HETEROGENEOUS INTEGRATION ROADMAP

2019 Edition

Chapter 13: Co-Design for Heterogeneous Integration

http://eps.ieee.org/hir

We acknowledge with gratitude the use of material and figures in this Roadmap that are excerpted from original sources.
Figures & tables should be re-used only with the permission of the original source.

IEEE ELECTRONICS PACKAGING SOCIETY · ELECTRON DEVICES SOCIETY · IEEE Photonics Society · ASME SETTING THE STANDARD · semi

Table of Contents

To download additional chapters, please visit
http://eps.ieee.org/hir

Chapter 13: Co Design for Heterogeneous Integration

1. Executive Summary

Semiconductor packaging traditionally provides the interconnection scheme between functional layers of systems while also providing protection for dice from mechanical, thermal and electrical stress. Market focus on IoT, health, automotive and communication is driving the quest for more creative integration methods. With Moore's law scaling slowing down, complexity, size variability and the presence of uncertainty have rendered the design task more daunting. Multi-domain (chip, package, board) and multi-physics (thermal, optical, electrical, mechanical) challenges have given rise to the need for co-design tools and methodologies. Interdependence between different domains in the design flow dictates that analysis and design optimization in these domains must be performed concurrently. For heterogeneous integration, this implies the replacement of a segregated approach with one based on synthesis, with a parallel and integrated flow for design, analysis and optimization.

The emergence of new materials and devices with applications in computing, communication, and health care is expanding the stage for heterogeneous integration and creating new opportunities for co-design in which multi-physics, multi-scale and multi-level analyses will provide the needed solutions. New paradigms and new system architectures are being explored to better exploit these emerging technologies.

This chapter focuses on current state-of-the-art, challenges and potential solutions for Co-Design. It compliments and aligns with chapter 14 on Modeling and Simulation, which details the needs for robust modeling and simulation tools in the context of heterogeneous integration. For example, electrical, thermal, and mechanical interactions across the chip-package-board domains can no longer be ignored. New modeling and simulation tools must accurately predict the physical (e.g. electro-thermal, thermo-mechanical, etc) coupling between multiple semiconductor components and the package/system that contains them. Although modeling tools are available for predicting electro-thermal phenomena at a component level (e.g. hot-spots), there is currently no capability to predict these interactions within a co-design optimization environment where different design teams (chip, package, board) collaborate with design/model data that can be shared to support effective trade-off analysis and optimization for a whole system. In addition to this, thermo-mechanical interactions across the chip-package-board domains require robust and accurate modeling tools that can predict stress for use in new physics-of-failure based reliability models. Developments in physics-based models (multi-physics & multi-scale) and simulation tools will be complimented with new applications of AI and Machine Learning to address these challenges. The chapter on modeling & simulation provides further details on these challenges and possible solutions.

Scope

The TWG explored challenges and potential solutions associated with

- Chip-Package-Board Design Flow
- Pathfinding Solutions
- Architecture
- Silicon Photonics
- Neuromorphic Computing
- Quantum Computing

The chapter explores how co-design practices need to be defined in the context of heterogeneous integration. It addresses the traditional chip-package-board design flow as well as current capabilities and future challenges. The vision for co-design is expected to create an environment where design closure is achieved with a minimum number of iterations, meeting all requirements for performance and cost. This environment must leverage from currently available technologies – specifically, computing power, algorithms and artificial intelligence.

2. System Co-Design

Traditionally, integrated circuit (IC) design is performed without consideration of the package and the board. Co-design addresses the discontinuity that exists between IC design and packaging. The objective is to streamline the process of assembling and optimizing the IC, package and printed circuit board (PCB) while applying constraints pertaining to the physical and logical interactions between these design domains.

User-friendly design flows require the creation of signal and power models for the IC, package and PCB which are then combined for system-level simulation. These steps help dictate guidelines with constraints that depend on products and applications.

Historically, packaging technologies evolved around two-dimensional methods. The need for heterogeneous integration has fostered interest in 3D stacking and packaging strategies [1]. These are facilitated by the inception of more advanced interconnection schemes such as TSV, micro-balls or direct Cu-Cu bonding and interposers. Flip chip BGAs operating in the gigahertz range with high densities of I/Os help achieve higher levels of integration while reducing the dependence on Moore's law. Interconnection schemes have also experienced dramatic changes with the demand for higher speed [5], [6]. They have evolved from passive channels made of impedance-controlled traces on top or embedded in a dielectric to complex structures that involve analog and mixed-signal components performing equalization and clock/data recovery. To achieve required performance, planning in all three domains of integration is of paramount importance.

Packaging tradeoffs must be identified and evaluated in advance during system design. In addition, early planning is critical for a multi-domain design flow. Aligning to a good co-design space before layout and signoff is important. Floorplans become very demanding at early stages of the design flow and it becomes prohibitively difficult to conform them with board and package flows without expensive tradeoffs as the design progresses. Early floorplans along with early bump and ball maps can facilitate easy evaluations across metrics in all three domains.

In chip design, the package and board model is used as a load. In package design, the load is the chip-level I/O buffer model or the board model. Conversely, from the board, the loads are the package I/O buffer models. One option is to use the package as the "host" or "master" domain whose task is to operate as an intermediary between the IC and the PCB. As such the packaging gains influence on the choice of PCB technology and can play a role in optimizing floor-planning and bump placement on the IC die. Since some packaging technologies can be far more expensive than the integrated circuits that they house, their physical design must be cost-conscious so as not to force the redesign of an interposer or substrate layout. Electrical budget decisions such as noise margins are decided prior to physical design and are monitored during package implementation. Auto-routers can help facilitate the task, but they need to closely cooperate with simulation tools.

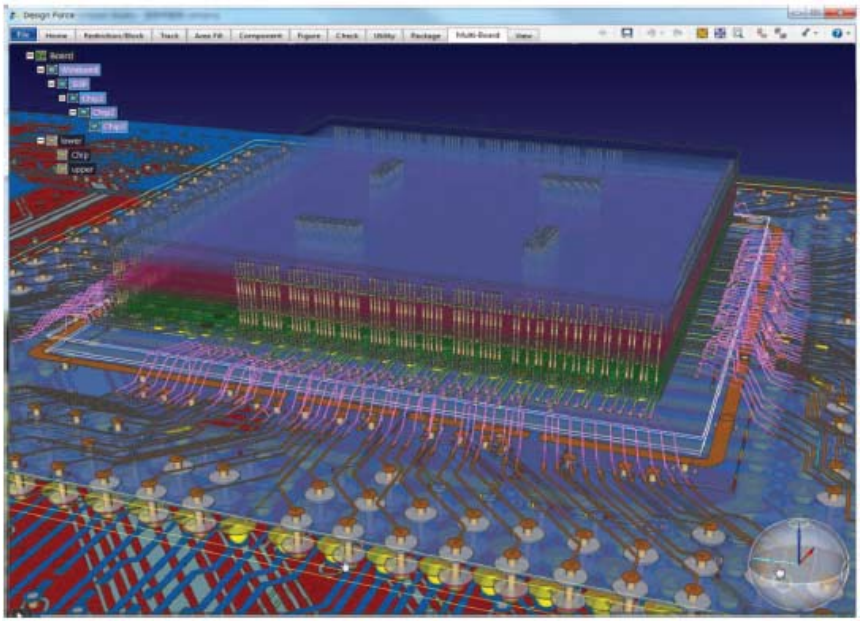

Figure 1. Domains in Co-Design Environment (Source [1]).

The ideal co-design expert system must possess the ability to simultaneously oversee the floorplan of the IC, the package, and the PCB while having the methodology for translating information between the domains and intelligently changing the design at any step within the design cycle. In addition, when bump pad pitches on the package are modified, visualization should permit capturing resulting changes on the underlying IC. Co-design optimization is achieved through floorplan visualization with the ability to track physical scaling and offset relationships between the various domains.

Traditional library components that manage connection pins and physical outline for boards and packages are too simplistic to operate in a co-design scenario. Because of the complexity and the level of layout detail that is required, the co-design vehicle should include details on trace routing, via location and internal active and passive components.

Co-design can ideally be used to optimize fan-out from I/O buffers, minimize the layers of PCB for signal traces and optimize interposer layer count. For this to happen, the noise budgets for chips and packages must be made available. For instance, for low-voltage differential signaling, trace bumps must remain within close proximity and may need additional layers to prevent crosstalk. To facilitate power distribution analysis and I/O planning, co-design tools must offer the capability to extract interconnect parameters from the buffers to the package pins.

One critical issue is to manage the connectivity between the various levels. Intra-level and inter-level connectivity must be managed concurrently. The co-design approach must facilitate pin mapping, wire-bond placement and other logical connectivity constraints in a seamless manner. Electromagnetic compatibility concerns at the package and board levels create constraints on I/O routing, power distribution networks and even location of sensors, antennas and reactive elements. This information is to be used for accounting for obstructions, RDL routing and I/O placement. In a co-design environment this IC-level information must be shared with the package and board levels.

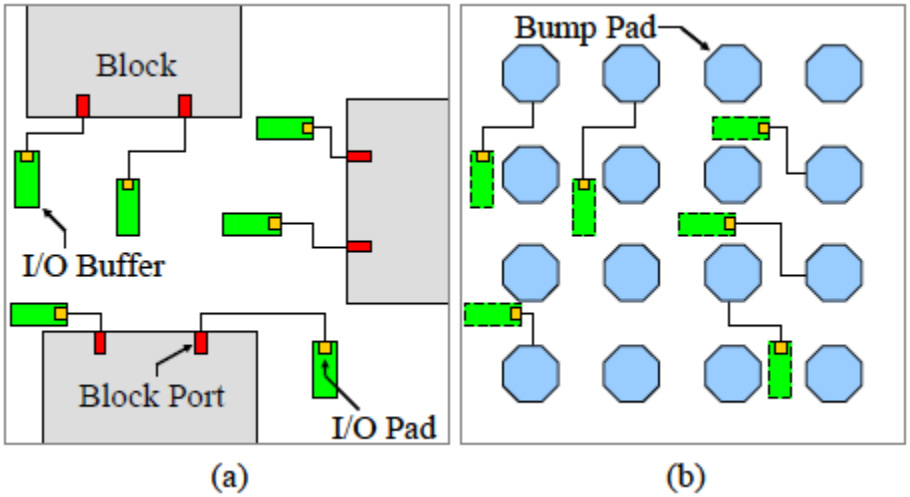

Figure 2. (a) Routing from block ports to I/O pads (b) Package level – routing from I/O pads to bump pads (source [2])

When changes are made in a domain, the information must be propagated to the other domains. Typical changes include net name changes, information about addition or deletion of signals, and scaling. Changes to the physical detail may include addition or deletion of pins, modification of pin pitch, or re-locations and re-orientations. In addition, accurate visualization of the hierarchy across all three domains can help achieve optimum connectivity and pin placement and the ability to manage cross-domain connectivity.

Database types and data formats used for package and boards are different from the IC level. To manage connectivity, traditional design flows rely on drawings and spreadsheets with labeling and names for the different domains. In particular, at the package level, spreadsheets and tables are the preferred database format because present-day CAD tools do not support the various flows that are in existence. At the board level, graphical schematics are more commonly used for connectivity management. With increasing design complexity, this method becomes intractable, making it difficult to track changes and capture co-dependencies that exist between the domains. In addition, cost-related constraints make it difficult to rely on spreadsheets and map models which are no longer sufficient for a true co-design environment. For every co-design approach, concurrent database support is needed.

Placement and routing algorithms are at the heart of co-design strategy and have seen increasing challenges due to increasing complexity in integration. These algorithms have the objective of determining the optimum layout while minimizing real estate and delay, improving signal integrity, and reducing cost. They also must possess computational efficiency. With the increasing number of I/Os and added constraints resulting from the multi-domain and multi-physics nature of co-design, balancing the goals of performance and routability while achieving optimal execution in placement is the nature of the challenge. More specifically, it is a bottleneck to concurrently optimize pin assignment and pin routing simultaneously in the different design domains (chip, package, board). At a premium are fast escape routing algorithms that handle bump assignment, RDL routing and substrate routing with accuracy [4].

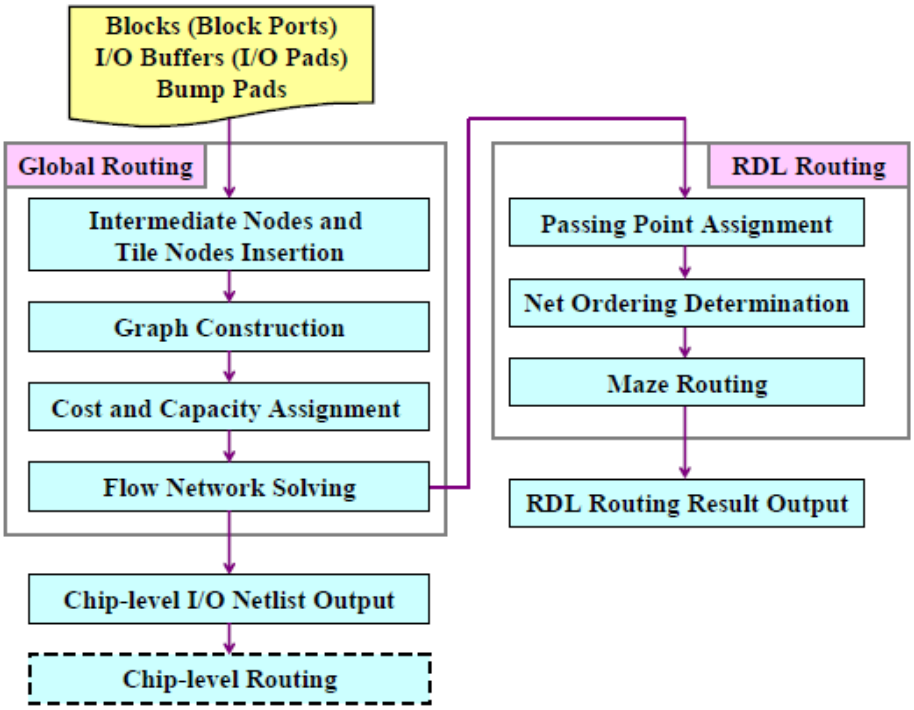

Figure 3. The Routing Flow of Chip-Package Co-design. (Source [3])

Verification is another component of a design flow that must be re-conceived for a co-design adaptation. One key step in the co-analysis of chip, package, and board systems is the reduction of complexity. At the verification stage, this is often achieved through macro-modeling and behavioral modeling. It is also necessary for the frequency dependence of interconnects to be taken into account. Macro-modeling is often used for circuit blocks for which the behavior is linear. PDN analysis often requires one to simulate systems for which equivalent circuits are not available [8]. From this, a macro-model can be generated to produce a SPICE (Simulated Program with Integrated Circuit Emphasis) stamp which reproduces the response of the system with sufficient accuracy. As an alternative, circuit netlists can be derived from macro-models for use with SPICE [10].

The complexity of integrated circuits makes transistor-level simulations intractable, resulting in a shift to behavioral modeling, which not only provides a simpler and faster-to-simulate representation of the IC but also protects the intellectual property (IP). Models from the input/output buffer information specification (IBIS) aim to achieve that goal and are a standard in the electronics industry, also known as ANSI/EIA-656A [7]. Other techniques involving behavioral models for nonlinear systems, such as the poly-harmonic distortion (PHD) method, are poised to provide viable solutions [11].

Given the complexity in current IC packaging technologies, choosing the appropriate package configuration for a product using the cost, performance and reliability criteria can be a daunting task. Such a decision must be made at an early stage. Product definition in the early phases of implementation requires an expert system that can quickly predict and combine a complex assembly of components from a limited data set. Pathfinding often operates with incomplete and inaccurate inputs while providing viable solutions. It requires hundreds if not thousands of man-hours to develop a viable co-design solution.

While optimizing I/O across all the domains, simulation needs to be an integral part of the process. Pathfinding makes it possible to run some feasibility studies of different packaging options and to drive IC placement and routing with packaging constraints. This effort must bring electrical constraints and routability into packaging solutions.

It is not sufficient to transfer the design to the next team; additional collaboration is a more viable path to a successful product. In addition, useful tools can only be developed with interaction between developer and user. It is imperative that chip and package designers work together to assess early-on the system-wide impact of various packaging scenarios. Design teams need to bridge the process between pathfinding and production design. Experience has shown that tools must and will mature with customers. It is a very beneficial model, because software can be re-configured and updated continuously depending on the needs and specific applications. New methodologies need to be introduced and can only succeed with support from the major EDA players.

Not addressed in this survey is the need for true hierarchical schematics across the IC/Package/Board domain. This would entail compiling IC details required for co-design/co-optimization and represent them as one or two layers in order to integrate them with package and board-level schematics. Also not covered is the impact of higher performance (multiple voltages, higher signaling speeds, etc.), integration of multiple areas of analysis, scaling (bump pitch, BGA pitch, board level parameters, etc.) to co-design methods and tool flows. The impact of these trends on co-design requirements and the ensuing challenges will be reported in the next edition of the HIR roadmap.

3. Emerging Pathfinding Methodologies

In the conventional system design flow, where dice are designed and optimized individually and handed over to the package designer, pathfinding refers to finding the solution for package connectivity that results in optimal system interconnect performance. However, in the context of co-design for heterogeneous integration, pathfinding needs to ensure ICs, packages and PCBs are co-optimized, taking into account multiple physical domains and multiple system configurations [5].

State-of-the-art EDA tools (such as Mentor Graphics Xpedition Substrate Integrator [12] and Cadence OrbitIO Interconnect Designer [13]) have successfully tackled some of the key Pathfinding features required for enabling 3D Chip-Package-PCB co-design [5]:

- Unified workflow, including partitioning, floorplanning, design of system-level interconnects, route pathway exploration and feasibility analysis
- Capability to create abstract package models and virtual die models from multiple sources
- Ability to visualize and modify component placement scenarios and make connectivity changes in a preliminary floorplan
- Provision of dynamic manipulation of pin arrays within the abstract models
- Preserving signal assignments and rules while making adjustments to the physical pin array
- Support for multiple package variables and PCB form factors to verify and compare different system configurations
- Standardization of interface data of system blocks, usage of open-source file formats and APKs
- Enabling the interaction of design tools from different EDA vendors

Some of the difficult challenges for heterogeneous integration co-design include:

- **Routability estimation and optimal routing algorithms** are facing nets that may span across multiple die, with heterogeneous topologies. Academic efforts towards extraction of inter-die coupling [15] and power delivery pathfinding [16] have been proposed, but given the complexity of the problem, novel analytic and heuristic estimation algorithms will need to be integrated in the EDA tools . An up-to-date overview of open challenges in **multi-domain simulation and verification of 3D chip stacks** is given in Section 4.3 of [18].
- While each element of the system might be fully optimized using the conventional design flows, the overall system is unlikely to reach the same degree of optimization [17]. There is a clear need for **co-optimization of the global system**, using the co-design flow which allows parallel design of package and die, with adequate level of abstraction to support iterative modifications to subsystems, ideally performed in an automated manner (e.g. using machine learning [33] or other suitable algorithms).
- Co-design CAD tools need the **ability to quickly provide accurate models of physical structures and interconnects for electrical and thermal analysis**. Due to strong coupling of physical domains in a 3D stack, such modeling and simulation techniques need to be scalable, linking across different design levels. Recent academic work in the field [20] proposed transaction-level-based pathfinding, complementing previous RTL-based approaches, while commercially available EDA tools are leveraging standalone SI/PI, computational fluid dynamics thermal modeling and substrate fabrication checking tools [14].
- In particular, due to the high density of tightly packed 3D systems, **the need for thermal pathfinding and co-optimization** needs to be addressed concurrently with electrical routing. Each component of a 3D heterogeneous system can be thermally optimized on its own (locally) to manage hot spots, junction temperature and power density. However, the integrated system (global) thermal management may be drastically different, due to stacking of high-performance chips in miniature packages resulting in increased heat flux, reduced cooling channel space and extreme conditions for hot spots in thin chips

[24]. A unified co-optimization flow of the global system which takes into account thermal properties [25], [26], [32] will be key to successful 3D integration of high-performance systems. See chapter 20 on Thermal Design.

- With the increasing complexity of today's chips, packages and boards, designing each in isolation is no longer feasible. For enabling heterogeneous integration, there is a clear need for **true hierarchical schematics across the IC/Package/Board domain**. Available co-design EDA tools [27], [28] have enabled a single hierarchical schematic to drive both IC and package layout while providing LVS checking, along with automating the library development process. However, to enable efficient co-optimization, the IC details need to be abstracted and integrated with package and board-level schematics [29], [30], [31].
- In order to support increasing performance and miniaturization, individual system building blocks are **growing in complexity** (multi-supply/multi-voltage techniques, higher signaling speeds and encoding techniques) and **reducing in scale** (bump pitch, BGA pitch, board level parameters). While current tool flows are equipped to tackle the problems for individual building blocks [34], [35], the tools will need to be tightly integrated into a coherent co-design methodology.
- Integration of heterogeneous devices into a 3D system necessitates a **multi-physics aware pathfinding environment** that considers the impact of material characteristics (e.g. CTE, Poisson ratio, Young's modulus, insertions loss). The environment needs to encompass electrical, mechanical, magnetic, optical, acoustic, and fluidic models of system building blocks, with the goal of enabling a holistic system-level architecture and interconnect optimization. While some systems lend themselves naturally to being packaged away into models with only electrical ports (e.g. MEMS processes), other integration cases (photonics, micro-fluidics) require non-electrical interfacing to the 3D system [18], making it difficult to capture in a pure EDA tool, and workarounds/customizations are needed to approximate it.
- **Design-for-Test and testability constraints for 3D integration** present a unique challenge to the pathfinding methodology, given the testing modality of individual system building blocks: IC bumps may have to be probed with power and routing topologies very different than in the final system. Pathfinding flow needs to enable IC/package bump layout that satisfies both sets of design constraints, similar to designing towards multiple PCB target systems in the conventional EDA flow [5]. In addition to the goal of enabling die testing that results in a high unit-to-die ratio, the interposer itself needs to be designed for testability, since it will be assembled to valuable known good die [18]. See chapter 17 on Test.
- Especially in very high complexity designs, **architectural considerations, such as redundancy, built-in self-test, (self) repair capabilities as well as failure monitoring functions** need to be considered during Pathfinding. They contribute to achieving acceptable yields and low failure rates. These measures result in cost-effective operation and long-term reliability. Co-design tools and algorithms need to support designing with redundancy and repair capabilities in mind, following the DfT design flows proposed for 3D memory chip design [22] and large-scale 3D integration [23].

4. System Architecture Considerations

VLSI design methodology flows by transferring specifications, constraints and details from one design layer to the next with virtually strict boundaries between the layers. In order to ensure design robustness, guard-bands are included when passing down the information, which inevitably causes over-design. With decreasing design margins, many designs can ill afford such guard-bands under required bill-of-materials (BOM) cost and time-to-market constraints. Thus, it is highly desirable to explore methodologies that enable more flexible optimization. Architecture co-design is just such an approach that increases the optimization space across various design layers by removing the boundaries between layers.

Co-design for architecture involves preceding and back-end design layers. For the design layers at the top, it is more associated with system or software co-design, which includes:

- **Algorithm and architecture co-design** typically utilizes the hardware architecture feature to favor more efficient data access or processing. It is also possible to modify architecture details, such as deeper cache hierarchy, to improve runtime and efficiency of the algorithm.

- **Operating system (OS) and architecture co-design** provides more direct access to either OS services or the underlying cores, thereby creating a convenient programming environment at the cost of robustness.
- **Security and architecture co-design** can eliminate inherent software vulnerabilities by optimizing the underlying architecture (as well as OS) to enforce software security policies and semantics.

On the other hand, the architecture can also be co-designed using the back-end details:

- **RTL and architecture co-design** typically combines or embeds the accurate RTL simulation into fast architecture simulation to provides more direct but accurate evaluations of the underlying core.
- **Interconnect and architecture co-design** accounts for the impact of interconnect parasitics, which is more evident at advanced nodes, but relies on a segment of interconnect layout to provide insight into the system performance at post-layout.
- **Device and architecture co-design** occurs more often for simpler architectures, such as SRAM. The unique characteristics of devices can then be directly reflected in the architecture design and functions as the motivation for architecture optimization.
- **Technology and architecture co-design** makes use of the underlying architecture features to determine the design rules and options for the technologies of interest.

However, it is not a trivial task to conduct architecture co-design. Several key limitations or challenges faced by architecture co-design will:

- Improve co-simulation efficiency due to the limitation that different design layers may employ tools with different complexities, causing speed differences of orders of magnitude. When all the design layers are connected and simulated altogether, a slower simulator for a layer may block the faster ones from completing.
- Provide communication mechanisms between different design layers. It is necessary to identify the minimum key information that needs to be interchanged to prevent unnecessary interruption.
- Support growing functionalities and heterogeneity requirements driven by the emerging devices.
- Support thermal, power integrity, and reliability management at advanced technology nodes.
- Support asynchronous accelerator or co-process design.
- Reduce time-to-market through co-design which demands details and constraints for thermal, electrical and mechanical parts.
- Drive down cost at the back-end stage to optimize design margins for layout.
- Provide models of available/proven system building blocks – at different levels of abstraction – to reduce system design time and risk of failure.

There have been industry-wide efforts in architecture design to include more functions, more materials, and highly energy-efficient co-processors [36]-[42]. Various innovations in architecture design for power and performance are highly desired to provide both the desired performance and power dissipation. The co-design methodologies account for information from other design layers to raise the potential optimization space. A trade-off between or among different layers is the goal of architecture co-design.

5. Co-design in Silicon-Photonics

The co-design of silicon-photonic systems involves multiple domains – electrical, optical, thermal and mechanical. The behavior of the system in these four domains needs to be carefully accounted for during the design process. The co-design process typically involves schematic capture, circuit simulation, circuit layout and design verification [43]. For schematic capture, various commercial and non-commercial schematic editors exist. Both electrical and photonic components of the system can be composed hierarchically by breaking the components into sub-circuits. There are four approaches for simulating silicon-photonic systems –

- we can use a photonic circuit simulator to design photonics and electronics together;
- we can use an electronic circuit simulator to design photonics and electronics together;
- we can partition the design into its electronic and photonic components and simulate the components using separate electronic and photonic circuit simulators; and
- we can co-simulate electrical and photonic components using separate electronic and photonic circuit simulators [44].

Given the extensive design tools and design automation flows that are currently available for electronic systems, the second approach above can enable us to simulate large silicon-photonic systems. For circuit layout, one can use a schematic-driven layout (SDL) approach. Depending on the hierarchy and sub-circuits in the silicon-photonic system, the SDL approach can pre-place individual components of the system, and then the designer can connect the components. However, connection of the photonic components is more challenging than the electrical components because photonic components do not follow the Manhattan-style routing rules and typically use a single routing layer. For verification, the design rule checker (DRC) and layout versus schematic (LVS) verification steps can be used for verifying both electrical and photonic components. Overall, a design environment that can accurately account for cross-domain effects is critical for designing optimal silicon-photonic systems.

The electrical and optical components have different timescales. Electrical components typically run sub-5 GHz. Optical components have a broader range, and wavelength division multiplexed (WDM) optical signals can be spread over several THz (even 10THz) of bandwidth. Thermal time constants are of the order tens of microseconds. However, the timescales for heating and cooling of electrical and photonic components are not always the same.

When co-designing silicon-photonic systems, we need to carefully account for the interaction between electrical, optical and thermal domains. For example, photonic circuits are very sensitive to thermal changes – so when using electrical and photonic integrated circuits, they need to be designed and placed strategically such that heat generated due to power consumption in the electrical circuits does not impact the functionality of the photonic circuits. A ring modulator has optical power limitations, along with dynamic self-heating due to both two-photon absorption (TPA) and linear absorption, and provides photocurrent because of defect absorption. This photocurrent can be used for feedback-based control of the silicon-photonic system.

6. Co-Design for Future Heterogenous Systems

System design and heterogeneous integration are key enablers for overcoming the challenges arising from the increasing complexity of embedded systems. In this section, we cover the design challenges for future heterogeneous systems based on neuromorphic and quantum computing.

Co-Design for Neuromorphic Computing

The architecture of digital microprocessors is fundamentally different from the biological brain. The brain consumes about 20 W and is a massively parallel structure of neurons interconnected through synapses, whereas microprocessors are mostly based on a von Neumann architecture. Inspired by biological functioning of the brain, neuromorphic architectures do not adopt von Neumann architecture by collocating memory close to the processor but structured to provide massive parallelism, high energy efficiency, reconfigurability, fault tolerance and integrability with CMOS technologies. Neuromorphic computing spans a broad range of scientific disciplines, from material science to devices, computer science to neuroscience, which are required to design an energy-efficient neuromorphic computing system.

TABLE 1. STATE OF THE ART ON NEUROMORPHIC HARDWARE

Company/ Project	Technology	Energy Consumption	Learning and Autonomy	Integration density
IBM/ TrueNorth	Digital ASIC at 28nm CMOS	High, 26pJ per synaptic operation	No	1M neurons, 256M synapses
Intel/ Loihi	Image recognition, control of robots, etc.	Claims up to 1000 x higher efficiency than GPUs; 23.6pJ per synaptic operation	Yes	130k neurons, 130M synapses
Zhejiang & Hangzhou Dianzi U. (China)/Darwin	Digital ASIC at 180nm CMOS	Unknown, probably high	No	Max 32k neurons with 1000M synapses
Human Brain Project (EU)/ BrainScaleS	Mixed-signal wafer-scale ASIC at 180nm CMOS	Claims 1000 x higher efficiency than traditional chips	No	180k neurons, 40M synapses
Human Brain Project (EU)/SpiNNaker	ARM boards at 130nm CMOS	High	No	1K neurons/core, 1M cores
DARPA & HRL Labs (US)/ SyNAPSE	Mixed-signal 180nm CMOS	17pJ synaptic operation	Yes	1024 neurons, 64k synapses

Current CMOS-based devices and emerging devices (e.g., memristor, spintronics, magnetic, etc. [46-53]) are exploited to emulate the functionality of neurons and synapses. See chapter 16 on emerging research devices. Just as biological neural systems are composed of networks of neurons and synapses that learn and evolve, so must the computational building blocks of neuromorphic computing systems learn and evolve to address the problems presented to them. Neuromorphic hardware platforms can emulate both the ultra-high density and the ultra-low power consumption of biological neural networks. Non-volatile memories (e.g., memristors) are one of the key enablers for biologically inspired computing [46].

Neuromorphic computing has been mainly explored based on neural networks. There are two main classes – (1) Artificial Neural Networks (ANNs) and (2) Spiking Neural Networks (SNNs) – that are being developed in either digital or analog form and implemented typically in CMOS technology. Some of the notable hardware implementations have been listed in Table 1 [54].

Co-Design Challenges

Neuromorphic design challenges arise from different domains (digital, analog, mixed-signal) to implementing neuromorphic hardware. The interface between different domains introduces design challenges, and efforts are dedicated to increasing the accuracy of classification and computing with fewer bits and with lower precision. As a path to improve computing efficiency (and saving power), near-memory processing and computing (i.e., in-memory processing) are crucial. The design implementation challenges for neuromorphic computing are mainly related to SNR (signal to noise ratio), the variability of analog design, and mixed-signal design issues.

To build large-scale neuromorphic systems, there are several design-specific challenges. One challenge arises from the device type for emulating neurons and synapses, such as chemical vs solid-state implementation. Using chemical-based devices will introduce design challenges about how these devices can be scalable, controllable and reconfigurable at the system-level. Another challenge arises for cross-domain design needs (analog, digital, mixed-signal) which raises challenges on system-level design, test and reliability. Moreover, a challenge arises from the need for dense packaging of neurons to achieve volume comparable to biological brains. This might imply dense 3D technology packaging, which challenges package assembly, power delivery, heat removal, I/O count, and system level design (place & route, interconnects, power/ground/clock signals). Here, we elaborate on these challenges in detail:

- **Cross-Layer Design (Material, Device, Circuits, Systems)** – Neuromorphic computing involves research competencies across layers from materials, devices, circuits, systems and packaging. Research is needed to bridge the gap between materials research, device design and system-level performance requirements. Current non-volatile memories (NVM) are limited in being able to take full advantage of a neuromorphic computing paradigm. Thus, challenges arise from cross-layer design to consider simultaneously behavior of novel devices and materials (e.g. switching properties of non-volatile devices) with system design to meet the requirements of neural networks (e.g. training and inference, symmetric up and down conductance changes in NVM devices, device-to-device variability, high bandwidth, noise sensitivity of analog elements, weight updates, area, power and scalability of design). In parallel to mainstream NVMs, new devices based on electro-chemical synaptic elements (ECRAM) [57] are also being investigated as novel devices for enabling AI neuromorphic computing.
- **Multiple Domain Co-Design (Analog, Digital, Mixed-Signal)** – At the core of neuromorphic computing there are matrix manipulations done on arrays of non-volatile memories in the analog domain. These computations are performed locally in memory to avoid moving back and forth the weights from memory to the computing unit. However, not all the operations can be performed in the analog array; therefore, neuromorphic systems will need to embed the analog array into a digital backbone [58]. This will also raise challenges on the analog/digital interface such as insertion of ADCs and DACs, which might further raise concerns on noise, timing variations and power consumption issues. Thus, the focus will be given to explore both mixed digital-analog and pure analog approaches. Moreover, NVM devices should have symmetric conductance variations, and investigation should be carried on utilizing novel material NVMs such as phase-change materials (PCM). Additional design challenges arise from multi-physics simulations of neuromorphic devices to circuits (analog and digital), power/timing sign-off, power/signal/thermal integrity across domains, reliability (i.e. system-level connectivity and latency) and power delivery challenges.

- **Packaging** – Neuromorphic computing requires massive parallelism and significant data movement. One of the challenges arises in the system design and package integration. Cost-effective 3D solutions (3D stacking and monolithic integration) packaging technology should be explored to enable massive parallelism and increase in energy efficiency.

Co-Design for Quantum Computing

Quantum computing has become a very active and promising topic in recent years. Quantum computers hold the promise of solving certain set of complex problems that are intractable for even the most powerful current supercomputers. Those problems include integer factorization, molecule simulation, search and optimization that have application in fields such as cryptography, chemistry, pharmacy, medicine, artificial intelligence and machine learning [57, 58].

The idea of building a quantum computer was introduced by the physicist Richard Feynman [59] in the early 80's. He proposed building a quantum machine to simulate quantum systems. Since then, many quantum algorithms have been developed, one of the most famous being the Shor's algorithm [60] for factoring large numbers that has theoretically proven to have exponential speedup compared to its best classical counterpart. Another example is Grover's algorithm [61] for solving unstructured search problems with quadratic speedup. In addition, different quantum technologies are being implemented.

With the development of quantum algorithms and advances in quantum hardware, main IT companies (Intel, Google, IBM, Alibaba) and research groups are working on building the first universal quantum computer. This requires contributions from several fields of knowledge, including Physics, Mathematics, Computer Science, and Electrical and Computer Engineering.

Background and overview

The basic unit of information in quantum computing is called quantum bit or qubit. A qubit can be in any of its basis states |0> or |1> but also in a superposition of both. This is mathematically described by $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers and satisfy $|\alpha|^2+|\beta|^2=1$. When a qubit is measured, it only provides a binary value '0' or '1'(measurement result) with probability $|\alpha|^2$ and $|\beta|^2$, respectively. In addition, the quantum state is projected onto the corresponding |0> or |1> state. Due to this probabilistic behavior of the measurement, quantum computation is non-deterministic and an algorithm needs to be run several times and averaged to get the correct result. By combining qubits and exploiting superposition and entanglement, quantum computers can be faster than classical computers. In order to perform quantum computation, the states of the qubits have to be changed. This can be done by applying quantum gates (or operations) when the circuit model of computation is adopted. An alternative approach is adiabatic quantum computing [62].

The main issue with qubits is their fragility. They easily decohere – that is, loose their information extremely fast due just to interaction with the environment. In addition, quantum gates are faulty, having error rates ~10^{-2}-10^{-3}. In order to make quantum computing fault-tolerant (FT), quantum error correction (QEC) and fault-tolerant mechanisms are required [63, 64]. To this purpose, a logical qubit is encoded into multiple unreliable physical qubits and FT operations are applied on them. In addition, the quantum system is continuously monitored to detect and correct for possible errors. The use of QEC significantly increases the number of qubits required (up to four orders of magnitude) and imposes some extra requirements on the system design. One of the most promising QEC codes is surface code.

As will be explained in the next section, there are different ways of implementing qubits. Current quantum processors consist of tens of 'noisy' qubits. Although quantum processors have been able to demonstrate small quantum algorithms [65] and quantum error detection [66, 67, 68], quantum advantage and supremacy[1] still need to be proven.

The main challenges in quantum technologies are: i) improving qubit lifetime and gate fidelity and ii) scalability, to build large-scale FT quantum systems. We are now entering the Noisy Intermediate-Scale Quantum (NISQ) era [69] that refers to quantum processors with 50 to a few hundreds of qubits and with imperfect control over them.

Qubit implementations

There are different quantum technologies that are being developed which differ in their implementation and properties. The most relevant ones are superconducting qubits, ion traps, silicon-based qubits, topological qubits,

[1] Quantum advantage refers to the potential of quantum processors to solve problems faster than a classical computer. Quantum supremacy refers the moment when a quantum computer will solve problems that are unsolvable by classical computers.

and photonic qubits. The main properties and challenges for each of these technologies, such as number of qubits, qubit lifetime, gate fidelity, gate time, connectivity and scalability, are described in [70,71,72]. It is still unclear which technology will succeed, but nowadays superconducting and ion-trap qubits seem to be the most promising candidates as they have achieved the larger number of qubits. On the other hand, silicon quantum dots might be easier to scale up (smaller components), photons can be operated at room temperature, and topological qubits might have long lifetime and gate fidelities making it easier to achieve FT computation.

Current co-design challenges

As quantum computation is a relatively new field, there is no clear design process when building a quantum system. The first 'quantum computer' prototypes are being developed and some of them are accessible through the Internet (quantum in the cloud) [73, 74]. The main challenges are related to build a scalable fault-tolerant quantum system and can be divided as:

- **Quantum hardware**: For building a large-scale quantum computer, quantum hardware must meet a set of requirements known as DiVincenzo criteria [75]. Although some of the qubit implementations such as ion-traps seem to already fulfil them, in general, lower error rates still need to be achieved by enhancing coherence time and gate fidelity. Most of the technologies are also facing scalability issues – that is, to integrate a larger number of qubits and quantum error correction mechanisms [76, 77]. Some solutions include using crossbars of qubits and even connecting small arrays of quits using long-distance qubit couplers (silicon spin qubits) [78, 79]. Others propose to define unit cells that can be replicated and vertical I/O interconnects using through-silicon vias or flip-chip bonding (for superconducting qubits) [80] or build micro-fabricated traps and photonic interconnects or using modules (for ion traps) [81, 82, 83].
- **Control electronics**: General purpose instruments, such as arbitrary waveform generators (AWGs) and digitizers, are used to operate and control the qubits, which would be unfeasible for large systems because of their size and cost. In addition, some quantum technologies such as solid-state qubits need to be placed at cryogenic temperatures, posing a connectivity challenge between the qubits and the control electronics that are at room temperature. Proposed solutions include multiplexing, moving classical control electronics closer to the quantum chip (cryogenic CMOS electronics) or even moving qubits to higher temperatures (hot silicon quantum dot qubits).[79, 84, 85, 86, 87]
- **System design and integration (SW-HW co-design)**: a computing system is composed of software (SW) and hardware (HW) layers. To bridge quantum applications with quantum devices, quantum SW as well as HW layers that serve as an interface between SW and quantum processors need to be developed. Software platforms have been built to connect to different quantum processors and/or quantum computer simulators [88]. However, as quantum technology still needs to mature and scale up, there are limited works that explore the entire system design [89, 90, 91, 92]. High-level quantum programming languages and compilers already exist to easily describe quantum algorithms and translate them into low-level quantum instructions, most of them expressed in a quantum assembly language (QASM) [88]. Furthermore, the first quantum instruction set architectures and corresponding microarchitecture have been proposed that target small quantum processors [93, 94, 95, 96]. The main challenge here is to add 'programmability' to current quantum processors and exploring the design trade-off between hardware and software as the number of qubits grows. In addition, although the term quantum computer is extensively used, what we will have in the next coming years is a quantum co-processor or accelerator in which some parts of the computation will be offloaded (hybrid classical-quantum computing paradigm) [97, 98, 99, 100].

Acknowledgments

The Co-Design Technical Working Group (TWG) appreciates very much the contribution of volunteers and their companies/institutions in making this chapter a reality for our profession and our industry:

José Schutt-Aine, chair; Christopher Bailey, co-chair

Pavle Milosevic	Aida Todri-Sanial	Carmen G. Almudever
Cheng Zhuo	Ajay Joshi	Milos Popovic
Ambrish Varma	Herb Reiter	Narayanan Terizhandur Varadharajan
Richard Rao	Brandon Wang	Youngsoo Lee
Muhannad S. Bakir	Vaishnav Srinivas	Kambiz Samadi

References

[1] Kiyohisa Hasegawa, Kazunari Koga, Humair Mandavia, "Improving design performance with system-level co-design and multi-physics analysis: Thermal and stress-aware design for three dimensional stacked IC package", 2016 Pan Pacific Microelectronics Symposium (Pan Pacific), March 2016.

[2] Meng-Ling Chen, Tu-Hsiung Tsai, Hung-Ming Chen Shi-Hao Chen, "Routability-Driven Bump Assignment for Chip-Package Co-Design", 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC), pp 519-524, Jan 2014.

[3] Jia-Wei Fang and Yao-Wen Chang, "Area-I/O Flip-Chip Routing for Chip-Package Co-Design", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp 711-721, Volume 29, Issue 5, May 2010.

[4] L. Luo, T. Yan, Q. Ma, M. D. F. Wong, and T. Shibuya, "B-escape: a simultaneous escape routing algorithm based on boundary routing," in Proc. of International Symposium on Physical Design, pp. 19-25, 2010.

[5] G. Brist et al., "A novel approach to IC, package and board co-optimization," in Proc. Sixteenth Int. Symp. Quality Electronic Design, Mar. 2015, pp. 512–518.]

[6] John Park, "The bifurcation of advanced IC packaging", 2019 IMAPS Device Packaging Conference, AZ

[7] IBIS Homepage, [Online]. Available: http://www.eigroup.org/ibis/

[8] D. Saraswat, R. Achar, and M. Nakhla, "A fast algorithm and practical considerations for passive macromodeling of measured/simulated data," IEEE Trans. Adv. Packag., vol. 27, no. 1, pp. 57–70, Feb. 2004.

[9] Chua, L. IEEE Trans. Circuit Theory 18, 507–519 (1971). Chua, L. IEEE Trans. Circuit Theory 18, 507–519 (1971).

[10] Jose Schutt-Aine, Robust SPICE circuit generation using scattering parameters", Proceedings of the 2014 IEEE International Symposium on Electromagnetic Compatibility (EMC), pp. 230-233, Rayleigh, NC August 2014.

[11] D. E. Root, J. Verspecht, D. Sharrit, J. Wood, and A. Cognata, "Broadband polyharmonic distortion (PHD) behavioral models from fast automated simulations and large-signal vectorial network measurements," IEEE Trans. Microwave Theory Tech., vol. 53, no. 11, pp. 3656–3664, Nov. 2005.

[12] http://s3.mentor.com/public_documents/datasheet/pcb/xpediton-substrate-integrator-ds.pdf

[13] https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/ic-package-design-analysis/sigrity-orbitio-ds.pdf

[14] https://www.mentor.com/company/news/mentor-xpedition-integrator-flow-ic-pack-board-design

[15] Y. Peng, D. Petranovic, K. Samadi, P. Kamal, Y. Du, S. K. Lim, "Inter-die Coupling Extraction and Physical Design Optimization for Face-to-Face 3D ICs", IEEE Trans. on NANO, 2017.

[16] S. Kim, A. B. Kahng, S. Kang and B. Xu. "Power Delivery Pathfinding for Emerging Die-to-Wafer Integration Technology", in Proc. DATE, 2019

[17] T. Horst, R. Fischbach, J. Lienig, "Design methodologies and co-design options for novel 3D technologies", Proc. of ANALOG 2018, pp. 181-186, Sep. 2018.

[18] Johann Knechtel, Ozgur Sinanoglu, Ibrahim (Abe) M. Elfadel, Jens Lienig, Cliff C. N. Sze, "Large-Scale 3D Chips: Challenges and Solutions for Design Automation Testing and Trustworthy Integration", IPSJ Transactions on System LSI Design Methodology, vol. 10, pp. 45, 2017.

[19] D. Milojevic et al., "Design issues in heterogeneous 3D/2.5D integration", Design Automation Conference (ASP-DAC) 2013 18th Asia and South Pacific, 22–25 Jan. 2013.

[20] S. Priyadarshi, W. R. Davis, M. B. Steer, and P. D. Franzon, "Thermal pathfinding for 3-D ICs," IEEE Trans. Compon., Packag., Manuf. Technol., vol. 4, no. 7, pp. 1159–1168, Jul. 2013.

[21] Johann Knechtel and Jens Lienig, Physical Design Automation for 3D Chip Stacks: Challenges and Solutions. In Proceedings of the 2016 International Symposium on Physical Design (ISPD '16). ACM, New York, NY, USA, 3-10.

[22] Xiaodong Wang, Dilip Vasudevan, and Hsien-Hsin S. Lee. "Global built-in self-repair for 3D memories with redundancy sharing and parallel testing." 2011 IEEE International 3D Systems Integration Conference (3DIC), 2011 IEEE International. IEEE, 2012.

[23] Koyanagi, Mitsumasa. "Heterogeneous 3D integration—Technology enabler toward future super-chip." 2013 IEEE International Electron Devices Meeting. IEEE, 2013.

[24] Ma, He, Daquan Yu, and Jun Wang. "Thermal analysis and heat dissipation optimization of 3D packaging with TSV interposer." In 2012 13th International Conference on Electronic Packaging Technology & High Density Packaging, pp. 700-705. IEEE, 2012.

[25] Y. Chen, E. Kursun, D. Motschman, C. Johnson, Y. Xie, "Through silicon via aware design planning for thermally efficient 3-D integrated circuits", IEEE Trans. Computer-Aided Design Integr. Circuits Syst., vol. 32, no. 9, pp. 1335-1346, Sep. 2013.

[26] Xie, Jianyong, and Madhavan Swaminathan. "Electrical-thermal co-simulation of 3D integrated systems with micro-fluidic cooling and Joule heating effects." IEEE Transactions on Components, Packaging and Manufacturing Technology 1.2 (2011): 234-246.

[27] https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/ic-package-design-analysis/virtuoso-system-design-platform-ds.pdf

[28] https://www.zuken.com/en/product/cr-8000/chip-package-board-co-design/
[29] Satomi, Yuuta, et al. "Optimization of Full-Chip Power Distribution Networks in 3D ICs." 2018 IEEE 3rd International Conference on Integrated Circuits and Microsystems (ICICM). IEEE, 2018.
[30] Bailey, Chris, and Stoyan Stoyanov. "Co-Simulation and modelling for heterogeneous integration of high-tech electronic systems." 2017 40th International Spring Seminar on Electronics Technology (ISSE). IEEE, 2017.
[31] Ma, Yue. First order Electro-thermal compact models and noise considerations for three-dimensional integration circuits. Diss. 2018.
[32] Wan, Zhimin, et al. "Co-design of multicore architectures and microfluidic cooling for 3D stacked ICs." Microelectronics Journal 45.12 (2014): 1814-1821.
[33] Jain, Rahul, Preeti Ranjan Panda, and Sreenivas Subramoney. "Machine learned machines: adaptive co-optimization of caches, cores, and on-chip network." 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2016.
[34] https://www.mentor.com/resources/techpubs/upload/mentorpaper_55437.pdf
[35] https://www.cadence.com/content/cadence-www/global/en_US/home/company/events/webinars/genus-synthesis-solution-webinar.html
[36] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "PRIME: A Novel Processing-in-memory Architecture for Neural Network Computation in ReRAM-based Main Memory," SIGARCH Comput. Archit. News, vol. 44, no. 3, pp. 27–39, Jun. 2016
[37] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, "An evaluation of high-level mechanistic core models," ACM Transactions on Architecture and Code Optimization (TACO), pp. 28:1–28:25, 2014.
[38] Intel Corp., "Nios II Processor," Mountain View, CA, USA, 2017
[39] D. Reis, M. Niemier, and X. S. Hu, "Computing in memory with fefets," in IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). New York, NY, USA: ACM, 2018, pp. 24:1–24:6.
[40] J. Liu, H. Zhao, M. Ogleari, D. Li, and J. Zhao, "Processing-in-memory for energy-efficient neural network training: A heterogeneous approach," in IEEE/ACM International Symposium on Microarchitecture, 2018, pp. 185–197.
[41] K. Czechowski and R. Vuduc, "A Theoretical Framework for Algorithm-Architecture Co-design," In IEEE/ACM Elsevier International Symposium on Parallel and Distributed Processing, May 20-24, pp. 245-267
[42] R. Barret, S. Dosanjh, R. Barrett, D. Doerfler, D. Hammond, S. Hemmert, M. Heroux, L. Rodrigues, and L. Justin. "Exascale design space exploration and co-design," Worlshop on Future Generation Computer.
[43] Wim Bogaerts and Lukas Chrostowski, "Silicon Photonics Circuit Design: Methods, Tools and Challenges", Laser & Photonics Reviews, vol. 12, issue 4, 2018.
[44] Cheryl Sorace-Agaskar, Jonathan Leu, Michael R. Watts, and Vladimir Stojanovic, "Electro-optical co-simulation for integrated CMOS photonic circuits with VerilogA," Opt. Express 23, 27180-27203 (2015).
[45] Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. Nature 453, 80–83 (2008).
[46] Zidan, M. A., Strachan, J. P. & Lu, W. D. Nat. Electron 1, 22–29 (2018).
[47] Abunahla, H. and Mohammad, B., "Synthesis and Characterization of Micro-Thick TiO2 and HfO2 Memristors," In Memristor Technology: Synthesis and Modeling for Sensing and Security Applications, ed. Heba Abunahla and Baker Mohammad, 31–51. Analog Circuits and Signal Processing. Cham: Springer International Publishing, 2018.
[48] Jo, Sung Hyun; Chang, Ting; Ebong, Idongesit; Bhadviya, Bhavitavya B.; Mazumder, Pinaki; Lu, Wei (2010): Na noscale memristor device as synapse in neuromorphic systems. In: Nano letters 10 (4), S. 1297–1301. DOI: 10.1021/nl904092h.
[49] Shi, Yuanyuan; Liang, Xianhu; Yuan, Bin; Chen, Victoria; Li, Haitong; Hui, Fei et al. (2018): Electronic synapses made of layered two-dimensional materials. In: Nat Electron 1 (8), S. 458–465. DOI: 10.1038/s41928-018-0118-9.
[50] Sangwan, V., Lee, H.-S., Bergeron, H., Balla, I., Beck, M., Chen, K.-S., and C. Hersam, M., "Multi-terminal memtransistors from polycrystalline monolayer molybdenum disulfide," Nature, pp. 500–504, 2018.
[51] J. Lappalainen, J. Mizsei, and M. Huotari, "Neuromorphic thermal-electric circuits based on phase-change VO2 thin-film memristor elements," 044501 (2019); https://doi.org/10.1063/1.5037990.
[52] Shinhyun Choi, Scott H. Tan, Zefan Li, Yunjo Kim, Chanyeol Choi, Pai-Yu Chen, Hanwool Yeon, Shimeng Yu, and Jeehwan Kim, "SiGe epitaxial memory for neuromorphic computing with reproducible high performance based on engineered dislocations," Nature Materials volume 17, pages335–340 (2018).
[53] Hao Jiang, Lili Han, Peng Lin, Zhongrui Wang, Moon Hyung Jang, Qing Wu, Mark Barnell, J. Joshua Yang, Huolin L. Xin & Qiangfei Xia, Sub-10 nm Ta Channel Responsible for Superior Performance of a HfO2 Memristor, Scientific Reports | 6:28525
[54] B. Rajendran, A. Sebastia, M. Schuker, N. Srinivasa, E. Eleftheriou, "Low-Power Neuromorphic Hardware for Signal Processing Applications", arXiv:1902"03690v2, Apr 2019.
[55] J. Tang et al., "ECRAM as Scalable Synaptic Cell for High-Speed Low-Power Neuromorphic Computing", IEDM 2018.
[56] W. Haensch, "Analog Computing for Deep Learning: Algorithms, Materials & Architectures", IEDM 2018.
[57] A. Montanaro, "Quantum algorithms: an overview," npj Quantum Information, vol. 2, 2016.
[58] S. P. Jordan, "Quantum algorithm zoo." [Online]. Available: http://math.nist.gov/quan%20tum/zoo/

[59] R. P. Feynman, “Simulating physics with computers,” International Journal of Theoretical Physics, vol. 21, p. 467–488, 1982.
[60] P.W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM Journal on Computing, vol. 26, pp. 1484–1509, 1999.
[61] L. K. Grover, “Quantum mechanics helps in searching for a needle in a haystack,” Phys. Rev. Lett., vol. 79, pp. 325–328, Jul 1997.
[62] T. Albash and D. A. Lidar, “Adiabatic quantum computation,” Rev. Mod. Phys., vol. 90, p. 015002, Jan 2018.
[63] S. J. Devitt, W. J. Munro, and K. Nemoto, “Quantum error correction for beginners,” Reports on Progress in Physics, vol. 76, no. 7, p. 076001, Jun 2013.
[64] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” Phys. Rev. A, vol. 86, p. 032324, Sep 2012.
[65] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright and C. Monroe, “Demonstration of a small programmable quantum computer with atomic qubits,” Nature, vol. 536, p. no. 7614, pp. 63-66, Aug 2016
[66] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Yu Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.-C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, A. N. Cleland and J. M. Martinis, “State preservation by repetitive error detection in a superconducting quantum circuit,” Nature, vol. 519, p. 66, Mar 2015.
[67] A. Córcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow, “Demonstration of a quantum error detection code using a square lattice of four superconducting qubits,” Nature Communications, vol. 6, 2015.
[68] D. Ristè, S. Poletto, M.-Z. Huang, A. Bruno, V. Vesterinen, O.-P. Saira, and L. DiCarlo, “Detecting bit-flip errors in a logical qubit using stabilizer measurements,” Nature Communications, vol. 6, 2015.
[69] J. Preskill, “Quantum Computing in the NISQ era and beyond,” Quantum, vol. 2, p. 79, Aug. 2018.
[70] A. Acín, I. Bloch, H. Buhrman, T. Calarco, C. Eichler, J. Eisert, D. Esteve, N. Gisin, S. J. Glaser, F. Jelezko, S. Kuhr, M. Lewenstein, M. F. Riedel, P. O. Schmidt, R. Thew, A. Wallraff, I. Walmsley, and F. K. Wilhelm, “The quantum technologies roadmap: a european community view,” New Journal of Physics, vol. 20, no. 8, p. 080201, aug 2018.
[71] P. Gerbert and F. Ruess, “The next decade in quantum computing – and how to play,” Boston Consulting Group, 2018
[72] G. Popkin, “Quest for qubits,” Science, vol. 354, no. 6316, pp. 1090–1093, 2016.
[73] D. Castelvecchi, “Ibm’s quantum cloud computer goes commercial,” Nature News, vol. 543, no. 7644, p. 159, 2017.
[74] Rigetti, “Rigetti quantum cloud services.” [Online]. Available: https://www.rigetti.com/qcs
[75] D. P. DiVincenzo, “The physical implementation of quantum computation,” Fortschritte der Physik, vol. 48, pp. 771–783, 2000.
[76] C. D. Hill, E. Peretz, S. J. Hile, M. G. House, M. Fuechsle, S. Rogge, M. Y. Simmons, and L. C. L. Hollenberg, “A surface code quantum computer in silicon,” Science Advances, vol. 1, no. 9, 2015.
[77] J. O’Gorman, N. H. Nickerson, P. Ross, J. J. Morton, and S. C. Benjamin, “A silicon-based surface code quantum computer,” vol. 2, 2016.
[78] R. Li, L. Petit, D. P. Franke, J. P. Dehollain, J. Helsen, M. Steudtner, N. K. Thomas, Z. R. Yoscovits, K. J. Singh, S. Wehner, L. M. K. Vandersypen, J. S. Clarke, and M. Veldhorst, “A crossbar network for silicon quantum dot qubits,” Science Advances, vol. 4, no. 7, 2018.
[79] L. M. K. Vandersypen, H. Bluhm, J. S. Clarke, A. S. Dzurak, R. Ishihara, A. Morello, D. J. Reilly, L. R. Schreiber, and M. Veldhorst, “Interfacing spin qubits in quantum dots and donors—hot, dense, and coherent,” npj Quantum Information, vol. 3, 2017.
[80] R. Versluis, S. Poletto, N. Khammassi, B. Tarasinski, N. Haider, D. J. Michalak, A. Bruno, K. Bertels, and L. DiCarlo, “Scalable quantum circuit and control for a superconducting surface code,” arXiv preprint arXiv:1612.08208, 2016.
[81] C. Monroe and J. Kim, “Scaling the ion trap quantum processor,” Science, vol. 339, 2013.
[82] R. C. Sterling, H. Rattanasonti, S. Weidt, K. Lake, P. Srinivasan, S. C. Webster, M. Kraft, and W. K. Hensinger, “Fabrication and operation of a two-dimensional ion-trap lattice on a high-voltage microchip,” Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., vol. 5, 2014.
[83] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger, “Blueprint for a microwave trapped ion quantum computer,” Science Advances, vol. 3, no. 2, 2017.
[84] D. J. Reilly, “Engineering the quantum-classical interface of solid-state qubits,” Npj Quantum Information, vol. 1, 2015.
[85] B. Patra, R. M. Incandela, J. P. G. van Dijk, H. A. R. Homulle, L. Song, M. Shahmohammadi, R. B. Staszewski, A. Vladimirescu, M. Babaie, F. Sebastiano, and E. Charbon, “Cryo-cmos circuits and systems for quantum computing applications,” IEEE Journal of Solid-State Circuits, vol. 53, pp. 309–321, 2018.
[86] J.P.G. van Dijk, E. Charbon, and F. Sebastiano, “The electronic interface for quantum processors,” Microprocessors and Microsystems, vol. 66, pp. 90–101, 2019.
[87] J. M. Hornibrook, J. I. Colless, I. D. Conway Lamb, S. J. Pauka, H. Lu, A. C. Gossard, J. D. Watson, G. C. Gardner, S. Fallahi, M. J. Manfra, and D. J. Reilly, “Cryogenic control architecture for large-scale quantum computing,” Phys. Rev. Applied, vol. 3, p. 024010, Feb 2015.

[88] R. LaRose, "Overview and Comparison of Gate Level Quantum Software Platforms," Quantum, vol. 3, p. 130, Mar. 2019.
[89] R. V. Meter and C. Horsman, "A blueprint for building a quantum computer," Communications of the ACM, vol. 56, pp. 84–93, 2013.
[90] Y. Shi, N. Leung, P. Gokhale, Z. Rossi, D. I. Schuster, H. Hoffman, and F. T. Chong, "Optimized compilation of aggregated instructions for realistic quantum computers," ArXiv preprint arXiv:1902.01474v2 [quantph], 2019.
[91] C. García Almudever, L. Lao, X. Fu, N. Khammassi, I. Ashraf, D. Iorga, S. Varsamopoulos, C. Eichler, A. Wallraff, L. Geck, A. Kruth, J. Knoch, H. Bluhm, and K. Bertels, "The engineering challenges in quantum computing," in Proceedings of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE). United States: IEEE, 2017, pp. 836–845.
[92] K. M. Svore, A. V. Aho, A. W. Cross, I. Chuang, and I. L. Markov, "A layered software architecture for quantum computing design tools," Computer, vol. 39, no. 1, pp. 74–83, Jan. 2006.
[93] S. Balensiefer, L. Kregor-Stickles, and M. Oskin, "An evaluation framework and instruction set architecture for ion-trap based quantum micro-architectures," SIGARCH Comput. Archit. News, vol. 33, no. 2, pp. 186–196, May 2005.
[94] R. S. Smith, M. J. Curtis, and W. J. Zeng, "A practical quantum instruction set architecture," ArXiv preprint arXiv:1608.03355v2 [quantinstruction set architecture," ArXiv preprint arXiv:1608.03355v2 [quantph], 2017.
[95] X. Fu, L. Riesebos, M. A. Rol, J. van Straten, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, V. Newsum, K. K. L. Loh, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudéver, L. Di-Carlo, and K. Bertels, "eqasm: An executable quantum instruction set architecture," 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 224–237, 2018.
[96] X. Fu, M. Rol, C. Bultink, J. van Someren, N. Khammassi, I. Ashraf, R. Vermeulen, J. De Sterke, W. Vlothuizen, R. Schouten, C. Almudéver, L. DiCarlo, and K. Bertels, "A microarchitecture for a superconducting quantum processor," IEEE Micro, vol. 38, no. 3, pp. 40–47, 2018.
[97] A. J. McCaskey, E. F. Dumitrescu, D. Liakh, and T. S. Humble, "Hybrid programming for near-term quantum computing systems," 2018 IEEE International Conference on Rebooting Computing (ICRC), pp. 1–12, 2018.
[98] L. Riesebos, X. Fu, A. A. Moueddenne, L. Lao, S. Varsamopoulos, I. Ashraf, J. van Someren, N. Khammassi, C. G. Almudever, and K. Bertels, "Quantum accelerated computer architectures," in 2019 IEEE International Symposium on Circuits and Systems (ISCAS), May 2019, pp. 1–4.
[99] W. Zeng, B. Johnson, R. Smith, N. Rubin, M. Reagor, C. Ryan, and C. Rigetti, "First quantum computers need smart software," Nature News, vol. 549, no. 7671, p. 149, 2017.
[100] K. Svore, A. Geller, M. Troyer, J. Azariah, C. Granade, B. Heim, V. Kliuchnikov, M. Mykhailova, A. Paz, and M. Roetteler, "Q#: Enabling scalable quantum computing and development with a high-level dsl," in Proceedings of the Real World Domain Specific Languages Workshop 2018, ser. RWDSL2018. New York, NY, USA: ACM, 2018, pp. 7:1–7:10.

Edited by Paul Wesling