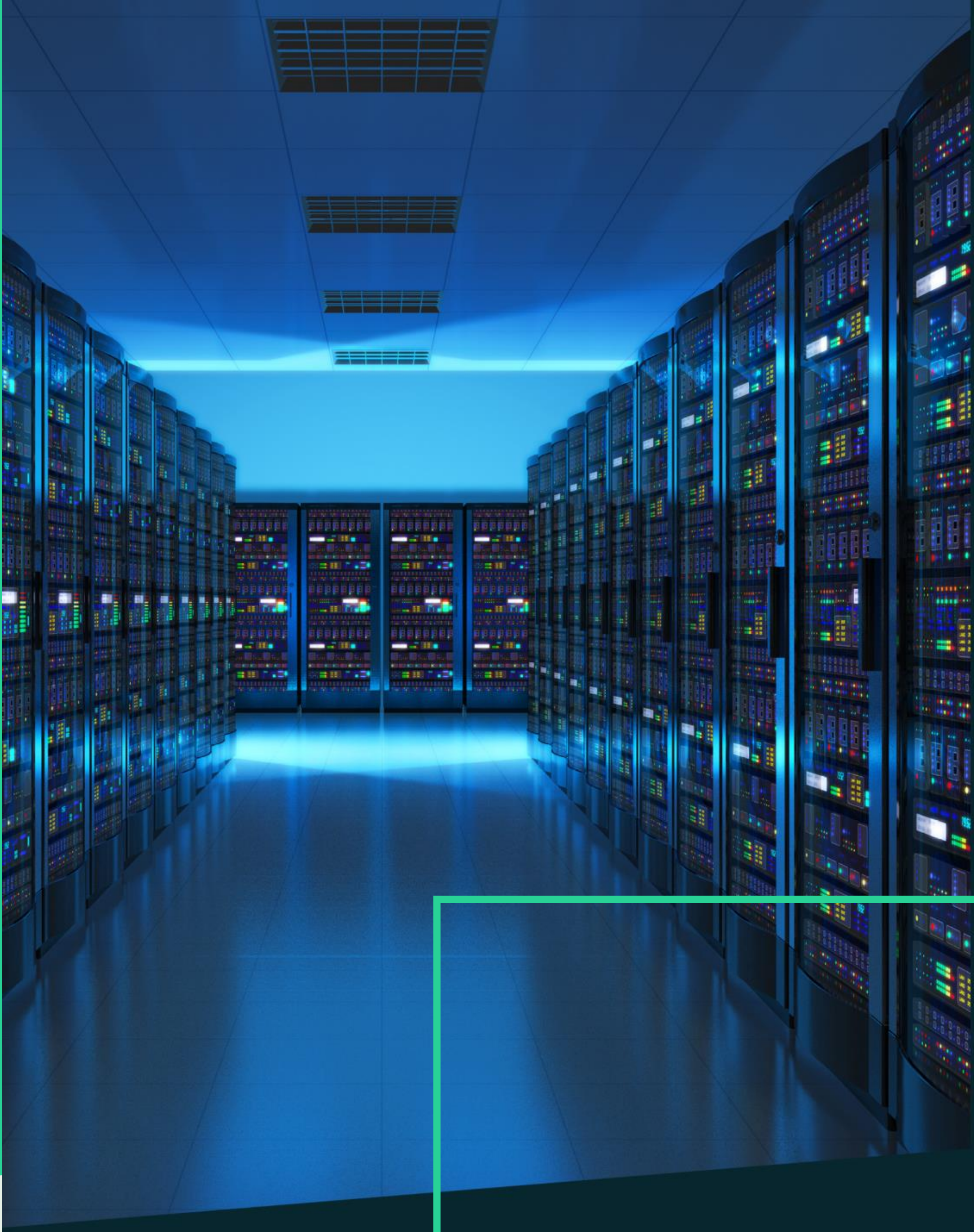

Future Computing

Overview of Technological Landscape
(Full Version)

Inhalt

Introduction

Nowadays progress in society, science, and technology is closely linked to the advances in IT and computing technologies. From healthcare to finance, transportation to manufacturing, agriculture to chemical industry: computing technologies affect almost every aspect of the modern economy. With the increasing complexity of problems, we seek to solve, as well as environmental impact of these technologies, there is a growing need for accessible, scalable, and more powerful computing that is also energy efficient.

For over 60 years, progress in computing power was driven by Moore's Law, which states that the number of transistors on a microprocessor chip doubles every two years. Today, though, we are approaching the physically defined limitations of this law and it is becoming increasingly difficult to double computing power every two years. As a result, experts are discussing various computational concepts, architectures and platforms that could – in the mid and long term – further expand the boundaries of modern computing.

In this paper we provide an overview of some of the most prominent technological approaches that promise to expand the current computing landscape by offering advantages in terms of computing power scaling, energy efficiency and applicability to a new range of problems. This includes new computing concepts such as quantum or neuromorphic computing as well as new computing platforms such as optical or chemical computing.

It is important to note that all future computing concepts utilize at least one new approach on these levels of abstraction:

- New basic concepts of computation e. g. based on quantum effects, based on principles of information processing inspired in the brain, based on non-deterministic emergent effects or on the dynamics of chemical reactions. Each of these directions also requires new computer hardware architectures.
- New platforms or information carriers: Some of these approaches require entirely new classes of devices e. g. molecule sensors in chemical computing, but most approaches can improve existing devices in a novel way (e. g. analog CMOS for neuromorphic computing or photonics for optical and quantum computing).

Although each approach has its unique strengths and challenges, it is anticipated that the potential benefits will result from combining various technologies in an efficient manner. This paper will explore technological approaches across different levels of abstraction. However, providing a comprehensive description of every conceivable computing approach is beyond the scope of this paper.

> Our goal is to offer guidance to anyone who seeks to gain a better understanding of the currently emerging computing landscape and to help evaluate these approaches based on potential benefits for businesses in the future.

How to read this document

In this paper, we categorize each future computing approach according to the framework presented below.

- **Short description:** An overview of the respective computing approach and functionality, including an estimation of the current technology readiness level (TRL).
- **Benefits:** Potential benefits of the technology as compared to classical digital computing available today.
- **Limitations:** Inherent theoretical or physical limitations of the approach, e. g. in terms of problems classes the technology can address.
- **Applications and Use-Cases:** Applications denote potential usage domains. A Use-Case is a demonstrated proof-of-concept or an implemented solution (when available).
- **Energy Consumption Consideration:** If at least a rough assessment about the energy efficiency (per computation) is available, it will be given. It is only an order-of-magnitude assessment indicating, how efficient this technology is compared with others.
- **OS / Software:** An outline of the software environment required for a particular technology.
- **Physical Area of Use:** We differentiate between three main deployment and consumption options of the technology: cloud-based, on-premises, or edge computing.
- **Development Challenges:** Existing technological challenges which needs to be overcome for a practical and industrial use of the technology.

At the end of the paper, we provide a curated list of resources for further reading on each specific approach.

List of used abbreviations

- ASIC - Application-specific integrated circuit
- AI - Artificial Intelligence
- CMOS – Complementary metal-oxide-semiconductor
- CPU – Central processing unit
- DNA – Deoxyribonucleic acid
- DNN – Deep neural network
- EU – European Union
- FPGA – Field-programmable gate array
- GPU – Graphic processing unit
- IC – Integrated Circuit
- IMC – In-memory computing
- IMP – In-memory processing
- MB – Megabyte. 1 MB = 1,024 kilobytes (kB) = 1, 048,576 bytes
- ML – Machine learning
- nm – Nanometer. 1 nm = 10^{-9} m
- PIC – Photonic integrated circuit
- PIM – Processing in memory
- QC – Quantum Computing
- QUBO – Quadratic unconstrained binary optimization
- RAM – Random Access Memory
- RNA – Ribonucleic acid
- RSA – Rivest–Shamir–Adleman, a public-key cryptosystem
- TRL – Technology readiness level
- VMM – vector matrix multiplication

Technology readiness levels

The Technology Readiness Level (TRL) scale was introduced into the EU funded projects arena in 2014 as part of the Horizon 2020 framework program[1]. This document uses the same scale with the following definitions:

- **TRL 1** – basic principles observed
- **TRL 2** – technology concept formulated
- **TRL 3** – experimental proof of concept
- **TRL 4** – technology validated in lab
- **TRL 5** – technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies)
- **TRL 6** – technology demonstrated in relevant environment (industrially relevant environment in the case of key enabling technologies)
- **TRL 7** – system prototype demonstration in operational environment
- **TRL 8** – system complete and qualified
- **TRL 9** – actual system proven in operational environment (competitive manufacturing in the case of key enabling technologies or in space)

[1] TRL Scale in Horizon Europe and ERC - explained - Enspire Science Ltd.

Technology		TRL	Benefits	Limitations	Applications	Area of Use	Challenges
Quantum Computing (QC)		0 – 6[2]	▪ Solving bigger, more complex problems ▪ Handling more data ▪ High energy efficiency	▪ Limited stability ▪ Not a universal computer ▪ No universal advantage	▪ Optimization ▪ Forecasting, scenario modelling ▪ Chemical simulation	▪ Cloud ▪ On-premises ▪ Edge	▪ Hardware scaling ▪ Identifying business-relevant problems ▪ Integration with classical computing ▪ Quantum storage
Neuromorphic Computing		2 – 7[3]	▪ High energy efficiency ▪ High processing speed ▪ Robustness	▪ Not efficiently representing numbers ▪ Based on incomplete approximations of the brain	▪ Medicine ▪ Computer vision ▪ Autonomous robots and self-driving cars	▪ Edge ▪ Cloud	▪ Distributing memory among multiple processors on a chip ▪ Scalable messaging and interconnect architectures ▪ New algorithms, software and design tools
Optical Computing	**Analog**	7	▪ Extremely parallelizable ▪ Ultra-low power consumption and low latency ▪ Mature fabrication nodes	▪ Power loss during electronic-optical conversions	▪ Deep learning ▪ Data encryption	▪ Cloud ▪ Edge	▪ Large area footprint ▪ Scaling
	Digital	4		▪ Intrinsically low light-light interaction hinders operation	▪ General use	▪ Cloud ▪ On-premises	▪ Alternative architectures needed ▪ Chip integration and material development ▪ Optical memory
Digital Annealing		9	(As compared to QC) ▪ Full connectivity among bits ▪ High precision in problem formulation ▪ Large problem sizes solvable ▪ No special environment needed	▪ Restricted to combinatorial optimization problems ▪ Close to optimal solution, no guarantee for the global optimum	▪ Traffic flow optimization ▪ Drug design ▪ Portfolio optimization ▪ Production planning	▪ Cloud ▪ On-premises	▪ Not every combinatorial optimization problem is suited ▪ Additional resources to formulate problems with higher order polynomials
DNA Computing		3 – 4	▪ Extremely parallelizable ▪ High energy efficiency ▪ High storage density (for DNA storage)	▪ Low signal propagation speed (hours) ▪ Weak scaling ▪ Propensity for discrete problems	▪ Combinatorial problems, search, scheduling, clustering ▪ Cryptography and intrusion detection	▪ Cloud	▪ Large-scale parallelization to compensate for slow signal propagation ▪ Lowering the operation costs ▪ Mapping business problems

[2] TRL depends on the physical realization of qubits (highest is for superconducting qubits)

[3] Differs for neuromorphic chips used for inference (TRL 6 – 7) and training (TRL 2 – 4)

1 Quantum Computing

Dr. Jannis Ehrlich (Fraunhofer IWM),
Dr. Oliver Oberst (IBM), Dr. Christoph
F. Strnadl (Software AG)

1.1 Short Description

Quantum computing (QC) is computing based on direct manipulations of quantum systems by discretely transforming their respective quantum states[4]. Instead of bits with just the two states »0« or »1«, present in classical computers, quantum computers use qubits which can have any state in between including a phase, such that one needs two real numbers to exactly describe the state of a single qubit. Also contrary to normal bits, several qubits can be entangled leading to an inseparable multi-qubit state, which needs exponentially more parameters to be exactly specified.[5]

A quantum computer is a statistical computer which returns the results of a (quantum) algorithm according to a certain probability distribution corresponding to the quantum state measured. Thus, the same instructions must be run many times to obtain a reasonable statistic for the »correct« (intended) result, and the resulting probability distribution must be post-processed for further computational use.

Like classical computing, QC comprises hardware, software, and quantum algorithms. Quantum simulation is another important concept within the generic QC domain. It has no direct counterpart in classical computing and refers to purpose-built quantum computers whose computation is limited to the (extremely fast) simulation of another quantum system.

As of today, many different physical realizations of quantum computers utilizing various hardware platforms, including e. g. ion traps, superconducting circuits, neutral atom arrays, various solid-state approaches, optical quantum computing etc. The question of the most economic quantum computing »hardware« has yet to be answered.

Technology Readiness Level

It strongly depends on the qubit realization technology, varying between 0 to 6 (7 being the demonstration of »quantum advantage«).[6] The quantum states are fragile leading to not negligible errors (see section Limitations), which in the future can be corrected by error correction. However, this approach might be superseded by error mitigation to reach quantum advantage.

1.2 Benefits

Quantum computing offers the promise of identifying and executing suitable (quantum) algorithms with significantly better scaling in problem size than classical

[4] adapted after N.D. Mermin 2007, *Quantum Computer Science*.
[5] Strnadl & Schöning (2022). Quantum Computing and Software AG. Part 1.
[6] Technology Readiness Level of Quantum Computing Technology (QTRL) (fz-juelich.de)

computers. This allows computations on much more data and the solution to problems that are out of reach for classical computers.

Furthermore, some quantum algorithms may provide a significant (polynomial or, in cases, even exponential) speed-up over conventional algorithms (i. e., dramatically reduced execution times). This feature of QC will widen the problem space towards more complex and longer-lasting algorithms including more data, larger systems, finer resolution, and better accuracy. Note that different hardware approaches to QC come with different additional benefits.

1.3 Energy Consumption Considerations

Based on the physical basic principles used by all technical implementations of quantum computing, the basic needed energy per computation operation is expected to be extremely low compared with state-of-the-art semiconductor technology. Nevertheless, a direct comparison between quantum computing and classical computing in terms of energy consumption is not possible up today, due to the different computation principles and the different kind of algorithms.

Besides, the main secondary energy spent for quantum computing is in fact the need of expensive cooling needed for some of the QC realizations, such as superconducting circuits (10 mK), trapped ions (4-10 K) or quantum dots in silicon (ca. 1 K). To achieve that, rather expensive in terms of needed energy Helium-based cooling is today used.

In addition to the cooling requirements, another factor that affects the energy consumption of quantum computing is the error correction process. Error correction is essential for ensuring the accuracy of quantum computations, but it requires additional qubits and computational overhead, which translates to higher energy consumption. Besides, QC could not run without classical IT to control, feed in and extract data. Despite these challenges, being "green" by physical principles quantum computing still offers the potential for significant energy savings compared to classical computing.

1.4 Limitations

- **Quantum computing does not extend the limits of computability**: It cannot compute what classical computers cannot – albeit much slower – compute.
- **Limited stability:** Quantum computers easily interact with their environment in uncontrolled or uncontrollable ways. This interaction can change the values of individual qubits and even destroy the entire quantum state, including all the information stored in it. This phenomenon is called decoherence and makes the construction of quantum computers more challenging than classical computers.
- **Not a universal computer**: The calculation model used by quantum computers is limited to a subset of algorithms[7] that are known in classical computing. This means that while quantum computers can perform certain types of computations much faster than classical computers, there are still many tasks that a quantum computer cannot perform or for which a classical computer may be better suited.

[7] Unitary transformations on finite dimensional (mostly discrete) Hilbert spaces, a subsection of classical Linear Algebra (viz. matrix calculations).

As a result, a quantum computer is not a universal computer and is not well-suited for running universal computations.

- **It seems to be hard coming up with good quantum algorithms.** The lack of quantum algorithms readily applicable to business seems to stem from the fact that our mind is rooted in the classical (i.e., non-quantum) world. We readily employ classical Boolean (i. e., two-valued) logic, classical reasoning, and causality (if -then-else) to cope with our environment – an approach, which has worked exceedingly well for our species for the last couple of 100,000 years. Classical thinking will only yield classical algorithms.
- **No universal quantum advantage exists**. While quantum computers have the potential to provide significant speedups for certain types of problems, this advantage does not apply to all conceivable algorithms or problems. For example, while quantum computers are expected to provide a significant speedup for factoring large numbers and breaking certain cryptographic schemes, they may not provide any advantage for other types of problems. While this limitation may seem discouraging, it is also reassuring in cases, such as in cryptography, where it is known that no universal attack against any cryptographic scheme can exist.

1.5 Applications

- **Optimization**: Optimization problems are ubiquitous in many fields, from finance and logistics to transportation and resource management. In the field of risk analysis, for example, quantum computing can be used to analyze and manage risk in financial portfolios, helping investors to make more informed decisions. In traffic and routing, quantum algorithms can be used to optimize traffic flow and reduce congestion, leading to more efficient and sustainable transportation systems. In resource management, quantum computing can be used to optimize the allocation of resources, such as energy and water, to improve efficiency and reduce waste. [8]
- **Forecasting and scenario modelling**: Quantum computing has the potential to significantly improve forecasting in a variety of fields, including risk, weather, and traffic. In risk analysis, quantum computing can be used to perform financial risk analysis more quickly and accurately than classical computers. In weather forecasting, quantum computing can help improve the accuracy and resolution of weather models. Quantum computing can also aid in predicting traffic patterns, allowing for more efficient routing. In addition to these specific applications, quantum computing can also be used to develop more advanced forecasting techniques in general. Quantum computing can also be used to simulate different scenarios, such as the impact of natural disasters on infrastructure or the spread of a virus in a population[9].
- **Simulation:** Quantum computers can simulate the behavior of large, complex systems more accurately and quickly than classical computers, allowing researchers to study phenomena that would otherwise be too difficult to analyze. Quantum computing can be used to simulate the behavior of materials at the atomic level as well as chemical reactions and properties, which is essential in fields such as materials science and drug discovery.

[8] Example: Industry Quantum Computing Applications QUTAC Application Group

[9] Example: Forecasting financial crashes with quantum computing. R. Orus, S. Mugel, E. Lizaso

- **Further special algorithms**: Quantum machine learning (e.g., very deep or complex models)[10], quantum AI (quantum neural networks), quantum multi-agent systems (e.g., with millions of agents), quantum chemistry, high energy physics.
- **Cryptography**: Quantum computing has the potential to impact the field of cryptography. One of the most famous and widely used cryptographic protocols, the RSA algorithm, relies on the difficulty of factoring large numbers to provide security. However, Shor's algorithm, a quantum algorithm, is capable of factoring large numbers exponentially faster than any classical algorithm. This means that a quantum computer could potentially break RSA encryption, rendering it useless for secure communication.

1.6 Physical Area of Use

- **Cloud:** To overcome the inherent complexity of building and operating quantum computers, most organizations are likely to use them through cloud-based services offered by third-party providers. These services will expose the functionality of quantum computers through APIs that can be accessed over the public Internet.
- **On-premises:** Option for organizations with dedicated capabilities, specialized requirements, and sufficient financial resources (research centers, universities, military and defense, public administration, banks). For example, banks could use on-premises quantum computers to create theoretically unbreakable cypher systems, while logistics providers could leverage the competitive advantage of sophisticated optimization algorithms by running them locally on their own quantum computers.
- **Edge** (Mobile): Currently there are also “non-stationary” approaches being developed to allow for smaller quantum computing systems to be administered with reasonable effort (e.g., rack-based) or on edge avoiding complex environmental shielding.

1.7 OS / Software

Several software frameworks e. g., QISKIT, Q#, Bracket and others have been developed to provide a domain-specific language (DSL) for specifying and executing quantum algorithms at the level of universal gates independent from the underlying realization of the physical qubits. Typically, the same DSLs can also be used to simulate the quantum algorithm. Note, though, that this form of “programming” is on a similar level as assembly language. Firmware on control electronics (FPGAs etc.) is also required to “run” a quantum computer. Optical quantum computers, as a special case here, operate as an accelerator and do not run software directly. Custom software is used for tuning and configuration of the optical QC device.

[10] Example: Supervised learning with quantum enhanced feature spaces, Havlíček, V., Córcoles, A.D., Temme, K. et al.

1.8 Development Challenges

- **Hardware scaling**: Even though typical quantum algorithms only need a small number of logical qubits; current QC technologies require a factor of thousand more physical qubits to form a single logical one due to:
  - Quantum Error Correction to create a few stable logical qubits out of a lot of (unstable) physical qubits.
  - Error mitigation to enable quantum advantage in the near term.
- **Lack of business-relevant quantum computing algorithms:** The prime challenge is to identify quantum algorithms that solve real business problems more efficiently than classical ones. To a large extent this is due to the (typically omitted) fact that one cannot directly map a business question to a quantum algorithm. One first must associate the business question at hand with a specific mathematical problem or algorithm, or classes thereof. Only then is it possible to evaluate whether there exists a suitable quantum algorithm and how many qubits are minimally needed to run it.
- **Efficient integration and communication of quantum and classical computing** is crucial, as most quantum algorithms only speed up specific steps in a larger classical algorithm. To fully realize the potential of quantum computing, the quantum and classical steps need to be efficiently combined. This requires developing new techniques for integrating quantum and classical computing, as well as optimizing the communication between the two types of systems.
- **Quantum computers cannot really »store« information:** Due to the inevitable decoherence time and the “no cloning” theorem[11], a quantum computer cannot store data for a duration longer than the computation process. Consequently, quantum algorithms must be designed in such a way that the data is processed, and the results obtained within the time frame allowed by the coherence time of the qubits. This imposes a fundamental limitation on the size and complexity of the problems that can be solved using quantum computers.

[11] The "no cloning" theorem in quantum mechanics states that it is impossible to make a perfect copy of an unknown quantum state. This means that it is not possible to simply copy the data in a quantum computer to a different location for storage, as is done in classical computers.

2 Neuromorphic Computing

Dr. Johannes Leugering (Fraunhofer IIS), Bert Klöppel (T-Systems), Ferdinand Pscheidl (Fraunhofer EMFT), Villads Pors Schjelde (Infineon Technologies AG), Dr. Mira Wolf-Bauwens (IBM)

2.1 Short Description

Neuromorphic Computing is a multidisciplinary field that aims to build sensors, processors, and algorithms based on principles observed in neurobiology. Most neuromorphic systems are designed according to the special requirements of neural network models: Neural networks are composed of many individually simple neurons that all operate in parallel and communicate only via direct synaptic connections. Likewise, neuromorphic systems typically feature many simple processing elements that run concurrently and communicate via point-to-point connections instead of shared memory. These basic features, which mimic the morphology of biological neural systems, give the field its name and distinguish it from conventional general-purpose computer architectures.

Although building neuromorphic circuits originated in the intent of modelling biological processes in the human brain in detail to learn more about it, most current neuromorphic computing does not aim to do that. It rather tries to adopt some of the characteristics of the human brain to design algorithms and hardware, that brings power efficiency and latency of computing close to the human brain while solving typical machine learning tasks like image classification or voice detection with a human-level performance. The main aspects of the biological model that are leveraged are binary and asynchronous spike-based communication and continuous time computation, which are often implemented using analog, mixed-signal or asynchronous digital circuits.

Technology Readiness Level

While the neuromorphic computing field is still dominated by academic research, several start-ups and research division of larger companies have demonstrated prototypes and first products in recent years. In Germany, several research organizations, universities and spin-offs have also announced neuromorphic products, but so far only few products are commercially available of the shelf.

Practical maturity should further discriminate between sensing (e.g., event-based sensors using neuromorphic principles), inference (e.g., classification of patterns) and training (extracting implicit knowledge by observation): Neuromorphic sensors, in particular event-based vision, were among the first applications of neuromorphic computing and are already commercially available off the shelf (maturity 10). Inference is also quite advanced, (maturity 6-7, with expected prototypes and first products within the next five years), but training solely done by neuromorphic chips is still in a quite early research stage (maturity 2-4). Therefore, hybrid scenarios (technical split between training and inference) are likely to be run first. Most early use cases can be expected for relatively stable edge scenarios, perhaps in combination with neuromorphic sensors.

2.2 Benefits

- Higher energy efficiency (see below)
- Execution speed
- Robustness against noise, local failures, and device variability
- Enhancing efficiency and enabling growth of network size for DNN applications.

2.3 Energy Consumption Considerations

From the energy efficiency perspective, neuromorphic computing is significantly better than classical, synchronous running implementations. Studies claim a factor between 4 and 16 using the same semiconductor technology as classical processor chips[12].

Alternative semiconductor materials are also under investigation, aiming especially to improve memory density (in particular for non-volatile memories), integrate memory and computing more closely to reduce data transfers, or to improve efficiency at the expense of computing accuracy, which is possible for intrinsic fault tolerant approaches like neuromorphic computing[13].

Due to this improved computational energy efficiency, current neuromorphic use cases focus mostly on near edge intelligence with focus on computer vision and high dimensional signal processing.

2.4 Limitations

- For neuromorphic computing with spiking neural networks spike-rates are an inefficient way of representing numbers. This makes a direct conversion of deep neural networks inefficient. More efficient approaches are an active area of research but currently both software support and competitive network architectures are still missing. Therefore, neuromorphic chips are currently not competitive for many tasks that are done well by conventional computers[14].
- Neuromorphic computing is motivated by our current, incomplete understanding of the brain, which may not capture critical aspects of cognition such as the role of neurotransmitters and hormones, different neuron morphologies and cell types, and specific neural circuits. If cognition relies on these phenomena, current neuromorphic computing approaches might result in too incomplete approximations to be of practical use.

2.5 Applications

- **Medicine**: Neuromorphic computing is effective in receiving and responding to environmental data and can be made compatible with the human body using organic materials. This technology could be used to improve drug delivery systems

[12] Rao, A., Plank, P., Wild, A. et al. A Long Short-Term Memory for AI Applications in Spike-based Neuromorphic Hardware. Nat Mach Intell 4, 467–479 (2022).
[13] Emerging In-memory Computing for Neural Networks (fraunhofer.de)
[14] Are neuromorphic systems the future of high-performance computing? – Physics World

by releasing a drug when it senses a change in body conditions such as insulin and glucose levels. It could also enhance prosthetics by providing a more realistic and fluid experience through its ability to receive and respond to external signals.

- **Computer vision** [15]: Neuromorphic computing can be applied to event-based vision sensors that generate images similarly to the human eye. These sensors respond to changes in light intensity extremely quickly. This fast response time does no longer lead motion blur or delayed response, making them ideal for uses in robot vision and virtual and augmented reality technology.
- **Autonomous robots and self-driving cars** [16]**:** Due to their low energy consumption and short reaction times needed to process vast quantities of data, neuromorphic chips are well-suited for on-board processing in self-driving cars, drones, or satellites, i.e., where decisions need to be made autonomously and cloud-connectivity and power-budget are limited. In addition, neuromorphic chips offer potentially higher reliability in high noise environments. These benefits could make autonomous robots and driverless cars more economical, safer, and more suitable for varying environments.

2.6 Physical Area of Use

- **Edge:** Neuromorphic chips are particularly well-suited for use on the edge, as they can process data more efficiently and quickly with limited cloud-connectivity. Since edge applications also require less infrastructure and are thus cheaper to develop, a lot of neuromorphic computing research has focused on this sector so far.
- **Cloud** solutions are used for tasks that require more computational power and data processing capabilities. Several organizations pursue this direction by building wafer-scale chips or cloud computing centers based on neuromorphic architectures. Ultimately, the decision of whether to use neuromorphic chips on the edge or in the cloud will depend on the specific needs and constraints of the application.

2.7 OS / Software

Four basic features of spiking neural network (SNN) software can be distinguished: Design, training, simulation, and compilation to hardware. The existing software tools discussed later often provide multiple of these features.

Software to design SNNs needs to provide a specification format of describing a SNN architecture (e.g., PyNN) and should have an API to easily build up a SNN specification from reusable subnetworks (e.g., Nengo) and neuron models. A design tool could be extended by neural architecture search tools, which is a promising research direction to find more efficient SNN architectures that use less neurons but exploit spike timing and neuron dynamics in deep and recurrent networks.

A SNN training tool should be able to apply a programmable or selectable training algorithm to a specified SNN architecture. Existing training tools often rely on pretrained DNNs and convert them to SNNs with or without post-conversion optimization (e.g., snntoolbox), which reduces development time but tends to yield suboptimal results. Some training algorithms try to improve on this by using the

[15] PureSentry contamination detection | Cambridge Consultants
[16] Edge-AI & Neuromorphic – Efficient Processing of Time Series Data for Control & Prediction

precise spike timing to encode information, or by constructing a surrogate model that can be optimized with the same gradient-based methods as DNNs (e.g., snnTorch or Norse).

Due to the popularity of Deep Learning, a lot of the tools used for neuromorphic computing are derived from tools developed for DNNs (e.g. Pytorch), but these tools fail to leverage the full potential, since they don't capture the specifics of neuromorphic hardware such as analog or event-based processing. Therefore, the field will need to develop its own paradigms and software frameworks alongside the hardware itself.

When deploying neural networks on neuromorphic hardware, it is also necessary to implement a compiler that converts a given network and its parameters to programming instructions and byte-streams for the hardware. This can also include additional steps, such as a placer & mapper and router, which optimize the abstract model to the specific resources available in hardware.

2.8 Development Challenges

- **Being able to distribute large amounts of memory among many processors on a single chip:** Neuromorphic chips rely on the parallelization of a very large number of processes. For this to work efficiently, the required data needs to be placed in memory close to these processors, which requires new architectures and memory technologies. The availability of (cache) memory is already (one of) the main limiting factor(s) of conventional processor design and will be a major challenge for neuromorphic computing.
- **Developing competitive neural network architectures for neuromorphic devices:** To be successful, neuromorphic computing will have to compete against neural networks deployed on conventional hardware such as GPUs, which have been thoroughly optimized for each other. This can only be achieved by finding new algorithms and neural network topologies that are similarly optimized for neuromorphic devices.
- **Developing scalable messaging and interconnect architectures**: The parallelization of neuromorphic devices also requires a previously unseen amount of communication between these processors. Since communication can be highly irregular and sparse, completely new interconnect architectures and design tools are needed.
- **New algorithms, software and design tools** must be developed to incorporate and leverage the specifics of neuromorphic circuits. Current hardware is developed to optimize performance for current neural network algorithms, which are in turn optimized to run well on current hardware. Breaking this cycle requires a large effort of hard- and software co-development.
- **Device variability and noise of analog circuits**: For neuromorphic devices that rely on analog computation or new materials, the non-ideal effects and inherent device variability can be addressed on the application level, i.e., inside the neural network. This contrasts with conventional digital design, which aims to compensate for these effects already on the lowest circuit level. This means that design workflows must be changed, and new methods for verification must be developed.

3 Optical Computing

Dr. Michael Kissner (Akhetonics)

3.1 Short Description

Optical Computing promises to remedy some of the problems that plague electronics such as power consumption and data bottlenecks. The idea dates back to the 1950's, with von Neumann himself holding one of the first patents on an electromagnetic (”optical”) transistor[17] and the famed Bell Laboratories having created one of the first optical computers in the late 1980's[18]. But as with most technologies that aren't mature enough to enter the market, optical computing has gone through several hype and winter cycles over the years, and we are currently entering a new hype cycle.

There are two common approaches: The digital domain, which aims to replace the standard electronic processor and the analog domain, focusing on artificial intelligence (AI) acceleration. There is also an optical approach in QC which is out of focus of this section.

Digital:
By using all-optical transistors as building blocks, any computer architecture can be converted to an all-optical processor (be it von Neumann, Harvard, or other). This would make computing extremely energy efficient and a lot faster. Furthermore, by using multiple wavelengths, it is highly parallelizable without the need to add more transistors.

Currently, there are sufficient all-optical transistor concepts for which the research phase is completed. These can be used for optical digital computing. With switching speeds approaching the Petahertz domain, they promise a speed-up compared to current electronic transistors found in modern CPUs.

Currently, optical digital computing is the main approach to optical computing that promises to be all-optical without the need for electronics to manipulate data. Essentially, all-optical digital computing is at a similar point as electronic computing was in the 1970s, with similar large footprints and integration challenges. **The TRL is currently around 4.**

Analog
Optical analog computing is used to directly model mathematical functions in the optical domain. Using linear optical devices (such as Mach-Zehnder-Interferometers), it is possible to do vector-matrix multiplications (VMM) which can be implemented in AI systems. With more complex setups, Fourier transforms are also possible using linear devices. So far, non-linear mathematical operations have not been explored in detail.

The working principle exploits the wave properties of light: By constructively interfering two light beams one can »add« them and by attenuating (or dimming) one can »multiply« the first result by a constant. Using these two linear operations in

[17] US2815488A - Non-linear capacitance or inductance switching, amplifying, and memory organs - Google Patents

[18] A. Huang. “Architectural considerations involved in the design of an optical digital computer”. Proceedings of the IEEE, 72(7):780786, 1984

sequence allows for the famed vector-matrix multiplication. **The TRL is currently around 7, with the first companies releasing their products.**

The inability to perform operations other than linear vector-matrix multiplications has led to significant limitations in the use of analogue processors for AI acceleration. Existing approaches rely on continuous conversion between the electrical and optical domains, which compromises efficiency (see further "Limitations"). There are companies working on performing some of the non-linear computations optically. However, there is currently no roadmap to all-optical analogue computing, and all approaches require some conversion to the electronic domain at some point during the computation.

3.2 Benefits:

- **Extremely parallelizable:** Photons do not interact strongly with other photons or even some materials. This allows a high degree of parallelization, since it is possible to have multiple light beams with different wavelengths doing parallel computation without crosstalk.
- **Ultra-low power consumption and low latency:** Using almost perfectly transparent materials allows for extremely efficient waveguides that act almost like a superconductor.
- **130/250 nanometer (nm) fabrication nodes:** No leading-edge manufacturing processes are required.

3.3 Energy Consumption Considerations

Based on technology prototypes aiming on the key optical benefits – massive parallelism and ultra-high speed – used for AI-relevant tasks, considerable increase in energy efficiency could be expected. Some basic research calculated and empirically verified improvements ranging from 1 to 3 orders of magnitude (factor 10 up to 1000). Lower primary energy consumption per operation and drastically reduced computation time are main reasons for this improvement. Due to the very early state of maturity, especially the very high figures must be handled carefully. Finally, optical computing, like quantum computing, needs significant effort and therefore energy for classical computing power for data preparation and ingest plus later extraction and interpretation.[19]

3.4 Limitations

- **Digital**
  Low light-light interaction: To create an all-optical transistor, two light beams must interact with each other. This can only be done using highly specialized materials, long interaction lengths or very intense light beams.
- **Analog**
  Electronic processing bottleneck: Not many operations that can be performed using a completely linear PIC. This has severely limited the use of these analog processors to AI acceleration, since all modern neural networks also require a non-linear

[19] Photonic tensor cores for machine learning | Applied Physics Reviews | AIP Publishing

activation function. So far, the approach has been to repeatedly convert between the optical domain (to do the linear operation) and the electronic domain (to do all the non-linear operations). Converting between optical and electronic domains repeatedly leads to power and speed losses.

3.5 Applications

- **Digital**
  General Purpose: Use-cases include anything that a regular CPU, GPU or ASIC can currently do.
- **Analog**
  Application Specific: Linear Mathematical Operations and Fourier Transforms are the current main use-cases.
  - **Deep learning**: VMM is the main computation component in deep learning, so AI accelerators mostly use the analog approach.
  - **Data encryption**: Fourier Transforms have proven to be quite useful in the field of fully homomorphic encryption (FHE), enabling secure data processing, which has thus far been held back by the speed of computation.

3.6 Physical Area of Use

- **Digital**: cloud and on-premises
- **Analog:** cloud and edge

3.7 OS / Software

Digital: Same as in electronics, OS / Software can be optimized for Optical Digital Computing. While it is expected that a different architecture is used and a new type of instruction set, the overall programming principle will be abstracted away from the developer (through libraries and compilers).

Analog: Operates as an accelerator and does not run Software directly. Custom Software used for tuning and setting the device.

3.8 Development Challenges

- **Very large footprint** of the IC due to low transistor densities, leading to larger processors at the moment.
- **Digital**
  - **Need for alternative architecture approaches:** von Neumann architecture as found in electronic digital computers is not well suited for the optical domain as one of the major benefits is the high data bandwidth achievable using light. Alternatives are being explored which have dropped out of favor in electronics, but are still interesting in the optical domain, such as the Harvard architecture, the pushdown automata or the finite-state machine.

  - **Chip integration** of non-linear photonics and the associated material development onto a photonic integrated circuit (PIC) to allow for scaling of the circuits.
  - **Optical memory** is still not as abundantly available as in electronics.
- **Analog**
  - **Scaling** without sacrificing latency and power and reducing the input/output bottleneck introduced by electronics.

4 Digital Annealing

Dr. Stefan Walter (Fujitsu)

4.1 Short Description

Digital Annealing systems are quantum-inspired special purpose hardware devices, designed to solve large combinatorial optimization problems very fast. The problems need to be formulated as an Ising model or equivalently as a Quadratic Unconstrained Binary Optimization (QUBO). Digital Annealing systems search for the lowest energy value of the energy function. In many cases the basis of the search algorithm is a Simulating Annealing approach which is enhanced using hardware techniques such as GPUs, FPGAs or ASICs, and/or quantum-inspired algorithms. The hardware approaches specifically tailored to solve Ising type problems make Digital Annealing systems a powerful and competitive computing technology.

Technology Readiness Level

Digital Annealing is a mature technology, proven by actual systems in operational environments. Therefore TRL 9 can be assigned.

4.2 Benefits

The key advantages of Digital Annealing *compared to quantum computing* approaches are:

- **Full connectivity among bits** allows for solving more complex and more realistic problem scenarios without an additional overhead of embedding the problem, i.e., all bits can be used to encode the problem.
- **High gradation** allows for high precision (high inter-bit coupling resolution) in the formulation of the combinatorial optimization problem.
- **Large problem sizes solvable:** Existing annealers can handle between 100 000 bits up to 10 million bits using a combination of hardware and software technologies.
- **No special environment needed:** Digital Annealer can be implemented in standard 19-inch rack enclosures used in data centers. Compared to most quantum computing approaches, no cryogenic or vacuum environment is needed.

4.3 Energy Consumption Considerations

A benefit of special purpose hardware is their relatively low energy consumption. Depending on the realization, their power consumption is comparable to the CPUs and GPUs. As in the case of traditional hardware, there is of course power needed for the environment, e.g., server power, data center cooling, etc.

4.4 Limitations

- **Special purpose:** Digital Annealing systems are restricted to solving combinatorial optimization problems only.
- **Only close to optimal solution:** Digital Annealing systems are based on simulated annealing. There is no guarantee for the global optimum of the optimization problem. However, often a close to optimal solution, obtained in a very short time, is sufficient. QUBO problems are formulated with binary variables. To represent integer and continuous variables slack bits are needed which then are no longer available for the problem description.

4.5 Applications

Digital Annealing systems, as their quantum counterpart Quantum Annealer, are capable of solving combinatorial optimization problems, i.e., searching for a minimum energy solution of a given energy function. The problems need to be formulated as an Ising model or equivalently as a Quadratic Unconstrained Binary Optimization (QUBO). Combinatorial optimization problems are very complex optimization problems (in many cases instances of NP-complete problems) often with a huge number of variables and as a consequence the search space is usually too large to search exhaustively by brute force methods. Prominent problem examples include the travelling salesman problem, graph partitioning, graph coloring, and Boolean satisfiability problems. Although being restricted combinatorial optimization problems, those kinds of problems represent a significant number of challenges in many industries, e.g.:

- **Mobility**: e. g. traffic flow optimization[20], scheduling
- **Life science**: e. g. drug design
- **Finance**: e. g. portfolio optimization, arbitrage optimization
- **Manufacturing**: e. g. production planning [21], transport planning, assignment problems, warehouse optimization

4.6 Physical Area of Use

- **Cloud**: with APIs providing suitable interfaces for access from the public internet.
- **On-premises**: in case of special requirements to the solution (such as security issues or latency and real time demands). On-premises operation requires no dedicated environment since most Annealing systems can easily be integrated in existing standard data center infrastructure.

4.7 OS / Software

All Digital Annealing systems have in common that they solve combinatorial optimization problems formulated as an Ising or QUBO model. In these types of problems, the whole optimization problem and its business logic is described by a

[20] MOZART: Traffic management through traffic signal control by Quantum-Inspired (fujitsu.com)
[21] Fujitsu, Toyota Systems leverage Fujitsu's Quantum-Inspired Digital Annealer to streamline automobile production sequence : Fujitsu Global

matrix containing the coupling between bits. Compared to other computing approaches and especially to quantum gate computing, there is no need to develop novel algorithms which are then used to solve certain types of problems. It is just the matrix which in most cases needs to be handed over, using an API, to the Digital Annealing device. The underlying algorithm (digital annealing, simulated annealing, or simulated bifurcation) then solves the optimization problem.

There are some SDKs to conveniently create Ising and QUBO polynomials or matrixes, such as Fujitsu's DADK or D-Wave's Ocean. Some quantum SDKs such as Qiskit are also able to handle QUBOs. However, any custom method to populate a matrix with weights reflecting the business problem is sufficient. Each Digital Annealing system comes with specific parameters (such as e.g., maximum allowed solver time, etc.) that can be adjusted using the API.

4.8 Development Challenges

- **Not every combinatorial optimization problem is suited:** Every linear (mixed/integer) optimization problem can be cast into QUBO form and can then be solved using a Digital Annealing system. In many cases, such an approach won’t provide the potential speed up or an increased solution quality. Therefore, it is of great importance to identify business problems benefitting from using a Digital Annealing system or allowing to think beyond the border line of limitations caused by linear solvers. In addition, it is essential to develop and efficiently formulate the QUBO such that the business logic is correct.
- **Additional resources (i. e. bits) required for problems using higher order polynomials to be cast into QUBO form:** Often business problems require the use of higher order polynomials in their formulation. To use Digital Annealing systems for such kind of problems, higher order polynomials need to be cast into QUBO form which requires additional resources (i.e., bits). As it is the case with Quantum Annealing, identifying the right problem to be solved is a crucial task and in any case, it needs to be tested whether the novel Digital Annealing approach leads to a lower time to solution or provides a better solution.

5 DNA Computing

Dr. Christoph F. Strnadl (Software AG)

5.1 Short Description

DNA computing is a subset of molecular or chemical computing involving DNA, RNA, or highly related molecules. DNA computing (in the wider sense [22]) means computing by using a) the structure of synthetic oligonucleotids (shorter strands of DNA or RNA[23]) to store information (**DNA storage)** and the related chemical machinery (e.g., enzymes, ribozymes, ribosomes, DNAzymes, and others) and mechanisms (e.g., hybridization or annealing[24], exonuclease digestion, PCR) to deliberately manipulate this information (**DNA computing** in the narrower sense).

There are several different computational models how to realize DNA computing such as

- **DNA strand displacement** (DSD) yielding amplifiers, Boolean logic gates, chemical reaction, networks, oscillators, molecular diagnostics, and neural networks: A single strand DNA molecule replaces an (existing) strand on a double-stranded DNA molecule and is later also moved.
- **DNA-based switching circuits** based on DNA logic gates up to more sophisticated DNA neural networks.
- **Sticker model** implementing a register machine. Here, one lets smaller strands of single-stranded DNA hybridize ("stick") to a larger (single strand) "memory strand" to encode numbers. Advantage here is that this works without any enzymes or PCR simply based on annealing, i.e., letting two single-strand DNA pieces connect to the complementary piece.
- **Surface-based DNA computing / DNA computing on a scaffold**: Here, DNA molecules are fixated ("sticked") onto an underlying substrate medium – as opposed to floating around in a suitable solution.
- **Combinatorial DNA approaches** where (very roughly speaking) potential solutions are encoded in DNA and then subject to an appropriate filtering process discarding all non-optimal (or non-fitting) solutions.[25]
- **DNA "origami"**: Automatic nanoscale DNA assembly and folding resulting in non-arbitrary two- or three-dimensional shapes ("tiles").
- **DNA "chips"**, i.e., microarrays of DNA.
- **DNA "navigators" and "DNA walkers":** A single DNA molecule "travels" all possible paths on a surface prepared (e.g., by DNA origami) for the problem to be solved. Both, autonomous and stepped (i.e., externally triggered) locomotion are achievable.

[22] When referring to "DNA computing" in general, it is meant to always include RNA computing and other variants as well. If we want to distinguish between RNA computing and DNA computing, this will always be specifically indicated in the text.
[23] Typical lengths reach from 20 to some hundreds of nucleotides.
[24] This is the reaction where to complementary strands of single-strand DNA reconnect or recombine into a double-strand DNA molecule. This is called "hybridization" for historic reasons and "annealing" because it happens if you cool down a hot, (so-called denaturalized) "soup" of DNA single-strand molecules.
[25] Adelman (1994) used this technique to solve the travelling salesperson problem.

- **Cellular DNA computing**, that is DNA computing directly in a (living) cell. This form of DNA computing has some promising applications in nanomedicine.
- **"Computing without computing"**, i.e., computing through stopping naturally automatically executing chemical processes to do something meaningful (e.g., ligation prevention).

Technology Readiness Level

Depending on the implementation of DNA computing, TRL may be assessed at 3 (experimental proof of concept) or 4 (technology validated in lab). DNA computing, as of today, has **not been validated in any relevant application environment outside** research – possibly except for some startups who are currently trying to establish TRL 5 (technology validated in relevant environment; industrially relevant environment in the case of key enabling technologies). For **DNA storage**, current write speeds are approximately **500 byte/s.**

5.2 Benefits

DNA computing

- **Parallelization**: Using sufficiently many DNA strands, a large class of NP-complete combinatorial problems (Hamilton path, Boolean satisfaction) can be solved very fast (e.g., by first encoding every potential combination and then selecting only the required subset). Scalability goes to 10^{18} DNA strands in 1 liter of water.
- **Biological error-correction mechanisms available** (in theory) for correcting copying and processing areas, limited to the natural pathways of ribonucleic acids in living cells.
- **extraordinary energy efficiency** at the level of 1 J/10^{19} operations (= 1 J/10 Eops = 1 J/10,000 Pops[26][27]). This is about one billion (= 1,000 million = 10^9) times more energy efficient than current day electronic devices.[28]

DNA storage

- Besides its incredible storage density (seven orders of magnitude (10^7) more than tape storage and three orders more than flash memory), DNA storage excels with a (evolutionary) proven track record as information bearer at longevity, durability (up to 1 million years), and energy efficiency (eight orders of magnitude (10^8) better than that of flash memory).

5.3 Limitations

DNA computing

- **Low signal propagation speed**: DNA reactions are slow and even simple operations typically complete in hours (instead of milliseconds). Only the huge parallelization will yield acceptable speed up (10^{18} processors / 10^4 seconds (= several hours) for 1

[26] Exa (E) = 10^{18}, Peta (P) = 10^{15}

[27] ops = operations per second

[28] The chemical reaction of hybridization (or annealing), i.e., letting two strands of single-strand DNA molecules, releases energy.

operation = 10^{14} ops = 0.1 Pops (Peta-ops). The current fastest supercomputers have 1,000 PFlops[29].

- **Weak scaling:** The amount of DNA required scales exponentially with the size of (most) problems even though the number of steps of the underlying chemical reaction network only increases polynomial.
- **Optimal for discrete problems but not continuous ones.** By construction, DNA/RNA computing is optimal for discrete problems even though many real-world problems involve continuous search spaces.

DNA storage

- **Access latency**: Currently at minutes or hours. Undoubtedly, this will be shortened, but there are intrinsic physico-chemical limits to what one can achieve[30]. Current implementations achieve up to **500 byte/s** throughput (for some MB of data). Prototypes are expected to run at a factor of 10,000 in mid-2023.

5.4 Application Areas

- **Combinatorial problems**: travelling salesperson problem (= Hamiltonian path problem); SAT (satisfiability problem) of Boolean formulas, graph coloring
- **Content-based similarity search** as opposed to exact search and information retrieval (e.g., based on a unique key). Here, similar features of records are mapped to DNA sequences which are more likely to hybridize.
- **Random-number generation** (RNG) provided by automated DNA synthesis, enabling a true random number output of approx. 0.3 MB/s based on an independent source of entropy that is air-gapped and orthogonal to other RNG sources. This entropy can be used directly or as a secure seed into a pseudo-RNG
- **Cryptography** including unbreakable encryption schemes, steganography (hiding information in some other medium)
- **Intrusion/attack detection**
- **Scheduling**
- **Clustering**
- **Bio-sensing and intelligent diagnostics of molecular-level conditions** (e.g., sickness, presence of pathogens or other biomarkers) of the human system
- **Nanomedicine**, e.g., providing *in vivo* (foremost) RNA computing systems implementing some Boolean logic for controlling cell-level or molecular pathways
- **DNA barcoding and product tagging**
- **Switchable “smart” materials controlled by logically processed signals**
- “**DNA of Things” (DoT)**: Based on 3D-printing of objects with memory, by encoding the desired information in DNA, encapsulating the DNA in silica particles and fusing the DNA-containing particles into various materials usable in 3D-printing, e.g., for providing a unique identity to the thing. Another example cold be bioelectronic

[29] Flops = floating point operations per second
[30] The limited speed of DNA computing's operations is fundamentally limited by physico-chemical laws of nature (e.g., mass action law) and cannot be overcome even theoretically as long as DNA computing (and storage) is confined to DNA, RNA, or other complex molecules.

devices such as biofuel cells controlled by external signals, or signal-controlled release processes for certain chemicals.

- Long-term highly energy efficient and fully biodegradable **information storage & archiving**

5.5 Physical Area of Use

- **Cloud**: Due to the (current) complexity of operating DNA computers including DNA storage[31], this will most likely result in a cloud computing (SaaS) model.

5.6 OS / Software

By its very design, DNA computing does not run on any common digital substrate where the terms “operating system” or “software” can be used in the ordinary sense. Nevertheless, research has been active to provide higher level abstractions in the form of **programming languages or simulators** on top of the individual molecular or chemical reaction pathways to facilitate controlling (“programming”) the essentially chemical test tubes or micro-reactors[32].

- **gro** (from “growth”) is an open-source software package that combines a distributed systems and parallel computing approach with the simulation of up to a few thousand bacterial cells growing in a 2D environment[33]. It is based on the behavior of Escherichia coli (bacteria) like microcolonies and tracks up to 10 generations before the computation is saturated.
- **Cello** is an open-source framework can be used to design computational DNA circuits that encodes the electronic logic circuits. The resulting complete DNA sequence can be executed as a circuit inside living cells, such as bacteria or viruses. Primarily, their database uses NOR and NOT gates to represent any logic function. It is based on Verilog.[34]
- **Visual DSD** is a domain-specific programming language (DSL) based on the DNA strand displacement (DSD) mechanism. It uses a web-based graphical interface to realize the computational circuit, in order to construct the reaction network by using the provided DNA species, while this approach benefits in analyzing the reaction network circuit without manual construction. It originated from the Microsoft and Duke University partnership on DNA computing. [35]
- **Visual GEC** is a programming language and software tool for designing genetic circuits.[36]

5.7 Development Challenges

- **Large-scale parallelization**: Due to the intrinsically low signal propagation speed of chemical reactions DNA computing must be parallelized to an unprecedented extent to yield acceptable computational throughput technical realization.,

[31] Today, this means running a state-of-the-art genetic laboratory.
[32] Sawlekar, Rucha, and George Nikolakopoulos. "A Survey of DNA-based Computing Devices and their Applications." 2021 European Control Conference (ECC). IEEE, 2021.
[33] Specification and Simulation of Synthetic Multicelled Behaviors Seunghee S. Jang, Kevin T. Oishi, Robert G. Egbert, and Eric Klavins ACS Synthetic Biology 2012 1 (8), 365-374m
[34] Cello | CIDAR Lab
[35] Programming DNA Circuits - Microsoft Research
[36] Genetic Engineering of Living Cells - Microsoft Research

- **Automation of DNA manipulations.** Chemical reactions governing DNA computing are currently often performed manually by lab technicians using an array of semi-automated lab equipment (e.g., pipette dispensers, etc.). All of this will have to be fully automated (e.g., by suitable robots) before larger scale commercial DNA computing may be accomplished.
- **Lowering the costs:** The current cost base for manipulating DNA strands in a lab needs to be reduced by a factor of 1 million (e. g. through automatization or better error-correction) to make it feasible.
- **Mapping of business problems:** DNA encoding of problem data is often unique to one problem and cannot easily be transferred to different problems (the so-called "DNA word design" problem). To limit copying errors, DNA sequences should be unique with a good 3D-structure (hairpin or linear), which further reduces encoding flexibility. It is also more difficult to encode continuous problems or problems with more than a single solution, although there are some promising results seemingly overcoming these problems (in theory).
- **Compounding of experimental errors**: manipulating DNA strands (as of to-date) is an error-prone process (scales with the number of steps) and needs active error correction at every step. This is typically done by replicating DNA storage or the computations as the biological error-correction mechanisms typically are not available 1:1 for computations.
- **Limited computational complexity**: The realm of computations accessible to DNA computing seems to be limited and much narrower than even classical computing. For instance, the encoding of data types (e.g., a stack) is difficult; modularity of a DNA computing algorithm is also difficult to achieve. Furthermore, for error-free operations, DNA designs should feature specific distribution of the underlying bases[37] reducing design flexibility.
- **Discrete problems with a single optimum favored**: By construction, DNA computing runs on a discrete substrate even though many real-world problems exhibit continuous search spaces. Additionally, problems often lack a single optimal solution but feature multiple solutions. This needs to be sufficiently recognized and suitably reflected in the algorithms applied to solve this type of problems.
- **Costs**: Currently, the costs of DNA computing render it uneconomical. Costs (roughly) need to go down by 6 orders of magnitude (= 10^6 = 1 million) to render it feasible.

[37] The percentage of some of the four bases (typically G and C out of the four bases A, C, G, and T) needs to be sufficiently high to lead to stable 3-dimensional structures (hairpin or linear formation) of the DNA strands.

6 Further computing approaches

Dr. Roman Bansen (Agentur für Innovation in der Cybersicherheit), Dr. Christoph F. Strnadl (Software AG)

- **Smartdust**
  Smartdust usually refers to a system of many tiny microelectromechanical devices such as sensors or robots. They are operated wirelessly on a computer network and are distributed over an area to perform tasks, usually sensing, and communicate via radiofrequency38 . There are, however, theoretical concepts for these tiny devices to contain a simple processor. Intelligent control could use mesh networking of these devices in different ways from those of traditional computer networks.

- **Brain-Computer interface**

  Brain-machine interfaces or brain-computer interfaces refer to the direct communication between the brain's electrical activity and an external computer. While usually used for augmenting or repairing human cognitive or sensory-motor functions and technically not a computing approach, it might radically change our way to operate and interact with computers in the future.[39] [40] [41]

- **Spintronics**
  The term spintronics, or spin transport electronics, generally describes any kind of application or solid-state device which does not only use the fundamental electric charge of electrons but also their intrinsic spin. This adds a further degree of freedom to be utilized. Technically, this includes a whole range of already commercialized devices like the read heads of magnetic hard drives or magneto-resistive RAM and even certain approaches in quantum computing the focus with respect to future computing lies on the development of spin-based transistors, which could have several potential advantages over classical transistors.

- **Bio-neuronal networks**

  In contrast to the artificial neural networks in neuromorphic computing, this approach uses biologically real nerve cells (neurons) as an interface to silicon-based electronics. Currently, the Australian startup Cortical Labs is looking to commercialize this »dishbrain« technology. Similar approaches are being pursued by researchers at the University of Texas in Austin[42].

- **Approximate computing**
  The umbrella term »approximate computing« includes several different

[38] Iyer, V., Gaensbauer, H., Daniel, T.L. et al. Wind dispersal of battery-free wireless devices. Nature 603, 427–433 (2022).
[39] Science & Tech Spotlight: Brain-Computer Interfaces | U.S. GAO
[40] 3 Brain-Computer Interface Technology Trends (patsnap.com)
[41] Frontiers | Progress in Brain Computer Interface: Challenges and Opportunities (frontiersin.org)
[42] Kireev, D., Liu, S., Jin, H. et al. Metaplastic and energy-efficient biocompatible graphene artificial synaptic transistors for enhanced accuracy neuromorphic computing. Nat Commun 13, 4386 (2022).

computation techniques that – in contrast to a classical computer – return possibly inaccurate results. They could be used for any kind of application where an approximate result is sufficient for its purpose. In many non-critical scenarios, approximation within certain boundaries could provide significant gains in performance and energy, while still achieving acceptable result accuracy. Approaches and strategies for approximate computing include, amongst others, the use of approximate arithmetic circuits, approximate storage and memory or software-level approximation[43, 44].

- **Associative machine**
  An associative machine is a freely programmable machine composed of associative memories. In contrast to a classical Von Neumann processor, an associative machine is not built around an arithmetic logic unit but consists of associative units. Usually, associative matrices are used as associative memories. This way, the associative machine acquires properties of fault tolerance, which makes it particularly interesting for pattern recognition, completion and extraction. Associative machines require completely different programming paradigms, called associative programming, where a program line is associated with its successor and data can be queried in a fault-tolerant manner[45, 46].

- **In-memory computing (IMC) / In-memory processing (IMP) / Processing in memory (PIM)**
  In today's widespread Von Neumann computing architecture, the memory interface is a major hurdle for overall processing speed, slowing down many calculations. The IMC architecture eliminates the detour via the processor's main memory: Processing and storage of data takes place on the same chip. As stored data is accessed much more quickly when it is placed in RAM or flash memory, in-memory processing allows data to be analyzed much faster. Moreover, complex algorithms can potentially be processed with significantly lower power consumption. In-memory computing is sometimes also classified as a neuromorphic computing approach. The way it functions, and the combination of fast computing power and low energy consumption make IMC particularly interesting for AI applications[47].

- **Field Programmable Gate Array (FPGA)**
  An FPGA is an integrated circuit consisting of (i) input/output blocks and (ii) user-configurable logic blocks, which are linked together via (iii) programmable interconnections. Therefore, engineers can configure FPGAs according to desired requirements after they have been manufactured and deployed (hence, field programmable, contrary to ASICs – application specific ICs).[48] Despite their size, cost and power disadvantages compared to ASICs, they are used for hardware acceleration (e. g. encryption, video format conversions, AI/ML algorithms) or to enhance security in heterogeneous and/or changing environments where the deployment of classical ICs (CPUs, GPUs) or ASICs no longer is cost-effective.

[43] W. Liu, F. Lombardi and M. Schulte, "Approximate Computing: From Circuits to Applications [Scanning the Issue]," in Proceedings of the IEEE, vol. 108, no. 12, pp. 2103-2107, Dec. 2020

[44] Approximate Computing | SpringerLink

[45] assoziativmaschine.de

[46] Assoziativcomputer: Hildesheimer Grüße an die NSA - Digital - FAZ

[47] S. Ghose, A. Boroumand, J. S. Kim, J. Gómez-Luna and O. Mutlu, "Processing-in-memory: A workload-driven perspective," in *IBM Journal of Research and Development*, vol. 63, no. 6, pp. 3:1-3:19, 1 Nov.-Dec. 2019

[48] Michael Mattioli. 2022. FPGAs in Client Compute Hardware: Despite certain challenges, FPGAs provide security and performance benefits over ASICs.

7 Further Reading

Quantum Computing

- WEF insight Report »State of Quantum Computing: Building a Quantum Economy«
- Leitfaden »Quantentechnologien in Unternehmen«
- Quantum Computing White Paper (Software AG)
- Industry Quantum Computing Applications, QUTAC Application Group

Neuromorphic Computing

- The Femtojoule Promise of Analog AI - IEEE Spectrum
- J. Leugering, »Neuromorphe Hardware – Hardware für neuronale Netze«, DESIGN & ELEKTRONIK, 7/2020

Optical Computing

- Pierre Ambs, »Optical Computing: A 60-Year Adventure«, Advances in Optical Technologies, vol. 2010, Article ID 372652, 15 pages, 2010.
- Miller, D. Are optical transistors the logical next step?. *Nature Photon* **4**, 3–5 (2010)

Digital Annealing

- Mohseni, N., McMahon, P.L. & Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nat Rev Phys* **4**, 363–379 (2022).
- Kochenberger, G., Hao, JK., Glover, F. *et al.* The unconstrained binary quadratic programming problem: a survey. *J Comb Optim* **28**, 58–81 (2014).

DNA Computing

- Katz E (2020): DNA Computing: Origination, Motivation, and Goals – Illustrated Introduction
- Meiser, Linda C., et al. »Synthetic DNA applications in information technology.« Nature Communications 13.1 (2022): 1-13.

Bitkom represents more than 2000 companies of the digital economy. Their annual turnover for IT and telecommunications services alone is 190 billion Euro, including exports of 50 billion Euro. Bitkom members employ more than 2 million people in Germany. These members number more than 1,000 small and medium-sized enterprises, over 500 start-ups and virtually all global players. They offer software, IT services, telecommunications or Internet services, manufacture devices and components, are active in the area of digital media or form part of the digital economy in some other way. 80% of the companies have their head office in Germany, while 8% each come from Europe and the USA, and 4% from other regions. Bitkom promotes and drives the digital transformation of the German economy, and champions broad involvement of society in digital developments. The objective is to make Germany one of the world's leading digital business locations.

Publisher
Bitkom e.V.
Albrechtstr. 10 | 10117 Berlin

Contact Person
Dr. Natalia Stolyarchuk | Head of Future Computing & Microelectronics
T 030 27576-187 | n.stolyarchuk@bitkom.org

Responsible Bitkom Working Group
AK High Performance Computing and Quantum Computing

Authors and Contributors:
Dr. Roman Bansen (Agentur für Innovation in der Cybersicherheit), Dr. Jannis Ehrlich (Fraunhofer IWM), Dr. Michael Kissner (Akhetonics), Bert Klöppel (T-Systems), Dr. Johannes Leugering (Fraunhofer IIS), Dr. Oliver Oberst (IBM), Ferdinand Pscheidl, Villads Pors Schjelde (Infineon Technologies AG), Dr. Christoph F. Strnadl (Software AG), Dr. Stefan Walter (Fujitsu), Dr. Mira Wolf-Bauwens (IBM).

Titelbild
© Scanrail – stock.adobe.com

This publication constitutes general non-binding information. The contents reflect the opinion of Bitkom at the time of publication. Although the information has been prepared with the greatest possible care, no claim to factual correctness, completeness and/or up-to-dateness exists. This publication cannot take into account the specific circumstances of individual cases. Any use of this information is therefore the responsibility of the reader. Any liability is excluded. All rights, including the reproduction of excerpts, are held by Bitkom or the respective rights holders.