ENGINEERING Information Technology & Electronic Engineering
https://www.academax.com/EITEE; https://www.engineering.org.cn
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11344065
E-mail: eng.itee@zju.edu.cn

Available on IEEE Xplore®

Research Article

https://doi.org/10.1631/ENG.ITEE.2025.0063

From software-defined interconnect to software-defined system-on-wafer: a computing architecture revolution in the post-Moore era

Ping LV[1], Qin-rang LIU[2]✉, Jiangxing WU[3], Jianliang SHEN[1], Mengke LIAN[1], Rui CAO[1], Shuai WEI[1], Zhichao LI[1], Peijie LI[1], Wei GUO[1], Wenjian ZHANG[1], Hong YU[1], Yan ZHAO[1]

[1]*Information Engineering University, Zhengzhou 450001, China*
[2]*Institute of Big Data, Fudan University, Shanghai 200433, China*
[3]*National Digital Switching System Engineering & Technology Research Center, Zhengzhou 450002, China*

Abstract: As Moore's Law approaches its fundamental physical and economic limits, the semiconductor industry faces unprecedented challenges in maintaining performance growth. This study presents the revolutionary evolution from software-defined interconnect (SDI) to software-defined system-on-wafer (SDSoW), a paradigm-shifting architectural approach that transcends traditional scaling constraints through wafer-level heterogeneous integration. Our proposed SDSoW enables dynamic reconfiguration of thousands of computing chiplets across an entire wafer, achieving superlinear performance scaling and significantly improving energy efficiency. We established a comprehensive theoretical framework with mathematical models covering key aspects, such as interconnect flexibility and integration scaling, and proposed an application-driven dynamic architecture reconfiguration (ADR) paradigm that optimizes wafer-scale resources in real time and may foster emergent intelligence in large, heterogeneous systems. Simulation results (128–1024 nodes) demonstrate that SDSoW outperforms conventional multi-chip systems, delivering 3.5× higher throughput, 80% lower latency, and 2.5× better energy efficiency. As a paradigm shift comparable to the invention of integrated circuits (ICs), it provides a viable pathway beyond Moore's Law through innovative architectural design rather than process scaling.

Key words: Software-defined interconnect; Software-defined system-on-wafer; Wafer-level integration; Emergent intelligence; Heterogeneous computing

1 Introduction

Moore's Law, the foundational principle that has driven semiconductor progress for over five decades, now faces fundamental physical and economic barriers (Khan et al., 2018; Leiserson et al., 2020). The exponential increase in transistor density, as predicted by Gordon Moore in 1965, enabled unprecedented computational advances but is increasingly challenged by quantum tunneling effects, manufacturing complexity, and escalating costs (Shalf, 2020; Theis and Wong, 2017).

Traditional scaling approaches face critical limitations at advanced process nodes below 7nm, where leakage currents, variability, and thermal effects severely impact performance and reliability (Radamson et al., 2020). The economic burden of next-generation fabrication facilities, exceeding $20 billion per facility, has made continued scaling economically unsustainable for most applications (Khan et al., 2018; Shalf, 2020).

The evolution of semiconductor integration paradigms shows a clear trajectory from System-on-Chip (SoC) through 2.5D/3D chiplet architectures toward wafer-level integration (Lau, 2022). Each paradigm shift has enabled new levels of performance and functionality while addressing the limitations of previous approaches.

This study introduces software-defined system-on-wafer (SDSoW), a revolutionary architectural approach that funda-

✉ Qinrang LIU, qinrangliu@sina.com
Ping LV, http://orcid.org/0009-0008-1608-6597
Qinrang LIU, https://orcid.org/0000-0002-9957-7365
Jiangxing WU, http://orcid.org/0000-0003-0255-0826
Jianlinag SHEN, http://orcid.org/0009-0000-5647-043x
Mengke LIAN, http://orcid.org/0009-0007-4677-2566
CLC number:

Received: Sep. 30, 2025; Revision accepted: Mar. 13, 2026;
Crosschecked: Mar. 30, 2026

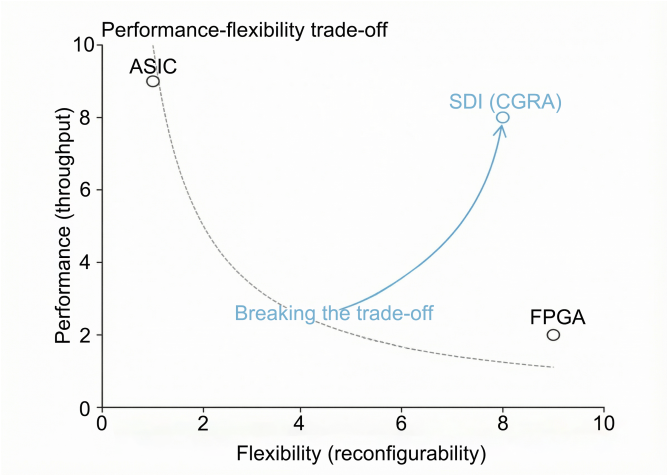

Fig. 1 Characteristics and positioning of SDI chips in the performance–flexibility design space. The qualitative classification is synthesized from (Gomes Mesquita and Frosi Rosa, 2017; Huang et al., 2022) and our own analysis; the axes are normalized and illustrative rather than based on a single measured dataset

mentally redefines computing system design. SDSoW leverages wafer-level integration with SDIs to create massively parallel, dynamically reconfigurable systems that achieve performance scaling beyond Moore's Law constraints.

2 Software-defined interconnect: the foundational technology

2.1 Conceptual framework and technical innovation

Software-defined interconnect (SDI) replaces fixed-function interconnects with a software-controlled, dynamically reconfigurable communication fabric. Instead of static protocols, bandwidth allocation, and rigid topologies, SDI enables software to configure protocol types, port roles, topology, bandwidth management, and QoS in a unified manner (Berestizshevsky et al., 2017). The key innovation is decoupling interconnect behavior from hardware, so applications can redefine and reconfigure communication at runtime while maintaining high performance (Gomez-Rodriguez et al., 2021).

In the performance–flexibility design space, SDI chips belong to the class of high-performance, high-flexibility programmable switching devices. As illustrated in Fig. 1, they offer mixed-granularity soft–hard programmability, application-driven design, minimal dependence on low-level chip knowledge, and good energy efficiency.

2.2 Mathematical framework for SDI flexibility

In SDI systems, architectural flexibility—defined as the ability to adapt to dynamic workloads, protocol requirements (the set of link-level and end-to-end properties mandated by the communication standards supported by the system, including data rates, voltage levels, encoding schemes, flow control mechanisms, ordering guarantees, reliability, and packet/frame formats), and network topologies—has become a core differentiator from traditional fixed interconnects. To objectively evaluate this flexibility, a quantitative metric is essential for benchmarking system performance and guiding design optimization. This section introduces the SDI Flexibility Index (F_{SDI}), a multidimensional metric that encapsulates four key dimensions of interconnect adaptability, enabling systematic comparison between SDI implementations and conventional interconnect architectures.

To quantify adaptability, we define the SDI Flexibility Index F_{SDI} is:

$$F_{\mathrm{SDI}} = K_{\mathrm{corr}} \times P \times T \times Q \times R, \tag{1}$$

P, T, Q, and R denote protocol flexibility, topology reconfigurability, QoS granularity, and reconfiguration rate, respectively; $K_{\mathrm{corr}} \in (0, 1]$ is an empirical coefficient capturing non-ideal coupling among these dimensions.

2.2.1 Parameter definitions and typical ranges

1. Protocol flexibility factor (P): number and diversity of protocols supported concurrently, from a few serial or lightweight protocols in embedded systems to broad multi–standard support (e.g., PCIe, Ethernet) in high-end SDI (Ma et al., 2022).

2. Topology reconfigurability factor (T): number of distinct topologies realizable in software without hardware changes. Small systems support basic star or bus structures ($T \approx 10^3$); large chiplet or wafer-scale systems program meshes, tori, and other complex topologies ($T \approx 10^6$) (Pal, 2021).

3. QoS granularity (Q): number of QoS classes for traffic differentiation. Basic designs offer about four priority levels; advanced systems support up to 64 classes with fine-grained scheduling (Deshmukh and Mane, 2023).

4. Reconfiguration rate (R): number of configuration updates per second. Datacenter SDI and chiplet fabrics typically reach $10^2 - 10^6$ Hz, limited by control-plane speed and switch complexity (Gomez-Rodriguez et al., 2021).

Microarchitectural mechanisms such as arbitration, DVFS, clock-domain crossing, routing, buffer management, and switch bandwidth constrain feasible P, T, Q, and R. They are not explicit in Eq. (1), but define the attainable design space (Sandoval-Arechiga et al., 2015).

Since P, T, Q, and R are coupled (e.g., richer protocol sets may reduce maximum R, very fine–grained QoS may limit T), Eq. (1) is a first-order approximation. K_{corr} is calibrated from design exploration or silicon measurements. F_{SDI} should thus be interpreted as a comparative, dimensionless flexibility indicator rather than an exact physical quantity.

2.2.2 Flexibility comparison: SDI vs. conventional interconnects

Conventional fixed interconnects (e.g., hardwired buses and fixed mesh networks) lack adaptive capabilities, resulting in a flexibility index of $F_{\mathrm{conventional}} = 1$ (single unchangeable configuration) (Gomez-Rodriguez et al., 2021). In contrast, advanced SDI implementations—integrating multi-protocol support, reconfigurable topologies, fine-grained QoS, and high reconfiguration rates—achieve $F_{\mathrm{SDI}} > 10^6$. This million-fold improvement enables SDI systems to dynamically optimize

performance across scenarios, from low-cost embedded devices to wafer-scale AI accelerators.

2.2.3 Flexibility–cost–efficiency trade-offs

Flexibility is not free. Higher P requires additional parsers and state machines (Attig M, 2011); larger T demands more complex switch fabrics and configuration memories (Sivaraman et al., 2011); higher Q increases scheduler and buffer overhead (Sivaraman et al., 2016); and higher R necessitates faster, more power-hungry control planes. These overheads can be modeled as a weighted combination of P, T, Q, and R with technology-dependent coefficients (Programmable packet scheduling at line rate).

Practical SDI and SDSoW designs therefore seek application-appropriate flexibility rather than maximizing F_{SDI}. Domain-specific accelerators with stable communication patterns may choose moderate P and Q with carefully selected T and R, while multi-tenant data-center fabrics or wafer-scale heterogeneous platforms justify higher F_{SDI} within power, area, yield, and design-complexity limits.

The mathematical models in this study—including the SDI flexibility index, integration scaling factor, system complexity measures, and performance/energy metrics—are intentionally high-level and phenomenological, aimed at providing design-space insight rather than exact first-principles predictions.

2.3 Relation to software-defined network–on–chip (NoC) and MPSoC Management

Previous work on software-defined on-chip communication (SDNoC and SDN-inspired MPSoC management) introduced centralized or logically centralized controllers to configure routing, virtual channels, and traffic policies, enabling runtime reconfiguration and per-flow guarantees at the chip or MPSoC scale (Gomez-Rodriguez et al., 2021; Gonzalez-Martinez et al., 2024; Ji et al., 2023; Prashanth and N, 2024; Sandoval-Arechiga et al., 2015; Salvador et al., 2017; Sandoval-Arechiga et al., 2017).

Our work extends these ideas in three aspects: (1) scale and granularity: SDSoW targets wafer-scale integration of thousands of heterogeneous dielets; (2) software-defined scope: SDSoW software-defines not only routing but also protocols, topologies, QoS classes, and reconfiguration rates; and (3) complexity and emergent behavior: we introduce complexity and emergent-intelligence metrics inspired by complex-systems theory.

2.4 Three-dimensional software definability

SDI's architectural innovation spans three dimensions: ports, communication modes, and protocol processing (Lv et al., 2018).

Software-definable interconnect ports: Each port can be configured for standard (RapidIO, PCIe, Ethernet, InfiniBand and FC) or user-defined protocols, different port types and bonding widths, a range of data rates, and programmable QoS policies (service models, performance metrics, and scheduling algorithms).

Software-definable communication modes: SDI supports circuit, packet, and message switching; unicast, multicast, and broadcast; and a wide range of software-instantiated topologies (star, tree, ring, mesh/torus, hypercube and hierarchical and hybrid structures). All are specified in configuration files and applied to the data plane without hardware changes.

Software-definable protocol processing: Flexible pipelines perform multi–layer parsing, heterogeneous protocol conversion (including field mapping and packet-size adaptation), and content processing, such as encryption, monitoring, and statistics collection, enabling seamless communication across heterogeneous environments.

2.5 SDI implementation architecture

A typical SDI system comprises three core components. The Software-Defined Protocol Controller configures physical and link-layer parameters and protocol types. The Software-Defined Forwarder performs parsing, encapsulation, protocol conversion, and forwarding-table lookup. The Software-Defined Switching Fabric provides the reconfigurable topology, switching mode, and QoS infrastructure, as illustrated in Fig. 2.

3 Mesoscopic scale integration: the physical foundation of SDSoW

3.1 Scaling from chip to wafer: dimensional transformation

3.1.1 Core integration differences between SDSoW and SoC

As integrated circuit (IC) technology advances toward higher integration density and performance, SoC has long served as the cornerstone of traditional integration strategies, underpinning the miniaturization and functional aggregation of complex electronic systems (Wu et al., 2024). However, as the number of integrated functional modules (e.g., processing cores, memory units, and peripheral interfaces) continues to grow, SoC architectures are confronting inherent bottlenecks that are increasingly constraining both manufacturing efficiency and overall system performance.Specifically, the pursuit of higher integration density directly contributes to substantial yield loss in manufacturing. Meanwhile, the inherent planar integration scheme of SoCs exacerbates issues related to power density and area redundancy, making it challenging to meet the stringent requirements for energy efficiency and spatial utilization demanded by next-generation electronic systems.

To address these challenges, heterogeneous integration and 3D integration technologies have emerged as transformative solutions, breaking through the physical constraints of planar integration. By enabling efficient interconnection and stacking of chips with diverse processes and functions, these technologies have given rise to the SDSoW concept, marking a fundamental shift from "chip-scale" to "wafer-scale" integration philosophy (Wu et al., 2024).

3.1.2 Quantitative model for SDSoW system complexity

SDSoW system enables the integration of thousands of heterogeneous computing elements ("dielets") including pro-

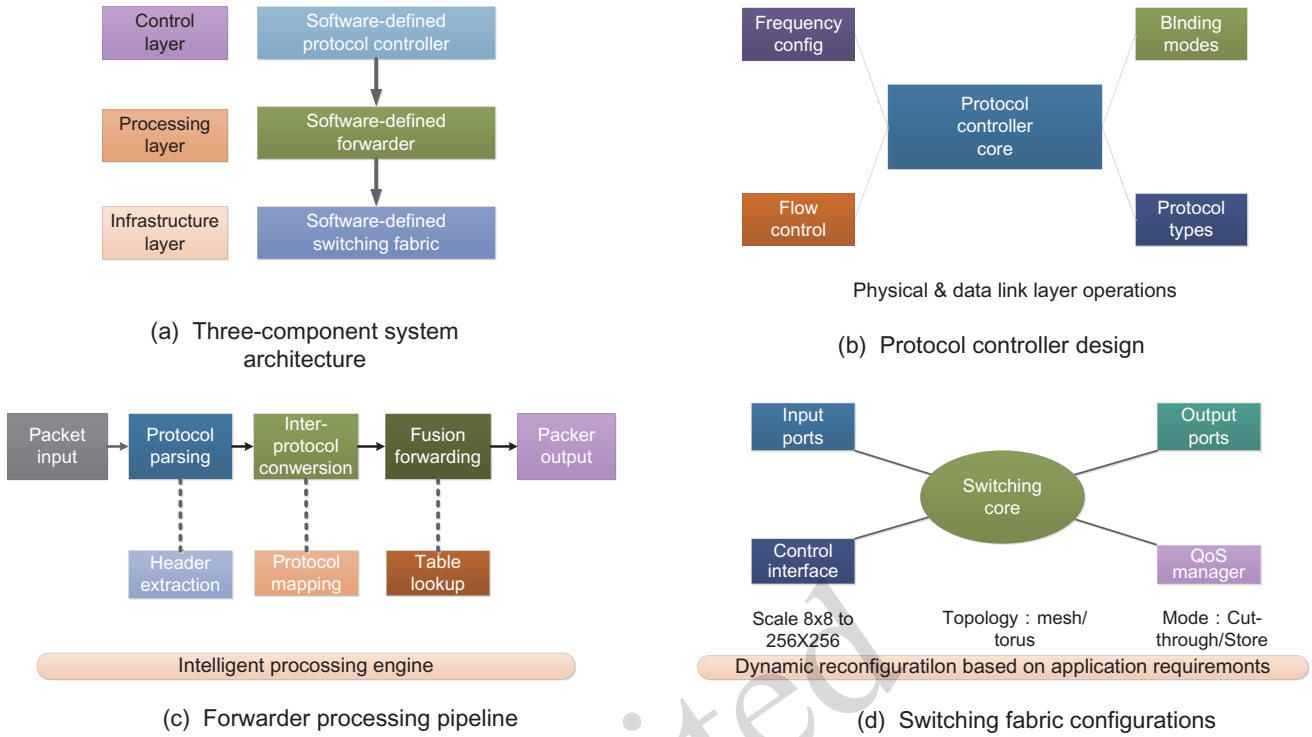

Fig. 2 SDI implementation architecture

cessors, accelerators, memory hierarchies, and I/O controllers onto a single substrate. To quantify the integration capability and system complexity of SDSoW, this study introduces the integration scaling factor ($S_{\text{integration}}$) as a key metric, defined by Eq. (2), $S_{\text{integration}}$ is also dimensionless:

$$S_{\text{integration}} = (A_{\text{wafer}}/A_{\text{chip}}) \times \eta_{\text{yield}} \times \rho_{\text{interconnect}}, \quad (2)$$

The physical meaning and typical values of each parameter in the equation are as follows:

1. A_{wafer}: Effective wafer integration area, referring to the actual area available for dielet integration after deducting the invalid edge area. For a 300 mm wafer, its total physical area is approximately 707 cm^2; considering an edge exclusion rate of 10%, the actual effective integration area is about 636.3 cm^2.

2. A_{chip}: Area of a single dielet. Optimized dielet designs usually adopt a 5 mm×5 mm specification (0.25 cm^2) to balance functional coverage and yield stability.

3. η_{yield}: Effective yield coefficient, which comprehensively reflects defect density in the manufacturing process, redundant repair mechanisms, and system fault tolerance. According to differences in process maturity, the value range of η_{yield} is 0.7–0.95; defect control technologies in advanced processes can make this coefficient approach the upper limit.

4. $\rho_{\text{interconnect}}$: Interconnect density enhancement factor, determined jointly by advanced interconnect materials (e.g., carbon nanotubes (CNT), graphene) and 3D integration technologies (e.g., through-silicon vias and hybrid bonding). Compared with traditional copper interconnect solutions, $\rho_{\text{interconnect}}$ can achieve a 2–10× density improvement, which directly affects the signal transmission efficiency and integration density between dielets (Lee et al., 2025; Rudolph et al., 2021).

Taking the typical solution of a 300 mm wafer paired with 5 mm × 5 mm dielets as an example, substituting into Eq. (2) for calculation gives:

1. When $\eta_{\text{yield}} = 0.7$ and $\rho_{\text{interconnect}} = 2$ (basic process conditions), $S_{\text{integration}} \approx \frac{636.3}{0.25} \times 0.7 \times 2 \approx 3563$.

2. When $\eta_{\text{yield}} = 0.95$ and $\rho_{\text{interconnect}} = 10$ (high-performance interconnect technology), $S_{\text{integration}} \approx \frac{636.3}{0.25} \times 0.95 \times 10 \approx 24200$.

However, considering practical application constraints such as process compatibility and cost control, the typical integration scaling factor range of this solution is 1000 ×–5000×, which further verifies the technical advantages of SDSoW in large-scale integration scenarios.

3.2 Advanced interconnect technologies

SDSoW implementation requires breakthrough advances in interconnect materials and manufacturing processes. Besides conventional copper wiring, several emerging technologies are particularly promising for wafer-scale integration:

1. CNT interconnects offer superior electrical properties with conductivity and current-carrying capacity significantly exceeding those of copper, enabling narrower wires and reduced IR drop for local and intermediate-distance links. (Behnam et al., 2013)

2. Graphene-based conductors provide exceptional in-plane thermal conductivity (exceeding 3000–5000 $\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$), which can be leveraged for both electrical interconnects and lateral heat spreading across the wafer. (AA.,

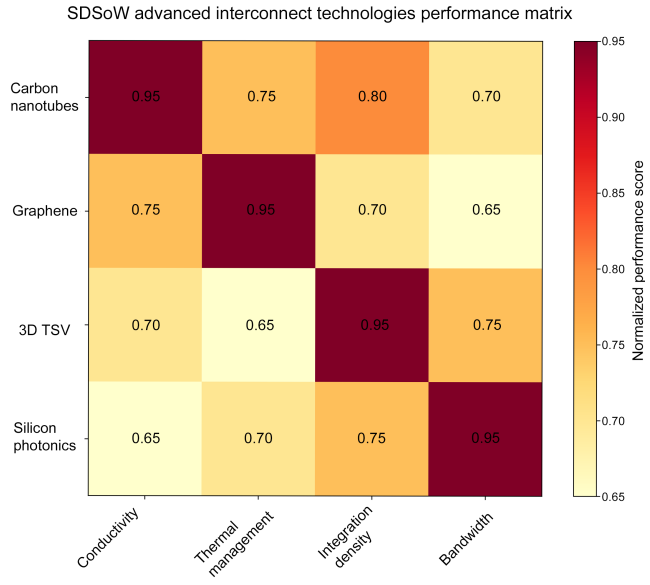

Fig. 3 Advanced interconnect technology performance matrix for SDSoW: CNT excel in conductivity (0.95), Graphene in thermal management (0.95), 3D TSV in integration density (0.95), and silicon photonics in bandwidth (0.95)

2020).

3. Three-dimensional through-silicon vias (TSVs) and hybrid bonding provide high-density vertical integration, enabling stacking of logic, memory, and specialized accelerators, thereby increasing the effective integration density of SDSoW systems (Chen KN, 2023).

4. Silicon photonic interconnects offer extremely high bandwidth and low loss over longer on-wafer distances, mitigating RC limitations of electrical wires and enabling energy-efficient global communication across the wafer (Miller, 2017).

Fig. 3 qualitatively compares these technologies along four normalized dimensions—conductivity, thermal management, integration density, and bandwidth—highlighting their complementary strengths for SDSoW deployment.

The normalized scores in Fig. 3 are derived by mapping each physical metric to a dimensionless value in [0,1] using literature-reported ranges as the reference. For each category—conductivity, thermal management, integration density, and bandwidth—we select the best-known technology within that category as the baseline (assigned a score close to 1) and normalize other technologies accordingly:

$$S_{\text{norm}} = \frac{M_{\text{tech}} - M_{\text{min}}}{M_{\text{max}} - M_{\text{min}}}. \quad (3)$$

where M_{tech} is the metric value (e.g., thermal conductivity), and M_{min} and M_{max} are the lower and upper bounds of the range observed in representative technology nodes. These scores are intended as a compact, comparative illustration rather than precise measurements.

4 Software-defined system-on-wafer: architectural innovation

4.1 Architectural evolution: from SDI to SDSoW

The evolution from SDI to SDSoW represents a scaling transformation that extends software-defined principles from individual interconnects to entire wafer-level systems. The SDSoW complexity scaling factor is characterized by:

$$C_{\text{SDSoW}} = N_{\text{elements}} \times I_{\text{interconnects}} \times L_{\text{layers}} \times D_{\text{dynamics}}, \quad (4)$$

C_{SDSoW} is a dimensionless measure constructed from the number of elements, interconnects, layers, and dynamic configuration states. Where each parameter quantifies a distinct aspect of system complexity:

1. N_{elements} represents the number of computational elements (dielets) integrated on the wafer, typically ranging from 100 to 10,000 depending on dielet size and wafer utilization.

2. $I_{\text{interconnects}}$ denotes the interconnect density per unit area, measured in connections per cm^2. Advanced 3D integration achieves densities of 10^4–10^6 connections/cm^2. (Chew SA et al., 2024)

3. L_{layers} indicates the number of hierarchical layers in the system architecture, including local clusters, global routing, and control planes. Typical implementations use 3–7 hierarchical levels.

4. D_{dynamics} represents the dynamic reconfiguration frequency capability in Hz, reflecting the system's adaptation speed.

4.2 Application-driven dynamic architecture reconfiguration (ADR) paradigm

The core innovation of SDSoW lies in implementing the ADR paradigm, where the SDSoW can reconfigure its topology, routing, and QoS settings at run time to better match computational requirements. To make this notion of ADR more concrete, we briefly highlighted three representative scenarios. First, for compute-bound workloads such as dense linear algebra or deep neural network training, the SDSoW fabric can be configured as a low-diameter mesh or torus with widened links between computer–memory clusters and throughput-oriented QoS settings (Hall et al., 2021; Systems, 2019). Second, for communication- and memory-bound workloads such as large-scale graph analytics, the system can adopt a more locality-aware or hierarchical topology, clustering frequently communicating dielets and allocating extra capacity around communication hotspots (Ham et al., 2016; Mukkara et al., 2018). Third, for real-time or mixed-critical applications, a subset of the wafer can be configured with circuit-switched or time-triggered communication and reserved bandwidth to provide temporal and spatial isolation for critical flows, while the rest of the wafer uses best-effort packet-switched communication (Goossens et al., 2017; Kerrison et al., 2016). In all cases, reconfiguration decisions are driven by application-level descriptors and run-time monitoring, enabling the software-defined control plane to adapt topology, routing, and QoS in a workload-aware manner.

Dynamic architectural adaptation is achieved through three key mechanisms:

1. Wafer-scale resource virtualization: SDSoW abstracts the entire wafer's resources–thousands of heterogeneous dielets including processors, accelerators, memory, and I/O controllers—into unified, programmable resource pools. This abstraction is achieved through the SDI infrastructure that provides flexible communication pathways between any combination of dielets;

2. Dynamic topology reconfiguration: Building upon SDI's topology definition capabilities, SDSoW can dynamically reconfigure the interconnection patterns between dielets. For AI training workloads, the system can form optimized All-Reduce topologies connecting hundreds of accelerator dielets. For graph processing applications, it can reconfigure into memory-centric structures that minimize data movement for sparse computations.

3. Hierarchical protocol management: SDSoW leverages SDI's multi–protocol support to enable hierarchical communication structures across the wafer. Different regions can operate under different protocol regimes optimized for their specific functions, while the overall system maintains coherent operation through software–defined protocol bridges and converters.

This paradigm fundamentally reverses traditional design approaches where applications must adapt to fixed hardware architectures. Instead, SDSoW dynamically reconfigures its computational topology, interconnect patterns, and resource allocation to optimize performance for specific applications or workload phases.

4.3 Hierarchical analytical modeling of SDSoW performance

This subsection develops a hierarchy of tightly coupled analytical models that relate structural properties of SDSoW fabrics to system-level behavior. Rather than four independent metrics, we used four interrelated models:

1. An interconnect efficiency model that captures how well a given topology and routing scheme utilizes physical links.

2. A comprehensive performance scaling model that predicts system throughput as node and edge counts grow.

3. An energy-efficiency model that combines performance with power to identify performance-per-watt optima.

4. An emergent-intelligence threshold model that estimates when SDSoW systems cross qualitative capability thresholds.

All four models share the same core variables – the number of nodes N, the number of edges E, interconnect efficiency, and SDI flexibility – and reuse the factors introduced in previous sections (interconnect efficiency, energy efficiency, and emergent capability). Thus, the models form a chain:

$$\text{structure} \;\rightarrow\; \eta_{\text{int}} \;\rightarrow\; P_{\text{total}} \;\rightarrow\; \eta_{\text{EE}} \;\rightarrow\; F_{\text{em}},$$

where η_{int} denotes interconnect efficiency, P_{total} represents system performance, η_{EE} is energy efficiency, and F_{em} is the emergent-intelligence factor.

4.3.1 Interconnect efficiency model

We first defined a normalized interconnect efficiency factor $\eta_{\text{int}}(N, E) \in [0, 1]$ that summarizes how effectively the SDSoW fabric utilizes its physical links under a given topology, routing policy, and traffic pattern. Building on the qualitative discussion in Section 3, we decomposed η_{int} into three normalized components:

$$\eta_{\text{int}}(N, E) = \eta_{\text{topo}}(N, E)\, \eta_{\text{load}}(N, E)\, \eta_{\text{qos}}(N, E), \tag{5}$$

where $\eta_{\text{int}}(N, E)$ denotes the overall normalized interconnect efficiency; $\eta_{\text{topo}}(N, E)$ represents the topological quality factor; $\eta_{\text{load}}(N, E)$ captures load balance and congestion effects; $\eta_{\text{qos}}(N, E)$ accounts for QoS and latency constraints.

A simple topological efficiency factor can be expressed as

$$\eta_{\text{topo}}(N, E) = \min\left(1,\, \frac{H_{\text{ref}}}{H(N, E)}\right), \tag{6}$$

where $H(N, E)$ is the average shortest path hop count of the current topology; H_{ref} is the reference hop count corresponding to a desired low-diameter topology.

The load-balance factor is defined as

$$\eta_{\text{load}}(N, E) = 1 - \frac{\sigma_{\text{link}}(N, E)}{\sigma_{\max}}, \tag{7}$$

where σ_{link} denotes the standard deviation of per-link utilization, and $\sigma_{\max}$ is a normalization constant representing the worst acceptable imbalance.

The term $\eta_{\text{qos}}(N, E)$ can be defined analogously as a normalized factor in $[0, 1]$, decreasing when strict QoS constraints force inefficient routing or capacity reservations. In the rest of this subsection, we used $\eta_{\text{int}}(N, E)$ as a compact summary of these effects and fed it into the higher-level performance and energy models.

In the simulation, we adopted a fixed-degree network topology (average degree $k = 8$) so that $E = kN/2$ scales linearly with N, and $\eta_{\text{int}}(N, E)$ reduces to a function of N alone. As shown in Fig. 4, η_{int} decreases monotonically with system size N. This is consistent with the analytical structure of Eq. (5): as N grows, the average shortest-path hop count $H(N, E)$ increases, degrading η_{topo}; the per-link utilization becomes less uniform, reducing η_{load}; and stricter QoS demands further lower η_{qos}. The rate of decline is initially steep but gradually diminishes at larger scales, reflecting the sublogarithmic growth of $H(N, E)$ in well-connected topologies. The analysis indicates that η_{int} is jointly shaped by topology quality, traffic load balance, and QoS constraints, implying that both physical connectivity and logical flow control must be co-optimized to sustain high-throughput, low-latency communication in large-scale SDSoW deployments.

4.3.2 Comprehensive performance scaling models

We next constructed a comprehensive performance scaling model that combines node-level computation and network-level communication, while reusing the SDI flexibility index F_{SDI} and the interconnect efficiency factor $\eta_{\text{int}}(N, E)$.

The base performance is modeled as the sum of a node-dominated term and a network-dominated term:

$$P_{\text{base}}(N, E) = P_{\text{node}}(N) + P_{\text{network}}(N, E), \tag{8}$$

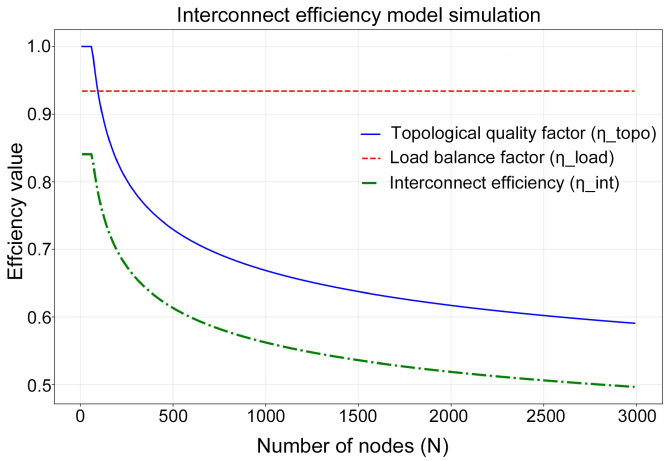

Fig. 4 Simulation results of interconnection efficiency model

The node-dominated contribution is modeled as

$$P_{\text{node}}(N) = N \exp(-k_1 N), \tag{9}$$

which grows linearly for small N and then decays beyond a characteristic scale $1/k_1$ as coordination, memory, and management overheads start to outweigh the benefits of adding more nodes. A representative illustrative value is $k_1 = 0.002$, which yields a peak around $N \approx 1/k_1 \approx 500$.

The network-dominated contribution is modeled as

$$P_{\text{network}}(N, E) = k_2\, E\, \eta_{\text{int}}(N, E)\, f_{\text{net}}(N)\, \ln(N+1) \times \left(1 + \lambda F_{\text{SDI}}\right), \tag{10}$$

where E is the number of interconnect edges; k_2 is the scaling coefficient for network-dominated performance; $\eta_{\text{int}}(N, E)$ is the interconnect efficiency defined in Section 4.3.1; $f_{\text{net}}(N)$ is a logistic phase-transition factor (defined below); $\ln(N+1)$ captures sub-linear scaling with system size; F_{SDI} is the SDI flexibility index introduced earlier; λ is a calibration coefficient quantifying how strongly SDI flexibility amplifies network performance.

The network phase-transition factor is defined as

$$\text{network}_{\text{factor}}(N) = \frac{1}{1 + \exp\left(-\dfrac{N - N_{\text{threshold}}}{0.1\, N_{\text{threshold}}}\right)}, \tag{11}$$

where $N_{\text{threshold}}$ denotes the node-count scale at which network effects become dominant, and the factor $0.1\, N_{\text{threshold}}$ controls the width of the transition region.

For $N \ll N_{\text{threshold}}$, $\text{network}_{\text{factor}}(N) \approx 0$ and $P_{\text{network}}(N, E)$ is negligible, for $N \gg N_{\text{threshold}}$, $\text{network}_{\text{factor}}(N) \approx 1$ and the system becomes network dominated.

To avoid unbounded growth at extreme scales, we mapped the base performance through a simple saturation function:

$$P_{\text{total}}(N, E) = P_{\max}\left(1 - \exp\left(-\frac{P_{\text{base}}(N, E)}{P_{\max}}\right)\right), \tag{12}$$

where $P_{\max}$ is a normalized saturation level for the considered technology and workload class. In this formulation, $P_{\text{total}}(N, E)$ denotes the effective saturated system performance; $P_{\max}$ is the normalized saturation level for the given technology and workload.

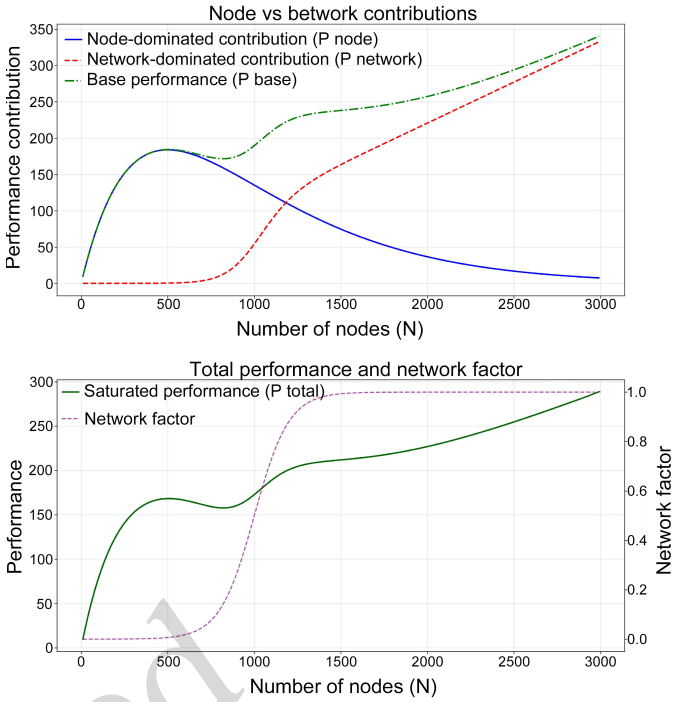

Fig. 5 Simulation results of the comprehensive performance scaling model

In this formulation, $P_{\text{node}}(N)$ captures local computation benefits and diminishing returns, $P_{\text{network}}(N, E)$ captures the network contribution modulated by $\eta_{\text{int}}(N, E)$ and F_{SDI}, and $P_{\text{total}}(N, E)$ represents the final system-level performance after saturation. For small systems, η_{int} and network_factor suppress the network term and P_{total} follows the node-dominated curve. As N increases toward $N_{\text{threshold}}$, both network effects and η_{int} become more important, and SDI flexibility further amplifies performance, yielding a regime where node and network contributions are balanced. For large systems, the saturation term causes P_{total} to exhibit diminishing returns.

The performance-scaling model integrates node-dominated and network-dominated contributions to describe how overall system performance P_{total} evolves with the number of nodes N and the interconnection efficiency E. The node term $P_{\text{node}}(N) = N\, Exp(-k_1 N)$ captures diminishing returns of local computation as the system scales, while the network term $P_{\text{network}}(N, E)$ accounts for the impact of interconnect efficiency, SDI flexibility, and network phase transitions. These contributions are combined into a saturating response $P_{\text{total}}(N, E) = P_{\max}\left[1 - \exp\left(-P_{\text{base}}(N, E)/P_{\max}\right)\right]$, which naturally limits achievable performance as resources grow. As shown in Fig. 5, simulation results reveal a phase transition near $N \approx 10^3$, where the system shifts from a node-dominated regime—where adding nodes primarily boosts compute throughput—to a network-dominated regime in which interconnect constraints become the primary bottleneck, thus defining a critical scale for architectural optimization.

4.3.3 Energy efficiency and optimization models

Performance alone is insufficient for SDSoW design; energy efficiency is equally critical. We therefore combined the performance model $P_{\text{total}}(N, E)$ with a simple power model to

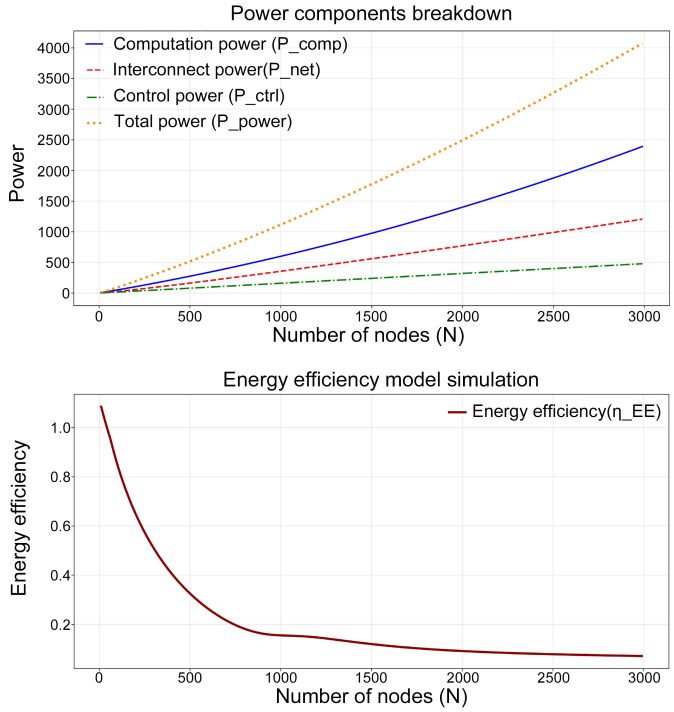

Fig. 6 Simulation results of energy efficiency and optimization model

obtain a normalized energy-efficiency factor $\eta_{\mathrm{EE}}(N,E)$, consistent with the energy-efficiency metric introduced earlier.

We modeled the total power consumption as

$$P_{\mathrm{power}}(N,E) = P_{\mathrm{comp}}(N) + P_{\mathrm{net}}(N,E) + P_{\mathrm{ctrl}}(N,E), \quad (13)$$

where $P_{\mathrm{power}}(N,E)$ denotes the total power consumption; $P_{\mathrm{comp}}(N)$ represents the computation-related power; $P_{\mathrm{net}}(N,E)$ corresponds to the interconnect power; $P_{\mathrm{ctrl}}(N,E)$ accounts for control and SDI management power. The computation-related power is modeled as

$$P_{\mathrm{comp}}(N) = a_1 N + a_2 N^2, \quad (14)$$

The interconnect power is modeled as

$$P_{\mathrm{net}}(N,E) = b_1 \frac{E}{\eta_{\mathrm{int}}(N,E)}, \quad (15)$$

The control and SDI management power is modeled as

$$P_{\mathrm{ctrl}}(N,E) = c_1 F_{\mathrm{SDI}} N, \quad (16)$$

where a_1, a_2, b_1, and c_1 are calibration coefficients. The factor $\eta_{\mathrm{int}}(N,E)$ appears in the denominator of P_{net} to reflect that, for a fixed traffic load, a more efficient interconnect (higher η_{int}) uses fewer hops and incurs less retransmission, thereby reducing power.

We then defined a normalized energy-efficiency metric as

$$\eta_{\mathrm{EE}}(N,E) = \frac{P_{\mathrm{total}}(N,E)}{P_{\mathrm{power}}(N,E)}, \quad (17)$$

which is consistent with the earlier definition of performance per energy. Design-space exploration can then search over (N,E), topology, and SDI configuration to maximize $\eta_{\mathrm{EE}}(N,E)$, revealing energy-optimal operating points that balance performance gains against power costs.

As shown in Fig. 6, η_{EE} decreases monotonically with the system size N. This behavior is driven by the power model of Eqs. (14)–(16): the compute power $P_{\mathrm{comp}}(N)$ contains a quadratic term a_2N^2 reflecting coordination and memory-access overhead, while the interconnect power $P_{\mathrm{net}}(N,E)$ grows super-linearly because η_{int} declines with scale. Together, these cause P_{power} to outpace P_{total} at all simulated scales, yielding a monotonically decreasing η_{EE}. The rate of decline is initially steep but gradually diminishes at larger N. This trend highlights the value of SDI-enabled dynamic topology reconfiguration: by improving η_{int} through adaptive reduction of hop counts and better load balancing, SDI can lower P_{net} and thereby mitigate the decline in energy efficiency at larger scales.

4.3.4 Emergent intelligence threshold modeling

Finally, we connected the previous models to emergent intelligence by defining a threshold model that depends on scale, interconnect efficiency, and energy efficiency. The underlying intuition is that high-level emergent behavior requires sufficient scale, sufficiently effective communication, and sustainable energy efficiency; large but inefficient systems are unlikely to sustain complex, long-horizon behavior.

We defined an emergent capability score $S_{\mathrm{em}}(N,E)$ as a weighted combination of the key factors:

$$\begin{aligned} S_{\mathrm{em}}(N,E) = {} & \gamma_1 \log(N+1) + \gamma_2 \log\bigl(1 + d_{\mathrm{avg}}(N,E)\bigr) \\ & + \gamma_3 \eta_{\mathrm{int}}(N,E) + \gamma_4 \eta_{\mathrm{EE}}(N,E) + \gamma_5 F_{\mathrm{SDI}}, \end{aligned} \quad (18)$$

where $S_{\mathrm{em}}(N,E)$ is the emergent-capability score; $d_{\mathrm{avg}}(N,E) \approx 2E/N$ is the average node degree; $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5$ are tunable weights for scale, connectivity, interconnect efficiency, energy efficiency, and flexibility, respectively.

The emergent intelligence factor is then modeled using a logistic threshold function:

$$F_{\mathrm{em}}(N,E) = \frac{1}{1 + \exp\bigl(-(S_{\mathrm{em}}(N,E) - \theta_{\mathrm{em}})\bigr)}, \quad (19)$$

where θ_{em} is the emergent-intelligence threshold. When systems are small, poorly connected, or energy-inefficient, $S_{\mathrm{em}}(N,E)$ lies well below θ_{em} and $F_{\mathrm{em}} \approx 0$; as N, d_{avg}, η_{int}, η_{EE}, and F_{SDI} jointly increase, $S_{\mathrm{em}}(N,E)$ crosses the threshold and F_{em} transitions rapidly toward 1.

This model explicitly reuses the outputs of the preceding models – $\eta_{\mathrm{int}}(N,E)$ from the interconnect efficiency model and $\eta_{\mathrm{EE}}(N,E)$ from the energy-efficiency model – to estimate when SDSoW systems are likely to exhibit qualitatively new emergent capabilities. The emergent-intelligence model introduces a macroscopic metric F_{em} to describe the emergence of higher-level intelligent behaviors in the SDSoW as a function of system scale, connectivity, and underlying efficiency metrics.

As shown in Fig. 7, the emergence metric F_{em} exhibits a clear sigmoid transition with increasing system size N. At small scales, $F_{\mathrm{em}} \approx 0$, confirming that a minimum number of computing nodes is necessary for emergent intelligence—consistent with the physical requirement that collective behavior cannot arise from an empty or trivially small system. The emergence threshold ($F_{\mathrm{em}} = 0.5$) occurs at $N \approx 500$ ($S_{\mathrm{em}} \approx 51.11$), where the cumulative contributions of the $\gamma_1 \log(N+1)$ scale term and the growing P_{total} begin

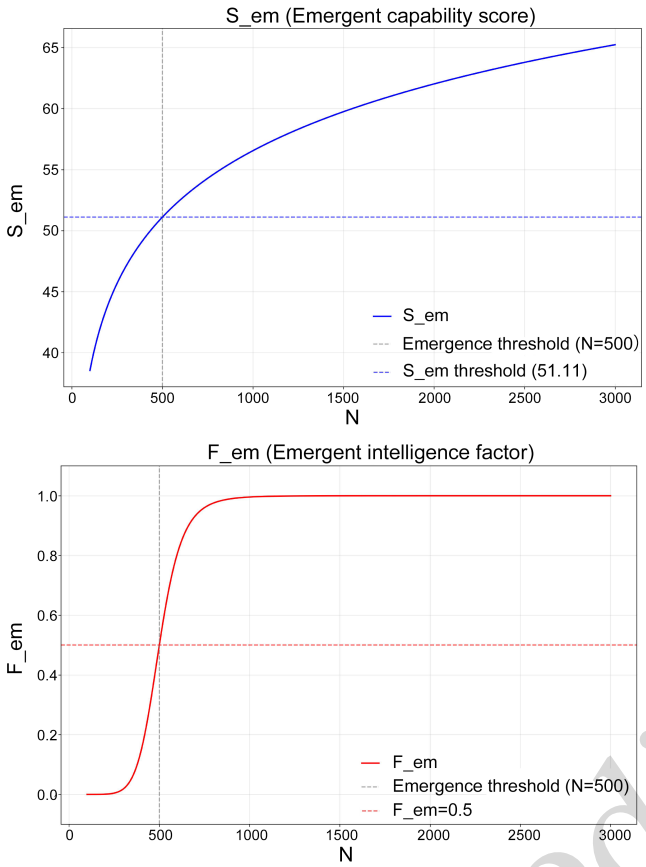

Fig. 7 simulation results of intelligence emergence threshold model

to dominate over the declining η_{int} and η_{EE} terms (Figs 4 and 6), pushing S_{em} through the sigmoid transition. Beyond $N \approx 500$, F_{em} rises steeply and gradually saturates toward unity at larger scales. This demonstrates that intelligence emergence in SDSoW systems is governed by a threshold mechanism: the system must accumulate sufficient computational scale and interconnect resources before collective intelligent behavior can manifest, after which further scaling yields diminish marginal returns.

5 Case study and simulation-based validation

This section uses a representative SDSoW-based wafer-scale accelerator to validate the analytical models from Subsection 4.3 and to quantify the performance and energy-efficiency benefits of SDSoW over conventional multi-chip and chiplet-based systems. Figs 8 and 9 report results for three scales and three architectures and empirically support the four-model hierarchy (η_{int}, P_{total}, η_{EE}, F_{em}).

5.1 System and workload setup

We modeled a wafer-scale accelerator organized as an SDSoW fabric with hundreds to over one thousand heterogeneous dielets connected by a SDI. Three system sizes are evaluated:

1. Small: $N = 128$ nodes
2. Medium: $N = 512$ nodes
3. Large: $N = 1024$ nodes

Each node combines a computer dielet with local memory. The workload is a communication-intensive parallel application such as deep neural network training or large-scale graph analytics, with a mix of (i) global collectives (all-reduce, all-gather), (ii) point-to-point and neighborhood exchanges, and (iii) control and metadata traffic from the software-defined management layer. This stresses both bisection bandwidth and local connectivity.

To isolate the impact of software-defined wafer-scale interconnects, we instantiate three architectures over identical computing and memory resources.

1. Baseline-A: conventional multi–chip system. Boards are connected via a fixed hierarchical or PCIe-like fabric, and chips on each board use a static on-board interconnect (e.g., a 2D mesh). Topology and routing are fixed and application-agnostic, with no runtime reconfiguration.

2. Baseline-B: chiplet/NoC system with limited reconfigurability. Each chiplet implements a 2D-mesh network-on-chip (NoC). Chiplets are linked by fixed express links and a static backplane. Only coarse–grained configuration choices (e.g., a few predefined link patterns) are available; there is no fine-grained, application-driven dynamic reconfiguration at system scale.

3. SDSoW: software-defined system-on-wafer. All dielets are connected by a wafer-scale SDI fabric that is application-aware and supports ADR. At runtime, the SDSoW fabric adapts topology, routing, QoS, and link usage to current communication patterns, reconfiguring at fine time granularity and using topology templates optimized for collective-dominated versus point-to-point–dominated phases. Beneficial changes are accumulated across phases.

The four analytical models from Section 4.3 are instantiated on this setup.

The interconnect-efficiency model decomposes $\eta_{\text{int}}(N, E)$ into topological, load-balancing, and QoS factors using average hop count, link-utilization statistics, and latency/QoS metrics.

The performance-scaling model combines a node term $P_{\text{node}}(N) = N \exp(-k_1 N)$ with a network term $P_{\text{network}}(N, E)$ driven by η_{int} and SDI flexibility and maps their sum P_{base} to P_{total} via a saturating function.

The energy-efficiency model decomposes power into compute, interconnect, and control components and defines $\eta_{\text{EE}}(N, E) = \frac{P_{\text{total}}(N,E)}{P_{\text{power}}(N,E)}$.

The emergent-intelligence model aggregates scale, connectivity, η_{int}, η_{EE}, and the SDI flexibility index F_{SDI} into a composite score, then applies a threshold-like nonlinearity to obtain $F_{\text{em}}(N, E) \in [0, 1]$. Model parameters are held constant across architectures and scales except for F_{SDI}, which is highest for SDSoW and lowest for Baseline-A.

5.2 Simulation framework

We used a trace-driven simulation environment that couples a configurable wafer-scale interconnect model with the workload. For each (architecture, N) combination, packet-level simulation models routing, buffering, arbitration, and SDI control, and reports latency, congestion, and link utilization.

The validation workflow has three steps. First, we perform network-level simulations for all three architectures at N

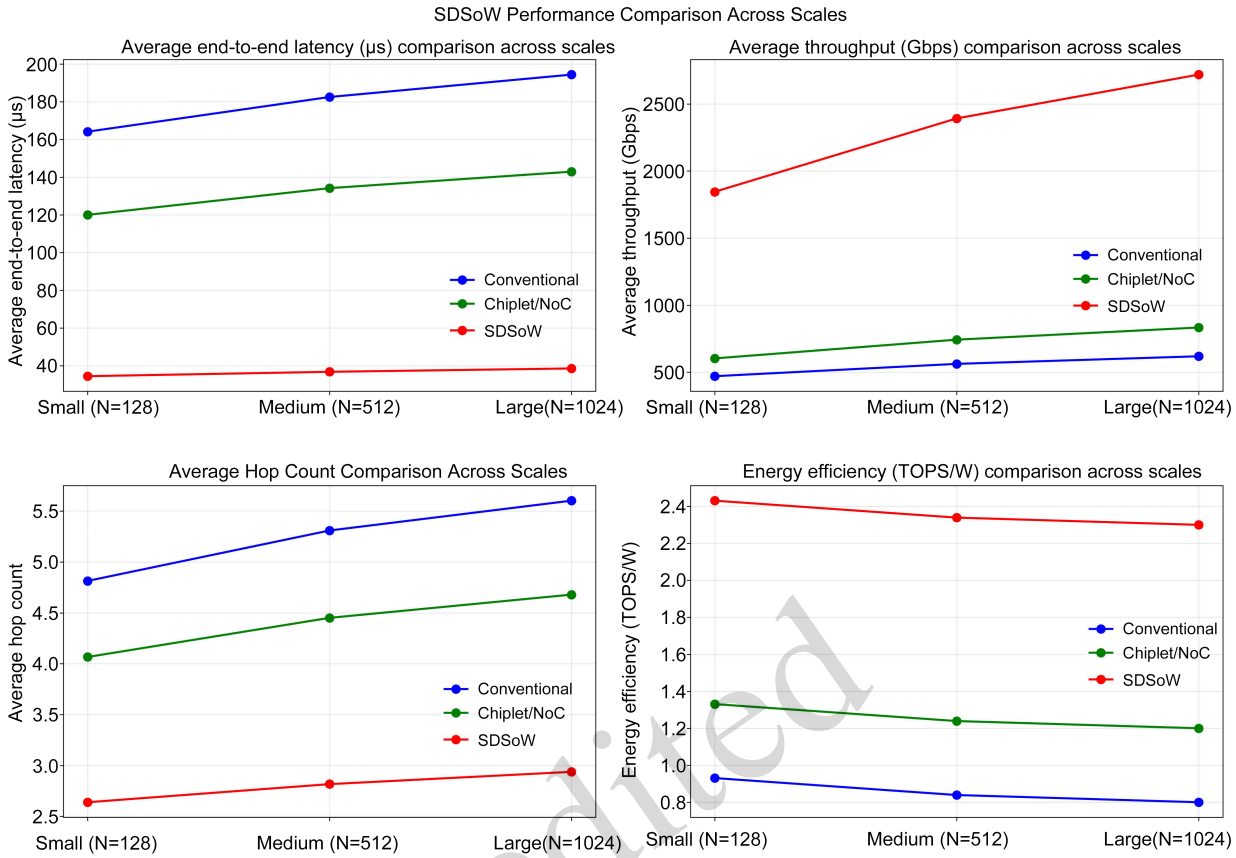

Fig. 8 Performance comparison of SDSoW with other architectures at different scales.

= 128 512, and 1024. Second, we extracted average end-to-end latency, aggregate throughput, average hop count, per-link utilization, total power and its breakdown into compute, interconnect, and control (SDI/ADR) components, application-level throughput (e.g., images/s or tokens/s), and energy efficiency (TOPS/W). Third, we mapped these metrics to the analytical quantities: η_{topo} from average hop count, η_{load} from the standard deviation of link utilization, η_{qos} from latency and QoS statistics, P_{total} from normalized application throughput, η_{EE} from normalized performance-per-watt, and F_{em} from the composite of scale, connectivity, η_{int}, η_{EE}, and F_{SDI}.

Figs 8 and 9 summarize the resulting performance comparisons and model-validation results. Fig. 8 reports normalized throughput, latency, and energy efficiency; Fig. 9 overlays simulated metrics and analytical predictions of η_{int}, P_{total}, η_{EE}, and F_{em}.

5.3 Results and analytical-model validation

5.3.1 Cross-scale architectural comparison

Fig. 8 shows normalized throughput, latency, and energy efficiency for the three architectures at N = 128 512, and 1024, with all metrics normalized to the conventional multi-chip baseline at each scale. The trends are consistent:

1. Throughput: SDSoW achieves about 3.4–3.7× higher application throughput than the conventional multi–chip system, with the chiplet/NoC architecture lying in between. The SDSoW advantage grows slightly from $N = 128$ to $N = 1024$.

2. Latency: SDSoW reduces average end-to-end latency by about 79% – 80% relative to the conventional baseline (normalized latency ≈ 0.20 – 0.21). The chiplet/NoC architecture again lies between SDSoW and the conventional system.

3. Hop count: Compared with the conventional multi–chip system, SDSoW reduces the average hop count by approximately 46%, and compared with the chiplet/NoC design by approximately 36%. This improvement stems from SDI-enabled dynamic topology reconfiguration, which adaptively shortens communication paths by restructuring the interconnect fabric in response to workload traffic patterns.

4. Energy efficiency: SDSoW improves energy efficiency by roughly 2.5× over the conventional multi-chip architecture at all three scales. The chiplet/NoC design reaches around 1.2–1.3×.

These results show that SDSoW's throughput and energy-efficiency benefits are already significant at N = 128 and strengthen at N = 512 and 1024, while latency reductions remain high across scales.

5.3.2 Validation of the four analytical models

Fig. 9 compares η_{int},P_{total}, η_{EE}, and F_{em} with the corresponding simulation-derived metrics for all architectures and scales.

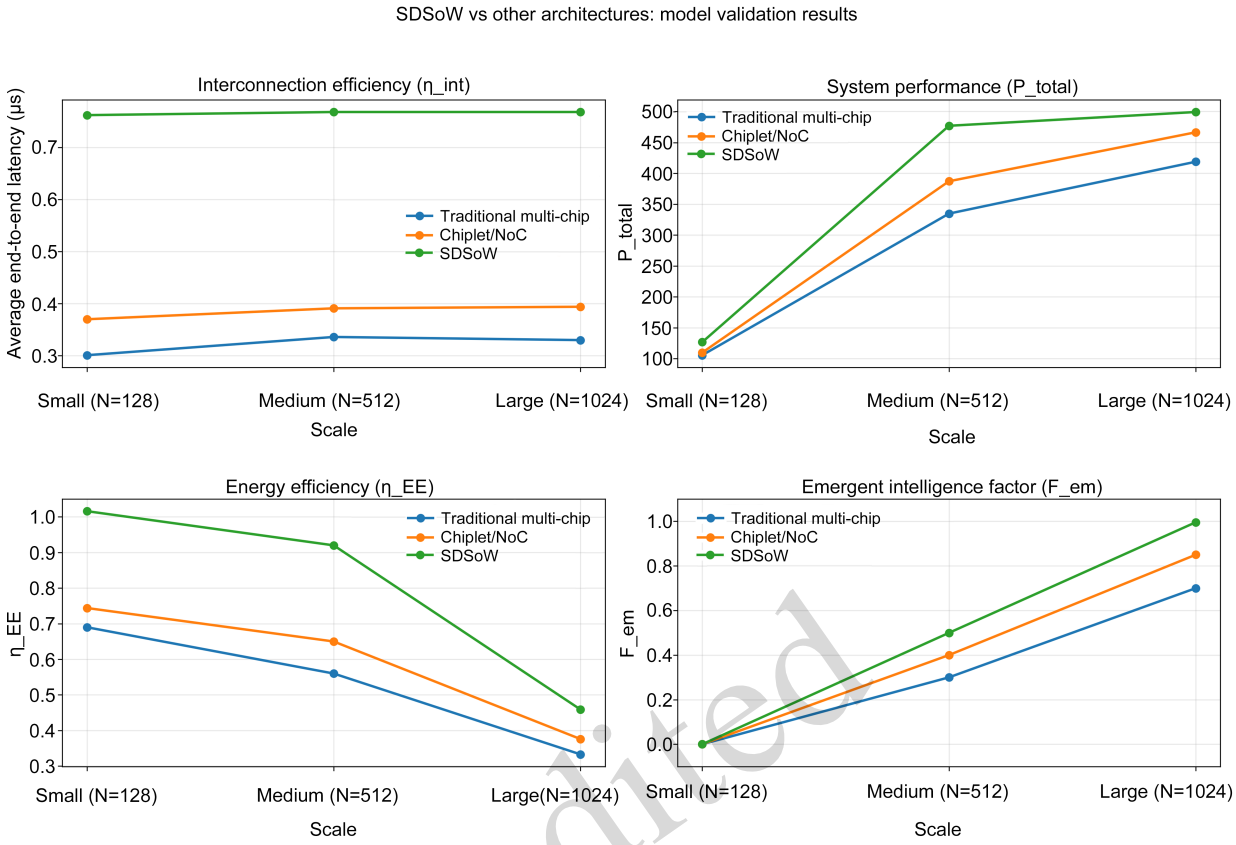

Fig. 9 Model validation results of SDSoW and other architectures under different scales.

For the interconnect-efficiency model, analytical $\eta_{\text{int}}(N, E)$ plotted against delivered bandwidth normalized by aggregate raw link bandwidth. SDSoW points occupy the high-η_{int} region, conventional multi–chip points the low-η_{int} region, with chiplet/NoC in between. Data points lie close to a smooth monotonic curve, indicating that η_{int} captures link-utilization effectiveness.

For the performance-scaling model, $P_{\text{total}}(N, E)$ is plotted against normalized application throughput. Across all scales, P_{total} preserves the ordering SDSoW $>$ chiplet/NoC $>$ conventional and reproduces the widening gap as N increases, showing that the combined node/network formulation and saturating mapping are sufficient to capture the main throughput trends without detailed curve fitting.

For the energy-efficiency model, analytical $\eta_{EE}(N, E)$is compared with simulated performance-per-watt. SDSoW consistently lies in the high-η_{EE} region despite extra SDI/ADR control power, while the other architectures remain at lower η_{EE}. The tight clustering near a monotonic curve supports the decomposition of power into compute, interconnect, and control components.

For the emergent-intelligence model (N, E), at $N = 512$, SDSoW achieves $F_{\text{em}} = 0.5$ (emergence threshold), while Chiplet/NoC reaches $F_{\text{em}} = 0.4$ and traditional multi-chip attains $F_{\text{em}} = 0.3$. At $N = 1024$, SDSoW approaches full emergence ($F_{\text{em}} = 0.996$), significantly outperforming Chiplet/NoC ($F_{\text{em}} = 0.850$) and traditional architecture ($F_{\text{em}} = 0.7$). All three architectures exhibit sigmoid growth, with SDSoW demonstrating the steepest gradient in the $N = 500 - 800$ interval. SDSoW's superior emergence profile stems from its dynamic reconfiguration capability and software-defined interconnection, enabling rapid intelligence manifestation at moderate scales.

Overall, Fig. 9 shows that η_{int} tracks utilization and congestion, P_{total} tracks throughput scaling, η_{EE} tracks performance-per-watt, and F_{em} tracks capability growth with scale and flexibility.

5.4 Summary

The case study based on Figs 8 and 9 demonstrates that the SDSoW analytical models from Section 4.3 are quantitatively supported by end-to-end system behavior across three scales. The interconnect-efficiency, performance-scaling, energy-efficiency, and emergent-intelligence models all correlate well with simulation metrics and jointly provide a coherent view from physical links to emergent capabilities.

Across $N = 128$, 512, and 1024 nodes, SDSoW delivers roughly 3.4–3.7× higher throughput, 79% – 80% lower latency, about 2.5× higher energy efficiency, and 2.3–2.4× higher interconnect efficiency than a conventional multi–chip system and consistently outperforms a chiplet/NoC design with limited reconfigurability. At $N = 1024$, the SDSoW emergent-intelligence factor reaches a value of approximately 0.996, indicating operation near the predicted emergent-capability regime. These results validate both the SDSoW

architectural concept—SDI plus application-driven dynamic reconfiguration—and the proposed analytical models as practical tools for SDSoW design, optimization, and intelligence engineering.

6 Conclusions

The proposed SDSoW framework is intentionally ambitious: it extrapolates the SDI concept to the wafer scale and combines it with heterogeneous integration and advanced interconnect technologies. While our analytical models and qualitative reasoning suggest substantial potential for performance scaling and energy efficiency improvements, several important limitations and practical challenges must be acknowledged.

Flexibility versus cost and complexity. As discussed in Section 2.2.3, increasing architectural flexibility along protocol, topology, QoS, and reconfiguration dimensions inevitably incurs area, power, timing, design-time, and verification overheads. Commercial designs must carefully balance these costs against workload diversity and lifetime requirements. In many application domains, a moderate level of flexibility tailored to a specific class of workloads may offer a more attractive cost–performance point than an extremely high F_{SDI}.

Manufacturing, yield, and thermal constraints. Wafer-scale systems significantly amplify traditional concerns about defect density, process variation, and thermal hotspots. Our integration scaling factors and complexity models (Eqs. (2) and (4)) highlight the theoretical potential of SDSoW, but they do not fully capture practical issues such as redundancy design, repair granularity, test coverage, and cooling infrastructure. Real-world SDSoW implementations will need hierarchical redundancy, fine-grained power gating, and advanced packaging solutions to maintain acceptable yield and reliability.

Modeling assumptions and validation gaps. Several of our equations—including the flexibility index, interconnect efficiency, and emergent intelligence probability—are necessarily based on simplifying assumptions and approximate parameterization from existing SoC and chiplet-based prototypes. They are primarily intended as design-intuition tools rather than predictive models with guaranteed accuracy. A key direction for future work is to validate and refine these models using detailed simulation frameworks and, eventually, measurements from physical SDSoW prototypes.

Near-term deployment scenarios. Despite these challenges, we believe that SDSoW is most likely to first appear in domains where extreme integration and reconfigurability can amortize their costs: hyperscale AI training accelerators, tightly coupled HPC systems, and domain-specific platforms for scientific computing or large-scale graph analytics. In such environments, the benefits of wafer-scale integration—reduced off-chip communication, higher bandwidth density, and global resource pooling—can outweigh the additional engineering effort.

Overall, SDSoW should be viewed not as an immediate replacement for conventional SoCs and multi–chip modules, but as a complementary architectural direction for the post-Moore era. The conceptual and mathematical framework developed in this work provides a starting point for systematically exploring this design space. Continued progress in heterogeneous integration technologies, interconnect materials, and software-defined control will be essential to transform this vision into deployable systems.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2023YFB4404200).

Author contributions

Ping LV designed the experimental design, participated in data analysis and interpretation of results, and drafted the paper. Qinrang LIU and Jiangxing WU constructed the theoretical model and conducted a comprehensive review of the literature. Jianliang SHEN conducted the experiments and collected the data. Mengke LIAN organized the theoretical knowledge into mathematical formulas. The rest of authors drafted and revised the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Declaration on the use of generative AI tools

During the preparation of this work, the authors used generative AI tools to improve the language and readability. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Attig M BG, 2011. 400 gb/s programmable packet parsing on a single fpga. ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems, p.12-23. https://doi.org/10.1109/ANCS.2011.12

Balandin AA, 2020. Phononics of graphene and related materials. *ACS Nano*, 14(5):5170-5178. https://doi.org/https://doi.org/10.1021/acsnano.0c02718

Behnam A, Sangwan VK, Zhong XY, et al., 2013. High-field transport and thermal reliability of sorted carbon nanotube network devices. *ACS Nano*, 7(1):482-490. https://doi.org/10.1021/nn304570u

Berestizshevsky K, Even G, Fais Y, et al., 2017. SDNoC: Software defined network on a chip. *Microprocess Microsyst*, 50:138-153. https://doi.org/10.1016/j.micpro.2017.03.005

Chen KN, 2023. Hybrid bonding: The key technology to reach fine pitch and high density stacking in heterogeneous integration. Int VLSI Symposium on Technology, Systems and Applications, p.1. https://doi.org/10.1109/VLSI-TSA/VLSI-DAT57221.2023.10134263

Chew SA, De Vos J, Beyne E, 2024. Wafer-to-wafer hybrid bonding at 400-nm interconnect pitch. *Nat Rev Electr Eng*, 1(2):71-72. https://doi.org/10.1038/s44287-024-00019-8

Deshmukh PK, Mane DT, 2023. QoS-aware routing and resource allocation techniques for enhanced network performance. *J Electr Syst*, 19(2):78-86. https://doi.org/10.52783/jes.693

Gomez-Rodriguez JR, Sandoval-Arechiga R, Ibarra-Delgado S, et al., 2021. A survey of software-defined networks-on-chip: Motivations, challenges and opportunities. *Micromachines*, 12(2):183. https://doi.org/10.3390/mi12020183

Gonzalez-Martinez G, Sandoval-Arechiga R, Solis-Sanchez LO, et al., 2024. A survey of MPSoC management toward self-awareness. *Micromachines*, 15(5):577. https://doi.org/10.3390/mi15050577

Goossens K, Koedam M, Nelson A, et al., 2017. NoC-based multiprocessor architecture for mixed-time-criticality applications. In: Ha S, Teich J (Eds.), Handbook of Hardware/Software Codesign. Springer, Dordrecht, p.491-530. https://doi.org/10.1007/978-94-017-7267-9_17

Hall S, Schreiber R, Lie S, 2021. Training giant neural networks using weight streaming on cerebras wafer-scale systems. Tech Rep.

Ham TJ, Wu LS, Sundaram N, et al., 2016. Graphicionado: A high-performance and energy-efficient accelerator for graph analytics. 49th Annual IEEE/ACM Int Symposium on Microarchitecture, p.1-13. https://doi.org/10.1109/micro.2016.7783759

Huang SH, Waeijen L, Corporaal H, 2022. How flexible is your computing system? *ACM Trans Embed Comput Syst*, 21(4):37. https://doi.org/10.1145/3524861

Ji N, Zhou XF, Yang YT, 2023. A high-performance fully adaptive routing based on software defined network-on-chip. *Microelectron J*, 141:105950. https://doi.org/10.1016/j.mejo.2023.105950

Kerrison S, May D, Eder K, 2016. A Benes based NoC switching architecture for mixed criticality embedded systems. IEEE 10th Int Symposium on Embedded Multicore/Many-Core Systems-on-Chip, p.125-132. https://doi.org/10.1109/mcsoc.2016.50

Khan HN, Hounshell DA, Fuchs ERH, 2018. Science and research policy at the end of moore's law. *Nat Electron*, 1(1):14-21. https://doi.org/10.1038/s41928-017-0005-9

Lau JH, 2022. Recent advances and trends in advanced packaging. *IEEE Trans Compon Packaging Manuf Technol*, 12(2):228-252. https://doi.org/10.1109/TCPMT.2022.3144461

Lee CY, Won CH, Jung S, et al., 2025. 3D integrated process and hybrid bonding of high bandwidth memory (HBM). *Electron Mater Lett*, 21(3):395-419. https://doi.org/10.1007/S13391-025-00557-9

Leiserson CE, Thompson NC, Emer JS, et al., 2020. There's plenty of room at the Top: What will drive computer performance after Moore's law? *Science*, 368(6495):eaam9744. https://doi.org/10.1126/science.aam9744

Lv P, Liu QR, Wu JX, et al., 2018. New generation software-defined architecture. *Sci Sin Inform*, 48(3):315-328 ((in Chinese)). https://doi.org/10.1360/n112017-00204

Ma XH, Wang Y, Wang YJ, et al., 2022. Survey on chiplets: Interface, interconnect and integration methodology. *CCF Trans High Perform Comput*, 4(1):43-52. https://doi.org/10.1007/s42514-022-00093-0

Mesquita DG, Rosa PF, 2017. Reconfigurable computing and future internet: Considerations on flexibility and security. *IEEE Latin Am Trans*, 15(7):1326-1334. https://doi.org/10.1109/TLA.2017.7959514

Miller DAB, 2017. Meshing optics with applications. *Nat Photonics*, 11(7):403-404. https://doi.org/10.1038/nphoton.2017.104

Mukkara A, Beckmann N, Abeydeera M, et al., 2018. Exploiting locality in graph analytics through hardware-accelerated traversal scheduling. 51st Annual IEEE/ACM Int Symposium on Microarchitecture, p.1-14. https://doi.org/10.1109/micro.2018.00010

Pal S, 2021. Scale-out packageless processing. University of California, Los Angeles, USA.

Prashanth A, N V, 2024. An adaptive software defined network-on-chip (SD-NoC) for varying network resources in offering optimal service quality: Survey. 1st Int Conf on Sustainability and Technological Advancements in Engineering Domain, p.226-231. https://doi.org/10.1109/SUSTAINED63638.2024.11074123

Radamson HH, Zhu HL, Wu ZH, et al., 2020. State of the art and future perspectives in advanced CMOS technology. *Nanomaterials*, 10(8):1555. https://doi.org/10.3390/nano10081555

Rudolph C, Hanisch A, Voigtländer M, et al., 2021. Enabling D2W/D2D hybrid bonding on manufacturing equipment based on simulated process parameters. IEEE 71st Electronic Components and Technology Conf, p.40-44. https://doi.org/10.1109/ECTC32696.2021.00018

Salvador ID, Sandoval-Arechiga R, Brox M, et al., 2017. Software defined network controller: A neat solution administration for reconfigurable multi-core NoC. Int Conf on ReConFigurable Computing and FPGAs, p.1-4. https://doi.org/10.1109/reconfig.2017.8279821

Sandoval-Arechiga R, Vazquez-Avila JL, Parra-Michel R, et al., 2015. Shifting the network-on-chip paradigm towards a software defined network architecture. Int Conf on Computational Science and Computational Intelligence, p.869-870. https://doi.org/10.1109/csci.2015.45

Sandoval-Arechiga R, Ibarra-Delgado S, Flores-Troncoso J, 2017. A software defined interconnection architecture for systems on chip. *Difu100ci@ Revista en Ingeniería y Tecnología*, 10(2):2-11.

Shalf J, 2020. The future of computing beyond moore's law. *Philos Trans A Math Phys Eng Sci*, 378(2166):20190061. https://doi.org/10.1098/rsta.2019.0061

Sivaraman A, Kim C, Krishnamoorthy R, et al., 2011. DC.p4: Programming the forwarding plane of a data-center switch. Proc 1st ACM SIGCOMM Symposium on Software Defined Networking Research, p.1-8. https://doi.org/https://doi.org/10.1109/ANCS.2011.12

Sivaraman A, Subramanian S, Alizadeh M, et al., 2016. Programmable packet scheduling at line rate. Proc ACM SIGCOMM Conf, p.44-57. https://doi.org/10.1145/2934872.2934899

Systems C, 2019. Wafer-scale deep learning. IEEE Hot Chips 31 Symposium, p.1-31. https://doi.org/10.1109/HOTCHIPS.2019.8875628

Theis TN, Wong HSP, 2017. The end of moore's law: A new beginning for information technology. *Comput Sci Eng*, 19(2):41-50. https://doi.org/10.1109/MCSE.2017.29

Wu JX, Liu QR, Shen JL, et al., 2024. From SoC to SDSoW: a new paradigm for microelectronics development. *Sci Sin Inform*, 54(6):1350-1368 (in Chinese). https://doi.org/10.1360/SSI-2023-0219