



OPEN ACCESS

EDITED BY

Mats Brorsson,
University of Luxembourg, Luxembourg

REVIEWED BY

Srikanth Thudumu,
Institute of Applied Artificial Intelligence and
Robotics (IAAIR), United States
Sarah L. Harris,
University of Nevada, Las Vegas, United States

*CORRESPONDENCE

Nikolaos Alachiotis
✉ n.alachiotis@utwente.nl

RECEIVED 07 February 2025

ACCEPTED 26 May 2025

PUBLISHED 24 June 2025

CITATION

Alachiotis N, van den Belt S, van der Vlugt S,
van der Walle R, Safari M, Endres Forlin B, De
Matteis T, Al-Ars Z, Jordans R, Sousa de
Almeida AJ, Corradi F, Baaij C and
Varbanescu A-L (2025) FPGA innovation
research in the Netherlands: present
landscape and future outlook.
Front. High Perform. Comput. 3:1572844.
doi: 10.3389/fhpcp.2025.1572844

COPYRIGHT

© 2025 Alachiotis, van den Belt, van der Vlugt,
van der Walle, Safari, Endres Forlin, De Matteis,
Al-Ars, Jordans, Sousa de Almeida, Corradi,
Baaij and Varbanescu. This is an open-access
article distributed under the terms of the
[Creative Commons Attribution License \(CC BY\)](#). The use, distribution or reproduction in
other forums is permitted, provided the
original author(s) and the copyright owner(s)
are credited and that the original publication
in this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

FPGA innovation research in the Netherlands: present landscape and future outlook

Nikolaos Alachiotis ^{1*}, Sjoerd van den Belt ¹,
Steven van der Vlugt ², Reinier van der Walle ²,
Mohsen Safari ³, Bruno Endres Forlin ¹,
Tiziano De Matteis ⁴, Zaid Al-Ars ⁵, Roel Jordans ^{6,7},
António J. Sousa de Almeida ¹, Federico Corradi ⁶,
Christiaan Baaij ⁸ and Ana-Lucia Varbanescu ¹

¹University of Twente, Enschede, Netherlands, ²Netherlands Institute for Radio Astronomy (ASTRON), Dwingeloo, Netherlands, ³SURF, Amsterdam, Netherlands, ⁴Vrije Universiteit Amsterdam, Amsterdam, Netherlands, ⁵Delft University of Technology, Delft, Netherlands, ⁶Eindhoven University of Technology, Eindhoven, Netherlands, ⁷Radboud Radiolab, Institute for Mathematics, Astrophysics and Particle Physics (IMAPP), Radboud University, Nijmegen, Netherlands, ⁸QBayLogic B.V., Enschede, Netherlands

Field programmable gate arrays (FPGA) have transformed digital design by enabling versatile and customizable solutions that balance performance and power efficiency, yielding them essential for today's diverse computing challenges. Research in the Netherlands in both academia and industry plays a major role in developing new innovative FPGA solutions. This survey presents the current landscape of FPGA innovation research in the Netherlands by delving into ongoing projects, advancements, and breakthroughs in the field. Focusing on recent research outcome (within the past 5 years), we have identified five key research areas: (a) FPGA architecture, (b) FPGA robustness, (c) data center infrastructure and high-performance computing, (d) programming models and tools, and (e) applications. This survey provides in-depth insights beyond a mere snapshot of the current innovation research landscape by highlighting future research directions within each key area; these insights can serve as a foundational resource to inform potential national-level investments in FPGA technology.

KEYWORDS

field programmable gate array (FPGA), applications, survey, high performance computing (HPC), architecture, security and reliability, programming models

1 Introduction

The current surge in computationally intensive workloads, such as artificial intelligence (AI) and high-performance computing (HPC), creates an unprecedented need for ever-increasing computational capacity. Most modern-day applications and technologies are data-driven and must process large volumes of data at high speeds or operate under stringent time constraints. Emerging technologies, such as distributed sensor networks in radio astronomy (Lenkiewicz et al., 2018) and next-generation sequencing in genetics (Schmidt and Hildebrandt, 2017), enable the accumulation of vast amounts of data, further intensifying the need for more computational power. Keeping up with the ever-increasing demand for computational capacity requires excessive amounts of energy, at a time when the need to reduce unsustainable energy consumption is urgent. AI applications,

powered by large language and generative models, are rapidly increasing in scale and complexity, further increasing their energy demand (Strubell et al., 2019). Given the increasing amounts of data to be processed and the high computational requirements of modern-day applications, the energy needed to sustain these applications can often not be supplied exclusively by renewable sources of energy (Bhattacharya and Qin, 2020; Katal et al., 2023). This raises the need for energy-efficient hardware able to facilitate the next generation of data-driven applications.

To enable the deployment of modern applications that need to process large amounts of data faster, energy-efficiently, and/or in real time, hardware acceleration is necessary. Various hardware technologies can be used for this purpose, such as graphic processing units (GPUs), field-programmable gate arrays (FPGAs), and application-specific integrated circuits (ASICs). A GPU is a massively parallel architecture that comprises numerous small processing cores (Owens et al., 2008), making it well suited for computationally intensive workloads such as graphics rendering and parallel computing, particularly for vector-processing operations. GPUs are widely deployed for accelerating AI/machine learning (ML) model training. An ASIC offers the highest performance and energy efficiency for an application by implementing in hardware only the logic actually needed by the target application. Designing and fabricating an ASIC, however, requires an extensive design process and is extremely costly (Yang et al., 2023). These limitations can be alleviated by using an FPGA. FPGAs are frequently used instead of GPUs as well, because of their Input/Output (I/O) capabilities, for example optical links or direct network connections.

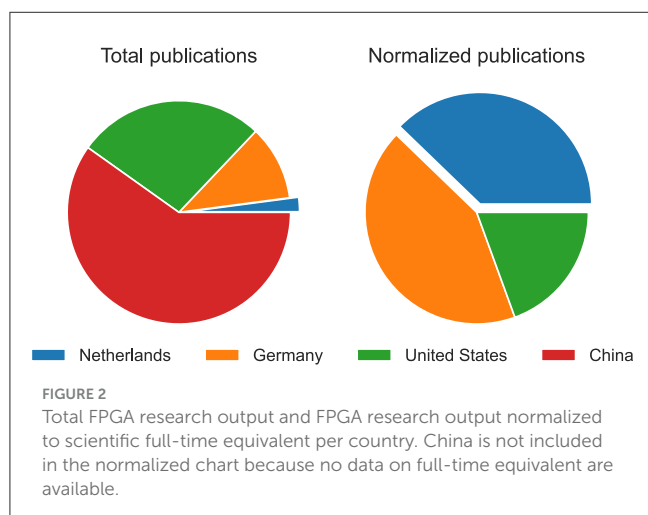
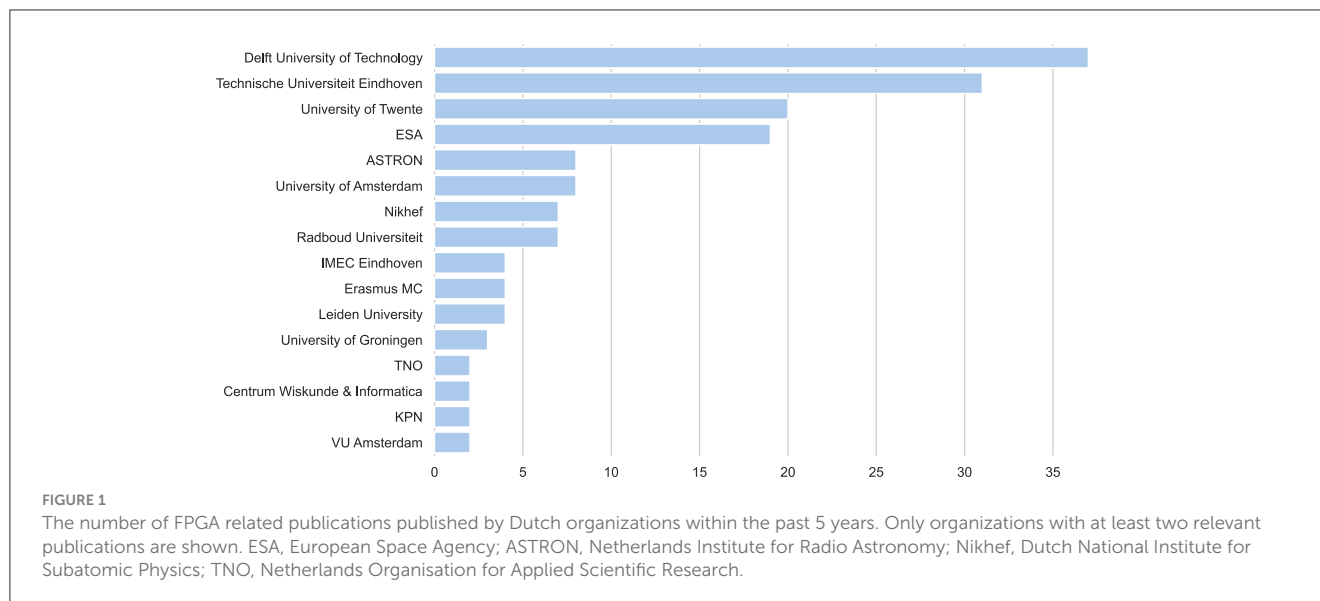
FPGAs have revolutionized digital design and prototyping due to their versatility and adaptability to varying computational requirements. Unlike ASICs, FPGAs are reprogrammable and can be reconfigured multiple times after manufacturing, allowing for cost-effectively realizing highly efficient custom computer architectures. FPGA technology relies on an array of configurable logic blocks that are interconnected through programmable routing channels; these elements' configuration results in the implementation of specialized digital circuits tailored to the specific requirements of a particular domain, application, or even workload. Eliminating unnecessary micro-architecture-level components and optimizing the design for a specific domain or application leads to considerably higher performance and energy efficiency than general-purpose processors without the prohibitively high costs for manufacturing an ASIC. FPGAs, however, present certain challenges as their programming involves a low-level design at the register-transfer level (RTL) using hardware description languages (HDLs), for example, VHSIC (Very High Speed Integrated Circuit) Hardware Description Language or Verilog. Unlike CPUs and GPUs, which can be programmed using high-level languages, using HDLs has a steeper learning curve and requires an extensive background in computer architecture, digital logic, and hardware design principles. To mitigate these challenges, substantial research has focused on raising the level of abstraction through high-level synthesis (HLS) tools, which allow developers to describe hardware behavior using C, C++, or similar languages. However, HLS also introduces certain challenges, such as the need to guide HLS tools to generate efficient architectures and understand how high-level

constructs map to the hardware. Moreover, effective FPGA development still demands careful consideration of hardware-specific concerns, for example, timing closure, parallelism, and resource constraints.

1.1 Context

The unique ability of modern FPGA technology to balance performance and power efficiency through customization makes it a pivotal technology in addressing the diverse and evolving challenges of today's highly heterogeneous computing landscape. Research in the Netherlands plays a major role in developing new innovative FPGA technologies. The country has a strong reputation for research excellence, particularly in areas such as technology, engineering, agriculture, environmental science, and health-care. Dutch universities and research institutions are actively involved in collaborative international projects, fostering a culture of knowledge exchange and cooperation, while Dutch companies contribute to global initiatives and advancements in fields like sustainable energy, water management, and digital technology. Besides universities and industry, public institutes such as the European Space Research and Technology Center (ESTEC), the Dutch National Institute for Subatomic Physics (Nikhef), and the Netherlands Institute for Radio Astronomy also conduct FPGA-related research. Figure 1 provides an overview of Dutch organizations that conduct research in this area and the number of recent publications per organization. The figure shows that research by universities and ESTEC constitutes the majority of the relevant published work. Two private companies, IMEC NL and KPN, have also contributed with at least two publications, indicating that private organizations also play a role in the scientific development of FPGA technology. Overall, FPGA technology is an active research field in the Dutch scientific community, with relevance in academia and industry.

Advancing FPGA technology aligns with European goals and strengthens the position of the Netherlands in a future where efficient large-scale computing will be increasingly important. In the past decade, the Netherlands has made significant scientific contributions to a large number of European research projects, accelerating the development of various technologies. Within the European Horizon 2020 funding program, the Netherlands has been successfully participating in scientific European projects when normalized to the number of scientific person-year effort in the country. To provide a broader perspective into the extent of involvement of the Netherlands in FPGA research, Figure 2 compares the FPGA research output of the Netherlands with three major contributors of FPGA research worldwide: Germany, China, and the United States. The number of publications per country has been obtained by adapting the Scopus query provided in Section 2 to the respective countries. The left chart in the figure shows the absolute research output per country, while the right chart shows the research output normalized to the number of scientific full-time equivalents (FTE) per country, obtained from reports published by the European Union (Eurostat, 2022) and the U.S. government (Education Statistics, 2022). While the absolute



number of FPGA research contributions from the Netherlands is smaller than those of other major contributors, when normalized to academic employment, the amount of Dutch contributions is comparable to that of Germany and higher than that of the United States.

Within Horizon 2020, the Netherlands has made a comparatively high contribution to the program pillar focusing on Excellent Science (Nederland en Horizon 2020, 2024). Current major strategic goals outlined by the European Commission are developing autonomous technologies, such as AI, and the development of technologies to combat climate change (Hessels et al., 2024). Both of these goals align with developing energy-efficient hardware acceleration. The Netherlands also houses the ESTEC, which is the main center for research and development of the European Space Agency (ESA). Space applications are another specific field in which FPGA technology has a prominent role. The flexibility and performance that FPGAs offer align with the needs of space applications. Recently, space-grade FPGA technology has been developed as a European research effort, strengthening the application of FPGAs in space projects (European Commission,

2024). Overall, Dutch contributions to research and technology are important on both European and global scale in various scientific and technological domains.

1.2 Survey focus

Harnessing the full potential of advanced FPGA-based systems demands more than just comprehending the inherent capabilities and limitations of FPGA technology. It necessitates a deep understanding of how these attributes align with and serve the diverse computational requirements across various domains and applications. To this end, we survey the present landscape of FPGA innovation research in the Netherlands. By delving into the ongoing projects, advancements, and breakthroughs in the field, this survey aims to provide valuable insights that go beyond a mere snapshot of the present landscape. More importantly, it aspires to serve as a foundational resource, possibly guiding future national-level investments in FPGA technology. By understanding the strengths, challenges, and emerging trends in FPGA research, policymakers, researchers, and industry stakeholders can make informed decisions about allocating resources and shaping strategies that will contribute to the continued growth and impact of FPGA technology. This survey endeavors to be a pivotal tool in fostering a forward-looking approach, ensuring that the nation remains at the forefront of FPGA innovation and leverages its potential for future technological advancements. Considering the relevance of FPGA technology in the European and Dutch contexts, there is ample reason for the Netherlands to invest in FPGA technology.

2 Survey method

We performed the literature collection in this survey study as a three-step process. First, we employed the online literature search tool Scopus (Scopus, 2024) to gather relevant literature from several publishers. Second, we performed a more in-depth search for

articles by specific publishers. Third, a selection of conferences and journals were manually checked for relevant papers in proceedings and articles, respectively, to include publications that were possibly omitted in previous steps. Finally, all duplicates were filtered out, and we manually reviewed the resulting publications for relevance.

2.1 Step 1: Scopus-wide search

We used Scopus to search for publications from more than 7,000 publishers. We used the following search criteria: (a) the abbreviation “FPGA” appears in the title, abstract, and/or keywords; (b) at least one of the authors has a Dutch affiliation; and (c) the publication year is 2019 or later. These translate into the following Scopus query: TITLE-ABS-KEY (FPGA) AND AFFILCOUNTRY (Netherlands) AND PUBYEAR > 2019. The search was performed using this query on the October 30, 2023, which delivered a list of 186 papers. As of this article’s publication, this query is expected to yield an increasing number of papers as more papers aligning with the search criteria are published. To put things in context, we also conducted a worldwide search for papers on FPGAs published in the last 5 years. The search resulted in a list of over 27,600 titles from 84 countries with at least 10 publications, indicating that the Netherlands contributes roughly 1% to the worldwide research on FPGAs and ranks on position 22 in terms of output volume. In Europe, Dutch FPGA research ranks 10th in terms of output volume.

2.2 Step 2: select publishers

In addition to the Scopus-based search, we conducted an in-depth search for relevant papers in proceedings and articles published by the Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE). Publications (co-)authored by Dutch institutes, published in 2019 or later, containing both the terms “FPGA” and “HPC” were queried for. Compared to the initial Scopus-based search, this new search is more specific in terms of keywords (by using one additional keyword) but more inclusive because it considers the entire text for keyword matches. We used advanced search tools provided by IEEE and ACM^{1,2} and obtained a list of 65 articles (including possible duplicates).

2.3 Step 3: select conferences and journals

We conducted a final search within topic-relevant conferences and journals; we restricted the scope of this manual search to conference proceedings and journal issues published in the last 5 years. The following conferences and journals were considered in this step:

- International Symposium on Field-Programmable Gate Arrays.
- International Symposium on Field-Programmable Custom Computing Machines.
- International Conference on Field Programmable Logic and Applications.
- International Conference on Field-Programmable Technology.
- ACM Transactions on Reconfigurable Technology and Systems.

This search delivered a list of 21 publications from Dutch-affiliated authors (including possible duplicates).

2.4 Final literature selection

We verified all collected publications for relevance and excluded duplicates. This process yielded a total of 212 relevant publications, which we further classified into research themes. A significant number of these papers (120, over 55%) focus on FPGA applications. Thus, we selected six applications for further review, based on the number of relevant papers and the significance of FPGAs in the application; the selection resulted in a shortlist of 49 application papers. These applications are discussed in Section 8. Other applications that are not reviewed in depth include network processing (Kundel et al., 2021), cryptography (Massolino et al., 2020), control systems (Moonen et al., 2021), and weather prediction (Singh et al., 2020). Section 3 introduces the research themes by which all reviewed papers were categorized.

3 Research themes

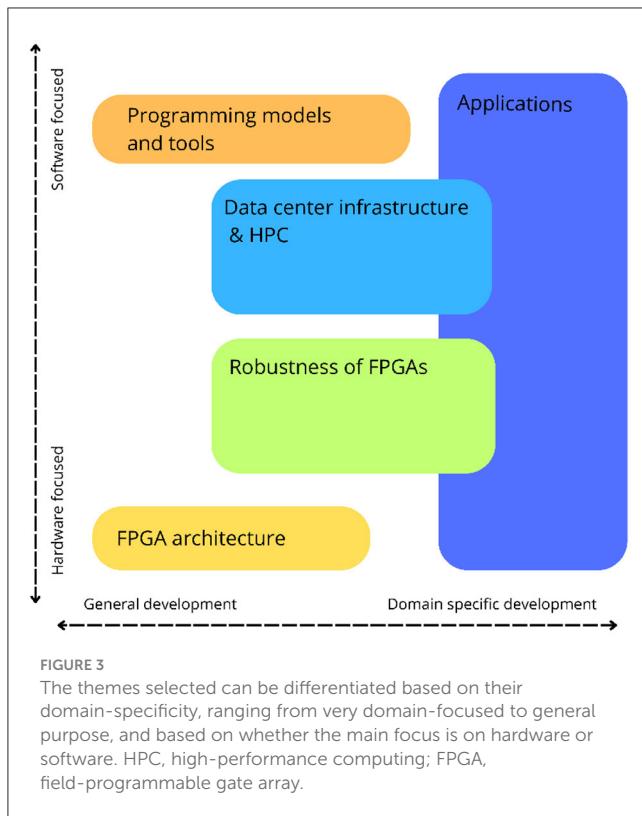
Based on the literature found through the process outlined in Section 2, articles with similar subjects or covering similar themes of research are grouped together. The themes are selected in such a way that most publications can be exclusively divided into one of the themes; that is, the themes should not have significant overlap. Furthermore, the themes should effectively separate domain-specific research and more generally applicable research. Considering these requirements the following list of themes is selected:

1. FPGA architecture.
2. Robustness of FPGAs.
3. Data center infrastructure & HPC.
4. Programming models & tools.
5. Applications.

Figure 3 illustrates how these themes cover both general and domain-specific development, and shows whether a theme is hardware- or software-focused. The theme “Applications” focuses on research applying FPGA technology to domain-specific problems. This can be in the form of hardware architectures for domain-specific applications, as well as software tools enabling FPGA technology in a specific domain of research. The themes “Programming models and tools” and “FPGA architecture” focus on research and development of solutions that are generally applicable in a wide range of domains, while the “Robustness of

¹ ACM Advanced Search. (2024). Available online at: <https://dl.acm.org/search/advanced>.

² IEEE Xplore Help (2024). Available online at: <https://ieeexplore.ieee.org/Xplorehelp/searching-ieee-xplore/advanced-search>.



FPGAs” and “Data center infrastructure & HPC” themes feature both hardware- and software-focused research. Moreover, these themes focus on a narrower selection of FPGA applications and can, therefore, be considered domain-specific.

Based on common subjects in each theme, the themes are further organized into subcategories. Table 1 shows the subcategories and prevalence of each subject based on the number of published articles. Finally, the most influential articles, based on the highest number of citations within each category (Google Scholar), is shown. This selection excludes survey publications.

Figure 4 shows the relative publications per theme out of the 141 selected publications, and Figure 5 illustrates an overview of publications per theme for each organization with more than one publication. It is clear that most major contributors to FPGA research publish mostly application-specific research. Out of the major contributors, Delft University of Technology focuses more on the “Data center & infrastructure” domain, while Eindhoven University of Technology is a larger contributor to the “FPGA architecture” theme. Following is a brief description of each theme:

- **FPGA architecture:** This research theme covers literature related to the design of novel digital hardware architectures. Efficient architectures, fast on-chip memory access, coarse-grained hardware design, and partially reconfigurable hardware are covered in this theme.
- **Data center infrastructure and HPC:** This theme includes literature on FPGAs used in HPC environments. This covers papers on the rapid processing of big data, and research toward distributed computing infrastructures deploying FPGAs. Furthermore, research focusing on employing FPGAs for processing communication between computing nodes using optical links is also covered here.

- **Programming models and tools:** This theme covers literature related to tools and models used to program FPGAs, ranging from research on HLS tools to tools that enable accessible hardware acceleration of conventional software. This theme also features research efforts on tools for accurate performance prediction of synthesized FPGA solutions.
- **Robustness of FPGAs:** This theme covers literature regarding the reliability and resilience of FPGAs to specific environments. Specifically, this subject is common in environments where resilience to radiation is prevalent. Furthermore, this theme expands on the security of FPGAs regarding cyberattacks.
- **Applications:** The literature on specific applications using FPGAs is more extensive than that of the other themes. This is expected because FPGAs can be applied in various fields, whereas the advancement of FPGA architectures and development tools is generally a more narrow area of research. Machine learning has been the main focus in recent years.

4 FPGA architecture

This section discusses how FPGAs enable advanced computer architecture concepts that eliminate common bottlenecks in existing computer architectures. Here, we discuss three main architectural topics that have received significant research attention in recent years, and their applications: near-memory computing (NMC), coarse-grained reconfigurable arrays (CGRAs), and network-on-chip (NoC).

4.1 Near-memory computing

Near-memory computing (NMC) is a promising approach to mitigate memory access bottlenecks in HPC systems. NMC bridges the widening gap between the computation capabilities of processors and the latency and bandwidth limitations of memory subsystems. This is specifically important for applications in which conventional Central Processing Unit (CPU) architectures struggle due to complex data access patterns, limited data reusability, and low arithmetic intensity. These challenges are attributed to the inefficiencies in conventional memory hierarchies. FPGAs play an important role in enabling NMC due to their reconfigurable nature, allowing for the co-location of computation with memory. This co-location significantly reduces the data movement that typically limits performance and increases energy consumption in traditional architectures. FPGAs facilitate the implementation of customized, application-specific data paths and processing units adjacent to memory, enabling more efficient data handling and processing.

4.1.1 Research topics

There are multiple research topics being investigated by Dutch organizations that focus on harnessing NMC to enhance the performance and energy efficiency of applications with complex data access patterns, specifically through the use of FPGA-based accelerators.

TABLE 1 Overview of highly cited papers per category, with number of published articles per theme and category in parentheses. HPC, high-performance computing; FPGA, field-programmable gate array; ESTEC, European Space Research and Technology Center; Nikhef, Dutch National Institute for Subatomic Physics.

Themes and Categories	Highly cited publications	Dutch affiliation
FPGA architecture (11)		
Near-memory processing (4)	Singh et al., 2021a	Eindhoven University
Coarse-grained reconfigurable architecture (4)	Wijtvliet et al., 2019	Eindhoven University
Network-on-Chip (3)	Ribot Gonzalez and Nelissen, 2020	Eindhoven University
Data center infrastructure and HPC (40)		
Big data processing and analytics (22)	Peltenburg et al., 2019	Delft University of Tech.
Distributed computing (5)	Bielski et al., 2018	Sintecs B.V.
Optical hardware communication (9)	Yan et al., 2018	Eindhoven University
HPC (4)	Katevenis et al., 2018	MonetDB Solutions
Programming models and tools (15)		
Programming models and frameworks (8)	Peltenburg et al., 2020b	Delft University of Tech.
Performance prediction (7)	Yasudo et al., 2018	University of Amsterdam
Robustness of FPGAs (26)		
Reliability (12)	Du et al., 2019	ESTEC
Hardware security (14)	Labafniya et al., 2020	Delft University of Tech.
Applications (49)		
Machine learning (12)	Rocha et al., 2020	IMEC NL
Astronomy (11)	Ashton et al., 2020	University of Amsterdam
Particle physics experiments (7)	Fernandez Prieto et al., 2020	Nikhef
Quantum computing (5)	Philips et al., 2022	Delft University of Tech.
Space (9)	Barrios et al., 2020	ESTEC
Bioinformatics (5)	Malakonakis et al., 2020	University of Twente

4.1.1.1 Genomics

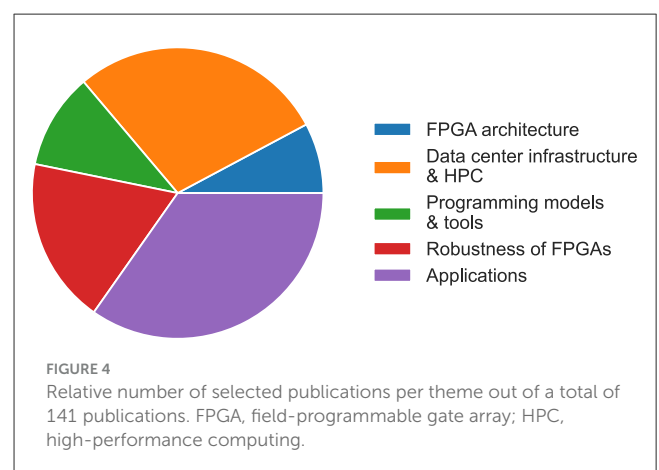
[Singh et al. \(2021a\)](#) propose improved computational solutions for genome analysis, which is an excessively expensive application in genomics. The study proposes a pre-alignment filter based on NMC to reduce the amount of data to be processed.

4.1.1.2 Weather prediction

Solving large-scale weather prediction simulations suffer from limited performance and high energy consumption due to complex irregular memory access patterns and low arithmetic intensity. [Singh et al. \(2022\)](#) propose an NMC solution with an FPGA and high-bandwidth memory (HBM) that is 5.3 times faster than the baseline CPU solution.

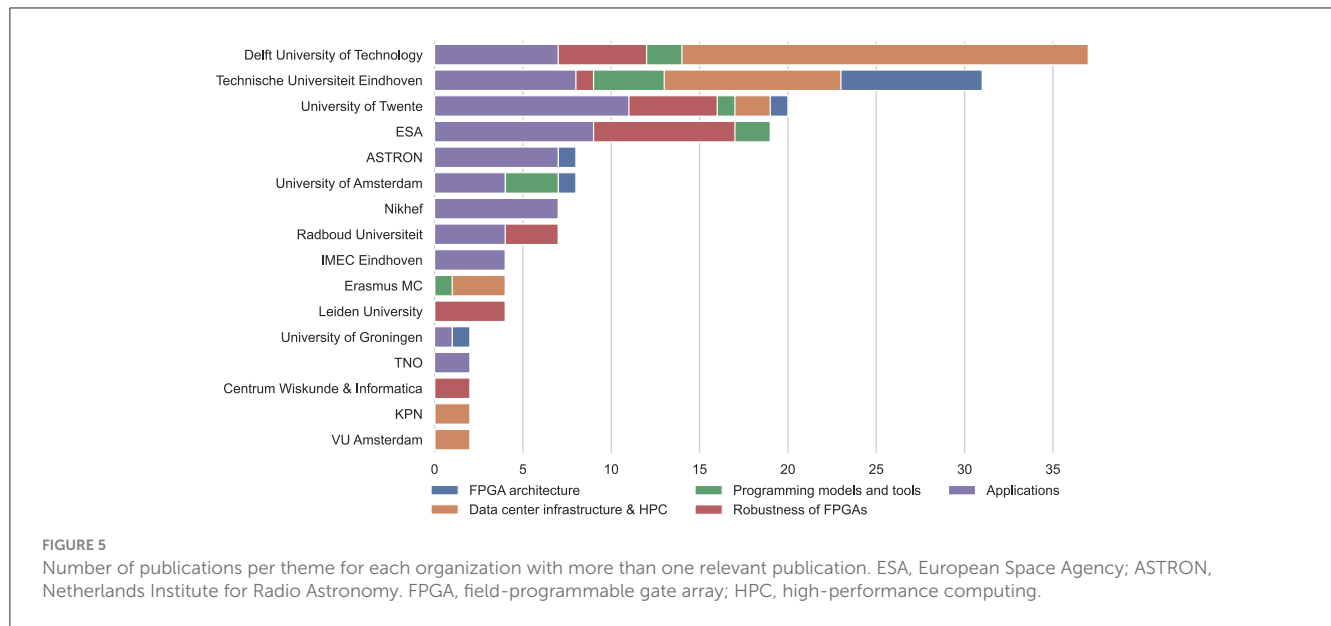
4.1.1.3 Phylogenetics

Reconstructing large evolutionary relationships among organisms is computationally expensive due to extensive calculation of probabilistic likelihood functions, which is a data-intensive, memory-bound operation. [Alachiotis et al. \(2021b\)](#) describe an NMC solution that addresses the problem of workload distribution in a disaggregated data center architecture, and results in improved, scalable performance, and nearly 3 times higher energy efficiency than multicore processing.



4.1.2 Future directions

Near-memory computing (NMC) is an emerging field that promises to have an impact on various applications domains in which data access patterns hinder performance including but not limited to, big data analytics, ML, and scientific simulations. Research continues to explore using FPGA-based accelerators for



a broader range of applications. The integration of emerging memory technologies like high bandwidth memory with FPGAs is anticipated to further enhance bandwidth and reduce the energy footprint of such applications (Singh et al., 2019). Furthermore, investigations to improve the NMC architecture by optimizing the granularity of data movement, refining memory hierarchies to match specific application patterns, and exploring the precision tolerance of various computational kernels for additional efficiency gains are ongoing. FPGAs' adaptability to various data access and processing patterns makes them attractive platforms for such optimizations (Singh et al., 2019). Moreover, as the NMC field progresses, there is a growing emphasis on software frameworks and tools that simplify the deployment of NMC solutions, making them accessible to a wider range of developers and applications (Abrahamse et al., 2022).

4.2 Coarse-grained reconfigurable architecture

Coarse-grained reconfigurable architecture (CGRAs) aim at improving the programming efficiency of FPGA platforms. Many applications do not require the bit-level reconfigurability that is provided by modern FPGA platforms. The trend toward coarser granularity is evident even in commercial FPGA architectures, as the number of digital signal processing (DSP) blocks and other specialized hardware accelerators increases with each new generation.

4.2.1 Research topics

Multiple CGRA-related topics are explored by Dutch organizations, both at application-mapping level and toward implementing novel accelerator structures.

4.2.1.1 Application-mapping templates

Charitopoulos et al. (2021) provide a template architecture that allows for device configuration at a coarser granularity

(e.g., arithmetic operations), thereby significantly reducing the architecture exploration space and significantly simplifying the work of application mapping tools. Due to the limited architecture exploration space, it becomes more likely that the application mapping process will find an efficient mapping. This work introduced a novel mapping improvement algorithm that decreased average distance per cell by up to 70% over original placements for various applications.

4.2.1.2 Accelerator architecture evaluation

Once a template architecture is designed, it can also be implemented directly on a chip, either as a standalone ASIC or as an accelerator integrated into a larger system. This further reduces the reconfiguration overhead compared to FPGAs and can result in overall improved performance (Wijerathne et al., 2022; Wijtvlief et al., 2019; de Bruin et al., 2024). FPGAs, however, are still used in this context, mainly for prototyping, obtaining activity traces of applications for energy estimation, and exploring the template architecture design.

4.2.2 Future directions

Many modern FPGA platforms are part of a larger system-on-chip (SoC). These are increasingly introducing hardware accelerators for common tasks (e.g., ML). CGRAs can play a key role in this context as they promise to maintain most of the high flexibility that an FPGA offers while providing improved energy consumption and performance for many applications. Furthermore, the more constrained architecture template of a CGRA alleviates the application-mapping burden, thereby benefiting human developers working on application mapping while simplifying the task of high-level synthesis tools.

4.3 Network-on-chip

Network-on-Chips (NoCs) play an important role in the development of complex, multicore SoCs, addressing

the limitations of traditional bus architectures in scalability, bandwidth, and power efficiency. As embedded systems require advanced functionalities, leading to an increased number of processing elements (PEs) integrated into SoCs, the demand for efficient communication architectures has escalated. NoCs provide a scalable solution by facilitating parallel data transmission among multiple PEs through router-based packet switching networks. Complex hardware designs on FPGAs rely on NoCs to improve the efficiency of data communication. In addition, FPGAs serve as a popular platform in NoC development due to their reconfigurable nature, allowing for evaluating, prototyping, and testing of various NoC architectures with different topologies and routing strategies. This flexibility is crucial in optimizing NoCs for specific applications, including real-time systems, where timely data delivery within predefined deadlines is essential.

4.3.1 Research topics

A number of papers have been published by Dutch organizations that focus on enhancing NoC architectures for real-time applications on FPGA platforms. These papers introduce novel NoC designs that aim to address the challenges of designing NoCs on FPGAs while minimizing hardware resources and power consumption.

4.3.1.1 In-order NoCs

Traditional deflection-based NoCs, while efficient in resource usage, struggle with maintaining the order of flits, which is critical for various applications. IPDeN (González et al., 2022) addresses this challenge and ensures in-order flit delivery by incorporating a small, constant-size buffer in each router. This architecture requires $3\times$ fewer hardware resources compared to virtual-channel-based solutions.

4.3.1.2 FPGA-optimized NoCs

To address the increasing communication volume between computation nodes integrated in FPGAs, Ribot González and Nelissen (2020) propose HopliteRT*, anNoC architecture for real-time systems optimized for the constraints of FPGA platforms. HopliteRT* supports priority-based routing and a novel network topology to improve worst-case packet traversal time. Experiments show that the proposed NoC allows for an improvement of at least two times the traversal time of high-priority packets for negligible additional hardware costs.

4.3.1.3 Predictable NoCs

Worst-case communication time analysis in NoCs is an important topic in real-time system design but is rather challenging due to the unpredictable nature of NoC communication. nDimNoC (Ribot González et al., 2021) is a new D-dimensional NoC that provides real-time guarantees for SoC systems. nDimNoC uses the circulant topologies' properties to ensure bounded worst-case communication delays. It requires 5 times less silicon than routers that use virtual channels.

4.3.2 Future directions

Future research on FPGA-based NoCs is expected to focus on further optimizations for real-time applications and efficient

hardware utilization. New, more sophisticated routing policies and buffer management strategies can be explored to further enhance throughput and reduce latency. Commercial architectures such as the AMD Versal Adaptive SoC already contain hard NoC infrastructure. Given that data movement is a dominant challenge for many applications, it is likely that similar advancements will be adopted in other FPGA devices. Additionally, NoC architectures can be further developed to support dynamically changing workloads and applications with varying timing properties without necessitating network reconfiguration. This adaptability is important in addressing the evolving needs of real-time and embedded systems, where the workload can vary over time. Furthermore, intelligent routing algorithms, such as those based on ML, are a promising avenue for future NoC research. This approach could lead to more adaptive and efficient NoC designs capable of self-optimization based on current network conditions and application requirements.

5 Data center infrastructure and HPC

In this section, we overview research and activities related to FPGAs in HPC and data centers. Using FPGAs in HPC and data centers raises a fundamental question: in which position can FPGAs play a significant role in the workflow of HPC and data centers? One historical position of FPGAs in this ecosystem is in the network and communication. This is due to the direct I/O connection capabilities of these devices, allowing them to attach to network components (e.g., switches and routers) through a dedicated network stack directly implemented on FPGAs. In the Microsoft Catapult project (Caulfield et al., 2016; Putnam et al., 2015), for instance, FPGAs are used in the Microsoft Bing search service as a reconfigurable logic layer between network switches and servers.

Another straightforward answer to this question is to deploy FPGAs as dedicated accelerators/co-processors. Due to their reconfigurability and flexibility, FPGAs enable hardware-software co-design and implementation of domain-specific applications. Moreover, FPGAs as accelerators facilitate spatial programming, for example, data flow implementations, to reduce data movement (from memory to compute units) compared to the traditional, control-based procedural programming (Licht et al., 2022). As an instance of FPGAs as accelerators, Fugaku extends its supercomputer center with a scalable FPGA-cluster system (Sano et al., 2023). Another example is through the AMD university program (Aad et al., 2021), where some research institutes around the world deploy Heterogeneous Accelerated Compute Clusters. These clusters support adaptive computing by incorporating FPGAs in their compute nodes to accelerate scientific applications. Despite substantial efforts to improve the programmability of these devices for software developers and end users, achieving high performance through an optimized implementation of an algorithm remains a significant challenge in most cases. Intel and VMware, in collaboration with research institutes and universities, established the Crossroads 3D-FPGA Academic Research Center (see footnote 3) to reconsider and find a permanent solution for this question. Their ambition is to define a fixed role for FPGAs as a central function in future data center servers. From their

perspective, FPGAs will serve as the core of servers, acting as data movement and transformation engines between the network, traditional compute units, accelerators, and storage.

The aforementioned activities indicate the important, yet ambiguous role of FPGAs in the future of HPC ecosystems and data centers. Bobda et al. (2022) provide an overview of existing academic and commercial efforts of employing in data centers. Among the commercial efforts, we observe that major data centers, such as Microsoft, Alibaba, Amazon, Baidu, and Huawei, benefit from FPGAs in their infrastructures (Firestone et al., 2018; Putnam et al., 2015; Caulfield et al., 2016; Ernst, 2020; Xilinx Case Study, 2024). Although this question is still open, and various *ad-hoc* solutions have been proposed, one important factor will be the economic advantage; it will depend on whether these solutions can deliver more performance with less energy consumption and lower costs across a range of applications. The rest of this section presents an overview of the FPGA research landscape in the Netherlands by organizing big data and high-performance processing into four categories: big data processing and analytics (Section 5.1), distributed computing (Section 5.2), optical hardware communication (Section 5.3), and HPC (Section 5.4).

5.1 Big data processing and analytics

Several studies in Dutch academia have assessed the domain of big data processing and analytics (Hoozemans et al., 2021a; Peltenburg et al., 2021b; Rellermeyer et al., 2019; Fang et al., 2020b), identifying opportunities for FPGA accelerators and describing the challenges faced in the wide adoption of FPGA technology. Peltenburg et al. (2021b) identify the programmability of the accelerators, the portability of the implementation, the interface design to the data, and the infrastructure for data movement to and from the accelerator and across kernels running on the accelerator as the main challenges. Solutions leveraging various existing technologies have been proposed, for example Apache Spark,³ Apache Arrow Flight,⁴ the IBM POWER architecture (Sadasiyam et al., 2017), and OpenCAPI,⁵ while the applications of FPGA accelerators in this domain involve database search (Fang et al., 2020b), real-time data analysis (Chrysos et al., 2019), graph-based processing (Iosup et al., 2023; Prodan et al., 2022), high-frequency trading (Chen et al., 2021), DNA analysis (Voicu and Al-Ars, 2019), and ML (Rellermeyer et al., 2019).

5.1.1 Research topics

Several challenges in using FPGAs effectively as accelerators for big data processing and analytics have been addressed by the Dutch research community.

5.1.1.1 Interface design and infrastructure

Many data structures used in databases do not map well onto the architecture of an FPGA, for example, the alignment of data format or the method of data retrieval, thus making processing

on an FPGA inefficient. Apache Arrow Flight (see footnote 1) organizes data movement in a coherent and transparent way across various systems and applications. Fletcher (Peltenburg et al., 2021b; Ahmad et al., 2022) extends Apache Arrow Flight with FPGA support and defines inter-kernel infrastructure between processing kernels implemented in FPGA, accelerating different application performances by 1.3–49 times. Complementary work provides (on-line) data conversion from the widely used Parquet (Peltenburg et al., 2020a) and JavaScript Object Notation (JSON) (Peltenburg et al., 2021a) formats to Arrow.

5.1.1.2 Frameworks and tooling

Several frameworks have been developed to ease programming FPGA accelerators for big data processing and analytics. Fletcher integrates FPGA accelerators with tools and frameworks that use Apache Arrow as their back end (Peltenburg et al., 2019). The open stream-oriented specification Tydi-spec (Peltenburg et al., 2020b) and language Tydi-lang (Tian et al., 2022) was developed to specify and implement complex, dynamically sized data structures onto hardware streams. SparkJNI enables heterogeneous CPU-FPGA systems based on the Apache Spark unified engine for large-scale data analytics (Voicu and Al-Ars, 2019), while Abrahamse et al. (2022) extend the ThymesisFlow (Pinto et al., 2020) memory disaggregation system with a framework leveraging IBM POWER9 and FPGA accelerators.

5.1.1.3 Compression and decompression

Both the data storage size and the bandwidth required to move data to and from storage present significant challenges in efficiently deploying accelerators. Data compression is used to mitigate these challenges by reducing both storage size and bandwidth requirements. However, the compression and decompression processes require considerable resources, and efforts have been undertaken to (de)compress data on FPGAs to either enable direct data processing on the FPGA itself or facilitate data transfers to another system component for further processing or storage. Solutions based on various (de)compression algorithms have been presented, such as Snappy (Fang et al., 2019), LZ77 (Fang et al., 2020a), and Zstd (Chen et al., 2021). Hoozemans et al. (2021b) present an energy-efficient, FPGA-based co-processor that supports several decompression algorithms.

5.1.2 Future directions

Research to develop frameworks that enable the efficient use of FPGA accelerators for big data processing and analytics is ongoing. By adopting high-level workflows tailored to these tasks, FPGA accelerators are becoming increasingly applicable within general data center infrastructures and applications. We see the work referred to in Section 5.1 being continued, as well as being extended with other partners in the industry (Al-Ars et al., 2023; Cromjongh et al., 2023; Reukers et al., 2023; Groet et al., 2024). Furthermore, one can not overstep the current rise of ML and AI, which, when applied to big data processing and analytics (Rellermeyer et al., 2019) can benefit from FPGA acceleration (Nechi et al., 2023). The previously listed technologies being developed in the Netherlands can enable using FPGAs as accelerators for ML and AI in big data analytics.

³ Apache spark. (2024). Available online at: <https://spark.apache.org/>.

⁴ Introducing apache arrow flight. (2019). Available online at: <https://arrow.apache.org/blog/2019/10/13/introducing-arrow-flight/>.

⁵ Opencapi consortium. (2024). Available online at: <https://opencapi.org>.

5.2 Distributed computing

Distributed computing involves the deployment of multiple computing nodes in parallel to increase performance and solve large computational problems. While research on distributed computing involving CPU and GPU nodes is well established, FPGAs' emergence as a new type of computational resources and accelerators within data center infrastructures introduces a new and challenging area of research. Dutch academia has mainly focused on applications that use distributed multi-FPGA systems for large-scale graph processing (Sahebi et al., 2023) and deep neural networks (DNNs) (Alonso et al., 2021).

5.2.1 Research topics

Several research topics have been investigated by Dutch researchers.

5.2.1.1 Communication overhead

Reducing communication is a key factor in distributed computing and, in particular multi-FPGA systems. By reducing communication overhead, computation time and latency decrease, and efficiency increases. To reach this goal, researchers propose interconnection frameworks to establish flexible, reliable, efficient, and custom communication protocols in multi-FPGA systems (Salazar-García et al., 2020; Salazar-García et al., 2021; Salazar-García et al., 2022). In addition to reducing latency, these frameworks are designed to work with various topology schemes and different FPGA technologies. Salazar-García et al. (2022) report up to 97.25% computational efficiency from the use of a multi-FPGA interconnection framework for distributed computing.

5.2.1.2 Partitioning and performance scaling

To increase performance in multi-FPGA systems, Alonso et al. (2021) propose an open-source, distributed resource partitioning and allocator tool on FPGAs for data flow architectures targeting DNN inference; it works in conjunction with the FINN compiler (Umuroglu et al., 2017). The authors show that their methodology enables super-linear scaling of throughput by benefiting from model parallelism and direct FPGA FPGA communication. Sahebi et al. (2023) propose a (multi-FPGA) framework for large-scale graph processing. The framework uses an offline partitioning mechanism and relies on Hadoop to map the graph onto the underlying hardware. The authors show that graph partitioning using an FPGA architecture results in better performance on large graphs that include millions of vertices and billions of edges. Their results indicate significant speedups over state-of-the-art CPU, GPU, and FPGA solutions.

5.2.2 Future directions

Several challenges exist in distributed computing using multi-FPGA systems, thereby necessitating further research in this direction. For instance, overcoming communication barriers and designing protocols for FPGA-FPGA communication is an ongoing research domain. Moreover, at the application level, developing (standard) Message Passing Interface (MPI)-like

collective communication libraries for multi-FPGA systems would be beneficial. Also, more case studies are needed to investigate and design efficient partitioning and workload distribution schemes for FPGA resources. Therefore, to bring ease of use and automation to distributed computing on FPGAs, developing libraries and tools is crucial.

5.3 Optical hardware communication

Optical hardware communication is at the forefront of addressing the critical challenges faced by contemporary data center network (DCN) infrastructures, such as bandwidth limitations, latency issues, and scalability concerns. Optical communication is a viable alternative to conventional electrical data pathways, offering significant improvements in terms of efficiency and performance. The integration of FPGAs into optical communication systems has been a key development, providing the necessary flexibility and speed for dynamic network reconfiguration and management.

5.3.1 Research topics

Exploring optical hardware communication using FPGAs encompasses a variety of innovative research topics covered by Dutch organizations.

5.3.1.1 Optical wireless datacenter networks

Zhang et al. (2022) have developed an optical wireless-DCN architecture that promises enhanced flexibility and scalability for DCNs, supporting high-speed optical packet-switching transmissions. FPGA-based switch schedulers are used to control the implementation based on semiconductor optical amplifier-based wavelength selectors and arrayed waveguide grating routers.

5.3.1.2 Disaggregated optical networks

The DACON project (Guo et al., 2022) introduces a disaggregated, application-centric optical network that uses hybrid optical switches and FPGA-based controllers, resulting in improved application performance and reduced latency. The tested applications using DACON achieve a runtime that is 1.46 times faster, with a latency reduction of 21%. Moreover, DACON reduces energy consumption by 13.4%–31.1%.

5.3.1.3 Low-latency edge networks

The Electro-Optical Communication group at Eindhoven University of Technology (TU/e) has proposed an edge DCN architecture that employs photonics and FPGA-based supervisory channels to achieve microsecond-time control and deterministic latency (Santana et al., 2023).

5.3.1.4 Nanosecond optical switching

A novel optical switching and control system has been designed to address the bandwidth bottlenecks of electrical switching, featuring a distributed network architecture with optical label channels and the optical flow control protocol (Xue and Calabretta, 2022).

5.3.1.5 Hybrid data center architectures

The HiFOST DCN architecture (Yan et al., 2018) integrates flow-controlled fast optical switches with modified top-of-the-rack switches, offering substantial improvements in latency and cost efficiency.

5.3.1.6 Beyond 5G networks

Santana et al. (2022) present a new edge cloud network design that uses FPGA-based controllers for rapid reconfiguration of optical networks, catering to the low-latency requirements of 5G applications and beyond.

5.3.2 Future directions

Through various ongoing developments, the field of optical hardware communication is poised for significant advancements. Efforts to mitigate Light Emitting Diode (LED) nonlinearity have led to the development of a Legendre-polynomials-based post-compensator optimized for FPGA implementation, enhancing the bit rate efficiency of high-speed visible light communication (VLC) systems (Niu et al., 2021). The introduction of a real-time FPGA-based implementation of a nonlinear LED model and post-compensator marks a substantial contribution to VLC technology, enabling high data rates over bandwidth-limited LEDs (Deng et al., 2022). A concurrency-aware mapping technique has been developed to reduce optical packet collisions in architecture-on-demand (AoD) network infrastructures, improving buffer utilization and execution time degradation in HPC systems (Meyer et al., 2018).

5.4 High performance computing

Benefiting from FPGAs in HPC applications is an active research area. Even though GPUs remain the most prevalent accelerator technology in HPC and AI-specific hardware is being increasingly adopted, FPGAs are also increasingly employed in HPC centers.

5.4.1 Research topics

Dutch institutes have been involved in European projects, for example ExaNeSt (Katevenis et al., 2018) and MANGO (Flich et al., 2018), to design large-scale heterogeneous compute systems. We can observe the important role of FPGAs in these projects, facilitating network communication or accelerating execution.

5.4.1.1 Architecture and system design

The ExaNeSt European project (Katevenis et al., 2018) deploys FPGAs as accelerators in a European exascale supercomputer based on low-cost, low-power Advanced RISC Machines Ltd (ARM) cores. They also employ an FPGA-based testbed for a low-latency, high bandwidth unified remote direct memory access (RDMA) interconnect, and present a custom FPGA-based switch to support inner-cabinet communications. The MANGO project (Flich et al., 2018) aims at addressing the PPP (performance, power, and predictability) space in HPC by exploring customizable and deeply heterogeneous accelerators. Their hardware concept consists

of general-purpose compute Nodes (GNs) with commercial accelerators such as Xeon Phi and NVIDIA GPUs, along with heterogeneous nodes (HNs). HNs are clusters of many-core chips coupled with customized heterogeneous computing resources, including high-capacity clusters of FPGAs.

5.4.1.2 Programming languages, tools, and applications

Within the ExaNeSt project, Katevenis et al. (2018) design a novel microarchitecture as top-of-rack switches. In one of their experiment, they port the OpenCL kernels of the molecular dynamics simulator LAMMPS (Plimpton, 1995) to FPGAs using HLS tools. They report that running the kernel on an FPGA requires 0.56 seconds while the 4 ARM cores requires 1.3 seconds. That is an improvement of more than a factor 2 in speed-up. Within the MANGO project, Flich et al. (2018) target three applications with significant Quality of Service (QoS) aspects: (1) online video transcoding, (2) rendering for medical imaging, and (3) error correcting codes in communication. The MANGO project relies on LLVM (Lattner and Adve, 2004) and their programming model is an extension of existing languages and libraries [e.g., OpenCL (Group, 2015)] for HPC by integrating the expression of new architectural features as well as QoS concerns and parameters. This is achieved by augmenting the runtime library Application Programming Interface (API) with new functions, pragmas and keywords to the existing HPC languages (e.g., clang C/C++ frontend).

5.4.1.3 Performance models

Combining the advantages of reconfigurability, dataflow computation, and heterogeneity results in reconfigurable dataflow platforms (RDPs) as a promising building block in next-generation, large-scale high-performance machines. RDPs rely on reconfigurable dataflow accelerators (RDAs) to realize multiple streaming pipelines, each comprising many parallel operations. Due to the heterogeneous hierarchy, RDPs' performance prediction is very challenging, in particular detecting bottlenecks within reasonable time and accuracy. Yasudo et al. (2021, 2018) propose a performance estimate framework for reconfigurable data flow applications named Performance Estimation for Reconfigurable Kernels and Systems (PERKS). It automatically extracts specific parameters from the application, hardware, and platform to calibrate the model. They use eight applications for their evaluation: AdPredictor (an online ML algorithm), N-body simulation, Monte Carlo simulation, sequence alignment, Asian option pricing, Jacobi solver, and regression/regularization solver. Their results show that PERKS achieves accuracy of 91% on these applications.

5.4.2 Future directions

Determining FPGAs' role in HPC necessitates more research from both data center architecture design and FPGA programming model. From a data center designs perspective, the positioning of FPGAs in the architecture of HPC centers needs more investigation. This also depends on the targeted application workflow and how FPGA can impact the most. From a user perspective, these devices' programmability is an important factor. Therefore, the gap between software developers and FPGA programming models and tools should be reduced further by using FPGA as mainstream HPC devices.

6 Programming models and tools

This section discusses approaches that boost developer productivity; Section 6.1 reviews programming models and frameworks that raise the abstraction level of describing hardware, while Section 6.2 presents techniques that predict performance of synthesized programs.

6.1 Programming models and frameworks

FPGA development is traditionally characterized by a steep learning curve, especially for non-experts. For this reason, HLS tools and, more generally, high-level programming models and frameworks have been proposed to increase productivity by raising the abstraction level. HLS tools became commercial products in the early 2010s. Since then, they have been used in various application domains, including, but not limited to, deep learning, multimedia, graph processing, and genome sequencing (Cong et al., 2022). HLS tools can reduce the average development time [up to two-thirds compared to RTL (Lahti et al., 2019)]. However, they still require considerable expertise to optimize the FPGA designs and achieve a quality of result that is on par with the one obtained through hardware description languages. For this reason, higher level programming models and frameworks are now being proposed. They allow developers to describe hardware in a more convenient formalism [e.g., Clash (Baaij et al., 2010),⁶ HeteroCL (Lai et al., 2019), PyLog (Huang et al., 2021), and DaCe (Ziogas et al., 2021), which currently support Haskell, Python or a Python-embedded DSL], and automatically, or via user-provided hints, generate optimized HLS/HDL descriptions.

6.1.1 Research topics

Several papers have been published by Dutch organizations focusing on reducing development time for hardware design using HLS programming models and tools.

6.1.1.1 Abeto framework

Sanchez et al. (2022) propose Abeto, a software tool for intellectual property (IP) management and workflow automation. Historically, no standard has been established for packing, documenting, and distributing IP core designs. This prevents their reusability, as each IP core has its unique learning curve and challenges for using them in an Electronic Design Automation (EDA) toolchain. Abeto allows the user to operate in a unified manner with heterogeneous IP cores and conveniently configure and launch the different stages of the IP workflow. To add an IP core, Abeto requires some auxiliary information to be provided: a database definition (containing information about the directory structure of the IP core) and a command dictionary (which includes the list of supported IP commands and how they must be executed). The tool has been validated against a subset of the ESA portfolio of IP cores,⁷ which constitutes a heterogeneous group of IP cores, demonstrating the tool's versatility.

⁶ Clash: a modern, functional, hardware description language. (2024). Available online at: <https://clash-lang.org/> (accessed May 15 2024).

6.1.1.2 Design flow for Gowin FPGAs

Vos et al. (2020) describe a method to create an open-source design flow for the Gowin LittleBee family of FPGAs. The design flow is based on well-known open-source tools such as Yosys and nextpnr, as well as the newly developed bitstream generator. The architectural details of the FPGA family were documented using input fuzzing and comparing results from the existing closed-source vendor tool flow. While the created open-source flow is capable of synthesizing a full RISC-V core, many aspects, such as DSPs, RAMs, and PLLs, are currently unsupported. The authors report that documenting the bitstream format for all of these features is the subject of future work.

6.1.1.3 AEx framework

Hirvonen et al. (2023) propose AEx, a framework for automated high-level synthesis of compiler programmable co-processors. AEx can be used to produce application-specific instruction-set (ASIP) architectures. ASIP processors have been proposed as a way to produce FPGA overlays starting from a software-programmable template. The program being executed can be easily changed, reducing design time and costs. The template being used by AEx is transport triggered architecture. AEx includes heuristics for design space exploration and pruning, aimed at finding the best architecture able to satisfy real-time execution time and clock frequency constraints. The user can then choose the results that better fit their needs (e.g., minimum resource utilization). Evaluation shows how the tool is able to produce results in a reasonable amount of time, achieving performance close to that of the fixed-function implementations generated by HLS vendor tools such as AMD/Xilinx Vitis.

6.1.1.4 Synthesis from Simulink models to FPGA for aerospace applications

Reconfigurable hardware is becoming an attractive solution for aerospace applications, thanks to its power efficiency and capabilities of in-flight configuration. Algorithms are usually expressed in model-based programming frameworks, for example MATLAB Simulink, but turning them into low-level hardware description languages can be cumbersome. Curzel et al. (2023) analyze solutions to automatically synthesize Simulink models. MATLAB already provides an automated method (HDL coder) to translate part of Simulink models into Verilog/VHDL, but this still requires a certain level of expertise. Therefore, the authors propose to apply HLS on the code generated by MATLAB's Embedded Coder tool, further automatizing the design process. Experiments with three benchmarks show that this solution is more efficient than relying on HDL coder, and it does not require specific hardware expertise.

6.1.1.5 HLS optimizations for post-quantum cryptography on Lattice FPGAs

Guerrieri et al. (2022) discuss the process of porting post-quantum cryptographic algorithms to an FPGA using HLS. While it can be reasonably straightforward to port an existing CPU implementation to an FPGA, the performance can be low, and

⁷ https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Microelectronics/ESA_HDL_IP_Cores_Portfolio_Overview

resource utilization is not optimal. The authors discuss how, applying well-known HLS-specific optimization techniques, the code can be rewritten to leverage the capabilities of HLS tools and produce more efficient designs, reducing the computation latency of up to two orders of magnitudes in specific cases.

6.1.1.6 Optimizations for heterogeneous HPC systems

Theodoropoulos et al. (2023) demonstrate the results of porting and optimizing industrial applications to two new heterogeneous HPC systems within the OPTIMA project. The results highlight the performance increase of using the available FPGA-based accelerators versus a pure software implementation running on the CPUs of the HPC system. The FPGA accelerators reduce compute time by 1.2–7 times for various industrial application domains.

6.1.1.7 A framework for heterogeneous cloud applications (VINEYARD)

Sidiropoulos et al. (2018) describe a framework to accelerate different parts of an application across different accelerators, like GPUs and FPGAs. They demonstrate the utility of this framework by creating a platform for computational neuroscience, called BrainFrame. The BrainFrame platform allows one to simulate spiking neural networks, and depending on the number of neurons and their interconnectivity, certain combinations of accelerators achieved the shortest simulation times.

6.1.1.8 Mapping data structures to hardware streams (Tydi)

Peltenburg et al. (2020b) describe a specification for mapping complex, dynamically sized data structures onto a fixed number of hardware streams.

6.1.2 Future directions

Traditional FPGA programming has been done using hardware description languages, which have a steep learning curve that does not favor adopting reconfigurable devices in the scientific and industry community. To address this issue, there is a collective effort to increase the abstraction level for FPGA designs without compromising performance. Achieving this goal requires a multidisciplinary approach that involves programming languages, compilers, and optimization techniques. HLS tools and high-level approaches are being used in various application domains. Although the current direction in this field in the Netherlands is unclear, our analysis has pinpointed specific domains of interest within local research communities, such as aerospace and accelerated big data processing, that could benefit from more accessible programming methods for FPGA devices.

6.2 Performance prediction

FPGA design and development processes are time-consuming activities due to, among others, the very fine granularity reconfigurability of FPGA designs, which translates into a large design space and long synthesis time. For this reason, enabling quick performance prediction of synthesized programs to improve early-stage design analysis and exploration, and

performance debugging. We can distinguish between two main types of performance prediction models: analytical and ML-based. Analytical models [such as HLScope+ (Choi et al., 2017) and COMBA (Zhao et al., 2020)] analyze the source code and use mathematical modeling to estimate performance and resource utilization. They are able to produce quick estimates at the cost of reduced accuracy. ML-based models (O'Neal et al., 2018; Ustun et al., 2020; Sun et al., 2021), by comparison, aim at improving prediction accuracy by considering device-specific features, but typically require long and expensive training procedures.

6.2.1 Research topics

Several papers have been published by Dutch organizations focusing on performance prediction of synthesized codes.

6.2.1.1 System performance prediction via transfer learning (LEAPER)

Singha et al. (2022) describe a method for predicting system performance and resource usage of FPGA accelerators using transfer learning. They trained a performance predictor model for an edge/embedded FPGA, and used transfer learning so that the model can also be used for cloud/high-end FPGAs. The method allows for design space exploration of mapping C/C++ programs to cloud/high-end FPGAs using HLS. The authors showed that it is 10 times faster than the state of the art, achieving 85% accuracy.

6.2.1.2 Modeling FPGA-based systems via few-shot learning

Machine learning based models are being proposed to provide fast and accurate performance predictions of FPGA-based designs. However, training these models is expensive due to the time-consuming FPGA design cycle. Singh et al. (2021b) propose a transfer-learning-based approach for FPGA-based systems, to adapt an existing ML model, trained for a specific device, to a new, unknown environment, reducing the training costs.

6.2.1.3 Energy and area estimation for coarse-grained reconfigurable architectures

Design space exploration is often required to achieve good Pareto points when creating reconfigurable architectures. Wijtvlit et al. (2021) introduce the CGRA-EAM model for energy and area estimation for CGRAs. It achieves a 15.5% error for energy and 2.1% error for area estimation for the Blocks (Wijtvlit et al., 2019) CGRA. The novelty of this work lies in its focus on CGRAs and its ability to handle multiple different applications running on a CGRA.

6.2.1.4 Analytical performance estimation for large-scale reconfigurable dataflow platforms

Yasudo et al. (2018, 2021) introduced and further expanded PERKS, a performance estimation framework for reconfigurable dataflow platforms. The authors propose that reconfigurable accelerators, such as FPGAs, will play an important role in future exascale computing platforms and that such a framework is essential in the efficient deployment of applications on heterogeneous platforms with reconfigurable accelerators. The PERKS framework uses parameters from the target platform and the application to build an analytical model to predict the

performance of multi-accelerator systems. Experimental results with different reconfigurable dataflow applications are presented, showing that the framework can predict the performance of current workloads with high accuracy.

6.2.1.5 Memory and communication profiling for accelerator-based platforms

Ashraf et al. (2018) present MCPROF, an open-source memory-access and data-communication profiler. The tool provides a detailed profile of memory-access behavior for heterogeneous systems (CPUs, GPUs, and FPGAs) for C/C++ applications, as well as data-communication-aware mapping of applications on these architectures. Comparison with the state of the art show that the proposed profiler has an order of magnitude, on average, lower overhead than state-of-the-art data-communication profilers over a wide range of benchmarks. A case study with several image processing applications for heterogeneous multi-core platforms containing an FPGA and a GPU as accelerators was conducted. The authors also demonstrate that the tool can provide insights into whether a specific accelerator (GPU or FPGA) is a good fit for the application.

6.2.1.6 Delay prediction for ASIC HLS

The delay estimates of HLS tools can often deviate significantly from results obtained from logic synthesis. De et al. (2023) propose a hybrid model by incorporating graph-based learning models, which can infer structural features from a design, into traditional non-graph-based learning models for delay estimates. The hybrid model improves delay prediction by 93% compared to the delay prediction reported by a commercial HLS tool.

6.2.2 Future directions

Performance prediction for FPGAs and, more generally, reconfigurable dataflow devices is challenging. Traditionally, this was addressed with analytical approaches, but more recently, ML-based approaches are becoming mainstream, and we can expect them to become a popular option given their successes in recent research. With CGRA-like architectures being commercialized, the interest in exploring these devices is increasing, and we expect future work to focus more on the predictability of the performance of such devices. Performance prediction is a device-specific task, yet only a few of the reviewed papers favor the reproducibility of the presented results, or publicly release the code used to generate these results. We should shift toward more reproducible and transparent research to foster collaborations and facilitate general progress. Thus, we advocate for the need for open-source practices in performance prediction as well.

7 Robustness of FPGAs

This section discusses deployment challenges and ongoing research efforts in Dutch institutions toward achieving robust FPGA implementations. The robustness of FPGAs and the designs they support is a multidimensional challenge, affecting several stages of an FPGA-based systems' production chain. Here, we delve into two critical aspects: reliability (Section 7.1) and security

(Section 7.2). We discuss the intricacies, and present collaborative efforts required to address these challenges.

7.1 Reliability

This section delves into FPGA devices' reliability in high-radiation environments, such as space and particle colliders—a topic that is at the forefront of research by ESTEC in the Netherlands. FPGAs, particularly those based on Static Random Access Memory (SRAM), are vulnerable to charged or high-energy particles. The Configuration Random Access Memory (RAM) of SRAM-based FPGAs is prone to single event upsets (SEUs), which can alter bits within the configuration memory. These alterations can potentially change the logic functions and connections within the device, leading to single event functional interruptions. To mitigate these risks, protective measures such as triple modular redundancy or error correcting code scrubbers are often implemented. However, these solutions can complicate the design and add significant overhead. As an alternative, Flash-based FPGAs offer a robust option because their configuration memory is inherently immune to SEUs, although they face constraints in terms of computing resources and less mature toolchains.

7.1.1 Research topics

FPGAs serve as an ideal test platform for advancing the maturity of RTL designs, especially in the context of technology readiness levels (TRLs) in space applications. FPGAs can be used to simulate early-stage prototypes, allowing for iterative refinement. In subsequent phases, they can be used to validate system components under simulated space-like conditions, ensuring RTL designs work as intended in the actual hardware configuration. In the final stages, FPGAs integrated with RTL designs are tested in operational environments, including actual space missions, demonstrating the systems performance in real-world conditions. This approach provides a flexible and cost-effective method for testing and validating technology and helps manage risks associated with deploying new technologies in space. Dutch research on FPGA devices focuses primarily on technology validation to assess and enhance their resilience to adverse conditions such as radiation, alongside the development of novel reliability techniques to mitigate vulnerabilities like high susceptibility to SEUs. Additionally, FPGAs are extensively employed for testing low TRL innovations and for realizing robust operational systems in various applications. The following elaborates on these core research areas.

7.1.1.1 Technology validation of FPGA devices

This topic assesses the resilience of FPGA devices under the effect of charged and high-energy particles. The Payload-XL project (Viel et al., 2023) exemplifies practical deployment, validating the BRAVE FPGA's performance in orbit and demonstrating the system's real-world reliability under space conditions. Studies that use ultra-high-energy heavy ions (Vlagkoulis et al., 2021; Du et al., 2019) characterize and quantify how various FPGA technologies respond to

environmental challenges, such as in space. [Leon et al. \(2021c\)](#) employ diverse benchmarks to evaluate the performance of new radiation hardened devices.

7.1.1.2 Advancements in FPGA device reliability

Research here develops innovative techniques aimed at enhancing FPGA reliability. [De Sio et al. \(2023\)](#) describe PyXEL's bitstream analysis tool for enhancing FPGA robustness by automating reliability analysis and facilitating mitigation solutions. [Vlagkoulis et al. \(2022\)](#) present configuration memory scrubbing methods that integrate mixed two-dimensional coding techniques, while [Mousavi et al. \(2023\)](#) discuss scrubbing methods that strive to reduce the mean time to repair, thereby significantly improving error correction capabilities.

7.1.1.3 FPGAs as research and development platforms

Various studies explore the role of FPGAs as both experimental test beds and final platforms for payload systems ([Viel et al., 2023](#); [Forlin et al., 2023](#); [Bohmer et al., 2023](#)). [Gambardella et al. \(2022\)](#) conducted accelerated radiation tests on quantized neural networks, while [Anders et al. \(2023\)](#) and [Hoozemans et al. \(2018\)](#) underline the capabilities of FPGAs to support the development of new processor architectures, such as RISC-V processors, and new computing paradigms, such as polymorphic very long instruction word (VLIW) processors, respectively.

7.1.2 Future directions

The future of reliability work on FPGAs in the Netherlands is increasingly shaped by the demand from the ESTEC ESA center and the expanding new space economy for advanced space applications. Dutch companies, such as Technolution, with its FreNox RISC-V soft-core, play a pivotal role in building this high-reliability ecosystem. As new FPGA providers in Europe, like NanoExplore, emerge, the need for technology-agnostic designs that enable the seamless integration of external IPs becomes more critical. The primary focus moving forward will be on expanding this ecosystem by adapting and validating existing common IPs for high-reliability applications while maintaining ease of integration across various platforms.

7.2 Hardware security

This section explores the security challenges and innovative solutions in FPGA technology. FPGAs present unique security risks due to their reconfigurability and wide deployment range, from cloud data centers to edge devices. Their flexibility and complex ecosystem makes them susceptible to various threats, including hardware Trojans, fault injection, and side-channel attacks.

Security, particularly in the realm of FPGAs, hinges on establishing and maintaining trust and isolation. This trust permeates the entire supply chain of the FPGA market, as detailed by [Zhang and Qu \(2014\)](#). Trust must exist between suppliers and consumers for instance, between foundries and FPGA vendors and is subsequently extended to downstream users. For example, system developers rely on FPGA vendors to ensure that their hardware and EDA tools are free from hardware Trojans, as

discussed in various studies ([Zeitouni et al., 2021](#); [Labafniya et al., 2020](#); [Nikiema et al., 2023](#); [Palumbo et al., 2022](#)). The direct relationships of trust within this framework are illustrated in [Figure 6](#). We observe that, at certain stages in the supply chain, the dynamics between service providers and consumers go beyond mere transactional interactions and become operational. This shift is represented by the dotted line in the figure. Typical examples include the relationship between end users and system developers or between cloud service providers and end users. In these scenarios, interactions between the involved parties are more reciprocal, and the focus expands beyond trust. Critical issues then include ensuring isolation within cloud environments, protecting against malicious users, and implementing countermeasures to various attacks, as highlighted in recent research ([Zeitouni et al., 2021](#); [Koylu et al., 2022](#); [Garaffa et al., 2021](#); [Socha et al., 2020](#)).

7.2.1 Research topics

In the Netherlands, research on security spans multiple technical domains, employing a variety of techniques to ensure operational level guarantees.

7.2.1.1 Vendor side

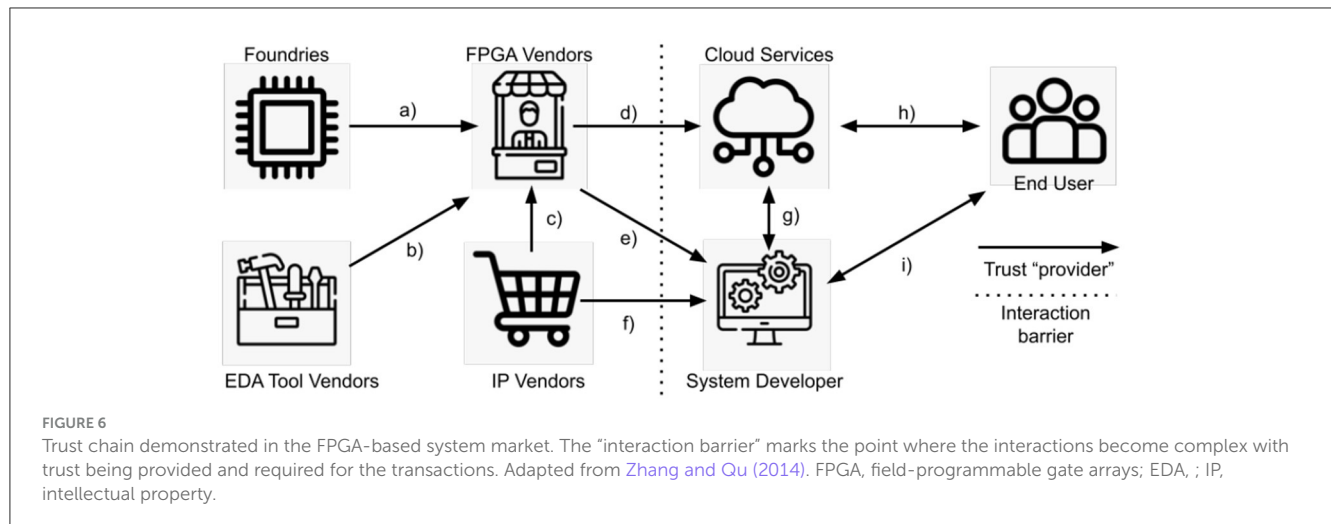
Vendor side research (left side of [Figure 6](#)) assesses trust and security in the system. These efforts predominantly aim to detect and neutralize hardware Trojans, as discussed in various studies ([Zeitouni et al., 2021](#); [Labafniya et al., 2020](#); [Nikiema et al., 2023](#); [Palumbo et al., 2022](#)). In cases involving hardware Trojans, while FPGA vendors might not be directly accountable for the intrusion, they inherit a trust relationship from upstream entities such as foundries.

7.2.1.2 User-side

By comparison, much of the research focus in the Netherlands is directed toward the interactions depicted on the right-hand side of the dotted line in [Figure 6](#). This includes examining the connections among cloud services, system developers, and end users. Key focal areas include the development of hardware primitives like programmable unclonable functions ([Jin et al., 2022, 2020](#)) and roots-of-trust ([Nikiema et al., 2023](#)), and evaluation of fault injection and protection mechanisms ([Miteloudi et al., 2022](#); [Nikiema et al., 2023](#); [Köylü et al., 2022](#); [Koylu et al., 2022](#)). Additionally, significant attention is given to detecting and mitigating side-channel attacks ([Lahr et al., 2020](#); [Miteloudi et al., 2021](#); [Socha et al., 2020](#); [Garaffa et al., 2021](#)) and micro-architectural attacks ([Arikan et al., 2022](#); [Nikiema et al., 2023](#)).

7.2.2 Future directions

Current research in the Netherlands extensively uses FPGA systems across all stages of end development. However, there is relatively less focus on enhancing the resilience and security of FPGA production, likely due to the absence of commercial players developing these systems within the country. As a result, future research is expected to increasingly emphasize the integration of FPGAs into system-level solutions, addressing critical issues in security from an end-user perspective. The inherent flexibility of FPGA platforms also presents significant opportunities for



innovative system design. They can serve as an initial gateway into a broader design ecosystem, where the development of IPs is validated for functional metrics, such as side-channel leakage and protection mechanisms. This approach ensures that new designs meet stringent standards before deployment, fostering a robust and secure technological landscape.

8 Applications

FPGAs have emerged as powerful accelerators for a wide range of applications. In this section, we discuss FPGA-based solutions in ML (Section 8.1), astronomy (Section 8.2), particle physics experiments (Section 8.2.6), quantum computing (Section 8.3), space applications (Section 8.4), and bioinformatics (Section 8.5).

8.1 Machine learning

In the field of ML, and in particular deep learning, hardware acceleration plays a vital role. GPUs are the predominant method for hardware acceleration due to their high parallelism, but FPGA research is showing promising results. FPGAs enable inference at greater speeds and better power efficiencies when compared to GPUs ([Nurvitadhi et al., 2016](#)) by designing model-specific accelerated pipelines ([Irmak et al., 2021a](#)). Through the co-design of ML models and ML hardware on FPGAs, models are accelerated without compromising on performance metrics and utilizing limited FPGA resources. In addition, the flexibility of the FPGAs architecture enables the realization of unconventional deep learning technology, such as spiking neural networks.

8.1.1 Hardware acceleration

Ample research on hardware acceleration focuses on accelerating existing neural network architectures. One common class of architectures is convolutional neural networks (CNNs), which learn image filters to identify abstract image features. CNNs are often deployed in embedded applications that require

real-time image processing and low energy consumption, making FPGAs suitable candidates for CNN acceleration. [Irmak et al. \(2021a\)](#) propose implementing the LeNet architecture using Vitis HLS, pipelining the CNN layers, and outperforms other FPGA based implementations at a processing time of $70\mu s$. One downside to this approach is the inflexibility of designing a specific model architecture in HLS which can be resolved by using partial reconfiguration ([Irmak et al., 2021c](#)). To increase CNN throughput, further parallelization can be exploited, and in combination with the use of the high bandwidth OpenCAPI interface, can achieve a latency of less than $10\mu s$ on the LeNet-5 model, streaming data from an High-Definition Multimedia Interface (HDMI) ([Ji et al., 2023](#)). In each of these implementations, fully pipelined CNNs are possible due to the limited number of parameters in small CNNs. As larger pipelined networks are deployed on FPGAs, parallelization puts strain on the available resources, and in particular, the amount of on-chip-memory becomes a bottleneck. A proposed solution to this is using frequency compensated memory packing ([Petrica et al., 2020](#)).

In addition to CNN acceleration, general neural network acceleration has been developed by means of a programmable tensor processing unit as an overlay on an FPGA accelerator ([Diamantopoulos et al., 2020](#)). Deep learning acceleration using FPGAs is also relevant to space technology research. Because the reprogrammability of FPGAs make them a suitable contender for deployment on space missions, FPGA implementations of existing deep learning models are being benchmarked for space applications ([Rapuano et al., 2021](#); [Petrica et al., 2020](#)).

8.1.2 Spiking neural networks

Spiking neural networks (SNNs) are computational models formed using spiking neuronal units that operate in parallel and mimic the basic operational principles of biological systems. These features endow SNNs with potentially richer dynamics than traditional artificial neural network models based on the McCulloch Pitts point neurons or simple rectified linear unit (ReLU) activation functions that do not incorporate timing information. Thus, SNNs excel in handling temporal

information streams and are well suited for innovative non-von Neumann computer architectures, which differ from traditional sequential processing systems. SNNs are particularly well suited for implementation in FPGAs due to their massive parallelism and requirement for significant on-chip memories with high-memory bandwidth for storing neuron states and synaptic weights. Additionally, SNNs use sparse binary communication, which is beneficial for low-latency operations because both computing and memory updates are triggered by events. FPGAs' inherent flexibility allows for reprogramming and customization, which enable reprogrammable SNNs in FPGAs, resulting in flexible, efficient, and low-latency systems (Corradi et al., 2021; Irmak et al., 2021b; Sankaran et al., 2022). Corradi et al. (2024) demonstrated the application of a spiking convolutional neural network (SCNN) to population genomics. The SCNN architecture achieved comparable classification accuracy to state-of-the-art CNNs while processing only about 59.9% of the input data, reaching 97.6% of CNN accuracy for classifying selective-sweep and recombination-hotspot genomic regions. This was enabled by the SCNN's capability to temporize genetic information, allowing it to produce classification outputs without processing the entire genomic input sequence. Additionally, when implemented on FPGA hardware, the SCNN model exhibited over three times higher throughput and more than 100 times greater energy efficiency than a GPU implementation, markedly enhancing the processing of large-scale population genomics datasets.

8.1.3 Model/hardware co-design

Previous examples demonstrate that existing deep neural network models can be accelerated using FPGAs. Typically, research in this area focuses on designing an optimal hardware solution for an existing model. A more effective approach, however, is to co-design the model and the hardware accelerator simultaneously. However, simultaneous co-design of DNN models and accelerators is challenging. DNN designers often need more specialized knowledge to consider hardware constraints, while hardware designers may need help to maintain the quality and accuracy of DNN models. Furthermore, efficiently exploring the extensive co-design space is a significant challenge. This co-design methodology leads to better performance, leveraging FPGAs' flexibility and rapid prototyping capabilities. For example, Rocha et al. (2020) by co-designing the bCorNET framework, which combines binary CNNs and LSTMs, they were able to create an efficient hardware accelerator that processes heart rate estimation from photoplethysmography signals in real time. The pipelined architecture and quantization strategies employed allowed for significant reductions in memory footprint and computational complexity, enabling real-time processing with low latency.

In SNNs, encoding information in spike streams is a crucial co-design aspect. SNNs primarily use two encoding strategies: rate-coding and time-to-first-spike (TTFS) coding. Rate coding is common in SNN models, encoding information based on the instantaneous frequency of spike streams. Higher spike frequencies result in higher precision but at the cost of increased energy consumption due to frequent spiking. While rate coding offers accuracy, it reduces sparsity. In FPGA implementations, rate

coding is often used for its robustness, simplicity, ease of training through the conversion of analog neural networks to spiking neural networks, and practicality in multi-sensor data fusion, where it helps represent real values from various sensors (radars, cameras) even in the presence of jitter or imperfect synchronization (Corradi et al., 2021). Conversely, TTFS coding has been demonstrated in SNNs implemented on FPGAs to enhance sparsity and has the potential of reducing energy consumption by encoding information in spike timing. For instance, Pes et al. (2024) introduced a novel SNN model with active dendrites to address catastrophic forgetting in sequential learning tasks. Active dendrites enable the SNN to dynamically select different sub-networks for different tasks, improving continual learning and mitigating catastrophic forgetting. This model was implemented on a Xilinx Zynq-7020 SoC FPGA, demonstrating practical viability with a high accuracy of 80% and an average inference time of 37.3 ms, indicating significant potential for real-world deployment in edge devices.

8.2 Astronomy

Observations in astronomy are done at different modalities and wavelengths, such as detection of a range of particles (e.g., Cherenkov detector based systems such as KM3NeT; Bagley, 2009), gravitational waves, optical observations, gamma and x-ray observations and radio [e.g., Westerbork Synthesis Radio Telescope (WSRT) (van Cappellen et al., 2022), LOw-Frequency ARray (LOFAR) (van Haarlem et al., 2013), and Square Kilometer Array (SKA) (Hall, 2005)]. Observations can be done from space or from Earth; in this section, we limit the scope to ground-based astronomy. A common denominator for instruments required for observing different modalities and different wave lengths is that the systems need to be very sensitive in order to observe very faint signals from outside the Earth's atmosphere. Instruments are typically large and/or distributed over a large area to achieve good sensitivity and resolution. Different modalities and wavelengths require distinct types of sensors, cameras, or antennas to convert observed phenomena into electrical signals. Each system is tailored to its specific modality and wavelength, necessitating specialized components to accurately capture and translate the data. At a certain stage in the signal chain, the electrical signal is converted into the digital domain, where it undergoes multiple processing stages. This processed signal ultimately results in an end product that can be used by scientists for analysis and research purposes. Systems can roughly be split into two parts, a front end and a back end. The front end requires interfacing with and processing of data from the sensor; electronics commonly deployed in the front end are constrained in space (size), temperature, power, cost, RFI, environmental conditions, and serviceability. The back end processes data produced by the front end(s) either in an online or offline fashion, which is usually done with server infrastructure in a data center. In the back end, the main challenges are the high data bandwidth and large data size coming from the front-ends. Although the environment is more flexible, systems are still constrained in space, power, and cost.

FPGAs have been used in astronomy instrumentation for quite some time, as they are efficient in interfacing with analog-to-digital converters (ADCs) and well suited to the conditions faced in instrumentation front ends [e.g., NCLE (Karapakula et al., 2024)]. Moreover, FPGA are also used further down the processing stages for various signal processing operations, both in the front ends [e.g., Uniboard2 in LOFAR (Schoonderbeek et al., 2019)] as well as in the back ends of systems [e.g., MeerKAT (van der Byl et al., 2022) and SKA (Hampson et al., 2016)]. GPUs represent a good alternative in back-end processing [e.g., LOFAR's system COBALT (Broekema et al., 2018)] as well. The work by Veenboer and Romein (2019) describes a trade-off between using a GPU and an FPGA accelerator in the implementation of an image-processing operation in a radio telescope back-end.

8.2.1 Hardware development for the radio neutrino observatory in Greenland

The Radio Neutrino Observatory in Greenland (Smith et al., 2022) is a radio detection array for neutrinos. It consists of 35 autonomous stations deployed over a 5×6 km grid near the National Science Foundation (NSF) Summit Station in Greenland. Each station includes an FPGA-based phased trigger. The station has to operate in a 25 W power envelope. Implementing an FPGA seems to be favorable due to environmental conditions and operation constraints.

8.2.2 Implementation of a correlator onto a hardware beam-former to calculate beam-weights

The apertif phased array feed (PAF) (van Cappellen et al., 2022) is a radio telescope front end used in the WSRT system in the Netherlands. FPGAs are used for antenna read out as well as signal processing close to the antenna. Schoonderbeek et al. (2020) describe transforming and implementing a beamformer algorithm on FPGA to build a more efficient system.

8.2.3 Near memory acceleration and reduced-precision acceleration for high resolution radio astronomy imaging

Corda et al. (2020) describe implementing a 2D FFT on FPGA, leveraging near-memory computing. The 2D FFT is applied to an image-processing implementation on FPGA in the back end of a radio telescope and compared with implementations on CPU and GPU. The FPGA implementation is 120 times faster than a CPU implementation, and it is as fast as a GPU implementation but with reduced bandwidth and memory requirements. Corda et al. (2022) explore reduced-precision computation on an FPGA for the same image-processing application and increases the energy efficiency by 3.46 times compared to a CPU.

8.2.4 The MUSCAT readout electronics backend: design and pre-deployment performance

The MUSCAT is a large single-dish radio telescope with 1,458 receives in the focal plane. The system uses FPGA-based electronics

to read out and pre-process the data from the receivers (Rowe et al., 2023).

8.2.5 Cherenkov telescope arrays

Three different contributions have been made to three different Cherenkov-based telescope arrays. Ashton et al. (2020) describe a system for the High Energy Stereoscopic System, where a custom board with ARM CPU and an FPGA is used to read out and pre-process a custom designed NECTAr digitizer chip in the front end of the system. After pre-processing, the data are distributed to a back-end over Ethernet. Sanchez-Garrido et al. (2021) present the design of a Zynq FPGA SoC-based platform for White Rabbit time synchronization in the ZEN-CTA telescope array front ends. Data captured and pre-processed at the front ends are distributed over Ethernet to the back end including the timestamp. Aiello et al. (2021) outline the architecture and performance of the KM3Net front-end firmware. The KM3Net telescope consists of two deep-sea 3-D sensor grids being deployed in the Mediterranean Sea. A central logic board with FPGA in the front end serves as a time to digital converter to record events and time at the sensors; the data are transmitted and further processed in a back end onshore.

FPGAs are mainly used for front end sensor interfacing and pre-processing. Corda et al. (2020, 2022) underline that FPGAs are also still relevant in the back-end, providing improved performance over CPU and on-par performance with GPU accelerators. FPGA are expected to remain the dominant choice for platforms in astronomy instrumentation front ends due to the strong interfacing capabilities and the adaptability and suitability to the constraints imposed by instrumentation front-ends. In the back end, FPGAs provide a viable solution to application acceleration, but will have to compete with other accelerator architectures, for example GPUs (Veenboer and Romein, 2019). An emerging new technology are the artificial intelligence engines in the AMD Versal Adaptive SoC. The work from van Wijhe et al. (2024) evaluated the AI engines' for a signal-processing application in radio astronomy. AI engines' flexibility and programmability, combined with the FPGA's interfacing capabilities can lead to a powerful platform for telescope front ends.

8.2.6 Particle physics experiments

The Large Hadron Collider (LHC) features various particle accelerators to facilitate particle physics experiments. Experiments performed using particle accelerators can produce massive amounts of data that need to be propagated and preprocessed at high speeds before the reduced relevant data are recorded for offline storage. FPGAs are widely employed throughout systems' LHC particle accelerators, such as ATLAS and LHCb, for their high-bandwidth capabilities, and the flexibility that reconfigurable hardware offers without requiring hardware alterations to the system. Recently both the ATLAS and LHCb particle accelerators have been commissioned for upgrades. The Dutch Institute for Subatomic Physics (Nikhef) is one of the collaborating institutes working on the LHC accelerators.

LHCb is a particle accelerator that specializes in experiments that study the bottom quark. Major upgrades to the LHCb that enable handling a higher collision rate require new front end and back end electronics. To facilitate the back end of the upgrade, the LHCb implements the custom PCIe40 board, which features an Intel Arria 10 FPGA. Four PCIe40 boards are dedicated to controlling part of the LHCb system, and 52 PCIe40 boards are used to read out each of the detectors slices, producing an aggregated data rate of 2.85 Tb/s (Fernandez Prieto et al., 2020).

ATLAS is one of the general particle accelerators of the LHC. ATLAS uses two trigger stages in order to record only the particle interactions of interest. In an upgrade to the ATLAS accelerator, ASIC-based calorimeter trigger preprocessor boards are replaced by FPGA-based hardware. Using FPGAs for this purpose allows implementing enhanced signal-processing methods (Aad et al., 2020). After the two trigger stages, FPGAs are deployed to process the triggers for tracking particles (Aad et al., 2021).

Ongoing upgrades to the LHC particle accelerators, referred to as High Luminosity LHC (HL-LHC), will facilitate higher energy collisions. HL-LHC will produce increased background rates. To reduce false triggers due to background, the New Small Wheel checks for coinciding hits. Each trigger processor features Virtex-7, Kintex Ultrascale and Zynq FPGAs (Iakovidis et al., 2023). Interaction to and from front-end hardware is done through Front End Link eXchange (FELiX) boards. As part of the HL-LHC upgrades, each FELiX board must facilitate a maximum throughput of 200 Gbps. To enable this, RDMA over Converged Ethernet as part of the FELiX FPGA system is proposed (Vasile et al., 2022, 2023). The performance of the FELiX upgrade in combination with an upgraded Software ReadOut Driver satisfies the data transfer requirements of the upgraded ATLAS system (Gottardo, 2020).

8.3 Quantum computing

Quantum computing promises to help solve many global challenges of our time, such as quantum chemistry problems for designing new medicines, predicting material properties for efficient energy storage, and handling big data needed for complex climate physics (Gibney, 2014). The most promising quantum algorithms demand systems comprising thousands to millions of quantum bits (van Meter and Horsman, 2013), the quantum counterpart of a classical bit. A quantum processor comprising up to 50 qubits has been realized using solid-state superconducting qubits (Arute et al., 2019), but its operation requires a combination of cryogenic temperatures below 100 mK and hundreds of coaxial lines for qubit control and readout. While in systems with a few qubits, this can be controlled using off-the-shelf electronic equipment, such approach becomes infeasible when scaling qubit systems toward thousands or millions of qubits that are required for a practical quantum computer.

A means to tackle the foreseeable bottleneck in scaling the operation of qubit systems is to integrate FPGA technology in the control and readout of solid-state qubits. FPGAs have

been used to generate highly stable waveforms suitable for the control of quantum bits with latency significantly lower than software alternatives (Ireland et al., 2020). In systems of semiconductor spin qubits, FPGAs have provided in-hardware syncing of quantum dot control voltages with the signal acquisition and buffering and thus enabled the observation of real-time charge-tunneling events (Haarman et al., 2023). FPGAs have also been used to configure and synchronize a cryo-controller with an arbitrary waveform generator required to generate complex pulse shapes and perform quantum operations (Xue et al., 2021). Such a setup has enabled the demonstration of universal control of a quantum processor hosting six semiconductor spin qubits (Philips et al., 2022). FPGAs have proven to be essential for implementing quantum error correction algorithms, which are critical for mitigating the effects of dephasing and decoherence in solid-state qubits. In qubit systems based on superconducting quantum circuits, the first efficient demonstration of quantum error correction was made possible by an FPGA-controlled data acquisition system, which provided dynamic real-time feedback on the evolution of the quantum system (Ofek et al., 2016). It has been further predicted that FPGA can enable highly efficient quantum error correction based on neural-network decoders (Overwater et al., 2022).

FPGA technology has proven invaluable in the development of the emerging research field of quantum computing. However, the complexity of programming FPGA circuits hinders their implementation in quantum computing systems. Commercial efforts have been made toward providing graphical tools for designing FPGA programs, namely, the Quantum Researchers Toolkit by Keysight Technologies and the FPGA-based multi-instrument platform Moku-Pro developed by Liquid Instruments. These tools are essential for implementing customized algorithms without the need for dedicated expertise in hardware description languages. Future research is also needed in integrating FPGAs in cryogenic platforms required to operate qubit systems. Such capability has already been demonstrated; commercial FPGAs can operate at temperatures below 4 K and be integrated in a cryogenic platform for qubit control (Homulle et al., 2017). These efforts provide evidence that FPGA technology is of great interest for enabling a scalable and practically applicable quantum computer.

8.4 Space

The flexibility of FPGA technology makes it a suitable platform for many applications on-board space missions. The ESTEC, as part of the ESA actively explores FPGA technology for space applications, and has an extensive portfolio of FPGA IP cores.⁸

FPGAs can flexibly route their input and output ports and be configured to support many different communication protocols. This makes FPGAs good contenders as devices that communicate with the various hardware platforms and sensors

⁸ Esa hdl ip cores portfolio overview. (2024). Available online at: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Microelectronics/ESA_HDL_IP_Cores_Portfolio_Overview (accessed June 03, 2024).

on a space mission. FPGAs have been implemented as interface devices in novel on board ML and digital signal-processing implementations (Leon et al., 2021b,a; Karapakula et al., 2024).

An on board task for which FPGAs are used is hyperspectral imaging. This type of on board imaging produces vast amounts of data. To reduce transmission bandwidth requirements when transmitting the sensory data to Earth, real-time on board compression handling high data rates is required. FPGAs are well suited for such tasks, and research has been done on using space-grade radiation-hardened FPGAs (Barrios et al., 2020) as well as commercial off-the-shelf (COTS) FPGAs (Rodriguez et al., 2019) for on-board hyperspectral image compression. COTS devices are generally cheaper than space-grade devices, but well suited higher susceptibility to radiation-induced effects makes them challenging to employ.

Communication between on board systems often requires high data-rates and is susceptible to radiation-induced effects. To deal with the unique constraints of space applications, dedicated communication protocols such as SpaceWire and its successor, SpaceFibre, have been developed. These protocols are available as FPGA IP implementations, and testing environments of SpaceFibre have been developed (Mystkowska and Steenari, 2022; Bargholz et al., 2022). SpaceWire can interface with the common AXI4 protocol using a dedicated bridge (Rubattu et al., 2022), enabling its integration with SpaceWire interfaces. Direct memory access (DMA) allows peripherals to transfer data to and from an FPGA without going through a CPU. The application of DMA in space is being investigated; however its application, as of now, is limited because DMA is susceptible to radiation-induced effects (Portaluri et al., 2022).

8.5 Bioinformatics

FPGA technology has been extensively explored for accelerating bioinformatics kernels. Bioinformatics is an interdisciplinary scientific field that combines biology, computer science, mathematics, and statistics to analyze and interpret biological data. The field primarily focuses on the development and application of methods, algorithms, and tools to handle, process, and analyze large sets of biological data, such as DNA sequences, protein structures, and gene expression patterns. Continuous advances in DNA sequencing technologies (Hu et al., 2021) have led to the rapid accumulation of biological data, creating an urgent need for high-performance computational solutions capable of efficiently managing increasingly larger datasets.

Shahroodi et al. (2022) describe a hardware/software co-designed framework to accelerate and improve energy consumption of taxonomic profiling. In metagenomics, the main goal is to understand the role of each organism in our environment to improve our quality of life, and taxonomic profiling involves identifying and categorizing the various types of organisms present in a biological sample by analyzing DNA or protein sequences from the sample to determine which species or taxa are represented. The study focuses on boosting performance of table lookup, which is the primary bottleneck in taxonomic profilers, by proposing a processing-in-memory hardware accelerator. Using large-scale simulations, the authors report an average of 63.1% faster execution

and orders of magnitude higher energy efficiency than the widely used metagenomic analysis tool Kraken2 (Wood et al., 2019) executed on a 128-core server with AMD EPYC 7742 processors operating at 2.25 GHz. An FPGA was used for prototyping and emulation purposes.

Corts et al. (2022) employ FPGAs to accelerate the detection of traces of positive natural selection in genomes. The authors designed a hardware accelerator for the ω statistic (Kim and Nielsen, 2004), which is extensively used in population genetics as an indicator of positive selection. In comparison with a single CPU core, the FPGA accelerator can deliver up to 57.1 times faster computation of the ω statistic, using the OmegaPlus (Alachiotis et al., 2012) software implementation as reference.

Malakonakis et al. (2020) use FPGAs to accelerate the widely used phylogenetics software tool Randomized Accelerated Maximum Likelihood (RAxML) (Stamatakis, 2014). The study implements the phylogenetic likelihood function (PLF), which is used for evaluating phylogenetic trees, on a Xilinx ZCU102 development board and a cloud-based Amazon Web Services (AWS) EC2 F1 instance. The first system (ZCU102) can deploy two accelerator instances, operating at 250MHz, and delivers executions up to 7.7 times faster than sequential software execution on a AWS EC2 F1 instance. The AWS-based accelerated system is 5.2 times faster than the same software implementation.

Alachiotis et al. (2021a) also target the PLF implementation in RAxML and propose an optimization method for data movement in Peripheral Component Interconnect (PCI)-attached accelerators using tree-search algorithms. They developed a software cache controller that leverages data dependencies between consecutive PLF calls to cache data on the accelerator card. In combination with double buffering over Peripheral Component Interconnect Express (PCIe), using this approach improved the performance of an FPGA-based PLF accelerator by nearly 4 times. Executing the complete RAxML algorithm on an AWS EC2 F1 instance, the authors observed that processing protein data was up to 9.2 times faster than using a 2.7 GHz Xeon processor in the same cloud environment.

With genomic datasets continuing to expand, bioinformatics analyses are likely to increasingly rely on cloud computing in the future. This shift will be supported by new programming models and frameworks designed to address the data-movement challenges posed by cloud-based hardware accelerators. These accelerators, such as FPGAs and GPUs, need data transfers from the host processor, which can significantly impact execution times and negate gains from computation improvements. Fortunately, similar data-movement concerns exist for both FPGAs and GPUs, and ongoing engineering efforts are likely to converge on common solutions (Corts and Alachiotis, 2023). This will help bring optimized, hardware-accelerate processing techniques into more widespread use among computational biologists and bioinformaticians in the future.

9 Research and development in industry

To understand the alignment between FPGA research in Dutch academia and industry, a questionnaire was initially distributed to Dutch companies working in FPGA technology. Nine companies

responded, representing a broad range of FPGA applications, from custom hardware architecture development to programming models, tools, and application-specific designs. Thereafter, we conducted interviews with several industry representatives in the Netherlands, reviewed the initial survey findings with them, and integrated their feedback into this analysis. The survey posed nine questions, addressing the following subjects:

1. The themes and applications where companies apply FPGAs (questions 1, 2).
2. The reasons for companies to choose FPGA technology (question 3).
3. To what extent the companies create custom FPGA designs (question 4).
4. The satisfaction of companies with respect to the available FPGA technology on the market (questions 5, 6).
5. The most beneficial advances in FPGA technology to companies (question 7).
6. The involvement of companies in academic research (questions 8, 9).

The companies involved and consented to their names being published are ProDrive Technologies, HDL Works, Sioux Technologies, Technolution, QBayLogic, Demcon and Core-Vision.⁹ Other companies choose to remain anonymous.

The most frequently cited applications for FPGAs are control systems, near-memory processing and resilient designs. Table 2 shows an overview of the most cited applications (each company could submit multiple applications). Most companies also reported that they use existing FPGA technology and software to develop custom IP designs for their products, while a fraction (three companies) also highlighted their focus on custom software development to create FPGA solutions.

Companies cite a variety of reasons for choosing FPGAs over GPUs or CPUs in their solutions. The most frequently mentioned motivations are shown in Table 2 (there can be more than one motivation per company). These motivations align with the typical advantages of FPGAs, including high parallelism, high throughput, and substantial processing bandwidth. Additionally, companies appreciate the flexibility in hardware design that FPGAs offer, allowing for adjustments throughout a product's lifetime without major redesigns. Several companies also highlight the deterministic behavior and precise timing of FPGAs as key factors in their decision-making. Furthermore, the cost-effectiveness of FPGAs for implementing custom hardware designs—especially for low-volume products, compared to developing a custom ASIC—plays a significant role in their preference. Other advantages mentioned include flexible hardware interfacing, low latency, low power consumption, and sufficient I/O bandwidth.

Most companies express general satisfaction with the current state of accessible FPGA technology, with only two reporting dissatisfaction. However, they identify several developments that could enhance their FPGA designs, such as on-chip memory for smaller FPGA devices, improved quality of IP cores, better

software for managing HDL dependencies and packages, and high-level languages that retain the details of hardware design. Many companies suggest that FPGA technology would be more attractive if high-level design efforts shifted focus from making FPGAs easier for software developers to enhancing the capability of hardware designers to work at a higher level of abstraction. Overall, tooling is regarded as a critical area for improvement. Frequent changes in vendor tooling often lead companies to develop custom solutions. Additionally, vendor-specific tools create varying workflows, making it challenging to work with devices from different manufacturers. While open-source tools exist, they are generally not considered suitable for professional use.

In conclusion, companies employ FPGA technology when their solutions require the specific advantages that FPGAs provide, such as low latency, high processing bandwidth, deterministic behavior, and flexible interfacing. The accessibility and adoption of FPGAs in the industry would significantly improve with the development of vendor-agnostic, open-source toolchains that are suitable for professional use. Moreover, creating a high-level programming language tailored for hardware designers, instead of focusing on accessibility for software developers, would greatly enhance FPGA design in industrial applications.

10 Conclusion

This article presents an overview of the current research landscape, applications, and future potential of FPGA technology in the Netherlands. It highlights how academia and industry in the Netherlands play an important role in developing new and innovative FPGA-based technologies and solutions to address various important societal challenges ranging from health care to power efficiency.

10.1 Summary and main findings

We selected a total of 212 relevant FPGA-related papers published in the past 5 years. Most contributions to the national FPGA research effort stem from 16 organizations, including major Dutch universities, research institutes, and industry. Many of these publications are collaborations of national and international partners, mostly European. We note that the survey covers about 1% of the relevant world wide FPGA publications (from the same period), with the Netherlands ranking 22nd in the world and 10th in Europe in relevant FPGA-research output. We classified the selected papers into five major themes: (a) FPGA architecture, with 11 published papers, (b) data center infrastructure & HPC, with 40 papers, (c) programming models & tools, with 15 papers, (d) robustness of FPGAs, with 26 papers, and (e) applications, with 120 papers. We paid specific attention and reviewed in depth popular FPGA applications, that is those applications with a significant number of relevant publications in which FPGAs play an important role. We found 49 application papers over 6 subjects; these publications indicate that FPGAs have emerged as powerful accelerators for a wide range of applications, such as ML, astronomy, particle physics experiments, quantum computing, space applications, and bioinformatics.

⁹ <https://prodrive-technologies.com>, <https://hdlworks.com>, <https://sioux.eu>, <https://technolution.com>, <https://qbaylogic.com>, <https://demcon.com>, <https://www.core-vision.nl>

TABLE 2 Most cited applications where FPGAs are used, and reasons for preferring a design involving FPGAs. FPGA, field-programmable gate array; ASIC, application-specific integrated circuit.

Applications for FPGA design	Companies	What makes FPGAs the preferred technology	Companies
Control systems	7	High degree of parallelism	4
Near-memory processing	3	High bandwidth and throughput	3
Reliability and resilient design	3	Flexibility of development	3
Machine learning	2	Custom hardware interfacing	2
Distributed computing	2	Low latency	2
Coarse-grained architecture design	2	Affordable compared to ASIC design	2

This survey highlights the significant contributions of FPGA research in the Netherlands across multiple domains related to FPGA technology. Dutch research on FPGA architectures has led to improvements in execution time ($5\times$) and energy efficiency ($3\times$) through Near Data Processing (NDP), the reduction of communication overheads (70% reduction in average distance per cell) through application mapping frameworks to CGRAs, and higher resource efficiency ($3\times$ – $5\times$) by using NoC architectures over virtual channels. In the areas of data center infrastructure and HPC, contributions have reported improvements of 1–2 orders of magnitude in execution time for data analytics, high computational efficiency (97.25%) in distributed multi-FPGA systems, and energy improvements (31.1%) enabled by optical hardware communication. Dutch efforts have also focused on developing programming models and tools to accelerate and simplify FPGA development, and performance models to predict system performance faster ($10\times$) and more accurately (85%). Furthermore, extensive research has addressed FPGA systems' robustness leading to enhanced device reliability and hardware security. Furthermore, a plethora of FPGA-based applications have been reported, from improving accuracy of SCNNs mapped to FPGA and optimized for population genetics (97.6%) to reducing processing times for high resolution imaging in radio astronomy ($120\times$).

Our survey revealed many future directions across all themes. In terms of architecture (see Section 4), specialized FPGA architectures (e.g., for in-memory computing) and embedding FPGA technology as part of both simple and complex computing systems beyond the computation (e.g., for on-chip and off-chip communication or memory systems) are promising paths toward reducing the memory gap of traditional systems. For data centers and HPC infrastructure (see Section 5), we identify a promising research direction in FPGA-based acceleration, in general, and tooling to enable its seamless integration into large-scale systems, in particular; a second research direction should focus on supporting novel communication protocols and technologies, to further improve data movement in supercomputers, data centers, and the computing continuum. For programming models and tools (see Section 6), the main research developments target high-level tooling for easier development and deployment, accessible for more educated users other than HDL experts, and tooling support for performance analysis, modeling, and prediction, which will help solution feasibility analysis and fair comparison against competing technologies. For FPGAs' robustness (see Section 7), reliability and security are key aspects where more research is needed in

architecture and tooling for moving from proof of concept to complex (eco)systems, where FPGAs can guarantee such essential non-functional requirements in the most adverse conditions (e.g., for deployment in space). Finally, for applications, future research should focus on the adoption of FPGAs in more application domains where energy-efficient acceleration is essential; we suggest further research toward more efficient developments in AI (for both training acceleration and latency-sensitive deployment), as well as for various computing continuum layers, fast/specialized communication, and HPC kernels. We further expect more developments toward successful acceleration in simulation and digital twin applications, where FPGAs could play an interesting role in a simulation emulation continuum.

10.2 Limitations

The literature was collected by requiring at least one author to be affiliated with a Dutch organization. However, this criterion does not provide insights into the degree of involvement of the Dutch affiliated author in the research. It is possible that a co-author with a Dutch affiliation made only a minor contribution to the overall work presented in the paper. Given the limitations of the available data, determining the specific contributions of individual authors is not feasible; hence, this survey does not include an analysis of the contributions of individual authors for each paper. In addition, it is likely that key advancements from industry are not included in this survey, as it is a common practice for companies to disseminate their innovations through non-academic channels, such as white papers and/or patents. We have made efforts to engage with the industry during the course of this work and have included several papers authored by researchers at companies, as well as input provided through the questionnaire. However, this survey is primarily based on peer-reviewed publications to ensure the validity and reliability of the included contributions. This survey categorized FPGA literature by subject and provided insights into the content of FPGA research by Dutch organizations. Moreover, literature categories were grouped per Dutch organization, identifying the primary FPGA-related research direction of each Dutch organization. The survey does not identify which research groups in each organization are focusing on a specific domain of FPGA research. As future work, mapping out these research groups, and identifying the lines of research connecting them, can give further insights into the directions of FPGA research in the Netherlands. Moreover, this can aid in

identifying which FPGA-related lines of research are primarily established in the Netherlands.

10.3 Future work

Our detailed analysis of the last half-decade of Dutch FPGA research emphasizes that, while the technology holds tremendous potential, challenges exist, particularly in terms of development tools, performance predictability, and hardware security. Future research will need to focus on improving these aspects, especially regarding user-friendly programming models and more robust performance estimation frameworks, especially considering the growing availability of heterogeneous chips that tightly integrate FPGA technology with other computing architectures, such as general-purpose processors and domain-specific (mostly AI) accelerators. The continued investment in FPGA technology will be vital for maintaining the Netherlands' competitive edge and addressing the growing demand for energy-efficient, HPC solutions. Furthermore, fostering open-source tools, as well as deeper industry-academia collaboration, will be essential for sustaining long-term growth and innovation in this rapidly evolving field.

To further enlarge the scope of this analysis, we welcome broader community-driven research, to survey the developments and needs of FPGA research and innovation at a European or even world wide level. Such analysis will reveal additional collaboration opportunities, cross-theme and cross-domain, across multiple layers of modern computing systems. While we recognize the scale and scope of such an effort are significantly larger, we argue that it is an effective way to map the current landscape of FPGA research, and further focus on relevant research challenges and opportunities.

Author contributions

NA: Conceptualization, Funding acquisition, Investigation, Writing – original draft, Writing – review & editing. SB: Conceptualization, Data curation, Investigation, Writing – original draft, Writing – review & editing. SV: Investigation, Writing – original draft, Writing – review & editing. RW: Investigation, Writing – original draft, Writing – review & editing. MS: Investigation, Writing – original draft, Writing – review & editing. BE: Investigation, Writing – original draft, Writing – review &

editing. TD: Investigation, Writing – original draft, Writing – review & editing. ZA-A: Investigation, Writing – original draft, Writing – review & editing. RJ: Investigation, Writing – original draft, Writing – review & editing. AS: Investigation, Writing – original draft, Writing – review & editing. FC: Investigation, Writing – original draft, Writing – review & editing. CB: Investigation, Writing – original draft, Writing – review & editing. A-LV: Investigation, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This study was supported by 4TU.NIRICT project “NIRICT2023 - The FIRE Symposia Series: FPGA Innovation Research Exchange”, and by SURF Labs through its innovation program on “Advanced Computing”.

Conflict of interest

CB was employed at QBayLogic B.V.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Aad, G., Abbott, B., Abbott, D. C., Abed Abud, A., Abeling, K., Abhayasinghe, D. K., et al. (2020). Performance of the upgraded PreProcessor of the ATLAS Level-1 Calorimeter Trigger. *J. Instrument.* 15:P11016. doi: 10.1088/1748-0221/15/11/P11016
- Aad, G., Abbott, B., Abbott, D. C., Abud, A. A., Abeling, K., Abhayasinghe, D. K., et al. (2021). The ATLAS fast TracKer system. *J. Instrument.* 16:P07006. doi: 10.1088/1748-0221/16/07/P07006
- Abrahamse, R., Hadnagy, A., and Al-Ars, Z. (2022). “Memory-disaggregated in-memory object store framework for big data applications,” in *Proceedings - 2022 IEEE 36th International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2022*, 1228–1234. doi: 10.1109/IPDPSW55747.2022.00211
- Ahmad, T., Al-Ars, Z., and Hofstee, H. P. (2022). “Benchmarking Apache Arrow Flight - A wire-speed protocol for data transfer, querying and microservices,” in *Proceedings of the 4th Benchmarking in the Datacenter: Expanding to the Cloud 2022, BID 2022*, 22. doi: 10.1145/3527199.3527264
- Aiello, S., Albert, A., Garre, S. A., Aly, Z., Ameli, F., Andre, M., et al. (2021). Architecture and performance of the KM3NeT front-end firmware. *J. Astron. Telesc. Instr. Syst.* 7:016001. doi: 10.1117/1.JATIS.7.1.016001
- Alachiotis, N., Brokalakis, A., Amourgianos, V., Ioannidis, S., Malakonakis, P., and Bokalis, T. (2021a). Accelerating phylogenetics using FPGAs in the cloud. *IEEE Micro* 41, 24–30. doi: 10.1109/MM.2021.3075848

- Alachiotis, N., Skrimponis, P., Pissadakis, M., and Pnevmatikatos, D. (2021b). Scalable phylogeny reconstruction with disaggregated near-memory processing. *ACM Trans. Reconfig. Technol. Syst.* 15:3484983. doi: 10.1145/3484983
- Alachiotis, N., Stamatakis, A., and Pavlidis, P. (2012). Omegaplus: a scalable tool for rapid detection of selective sweeps in whole-genome datasets. *Bioinformatics* 28, 2274–2275. doi: 10.1093/bioinformatics/bts419
- Al-Ars, Z., Petri-Koenig, J., Hoozemans, J., Dierick, L., and Hofstee, H. P. (2023). "Ocoray: Framework for scalable FPGA cluster acceleration of python big data applications," in *Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis, SC-W '23* (New York, NY, USA: Association for Computing Machinery), 539–546. doi: 10.1145/3624062.3624541
- Alonso, T., Petrica, L., Ruiz, M., Petri-Koenig, J., Umuroglu, Y., Stamelos, I., et al. (2021). Elastic-DF: scaling performance of DNN inference in FPGA clouds through automatic partitioning. *ACM Trans. Reconfig. Technol. Syst.* 15:3470567. doi: 10.1145/3470567
- Anders, J., Andreu, P., Becker, B., Becker, S., Cantoro, R., Deligiannis, N. I., et al. (2023). "A survey of recent developments in testability, safety and security of RISC-V processors," in *Proceedings of the European Test Workshop, 2023-May*. doi: 10.1109/ETS56758.2023.10174099
- Arikan, K., Palumbo, A., Cassano, L., Reviriego, P., Pontarelli, S., Bianchi, G., et al. (2022). Processor security: detecting microarchitectural attacks via count-min sketches. *IEEE Trans. Very Large Scale Integr. Syst.* 30, 938–951. doi: 10.1109/TVLSI.2022.3171810
- Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature* 574, 505–510. doi: 10.1038/s41586-019-1666-5
- Ashraf, I., Khammassi, N., Taouil, M., and Bertels, K. (2018). Memory and communication profiling for accelerator-based platforms. *IEEE Trans. Comput.* 67, 934–948. doi: 10.1109/TC.2017.2785225
- Ashton, T., Backes, M., Balzer, A., Berge, D., Bolmont, J., Bonnefoy, S., et al. (2020). A NECTAr-based upgrade for the Cherenkov cameras of the H.E.S.S. 12-meter telescopes. *Astroparticle Phys.* 118:102425. doi: 10.1016/j.astropartphys.2019.102425
- Baaij, C., Kooijman, M., Kuper, J., Boeijink, W., and Gerards, M. (2010). "Clash: structural descriptions of synchronous hardware using haskell," in *Proceedings of the 13th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools (IEEE)*, 714–721. doi: 10.1109/DSD.2010.21
- Bagley, P. (2009). *KM3NeT: Technical Design Report for a Deep-Sea Research Infrastructure in the Mediterranean Sea Incorporating a Very Large Volume Neutrino Telescope*.
- Bargholz, M., Rönninger, W., and Siegle, F. (2022). "An optical spacefibre testbed for transceiver evaluation and validation of a design-time configurable router," in *2022 International SpaceWire SpaceFibre Conference (ISC)*, 01–08.
- Barrios, Y., Sanchez, A. J., Santos, L., and Sarmiento, R. (2020). SHyLoC 2.0: a versatile hardware solution for on-board data and hyperspectral image compression on future space missions. *IEEE Access* 8, 54269–54287. doi: 10.1109/ACCESS.2020.2980767
- Bhattacharya, T., and Qin, X. (2020). "Modeling energy efficiency of future green data centers," in *2020 11th International Green and Sustainable Computing Workshops (IGSC)*, 1–3. doi: 10.1109/IGSC51522.2020.9291049
- Bielski, M., Syrigos, I., Katrinis, K., Syrivelis, D., Reale, A., Theodoropoulos, D., et al. (2018). "DReDBox: Materializing a full-stack rack-scale system prototype of a next-generation disaggregated datacenter," in *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*, 1093–1098. doi: 10.23919/DATE.2018.8342174
- Bobda, C., Mbongue, J. M., Chow, P., Ewais, M., Tarafdar, N., Vega, J. C., et al. (2022). The future of FPGA acceleration in datacenters and the cloud. *ACM Trans. Reconfigurable Technol. Syst.* 15, 142. doi: 10.1145/3506713
- Bohmer, K., Forlin, B., Cazzaniga, C., Rech, P., Furano, G., Alachiotis, N., et al. (2023). "Neutron radiation tests of the NEORV32 RISC-V SoC on Flash-Based FPGAs," in *Proceedings - IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT*. doi: 10.1109/DFT59622.2023.10313556
- Broekema, P. C., Mol, J. J. D., Nijboer, R., van Amesfoort, A., Brentjens, M., Loose, G. M., et al. (2018). Cobalt: a gpu-based correlator and beamformer for lofar. *Astronomy Comput.* 23, 180–192. doi: 10.1016/j.ascom.2018.04.006
- Caulfield, A. M., Chung, E. S., Putnam, A., Angepat, H., Fowers, J., Haselman, M., et al. (2016). "A cloud-scale acceleration architecture," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (IEEE), 1–13. doi: 10.1109/MICRO.2016.7783710
- Charitopoulos, G., Pnevmatikatos, D. N., and Gaydadjiev, G. (2021). MC-DeF: creating customized CGRAs for dataflow applications. *ACM Trans. Archit. Code Optim.* 18:3447970. doi: 10.1145/3447970
- Chen, J., Daverveldt, M., and Al-Ars, Z. (2021). "FPGA acceleration of zstd compression algorithm," in *2021 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2021 - In conjunction with IEEE IPDPS 2021*, 188–191. doi: 10.1109/IPDPSW52791.2021.00035
- Choi, Y.-k., Zhang, P., Li, P., and Cong, J. (2017). "Hlscope+: fast and accurate performance estimation for FPGA hls," in *Proceedings of the 36th International Conference on Computer-Aided Design, ICCAD '17* (IEEE Press), 691–698. doi: 10.1109/ICCAD.2017.8203844
- Chrysos, G., Papapetrou, O., Pnevmatikatos, D., Dollas, A., and Garofalakis, M. (2019). "Data stream statistics over sliding windows: How to summarize 150 million updates per second on a single node," in *Proceedings - 29th International Conference on Field-Programmable Logic and Applications, FPL 2019*, 278–285. doi: 10.1109/FPL.2019.00052
- Cong, J., Lau, J., Liu, G., Neuendorffer, S., Pan, P., Vissers, K., et al. (2022). FPGA hls today: Successes, challenges, and opportunities. *ACM Trans. Reconfigurable Technol. Syst.* 15:3530775. doi: 10.1145/3530775
- Corda, S., Veenboer, B., Awan, A. J., Kumar, A., Jordans, R., and Corporaal, H. (2020). "Near memory acceleration on high resolution radio astronomy imaging," in *2020 9th Mediterranean Conference on Embedded Computing, MECO 2020*. doi: 10.1109/MECO49872.2020.9134089
- Corda, S., Veenboer, B., Awan, A. J., Romein, J. W., Jordans, R., Kumar, A., et al. (2022). Reduced-precision acceleration of radio-astronomical imaging on reconfigurable hardware. *IEEE Access* 10, 22819–22843. doi: 10.1109/ACCESS.2022.3150861
- Corradi, F., Adriaans, G., and Stuijk, S. (2021). "Gyro: a digital spiking neural network architecture for multi-sensory data analytics," in *ACM International Conference Proceeding Series*, 9–15. doi: 10.1145/3444950.3444951
- Corradi, F., Shen, Z., Zhao, H., and Alachiotis, N. (2024). "Accelerated spiking convolutional neural networks for scalable population genomics," in *Proceedings of the 14th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies*, 53–62. doi: 10.1145/3665283.3665285
- Corts, R., and Alachiotis, N. (2023). A survey of processing systems for phylogenetics and population genetics. *ACM Trans. Reconfigurable Technol. Syst.* 16:3588033. doi: 10.1145/3588033
- Corts, R., Sterenborg, N., and Alachiotis, N. (2022). "Accelerated LD-based selective sweep detection using GPUs and FPGAs," in *Proceedings - 2022 IEEE 36th International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2022*, 196–205. doi: 10.1109/IPDPSW55747.2022.00044
- Cromjongh, C., Tian, Y., Hofstee, P., and Al-Ars, Z. (2023). "Tydi-chisel: collaborative and interface-driven data-streaming accelerators," in *2023 IEEE Nordic Circuits and Systems Conference (NorCAS)*, 1–7. doi: 10.1109/NorCAS58970.2023.10305451
- Curzel, S., Fiorito, M., Cueva, P. L., Jorge, T., Tsiordas, T., and Ferrandi, F. (2023). "Exploration of synthesis methods from simlink models to FPGA for aerospace applications," in *Proceedings of the 20th ACM International Conference on Computing Frontiers 2023, CF 2023*, 243–249. doi: 10.1145/3587135.3592766
- de Bruin, B., Vadiel, K., Wijtvliet, M., Jäskeläinen, P., and Corporaal, H. (2024). "R-blocks: an energy-efficient, flexible, and programmable cgra," in *ACM Transactions on Reconfigurable Technology and Systems*. doi: 10.1145/3656642
- De Sio, C., Azimi, S., Sterpone, L., Codinachs, D. M., and Decuzzi, F. (2023). "PyXEL: exploring bitstream analysis to assess and enhance the robustness of designs on FPGAs," in *Proceedings - 2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods, and Applications to Circuit Design, SMACD 2023*. doi: 10.1109/SMACD58065.2023.10192116
- De, S., Shafique, M., and Corporaal, H. (2023). Delay prediction for asic hls: Comparing graph-based and nongraph-based learning models. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 42, 1133–1146. doi: 10.1109/TCAD.2022.3197977
- Deng, X., Zhang, M., Pan, W., Gao, Z., Mo, J., Chen, C., et al. (2022). Physics-based LED modeling and nonlinear distortion mitigating with real-time implementation. *IEEE Photonics J.* 14:3210021. doi: 10.1109/JPHOT.2022.3210021
- Diamantopoulos, D., Ringlein, B., Purandare, M., Singh, G., and Hagleitner, C. (2020). "Agile autotuning of a transprecision tensor accelerator overlay for tvn compiler stack," in *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*, 310–316. doi: 10.1109/FPL50879.2020.00058
- Du, B., Sterpone, L., Azimi, S., Codinachs, D. M., Ferlet-Cavrois, V., Polo, C. B., et al. (2019). Ultrahigh energy heavy ion test beam on xilinx kintex-7 SRAM-Based FPGA. *IEEE Trans. Nucl. Sci.* 66, 1813–1819. doi: 10.1109/TNS.2019.2915207
- Education Statistics, N. C. (2022). *Full-time-equivalent (fte) staff, fte faculty, and ratios of fte students to fte staff and fte faculty in public degree-granting postsecondary institutions, by level of institution and state or jurisdiction: Fall 2022*. Available online at: https://nces.ed.gov/programs/digest/d23/tables/dt23_314.50.asp (accessed April 8, 2025).
- Ernst, D. (2020). *Competing in Artificial Intelligence Chips: China's Challenge Amid Technology War*. Centre for International Governance Innovation. Available online at: <https://www.cigionline.org/publications/competing-artificial-intelligence-chips-chinas-challenge-amid-technology-war/> (Accessed October 1, 2024).
- European Commission (2024). *Space Qualification and Validation of High Performance European Rad-Hard FPGA | OPERA Project | Fact Sheet | H2020 | CORDIS*

| European Commission — cordis.europa.eu, howpublished. Available online at: <https://cordis.europa.eu/project/id/821969> (accessed February 19, 2024).

Eurostat (2022). *Randd personnel by sector of performance, professional position and sex*.

Fang, J., Chen, J., Lee, J., Al-Ars, Z., and Hofstee, H. P. (2019). "A Fine-grained parallel snappy decompressor for FPGAS using a relaxed execution model," in *Proceedings - 27th IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM 2019*, 335. doi: 10.1109/FCCM.2019.00076

Fang, J., Chen, J., Lee, J., Al-Ars, Z., and Hofstee, H. P. (2020a). An efficient high-throughput LZ77-based compressor in reconfigurable logic. *J. Signal Process. Syst.* 92, 931–947. doi: 10.1007/s11265-020-01547-w

Fang, J., Mulder, Y. T., Hidders, J., Lee, J., and Hofstee, H. P. (2020b). In-memory database acceleration on FPGAs: a survey. *Vldb J.* 29, 33–59. doi: 10.1007/s00778-019-00581-w

Fernandez Prieto, A., Regueiro, P. V., Hennessy, K., Buytaert, J., Van Beuzekom, M., Cid, E. L., et al. (2020). Phase I upgrade of the readout system of the vertex detector at the LHCb experiment. *IEEE Trans. Nucl. Sci.* 67, 732–739. doi: 10.1109/TNS.2020.2970534

Firestone, D., Putnam, A., Mundkur, S., Chiou, D., Dabagh, A., Andrewartha, M., et al. (2018). "Azure accelerated networking: SmartNICs in the public cloud," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 51–66.

Flich, J., Agosta, G., Ampletzer, P., Alonso, D. A., Brandolese, C., Cappe, E., et al. (2018). Exploring manycore architectures for next-generation HPC systems through the MANGO approach. *Microprocess. Microsyst.* 61, 154–170. doi: 10.1016/j.micpro.2018.05.011

Forlin, B. E., Van Huffelen, W., Cazzaniga, C., Rech, P., Alachiotis, N., and Ottavi, M. (2023). "An unprotected RISC-V Soft-core processor on an SRAM FPGA: is it as bad as it sounds?" in *Proceedings of the European Test Workshop, 2023-May*. doi: 10.1109/ETS56758.2023.10174076

Gambardella, G., Fraser, N. J., Zahid, U., Furano, G., and Blott, M. (2022). "Accelerated radiation test on quantized neural networks trained with fault aware training," in *IEEE Aerospace Conference Proceedings, 2022-March*. doi: 10.1109/AERO53065.2022.9843614

Garaffa, L. C., Aljuffri, A., Reinbrecht, C., Hamdioui, S., Taouil, M., and Sepúlveda, J. (2021). "Revealing the secrets of spiking neural networks: the case of izhikevich neuron," in *Proceedings - 2021 24th Euromicro Conference on Digital System Design, DSD 2021*, 514–518. doi: 10.1109/DSD53832.2021.00083

Gibney, E. (2014). Physics: quantum computer quest. *Nature* 516, 24–26. doi: 10.1038/516024a

González, Y. R., Nelissen, G., and Tovar, E. (2022). "Ipdn: Real-time deflection-based noc with in-order flits delivery," in *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 160–169. doi: 10.1109/RTCSA55878.2022.00023

Gottardo, C. A. (2020). "FELIX and SW rod commissioning of the new ATLAS readout system," in *2020 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2020*. doi: 10.1109/NSS/MIC42677.2020.9507984

Groet, P., Hoozemans, J., Grapentin, A., Eberhardt, F., Al-Ars, Z., and Hofstee, H. P. (2024). Leveraging apache arrow for zero-copy, zero-serialization cluster shared memory. *arXiv preprint arXiv:2404.03030*.

Group, K. (2015). *The open standard for parallel programming of heterogeneous systems*. Available online at: <https://www.khronos.org/OpenGL/> (accessed September 9, 2024).

Guerrieri, A., Silva Marques, G. D., Regazzoni, F., and Upegui, A. (2022). "Optimizing lattice-based post-quantum cryptography codes for high-level synthesis," in *Proceedings - 2022 25th Euromicro Conference on Digital System Design, DSD 2022*, 777–784. doi: 10.1109/DSD57027.2022.00109

Guo, X., Xue, X., Yan, F., Pan, B., Exarchakos, G., and Calabretta, N. (2022). DACON: a reconfigurable application-centric optical network for disaggregated data center infrastructures [Invited]. *J. Opt. Commun. Network.* 14, A69–A80. doi: 10.1364/JOCN.438950

Haarman, T., Almeida, A. S., d., Heskes, A., Zwanenburg, F., and Alachiotis, N. (2023). "FPGA-accelerated quantum transport measurements," in *2023 International Conference on Field Programmable Technology (ICFPT)*, 44–52. doi: 10.1109/ICFPT59805.2023.00010

Hall, P. J. (2005). *The Square Kilometre Array: An Engineering Perspective*. Cham: Springer. doi: 10.1007/1-4020-3798-8

Hampson, G., Buntun, J., Gunst, A., Baillie, P., and Vaate, J. (2016). "Introduction to the ska low correlator and beamformer system," in *Proceedings Volume 9906, Ground-based and Airborne Telescopes VI*, 99062S. doi: 10.1117/12.2231524

Hessels, L., Tai, S.-Y. T. T., Jansen, J., and Deuten, J. (2024). *Europese wetenschap en innovatie in een nieuw geopolitiek speelveld | Rathenau Instituut*. Available online at: <https://www.rathenau.nl/nl/werking-van-het-wetenschapssysteem/europese-wetenschap-en-innovatie-een-nieuw-geopolitiek-speelveld> (accessed February 19, 2024).

Hirvonen, A., Leppänen, T., Hepola, K., Multanen, J., Hoozemans, J., and Jääskeläinen, P. (2023). AEX: automated high-level synthesis of compiler programmable co-processors. *J. Signal Process. Syst.* 95, 1051–1065. doi: 10.1007/s11265-023-01841-3

Homulle, H., Visser, S., Patra, B., Ferrari, G., Prati, E., Sebastiano, F., et al. (2017). A reconfigurable cryogenic platform for the classical control of quantum processors. *Rev. Sci. Instr.* 88:045103. doi: 10.1063/1.4979611

Hoozemans, J., Peltenburg, J., Nonnemacher, F., Hadnagy, A., Al-Ars, Z., and Hofstee, H. P. (2021a). FPGA acceleration for big data analytics: challenges and opportunities. *IEEE Circ. Syst. Mag.* 21, 30–47. doi: 10.1109/MCAS.2021.3071608

Hoozemans, J., Tervo, K., Jaaskelainen, P., and Al-Ars, Z. (2021b). "Energy efficient multistandard decompressor ASIP," in *ACM International Conference Proceeding Series, Part F 174233*, 14–19. doi: 10.1145/3456172.3456218

Hoozemans, J., van Straten, J., and Wong, S. (2018). Increasing resource utilization in mixed-criticality systems using a polymorphic VLIW processor. *J. Syst. Archit.* 84, 2–11. doi: 10.1016/j.sysarc.2018.01.003

Hu, T., Chitnis, N., Monos, D., and Dinh, A. (2021). "Next-generation sequencing technologies: an overview. *Human Immunol.* 82, 801–811. doi: 10.1016/j.humimm.2021.02.012

Huang, S., Wu, K., Jeong, H., Wang, C., Chen, D., and Hwu, W.-M. (2021). Pylog: an algorithm-centric python-based FPGA programming and synthesis flow. *IEEE Trans. Comput.* 70, 2015–2028. doi: 10.1145/3431920.3439478

Iakovidis, G., Levinson, L., Afik, Y., Alexa, C., Alexopoulos, T., Ameal, J., et al. (2023). The New Small Wheel electronics. *J. Instrument.* 18:P05012. doi: 10.1088/1748-0221/18/05/P05012

Iosup, A., Prodan, R., Varbanescu, A. L., Talluri, S., Magalhaes, G., Hokstam, K., et al. (2023). "Graph greenifier: towards sustainable and energy-aware massive graph processing in the computing continuum," in *ICPE 2023 - Companion of the 2023 ACM/SPEC International Conference on Performance Engineering*, 209–214.

Ireland, J., Protheroe, S., Williams, J., Belcher, A., Dekker, R., Schaapman, K., et al. (2020). "Real-time quantum-accurate voltage waveform synthesis," in *2020 Conference on Precision Electromagnetic Measurements (CPEM)*, 1–2. doi: 10.1109/CPEM49742.2020.9191715

Irmak, H., Alachiotis, N., and Ziener, D. (2021a). "An energy-efficient FPGA-based convolutional neural network implementation," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 1–4. doi: 10.1109/SIU53274.2021.9477823

Irmak, H., Corradi, F., Detterer, P., Alachiotis, N., and Ziener, D. (2021b). A dynamic reconfigurable architecture for hybrid spiking and convolutional FPGA-based neural network designs. *J. Low Power Electr. Applic.* 11:32. doi: 10.3390/jlpea11030032

Irmak, H., Ziener, D., and Alachiotis, N. (2021c). "Increasing flexibility of FPGA-based CNN accelerators with dynamic partial reconfiguration," in *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*, 306–311. doi: 10.1109/FPL53798.2021.00061

Ji, M., Al-Ars, Z., Hofstee, P., Chang, Y., and Zhang, B. (2023). Fpqnet: fully pipelined and quantized CNN for ultra-low latency image classification on FPGAs using opencapi. *Electronics* 12:4085. doi: 10.3390/electronics12194085

Jin, C., Burleson, W., Van Dijk, M., and Rührmair, U. (2020). "Erasable PUFs: formal treatment and generic design," in *ASHES 2020 - Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security*, 21–33. doi: 10.1145/3411504.3421215

Jin, C., Burleson, W., van Dijk, M., and Rührmair, U. (2022). Programmable access-controlled and generic erasable PUF design and its applications. *J. Cryptogr. Eng.* 12, 413–432. doi: 10.1007/s13389-022-00284-z

Karakakula, S., Brinkerink, C., Vecchio, A., Pourshaghagh, H. R., Dolron, P., Jordans, R., et al. (2024). Architecture design and ground performance of netherlands-china low-frequency explorer. *Radio Sci.* 59, 1–39. doi: 10.1029/2023RS007906

Katal, A., Dahiya, S., and Choudhury, T. (2023). Energy efficiency in cloud computing data centers: a survey on software technologies. *Cluster Comput.* 26, 1845–1875. doi: 10.1007/s10586-022-03713-0

Katevenis, M., Ammendola, R., Biagioni, A., Cretaro, P., Frezza, O., Lo Cicero, F., et al. (2018). Next generation of Exascale-class systems: ExaNeSt project and the status of its interconnect and storage development. *Microprocess. Microsyst.* 61, 58–71. doi: 10.1016/j.micpro.2018.05.009

Kim, Y., and Nielsen, R. (2004). Linkage disequilibrium as a signature of selective sweeps. *Genetics* 167, 1513–1524. doi: 10.1534/genetics.103.025387

Köylü, T., a., Wedig Reinbrecht, C. R., Brandalero, M., Hamdioui, S., and Taouil, M. (2022). Instruction flow-based detectors against fault injection attacks. *Microprocess. Microsyst.* 94:104638. doi: 10.1016/j.micpro.2022.104638

Koylu, T., Garaffa, L., Reinbrecht, C., Zahedi, M., Hamdioui, S., and Taouil, M. (2022). "Exploiting PUF variation to detect fault injection attacks," in *Proceedings - 2022 25th International Symposium on Design and Diagnostics of Electronic Circuits and Systems, DDECS 2022*, 74–79. doi: 10.1109/DDECS54261.2022.9770154

Kundel, R., Nobach, L., Blendin, J., Maas, W., Zimmer, A., Kolbe, H. J., et al. (2021). OpenBNG: central office network functions on programmable data plane hardware. *Int. J. Netw. Manag.* 31:e2134. doi: 10.1002/nem.2134

- Labafniya, M., Picke, S., Etemadi Borujeni, S., and Mentens, N. (2020). On the feasibility of using evolvable hardware for hardware Trojan detection and prevention. *Appl. Soft Comput.* 91:106247. doi: 10.1016/j.asoc.2020.106247
- Lahr, N., Niederhagen, R., Petri, R., and Samardjiska, S. (2020). "Side Channel Information Set Decoding Using Iterative Chunking," in *Plaintext Recovery from the "Classic McEliece" Hardware Reference Implementation*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 881–910. doi: 10.1007/978-3-030-64837-4_29
- Lahti, S., Sjövall, P., Vanne, J., and Hämäläinen, T. D. (2019). Are we there yet? A study on the state of high-level synthesis. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* 38, 898–911. doi: 10.1109/TCAD.2018.2834439
- Lai, Y.-H., Chi, Y., Hu, Y., Wang, J., Yu, C. H., et al. (2019). "HeteroCL: a multi-paradigm programming infrastructure for software-defined reconfigurable computing," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'19)*, 242–251. doi: 10.1145/3289602.3293910
- Lattner, C., and Adve, V. (2004). "Llvm: a compilation framework for lifelong program analysis transformation," in *International symposium on code generation and optimization, 2004. CGO 2004 (IEEE)*, 75–86. doi: 10.1109/CGO.2004.1281665
- Lenkiewicz, P., Broekema, P. C., and Metzler, B. (2018). Energy-efficient data transfers in radio astronomy with software udp rdma. *Fut. Gener. Comput. Syst.* 79, 215–224. doi: 10.1016/j.future.2017.03.027
- Leon, V., Bezaitis, C., Lentar, G., Soudris, D., Reisis, D., Papatheofanous, E. A., et al. (2021a). "FPGA VPU co-processing in space applications: development and testing with DSP/AI benchmarks," in *2021 28th IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2021 - Proceedings*. doi: 10.1109/ICECS53924.2021.9665462
- Leon, V., Lentar, G., Petrongonas, E., Soudris, D., Furano, G., Tavoularis, A., et al. (2021b). "Improving performance-power-programmability in space avionics with edge devices: VBN on Myriad2 SoC," in *ACM Transactions on Embedded Computing Systems (TECS)*. doi: 10.1145/3440885
- Leon, V., Stamoulis, I., Lentar, G., Soudris, D., Gonzalez-Arjona, D., Domingo, R., et al. (2021c). Development and testing on the European space-grade BRAVE FPGAs: evaluation of NG-large using high-performance DSP benchmarks. *IEEE Access* 9, 131877–131892. doi: 10.1109/ACCESS.2021.3114502
- Licht, J. d. F., De Matteis, T., Ben-Nun, T., Kuster, A., Rausch, O. (2022). Python FPGA programming with data-centric multi-level design. *arXiv preprint arXiv:2212.13768*.
- Malakonakis, P., Brokalakis, A., Alachiotis, N., Sotiriades, E., and Dollas, A. (2020). "Exploring modern FPGA Platforms for faster phylogeny reconstruction with RAXML," in *Proceedings - IEEE 20th International Conference on Bioinformatics and Bioengineering, BIBE 2020*, 97–104. doi: 10.1109/BIBE50027.2020.00024
- Massolino, P. M. C., Longa, P., Renes, J., and Batina, L. (2020). A compact and scalable hardware/software co-design of SIKE. *Cryptol. ePrint Archive* 2020, 245–271. doi: 10.46586/tches.v2020.i2.245-271
- Meyer, H., Sancho, J. C., Mrdakovic, M., Miao, W., and Calabretta, N. (2018). Optical packet switching in HPC. An analysis of applications performance. *Future Gener. Comput. Syst.* 82, 606–616. doi: 10.1016/j.future.2017.02.041
- Miteloudi, K., Batina, L., Daemen, J., and Mentens, N. (2022). "ROCKY: rotation countermeasure for the protection of keys and other sensitive data," in *Embedded Computer Systems: Architectures, Modeling, and Simulation: 21st International Conference, SAMOS 2021*, 288–299. doi: 10.1007/978-3-031-04580-6_19
- Miteloudi, K., Chmielewski, L., Batina, L., and Mentens, N. (2021). "Evaluating the ROCKY countermeasure for side-channel leakage," in *IEEE/IFIP International Conference on VLSI and System-on-Chip, VLSI-SoC, 2021-October*. doi: 10.1109/VLSI-SoC53125.2021.9606973
- Moonen, N., Vogt-Ardatjew, R., and Leferink, F. (2021). Simulink-based FPGA control for EMI investigations of power electronic systems. *IEEE Trans. Electromag. Compatib.* 63, 1266–1273. doi: 10.1109/TEM.2020.3042301
- Mousavi, M., Pourshaghghi, H. R., Kumar, A., and Corporaal, H. (2023). MTTR reduction of FPGA scrubbing: exploring SEU sensitivity. *Microprocess. Microsyst.* 101:104841. doi: 10.1016/j.micpro.2023.104841
- Mystkowska, G., and Steenari, D. (2022). "Simulation and hardware validation of SerDes links for SpaceFibre," in *2022 International SpaceWire & SpaceFibre Conference (ISC)*
- Nechi, A., Groth, L., Mulhem, S., Merchant, F., Buchty, R., and Berekovic, M. (2023). FPGA-based deep learning inference accelerators: where are we standing? *ACM Trans. Reconfigurable Technol. Syst.* 16:3613963. doi: 10.1145/3613963
- Nederland en Horizon 2020 (2024). Rathenau Instituut. Available online at: <https://www.rathenau.nl/nl/wetenschap-cijfers/nederland-en-horizon-2020> (accessed February 19, 2024).
- Nikiema, P. R., Palumbo, A., Aasma, A., Cassano, L., Kritikakou, A., Kulmala, A., et al. (2023). "Towards dependable RISC-V cores for edge computing devices," in *Proceedings - 2023 IEEE 29th International Symposium on On-Line Testing and Robust System Design, IOLTS 2023*. doi: 10.1109/IOLTS59296.2023.10224862
- Niu, Y., Deng, X., Fan, W., Mo, J., Chen, C., Linnartz, J. P. M., et al. (2021). "LED nonlinearity post-compensator with legendre polynomials in visible light communications," in *Proceedings of the 16th IEEE Conference on Industrial Electronics and Applications, ICIEA 2021*, 1322–1327. doi: 10.1109/ICIEA51954.2021.9516222
- Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., and Marr, D. (2016). "Accelerating binarized neural networks: comparison of FPGA, CPU, GPU, and ASIC," in *2016 International Conference on Field-Programmable Technology (FPT)*, 77–84. doi: 10.1145/FPT.2016.7929192
- Ofek, N., Petrenko, A., Heeres, R., Reinhold, P., Leghtas, Z., Vlastakis, B., et al. (2016). Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* 536, 441–445. doi: 10.1038/nature18949
- O'Neal, K., Liu, M., Tang, H., Kalantar, A., DeRenard, K., and Brisk, P. (2018). "Hlspredict: cross platform performance prediction for FPGA high-level synthesis," in *Proceedings of the International Conference on Computer-Aided Design, ICCAD '18 (New York, NY, USA: Association for Computing Machinery)*. doi: 10.1145/3240765.3264635
- Overwater, R. W. J., Babaie, M., and Sebastiano, F. (2022). Neural-network decoders for quantum error correction using surface codes: a space exploration of the hardware cost-performance tradeoffs. *IEEE Trans. Quant. Eng.* 3, 1–19. doi: 10.1109/TQE.2022.3174017
- Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C. (2008). Gpu computing. *Proc. IEEE* 96, 879–899. doi: 10.1109/JPROC.2008.917757
- Palumbo, A., Cassano, L., Luzzi, B., Hernández, J. A., Reviriego, P., Bianchi, G., et al. (2022). Is your FPGA bitstream Hardware Trojan-free? Machine learning can provide an answer. *J. Syst. Archit.* 128:102543. doi: 10.1016/j.sysarc.2022.102543
- Peltenburg, J., Hadnagy, A., Brobbel, M., Morrow, R., and Al-Ars, Z. (2021a). "Tens of gigabytes per second JSON-to-Arrow conversion with FPGA accelerators," in *2021 International Conference on Field-Programmable Technology, ICFPT 2021*. doi: 10.1109/ICFPT52863.2021.9609833
- Peltenburg, J., Van Leeuwen, L. T., Hoozemans, J., Fang, J., Al-Ars, Z., and Hofstee, H. P. (2020a). "Battling the CPU bottleneck in apache parquet to arrow conversion using FPGA," in *Proceedings - 2020 International Conference on Field-Programmable Technology, ICFPT 2020*, 281–286. doi: 10.1109/ICFPT51103.2020.00048
- Peltenburg, J., Van Straten, J., Brobbel, M., Al-Ars, Z., and Hofstee, H. P. (2020b). Tydi: an open specification for complex data structures over hardware streams. *IEEE Micro* 40, 120–130. doi: 10.1109/MM.2020.2996373
- Peltenburg, J., van Straten, J., Brobbel, M., Al-Ars, Z., and Hofstee, H. P. (2021b). Generating high-performance FPGA accelerator designs for big data analytics with fletcher and apache arrow. *J. Signal Process. Syst.* 93, 565–586. doi: 10.1007/s11265-021-01650-6
- Peltenburg, J., Van Straten, J., Wijtemans, L., Van Leeuwen, L., Al-Ars, Z., and Hofstee, P. (2019). "Fletcher: a framework to efficiently integrate FPGA accelerators with apache arrow" in *Proceedings - 29th International Conference on Field-Programmable Logic and Applications, FPL 2019*, 270–277. doi: 10.1109/FPL.2019.00051
- Pes, L., Luiken, R., Corradi, F., and Frenkel, C. (2024). "Active dendrites enable efficient continual learning in time-to-first-spike neural networks," in *International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. doi: 10.1109/AICAS59952.2024.10595872
- Petrica, L., Alonso, T., Kroes, M., Fraser, N., Cotofana, S., and Blott, M. (2020). "Memory-efficient dataflow inference for deep CNNs on FPGA," in *2020 International Conference on Field-Programmable Technology (ICFPT)*, 48–55. doi: 10.1109/ICFPT51103.2020.00016
- Philips, S. G. J., Madzik, M. T., Amitonov, S. V., de Snoo, S. L., Russ, M., Kalhor, N., et al. (2022). Universal control of a six-qubit quantum processor in silicon. *Nature* 609, 919–924. doi: 10.1038/s41586-022-05117-x
- Pinto, C., Syrivelis, D., Gazzetti, M., Koutsovasilis, P., Reale, A., Katrinis, K., et al. (2020). "Thymesisflow: a software-defined, hw/sw co-designed interconnect stack for rack-scale memory disaggregation," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 868–880. doi: 10.1109/MICRO50266.2020.00075
- Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* 117, 1–19. doi: 10.1006/jcph.1995.1039
- Portaluri, A., Azimi, S., De Sio, C., Sterpone, L., and Codinachs, D. M. (2022). "Radiation-induced effects on DMA data transfer in reconfigurable devices," in *Proceedings - 2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design, IOLTS 2022*. doi: 10.1109/IOLTS56730.2022.9897262
- Prodan, R., Kimovski, D., Bartolini, A., Cochez, M., Iosup, A., Kharlamov, E., et al. (2022). "Towards extreme and sustainable graph processing for urgent societal challenges in Europe," in *Proceedings - 2022 IEEE Cloud Summit, Cloud Summit 2022*, 23–30. doi: 10.1109/CloudSummit54781.2022.00010

- Putnam, A., Caulfield, A. M., Chung, E. S., Chiou, D., Constantinides, K., Demme, J., et al. (2015). A reconfigurable fabric for accelerating large-scale datacenter services. *IEEE Micro* 35, 10–20. doi: 10.1109/MM.2015.42
- Rapuano, E., Meoni, G., Pacini, T., Dinelli, G., Furano, G., Giuffrida, G., et al. (2021). An FPGA-based hardware accelerator for cnns inference on board satellites: benchmarking with myriad 2-based solution for the cloudscout case study. *Rem. Sensing* 13:1518. doi: 10.3390/rs13081518
- Rellermeyer, J. S., Omranian Khorasani, S., Graur, D., and Parthasarathy, A. (2019). “The coming age of pervasive data processing,” in *Proceedings - 2019 18th International Symposium on Parallel and Distributed Computing, ISPD 2019*, 58–65. doi: 10.1109/ISPD.2019.00011
- Reukers, M. A., Tian, Y., Al-Ars, Z., Hofstee, P., Brobbel, M., Peltenburg, J., et al. (2023). An intermediate representation for composable typed streaming dataflow designs. *ArXiv, abs/2308.13436*.
- Ribot Gonzalez, Y., and Nelissen, G. (2020). HopliteRT*: Real-Time NoC for FPGA. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* 39, 3650–3661. doi: 10.1109/TCAD.2020.3012748
- Ribot González, Y., and Nelissen, G. (2020). Hoplitert*: Real-time noc for FPGA. *IEEE Trans. Comput.-Aided Design Integr. Circ. Syst.* 39, 3650–3661. doi: 10.1109/TCAD.2020.3012748
- Ribot González, Y., Nelissen, G., and Tovar, E. (2021). “nDimNoC: real-time D-dimensional NoC,” in *33rd Euromicro Conference on Real-Time Systems (ECRTS 2021)*, volume 196 of *Leibniz International Proceedings in Informatics (LIPIcs)* (Dagstuhl, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik), 1–5.
- Rocha, L. G., Biswas, D., Verhoef, B. E., Bampi, S., Van Hoof, C., Konijnenburg, M., et al. (2020). Binary CorNET: accelerator for HR estimation from Wrist-PPG. *IEEE Trans. Biomed. Circuits Syst.* 14, 715–726. doi: 10.1109/TBCAS.2020.3001675
- Rodriguez, A., Santos, L., Sarmiento, R., and Torre, E. D. L. (2019). Scalable hardware-based on-board processing for run-time adaptive lossless hyperspectral compression. *IEEE Access* 7, 10644–10652. doi: 10.1109/ACCESS.2019.2892308
- Rowe, S., Tapia, M., Barry, P. S., Karkare, K. S., Papageorgiou, A., Ade, P. A., et al. (2023). The MUSCAT Readout Electronics Backend: Design and Pre-deployment Performance. *J. Low Temp. Phys.* 211, 289–301. doi: 10.1007/s10909-022-02868-9
- Rubattu, C., Errico, W., and Marinis, K. (2022). “A low-overhead AXI bridge on SpaceWire for multi-device spacecraft systems,” in *2022 International SpaceWire & SpaceFibre Conference (ISC)*.
- Sadasivam, S. K., Thompto, B. W., Kalla, R., and Starke, W. J. (2017). Ibm power9 processor architecture. *IEEE Micro* 37, 40–51. doi: 10.1109/MM.2017.40
- Sahebi, A., Barbone, M., Procaccini, M., Luk, W., Gaydadjiev, G., and Giorgi, R. (2023). Distributed large-scale graph processing on FPGAs. *J. Big Data* 10, 1–28. doi: 10.1186/s40537-023-00756-x
- Salazar-Garcia, C., Chacon-Rodriguez, A., Rfmolo-Donadfo, R., Garcia-Ramirez, R., Solorzano-Pacheco, D., Gonzalez-Gomez, J., et al. (2022). “A custom interconnection multi-FPGA framework for distributed processing applications,” in *35th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design, SBCCI 2022 - Proceedings*. doi: 10.1109/SBCCI5532.2022.9893238
- Salazar-Garcia, C., Garcia-Ramirez, R., Rimolo-Donadio, R., Strydis, C., and Chacón-Rodríguez, A. (2021). “PlasticNet+: extending multi-FPGA interconnect architecture via Gigabit transceivers,” in *Proceedings - IEEE International Symposium on Circuits and Systems, 2021-May*. doi: 10.1109/ISCAS51556.2021.9401058
- Salazar-Garcia, C., González-Gómez, J., Alfaro-Badilla, K., García-Ramírez, R., Rimolo-Donadio, R., Strydis, C., et al. (2020). “Plasticnet: a low latency flexible network architecture for interconnected multi-FPGA systems,” in *2020 IEEE 3rd Conference on PhD Research in Microelectronics and Electronics in Latin America (PRIME-LA) (IEEE)*, 1–4. doi: 10.1109/PRIME-LA47693.2020.9062749
- Sanchez, A. J., Barrios, Y., Ventura, D., Santos, L., and Sarmiento, R. (2022). “Abeto framework: a Solution for Heterogeneous IP Management,” in *Proceedings - 2022 25th Euromicro Conference on Digital System Design, DSD 2022*, 720–725. doi: 10.1109/DSD57027.2022.00101
- Sanchez-Garrido, J., Jurado, A., Jimenez-Lopez, M., Balzer, A., Prokoph, H., Stephan, M., et al. (2021). A white rabbit-synchronized accurate time-stamping solution for the small-sized cameras of the cherenkov telescope array. *IEEE Trans. Instrum. Meas.* 70:3013343. doi: 10.1109/TIM.2020.3013343
- Sankaran, A., Dettler, P., Ni, D., Kannan, K., Alachiotis, N., Ni, A., et al. (2022). “An event-driven recurrent spiking neural network architecture for efficient inference on FPGA architecture for efficient inference,” in *ICONS '22: Proceedings of the International Conference on Neuromorphic Systems 2022*. doi: 10.1145/3546790.3546802
- Sano, K., Koshiba, A., Miyajima, T., and Ueno, T. (2023). “ESSPER: elastic and scalable FPGA-cluster system for high-performance reconfigurable computing with supercomputer fugaku,” in *ACM International Conference Proceeding Series (Association for Computing Machinery)*, 140–150. doi: 10.1145/3578178.3579341
- Santana, H., Mefleh, A., and Calabretta, N. (2023). “SOA-based optical networks with sub-microsecond control plane for low-latency applications,” in *International Conference on Transparent Optical Networks, 2023-July*. doi: 10.1109/ICTON59386.2023.10207247
- Santana, H., Pan, B., Kraemer, R., Prifti, K., Mefleh, A., and Calabretta, N. (2022). “Transparent and fast reconfigurable optical network with edge computing nodes for beyond 5G applications,” in *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2022*, 820–825. doi: 10.1109/CSNDSP54353.2022.9907904
- Schmidt, B., and Hildebrandt, A. (2017). Next-generation sequencing: big data meets high performance computing. *Drug Discov. Today* 22, 712–717. doi: 10.1016/j.drudis.2017.01.014
- Schoonderbeek, G., Hut, B., Kooistra, E., Mika, P. H. J., Gunst, A. W., and Van Cappellen, W. A. (2020). Implementation of a correlator onto a hardware beam-former to calculate beam-weights. *J. Astron. Instr.* 9:75. doi: 10.1142/S2251171720500075
- Schoonderbeek, G. W., Szomoru, A., Gunst, A. W., Hiemstra, L., and Hargreaves, J. (2019). Uniboard2, a generic scalable high-performance computing platform for radio astronomy. *J. Astron. Instr.* 08:1950003. doi: 10.1142/S225117171950003X
- Scopus (2024). *Abstract and citation database*. Available online at: <https://www.elsevier.com/products/scopus> (accessed November 10, 2023).
- Shahroodi, T., Zahedi, M., Singh, A., Wong, S., and Hamdioui, S. (2022). “KrakenOnMem: a memristor-augmented HW/SW framework for taxonomic profiling,” in *Proceedings of the International Conference on Supercomputing*, 14. doi: 10.1145/3524059.3532367
- Sidiropoulos, H., Chatzikonstantis, G., Soudris, D., and Strydis, C. (2018). The vineyard framework for heterogeneous cloud applications: the brainframe case. *IEEE Comput. Soc.* 2018, 70–75. doi: 10.1109/DASIP.2018.8597119
- Singh, G., Alser, M., Cali, D. S., Diamantopoulos, D., Gomez-Luna, J., Corporaal, H., et al. (2021a). FPGA-based near-memory acceleration of modern data-intensive applications. *IEEE Micro* 41, 39–48. doi: 10.1109/MM.2021.3088396
- Singh, G., Chelini, L., Corda, S., Awan, A. J., Stuijk, S., Jordans, R., et al. (2019). Near-memory computing: past, present, and future. *Microprocess. Microsyst.* 71:102868. doi: 10.1016/j.micpro.2019.102868
- Singh, G., Diamantopoulos, D., Gómez-Luna, J., Stuijk, S., Mutlu, O., and Corporaal, H. (2021b). “Modeling FPGA-based systems via few-shot learning,” in *FPGA '21: The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 146–146. doi: 10.1145/3431920.3439460
- Singh, G., Diamantopoulos, D., Hagleitner, C., Gómez-Luna, J., Stuijk, S., Mutlu, O., et al. (2020). “NERO: a near high-bandwidth memory stencil accelerator for weather prediction modeling,” in *Proceedings - 30th International Conference on Field-Programmable Logic and Applications, FPL 2020*, 9–17. doi: 10.1109/FPL50879.2020.00014
- Singh, G., Zürich, E., Gómez-luna, J., Christoph Hagleitner, S., Diamantopoulos, D., Hagleitner, C., et al. (2022). Accelerating weather prediction using near-memory reconfigurable fabric. *ACM Trans. Reconfig. Technol. Syst.* 15:3501804. doi: 10.1145/3501804
- Singha, G., Diamantopoulos, D., Gomez-Lunaa, J., Stuijk, S., Corporaal, H., and Mutlua, O. (2022). “Leaper: fast and accurate FPGA-based system performance prediction via transfer learning,” in *Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 499–508. doi: 10.1109/ICCD56317.2022.00080
- Smith, D., Aguilar, J. A., Allison, P., Beatty, J. J., Bernhoff, H., Besson, D., et al. (2022). “Hardware development for the radio neutrino observatory in Greenland (RNO-G),” in *Proceedings of Science* 395.
- Socha, P., Brejtnik, J., Balasch, J., Novotný, M., and Mentens, N. (2020). Side-channel countermeasures utilizing dynamic logic reconfiguration: Protecting AES/Rijndael and Serpent encryption in hardware. *Microprocess. Microsyst.* 78:103208. doi: 10.1016/j.micpro.2020.103208
- Stamatakis, A. (2014). Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30, 1312–1313. doi: 10.1093/bioinformatics/btu033
- Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.
- Sun, Q., Chen, T., Liu, S., Miao, J., Chen, J., Yu, H., et al. (2021). “Correlated multi-objective multi-fidelity optimization for hls directives design,” in *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, 46–51. doi: 10.23919/DAT51398.2021.9474241
- Theodoropoulos, D., Michel, O., Malakonakis, P., Georgopoulos, K., Isotton, G., Pnevmatikatos, D., et al. (2023). “Optimizing industrial applications for heterogeneous hpc systems: The optima project intermediate stage,” in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. doi: 10.23919/DAT56975.2023.10136980
- Tian, Y., Reukers, M. A., Al-Ars, Z., Hofstee, P., Brobbel, M., Peltenburg, J., et al. (2022). “Tydi-lang: a language for typed streaming hardware,” in *Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*. doi: 10.1145/3624062.3624539
- Umuroglu, Y., Fraser, N. J., Gambardella, G., Blott, M., Leong, P., Jahre, M., et al. (2017). “Finn: a framework for fast, scalable binarized neural network inference,” in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 65–74. doi: 10.1145/3020078.3021744

- Ustun, E., Deng, C., Pal, D., Li, Z., and Zhang, Z. (2020). "Accurate operation delay prediction for FPGA hls using graph neural networks," in *Proceedings of the 39th International Conference on Computer-Aided Design, ICCAD '20* (New York, NY, USA: Association for Computing Machinery). doi: 10.1145/3400302.3415657
- van Cappellen, W. A., Oosterloo, T. A., Verheijen, M. A. W., Adams, E. A. K., Adebahr, B., Braun, R., et al. (2022). Apertif: phased array feeds for the westerbork synthesis radio telescope: System overview and performance characteristics. *Astronomy Astrophys.* 658:A146. doi: 10.1051/0004-6361/202141739
- van der Byl, A., Smith, J., Martens, A., Manley, J., van Balla, T., Rust, A., et al. (2022). MeerKAT correlator-beamformer: a real-time processing back-end for astronomical observations. *J. Astron. Telesc. Instr. Syst.* 8:011006. doi: 10.1117/1.JATIS.8.1.011006
- van Haarlem, M. P., Wise, M. W., Gunst, A. W., Heald, G., McKean, J. P., Hessels, J. W. T., et al. (2013). Lofar: the low-frequency array. *Astronomy Astrophys.* 556:A2. doi: 10.1051/0004-6361/201220873
- van Meter, R., and Horsman, D. (2013). A blueprint for building a quantum computer. *Commun. ACM* 56, 84–93. doi: 10.1145/2494568
- van Wijhe, V., Sprave, V., Passaretti, D., Alachiotis, N., Grutzeck, G., Pionteck, T., et al. (2024). "Exploring the versal ai engines for signal processing in radio astronomy," in *2024 34th International Conference on Field-Programmable Logic and Applications (FPL)*. doi: 10.1109/FPL4840.2024.00011
- Vasile, M., Martoiu, S., Boukadida, N., Antonescu, M., Ulmamei, A., Stoicea, G., et al. (2022). FPGA implementation of RDMA for ATLAS readout with FELIX at high luminosity LHC. *J. Instrument.* 17:C05022. doi: 10.1088/1748-0221/17/05/C05022
- Vasile, M., Martoiu, S., Boukadida, N., Stoicea, G., Micu, P., Dumitru, A., et al. (2023). Integration of FPGA RDMA into the ATLAS readout with FELIX in High Luminosity LHC. *J. Instrument.* 18:C01025. doi: 10.1088/1748-0221/18/01/C01025
- Veenboer, B., and Romein, J. W. (2019). "Radio-astronomical imaging: FPGAs vs gpus," in *Euro-Par 2019: Parallel Processing* ed. R. Yahyapour (Cham: Springer International Publishing), 509–521. doi: 10.1007/978-3-030-29400-7_36
- Viel, F., Gouveia, K. R., Costa, E., Oliveira, M., Boing, M., Mattos, A. M. P. D., et al. (2023). Payload-XL: a platform for the in-orbit validation of the BRAVE FPGA. *IEEE Embed. Syst. Lett.* 15, 93–96. doi: 10.1109/LES.2022.3191638
- Vlagkoulis, V., Sari, A., Antonopoulos, G., Psarakis, M., Tavoularis, A., Furano, G., et al. (2022). Configuration memory scrubbing of SRAM-based FPGAs using a mixed 2-D coding technique. *IEEE Trans. Nucl. Sci.* 69, 871–882. doi: 10.1109/TNS.2022.3151977
- Vlagkoulis, V., Sari, A., Vrachnis, J., Antonopoulos, G., Segkos, N., Psarakis, M., et al. (2021). Single event effects characterization of the programmable logic of xilinx Zynq-7000 FPGA using very/ultra high-energy heavy ions. *IEEE Trans. Nucl. Sci.* 68, 36–45. doi: 10.1109/TNS.2020.3033188
- Voicu, T. A., and Al-Ars, Z. (2019). "SparkJNI: a toolchain for hardware accelerated big data apache spark," in *2019 4th IEEE International Conference on Big Data Analytics, ICBDA 2019*, 152–157. doi: 10.1109/ICBDA.2019.8713201
- Vos, P. D., Kirchhoff, M., and Ziener, D. (2020). "A complete open source design flow for gowin FPGAs," in *2020 International Conference on Field-Programmable Technology (ICFPT)*.
- Wijerathne, D., Li, Z., Pathania, A., Mitra, T., and Thiele, L. (2022). HiMap: fast and scalable high-quality mapping on CGRA via hierarchical abstraction. *IEEE Trans. Comput-Aided Design Integr. Circ. Syst.* 41, 3290–3303. doi: 10.1109/TCAD.2021.3132551
- Wijtvliet, M., Corporaal, H., and Kumar, A. (2021). Cgra-eam-rapid energy and area estimation for coarse-grained reconfigurable architectures. *ACM Trans. Reconfig. Technol. Syst.* 14. doi: 10.1145/3468874
- Wijtvliet, M., Huisken, J., Waeijen, L., and Corporaal, H. (2019). "Blocks: redesigning coarse grained reconfigurable architectures for energy efficiency," in *Proceedings - 29th International Conference on Field-Programmable Logic and Applications, FPL 2019*, 17–23. doi: 10.1109/FPL.2019.00013
- Wood, D. E., Lu, J., and Langmead, B. (2019). Improved metagenomic analysis with kraken 2. *Genome Biol.* 20, 1–13. doi: 10.1186/s13059-019-1891-0
- Xilinx Case Study (2024). *Xilinx powers ALIBABA cloud FAAS with AI acceleration solution for e-commerce business*. Available online at: <https://www.xilinx.com/publications/powerd-by-xilinx/xilinx-alibaba-case-study.pdf>
- Xue, X., and Calabretta, N. (2022). Nanosecond optical switching and control system for data center networks. *Nat. Commun.* 13, 1–8. doi: 10.1038/s41467-022-29913-1
- Xue, X., Patra, B., van Dijk, J. P. G., Samkharadze, N., Subramanian, S., Corna, A., et al. (2021). Cmos-based cryogenic control of silicon quantum circuits. *Nature* 593, 205–210. doi: 10.1038/s41586-021-03469-4
- Yan, F., Xue, X., and Calabretta, N. (2018). HiFOS: a scalable and low-latency hybrid data center network architecture based on flow-controlled fast optical switches. *J. Optical Commun. Network.* 10, B1–B14. doi: 10.1364/JOCN.10.0000B1
- Yang, Z., Ji, S., Chen, X., Zhuang, J., Zhang, W., Jani, D., et al. (2023). "Challenges and opportunities to enable large-scale computing via heterogeneous chiplets," in *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)* (Incheon: IEEE Press), 765–770. doi: 10.1109/ASP-DAC58780.2024.10473961
- Yasudo, R., Coutinho, J., Varbanescu, A., Luk, W., Amano, H., and Becker, T. (2018). "Performance estimation for exascale reconfigurable dataflow platforms," in *Proceedings - 2018 International Conference on Field-Programmable Technology, FPT 2018*, 317–320. doi: 10.1109/FPT.2018.00062
- Yasudo, R., Coutinho, J. G., Varbanescu, A. L., Luk, W., Amano, H., Becker, T., et al. (2021). Analytical performance estimation for large-scale reconfigurable dataflow platforms. *ACM Trans. Reconfig. Technol. Syst.* 14:3452742. doi: 10.1145/3452742
- Zeitouni, S., Vliegen, J., Frassetto, T., Koch, D., Sadeghi, A. R., and Mentens, N. (2021). "Trusted configuration in cloud FPGAs," in *Proceedings - 29th IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM 2021*, 233–241. doi: 10.1109/FCCM51124.2021.00036
- Zhang, J., and Qu, G. (2014). "A survey on security and trust of FPGA-based systems," in *2014 International Conference on Field-Programmable Technology (FPT)* (IEEE), 147–152. doi: 10.1109/FPT.2014.7082768
- Zhang, S., Xue, X., Tangdiongga, E., and Calabretta, N. (2022). Low-latency optical wireless data-center networks using nanoseconds semiconductor-based wavelength selectors and arrayed waveguide grating router. *Photonics* 9:203. doi: 10.3390/photonics9030203
- Zhao, J., Feng, L., Sinha, S., Zhang, W., Liang, Y., and He, B. (2020). Performance modeling and directives optimization for high-level synthesis on FPGA. *IEEE Trans. Comput-Aided Design Integr. Circ. Syst.* 39, 1428–1441. doi: 10.1109/TCAD.2019.2912916
- Ziogas, A. N., Schneider, T., Ben-Nun, T., Calotou, A., De Matteis, T., de Fine Licht, J., et al. (2021). "Productivity, portability, performance: data-centric python," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21* (New York, NY, USA: Association for Computing Machinery). doi: 10.1145/3458817.3476176