renewables and announced in 2017 that their data centres will be energy neutral. However, as big consumers of elec-

tricity, HPC sites will also require a tighter integration of the electrical subsystem with both the local/global grids and the IT equipment. Modern UPS systems are primarily designed to mitigate electrical emergencies. Many researchers are exploring the use of UPS systems as energy storage to regulate load on the electrical grid both for economic reasons, to balance the load on the grid or to tolerate the burst of electricity generated from renewables. The Net-Zero data centre at HP and GreenDataNet [12] are examples of such technologies.

Among existing efforts for power management, the majority of these approaches are specific to HPC centers and/or specific optimization goals and are implemented as standalone solutions. As a consequence, these existing approaches still need to be hooked up to the wide range of software components offered by academic partners, developers or vendors. According to [13], existing techniques have not been designed to exist and interact simultaneously on one site and do so in an integrated manner. This is mainly due to a of application-awareness, lack of coordinated management across different granularities, and lack of standardised and widely accepted interfaces along with consequent limited connectivity between modules, resulting in substantially underutilized Watts and FLOPS. To overcome the above mentioned problems, various specifications and standardization are currently under development [14, 15]

6.2 Resiliency

Preserving data consistency in case of faults is an important topic in HPC. Individual hardware components can fail causing software running on them to fail as well. System software would take down the system if it experiences an unrecoverable error to preserve data consistency. At this point the machine (or component) must be restarted to resume the service from a well-defined state. The traditional failure recovery technique is to restart the whole user application from a user-assisted coordinated checkpoint taken at synchronization point. The optimal checkpoint period is a function of time/energy spent writing the checkpoint and the expected failure rate [16]. The challenge is to guess the failure rate, since this parameter is not known in general. If a failure could be predicted, preventive action such as the checkpoint can be taken to mitigate the risk of the pending failure.

No deterministic failure prediction algorithm is known. However, collecting sensor data and Machine Learning (ML) on this sensor data yields good results [17]. We expect that the Exascale machine design will incorporate sufficient sensors for the failure prediction and monitoring. This may be a significant challenge, as the number of components and the complexity of the architecture will increase. Therefore, also the monitoring data stream will increase, leading to a fundamental Big Data problem just to monitor a large machine. We see this monitoring problem as an opportunity for the EU funding of fundamental research in ML techniques for real-time monitoring of hardware facilities in general. The problem will not yet be solved in the (pre-)Exascale machine development. Therefore, we advocate a targeted funding for this research to extend beyond Horizon 2020 projects. The traditional failure recovery scheme with the coordinated checkpoint may be relaxed if fault-tolerant communication libraries are used [18]. In that case the checkpoints do not need to be coordinated and can be done per node when the computation reaches a well-defined state. When million threads are running in a single scalable application, the capability to restart only a few communicating threads after a failure is important.

The non-volatile memories may be available for the checkpoints; it is a natural place to dump the HBM contents. We expect these developments to be explored on the time scale of (pre-)Exascale machines. It is clear that the system software will incorporate failure mitigation techniques and may provide feedback on the hardware-based resiliency techniques such as the ECC and Chipkill. The software-based resiliency has to be designed together with the hardware-based resiliency. Such design is driven by the growing complexity of the machines with a variety of hardware resources, where each resource has its own failure pattern and recovery characteristics.

On that note the compiler assisted fault tolerance may bridge the separation between the hardware-only and software-only recovery techniques [19]. This includes automation for checkpoint generation with the optimization of checkpoint size [20]. More research is needed to implement these techniques for the Exascale and post-Exascale architectures with the new levels of memory hierarchy and increased complexity of the computational resources. We see here an opportunity for the EU funding beyond the Horizon 2020 projects.

Stringent requirements on the hardware consistency and failure avoidance may be relaxed, if an application algorithm incorporates its own fault detection and recovery. Fault detection is an important aspect, too. Currently, applications rely on system software to detect a fault and bring down (parts of) the system to avoid the data corruption. There are many application environments that adapt to varying resource availability at service level—Cloud computing works in this way. Doing same from within an application is much harder. Recent work on the "fault-tolerant" message-passing communication moves the fault detection burden to the library, as discussed in the previous section. Still, algorithms must be adopted to react constructively after such fault detection either by "rolling back" to the previous state (i.e. restart from a checkpoint) or "going forward" restoring the state based on the algorithm knowledge. The forward action is subject of a substantial research for the (pre-)Exascale machines and typically requires algorithm redesign. For example, a possible recovery mechanism is based on iterative techniques exploited in Linear Algebra operations [21].

The Algorithm Based Fault Tolerance (ABFT) may also use fault detection and recovery from within the application. This requires appropriate data encoding, algorithm to operate on the encoded data and the distribution of the computation steps in the algorithm among (redundant) computational units [22]. We expect these aspects to play a role with NMP. The ABFT techniques will be required when running applications on machines where the strong reliability constraint is relaxed due to the subthreshold voltage settings. Computation with very low power is possible [23] and opens a range of new "killer app" opportunities. We expect that much of this research will be needed for post-Exascale machines and therefore is an opportunity for EU funding beyond the Horizon 2020 projects.

6.3 Impact of Memristive Memories on Security and Privacy

This section discusses security and privacy implications of memristive technologies, including emerging memristive non-volatile memories (NVMs). The central property that differentiates such memories from conventional SRAM and DRAM is their non-volatility; therefore, we refer to these memories as "NVMs".

We cover potential inherent security risks, which arise from these emerging memory technologies and on the positive side security potentials in systems and applications that incorporate emerging NVMs. Further, we also consider the impact of these new memory technologies on privacy.

6.3.1 Background

The relevance of security and privacy has steadily increased over the years. This concerns from highly complex cyber-physical infrastructures and systems-of-systems to small Internet of Things (IoT) devices if they are applied for security critical applications [24]. A number of recent successful attacks on embedded and cyber-physical systems has drawn the interest not only of scientists, designers and evaluators but also of the legislator and of the general public. Just a few examples are attacks on online banking systems [25] and malware, in particular ransomware [26] and spectacular cyber attacks to critical infrastructures, as the Stuxnet attack [27], attacks on an industrial installation in German steel works [28] and on a driving Jeep car [29], to name but a few. Meanwhile entire botnets consisting of IoT devices exist [30]. These examples may shed light on present and future threats to modern IT systems, including embedded devices, vehicles, industrial sites, public infrastructure, and and HPC supercomputers. Consequently, security and privacy may determine the future market acceptance of several classes of products, especially if they are increasingly enforced by national and EU-wide legislation [31]. Consequently, security and privacy should be considered together with (and in certain cases weighted against) the more traditional system attributes such as latency, throughput, energy efficiency, reliability, or cost.

Historically, the networks connecting the system with the outside world and the software running on the system's components were considered as a source of security weaknesses, giving rise to the terms "network security" and "software security" [32]. However, the system's hardware components are increasingly shifting into the focus of attention, becoming the Achilles' heels of systems. Researchers have been pointing to hardware-related vulnerabilities since long times, including side channels [33], fault-injection attacks [34], counterfeiting [35], covert channels [36] and hardware Trojans [37]. Several potential weaknesses in hardware components were exposed; some of the

widely publicized examples were: counterfeit circuits in missile-defense installations in 2011 [38], potential backdoors in FPGAs (later identified as undocumented test access circuitry [39]) in 2012 [40], (hypothetical) stealthy manipulations in a microprocessor's secure random number generator in 2013 [41]. Very recently, two hardware-related security breaches, Meltdown [42] and Spectre [43] were presented. They exploit advanced architectural features of modern microprocessors and affect several microprocessors that are in use today.

Meltdown and Spectre are indicative of hardware-based attacks on high-performance microprocessors: On the one hand, it is difficult for an attacker to find such weaknesses (compared to many conventional methods, from social engineering to malware and viruses), and even when the weaknesses are known it may be difficult to develop and mount concrete attacks. On the other hand, once such an attack has been found, it affects a huge population of devices. It is also extremely difficult or may even be impossible to counteract because hardware cannot easily be patched or updated in field. Corrective actions, which require the replacement of the affected hardware components by (to be produced) secure versions are usually extremely costly and may even be infeasible in practice. Healing the problem by patching the software that runs on the component is not always effective and is often associated with a barely acceptable performance penalty [42]. Consequently, new architectural features should undergo a thorough security analysis before being used.

In this section, we consider potential implications of emerging memristors, and in particular memristive non-volatile memories (NVMs) and NVM-based computer architectures on security and privacy of systems (compared to conventional memory architectures). We will discuss both: the vulnerabilities of systems due to integration of emerging NVMs, and the potential of NVMs to provide new security functions and features.

6.3.2 Memristors and Emerging NVMs: Security Risks

The crucial property of NVMs is – rather expected – their non-volatility: An NVM retains the stored information even when it is disconnected from the power supply. The first obvious consequence is the *persistency of attacks:* if the adversary managed to place malicious content (e.g., software code or manipulated parameter values) into a device's main memory, this content will not disappear by rebooting the device or powering it off. (Of course, to get rid of the malware usually additional security measures are necessary.) This is in stark contrast to volatile memories where reboot and power-off are viable ways to "heal" at least the volatile memory of an attacked system; the same system with an NVM will stay infected.

The non-volatility can simplify *read-out attacks* on unencrypted memory content. In such attacks, sensitive data within an electronic component are accessed by an adversary with physical access to the device using either direct read-out or side-channels, e.g., measuring data-dependent power consumption or electromagnetic emanations. Usually, volatile memory must be read out in the running system, with the full system speed; moreover the system may be equipped with countermeasures, e.g., tamper-detectors which would delete the memory content once they identify the attempted attack. An exception are so-called cold boot attacks where the memory content may persist for several minutes or even hours [44]. An attacker who powered off a system with sensitive data in an NVM can analyze the NVM block offline.

It is currently not clear whether emerging memristive NVMs bear *new side-channel vulnerabilities.* For example, many security architectures are based on encrypting sensitive information and overwriting the original data in the memory by an encrypted version or randomness. It is presently not clear whether memristive elements within NVMs exhibit a certain extent of "hysteresis", which may allow the adversary to reconstruct the state, which a memory cell had before the last writing operation with some degree of accuracy. This property was discussed in [45] from the forensic point of view. Whether this vulnerability indeed exists, must be established for each individual NVM technology (like STT-RAM or ReRAM) by physical experiments. If it exists this might allow or at least support side-channel attacks.

First thoughts whether emerging NVMs would have impact on the vulnerability against implementation attacks can be found in [46]. The attack scenarios mentioned therein are typically counted as fault attacks and probing attacks. (In the field of implementation attacks the nomenclature is not always unique.) The authors conclude that ReRAMs would prevent these

attacks. In [46] experiments were not conducted but the authors announce future experiments. To our knowledge typical side-channel attacks (power, timing, cache etc.) have not been considered so far in the context of NVMs.

Some of the memristive NVMs are also prone to *active manipulations, enabling fault attacks.* For example, the recent paper [47] considers non-invasive magnetic field attacks on STT-RAMs, where the adversary overrides the values of the cell by applying either a static or an alternating magnetic field. The authors of [47] note that this attack can be mounted on a running system or in passive mode, where it could, e.g., compromise the boot process.

While all of the mentioned attack scenarios can have severe consequences already against an adversary who has physical access to the end-product, they may be even more dangerous if an attacker manages to compromise the system design and the manufacturing process, and was able to insert a *Hardware Trojan* into the circuitry. Trojans can be inserted during semiconductor manufacturing [48], they can be lurking in third-party intellectual-property cores [49], and even CAD tools used for circuit design may plant Trojans [50]. Emerging NVMs might facilitate both the establishment of Trojans in the system (e.g., by placing their trigger sequences in a non-volatile instruction cache) and also multiply the damaging potential of Trojans.

6.3.3 Memristors and Emerging NVMs: Supporting Security

On the positive side, memristors can be the basis for security primitives that are difficult or expensive to realize technically by conventional hardware and software. Depending on the scenario one such primitive might a *random number generator (RNG),* which is useful, for instance, for on-chip generation of secure cryptographic keys, signature parameters, nonces and for creating masks to protect cryptographic cores against side-channel analysis. Roughly speaking, RNGs can be divided into deterministic RNGs (DRNGs) (a.k.a. pseudorandom number generators) and true RNGs. The class of true RNGs can further be subdivided into physical RNGs (PTRNGs, using dedicated hardware) and non-physical trueRNGs (NPTRNGs) [51]. Memristors and NVMs on their basis might be beneficial for both DRNGs and true RNGs. For DRNGs, NVMs might be used to store the internal state, thus reducing the need for additional non-volatile memory, saving the copy process of the internal state to non-volatile memory, or reseeding upon each power-on. Of course, such NVM cells must be secure against read-out and manipulation since otherwise an attacker might be able to predict all future random numbers. In TRNGs, memristors might serve as sources of entropy (see e.g. [52] and [53]), providing sources for physical RNGs or for non-physical non-deterministic RNGs as Linux /dev/random, for instance. Whether this use is realistic depends on the outcome of physical experiments for individual memristive technologies. To this end, suitable random parameters (e.g., the duration of the transition between stable states) must be identified; then, a stochastic model (for PTRNGs) or at least a reliable lower entropy bound per random bit (for NPTRNGs) must be established and validated, and finally the entropy per bit must be estimated [54]; see also [55, 56, 57]. In [52] and [53] the authors focus only on the statistical properties of the generated random numbers, which are verified by NIST randomness tests.

Another possible memristor-enabled security primitive could be a Physically Unclonable Function (PUF). A PUF is a "fingerprint" of an individual circuit instance among a population of manufactured circuits [58]. It should reliably generate a unique, circuit-specific bitstring, and it shall be impossible to produce another circuit with the same fingerprint. PUFs are used for on-chip generation of secret keys and for authentication protocols, for instance, but also for tracking circuits and preventing their counterfeiting [59]. PUFs based on memory cells are well-known [60], and these insights can perhaps directly be applied to emerging NVMs [61]. However, the emerging near-memory and in-memory concepts where NVMs are tightly coupled with logic, create potentials for richer varieties of PUF behavior, such as "strong PUFs" which support challenge-response authentication protocols [62]. A strong PUF proposal based on memristive elements has been proposed in [63]. Moreover, it was suggested to leverage non-linearity of memristors to define "public PUFs" which overcome certain deficiencies of traditional PUFs [64].

An interesting question might be whether emerging memristive cells and NVM-enabled architectures are better or worse protected against *counterfeiting and reverse engineering* compared to conventional circuits. On the one hand, the designer can replace identifiable circuit structures by a regular fabric similar to reconfigurable gate-arrays that is controlled by values stored in an NVM. This makes it difficult for an

attacker to apply the usual flow to reconstruct the circuit functionality: depackage the circuit, extract its individual layers, and apply optical recognition to find logic gates, memory cells, interconnects, and other structures. In fact, if the content of the "configuration" NVM cells is lost during deprocessing, its functionality is irretrievably lost as well. Possibly, attackers may find ways to read out the values in memristive elements prior to deprocessing. In addition, memristors can power anti-counterfeiting solutions, like PUFs. As with other security attributes, the resistance of circuits to reverse engineering is a cat-and-mouse game where the defender invents new protections and the attacker finds way around this protection; NVMs could substantially change the rules of this game.

6.3.4 Memristors, Emerging NVMs and Privacy

Privacy stands in a non-trivial relationship with security, and therefore security implications of memristors can have positive or negative consequences for privacy [65]. On the one hand, security breaches that lead to unauthorized access to user data (e.g., leaked secret keys), or compromise their authenticity and integrity, are clearly detrimental for privacy (loss of privacy or of availability). To this end, all properties of NVMs that simplify attacks on encryption negative privacy impact, and all beneficial features of NVMs, e.g., schemes (e.g., read-out attacks or new side-channel attacks) have generation of secure secret keys, have positive consequences. Here, security and privacy requirements are consistent.

Security and privacy may get in conflict when it comes to methods which track in an undesired and unnecessary way individual circuit instances, e.g., by storing a unique identifier in an on-chip NVM, or by creating such an identifier using a PUF. This functionality is beneficial for security and in particular to prevent counterfeiting or overbuilding [59].

References

[1] L. Smarr. "Project GreenLight: Optimizing Cyberinfrastructure for a Carbon-Constrained World". In: *Computer* 43.1 (Jan. 2010), pp. 22–27. DOI: 10.1109/MC.2010.20.

[2] J. Shuja, A. Gani, S. Shamshirband, R. W. Ahmad, and K. Bilal. "Sustainable Cloud Data Centers: A survey of enabling techniques and technologies". In: *Renewable and Sustainable Energy Reviews* 62.C (2016), pp. 195–214.

[3] *The ECO2Clouds Project*. 2017. URL: http://eco2clouds.eu/.

[4] *The CoolEmAll Project*. 2017. URL: https://www.hlrs.de/about-us/research/past-projects/coolemall/.

[5] *The DEEP-ER Project*. 2017. URL: http://juser.fz-juelich.de/record/202677.

[6] *The CMOSAIC Project*. 2017. URL: http://esl.epfl.ch/page-42448-en.html.

[7] *TUE and iTUE*. 2017. URL: http://eehpcwg.lbl.gov/subgroups/infrastructure/tue-team.

[8] D. Wolpert and P. Ampadu. *Managing Temperature Effects in Nanoscale Adaptive Systems*. 2012. 174 pp. DOI: 10.1007/978-1-4614-0748-5.

[9] S. Li, H. Le, N. Pham, J. Heo, and T. Abdelzaher. "Joint Optimization of Computing and Cooling Energy: Analytic Model and a Machine Room Case Study". In: *2012 IEEE 32nd International Conference on Distributed Computing Systems*. 2012, pp. 396–405. DOI: 10.1109/ICDCS.2012.64.

[10] M. J. Ellsworth. *The Challenge of Operating Computers at Ultra-low Temperatures*. 2001. URL: https://www.electronics-cooling.com/2001/08/the-challenge-of-operating-computers-at-ultra-low-%20temperatures/.

[11] J. Hu. *Low Temperature Effects on CMOS Circuits*. 2017. URL: http://users.eecs.northwestern.edu/~jhu304/files/lowtemp.pdf.

[12] *The GreenDataNet Project*. 2017. URL: http://www.greendatanet-project.eu./.

[13] "A Strawman for an HPC PowerStack". In: *OSTI Technical Report* (Aug. 2018). URL: https://hpcpowerstack.github.io/strawman.pdf.

[14] URL: https://hpcpowerstack.github.io/.

[15] URL: https://www.osti.gov/biblio/1458125.

[16] J. S. Plank and M. G. Thomason. "Processor Allocation and Checkpoint Interval Selection in Cluster Computing Systems". In: *J. Parallel Distrib. Comput.* 61.11 (Nov. 2001), pp. 1570–1590. DOI: 10.1006/jpdc.2001.1757.

[17] D. Turnbull and N. Alldrin. *Failure prediction in hardware systems*. Tech. rep. University of California, San Diego, 2003. URL: http://cseweb.ucsd.edu/~dturnbul/Papers/ServerPrediction.pdf.

[18] G. E. Fagg and J. J. Dongarra. "FT-MPI: Fault Tolerant MPI, Supporting Dynamic Applications in a Dynamic World". In: *Recent Advances in Parallel Virtual Machine and Message Passing Interface: 7th European PVM/MPI Users' Group Meeting*. 2000, pp. 346–353. DOI: 10.1007/3-540-45255-9_47.

[19] T. Herault and Y. Robert. *Fault-Tolerance Techniques for High-Performance Computing*. 2015. 320 pp. DOI: 10.1007/978-3-319-20943-2.

[20] J. S. Plank, M. Beck, and G. Kingsley. "Compiler-Assisted Memory Exclusion for Fast Checkpointing". In: *IEEE Technical Committee on Operating Systems and Application Environments* 7 (1995), pp. 62–67.

[21] J. Langou, Z. Chen, G. Bosilca, and J. Dongarra. "Recovery patterns for iterative methods in a parallel unstable environment". In: *SIAM Journal on Scientific Computing* 30.1 (2007), pp. 102–116.

[22] K.-H. Huang and J. A. Abraham. "Algorithm-Based Fault Tolerance for Matrix Operations". In: *IEEE Transactions on Computers* C-33.6 (1984), pp. 518–528. DOI: 10.1109/TC.1984.1676475.

[23] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan. "Deep Learning with Limited Numerical Precision". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*. ICML'15. 2015, pp. 1737–1746.

[24] F. Regazzoni and I. Polian. "Securing the hardware of cyber-physical systems". In: *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2017, pp. 194–199.

[25] BSI für Bürger. *Online-Banking*. https://www.bsi-fuer-buerger.de/BSIFB/DE/DigitaleGesellschaft/OnlineBanking/onlinebanking_node.html.

[26] Bundesamt für Sicherheit in der Informationstechnik (BSI). *Ransomware - Bedrohungslage, Prävention & Reaktion*. https://www.bsi.bund.de/DE/Themen/Cyber-Sicherheit/Empfehlungen/Ransomware/Ransomware.pdf. Mar. 2016.

[27] R. Langner. "Stuxnet: Dissecting a Cyberwarfare Weapon". In: *IEEE Security & Privacy* 9.3 (2011), pp. 49–51.

[28] BBC News. *Hack attack causes 'massive damage' at steel works*. URL: http://www.bbc.com/news/technology-30575104.

[29] A. Greenberg. *Hackers remotely kill a Jeep on the highway—With me in it*. Wired. 2015. URL: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[30] K. on Security. *Source code for IoT botnet 'Mirai' released*. 2016. URL: https://krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released/.

[31] *The EU General Data Protection Regulation (GDPR) Portal*. URL: https://www.eugdpr.org/.

[32] C. Eckert. *IT-Sicherheit - Konzepte, Verfahren, Protokolle*. 6th ed. 2009.

[33] S. Mangard, E. Oswald, and T. Popp. *Power analysis attacks - Revealing the secrets of smart cards*. 2007.

[34] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache. "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures". In: *Proceedings of the IEEE* 100.11 (2012), pp. 3056–3076.

[35] F. Koushanfar, S. Fazzari, C. McCants, W. Bryson, M. Sale, and M. P. P. Song. "Can EDA combat the rise of electronic counterfeiting?" In: *DAC Design Automation Conference 2012*. 2012, pp. 133–138.

[36] Z. Wang and R. B. Lee. "Covert and side channels due to processor architecture". In: *ASAC*. 2006, pp. 473–482.

[37] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan. "Hardware Trojan attacks: Threat analysis and countermeasures". In: *Proceedings of the IEEE* 102.8 (2014), pp. 1229–1247.

[38] D. Lim. *Counterfeit Chips Plague U.S. Missile Defense*. Wired. 2011. URL: https://www.wired.com/2011/11/counterfeit-missile-defense/.

[39] S. Skorobogatov. *Latest news on my Hardware Security Research*. URL: http://www.cl.cam.ac.uk/~sps32/sec_news.html.

[40] S. Skorobogatov and C. Woods. "Breakthrough silicon scanning discovers backdoor in military chip". In: *Cryptographic Hardware and Embedded Systems – CHES 2012*. 2012, pp. 23–40.

[41] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson. "Stealthy dopant-level hardware Trojans". In: *Cryptographic Hardware and Embedded Systems - CHES 2013*. 2013, pp. 197–214.

[42] M. Lipp et al. *Meltdown*. 2018. URL: https://meltdownattack.com.

[43] P. Kocher et al. *Spectre attacks: Exploiting speculative execution*. 2018. URL: https://meltdownattack.com.

[44] A. Halderman et al. "Lest we remember: Cold boot attacks on encryption keys". In: *17th USENIX Security Symposium*. 2008, pp. 45–60.

[45] J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. McDonald, G. S. Rose, and B. Wysocki. "Nano meets security: Exploring nanoelectronic devices for security applications". In: *Proceedings of the IEEE* 103.5 (2015), pp. 829–849.

[46] Z. Dyka, C. Walczyk, D. Walczyk, C. Wenger, and P. Langendörfer. "Side channel attacks and the non volatile memory of the future". In: CASES. 2012, pp. 13–16.

[47] A. De, M. N. I. Khan, J. Park, and S.Ghoosh. "Replacing eFlash with STTRAM in IoTs: Security challenges and solutions". In: *Journal of Hardware and Systems Security* 1.4 (2017), pp. 328–339.

[48] R. Kumar, P. Jovanovic, W. P. Burleson, and I. Polian. "Parametric Trojans for fault-injection attacks on cryptographic hardware". In: *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography*. 2014, pp. 18–28.

[49] I. Polian, G. T. Becker, and F. Regazzoni. *Trojans in early design steps—An emerging threat*. TRUDEVICE Conf. 2016. URL: http://upcommons.upc.edu/handle/2117/99414.

[50] M. Potkonjak. "Synthesis of trustable ICs using untrusted CAD tools". In: *Design Automation Conference*. 2010, pp. 633–634.

[51] W. Schindler. "Random number generators for cryptographic applications". In: *Cryptographic Engineering*. 2009, pp. 5–23.

[52] C. Huang, W. Shen, Y. Tseng, C. King, and Y. C. Lin. "A Contact-resistive random-access-memory-based true random number generator". In: *IEEE Electron Device Letters* 33.8 (2012), pp. 1108–1110.

[53] H. Jiang et al. "A novel true random number generator based on a stochastic diffusive memristor". In: *Nature Communications* 8 (2017). DOI: 10.1038/s41467-017-00869-x.

[54] W. Schindler. "Evaluation criteria for physical random number generators". In: *Cryptographic Engineering*. 2009, pp. 25–54.

[55] *AIS 20: Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren. Version 3*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_pdf.pdf. May 2013.

[56] *AIS 31: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren. Version 3*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_pdf.pdf. May 2013.

[57] W. Killmann and W. Schindler. *A proposal for: Functionality classes for random number generators. Mathematical-technical reference AIS20 and AIS31, Version 2.0*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_20_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile&v=1. Sept. 2011.

[58] U. Rührmair and D. E. Holcomb. "PUFs at a glance". In: *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2014, pp. 1–6.

[59] F. Koushanfar. "Integrated circuits metering for piracy protection and digital rights management: An overview". In: *Proceedings of the 21st Edition of the Great Lakes Symposium on Great Lakes Symposium on VLSI*. 2011, pp. 449–454.

[60] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls. "FPGA intrinsic PUFs and their use for IP protection". In: *Cryptographic Hardware and Embedded Systems - CHES 2007*. 2007, pp. 63–80.

[61] P. Koeberl, Ü. Koçabas, and A.-R. Sadeghi. "Memristor PUFs: A new generation of memory-based physically unclonable functions". In: *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2013, pp. 428–431.

[62] U. Rührmair, H. Busch, and S. Katzenbeisser. "Strong PUFs: Models, Constructions, and Security Proofs". In: *Towards Hardware-Intrinsic Security: Foundations and Practice*. 2010, pp. 79–96.

[63] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott. "Memristive crypto primitive for building highly secure physical unclonable functions". In: *Scientific Reports* 5 (2015).

[64] J. Rajendran, G. Rose, R. Karri, and M. Potkonjak. "Nano-PPUF: A memristor-based security primitive". In: *2012 IEEE Computer Society Annual Symposium on VLSI*. 2012, pp. 84–87.

[65] N. Rathi, S. Ghosh, A. Iyengar, and H. Naeimi. "Data privacy in non-volatile cache: Challenges, attack models and solutions". In: *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2016, pp. 348–353.