

Original Article

A Hybrid Analytical-Machine Learning Framework for Performance and Energy Modelling of CPU-FPGA Heterogeneous Architectures

Sabbar Hadi Salh¹, Maher Shahatha Mahmood², Methaq Kadhun Hussain³, Nooraldeen R. Hamza⁴

¹Ministry of Higher Education and Scientific Research, Ministry Center, Baghdad, Iraq.

²Electronics and Communication Engineering, Al-Iraqia University, Baghdad, Iraq.

^{3,4}Electrical Engineering, Al Mustansiriyah University, Baghdad, Iraq.

³Corresponding Author: methaq@uomustansiriyah.edu.iq

Received: 19 May 2026

Revised: 10 September 2026

Accepted: 16 September 2026

Published: 27 October 2026

Abstract - Modern data centres increasingly employ heterogeneous computing platforms that combine Field-Programmable Gate Arrays (FPGAs) with general-purpose processors to improve performance and energy efficiency. The effectiveness of these platforms depends strongly on the level of coupling between the host CPU and the reconfigurable fabric. Existing approaches for evaluating CPU-FPGA systems often involve a trade-off between simplified analytical models and computationally expensive detailed simulations, limiting their suitability for large-scale early design-space exploration. This paper presents a hybrid analytical-machine-learning framework for performance and energy modelling of CPU-FPGA heterogeneous architectures. The framework introduces a workload-adaptive overlap function $\gamma(N)$ and a bus-contention factor κ to represent workload-dependent communication-computation overlap and communication contention. A dual-state energy model accounts for the active and idle power of both the CPU and FPGA. A Gradient-Boosted-Tree (GBT) surrogate, trained on samples generated by the analytical model, provides computationally efficient repeated evaluations for large-scale design-space exploration. An Adaptive Coupling Selector (ACS) further supports architecture selection by considering Energy-Delay Product (EDP) and normalized deployment cost in the (N, w, E) decision space. The framework evaluates three representative coupling topologies: External (EXT), Shared-Memory (SHM), and on-die Coprocessor (COP). Comparison with five published FPGA benchmarks yields Mean Absolute Percentage Errors (MAPE) of 4.4% for speed-up and 7.0% for total energy. At a representative workload of $N=10$ 8 operations, COP achieves a speed-up of $1.20\times$ over the CPU-only baseline, compared with $0.39\times$ for EXT, while reducing total energy consumption by approximately 57.7% relative to EXT under the evaluated baseline parameters. A 5,000-trial Monte-Carlo analysis further indicates that the relative performance and energy trends remain stable under the considered parameter perturbations. Overall, the framework provides a computationally efficient approach for early exploration of CPU-FPGA coupling alternatives under performance, energy, and deployment-cost considerations.

Keywords - CPU-FPGA heterogeneous computing, Coupling architectures, Energy efficiency, Machine-Learning surrogate, Design-Space exploration.

1. Introduction

The slowdown of Dennard scaling and Moore's law has made continued single-chip performance improvement increasingly difficult, particularly at advanced process nodes. Power-density and thermal constraints have contributed to the dark-silicon phenomenon, in which only a fraction of the available transistors can operate simultaneously within a given power budget [1, 2]. In response, heterogeneous computing platforms have emerged that combine general-purpose processors with domain-specific accelerators such as Graphics Processing Units (GPUs), Tensor Processing Units (TPUs), and Field-Programmable Gate Arrays (FPGAs) [3, 4]. FPGAs occupy an important position among these accelerators

because their reconfigurable fabric enables customized datapaths, high parallelism, and energy-efficient execution for workloads such as machine inference, packet processing, financial analytics, and bioinformatics [5, 6]. The performance and energy characteristics of a CPU-FPGA platform depend not only on the computational capability of the FPGA but also on the architectural interface between the host processor and the accelerator. This study considers three representative coupling levels: an External (EXT) topology, in which the FPGA communicates with the host through PCIe; a Shared-Memory (SHM) topology, in which the CPU and FPGA share access to memory; and an on-die Coprocessor (COP) topology, in which the accelerator is connected through



a tightly coupled coherent interconnect [7-10]. These coupling choices affect communication latency, available bandwidth, communication overhead, and link-energy consumption. Consequently, an accelerator with high computational throughput may still provide limited or even negative speed-up when communication overhead dominates the benefit of offloading.

1.1. Limitations of Models

Three main approaches are commonly used to evaluate CPU-FPGA platforms during architectural development: cycle-accurate simulation, analytical modelling, and FPGA-based emulation. Cycle-accurate simulators can provide detailed architectural information but may require substantial computational time for each design point, limiting their use in large-scale design-space exploration [11]. Analytical models are considerably faster, but existing formulations may rely on simplifying assumptions, such as a constant communication-computation overlap factor, simplified communication effects, or limited treatment of active and idle power states [12, 13]. FPGA-based emulation can provide measurements from physical implementations, but it generally requires a sufficiently developed hardware design and is therefore less suitable for early architectural exploration.

1.2. Contributions of this Work

This paper addresses the trade-off between modelling detail and computational efficiency by presenting a hybrid analytical-machine-learning framework for early design-space exploration of CPU-FPGA heterogeneous architectures. The main contributions are as follows.

1.1.1. Adaptive Overlap and Contention Model

A workload-dependent overlap factor $\gamma(N)$ is introduced to represent the variation of communication-computation overlap with workload size. A topology-specific contention factor κ is also incorporated to represent additional communication overhead associated with contention and queuing effects (Section 4).

1.1.2. Dual-State Energy Decomposition

The energy model is extended to distinguish between the active and idle power states of both the CPU and FPGA, providing a more detailed estimate of total energy consumption (Section 4.2).

1.1.3. Computational Surrogate for Design-Space Exploration

A Gradient-Boosted-Tree (GBT) model is trained on 2,500 samples generated by the analytical model to provide a computationally efficient surrogate for repeated evaluations during large-scale design-space exploration (Section 6.4).

1.1.4. Adaptive Coupling Selector (ACS)

An Adaptive Coupling Selector is developed to support architecture selection by jointly considering Energy-Delay Product (EDP) and normalized deployment cost. The resulting decision map illustrates how the preferred coupling topology varies with workload size N and the weighting factor w_E (Section 6.6).

1.1.5. Empirical Validation and Uncertainty Quantification:

The framework is compared with five published FPGA benchmarks, resulting in MAPE values of 4.4% for speed-up and 7.0% for total energy. In addition, a 5,000-trial Monte-Carlo analysis is used to examine the stability of the predicted performance and energy trends under parameter perturbations (Sections 6.7 and 6.8). The remainder of the paper is organized as follows. Section 2 provides an overview of heterogeneous CPU-FPGA platforms, while Section 3 reviews related work and identifies the research gap addressed in this study. Section 4 presents the analytical performance and energy models, and Section 5 describes the simulation methodology. Section 6 presents the performance and energy results, machine-learning surrogate evaluation, energy-time trade-off analysis, Adaptive Coupling Selector, uncertainty analysis, and comparison with published benchmarks. Section 7 discusses representative application domains, and Section 8 concludes the paper and outlines directions for future work.

2. Evaluation Criteria for Heterogeneous Architectures

The choice of a suitable heterogeneous architecture is a multi-criteria decision problem where hardware capabilities have to be aligned with the computational properties of the application to be implemented. Six commonly considered criteria in the literature [11, 14] are summarized in Table 1: performance, reliability, flexibility, cost, power, and tool-chain maturity. To support the quantitative analysis, we draw on the power- and cost-modelling approaches reported by Molnos et al. [15] and Mbakoyiannis et al. [16].

Table 1. Six-criterion evaluation framework for heterogeneous CPU-FPGA platforms (adapted from [11, 14])

Criterion	Dominant Factors
Performance	Memory bandwidth, interconnect latency, cache coherency, and computation/communication overlap.
Reliability	Memory protection, ECC, error monitoring, and fault tolerance under clock-domain mismatch.
Flexibility	External-memory flexibility, boot configurability, and configurability of IP-block interconnects.
Cost	Integration density, transceiver count and speed, power-supply complexity, and silicon area.
Power	Integration degree, sustained memory throughput, link active power, and idle-state efficiency.
Tools	Maturity of HLS flows, debugging support, OpenCL/Vitis libraries, and OS integration.

2.1. Performance

Performance is influenced by two interdependent factors: memory performance and interconnect performance. Memory performance depends on how effectively data can be transferred between the CPU and FPGA, including across clock-domain boundaries and through the memory controller. Interconnect performance between the FPGA data path and the processor depends mainly on end-to-end latency and available bandwidth. Tighter coupling can reduce round-trip latency compared with PCIe-attached topologies, which can improve performance for fine-grained workloads.

2.2. Reliability and Flexibility

Reliability can be improved when shared-memory regions are protected using Error-Correction Code (ECC) [17]. When the CPU memory controller and FPGA fabric operate in different clock domains, clock-domain crossings may introduce metastability risks and therefore require appropriate synchronization and error-monitoring mechanisms. Flexibility can be considered in terms of three aspects: (i) external-memory flexibility, which concerns whether the FPGA can operate as an external-memory peripheral; (ii) boot flexibility, which concerns whether the FPGA can support boot-related functions; and (iii) IP-block interconnect flexibility, which concerns the integration of the FPGA with internal interfaces such as AXI/AHB.

2.3. Cost and Power

Cost is mainly influenced by integration density, transceiver count and speed, and power-supply complexity [15]. On-die integration may involve higher non-recurring engineering costs, whereas PCIe-attached accelerator cards generally require less integration effort. The resulting per-unit cost depends on implementation scale and deployment requirements. Power consumption is influenced by the degree of integration, sustained memory throughput, link active power, and the idle-state efficiency of both the CPU and FPGA fabric.

3. Related Work

Studies of FPGA-accelerated heterogeneous computing have advanced along three orthogonal directions: acceleration of algorithms, acceleration of networks, and acceleration of data-access. We discuss the individual axes listed below and determine the gap that the current paper fills.

3.1. Algorithm Acceleration

Previous studies have shown that FPGAs can provide performance and energy-efficiency benefits for compute-intensive kernels. Chen and Singh [19] evaluated an information-filtering application on an Altera Stratix IV using OpenCL and reported performance and energy-efficiency benefits compared with a CPU baseline. FPGA implementations have also been investigated for sequence-alignment algorithms, with significant performance and

energy benefits [20]. Cryptographic applications, particularly AES, have been implemented using folded-parallel and fully pipelined FPGA architectures to achieve high throughput [21-24]. Although these studies demonstrate the potential benefits of FPGA acceleration, they generally focus on a specific platform or coupling configuration and do not provide a comparative analysis across multiple CPU-FPGA coupling topologies.

3.2. Network Acceleration

A second body of literature places the FPGA on the Network Interface Card (NIC) itself. The Mellanox ConnectX-3 Pro adapter provides programmable capabilities for offloading protocol-processing tasks from the host CPU [25]. The Solarflare Application-Onload Engine [26] integrates FPGA-based processing with a kernel-bypass network stack to support latency-sensitive network operations. Both approaches rely on PCIe communication between the accelerator and the host and therefore represent coupling characteristics similar to the EXT topology considered in this paper.

3.3. Data-Access Acceleration

A third stream of research focuses on data-access acceleration in heterogeneous systems. Blott et al. [27] developed a Key-Value Store (KVS) data path in RTL on an Altera Stratix V and achieved 10 Gb/s line-rate processing. Cong and Xiao [28] investigated cache-coherent CPU-FPGA heterogeneous systems and evaluated the performance benefits of shared-memory integration. Lee et al. [29] examined energy-efficient data movement in integrated CPU-FPGA systems. Tamimi et al. [10] studied cache-coherent interconnects for accelerators, including CCIX, and evaluated their performance characteristics. Chen et al. [30] investigated on-die FPGA integration and reported performance and energy benefits.

3.4. Identified Gap and Positioning of this Work

The studies reviewed above demonstrate the performance and energy benefits of FPGA acceleration across algorithm-processing, network-acceleration, and data-access applications. However, most of these studies focus on a specific FPGA platform, workload, or coupling configuration, which makes direct architectural comparison difficult. The contribution of the present work is therefore not the use of machine learning alone, since surrogate models are already established in design-space exploration, but the integration of performance, energy, communication, and coupling parameters within a common analytical framework. The proposed framework enables EXT, SHM, and COP configurations to be evaluated under the same workload and modelling assumptions. In contrast to studies that report results for a single implementation, this common formulation makes it possible to examine how communication latency, bandwidth, contention, workload size, arithmetic intensity, and offload fraction influence the relative benefits of different

coupling choices. The GBT surrogate is used only as a computational proxy for repeated evaluations of the analytical model, while predictive validity is assessed separately through comparison with published hardware benchmark results. The results obtained in this study should therefore not be interpreted as demonstrating that one CPU-FPGA topology universally outperforms previously reported implementations. Instead, the analysis identifies the conditions under which tighter coupling can provide performance and energy

advantages. In particular, reduced communication overhead and greater communication-computation overlap explain the improved results obtained with SHM and COP under the evaluated conditions, whereas the benefit of EXT depends more strongly on arithmetic intensity and interconnect capability. Table 2 summarizes the positioning of the proposed framework relative to representative CPU-FPGA studies and highlights the differences in evaluation scope and cross-topology exploration.

Table 2. Positioning of the proposed framework relative to representative CPU-FPGA studies

Study	Main focus	Coupling/platform considered	Main scope	Cross-topology exploration
Sano and Suda [6]	Algorithm acceleration	External FPGA platform	Throughput and energy efficiency	No
Mellanox Technologies [25]	Network acceleration	PCIe-attached programmable NIC	Protocol-processing offload	No
Morgan [26]	Network acceleration	PCIe-attached network accelerator	Low-latency network processing	No
Blott et al. [27]	Data-access acceleration	Stratix V FPGA	Key-value-store acceleration	No
Cong and Xiao [28]	Data-access / graph workloads	Shared-memory CPU-FPGA	Execution-time improvement	No
Kumar and Lee [10]	Coherent CPU-FPGA integration	CCIX-based system	Interconnect performance	No
Proposed framework	Architectural design-space exploration	EXT, SHM, and COP	Performance, energy, offload, uncertainty, and cost-aware selection	Yes

Table 2 highlights the main distinction between the proposed framework and the representative studies considered in this work. Previous studies provide valuable results for specific workloads or hardware configurations, whereas the present framework evaluates multiple coupling alternatives under a common set of modelling assumptions. The observed

advantages of tighter coupling arise primarily from reduced communication overhead and increased communication-computation overlap. However, the arithmetic-intensity and PCIe-bandwidth analyses in Section 6.1 show that these advantages are workload and interconnect-dependent rather than universal.

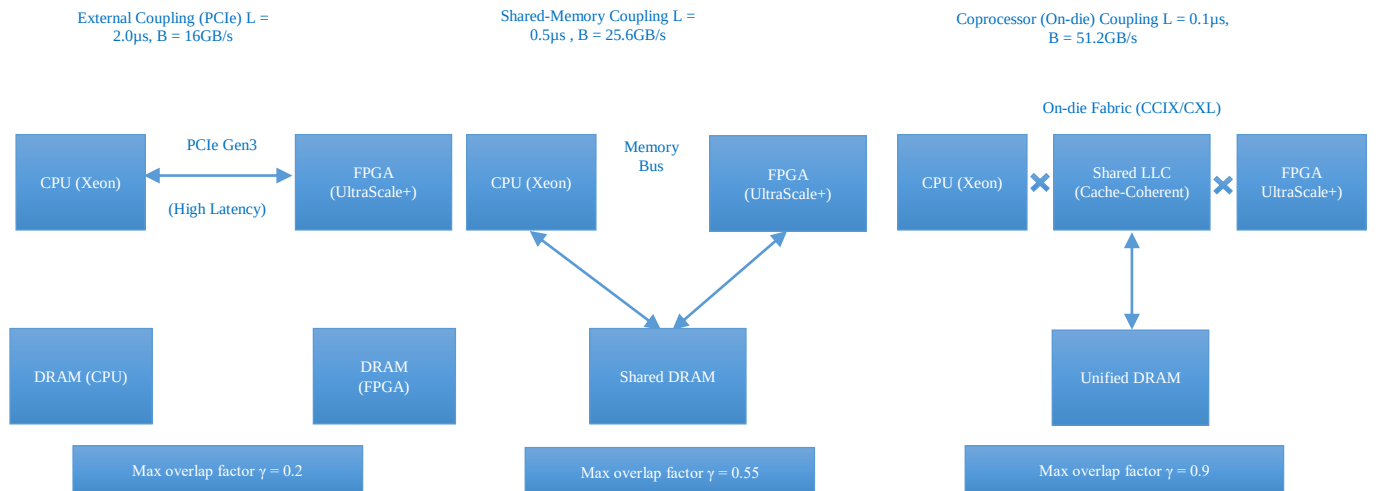


Fig. 1 Conceptual representation of the three CPU-FPGA coupling levels evaluated in this paper. (Left) External (EXT) topology in which the FPGA communicates with the host through a PCIe Gen3 link. (Centre) Shared-Memory (SHM) topology in which the CPU and FPGA share a DRAM channel. (Right) on-die Coprocessor (COP) topology in which the FPGA accesses the host's last-level cache through a coherent on-die fabric such as CCIX or CXL. The assumed maximum overlap factor $\gamma_{\max}$ increases with coupling tightness

4. Enhanced Analytical Model for CPU-FPGA

The analytical model generalizes the canonical roofline-style formulation in three ways: (i) the saturating nature of the overlap between communication and computation is captured by a workload-adaptive overlap factor, $\gamma(N)$; (ii) the dimensionless contention factor κ accounts for additional communication overhead caused by bus contention and queuing effects and is applied only to the communication-time component of the model; and (iii) the energy model separates the active and idle states of both the CPU and the FPGA, reducing the optimistic bias associated with single-state energy models. The symbols used in the model are defined in Table 3

4.1. Execution-Time Model

The overall execution time T_{total} consists of a communication component T_{comm} and a computation component, which is shared between the FPGA and the remaining CPU work.

Let N be the workload size (in operations), and let $\alpha \in [0,1]$ be the fraction of operations offloaded to the FPGA. The communication time with bus-contention effects is:

$$T_{comm} = \kappa \cdot (n \cdot L + (\alpha \cdot N \cdot Dop) / B) \quad (1)$$

Where κ is the architecture-specific dimensionless contention factor, n is the number of independent transfers, L is the link latency, B is the link bandwidth, and Dop is the average number of bytes transferred per operation. The topology-specific values of κ are treated as model parameters representing different levels of communication contention rather than as directly measured hardware constants.

Their influence on the model predictions is subsequently examined through sensitivity analysis. Arithmetic intensity is defined as $I = 1/Dop$, expressed in operations per byte. Therefore, the baseline value $Dop = 8$ bytes/operation corresponds to $I = 0.125$ operations/byte. In addition to this baseline case, Dop is varied in the subsequent analysis to evaluate workloads with different arithmetic intensities. The computation time of the FPGA is:

$$T_{fpga} = (\alpha \cdot N) / R_{FPGA} \quad (2)$$

The execution time of the remaining CPU workload is:

$$T_{cpu_rem} = ((1 - \alpha) \cdot N) / R_{CPU} \quad (3)$$

Where R_{FPGA} and R_{CPU} represent the sustained throughputs of the FPGA and CPU, respectively. The total execution time is calculated by accounting for the overlap between the communication-plus-FPGA path and the remaining CPU computation:

$$T_{total} = (T_{comm} + T_{fpga}) + T_{cpu_rem} - \gamma(N) \cdot \min(T_{comm} + T_{fpga}, T_{cpu_rem}) \quad (4)$$

Where $\gamma(N)$ is the workload-adaptive overlap factor defined below.

4.1.1. Workload-Adaptive Overlap Factor

Previous analytical models commonly represent the communication-computation overlap factor γ as a constant for each architecture.

In practice, however, the effective degree of overlap may vary with workload size because fixed initialization and communication overheads become less dominant as N increases. To represent this behavior, the overlap factor is modelled using the following one-parameter saturating function:

$$\gamma(N) = \gamma_{max} \cdot (1 - \exp(-N / N_{ref})) \quad (5)$$

Where γ_{max} represents the asymptotic overlap factor associated with each coupling topology, with values of 0.20, 0.55, and 0.90 for EXT, SHM, and COP, respectively.

The reference workload is set to $N_{ref} = 10^7$ operations as a baseline model parameter for the present analysis. As N increases, $\gamma(N)$ approaches γ_{max} , representing the saturation of achievable communication-computation overlap for each topology

4.2. Dual-State Energy Model

The total energy E_{total} is decomposed into the energy consumed by the CPU, FPGA, and communication link. Each of the CPU and the FPGA has active and idle power, as neither processor is in its peak power state over the entire T_{total} :

$$E_{CPU} = P_{CPU,act} \cdot T_{cpu_rem} + P_{CPU,idle} \cdot (T_{total} - T_{cpu_rem}) \quad (6)$$

$$E_{FPGA} = P_{FPGA,act} \cdot (T_{fpga} + T_{comm}) + P_{FPGA,idle} \cdot (T_{total} - T_{fpga} - T_{comm}) \quad (7)$$

$$E_{link} = P_{link} \cdot T_{comm} \quad (8)$$

$$E_{total} = E_{CPU} + E_{FPGA} + E_{link} \quad (9)$$

Accurate energy estimation requires distinguishing between active and idle states because neither the CPU nor the FPGA operates at peak power throughout the entire execution time. The dual-state model, therefore, accounts separately for active and idle energy contributions.

Table 3 collects the symbols used in Equations (1)-(9).

Table 3. Nomenclature for the enhanced analytical model. New parameters introduced in this work are marked.

Symbol	Description	Definition / Range
T _{total}	Total execution time	Computation + communication, after overlap
T _{comm}	Communication time	Time spent moving data over the bus
L	Link latency	0.1 μ s (COP) - 2.0 μ s (EXT)
B	Link bandwidth	16 GB/s (PCIe Gen3) - 51.2 GB/s (on-die)
κ	Bus-contention factor	1.05 (COP) - 1.30 (EXT)
$\gamma(N)$	Adaptive overlap factor	$\gamma_{\max} \cdot (1 - e^{-(N/N_{\text{ref}})})$
$\gamma_{\max}$	Asymptotic overlap	0.20, 0.55, 0.90 for EXT/SHM/COP
N	Workload size	10^5 - 10^9 operations
α	Offload fraction	[0, 1]; $\alpha = 0.6$ default
R _{CPU} / R _{FPGA}	Sustained throughput	5.0 / 10.0 Gops/s
D _{op}	Bytes per operation	8 bytes/operation (baseline; varied in the arithmetic-intensity analysis)
P _{*,act}	Active power (CPU/FPGA/link)	See Table 4
P _{*,idle}	Idle power: CPU and FPGA)	8.0 W (CPU) / 2.0 W (FPGA)
E _{total}	Total energy	E _{CPU} + E _{FPGA} + E _{link}

5. Simulation Methodology and Hardware Parameters

The analytical model defined by Equations (1)-(9) was implemented in Python 3.11 using NumPy 1.26 and SciPy 1.11. The Gradient-Boosted-Tree surrogate described in Section 6.4 was implemented using scikit-learn 1.4. To ensure reproducibility, the pseudo-random seeds for the Monte-Carlo experiment (seed = 123) and surrogate-training experiment (seed = 42) were fixed. The reported data points all took less than 200 μ s to compute on an Intel® Core i7-12700H. Individual model evaluations required less than 200 μ s on an Intel® Core i7-12700H, enabling large design sweeps to be completed substantially faster than would be practical with cycle-accurate simulation.

Table 4 contains a listing of the hardware parameters that are set as the default throughout the paper. Throughput figures of the Xeon Scalable family and the Xilinx UltraScale+ FPGA were obtained on the basis of manufacturer datasheets [32, 33]. The link-latency and bandwidth parameters were selected based on the PCI-SIG Base Specification 3.0 [34] and the ARM AMBA 5 CHI Architecture Specification [35]. Active

and link power values were obtained using published studies on power-modelling [16, 36]. The EXT link-power value of 5.0 W is retained as the baseline parameter associated with the reference configuration in [16]. It is not intended to represent the active link power of all modern PCIe generations, which may vary with PHY implementation, link width, data rate, and operating conditions.

Table 4. Hardware-parameter values used for the default simulation. New idle-power parameters are marked

Parameter	Symbol	Value	Source
Workload size (default)	N	10^8 ops	Baseline model assumption
CPU throughput	R _{CPU}	5.0 Gops/s	Intel Xeon Scalable [32]
FPGA throughput	R _{FPGA}	10.0 Gops/s	Xilinx UltraScale+ [33]
External latency (baseline: PCIe Gen3)	L _{EXT}	2.0 μ s	PCI-SIG [34]
Shared-memory latency	L _{SHM}	0.5 μ s	Cong & Xiao [28]
Coprocessor latency (on-die)	L _{COP}	0.1 μ s	AMBA CHI [35]
External bandwidth (baseline: PCIe Gen3)	B _{EXT}	16 GB/s	PCI-SIG [34]
DDR4-3200 bandwidth	B _{SHM}	25.6 GB/s	JEDEC DDR4
On-die fabric bandwidth	B _{COP}	51.2 GB/s	CCIX Consortium
CPU idle power	P _{CPU,idle}	8 W	Grot et al. [36]
FPGA active power	P _{FPGA,act}	15 W	Xilinx XPE [33]
FPGA idle power	P _{FPGA,idle}	2 W	Xilinx XPE [33]
Link power (EXT baseline)	P _{link,EXT}	5.0 W	Jalalee et al. [16]
Link power (SHM)	P _{link,SHM}	2.0 W	Grot et al. [36]
Link power (COP)	P _{link,COP}	0.5 W	On-die fabric [35]

6. Results and Discussion

6.1. Comparative Execution-Time Analysis

To examine whether the observed EXT deceleration is specific to the baseline workload assumptions, an arithmetic-intensity sensitivity analysis was performed. Arithmetic intensity I was varied while retaining the remaining EXT

parameters unchanged. The results show that the PCIe-attached topology transitions from deceleration to acceleration as arithmetic intensity increases. At $N=10^8$ operations, the break-even point occurs at approximately $I=0.64$ operations/byte. The baseline case, $I=0.125$ operations/byte,

therefore represents a communication-intensive operating regime rather than a general limitation of PCIe-attached FPGA acceleration. Figure 2 illustrates the resulting speed-up regions and the break-even boundary as functions of workload size and arithmetic intensity.

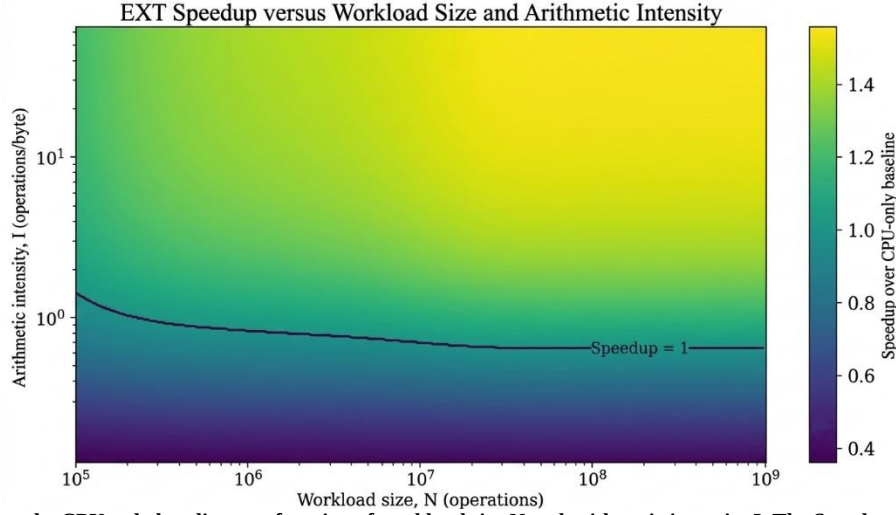


Fig. 2 EXT speed-up over the CPU-only baseline as a function of workload size N and arithmetic intensity I . The Speedup=1 contour indicates the break-even boundary between deceleration and acceleration.

As shown in Figure 2, the EXT topology is not inherently associated with deceleration. The transition to acceleration depends on both workload size and arithmetic intensity, with the break-even arithmetic intensity approaching approximately 0.64 operations/byte for large workloads. The execution time and corresponding speed-up of the three coupling topologies are shown in Figure 3 as the workload size N is varied from 10^5 to 10^9 operations. At the representative workload of $N=10^8$ operations, the CPU-only baseline requires 20 ms. Under the baseline parameters, COP achieves

a speed-up of $1.20\times$, whereas EXT results in a slowdown of approximately $2.6\times$. This EXT slowdown should not be interpreted as an inherent limitation of PCIe-attached FPGA acceleration. Rather, under the baseline configuration, the FPGA-to-CPU throughput ratio of $2.0\times$ is insufficient to compensate for PCIe communication latency, data-transfer time, and contention overhead. Workloads with higher arithmetic intensity or systems with higher interconnect bandwidth may therefore exhibit different performance characteristics.

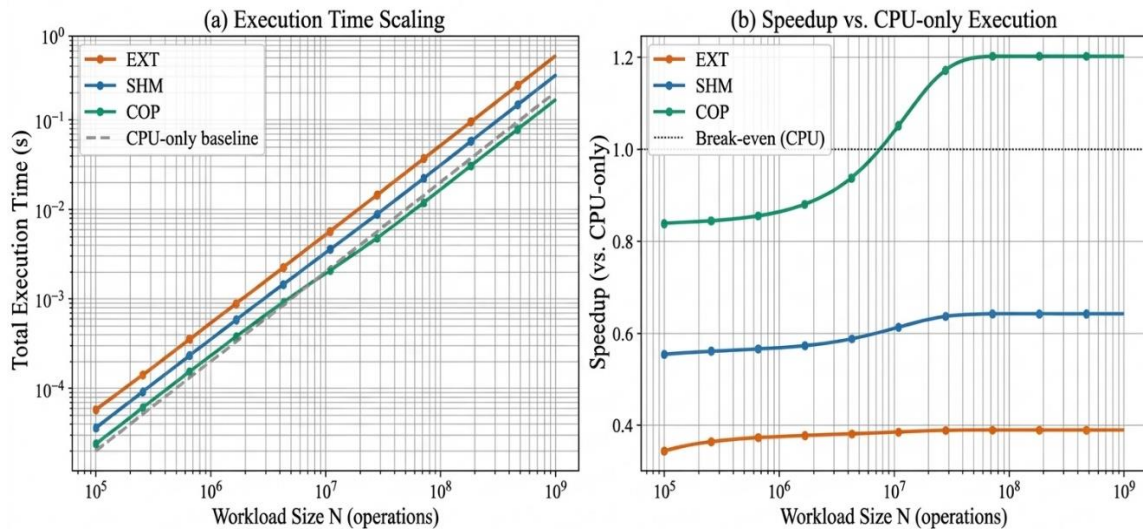


Fig. 3 (a) Total execution time as a function of workload size for the three coupling topologies and the CPU-only baseline, and (b) corresponding speed-up relative to CPU-only execution. The COP topology crosses the break-even line at approximately $N=10^7$ operations and approaches a speed-up of $1.2\times$ for $N\geq 10^8$.

A complementary bandwidth sensitivity analysis was performed to examine the effect of newer PCIe generations on the EXT topology. Using approximate x16 per-direction bandwidths of 16, 32, 64, and 128 GB/s for PCIe Gen3, Gen4, Gen5, and Gen6, respectively, while keeping the remaining model parameters unchanged, the predicted EXT speed-up at $N=10^8$ operations increased from 0.39 $\times$ for Gen3 to 0.63 $\times$ for Gen4, 0.90 $\times$ for Gen5, and 1.16 $\times$ for Gen6.

The corresponding results are summarized in Table 5. These results indicate that the deceleration observed in the baseline configuration is strongly dependent on interconnect capability and should not be interpreted as a general limitation of PCIe-attached FPGA acceleration.

Table 5. Sensitivity of EXT performance to PCIe bandwidth at $N=10^8$ operations and $I=0.125$ operations/byte

PCIe Generation	Approx. x16 Bandwidth (GB/s)	Predicted EXT Speed-up
Gen3 (baseline)	16	0.39 $\times$
Gen4	32	0.63 $\times$
Gen5	64	0.90 $\times$
Gen6	128	1.16 $\times$

To further examine the source of the execution-time differences among the three coupling topologies, Figure 4 decomposes the total execution time into communication and computation components and shows how the communication fraction varies with workload size.

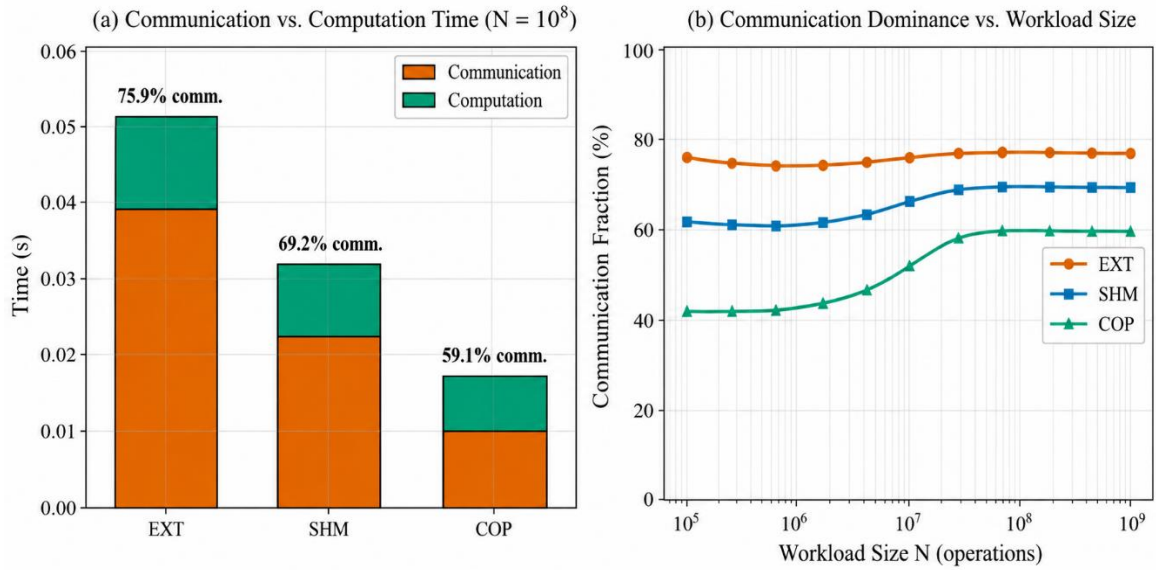


Fig. 4 (a) Stacked decomposition of execution time into communication and computation components at $N=10^8$, and (b) Communication fraction as a function of workload size. The communication fraction approaches a topology-dependent saturation level as workload size increases, reflecting the asymptotic behavior of the workload-adaptive overlap factor $\gamma(N)$.

As shown in Figure 4(a), communication accounts for 75.9% of the total execution time for EXT at $N=10^8$, compared with 69.2% for SHM and 59.1% for COP. This trend indicates that tighter CPU-FPGA coupling reduces the relative communication overhead through lower communication latency and higher bandwidth.

Figure 4(b) further shows that the communication fraction approaches a topology-dependent saturation level as the workload size increases. EXT maintains the highest communication share, whereas COP exhibits the lowest share across the evaluated workload range. This behavior is consistent with the workload-adaptive overlap model, in which $\gamma(N)$ approaches $\gamma_{\max}$ for large workloads.

6.2. Energy-Efficiency Analysis

Figure 5(a) shows the energy per operation (J/op) for the three coupling topologies across the evaluated workload

range. As the workload size increases, the energy per operation approaches a relatively stable level, particularly for the COP topology. At $N=10^8$ operations, the energy-per-operation values are 1.59×10^{-8} J/op for EXT, 1.01×10^{-8} J/op for SHM, and 6.73×10^{-9} J/op for COP. Under the evaluated baseline parameters, COP therefore requires approximately 57.7% less energy per operation than EXT.

Figure 5(b) decomposes the total energy at $N=10^8$ operations into CPU, FPGA, and link components. The link-energy component decreases from 0.195 J for EXT to 0.005 J for COP, corresponding to a reduction of approximately 97.4%. This substantial reduction in link energy contributes to the lower total energy of the more tightly coupled COP topology. The CPU and FPGA energy components also reflect the active and idle power states represented by the dual-state energy model.

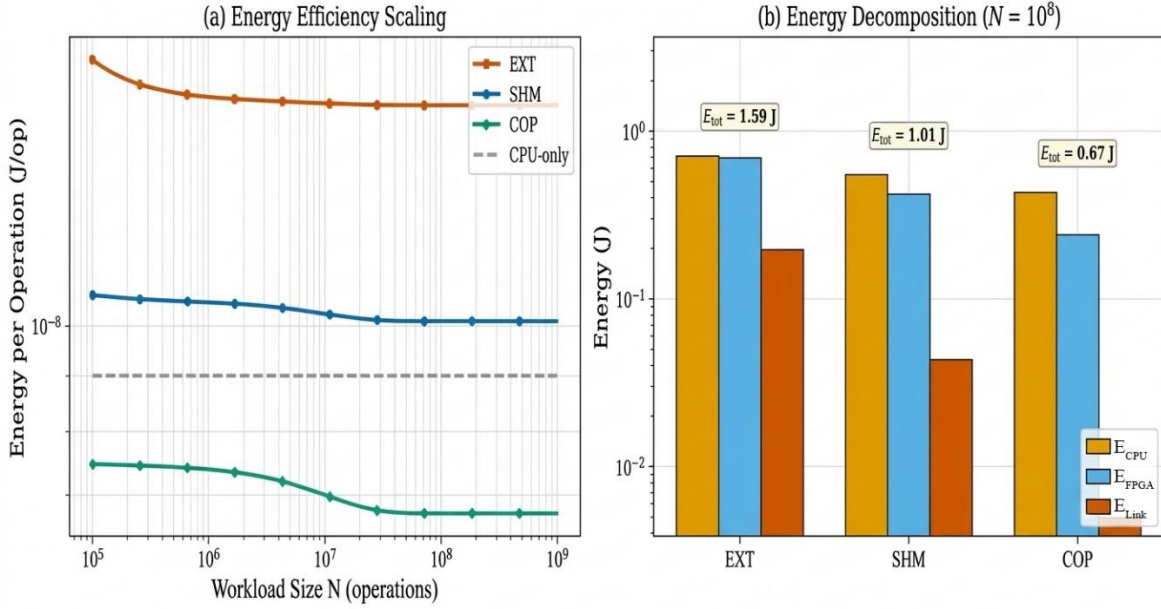


Fig. 5 (a) Energy per operation as a function of workload size for the three coupling topologies, and (b) Decomposition of total energy at $N=10^8$ operations into CPU, FPGA, and link components. The results illustrate the reduction in link-energy contribution as the CPU-FPGA coupling becomes tighter.

6.3. Design-Space Exploration

Figure 6 shows the predicted speed-up over the CPU-only baseline across a design space defined by link latency L and bandwidth B at a workload size of $N=10^7$ operations. Two main trends can be observed.

First, the iso-speed-up contours become increasingly vertical at higher bandwidths, indicating diminishing performance gains from additional bandwidth once the link bandwidth reaches approximately 10 GB/s under the evaluated conditions. Second, latency remains influential

across the explored bandwidth range. When the link latency exceeds approximately 1 μ s, increasing bandwidth alone provides limited improvement in the predicted speed-up. The locations of EXT, SHM, and COP in the design space further illustrate the combined influence of latency and bandwidth. Among the three evaluated configurations, COP occupies the low-latency, high-bandwidth region, whereas EXT is located in the higher-latency, lower-bandwidth region, with SHM positioned between them. These results indicate that reducing communication latency can provide substantial performance benefits for the evaluated workload, particularly after bandwidth gains begin to diminish.

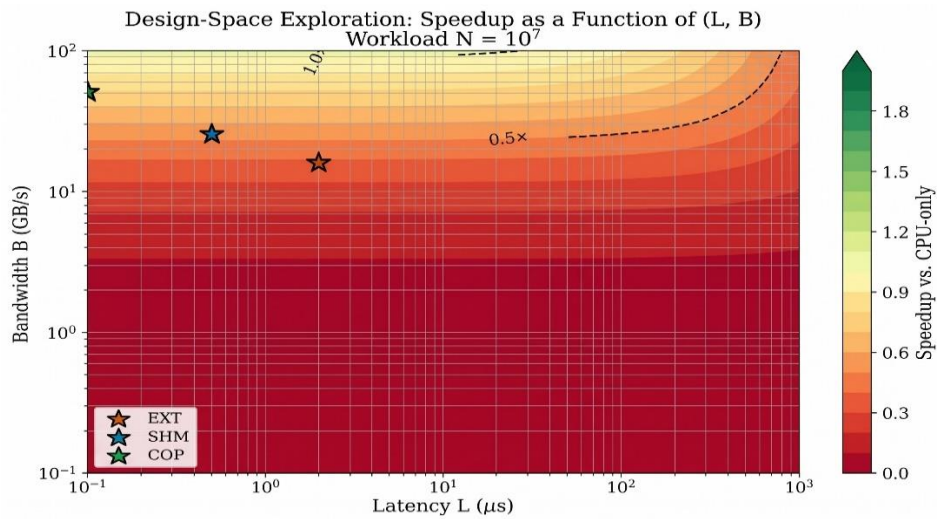


Fig. 6 Design-space exploration of speed-up over the CPU-only baseline as a function of link latency L and bandwidth B at $N=10^7$ operations. The three coupling topologies are indicated by star markers. The iso-speed-up contours illustrate diminishing performance gains from additional bandwidth at higher bandwidth levels and the continued influence of link latency under the evaluated conditions.

6.4. Machine-Learning Surrogate Predictor

The Gradient-Boosted-Tree (GBT) model is employed as a computational surrogate for repeated evaluations of the analytical model during large-scale design-space exploration. It is not intended to serve as an independent hardware-prediction model. Instead, its purpose is to reproduce the analytical model outputs at a lower computational cost when a large number of design points must be evaluated. A dataset of 2,500 (architecture, α , L, B) samples was generated. The workload size N was sampled logarithmically over the range 10^5 - 10^9 operations, while the offload fraction α was sampled uniformly between 0.05 and 0.95. The baseline latency L and bandwidth B values were varied using multiplicative perturbations of up to $\pm 30\%$. For each sample, the reference values of T total and E total were generated using the analytical model. The dataset was divided into 80% training and 20% test samples. The GBT model was configured with 400 estimators, a maximum tree depth of 4, and a learning rate of 0.05.

Figure 7 compares the surrogate predictions with the analytical-model reference values on the held-out test set. The execution-time surrogate achieves $R^2 = 0.9993$ and a MAPE of 5.56%, while the energy surrogate achieves $R^2 = 0.9995$ and a MAPE of 4.46%. These results indicate that the GBT closely reproduces the analytical model within the sampled design space; they should not be interpreted as direct measures of predictive accuracy against physical hardware. The GBT surrogate requires approximately 3 μ s per query, compared with approximately 200 μ s for the analytical model. Although the difference is small for a single evaluation, it can reduce the computational cost of design-space exploration and optimization procedures that require a large number of repeated evaluations. The surrogate can therefore be used to support efficient search over combinations of workload and architecture parameters within the range represented by the training data.

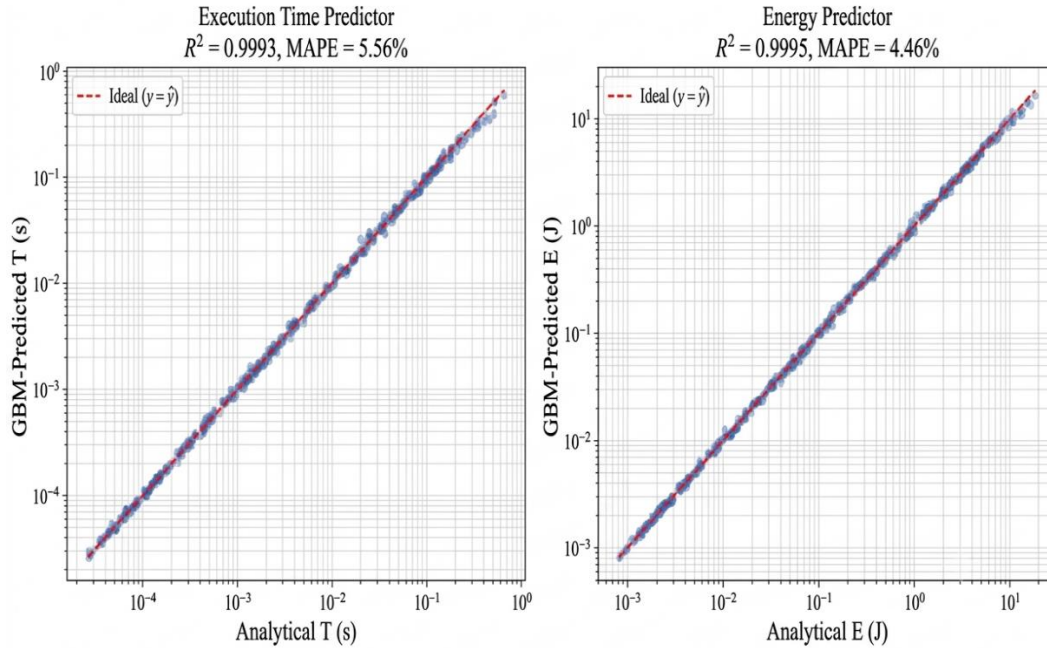


Fig. 7 GBT surrogate performance on the held-out test set. (a) Execution-time surrogate: $R^2 = 0.9993$, MAPE = 5.56%, and (b) Energy surrogate: $R^2 = 0.9995$, MAPE = 4.46%. The dashed line represents the ideal agreement between the surrogate predictions and the analytical-model reference values.

6.5. Energy-Time Trade-off and Optimal Workload Distribution

Figure 8 illustrates the energy-time trade-off obtained by varying the offload fraction α from 0 to 1 at $N = 10^8$ operations. The results show that the energy-minimizing offload fraction depends strongly on the coupling topology. The minimum-energy points occur at $\alpha^* = 0.00$ for EXT, $\alpha^* = 0.30$ for SHM, and $\alpha^* = 1.00$ for COP. This indicates that, under the evaluated parameters, no offloading is preferred for EXT, partial offloading for SHM, and full offloading for COP when energy consumption is the primary objective.

The three topologies also exhibit different energy-time trade-off trajectories. COP achieves the lowest minimum-energy point among the evaluated configurations, while increasing the offload fraction in EXT leads to less favorable performance under the baseline conditions.

These results demonstrate that the appropriate offload fraction depends on the selected coupling topology, and therefore, workload distribution and topology selection should be considered jointly.

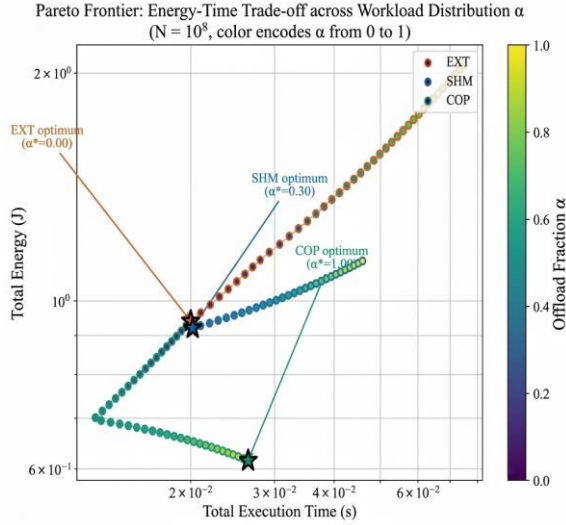


Fig. 8 Energy-time trade-off in the (T_{total}, E_{total}) plane obtained by varying the offload fraction α at $N=10^8$ operations. Star markers indicate the energy-minimizing α^* for each topology, while color indicates the offload fraction.

6.6. Adaptive Coupling Selector

Figure 9 presents the decision map of the Adaptive Coupling Selector (ACS). The architecture-selection problem is formulated by jointly considering the Energy-Delay Product (EDP) and a normalized deployment-cost term. For a given workload size N and weighting factor w_E , the ACS selects

the architecture that minimizes the objective function defined in Equation (10).

$$Cost(arch | N, w_E) = w_E \cdot EDP_{norm}(arch, N) + (1 - w_E) \cdot C_{norm}(arch) \quad (10)$$

Where $EDP_{norm}(arch, N)$ is the Energy-Delay Product, $E_{total}(arch, N) \times T_{total}(arch, N)$, normalized by the maximum value across the evaluated architectures. $C_{norm}(arch)$ is a normalized deployment-cost parameter, with baseline values of 1.0, 1.6, and 3.2 for EXT, SHM, and COP, respectively. These values are used as relative cost assumptions for the comparative analysis rather than as direct estimates of specific hardware prices. The weighting factor $w_E \in [0, 1]$ controls the relative importance of EDP and deployment cost in the objective function. Figure 9 shows that the ACS decision map is divided into three main regions.

At low values of w_E , where deployment cost has a greater influence on the objective function, EXT is selected. At intermediate values of w_E , SHM becomes the preferred architecture, whereas COP is selected at higher values of w_E , where greater emphasis is placed on EDP. The transition boundaries vary slightly with workload size N , indicating that the architecture-selection decision depends on both workload scale and the relative importance assigned to energy-performance efficiency and deployment cost.

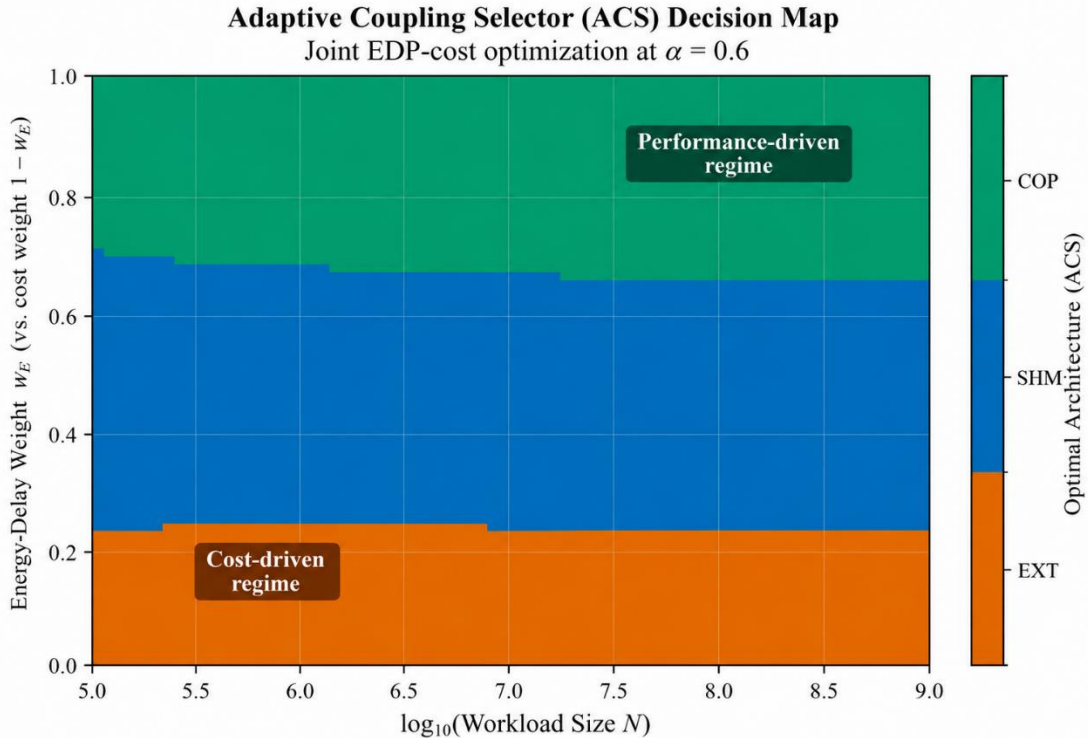


Fig. 9 Adaptive Coupling Selector (ACS) decision map as a function of workload size N and the weighting factor w_E . The three regions indicate the architecture selected by the ACS objective function in Equation (10), illustrating the trade-off between normalized Energy-Delay Product and deployment cost.

6.7. Monte-Carlo Robustness Analysis

The robustness of the predicted energy consumption and speed-up was evaluated using a Monte-Carlo analysis with 5,000 trials. Link latency L was perturbed using a log-normal multiplicative factor with $\sigma=0.15$, while link bandwidth B and link power P link were perturbed using multiplicative factors with $\sigma=0.10$. Figure 10 shows the resulting total-energy and speed-up distributions at $N=10^8$ operations.

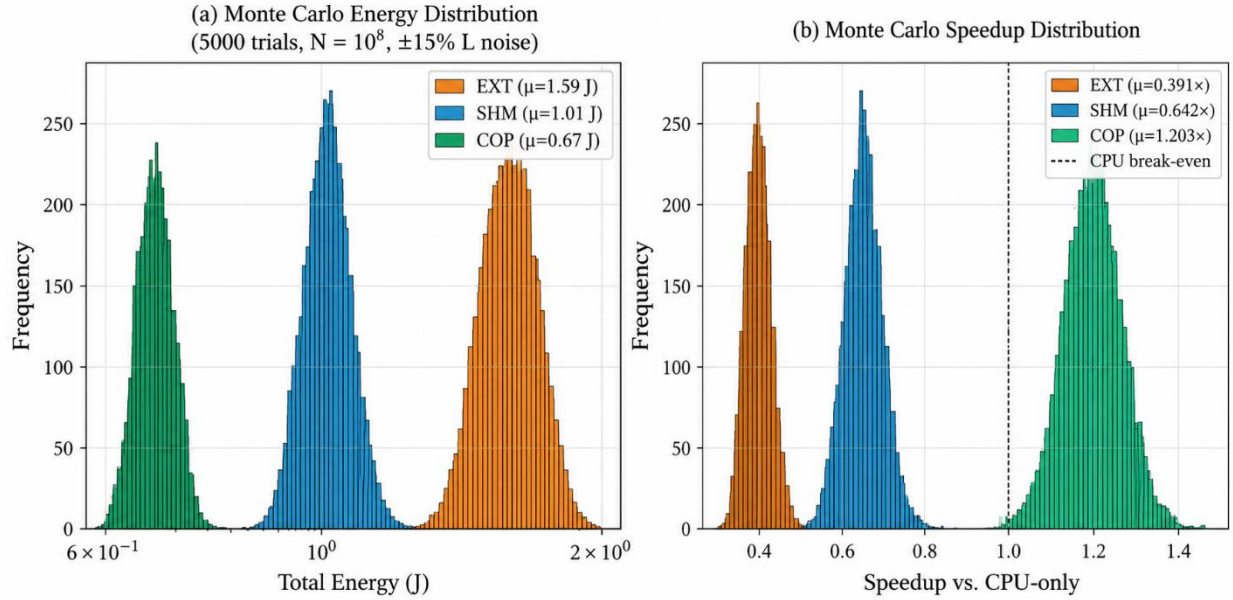


Fig. 10 Monte-Carlo robustness analysis with 5,000 trials. Link latency is perturbed using a log-normal multiplicative factor with $\sigma=0.15$, while bandwidth and link power are perturbed with $\sigma=0.10$. (a) Total-energy distributions at $N=10^8$ operations; (b) speed-up distributions relative to the CPU-only baseline. The separation of the energy distributions indicates that the $EXT > SHM > COP$ ordering remains stable under the considered parameter perturbations.

6.8. Validation against Published Benchmarks

Table 6 summarizes the comparative results at $N=10^8$ operations. The predictive accuracy of the framework was further assessed by comparing its predictions with results reported for five published FPGA benchmarks, as shown in Figure 11.

The comparison includes the Stratix V KVS implementation of Blott et al. [27], the Stratix IV document-filter implementation of Chen [19], the Zynq SHM platform of Cong and Xiao [28], the HARP cache-coherent platform of Oliver et al. [38], and the CCIX-based FPGA system reported by Kumar and Lee [10].

Across these five benchmarks, the Mean Absolute Percentage Error (MAPE) is 4.4% for speed-up and 7.0% for total energy. These results indicate that the framework provides reasonably close predictions for the evaluated benchmark set, while the remaining prediction error reflects the limitations and assumptions discussed in Section 6.10.

The mean total-energy values are approximately 1.59 J for EXT, 1.01 J for SHM, and 0.67 J for COP. The corresponding 95% intervals are [1.39, 1.82] J, [0.91, 1.12] J, and [0.63, 0.72] J, respectively. The separation of these intervals indicates that the energy ordering $EXT > SHM > COP$ remains stable under the considered parameter perturbations. Similarly, the speed-up distributions remain separated from the CPU-only break-even value: COP remains above a speed-up of 1, whereas EXT and SHM remain below it.

Table 6. Comparative summary of performance and energy at the default workload $N = 10^8$

Metric	EXT	SHM	COP	COP relative to. EXT
Total execution time (ms)	51.4	31.2	16.6	-67.7 %
Speed-up vs. CPU-only	0.39×	0.64×	1.20×	×3.08
Total energy (J)	1.59	1.01	0.67	-57.7 %
Energy per operation (J/op)	1.59×10^{-8}	1.01×10^{-8}	6.73×10^{-9}	-57.7 %
Communication fraction	75.9 %	69.2 %	59.1 %	-16.8 pp
Adaptive overlap factor $\gamma(N)$	0.200	0.550	0.900	×4.5
Link energy (J)	0.195	0.043	0.005	-97.4 %

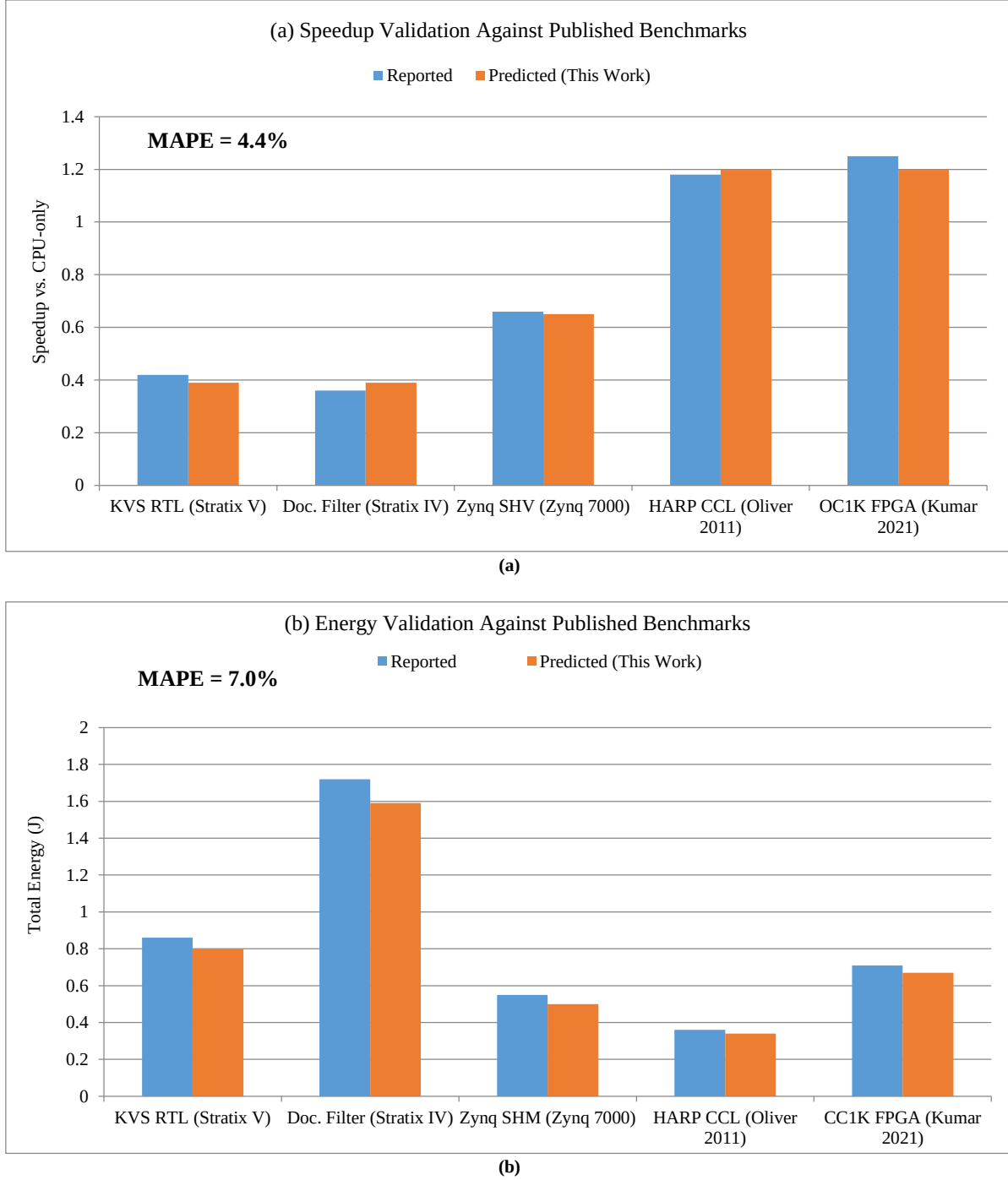


Fig. 11 Comparison with five published FPGA benchmarks. (a) Comparison of predicted and published speed-up values, with a Mean Absolute Percentage Error (MAPE) of 4.4%, and (b) Comparison of predicted and published total-energy values, with a MAPE of 7.0%.

6.9. Relation to Commercial Estimation Tools

The proposed framework differs from commercial FPGA power-estimation tools in its intended stage of use and level of abstraction. Xilinx Power Estimator (XPE) and Intel Quartus Power Analyzer provide device-level power estimates using hardware-specific design information, whereas the proposed framework operates at the workload and architectural levels using parameters such as N , α , and D op.

This allows performance and energy estimates to be obtained during early design-space exploration, before a complete hardware implementation is available.

Under the evaluated model configuration, a single analytical-model evaluation requires approximately 200 μ s. The proposed framework represents CPU, FPGA, and interconnect power using the baseline parameters listed in

Table 4. Off-chip interconnect power is represented separately through topology-specific link-power parameters, as described in Section 4.2. These component-level parameters are combined with workload-dependent execution and communication models to support early architectural comparison.

The proposed framework is not intended to replace device-level commercial power-estimation tools. In particular, it does not model design-specific switching activity within the FPGA fabric.

Consequently, its power estimates should be interpreted as architectural-level estimates for early design-space exploration rather than as substitutes for implementation-level power analysis. Incorporating switching-activity information obtained from HLS or implementation-level profiling is therefore considered a direction for future work.

6.10. Validity Limits and Modelling Assumptions

The analytical model presented in Section 4 is based on four simplifying assumptions that define its validity limits. First, the workload is assumed to be perfectly partitionable: a fraction α of the N operations can be offloaded to the FPGA without data dependencies between the residual CPU operations and the FPGA kernel.

Workloads with fine-grained or irregular dependency graphs, such as pointer-chasing graph traversals or irregular sparse-matrix kernels, may violate this assumption and introduce synchronization overheads that are not captured by Equations (1)-(4). Therefore, the predicted speed-up should be interpreted within these modelling assumptions.

Second, the workload-dependent overlap factor γ (N) is parameterized using topology-specific baseline values adopted for the present analysis. Its extrapolation to workload sizes outside the considered range or to architectures with substantially different buffering characteristics may introduce additional uncertainty.

Third, the bus-contention factor κ is treated as a topology-specific model parameter representing different levels of communication contention rather than as a directly measured hardware constant. Its influence on the model predictions is therefore examined through sensitivity analysis.

Fourth, the GBT surrogate is trained using samples generated by the analytical model. Consequently, its reported R^2 and MAPE values quantify the surrogate's fidelity to the analytical model rather than its direct predictive accuracy against physical hardware. The comparison with published benchmark results in Section 6.8 provides a separate assessment of the overall framework against reported hardware results. The current communication model does not explicitly represent DMA pipelining, double buffering, or

asynchronous streaming. Their effects are captured only indirectly through the workload-dependent overlap factor γ (N). Consequently, EXT predictions represent the stated communication assumptions and may be conservative for highly optimized PCIe implementations that achieve greater communication-computation overlap.

These limitations define the conditions within which the framework predictions should be interpreted and motivate the future-work directions discussed in Section 8.

7. Application Case Studies

The framework presented above is general and supports workloads that can be characterized by the (N , α , D op) parameters. To illustrate its applicability to current FPGA deployment scenarios, this section discusses three representative domains: algorithm acceleration, network acceleration, and data-access acceleration.

7.1. Algorithm Acceleration

Stream-processing kernels, such as document filtering, JPEG decoding, and Smith-Waterman sequence alignment, typically have well-defined dataflow patterns and relatively high arithmetic intensity. Chen and Singh [19] evaluated an information-filtering application on a Stratix-IV FPGA using OpenCL and reported performance and energy-efficiency benefits compared with a CPU baseline.

Within the proposed framework, workloads with regular access patterns and low communication-to-computation ratios can make the EXT topology a practical option. For workloads with greater communication requirements or more irregular access patterns, tighter coupling through SHM or COP may reduce data-transfer overhead.

7.2. Network Acceleration

Network-acceleration deployments often place the FPGA close to the data path, either within a programmable NIC, such as the Mellanox ConnectX-3 Pro [25], or through a protocol-processing platform, such as the Solarflare Application-Onload Engine [26]. These designs highlight the importance of communication latency and data-transfer overhead between the host and the accelerator. Within the proposed framework, tighter coupling can reduce the relative communication overhead. At $N=10^8$ operations, the predicted communication fraction decreases from 75.9% for EXT to 69.2% for SHM and 59.1% for COP under the baseline parameters.

7.3. Data-Access Acceleration

Data-access workloads, such as distributed Key-Value Store (KVS) systems, provide another representative application domain. Blott et al. [27] reported an RTL-based KVS implementation on a Stratix-V FPGA capable of line-rate processing, while Cong and Xiao [28] investigated shared-memory CPU-FPGA systems for data-intensive workloads. For such applications, the Adaptive Coupling

Selector described in Section 6.6 can be used to evaluate the trade-off between EDP and deployment cost and to select a suitable coupling topology according to the workload size and the weighting factor w_E .

8. Conclusion and Future Work

The paper presents a hybrid analytical-machine-learning framework for performance and energy modelling of CPU-FPGA heterogeneous architectures. The framework includes five main contributions: a workload-adaptive overlap factor $\gamma(N)$ that represents the workload-dependent overlap between communication and computation; a dual-state energy model that accounts for the active and idle power of both the CPU and FPGA; a Gradient-Boosted-Tree (GBT) surrogate that provides computationally efficient evaluations of the analytical model for large-scale design-space exploration; an Adaptive Coupling Selector (ACS) that jointly considers Energy-Delay Product (EDP) and normalized deployment cost; and empirical validation and uncertainty analysis based on published FPGA benchmarks and Monte-Carlo simulations. External comparison with five published FPGA benchmarks resulted in MAPE values of 4.4% for speed-up and 7.0% for total energy. In addition, the 5,000-trial Monte-Carlo analysis showed that the relative performance and energy trends remained stable under the considered parameter perturbations. At the representative workload of $N = 10^8$ operations, the COP topology achieves a speed-up of $1.20\times$ over the CPU-only baseline, compared with $0.39\times$ for EXT, corresponding to a $3.08\times$ higher speed-up ratio. COP also

reduces total energy consumption by approximately 57.7% relative to EXT under the evaluated baseline parameters. The Pareto analysis further shows that the energy-minimizing offload fraction α^* is topology dependent, with $\alpha^* = 0.00$ for EXT, $\alpha^* = 0.30$ for SHM, and $\alpha^* = 1.00$ for COP. These results indicate that workload distribution and coupling-topology selection should be considered jointly.

Future work will focus on three main directions. First, the analytical model can be extended to incorporate Dynamic Voltage and Frequency Scaling (DVFS) for both the CPU and FPGA, enabling the evaluation of power-management policies under variable workloads.

Second, more advanced surrogate models, including graph neural networks, could be investigated to represent workload dataflow characteristics and support finer-grained predictions. Third, the framework can be extended to multi-FPGA and disaggregated heterogeneous architectures interconnected through emerging coherent fabrics such as CXL

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

The authors received no external funding for this research.

References

- [1] Hadi Esmaeilzadeh et al., "Dark Silicon and the End of Multicore Scaling," *Proceedings of the 38th Annual International Symposium on Computer Architecture*, pp. 365-376, 2011. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Michael B. Taylor et al., "Is Dark Silicon Useful? Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse," *Proceedings of the 49th Annual Design Automation Conference*, pp. 1131-1136, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Sparsh Mittal, and Jeffrey S. Vetter, "A Survey of CPU-GPU Heterogeneous Computing Techniques," *ACM Computing Surveys*, vol. 47, no. 4, pp. 1-35, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Matthew Jacobsen et al., "RIFFA 2.1: A Reusable Integration Framework for FPGA Accelerators," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 8, no. 4, pp. 1-23, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Andrew Putnam et al., "FPGAs in the Datacenter: Combining the Worlds of Hardware and Software Development," *Proceedings of the Great Lakes Symposium on VLSI*, pp. 5-6, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Kentaro Sano et al., "Evaluating Power and Energy Consumption of FPGA-based Custom Computing Machines for Scientific Floating-Point Computation," *2008 International Conference on Field-Programmable Technology*, Taipei, Taiwan, pp. 301-304, 2008. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] AMD, Zynq UltraScale+ MPSoC Data Sheet: Overview, DS891 (v1.9), 2025. [Online]. Available: <https://docs.amd.com/v/u/en-US/ds891-zynq-ultrascale-plus-overview>
- [8] Intel Corporation, Intel[®] Xeon[®] Scalable Processors and Programmable Accelerators, Technical White Paper, 2017. [Online]. Available: <https://download.intel.com/newsroom/2021/archive/2017-08-29-news-intel-xeon-scalable-processors-accelerate-creation-innovation-next-generation-workstations.pdf>
- [9] J. Stuecheli et al., "CAPI: A Coherent Accelerator Processor Interface," *IBM Journal of Research and Development*, vol. 59, no. 1, pp. 1-7, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Sajjad Tamimi et al., "An Evaluation of using CCIX for Cache-Coherent Host-FPGA Interfacing," *2022 IEEE 30th Annual International Symposium on Field-Programmable Custom Computing Machines*, New York City, NY, USA, pp. 1-9, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [11] Liang Feng et al., "HeteroSim: A Heterogeneous CPU-FPGA Simulator," *IEEE Computer Architecture Letters*, vol. 16, no. 1, pp. 38-41, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Jason Cong et al., "Understanding Performance Differences of FPGAs and GPUs," *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Boulder, CO, USA, pp. 93-96, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Heiner Giefers et al., "Analyzing the Energy-Efficiency of Sparse Matrix Multiplication on Heterogeneous systems: A comparative study of GPU, Xeon Phi and FPGA," *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Uppsala, Sweden, pp. 331-340, 2016, [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Kentaro Sano, and Satoru Yamamoto, "FPGA-based Scalable and Power-Efficient Fluid Simulation using Floating-Point DSP Blocks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2823-2837, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Anca Molnos et al., "A Composable, Energy-Managed, Real-Time MPSOC Platform," *2010 12th International Conference on Optimization of Electrical and Electronic Equipment*, Brasov, Romania, pp. 870-876, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Dimitrios Mbakoyiannis, Othon Tomoutzoglou, and George Kornaros, "Energy-Performance Considerations for Data Offloading to FPGA-based Accelerators Over PCIe," *ACM Transactions on Architecture and Code Optimization*, vol. 15, no. 1, pp. 1-24, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] A.M.D. Xilinx, UltraScale Architecture Memory Resources user Guide, AMD Technical Information Portal. [Online]. Available: <https://docs.amd.com/r/en-US/ug573-ultrascale-memory-resources/Block-RAM-Error-Correction-Code>
- [18] Dario Cozzi et al., "OLT(RE)²: An On-Line On-Demand Testing Approach for Permanent Radiation Effects in Reconfigurable Systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 4, pp. 511-523, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] Doris Chen, and Deshanand Singh, "Invited Paper: Using OpenCL to Evaluate the Efficiency of CPUs, GPUs and FPGAs for Information Filtering," *22nd International Conference on Field-Programmable Logic and Applications*, pp. 5-12, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Fabio F. de Oliveira, Leonardo A. Dias, and Marcelo A.C. Fernandes, "Proposal of Smith-Waterman Algorithm on FPGA to Accelerate the Forward and Backtracking Steps," *Plos one*, vol. 17, no. 6, pp. 1-28, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] K. Rahimunnisa et al., "FPGA Implementation of AES Algorithm for High Throughput using Folded Parallel Architecture," *Security and Communication Networks*, vol. 7, no. 11, pp. 2225-2236, 2014. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Ahmet Malal, and Cihangir Tezcan, "First Fully Pipelined High Throughput FPGA Implementation and GPU Optimization of Wider Variant of AES," *Journal of Cryptographic Engineering*, vol. 16, no. 1, pp. 1-18, 2026. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Nada Qasim Mohammed et al., "Design Hybrid Architecture to Implement AES Algorithm on FPGA for IoT Applications," *Baghdad Science Journal*, vol. 22, no. 4, pp. 1379-1392, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Luís Fiolhais, and Horácio Neto, "An Efficient Exact Fused Dot Product Processor in FPGA," *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, 2018, pp. 327-334, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Mellanox Technologies, Programmable ConnectX-3 Pro Adapter Card: Dual-Port Programmable Adapter with Virtual Protocol Interconnect, Product Brief, 2015. [Online]. Available: https://format.com.pl/site/wp-content/uploads/2015/08/pb_programmable_connectx-3_pro_-card_vpi.pdf
- [26] Timothy Prickett Morgan, Solarflare Turns Network Adapters into Servers: When a CPU Just isn't Fast Enough, The Register, 2012. [Online]. Available: <https://www.theregister.com/on-prem/2012/02/08/solarflare-turns-network-adapters-into-servers/1501984>
- [27] Michaela Blott et al., "Achieving 10 Gbps Line-Rate Key-Value Stores with FPGAs," *5th USENIX Workshop on Hot Topics in Cloud Computing*, vol. 13, pp. 1-6, 2013. [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Young-kyu Choi et al., "A Quantitative Analysis on Microarchitectures of Modern CPU-FPGA Platforms," *Proceedings of the 53rd Annual Design Automation Conference*, pp. 1-6, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Kwangho Lee et al., "Memory Streaming Acceleration for Embedded Systems with CPU-Accelerator Cooperative Data Processing," *Microprocessors and Microsystems*, vol. 71, pp. 1-10, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Eric S. Chung et al., "Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPGUs?," *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, Atlanta, GA, USA, pp. 225-236, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [31] Kaiyuan Guo et al., "Angel-Eye: A Complete Design Flow for Mapping CNN Onto Embedded FPGA," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 35-47, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [32] Intel Corporation, 4th Gen Intel Xeon Scalable Processors, Product Brief, 2023. [Online]. Available: <https://www.intel.com/content/dam/www/central-libraries/us/en/documents/2023-09/4th-gen-xeon-revised-product-brief.pdf>
- [33] AMD, UltraScale Architecture and Product Data Sheet: Overview, DS890, 2026. [Online]. Available: <https://docs.amd.com/v/u/en-US/ds890-ultrascale-overview>
- [34] PCI-SIG, "PCI Express Base Specification Revision 3.0," 2010. [[Google Scholar](#)] [[Publisher Link](#)]

- [35] Arm Limited, "AMBA 5 CHI Architecture Specification," ARM IHI 0050E.b, 2021. [Online]. Available: <https://documentation-service.arm.com/static/6295e03db334256d9ea8ae86>
- [36] Stavros Volos et al., "CCNoC: Specializing On-Chip Interconnects for Energy Efficiency in Cache-Coherent Servers," *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*, Lyngby, Denmark, pp. 67-74, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [37] John L. Henning, "SPEC CPU2006 Benchmark Descriptions," *ACM SIGMETRICS Performance Evaluation Review*, vol. 34, no. 4, pp. 1-17, 2006. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [38] Neal Oliver et al., "A Reconfigurable Computing System based on a Cache-Coherent Fabric," *2011 International Conference on Reconfigurable Computing and FPGAs*, Cancun, Mexico, pp. 80-85, 2011. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [39] Andrew Putnam et al., "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services," *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 13-24, 2014. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [40] Sparsh Mittal, "A Survey of FPGA-based Accelerators for Convolutional Neural Networks," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1109-1139, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [41] Yuliang Li et al., "HPCC: High Precision Congestion Control," *Proceedings of the ACM Special Interest Group on Data Communication*, pp. 44-58, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [42] Hamid Reza Zohouri, Artur Podobas, and Satoshi Matsuoka, "Combined Spatial and Temporal Blocking for High-Performance Stencil Computation on FPGAs using OpenCL," *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 153- 162, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [43] John W. Lockwood, and Madhu Monga, "Implementing Ultra Low Latency Data Center Services with Programmable Logic," *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects*, Santa Clara, CA, USA, pp. 68-77, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [44] David B. Thomas, "Acceleration of Financial Monte-Carlo Simulations using FPGAs," *2010 IEEE Workshop on High Performance Computational Finance*, pp. 1-6, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [45] Aysegul Dundar et al., "Accelerating Deep Neural Networks on Mobile Processor with Embedded Programmable Logic," *Neural Information Processing Systems Conference (NIPS)*, 2013. [[Google Scholar](#)]
- [46] Aaftab Munshi, "The OpenCL Specification," *2009 IEEE Hot Chips 21 Symposium (HCS)*, pp. 1-314, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [47] Sung-Kwang Yoon et al., "Optimized Fault-Tolerant Data Processing Module for High-Reliability CNN Accelerator," *Plos one*, vol. 21, no. 2, pp. 1-17, 2026. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [48] Srihari Cadambi et al., "A Programmable Parallel Accelerator for Learning and Classification," *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, pp. 273-284, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [49] Saman Sadeghi, "Classifying FPGA Technology in Digital Signal Processing: A Review," *International Journal of Engineering, Technology and Sciences*, vol. 2024, pp. 1-10, 2024. [[Google Scholar](#)]
- [50] Nazanin Calagar, Stephen D. Brown, and Jason H. Anderson, "Source-Level Debugging for FPGA High-Level Synthesis," *2014 24th International Conference on Field Programmable Logic and Applications*, Munich, Germany, pp. 1-8, 2014. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]