electronics

MDPI

Review

Hardware Acceleration for Physical AI: Shared Operations and Design Requirements

Răzvan Stancu, Alina-Luiza Petcu, Costin-Cosmin Dumitru, Han-David Zhao and Georgian Nicolae *

Faculty of Electronics, Telecommunications and Information Technology, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania; ionut_razvan.stancu@upb.ro (R.S.); alina_luiza.petcu@upb.ro (A.-L.P.); costin.dumitru0303@upb.ro (C.-C.D.); han_david.zhao@upb.ro (H.-D.Z.)
* Correspondence: georgian.nicolae@upb.ro

Abstract

Embedded physical artificial intelligence (AI) combines sensing, learned inference and physical action under resource and timing constraints. This focused review synthesizes 62 detailed primary-source records across seven workload discussions and eight operation groups into conditional hardware requirements. Six cross-paper comparisons yield five adaptation judgments and one conditional functional match; none validates concurrent physical sharing. The reviewed audio fast Fourier transform (FFT) engine supports maximum lengths of 512/1024/2048 points at 32/16/8 bits per real or imaginary component, whereas the radar engine reaches 4096 points using half-precision arithmetic. Further configuration-specific contrasts expose memory and scheduling tradeoffs. An analytical MCUNetV2 example reduces peak activation storage from 1372 to 172 kB with 10% more multiply–accumulate operations. Published physical inverter tests at 40 kHz report parallel/sequential calculation times of 50/400 ns with 0.7% current total harmonic distortion in both; the corresponding synthesized designs use 200/70 digital signal processing cells. These findings motivate joint selection of numerical format, storage, execution granularity and timing budgets. A quantitative requirements matrix, explicit compatibility criteria and a conceptual architecture connect the evidence to configurable arithmetic, banked storage, state management, arbitration and software interfaces, with RISC-V as an implementation target. The framework distinguishes reusable operations and designs from concurrent resource sharing and specifies the numerical, task-quality and timing tests required to validate proposed mappings.

Keywords: physical AI; embodied AI; hardware accelerator requirements; computational kernels; RISC-V; embedded inference; radar and LiDAR; actuator control; numerical semantics; real-time execution

Check for updates

Academic Editor: John A. Kalomiros

Received: 22 September 2026
Revised: 23 October 2026
Accepted: 3 November 2026
Published: 7 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

An embedded robot or autonomous device may process images and audio, interpret environmental sensors, run learned inference and update its actuators. A researcher choosing an accelerator for this runtime must decide which computations deserve dedicated support and which capabilities could serve several tasks. A product-and-accumulation pattern can describe a neural layer, an adaptive filter or a dynamics calculation while concealing incompatible numerical behavior, state updates and deadlines. The key design question is therefore whether a stated capability can execute particular workload configurations within their application constraints.

Physical artificial intelligence (AI) has a broader meaning than embedded robotics. Wu et al. [1] distinguish physics-informed, generative physical and embodied AI. Here, the operational scope is deliberately narrower: runtime computation in embodied systems that combine sensing, learned inference or decision-making, and physical action. Training, simulation and world models enter when they explain a deployed representation, execution mechanism or transfer limitation. This review does not attempt comprehensive coverage of physical AI or of material and morphological computation.

The research goal is to turn recurring computations into testable requirements for accelerator reuse across this runtime. Computation and communication patterns already underpin the Berkeley taxonomy [2], domain-specific architecture research [3] and APEX processing-element exploration [4]. Building on that work, this review compares two source-linked workload configurations at a time. It checks numerical semantics, data movement, state and timing, records the proposed sharing level, and derives an acceptance test. Hardware designers can use the resulting records to select candidate execution units and memory organizations, identify required adaptations and reject mappings that lack essential evidence.

The technical scope comprises multimedia, AI and machine learning (ML), radar and light detection and ranging (LiDAR), and actuator control. Table 1 maps these categories to runtime roles without treating every classical algorithm as an AI system. AI/ML can occur within the other categories; this overlap is recorded rather than counted as independent evidence. The RISC-V instruction set architecture (ISA) is an implementation target, while generic embedded and other architectural evidence remain eligible.

Table 1. Technical categories and their role in the embedded physical AI scope. The categories overlap where a learned method contributes to sensing or control.

Category	Runtime Role	Requirements to Preserve
Multimedia	Audio, image and video acquisition, preprocessing and representation	Stream layout, buffering, transforms, filtering and source timestamps.
AI/ML	Learned perception, policies and selected action-decoding mechanisms	Numerical requirements, intermediate state, invocation latency and action representation.
Radar/LiDAR	Environmental sensing, localization and state estimation	Detection quality, indexed access, update ordering and observation age.
Actuator control	Optimization, dynamics, feedback and signal generation	Solver reliability, retained state, deadlines and synchronized actuator updates.

The contributions are as follows:

- A configured-workload comparison record that jointly checks numerical semantics, memory/state behavior and sensing-to-action timing, with explicit decision criteria and acceptance conditions.
- A quantitative synthesis of selected requirements and six compatibility cases that separates reported implementations from cross-paper reuse hypotheses.
- A capability taxonomy and conceptual architectural template linking the requirements to execution units, storage, scheduling and software, together with concrete validation workloads.

Section 2 compares the review with prior surveys and design frameworks. Section 3 explains retrieval, selection, evidence extraction and compatibility criteria. Section 4 analyzes the workload stages and consolidates their quantitative requirements. Section 5 develops

the six pairwise comparisons. Section 6 translates them into capabilities, an architectural template and timing/validation conditions. Section 7 examines RISC-V implementation mechanisms. Section 8 discusses design implications, evidence limits and future work, and Section 9 concludes. Appendices A–D retain retrieval records, source evidence, compatibility checks and dated citation observations.

2. Related Reviews and the Contribution Boundary

Computational patterns and domain-specific architectures already guide hardware design [2,3]. Operations-oriented image processing [5], accelerator taxonomies [6], radar/LiDAR architectures [7] and field-programmable gate array (FPGA) robotic computing [8] provide complementary application and implementation views. Specialized reviews address multiply–accumulate (MAC) arithmetic [9], spin-transfer-torque magnetoresistive random-access memory (STT-MRAM) computing [10] and RISC-V hardware/software for edge learning [11]. Physical and embodied AI reviews [1,12] provide the broader deployment context.

Table 2 compares prior organizing approaches, including APEX [4] and Timeloop [13], by scope, analysis unit, numerical/state treatment, timing and engineering output. ORIANNA [14] generates accelerators for optimization-based robotics, while DaDu-Corki [15] supplies algorithm–architecture codesign evidence combining learned trajectory generation and FPGA control.

Table 2. Prior approaches compared by workload granularity, numerical/state treatment, timing and acceptance boundary. Entries describe each approach's organizing scope, not an exhaustive absence/presence ranking.

Approach and Analysis Unit	Numerical, Memory and State Treatment	Timing, Acceptance and Sharing Boundary	Engineering Output
Accelerator taxonomy [6]; platforms	Fourteen aspects cover general properties, host coupling, architecture and software.	Published throughput/efficiency characterize devices; a new joint workload needs separate validation.	Platform classification, trends and challenges.
RISC-V edge learning [11]; hardware/software solutions	Explicit representation, parallelism and data-movement taxonomy.	Deep-learning implementation evidence; application state and concurrent schedules remain configuration-specific.	Implementation options and software/codesign gaps.
Physical/embodied AI [1,12]; methods and deployment	Physical models, representations and embodiment organize the comparison.	Benchmarks and deployment challenges frame evaluation at the AI/system level.	Field taxonomy and research directions.
APEX [4]; recurring computational subgraphs	Merged datapaths and generated compiler mappings specialize processing elements.	Evaluated mapped image/ML applications; area/energy comparisons concern those generated designs.	Hardware/compiler generation and design reuse.
Timeloop [13]; neural-network layer/mapping	Configurable compute precision, memory hierarchy, data staging and bandwidth constraints.	Performance/energy projections; model validation against existing designs.	Architecture/mapping exploration under declared constraints.
This review; paired configured workloads	Joint operand/accumulator, movement, mutable-state and dependency gates.	Sensing-to-action timing and task-quality tests; sharing level explicit. Physical sharing remains unvalidated for all six cases.	Conditional requirements, adaptation decisions and conceptual architecture.

The methodological contribution is the joint decision record: it makes a proposed sharing claim auditable against two configurations, preserves negative cases and identifies the evidence needed to accept or reject that claim. It complements architectural taxonomies and design-space exploration by specifying what an exploration must preserve. It does not introduce a new optimization algorithm or establish priority for hardware reuse. The quantitative synthesis (Section 4.8), decision criteria (Section 3.4) and architectural template (Section 6.1) make the review's engineering output explicit.

3. Review Method and Evidence Model

3.1. Scope, Search Streams and Selection

This focused critical review uses contrasting algorithm configurations to inform accelerator design. The retrieval and purposive selection procedure is specified below.

Two complementary retrieval streams were used in Web of Science Core Collection (WoS CC), Scopus and IEEE Xplore. The generic embedded stream avoids requiring RISC-V, while the RISC-V stream identifies relevant implementations and execution mechanisms. Eight families organize the generic stream: spectral transforms and action representations; adaptive filtering and neural convolution; attention and policy execution; detection and matching; geometry and state estimation; optimization and robot dynamics; closed-loop actuation and signal generation; and reusable execution and mapping mechanisms. The search terms for the eight algorithm families were refined through preliminary searches and checks against previously identified relevant papers. The resulting 24 database searches were conducted on 19 September 2026.

The principal research window was 1 January 2020 to 19 September 2026. Journal articles and conference papers were considered eligible. Publication identity and date were checked against primary sources; online publication and final issue year remained separate when they differed. Earlier foundations, standards and implementation documentation supported interpretation without being counted as recent primary implementations. A preprint and its published version were treated as one study, and related extensions were identified as a publication family when relevant. Appendix B records the primary-source locators and their relevance to the synthesis.

All 24 generic exports reconciled with the observed query counts. They contained 12,480 query-record occurrences, forming 7403 normalized digital object identifier (DOI) groups plus 382 separate no-DOI records. The resulting 7785 metadata groups were a discovery pool, not an included-study count. The separate RISC-V stream contained 3091 occurrences and 1565 metadata candidates. The streams overlapped and were not added. Appendix A reports the retrieval boundaries, query constructions, search-count figure and selection accounting.

Within the pools, purposive selection required a configured algorithm, a relevant operation or dependency mechanism, and extractable implementation, numerical, memory or timing evidence. A detailed source was retained for a distinctive configuration, a counterexample to an apparent match, or evidence of reuse within a system. Thus, fast Fourier transform (FFT) cases varied format and capacity; filtering varied mutable state; inference and control varied numerical requirements and dependencies; selection varied ordering and indexed access. Context sources supplied definitions, standards, benchmarks or review positioning. Sources without usable full text received no numerical implementation claims. Appendix A separates detailed extraction, contextual use, deferred candidates and unassessed records.

Here, "representative" means illustrative of a named mechanism or contrast, not statistically representative of the retrieval pool. Preliminary papers influenced the query vocabulary and the choice of contrasts; this researcher-driven component could favor familiar mechanisms. The examples are sufficient to establish the particular mismatch or acceptance condition documented in each record, but not to establish the prevalence, completeness or optimality of the resulting capability set. No saturation, unbiased recall, independent double screening or exhaustive exclusion of the remaining pool is claimed. The quantitative synthesis therefore counts only explicitly defined records and cases.

The final reference set contained 108 items, including 62 detailed primary-source records. The remainder provides related reviews, methodological context, foundations, standards, implementation documentation or a publication correction. These counts describe cited publications and evidence records, not independent experimental replications; related publication lineages are identified in the synthesis and Appendix B.

Appendix D preserves dated, database-specific citation observations for context; these counts did not determine source selection.

3.2. Configured Workloads and Traceability

A study can contain several configurations. The extraction unit therefore separates the study, publication version and configured workload. Its fields include algorithm variant, dimensions, phase, operations, dependencies, numerical requirements, memory organization, timing boundary, implementation and source locator. Missing quantities are recorded as not reported, rather than inferred from an algorithm name. Appendices B and C summarize the supporting source evidence and the corresponding design inferences.

Numerical descriptions distinguish inputs, weights or coefficients, products, accumulators and outputs. Memory descriptions identify layout, storage level, transfer direction and locality. Timing separates setup, update, transfer, execution and completion. Physical AI records additionally distinguish observation timestamps, policy invocation, action production and actuator updates. Evidence maturity applies to each claim: a paper may combine FPGA execution, simulated power and synthesized application-specific integrated circuit (ASIC) estimates. A fabricated device is distinct from online sensing, closed-loop validation or commercial deployment.

The reasoning chain is observation, configured condition, candidate requirement and acceptance test. The last two are review syntheses unless the source explicitly establishes them. A successful implementation establishes feasibility for its conditions, rather than a universal minimum precision or memory size. Cross-platform speedups and energy figures are not pooled because technology, workload, quality target and measurement boundary generally differ.

3.3. Three Meanings of Sharing

Shared mathematical operations are operations or computational patterns that occur in multiple workloads. Design reuse means a hardware design can be configured for different tasks or products. Physical sharing means tasks use the same instantiated resource. Physical sharing additionally requires state preservation, arbitration and a schedule compatible with their arrivals and deadlines.

A comparison names two configurations, a candidate capability and the intended sharing level. Figure 1 summarizes the procedure; Section 3.4 defines the evidence gates and decision rules.

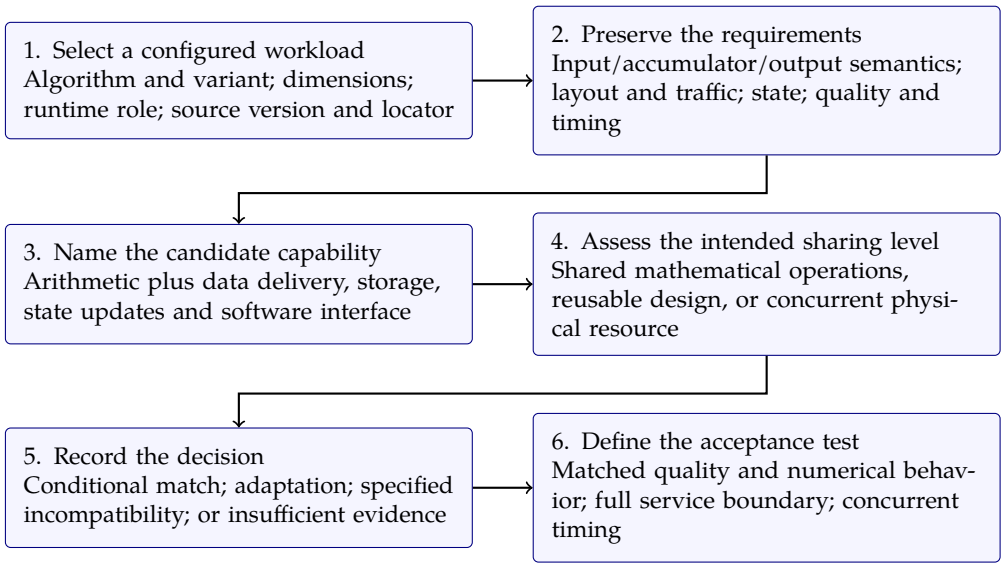

Figure 1. Procedure for translating reviewed algorithms and operations into testable accelerator requirements. Steps 3, 5 and 6 contain engineering synthesis; they do not imply that the cited sources implemented the proposed shared design.

3.4. Reproducible Compatibility Criteria

Table 3 defines a non-compensatory assessment: a favorable arithmetic result cannot cancel a timing failure. For each axis, record P (the stated condition is supported), A (a documented mismatch has an identified adaptation), F (the condition fails for the unchanged candidate), or U (required evidence is unavailable), together with the source locator and test. P describes support for that particular condition, not validation of a complete shared accelerator.

Table 3. Evidence gates for a configured capability. The criteria apply separately to design reuse and physical sharing.

Axis	Condition to Check	Evidence or Acceptance Test
Numerical semantics	Operand and accumulator formats, scaling, rounding, overflow and output tolerance agree.	Reference outputs; signal, inference or control quality after conversion.
Shape and data access	Dimensions, layout, capacity, ordering and transfers fit the proposed mapping.	Working-set and traffic accounting; mapped utilization and throughput.
State and lifecycle	Persistent objects, updates, invalidation and context ownership are preserved.	Cold/warm execution; update and context-isolation tests.
Dependencies	Reduction order, recursion, factorization, synchronization and nonlinear stages remain valid.	Complete dependency graph and operation-level functional tests.
Timing and arbitration	Arrival demand, blocking and interference satisfy each declared deadline and freshness limit.	Joint service trace or justified bound, including movement and completion.
Interface and quality	Invocation, completion, errors and task-level output semantics are defined.	End-to-end execution at the stated quality target.

An unchanged candidate with any F is incompatible under that constraint. An explicitly revised candidate "requires adaptation" when a known mismatch and its remedy

are recorded; the affected gates remain A until tested, and any U is retained alongside the verdict. With no known mismatch but an unresolved essential gate, the verdict is "insufficient evidence." "Conditionally compatible" requires agreement at the named functional boundary and a statement of its assumptions; it does not promote missing physical-sharing gates to P. Each status must be reproducible from its cited configuration.

For PC01, requesting 4096 points from the unchanged ECHOES capacity is F on shape; specifying decomposition and additional storage changes that proposed design-reuse verdict to A, with numerical conversion still requiring validation. For PC06, configurable minimum/maximum score-index selection is a conditional functional match only at an abstract interface that assumes common score widths, overflow and tie rules. Those semantics are not yet established for the configured pair, so numerical compatibility and concurrent arrivals remain U. All six cases retain U for concurrent physical sharing. This separation prevents the five adaptation cases from being counted as demonstrated compatible implementations.

4. Algorithms, Kernels and Operations from Sensing to Action

Table 4 maps illustrative algorithms to eight recurring operation groups. The seven workload discussions distinguish signal preprocessing, detection and matching, estimation, learned inference/action generation, optimization/dynamics, actuation and integrated execution. These are analytical categories, not mutually exclusive study populations. Codec and isolated-kernel cases serve as technical comparators where embodied deployment is unestablished.

Table 4. Recurring operations across the selected workloads. Presence identifies a comparison opportunity, not prevalence, runtime dominance or demonstrated physical sharing.

Operation Group	Representative Algorithms Across Domains	Requirement That Can Prevent Reuse
Products and reductions	Finite impulse response (FIR) and least mean squares (LMS) filtering [16]; convolutional neural network (CNN) convolution [17]; radar accumulation [18]; spatial dynamics [19]	Coefficient lifetime, signedness, scale, accumulator width, reduction length and structure.
Butterflies and separable transforms	Audio fast Fourier transform (FFT) [20]; video discrete cosine transform (DCT) [21]; radar FFT [18]; Frequency-Space Action Sequence Tokenization (FAST) [22]	Length, direction, precision, permutation, transpose storage and the stage where the transform executes.
Interpolation and indexed access	AOMedia Video 1 (AV1) motion search [23]; synthetic aperture radar (SAR) polar reformatting [24]; LiDAR scan matching [25]	Index generation, neighborhood shape, rounding, locality and map mutation.
Normalization and nonlinear functions	Quantized CNN [17] and attention [26]; guided iterative action generation [27]	Division, reciprocal/square root, exponentiation or their approximations; range and intermediate precision.
Comparison, ordering and selection	Video sum of absolute differences (SAD) minimum [28]; constant false alarm rate (CFAR) partial sorting [29]; LiDAR argmax [25]; motor candidate selection [30]	A minimum/index block does not supply partial sorting; score width, tie rules, candidate count and arrival order differ.
Factorization and dependent solves	State estimation [31]; model predictive control (MPC) quadratic programming [32,33]; robotic optimization [14]	Conditioning, rank, pivoting or regularization policy, factor reuse, residual tests and sequential dependencies.
State updates and recursive propagation	Adaptive coefficients [16]; recurrent map/state estimation [31]; robot dynamics [19]; predictive actuation [30]	Update order, persistent context, feedback availability and safe preemption points.
Lookup, counters and bounded event timing	Codec coefficients [34]; adaptive partial-product memories [35]; sinusoidal pulse-width modulation (SPWM) signal generation [36]	Read-only tables and mutable memories have different consistency needs; continuous edge timing differs from setup computation.

4.1. Signal Preprocessing and Multimedia Comparators

4.1.1. Transforms and Data Ordering

Transform reuse depends on precision, capacity and ordering. ECHOES [20] supports maximum FFT lengths of 512, 1024 and 2048 with 32-, 16- and 8-bit real and imaginary components, respectively, in measured silicon. FFT factorization [37] alone therefore does not select an implementation. Standardized codecs add retained windows and distinct transform modes: Low Complexity Communication Codec plus (LC3plus) [38] uses low-delay modified DCT (MDCT) analysis and a separate 16×16 type-II DCT (DCT-II). Versatile Video Coding (VVC) [39] supplies a further application context.

Within that context, Zhou and Pan's 8 × 8 DCT/inverse DCT (IDCT) [21] uses constant shift–add products, eight 8 × 12-bit transpose memories and a 33-cycle block schedule; row inputs are nine/eleven bits for DCT/IDCT and outputs are twelve bits. Hao et al. [40] reuse shift–add circuits and adder trees across DCT-II, type-VIII DCT and type-VII discrete sine transform (DST-VII), supporting 4–32-point transforms through a 32-sample interface. These evaluated configurations leave full-codec throughput and the temporal action-transform mapping in FAST [22] untested.

Radar imaging shows when companion operations determine the useful acceleration boundary. Lee et al. [41] reuse multipliers between 64–4096-point FFT/inverse FFT (IFFT) and synthetic aperture radar (SAR) phase compensation by reordering transposes. Jeong et al. [24] instead couple FFT with interpolation: coordinates and ratios use 32-bit floating point (FP32) before conversion, while phase-history data and FFT use 16-bit floating point (FP16). Their 2048 × 2048 execution time falls from 2.476 s with FFT-only acceleration to 0.766 s with both units. The reusable mechanism is consequently a scheduled arithmetic-and-data path, with conversion and interpolation included.

4.1.2. Fixed and Adaptive Filtering

Coefficient lifetime separates fixed interpolation from adaptive filtering. AV1 decoding [34] fixes filter tables and rounding, whereas encoder fractional motion estimation [23] permits a quality tradeoff: eight/six/four taps need 121/81/49 samples per 4 × 4 output, with average Bjøntegaard delta bit-rate (BD-BR) increases of 0.54/0.54/1.25% on the evaluated sequences. External traffic still depends on caching. Audio adaptation [16] additionally acquires errors and writes coefficients; its instruction-count reduction is not an execution-time measurement.

The distributed-arithmetic alternative [35] stores mutable partial products for 4/8/16-tap, Q1.3 input/coefficient configurations. Its eight-bit output and simulated/synthesized latency depend on serial or paired-bit processing. Narrowband adaptation [42] combines a 1024-point FFT, three-peak selection and adaptive sinusoidal state in an 18-bit fixed-point implementation; sensor output at 2 kHz is distinct from its 81.25 MHz FPGA clock, and piezo feedback remains prospective.

Deployment Oriented to memoRY (DORY) [17] instead uses 8-bit integer (INT8) operands and 32-bit integer (INT32) accumulation. Sharing its product path requires explicit rounding, delay-line access and coefficient writeback; preserving distributed arithmetic requires writable partial-product tables. For a RISC-V host, this decides whether register operands suffice or an operation needs persistent context, memory ordering and explicit completion. Section 7 develops these interface alternatives.

4.2. Detection and Spatial Matching

4.2.1. Radar Detection and Processing Schedules

Radar detection combines transforms, ordering and a service schedule. Heo et al. [18] report 3.00 ms for a 256 × 256 accelerated chain versus 23.89 ms with FFT-only acceleration,

using half-precision floating point. Cell averaging and order-statistic CFAR [43] then impose different data paths: Sim et al. [29] share sums and squares across phases but also need partial sorting in their $N = 16$ synthesis/timing case.

Pipeline readiness can be as important as arithmetic capacity. Petrović et al. [44] keep an early-alert branch continuously ready because their splitter waits for both branches. Flushing reduces measured input-block to output-block onset from 105.3 to 41.72 µs, excluding 0.07 µs of deserialization/alignment. Arbitration that stalls this branch changes that demonstrated property. Guo et al. [45] instead schedule FFT, CFAR and coordinate rotation digital computer (CORDIC) beamforming through single-port static random-access memory (SRAM). Their fabricated chip truncates 24-bit FFT components to sixteen bits and retains target addresses for later spectrum access. Its 12.67 ms frame time is calculated for specified chirp timing and 256 targets; testing uses simulated front-end input.

4.2.2. Video and LiDAR Matching

Video SAD search [28] retains a minimum and index over reference-window reductions; its calculated 4K30 point uses a 128 × 128 coding-tree unit (CTU) and 64 × 64 search range. LiDAR matching [25] instead transforms 32-bit fixed-point range/angle pairs and gathers six-bit stored occupancy values. Neither score-accumulator semantics nor binary points are established for transferring these configurations directly. The video figure excludes complete-encoder work; the LiDAR 2.3–2.4 W measurement includes the whole board.

Coarse-to-fine scan matching [46] reuses coordinate transforms, map reads and accumulation, then adds four-cell interpolation, gradients and Hessian-related terms for host-side solving. Its 30-cycle pipeline latency and six-cycle point interval describe different limits. This motivates a shared address-generation and state-transition interface alongside reduction/selection hardware. Table 5 consolidates the signal configurations; Appendix B retains source locators.

Table 5. Contrasting signal-processing configurations and the requirements they expose. Reported widths are not assumed to specify all intermediate arithmetic. These cases are a targeted comparison, not a frequency sample or a cross-platform performance ranking.

Configured Case	Operations and Retained Data	Requirement and Evidence Boundary
8 × 8 DCT/IDCT [21]	Constant shift–add butterflies; row/column transpose; twelve-bit row output.	Preserve direction, scale and bank layout. Approximate-image evaluation and FPGA implementation do not prove codec conformance.
AV1 encoder fractional motion estimation (FME) [23]	Four-, six- or eight-tap separable filters; reference window and intermediate rows.	Tap removal changes sample demand and coding quality. Synthesis power; external traffic depends on caching.
Chirp-scaling SAR [41]	FFT/IFFT plus phase multiplication; reordered transpose and phase-function storage.	Shared multiplier requires the demonstrated schedule. Double-data-rate (DDR) memory word width does not identify all arithmetic formats.
Polar-format SAR [24]	FP32 coordinate/ratio computation, FP16 interpolation data and FFT; DDR transfers.	Conversion and interpolation remain part of the task. Dataset execution time does not establish an airborne deadline.

Table 5. *Cont.*

Configured Case	Operations and Retained Data	Requirement and Evidence Boundary
Distributed-arithmetic LMS [35]	Q1.3 coefficients and inputs; mutable partial-product memories; shift-accumulation.	Preserve coefficient/lookup-table (LUT) update order and memory ports. Simulation and synthesis, not measured acoustic quality.
Narrowband adaptation [42]	FFT and peak selection; sinusoidal references; persistent coefficients and secondary-path state.	Sensor rate, FPGA clock and feedback response are distinct. Physical sensing; piezo feedback remains future work.
Early radar detection [44]	Streaming FFT, sums or insertion sorting, threshold and alert; parallel higher-order path.	Continuous readiness is an execution requirement. Measured pipeline interval excludes sensor and actuator response.
Integrated frequency-modulated continuous-wave (FMCW) chip [45]	FFT, CFAR, CORDIC; four spectral SRAMs and compact target/angle memories.	Single-port schedule and target capacity constrain sharing. Chip tested with simulated input; frame time calculated.
Nonlinear-optimization correlative scan matching (NLO-CSM) [46]	Transforms, four-cell gathers, gradients, small matrix accumulation; map and point buffers.	Separate coarse/fine modes and host solve. FPGA demonstration and ASIC layout have different evidence maturity.

4.3. State Estimation and Map Maintenance

Estimator structure determines both numerical work and retained state. LiDAR–inertial odometry via smoothing and mapping (LIO-SAM) [47] uses local-keyframe factor graphs, FAST-LIO [48] reduces inversion to the state dimension, and FAST-LIO2 [49] adds direct registration with a mutable ikd-Tree [50]. BLITZCRANK [51] establishes factor-graph acceleration as prior art, but a new graph still needs explicit dimensions, elimination order and storage layout.

Map work changes the timing requirement. FAST-LIO2's 24-state, 756-point configuration averages 1.82 ms on Intel and 5.23 ms on ARM with different thread counts; ARM occasionally exceeds the 10 ms scan period. FAST-LIVO2 [31] combines a 19-dimensional state, voxel hash/local octrees and three-level 8 × 8 image patches. Its central processing unit (CPU) experiment runs estimation/planning at 10 Hz and MPC at 100 Hz; the mean 80.4 ms per 100 ms window includes ten MPC calls. Both examples require map-access and observation-age accounting beyond matrix latency.

Table 6 compares numerical formulations. Square-root marginalization [52] uses QR decomposition into orthogonal and upper-triangular factors and avoids the tested conventional FP32 failures while retaining a small LDL^{T} camera solve; its 36% runtime advantage compares square-root FP32 with conventional 64-bit floating point (FP64). OpenVINS [53] provides the covariance-filter context for the square-root visual–inertial navigation system SR-VINS [54] and its later $\sqrt{\text{VINS}}$ development [55]. These related implementations support joint tests of triangular operations, state layout and conditioning.

Table 6. Estimator configurations that change the meaning of a shared matrix capability. Timing values refer to the stated configurations and measurement boundaries; they are not normalized across platforms.

Configuration	Operations and Retained State	Numerical or Timing Evidence	Requirement Implication
FAST-LIO2 [49]	Point residuals, state update; covariance and mutable point map.	Reported ARM mean 5.23 ms, occasional > 10 ms; word width not established.	Bound map-access delay and state age separately from matrix execution.
Square-root marginalization [52]	QR/nullspace elimination; square-root prior; small LDL^{T} solve.	FP32/FP64 dataset comparison; desktop execution.	Preserve rank handling and conditioning at each phase.
SR-VINS [54]	Permuted QR and triangular updates; clones and optional landmarks.	FP32 multi-state constraint Kalman filter (MSCKF)-only mean 1.1 ms; front-end excluded.	Fix state configuration before comparing arithmetic or memory demands.
$\sqrt{\text{VINS}}$ [55]	Cholesky LL^{T} (LLT) update plus QR propagation/ marginalization; initialization workspace.	Laptop and Jetson evaluated separately; same lineage as SR-VINS.	Distinguish startup, steady operation and platform-specific budgets.

A RISC-V partition can use vector arithmetic for regular updates while software coordinates indexed map mutation and ordered access. The record must still specify matrix structure, rank handling, precision, factor lifetime and trajectory/consistency error. This is an inferred partition, not a demonstrated ISA mapping.

4.4. Learned Perception and Action Generation

4.4.1. Convolution, Attention and Live Storage

The inference examples expose three distinct storage decisions. DORY [17] tiles INT8/INT32 convolution and overlaps transfers. The microcontroller unit (MCU) implementation of MCUNet [56] overwrites completed depthwise-convolution channels only after their consumers finish. MCUNetV2 [57] trades patch overlap for lower peak storage. On-Device Training [58] further changes tensor lifetimes through sparse backward graphs and interlaced updates, introducing an adaptation phase distinct from inference.

Attention additionally needs reductions, rescaling and nonlinear arithmetic. The Integer Transformer Accelerator (ITA) [26] couples INT8 tiles to softmax and wider accumulators, with 64 KiB SRAM, changing its post-layout estimate from 16.9 to 8.46 trillion operations per second per watt (TOPS/W). Tiny Transformers [59] shows why cold-start weight transfer and intermediate storage belong in the same record: hot-start timing excludes the initial transfer, while its separate enlarged-memory microbenchmarks are simulations. Integer-only Bidirectional Encoder Representations from Transformers (I-BERT) [60] and the integer-only vision transformer I-ViT [61] keep wider nonlinear paths; I-ViT combines INT8 products with INT32 accumulation and a fixed ten-iteration square-root routine. SOLE [62] buffers reduction statistics and nonlinear intermediates in a register-transfer-level (RTL) evaluated design.

Storage reuse also has different validity rules. FlashAttention [63] retains exact within-call maxima, normalizers and partial outputs; the small hls4ml design [64] instead illustrates explicit first-in, first-out (FIFO) buffering and partitioned key/value storage for a 50-token,

16-dimensional, 3244-parameter example without LayerNorm. FINN-T [65] buffers both runtime attention operands and mixes FP32 softmax functions with integer reductions. Its substituted normalization/activation functions and pipeline throughput must be distinguished from an autoregressive query.

MINOTAUR [66] supplies silicon evidence for a different numerical/storage configuration: 8-bit posit products, decoded 16-bit posit accumulation, 12 MB resistive random-access memory (RRAM) and 2 MB SRAM. Low-rank adaptation (LoRA) keeps trainable matrices in SRAM and fixed weights in RRAM; checkpointing exchanges storage for recomputation. Its power boundary excludes input/output (I/O).

4.4.2. Action Representation, Quality and Response Timing

RT-2 [67] introduces vision–language–action tokens, whereas π_0 [68] couples a vision–language backbone to flow matching. Action Chunking with Transformers (ACT) [69] predicts and aggregates overlapping chunks; its four-view, 14-joint, 100-action configuration omits the training encoder at inference. Its approximately 0.01 s RTX 2080Ti policy timing is distinct from motor-control timing.

Two counterexamples make action representation a hardware requirement. FAST [22] reduces a 14-dimensional, 50 Hz one-second chunk from 700 naive tokens to 53 on average through temporal DCT, quantization and byte-pair encoding. Yet autoregressive π_0-FAST takes approximately 750 ms per chunk versus 100 ms for diffusion π_0 on RTX 4090. OpenVLA-OFT [70], an optimized fine-tuning method for a vision–language–action (VLA) model, generates eight seven-dimensional actions in 72.9 ms on A100 for single-image LIBERO, versus 25-action chunks in 321 ms with three images and 14-dimensional state on ALOHA. Generated-action rates do not measure fresh-observation response. Its separate one-step diffusion ablation takes 73.1 ms but achieves zero LIBERO-Long success.

Iterative generation changes both dependencies and state lifetime. Diffusion Policy [71] retains visual conditioning while denoising; its real Push-T configuration has two observations, a prediction horizon of 16, an action horizon of 6, and 16 inference steps. The separate ten-step, approximately 0.1 s RTX 3080 example must remain a different record. Real-time chunking (RTC) [27] adds inference-time vector–Jacobian guidance and asynchronous chunk replacement. Its five-step brain floating-point 16 (BF16) result is a mean latency of 97 ms on the graphics processing unit (GPU) alone versus 76 ms for the base policy on RTX 4090; host/network overhead and sub-timestep synchronization require separate accounting.

Precision and cached observations must be evaluated at the task level. OpenVLA [72] runs at 1.2 Hz with INT8 and 3 Hz with 4-bit integer (INT4) arithmetic on A5000, against a 5 Hz nonblocking collection controller; conversion overhead contributes to the INT8 slowdown, while blocking tests obtain comparable success across precisions. VLA-Cache [73] demonstrates the cost of stale state: indiscriminate reuse lowers LIBERO-Spatial success from 84.4% to 74.2%, and adaptive reuse recovers 83.8%. Within-call exact tiling and cross-observation approximation therefore need different acceptance tests.

4.4.3. Foundation Models and Edge Generative Execution

These foundation-model policies motivate separate visual-encoding, token/trajectory-generation and action-delivery modes. An edge design needs weight residency, conversion and nonlinear support, chunk timestamps and an observation-triggered invalidation policy; sequential decoding or stale conditioning may dominate despite high matrix throughput. The cited GPU results are workload evidence, not evidence of embedded-device timings. For a RISC-V host, descriptors should expose tensor format/shape, persistent-context identity and completion at a useful action boundary. Software/vector fallbacks for guidance

derivatives and chunk management belong in the same response-time measurement. Table 7 summarizes the resulting requirements.

Table 7. Contrasting learned-computation requirements. Implications are derived design conditions, not measured cross-paper hardware-sharing results.

Configured Example	Operation and State Condition	Requirement and Evidence Boundary
MCUNet/MCUNetV2 [56,57]	Channel overwrite versus patch overlap and repeated convolution.	Prove overwrite legality; budget halo traffic. Distinguish on-device inference from analytical storage profiles.
I-ViT/SOLE [61,62]	Wider accumulation, live statistics, integer square root or compressed nonlinear intermediates.	Specify reduction completion, rescaling and approximation error. GPU measurement and RTL estimates are different evidence levels.
FlashAttention/VLA-Cache [63,73]	Exact within-call statistics versus approximate reuse across observations.	Require invalidation and quality tests for temporal reuse; similar arithmetic operations do not establish equivalent cache behavior.
ACT/Diffusion Policy [69,71]	Overlapping action chunks versus repeated denoising with retained visual conditioning.	Declare action horizon, observation age and repeated versus once-per-call work. Policy timing does not specify motor timing.
OpenVLA [72]	Lower weight precision changes memory traffic and conversion overhead.	Measure response timing and task success together; bit width alone does not predict useful throughput.

4.5. Model-Based Optimization and Robot Dynamics

4.5.1. Solver Phases and Retained Factors

The useful shared operation depends on the online solver phase. High-Performance Interior-Point Method (HPIPM) [32] performs structured quadratic programming (QP), factorization and refinement; its 24-state, three-control, 40-stage benchmark excludes condensing and sequential quadratic programming (SQP) overhead. Operator Splitting Quadratic Program (OSQP) [33,74] can retain direct-solver factors while its Karush–Kuhn–Tucker (KKT) matrix and relevant parameters remain unchanged. TinyMPC [75] instead caches Riccati quantities so the online alternating direction method of multipliers (ADMM) uses matrix–vector passes, projections and dual updates.

Table 8 contrasts the associated state and measurement boundaries. The acados [76] engine-control hardware-in-the-loop (HIL) case includes integration, sensitivities and QP work: observed maxima stay below 10 ms within a 50 ms period, with spikes at reference/constraint changes. TinyMPC's physical-flight configurations separate 100 Hz observation from 500 Hz tracking control or 100 Hz obstacle-avoidance control. MPCGPU [77] solves a block system with preconditioned conjugate gradients (PCG), making synchronized reductions and convergence central. In its 128-knot simulated-plant case, many solves beat the CPU baseline, but the slowest observed PCG solve exceeds the baseline's slowest.

Context switching and interference must therefore be evaluated with workspace residency, iteration policy and control quality.

Table 8. Different optimization phases require different capabilities even when their equations contain matrix products. The final column distinguishes demonstrated execution from an architectural inference.

Example	Work That Must Be Identified	Persistent Data	Evidence and Sharing Boundary
acados HIL [76]	Integration and sensitivities, real-time iteration and QP solution.	Integrator and solver workspaces; previous iterate.	Embedded HIL timing. A QP-only speedup does not account for the complete iteration.
TinyMPC [75]	Offline cache formation; online matrix–vector passes, projections and dual updates.	Gains, cost factors and penalty-specific caches; trajectory/slack/dual vectors.	MCU benchmarks and separate physical flight. Reuse requires a valid cache configuration.
MPCGPU [77]	Block-system construction, PCG reductions and parallel line search.	Warm start, block system, preconditioner and search vectors.	Measured GPU execution with simulated plant; variable iteration time and synchronization matter.

4.5.2. Robot Dynamics and Topology

Robot structure determines the schedule of otherwise small products. Robomorphic [19] uses 6×6 spatial matrices and 32-bit fixed point with 16 fractional bits. Dynamics-gradient work and RoboShape [78,79] expose topology-dependent parallelism, implemented in GRiD's generated kernels [80]. Its iiwa single-call mean is 2.9 µs on i7-10700K versus 12.9 µs on RTX3080 before GPU I/O; even batches of 256 spend 53–71% of total GPU time on transfers across the tested robots.

Dadu-RBD [81] reuses dynamics submodules and FIFO/feedback intermediates, with fixed/floating-point conversion for reciprocals. Function changes alter dataflow; robot changes require reconfiguration. Its reported IIWA derivative latency is 0.76 µs versus Robomorphic's 0.61 µs despite stronger batch throughput; these are implementation/tool-evaluation results. Table 9 separates such function reuse from topology flexibility. A candidate must budget retained context and single-request latency alongside throughput.

Table 9. Dynamics evidence separates reusable mathematical structure from the cost of executing and retaining it.

Mechanism	Reusable Component	Constraint on Reuse	Evaluation Boundary
GRiD [80]	Small spatial products and analytic-gradient subexpressions.	Topology-specific dependencies, shared-memory occupancy, batch size and CPU/GPU transfers.	Measured software kernels for models with 7, 12 and 30 degrees of freedom (DOF).
Dadu-RBD [81]	Dynamics submodules and stored intermediates across related functions.	Fixed robot configuration, forward/backward dependencies and feedback buffering.	FPGA implementation/tool evaluation; power estimated; physical robot benefit unestablished.
RoboShape [79]	Topology traversals and topology-derived matrices.	Morphology is an input to accelerator design.	Contextual prior art; no quantitative result re-extracted here.

4.6. Actuator Control and I/O Synchronization

4.6.1. Prediction, Selection and Output State

Physical control attaches a timing contract to candidate selection. Sun et al. [30] evaluate seven voltage vectors for a permanent-magnet synchronous motor (PMSM), averaging four 256 kHz measurements into a 64 kHz loop. Their field-oriented control (FOC) comparator runs at 16 kHz, so settling differences do not isolate algorithm choice at equal rates. Arithmetic formats and worst-case calculation time remain unresolved.

Zafra et al. [82] provide a matched-quality tradeoff for finite-control-set MPC (FCS-MPC): 16-bit fixed-point transforms and predictions select among eight switch states, retaining the result until the next sample. At 40 kHz, parallel/sequential calculations take 50/400 ns with 0.7% current total harmonic distortion (THD) in both physical inverter tests; synthesis uses 200/70 digital signal processing (DSP) cells. The smaller sequential design fits the 25 µs period, but acquisition and output holding remain outside calculation timing. Longer-horizon sphere decoding [83] is contextual evidence that candidate search can instead become integer least squares.

4.6.2. Feedback and Waveform Timing

FOC also requires synchronized operands and outputs. Only Mishra et al.'s shortest schedule [84] fits the 10 µs interval at 100 kHz on a 400 W PMSM; mixed-controller work [85] emphasizes processor/peripheral exchange. Intel AN 1000 [86] documents fixed-point transforms, proportional–integral (PI) control and synchronized analog-to-digital converter (ADC)/pulse-width modulation (PWM) interfaces, including 64 kHz hardware FOC. These are vendor configurations, not independent deployment evidence.

Yin et al. [36] connect waveform lookup, comparison and dead-time counters to a RISC-V host. Their 413-fold result compares software pulse-width calculation/interrupt initialization with register configuration, rather than closed-loop control. Table 10 collects the state and interface requirements that remain after arithmetic is accelerated.

Table 10. Physical control evidence and the conditions it places on proposed sharing.

Configuration	Arithmetic and State	Demonstrated Boundary	Condition for a Shared Resource
Inverter FCS-MPC [82]	Fixed-point transforms/predictions; cost minimum; previous and pending switch states.	Physical resistive–inductive (RL) load experiment; calculation timing and synthesis utilization reported separately.	Preserve scaling, selection and next-sample application; include acquisition and transfer in the schedule.
PMSM FOC [84]	PI integration, coordinate conversion and scheduled intermediates.	Physical motor tests at several sample rates; arithmetic width not extracted.	Operands and outputs must be ready within the selected synchronization schedule.
Mixed PMSM controller [85]	Processor computation with hardware-managed I/O timing.	Contextual interface evidence.	Include interconnect and peripheral service time, not only arithmetic latency.
Long-horizon FCS-MPC [83]	Integer least squares and sphere search.	Context only; configuration and timing unassessed here.	Do not assume equivalence to a fixed eight-state argmin.

4.7. Integrated Execution and Resource Sharing

The integrated examples provide evidence of reuse within specified formulations. ORIANNA [14] assigns 74/16/10% of drone-example latency to QR decomposition, equation construction and back substitution. Its extension [87] shares a block-sparse pipeline across localization, planning and control: factor computation overlaps elimination, but back substitution waits for triangularization. FPGA power is estimated in both studies.

Hierarchically Composed Processing for Efficient Robot Learning-Based Control (HiPER) [88] combines neural and model-based FlowMPPI phases through FP16 processing elements, hierarchical scheduling and distinct communication paths. Its horizon-40 configuration has 4 kB local SRAM per processing element and 2 MB global SRAM, with networks loaded before runtime. Cycle-accurate simulation and RTL synthesis evaluate static kernel orders/accesses, excluding the dynamic tree-search and mapping examples discussed by the authors.

Dong et al. [89] demonstrate scheduled coexistence of high-priority 50 Hz MPC and background DroNet on a RISC-V Vector (RVV) core: control occupancy falls from 28.5% to 3.3%, with DroNet at 7.7 frames/s in the 100 MHz scenario. The HIL artifact uses 20 MHz silicon with physics slowed to 0.2 times the real-time rate; a universal asynchronous receiver/transmitter (UART) transfer affects the response, processor power is measured, and rotor power is modeled. Architecture comparisons use RTL timing and synthesized area. This is stronger evidence than the six cross-paper hypotheses below, while not measuring both tasks' use of one particular arithmetic circuit.

4.8. Quantitative Configuration Matrix

Table 11 consolidates eleven configurations across the seven workload subsections. "NE" means not established in this extraction, rather than absent from the source. Each row distinguishes operand/accumulator formats, storage/traffic scope and timing maturity; unlike systems are not ranked.

Table 11. Quantitative requirements for illustrative configurations. Values are source-reported unless marked as derived; NE denotes an unresolved extraction field. Timing and maturity are attached to each row.

Configuration and Dimensions	Operands/Accumulation	Storage, Traffic and Live State	Timing Boundary and Evidence
ECHOES audio FFT [20]; maximum 512/1024/2048 points	Fixed-point complex C64/C32/C16: 32/16/8 bits per component; full intermediate widths NE.	A maximum-length input vector occupies 4 KiB in each mode (derived); excludes twiddles, outputs and buffering. Reordering state is required.	Matched latency NE here; measured silicon supports these format/capacity pairs.
Radar FFT chain [18]; 256 × 256 timing case	Half-precision floating point; accumulator details NE.	Stage buffers and FFT ordering; full working set and transfer bytes NE.	Full chain 3.00 ms versus 23.89 ms with FFT-only acceleration; FPGA evaluation.
LiDAR scan matching [25]; at most 512 points, 320 × 320 map	32-bit fixed-point range/angle pairs; 6-bit stored occupancy; score accumulation NE.	One map's packed value payload is 75 KiB (derived), excluding alignment, metadata and other buffers. Scan and matching state are separate.	FPGA board execution; matching-call latency and transfer bytes NE here.

Table 11. *Cont.*

Configuration and Dimensions	Operands/Accumulation	Storage, Traffic and Live State	Timing Boundary and Evidence
SR-VINS [54]; 15 clones, 200 tracks	FP32 arithmetic; distinct accumulator representation NE.	Square-root covariance, clone/track state and QR workspace; byte footprint NE.	Mean 1.1 ms for MSCKF-only update; software evaluation excludes feature tracking.
DORY neural convolution [17]; MobileNet-v1, width 1.0, 128 × 128 input; layer-dependent tiles	INT8 operands/INT32 accumulation.	Tiled activations and weights; overlapping transfers. Per-layer footprint and traffic must be specified.	Configuration-matched latency NE here; GAP8 deployment evidence.
MCUNetV2 memory example [57]; 224 × 224 input, first five blocks in 4 × 4 patches	Exact arithmetic fields NE for this analytical example.	Peak activations 1372 to 172 kB; 10% more total MACs. Halo overlap causes recomputation.	Analytical profile, not a measured board latency.
RTC flow policy [27]; five denoising steps	BF16 execution; internal accumulator widths NE.	Conditioning, guidance intermediates and overlapping action chunks; peak bytes/traffic NE.	Mean GPU-only 97 ms versus 76 ms baseline on RTX 4090; excludes host/network overhead.
HPIPM MPC [32]; 24 states, 3 controls, 40 stages	Double-precision benchmark; independent accumulator width NE.	Partially condensed matrices, factors and solver workspace; byte footprint NE.	QP benchmark excludes condensing and SQP overhead; CPU software evidence.
Robomorphic dynamics [19]; 6 × 6 spatial products	32-bit fixed point, 16 fractional bits; complete accumulation policy NE.	Topology-specific intermediates and recursive state; workspace bytes NE.	FPGA execution; simulated power and ASIC estimates are separate evidence.
Inverter FCS-MPC [82]; 40 kHz sampling	16-bit fixed-point variables; complete binary-point and accumulator details NE.	Previous/pending switch state; state bytes and transfer traffic NE.	Measured calculation 50/400 ns; both 0.7% current THD in physical RL-load tests.
HiPER FlowMPPI [88]; horizon 40	FP16 processing elements; separate accumulator fields NE.	4 kB local SRAM per processing element and 2 MB global SRAM; networks fit on chip after host loading.	Cycle-accurate simulation and RTL synthesis; measured-silicon latency NE.

The matrix yields concrete design constraints without inventing a common performance denominator. The LiDAR payload is derived as 320 × 320 × 6/8 = 76,800 bytes (75 KiB); it is not the complete allocated memory. ECHOES' three maximum-length input vectors each occupy 512 × 64/8 = 1024 × 32/8 = 2048 × 16/8 =4096 bytes, yet support different precision and lengths. MCUNetV2 reduces peak activation storage by approximately 8.0× for its stated example while increasing arithmetic work by 10%; the ratio is derived from 1372/172. The inverter's sequential calculation takes 0.4 µs within a 25 µs sample period, leaving 24.6 µs before accounting for acquisition and other work. These within-configuration calculations guide capacity, mapping and scheduling choices; they are not new experiments or cross-platform speedups.

Section 5 applies these distinctions to six pairwise comparisons and identifies the evidence needed before a proposed cross-domain sharing design can be accepted.

5. Worked Comparisons for Requirement Definition

5.1. Six Pairwise Compatibility Assessments

Table 12 summarizes six literature-derived compatibility hypotheses and their acceptance conditions. Appendix C retains the source configurations and case-specific checks. The worked examples below explain the principal numerical, state and scheduling decisions.

Table 12. Conditional compatibility cases. Outcomes apply only to the named capability and constraints; no row demonstrates physical sharing or a measured benefit from it.

Case	Compared Configurations	Assessment and Scope	Missing Proof or Acceptance Check
PC01	Audio and radar FFT modes [18,20]	Adaptation. The 4096-point radar mode exceeds all reported audio-engine capacities; numerical requirements also differ.	Matched length and quality; conversion, storage and ordering; full pipeline and concurrent service demand.
PC02	Quantized inference and hanging-chain MPC [17,32]	Adaptation. INT8 inference support does not establish double-precision factorization or solve support.	Numerical validation, condensed matrix shapes and timing that includes omitted condensing and SQP phases.
PC03	Video SAD search and N = 16 CFAR [28,29]	Adaptation. Reductions and extremum selection do not replace partial sorting and its state.	Intermediate formats, reduction lengths and phase schedule; retain the N = 16 timing boundary.
PC04	Q1.3 LMS and quantized CNN [17,35]	Adaptation. The LMS coefficient/LUT update and four-bit numerical format differ from tiled INT8 inference with INT32 accumulation.	Intermediate rounding and scaling, coefficient writeback, convergence, transfer cost and a concurrent schedule.
PC05	Attention tiles and robot dynamics [19,26]	Adaptation. Small structured products, numerical requirements and recursion constrain reuse of a tiled product datapath.	Conversion quality, mapped utilization, dependency-preserving execution and context/transfer cost.
PC06	Video minimum and LiDAR maximum selection [25,28]	Conditional functional match only for a configurable score/index selection block. Score generation and addressing remain separate.	Score widths, overflow and tie rules; context isolation, candidate arrival rates and a validated schedule.

The six cases comprise five adaptation judgments (PC01–PC05) and one conditional functional match (PC06): 5/6 and 1/6, respectively. Validated concurrent cross-domain physical-sharing designs number 0/6. These are descriptive counts of the selected comparisons, not estimates of compatibility across the literature. Table 13 separates these hypotheses from reuse actually reported within reviewed systems.

Table 13. Compatibility hypotheses and demonstrated reuse have different evidence boundaries.

Evidence Set	Supported Statement	Boundary Retained
PC01–PC05 (five cases)	Identified adaptation for a proposed capability.	No mapped joint design or demonstrated concurrent benefit.
PC06 (one case)	Conditional functional match for score/index selection.	Score semantics, arrival demand and arbitration still require tests.
ORIANNA [14]	Accelerator generation for optimization-based robotic applications.	Design reuse within its evaluated framework; not validation of PC01–PC06.
HiPER [88]	Reuse within a learning-based control execution architecture.	Cycle-accurate simulation and RTL synthesis; no measured-silicon claim.
Shared-core control/inference [89]	Evaluated scheduling and competing demand in the reported system.	Its hardware and simulation boundaries do not establish a new four-domain accelerator.

PC01's 4096-point capacity requirement is distinct from the radar source's 256 × 256 pipeline timing.

5.2. Audio and Radar FFT

PC01 requires added arithmetic or validated conversion/scaling between ECHOES' fixed-point modes [20] and the radar processor's half-precision arithmetic [18], together with sufficient transform capacity. Acceptance covers signal quality, permutation and twiddle storage.

Rejecting the unchanged mapping does not reject fixed-point radar processing. Complete pipeline service under combined arrivals and deadlines is tested by the protocol in Section 6.3.

5.3. Adaptive Filtering and Neural Convolution

Mapping PC04's LMS and inference paths to a product/reduction unit requires defined scale, accumulation and rounding, with coefficient writeback and separate delay-line context. Retaining distributed arithmetic instead requires coherent lookup-table updates. These concrete state requirements replace the under-specified audio example [16] that was retained briefly in Appendix C. Acceptance requires filter convergence, complete sample-service timing and measured context/transfer costs.

5.4. Control and ML Linear Algebra

For PC02, additional arithmetic or a validated reformulation must also preserve factorization/solve dependencies. Robomorphic Computing [19] adds a shape counterexample: short, morphology-dependent products require utilization measurements at their actual dimensions.

OSQP [33] supplies a separate prioritization rule. With an unchanged direct-solver KKT matrix, retained factors and repeated solves are the steady-state target; repeatedly accelerating factorization addresses the wrong phase. Matrix or relevant parameter updates invalidate this assumption. Numerical tolerance, mapped dimensions and phase frequency therefore belong in the record before selecting the execution unit.

5.5. Negative Cases and Unsupported Transfer Claims

Snitch [90] reports no benefit from adding floating-point repetition (FREP) to scaled vector addition $y \leftarrow ax + y$ (AXPY), because stores remain explicit. Its distance-only k-nearest-neighbor (kNN) timing, however, merely leaves complete kNN performance unresolved because sorting is excluded. Likewise, SPWM configuration time [36] is not closed-loop motor-control latency. The record distinguishes an observed failure to improve a measured kernel from an unsupported extrapolation to an unmeasured task.

6. A Conditional Requirements Framework

Table 14 translates reported conditions into candidate capabilities and validation requirements. The implications are engineering inferences; the cited conditions are source-reported observations.

Table 14. Example requirements. No row prescribes a universal hardware parameter.

Observed Condition	Conditional Implication	Validation Needed
FFT capacity in ECHOES [20] varies with precision	Treat format, length and storage jointly.	Quality and full timing for each mode.
DORY [17] includes transfers around neural arithmetic	Assess tiled storage and overlap alongside MAC support.	Shapes, traffic and mapping cost.
OSQP [33] reuses factors when its direct-solver KKT matrix is unchanged	Prioritize by setup/update frequency.	Update modes and retained-state cost.
CFAR [29] combines reductions and ordering	Preserve selection and phase scheduling.	Matching window and numerical semantics.
LiDAR scan matching [25] reduces indexed data	Assess address generation with arithmetic.	Working set and access distribution.
Robomorphic Computing [19] uses short spatial products	Test small-workload utilization and communication.	Application timing and precision.

A further integration example pairs RTC guidance [27] with predictive motor control [30]. The reported RTC configuration uses five BF16 denoising steps and a vector–Jacobian product; the controller evaluates seven voltage vectors from two current components at 64 kHz. A shared vector path would need policy-intermediate and observer-state retention, with bounded interference. Motor formats, joint execution times and concurrent schedules remain unresolved. This example motivates the partitioning rule below and is separate from PC01–PC06.

Service cost is distinct from application response time. A bookkeeping expression is

$$T_{\text{response}} = T_{\text{queue}} + T_{\text{configure}} + T_{\text{movement}} + T_{\text{compute}} + T_{\text{completion}}, \quad (1)$$

with overlapping intervals counted once. Setup is amortizable only while the retained context remains valid; the expression requires measured costs and a schedule before it predicts response time.

Each record links its originating comparison to either preservation of the reported requirements, an adaptation requiring a quality test, or an unresolved field. Table 3 determines acceptance, and Section 6.3 specifies the concurrent test.

6.1. Capability Taxonomy and Reference Architecture

Figure 2 consolidates the workload-to-requirement chain and a conceptual architectural template. Five capability families follow from the reviewed contrasts: products and reductions, transforms and permutations, indexed access and selection, nonlinear functions and dependent solves, and stateful timed I/O. Their limits are numerical/shape mismatch, format/capacity coupling, irregular traffic and ordering, dependency and quality constraints, and persistent-state/deadline obligations, respectively. The template is an engineering proposal derived from those requirements, not a synthesized or measured implementation.

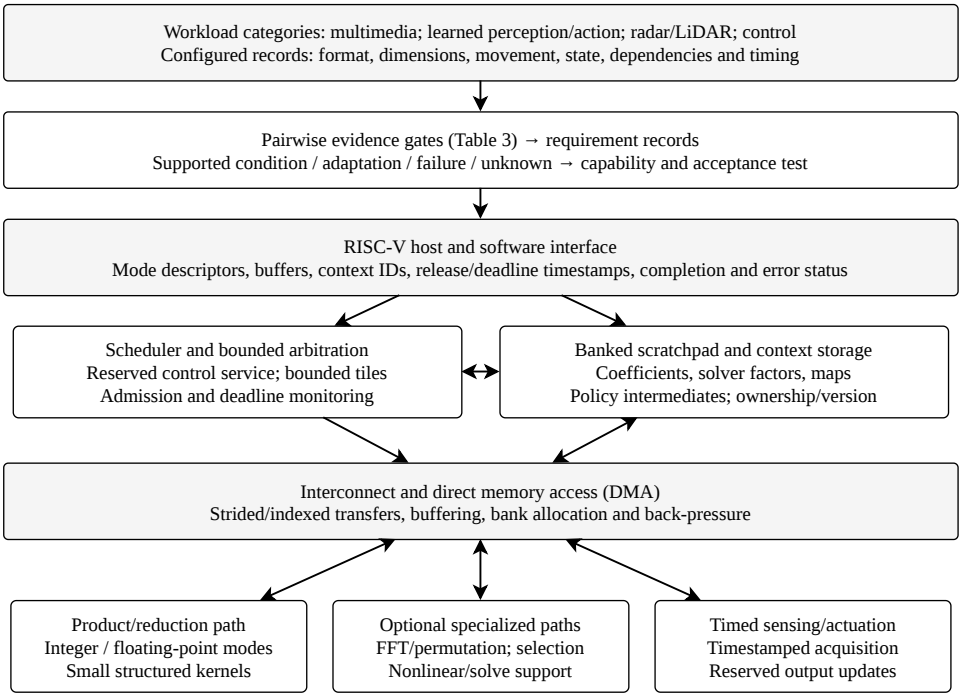

Figure 2. Proposed translation from evidence to architectural blocks. Numerical modes, capacities, lane counts and arbitration parameters are selected only after workload-specific validation. Arrows show logical control/data relationships, not a fixed physical bus topology.

Table 15 defines stream, inference and control/estimation modes. Descriptors identify shape, precision, buffer ownership, state version and completion semantics. Software saves or isolates mutable context at mode switches. Section 7 considers standard vector, custom-instruction and command-driven interfaces.

Table 15. Illustrative architectural modes and the conditions that decide their implementation.

Mode	Units and Retained Storage	Design Decision and Required Check
Stream	FFT/permutation, reductions, double buffers and twiddles.	Size storage jointly with format and length; validate 4096-point decomposition or use a separate engine.
Inference	Tiled product path, nonlinear support, weights and activation scratchpad.	Bound tile occupancy and traffic; test short shapes and context cost under control demand.
Control/estimation	Small products/solves, indexed loads, factors/maps/observer state and timed I/O.	Reserve service and state; validate numerical convergence and the complete sample-to-output deadline.

For example, the reported 64 kHz motor-control configuration [30] has a derived update period of $1/64{,}000 = 15.625$ µs. A candidate design adopting one update per period must budget acquisition, queuing, state transfer, and computation and output within its declared deadline; the period alone does not establish that budget. If an inference tile or DMA burst can block the control path beyond the remaining allowance, reduce its non-preemptible interval, reserve an independent bank/path, or retain a dedicated controller. This is a concrete partitioning rule; the present literature does not supply the joint execution times needed to select among those alternatives.

6.2. Timing Semantics from Observation to Actuation

Timing fields distinguish result availability from action application. For acquisition at t_s, the result-ready age is $A_r = t_r - t_s$, and age when action j is applied is $A_{a,j} = t_{a,j} - t_s$. Each sensor retains its timestamp and alignment procedure. Policy-call latency, invocation interval and actuator update interval describe different events; none alone measures observation age.

For a chunk executed at spacing Δ_a, age grows by $(K_{\text{exec}} - 1)\Delta_a$ from its first to last executed action. OFT's ALOHA configuration [70] executes 25 actions at 25 Hz, giving a derived growth of $(25 - 1)/25 = 0.96$ s. Acquisition, inference, queuing and first-action delay are additional. This schedule identity is not a measured total age; fresh low-level feedback is distinct from refreshing the policy observation.

Table 16 collects the timing notation; all timestamps must use a common clock or a documented alignment procedure.

Table 16. Timing notation and measurement boundaries.

Symbol	Meaning
T_{response}	Request-to-completion elapsed time at the declared interface.
T_{queue}, $T_{\text{configure}}$	Waiting for service and preparing the selected mode/context.
T_{movement}, T_{compute}, $T_{\text{completion}}$	Non-overlapping contributions of data movement, execution and completion; overlapped intervals counted once.
$t_s, t_r, t_{a,j}$	Acquisition time of the source observation, result-ready time and application time of action j.
A_r, $A_{a,j}$	Observation ages $t_r - t_s$ at result availability and $t_{a,j} - t_s$ at action application.
$K_{\text{exec}}, \Delta_a$	Number of actions actually executed from a chunk and their execution interval.
r_i, d_i, f_i	Release, absolute deadline and completion of request i in a replay or concurrent test.

6.3. Concrete Validation Workloads and Acceptance Protocol

Table 17 identifies concrete validation pairings, each retaining its source-specific numerical and task-quality requirements. These are proposed tests, not new experiments performed for this review.

Table 17. Concrete validation starting points for the proposed architecture. Freeze source version, inputs and configuration before comparison.

Workload or Pairing	Required Coverage	Acceptance Evidence
MLPerf Tiny keyword spotting/visual wake words [91]	Quantized inference; cold/warm weights; simultaneous sensing traffic.	Suite quality target, full inference latency and energy at the same boundary.
ECHOES and radar FFT configurations (PC01) [18,20]	Source-supported lengths and formats; an explicit 4096-point adaptation.	Signal/detection quality, storage/traffic, permutation and combined arrival service.
HPIPM hanging chain and OSQP repeated MPC [32,33]	24-state, 3-control, 40-stage HPIPM case; fixed and updated solver matrices.	Residual/constraint checks, setup versus repeated solve cost, state invalidation.
LiDAR scan matching with video selection (PC06) [25,28]	Up to 512 points, 320 × 320 map; score/index streams with ties and overflow.	Reference winners, map-update consistency, backpressure and worst observed response.
RTC policy plus predictive motor control [27,30]	Policy intermediates and observer state; burst arrivals and controller service.	Task/control quality, observation age and deadline misses under joint load.

For every pairing, first freeze the numerical reference, input traces, required quality and timing boundaries. Compare separate resources with time-multiplexed resources under identical inputs and quality targets; include a validated adapted version only when its conversion and state costs are charged. Exercise cold starts, maximum admitted dimensions, state updates, coincident releases and context switches. Measure complete requests rather than only arithmetic kernels, and report hardware execution separately from simulation or synthesis estimates.

For each request i, record release r_i, completion f_i and absolute deadline d_i. A replay passes its deadline check only if $f_i \leq d_i$ for every admitted request; report the number of misses, the largest $f_i - r_i$, and the observation ages of applied actions. Zero misses in a finite trace is empirical evidence for that trace, not a worst-case guarantee. A hard-real-time claim additionally needs bounded arrivals, execution, transfer and blocking, plus a justified scheduling analysis for the selected arbitration policy. Report area or energy savings only after these quality and timing gates pass and both resource configurations are evaluated at the same implementation boundary.

7. RISC-V Implementation Perspective

Requirements can map to existing standard instructions, custom ISA extensions or attached accelerators. These routes differ in software and state management. The official specification [92] defines V version 1.0 and embedded Zve subsets, with arithmetic, reduction, memory and permutation mechanisms. The implemented subset and numeric support must be identified. The vector-register length in bits (VLEN) does not determine lane count, bandwidth or achieved throughput.

Research extensions require their semantics, version, encoding, state and compiler support to be recorded separately from ratified instructions [92]. A memory-mapped or command-driven accelerator is not automatically an ISA extension. Selection among these interfaces depends on setup amortization, state lifetime and data placement.

7.1. Vector, Matrix and Streaming Mechanisms

Table 18 pairs arithmetic with its data-delivery requirements. Ara2's width-dependent register rearrangement and scalar/vector coherence [93], and Spatz's memory-port lim-

its [94], show why lane arithmetic alone does not determine useful throughput. Evaluate actual shapes and access patterns, including transitions between workloads.

Matrix specialization changes that balance. Quadrilatero [95] uses a 4×4 array, a matrix register file and explicit matrix transfers. In its reported INT8-to-INT32 cases, utilization is 76.0% for 64 × 64 × 64 multiplication and 44.2% for 64 × 16 × 64. These are configured cycle results; the physical estimates use post-synthesis 65-nm analysis. The reduction dimension and store overhead matter even though both cases use the same arithmetic. Small state-estimation or control matrices should therefore be evaluated at their actual dimensions before selecting a large tensor datapath.

Snitch [90] evaluates floating-point kernels within 128 KiB of local shared memory using cycle-accurate RTL timing and post-layout power estimates. Its AXPY and distance-only kNN boundaries are discussed in Section 5.5. Sparse stream semantic registers (SSRs) [96] instead couple products to indirection and coordinate intersection/union, requiring compatible value/index widths, row pointers and stream ordering.

Table 18. Reusable mechanisms and the requirements each exposes. The rows describe evidence boundaries, not a performance ranking.

Mechanism	Reported Support	Requirement to Test
Ara2 [93]	RVV 1.0 arithmetic, width-dependent register handling and scalar/vector memory coordination	Format-switch cost, tail handling, memory ordering and access throughput.
Spatz [94]	Compact vector register files and shared multibanked memory	Register reuse, bank pressure and port demand for actual kernel shapes.
Quadrilatero [95]	Matrix registers, mixed integer widths and FP32 products	Reduction dimension, transfer schedule and utilization outside long inner products.
Sparse SSR [96]	Sparse indirection, intersection and union	Value/index widths, stream order, stride restrictions and mutable structure.
FPnew [97]	Configurable arithmetic, conversion and mixed-format operations	Rounding, accumulation, exception policy and conversion overhead.
Vicuna [98]	In-order integer/fixed-point vector execution based on RVV draft 0.10	Timing assumptions, interference and compatibility with the required ISA version.
Gemmini [99]	Spatial arrays, software tiling, scratchpad, wider accumulation and virtual memory	Host overhead, translation, data staging and shared memory contention.

7.2. Numerical Semantics and Predictable Integration

FPnew [97] exposes arithmetic, division/square root, comparison and conversion groups, including fused operations with different source and destination formats. Its merged versus parallel format slices make sharing overhead explicit. Specify accumulation, conversion, rounding and exceptional values along with operand width. IEEE 754 [100] defines floating-point behavior; the workload supplies the accepted error.

Vicuna [98] evaluates constant-control-flow kernels with an in-order organization and stated memory/cache assumptions. It implements historical RVV draft 0.10. Predictable integration requires a blocking/preemption model and bounded memory interference beyond the configured kernel evidence.

Gemmini [99] couples spatial arrays to tiling, scratchpad and accumulator storage, and address translation. Its FPGA analysis, synthesis and fabricated integrations have

distinct measurement boundaries. For a software reference, the Common Microcontroller Software Interface Standard (CMSIS) DSP library [101] supplies non-RISC-V DSP kernels; preserve numerical behavior and placement when comparing them.

7.3. From a Requirement Record to Architecture Exploration

Timeloop [13], ZigZag [102] and Accelergy [103] support mapping, memory and energy exploration. Flexibility spans tiling, ordering, parallelization and array shape [104]. The requirement record supplies constraints on numerical behavior, mutable state and response deadlines for these explorations.

An initial capacity check can use a Roofline-style bound [105] at a declared memory boundary:

$$P_{\text{attainable}} \leq \min(P_{\text{compute}}, B_\ell I_\ell), \tag{2}$$

where B_ℓ is sustainable bandwidth and I_ℓ is the number of useful operations per transferred byte at level ℓ. State the multiply–add counting convention and include index traffic, conversion and host work within the declared boundary. This throughput bound does not determine schedulability. Heterogeneous energy models must likewise include CPU and transfer activity [106].

Mapped candidates should then pass the quality and concurrent timing protocol in Section 6.3. This extends kernel-level exploration to the sensing-to-action boundary.

8. Design Implications and Research Gaps

8.1. A Shared Base with Explicit Adaptation Boundaries

A configurable base with selective nonlinear, sparse or timed-I/O specialization is therefore a candidate to evaluate against dedicated units. Measured demand, adaptation cost and task quality determine the partition.

8.2. Reusable Evaluation Workloads

Validation should include short reductions and small matrices alongside large products and fixed as well as changing sparsity. Information-age concepts [107] describe timestamped updates; the physical task supplies the permissible age.

8.3. Limits of the Available Evidence

The targeted selection supports mechanism comparisons, not prevalence or market-adoption estimates. Database coverage, preliminary-paper influence and the unassessed retrieval pool limit generalization (Section 3.1). The partial 2026 window and dated citation counts cannot establish full-year trends, evidence quality or deployment.

Missing intermediate formats, memory traces and response distributions restrict numerical and sharing judgments. Author manuscripts may differ from the version of record, and implementation maturity varies by claim. No independent user study or new hardware experiment was performed.

8.4. Future Work in Hardware, Software, Benchmarking and Codesign

Hardware work should instantiate the template in Section 6.1 and compare shared and dedicated units at matched quality. Vary arithmetic modes, scratchpad banks, context capacity and arbitration, measuring conversion, state-save and interference costs. Audio/radar transforms and inference/control provide complementary stress cases.

Software work should encode dimensions, formats, state lifetime, preemption points and deadlines in compiler/runtime descriptors. Mapping tools should expose setup and

context costs, reject failed gates and retain software fallbacks. Independent designers should repeat the six judgments to test reproducibility.

Benchmarking should extend the pairings in Table 17 with DORY layers [17]. Publish traces, numerical references, configurations and measurement boundaries with deadline misses and observation-age distributions under the protocol in Section 6.3.

Codesign should jointly vary policy chunk length/replanning, solver updates and hardware allocation. Compare task success or tracking quality at matched freshness and energy, retaining the low-level feedback loop.

9. Conclusions

This focused review connects 62 detailed primary-source records to seven workload discussions, eight operation groups and six pairwise compatibility cases. Five cases require adaptation, and one identifies a conditional functional match for score/index selection. None of the six establishes concurrent physical sharing. These are descriptive results of the selected comparisons, not estimates of operation prevalence or new hardware measurements.

The numerical contrasts lead to concrete requirements. The audio FFT engine's maximum lengths of 512, 1024 and 2048 depend on its respective 32-, 16- and 8-bit real and imaginary components; the radar engine's 4096-point half-precision mode lies outside that unmodified envelope. Precision and storage therefore belong in the same capability specification. Likewise, 25 executed actions at 25 Hz imply 0.96 s of additional observation age from the first to the last action in that chunk. This derived schedule quantity excludes acquisition, inference and first-action delay, demonstrating why action throughput cannot replace a freshness requirement.

The consolidated matrix and conceptual architecture translate these findings into configurable arithmetic, banked storage, explicit state retention, bounded arbitration and software descriptors. Dedicated nonlinear, indexed-access or actuator support remains a workload-dependent choice. The decision record supplies the numerical, task-quality and timing gates for evaluating those choices; matched concurrent implementations and the tests in Section 6.3 are the next step for establishing area, energy or latency benefits.

Author Contributions: Conceptualization, R.S. and G.N.; methodology, R.S. and G.N.; validation, R.S. and G.N.; formal analysis, R.S. and A.-L.P.; investigation, R.S., A.-L.P., H.-D.Z. and C.-C.D.; data curation, A.-L.P., H.-D.Z. and C.-C.D.; writing (original draft preparation), R.S. and A.-L.P.; writing (review and editing), R.S., A.-L.P., H.-D.Z., C.-C.D. and G.N.; visualization, A.-L.P., H.-D.Z. and C.-C.D.; supervision, G.N.; project administration, G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Union–NextGenerationEU, through the National Recovery and Resilience Plan (PNRR), Component 9, Investment 4, under project SENSE, THINK @ POLITEHNICA BUCUREȘTI (SENTHIPoli) No. 6.PI/I4/C9. The content of this material does not necessarily represent the official position of the European Union or the Government of Romania.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACT	Action Chunking with Transformers
ADC	Analog-to-digital converter

ADMM	Alternating direction method of multipliers
ASIC	Application-specific integrated circuit
AV1	AOMedia Video 1
AXI	Advanced eXtensible Interface
AXPY	Scaled vector addition, $y \leftarrow ax + y$
BD-BR	Bjøntegaard delta bit rate
BERT	Bidirectional Encoder Representations from Transformers
BF16	Brain floating-point 16
CFAR	Constant false alarm rate
CMSIS	Common Microcontroller Software Interface Standard
CNN	Convolutional neural network
CORDIC	Coordinate rotation digital computer
CSM	Correlative scan matching
CTU	Coding-tree unit
CUDA	NVIDIA GPU-computing platform
CVAE	Conditional variational autoencoder
DCT	Discrete cosine transform
DCT-II	Type-II discrete cosine transform
DDR	Double data rate
DOF	Degrees of freedom
DOI	Digital object identifier
DORY	Deployment Oriented to memoRY
DSP	Digital signal processing
DST-VII	Type-VII discrete sine transform
FAST	Frequency-Space Action Sequence Tokenization
FCS	Finite control set
FCS-MPC	Finite-control-set model predictive control
FFT	Fast Fourier transform
FIFO	First in, first out
FIR	Finite impulse response
FMCW	Frequency-modulated continuous wave
FME	Fractional motion estimation
FOC	Field-oriented control
FREP	Floating-point repetition
HIL	Hardware in the loop
HiPER	Hierarchically Composed Processing for Efficient Robot Learning-Based Control
HPIPM	High-Performance Interior-Point Method
I-BERT	Integer-only BERT quantization method
I-ViT	Integer-only vision transformer quantization method
IDCT	Inverse discrete cosine transform
IFFT	Inverse fast Fourier transform
INT4/INT8/INT32	4-/8-/32-bit integer representation
ISA	Instruction set architecture
ITA	Integer Transformer Accelerator
KKT	Karush–Kuhn–Tucker
kNN	k-nearest neighbors
LC3plus	Low Complexity Communication Codec plus
LiDAR	Light detection and ranging
LIO-SAM	LiDAR–inertial odometry via smoothing and mapping
LLT	Cholesky factorization $LL^\top$
LMS	Least mean squares
LoRA	Low-rank adaptation
LUT	Lookup table
MAC	Multiply–accumulate

MCU	Microcontroller unit
MDCT	Modified discrete cosine transform
MFCC	Mel-frequency cepstral coefficient
ML	Machine learning
MPC	Model predictive control
MSCKF	Multi-state constraint Kalman filter
NA	Not available
NICE	Nuclei Instruction Co-unit Extension
NLO-CSM	Nonlinear-optimization correlative scan matching
OFT	Optimized fine-tuning
OSQP	Operator Splitting Quadratic Program
PCG	Preconditioned conjugate gradients
PE	Processing element
PI	Proportional–integral
PID	Proportional–integral–derivative
PMSM	Permanent-magnet synchronous motor
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PWM	Pulse-width modulation
QP	Quadratic programming
QR	Orthogonal–upper-triangular matrix factorization
RL	Resistive–inductive
RRAM	Resistive random-access memory
RRT	Rapidly exploring random tree
RTC	Real-time chunking
RTL	Register-transfer level
RVV	RISC-V Vector
SAD	Sum of absolute differences
SAR	Synthetic aperture radar
SIMD	Single instruction, multiple data
SLAM	Simultaneous localization and mapping
SPWM	Sinusoidal pulse-width modulation
SQP	Sequential quadratic programming
SR-VINS	Square-root visual–inertial navigation system
SRAM	Static random-access memory
SSR	Stream semantic register
STT-MRAM	Spin-transfer-torque magnetoresistive random-access memory
THD	Total harmonic distortion
TOPS/W	Trillion operations per second per watt
TVM	Machine-learning compiler framework (project name)
UART	Universal asynchronous receiver/transmitter
UAV	Unmanned aerial vehicle
VINS	Visual–inertial navigation system
VLA	Vision–language–action
VLEN	Vector-register length in bits
VVC	Versatile Video Coding

Appendix A. Search Records and Retrieval Boundaries

The 24 generic-family queries used Web of Science Core Collection Topic fields, Scopus title/abstract/keywords and IEEE Xplore metadata. Figure A1 gives raw retrieval counts; the exact queries, filters and export reconciliation follow. The 2026 window is partial.

Normalized DOI identity alone collapsed repeated records: 12,480 occurrences comprise 12,098 DOI-bearing occurrences (7403 groups) and 382 separate no-DOI occurrences. Removing 4695 duplicate DOI occurrences leaves 7785 metadata groups. Flags retain 53 DOI

groups with title variants, 138 same-title clusters across groups and three date-boundary occurrences; they neither merge distinct identities nor establish eligibility.

The 15 separate RISC-V exports contain 3091 occurrences: 1318 DOI groups and 247 separate no-DOI occurrences yield 1565 candidates.

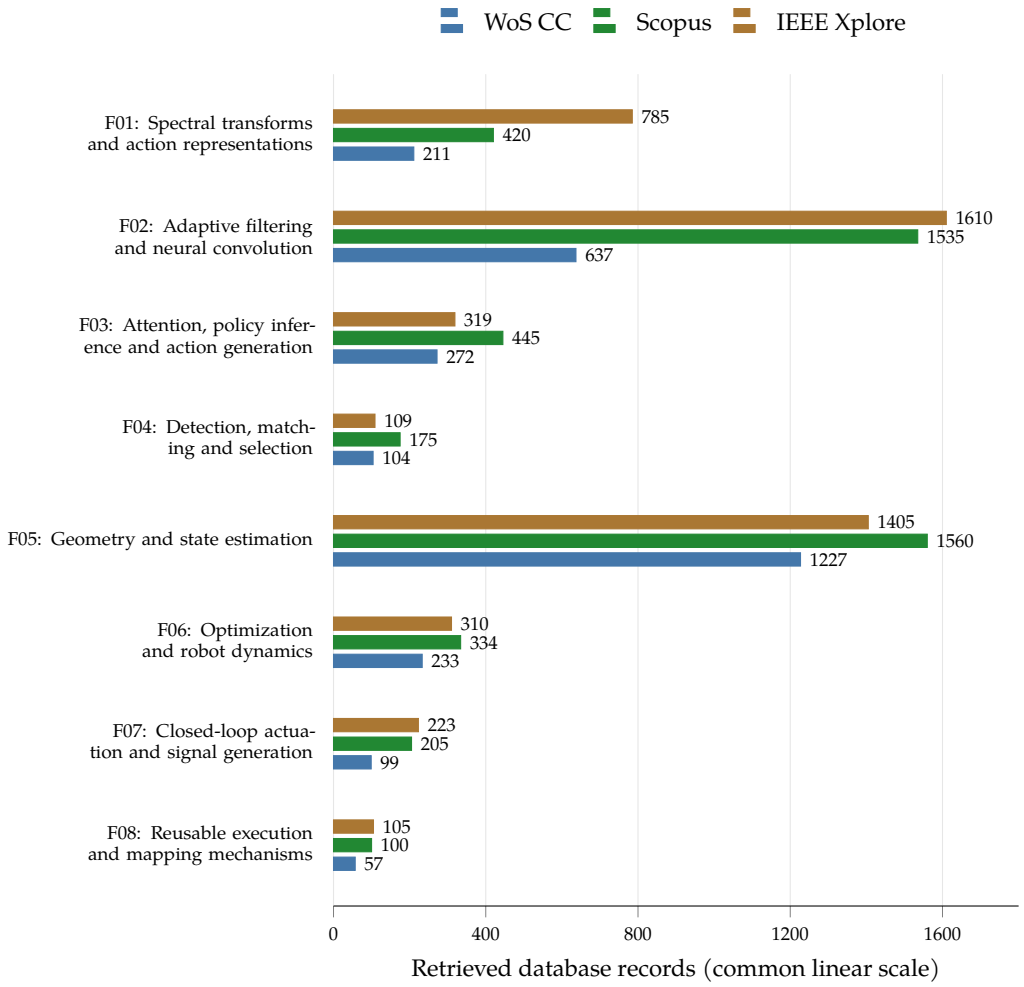

Figure A1. Raw retrieval counts for eight frozen query families across three databases, observed on 19 September 2026 within the 2020–2026 window. All 24 exports reconcile with the counts. Overlapping records, different search fields and the partial 2026 year preclude interpretations of study screening or influence.

Appendix A.1. Reproducible Generic Queries

The following factored representation preserves all terms and Boolean grouping of the 24 executed queries. The term SIMD denotes single instruction, multiple data. Within each numbered block, join quoted terms with OR and enclose the block in parentheses; join a family's blocks with AND. Let the resulting expression be Q. For Web of Science, execute `TS=(Q) AND PY=(2020-2026)`. For Scopus, execute `TITLE-ABS-KEY(Q) AND PUBYEAR > 2019 AND PUBYEAR < 2027`. For IEEE Xplore, prefix every quoted term with "All Metadata", enclose the complete expression in parentheses, and apply the 2020 to 2026 result-year filter. Q is a substitution variable, not a literal search term. Query identifiers follow G-DB-Fxx-01, where DB is WOS, SCOPUS or IEEE. Table A1 gives the exact term blocks, and Table A2 reports the corresponding result and export counts.

Table A1. Exact term blocks for the generic stream. Semicolons separate terms joined by OR; numbered blocks are joined by AND.

Family	Term Blocks
F01	1: "FFT"; "fast Fourier transform"; "discrete cosine transform"; "MDCT". 2: "audio"; "video"; "radar"; "action tokenization". 3: "hardware accelerator"; "hardware acceleration"; "FPGA"; "ASIC"; "embedded"; "microcontroller"; "RISC-V".
F02	1: "adaptive filtering"; "adaptive filter"; "active noise control"; "depthwise convolution"; "separable convolution". 2: "hardware accelerator"; "hardware acceleration"; "FPGA"; "ASIC"; "embedded"; "microcontroller"; "RISC-V".
F03	1: "vision-language-action"; "action chunking"; "policy inference"; "action tokenization"; "transformer inference". 2: "latency"; "real time"; "real-time"; "hardware"; "efficient"; "embedded".
F04	1: "CFAR"; "motion estimation"; "scan matching". 2: "hardware accelerator"; "hardware acceleration"; "FPGA"; "ASIC".
F05	1: "LiDAR-inertial"; "visual-inertial"; "LiDAR- inertial"; "visual inertial"; "factor graph"; "error state Kalman". 2: "hardware accelerator"; "FPGA"; "embedded"; "real-time"; "real time".
F06	1: "model predictive control"; "robot dynamics"; "robotic dynamics"; "inverse dynamics". 2: "hardware accelerator"; "hardware acceleration"; "FPGA"; "ASIC"; "RISC-V"; "GPU".
F07	1: "field oriented control"; "field-oriented control"; "motor control". 2: "hardware accelerator"; "hardware acceleration"; "FPGA"; "ASIC"; "RISC-V".
F08	1: "vector processor"; "SIMD"; "dataflow". 2: "embedded"; "robotics"; "RISC-V". 3: "matrix multiplication"; "linear algebra"; "FFT".

Table A2. Generic-stream result and export counts, 19 September 2026. Each displayed count equals the corresponding exported count.

Family	WoS Core	Scopus	IEEE
F01	211	420	785
F02	637	1535	1610
F03	272	445	319
F04	104	175	109
F05	1227	1560	1405
F06	233	334	310
F07	99	205	223
F08	57	100	105

Appendix A.2. RISC-V Queries and Retrieval Exceptions

The earlier RISC-V stream uses a domain block D and a required RISC-V block R. Domain terms are given in Table A3. Quote every domain term, join them with OR and enclose D in parentheses. For Web of Science and Scopus, R is `("RISC-V" OR "RISC V" OR RISCV)`. Execute `TS=(D AND R) AND PY=(2020-2026)` or `TITLE-ABS-KEY(D AND R) AND PUBYEAR > 2019 AND PUBYEAR < 2027`, respectively. For IEEE, prefix every term with `"All Metadata":` and quote RISCV as well; execute `D AND R`. The identifiers are R-DB-domain-01. All 15 runs were observed on 19 September 2026.

IEEE's year filter was verified for multimedia, AI/ML, radar and control. LiDAR had no added year filter: all 19 exports were dated 2022–2026. No language, open-access or

document-type restriction was reported. Table A4 preserves historical result/export counts; later indexing changes may alter reruns.

Table A3. Exact domain terms for the RISC-V stream; terms within each row are joined by OR.

Domain	Domain Block D
M (Multimedia)	`"multimedia"; "image processing"; "video coding"; "video compression"; "audio processing"; "audio coding"; "speech processing"; "audio denoising"; "active noise control".`
A (AI/ML)	`"machine learning"; "deep learning"; "neural network"; "neural networks"; "neural inference"; "tinyml".`
R (Radar)	`"radar"; "FMCW"; "CFAR"; "beamforming"; "range Doppler".`
L (LiDAR)	`"lidar"; "point cloud"; "point clouds"; "point-cloud".`
C (Control and actuators)	`"motor control"; "actuator control"; "model predictive control"; "robot dynamics"; "inverse dynamics"; "forward dynamics"; "state estimation"; "field oriented control"; "rigid body dynamics"; "pulse width modulation"; "SPWM".`

Table A4. RISC-V result and export counts, 19 September 2026. Displayed and exported counts agree; IEEE LiDAR retains the year-filter exception stated in the text.

Domain	WoS Core	Scopus	IEEE
M	51	95	78
A	674	1079	893
R	22	34	42
L	11	24	19
C	8	25	36

Appendix A.3. Selection Accounting and Calibration Limits

The 62 detailed records comprise 21 earlier anchors, 37 additions from the focused expansion, and four studies assessed after their initial deferral. The other 46 references provide context (Section 3.1). The expansion contributed nine detailed signal-processing records, ten AI/ML records, eleven control/estimation records and seven architecture records. These are editorial work allocations, not disjoint scientific domains or independent experimental replications.

During the initial focused expansion, the union of F01, F02 and F04 contained 3875 metadata groups for signal processing. Eighteen groups received an explicit candidate decision; 3857 remained unassessed by that expansion. Family-specific decision counts overlap and are not summed. At that stage, the AI/ML expansion retained ten detailed and five contextual references, deferred three candidates for access, and left one identified full text unassessed. Five of its retained references had an exact frozen-export match; primary tracing supplied additional candidates. The control/estimation assessment retained eleven detailed and nine contextual references and recorded fourteen other metadata-level decisions, including two possible no-DOI duplicates. The architecture assessment examined the 164 F08 group titles, without claiming a bulk full-text eligibility assessment. These different assessment units cannot be added to a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)-style screened-study total.

Full-text assessment subsequently added MINOTAUR [66], FINN-T [65], Hao et al.'s VVC transforms [40] and Liu et al.'s factor-graph accelerator [87]. Source locators are in Appendix B; these additions do not change historical query counts.

The fourteen calibration anchors were ECHOES [20], radar FFT [18], CFAR [29], LiDAR simultaneous localization and mapping (SLAM) [25], DORY [17], Snitch [90],

HPIPM [32], OSQP [33], Robomorphic Computing [19], VVC motion estimation [28], audio denoising [16], ITA [26], Tiny Transformers [59] and SPWM [36]. DOI lookup located all fourteen in WoS and Scopus and six in IEEE. The original WoS union recovered ten; adding terminology for audio denoising, active noise control, rigid-body dynamics, pulse-width modulation and generic hardware recovered fourteen. Scopus and IEEE revised unions recovered their fourteen and six DOI-located seeds. Because these seeds motivated the revisions, those outcomes are diagnostic and do not estimate unrestricted recall or precision. Preliminary novelty scouting and calibration are outside the two retrieval pools.

Appendix B. Sources and Their Relevance to the Synthesis

Table A5 gives source locators and evidence boundaries for all 62 detailed records. Numerical requirements are consolidated in Table 11; the common interpretation rules are in Sections 3.4 and 6.3.

Table A5. Source locators and configuration-specific evidence boundaries.

Source	Key Sections	Relevance to the Synthesis
ORIANNA [14]	Sections 5.2 and 6.1.	Four-application FPGA optimization prototype. Latency and estimated post-layout power have separate boundaries. Arithmetic widths, buffer capacities and deadline distributions remain unresolved; reuse is demonstrated within this framework.
Audio denoising [16]	Sections 3.1–3.3, 4.1, 5.1 and 6.2.	E203/Nuclei Instruction Co-unit Extension (NICE) integration and MCU200T acoustic experiments cover filtering and coefficient updates. Reduction is in instruction count. Tap counts, complete formats and sample deadlines remain unresolved.
DORY [17]	Sections 3.1, 3.2, 4 and 6.4.	Measured GAP8 tiled inference uses INT8 operands and INT32 accumulation. FPGA memory-capacity exploration is separate. Layer shape and memory configuration constrain transfer; control precision needs independent validation.
Radar FFT [18]	Sections 3.1, 3.2 and 4.	FPGA timing covers a 256 × 256 pipeline, with different work in full and FFT-only acceleration. Half-precision support up to 4096 points specifies capacity, not that timing configuration.
Robomorphic [19]	Sections 5.1 and 6.1–6.4.	Robot-specific fixed-point spatial products retain morphology and recursive dependencies. FPGA execution, simulated power and 12-nm ASIC estimates are distinct; the ASIC figures are not silicon measurements.
ECHOES [20]	Sections II-A, III and IV.	Measured 65-nm FFT and mel-frequency cepstral coefficient (MFCC) results couple component precision to maximum length. Application error, buffering, arbitration and shared audio/radar deadlines need separate validation.
DCT/IDCT [21]	Sections III-C, IV and V.	The 8 × 8 shift–add transform uses transpose storage and a 33-cycle schedule. FPGA/image tests do not establish normative decoder conformance or arbitrary-length temporal-transform support.
FAST [22]	Sections V-B/C and VI-A/B/E; Appendix C.	DCT, quantization and tokenization define action representation; runtime decoding uses the inverse pipeline. Robot/simulation policy results do not quantify transform hardware, decoding cost, precision or memory.
AV1 FME [23]	Sections IV, V and VI-D.	Approximate encoder interpolation changes sample demand and evaluated BD-BR, not normative decoding. External bandwidth depends on caching. Power is a 40-nm synthesis estimate with assumed switching.
Polar-format SAR [24]	Sections III and IV.	FP32 coordinate/interpolation work precedes FP16 data/FFT processing. FPGA timings distinguish FFT-only and combined acceleration. Image similarity is dataset-specific; sensor-to-image deadlines and airborne power remain unresolved.
LiDAR matching [25]	Sections 3.4, 4.2–4.5 and 5.7.	Pynq-Z2 evaluation covers 2D matching; coordinate and occupancy widths differ. The 2.3–2.4 W is whole-board power. Binary points and score accumulation remain unverified; 3D generalization is unsupported.

Table A5. *Cont.*

Source	Key Sections	Relevance to the Synthesis
ITA [26]	Sections III, IV and V-A/C/E.	Attention/softmax results use 22-nm post-layout estimates and synthetic activity. SRAM-inclusive and standalone efficiencies differ. Activation error does not establish task accuracy; 64 × 64 tiles do not establish small-recursion utilization.
RTC [27]	Sections 3.1–3.3 and 4.2; Appendices A.3 and A.5.	Robot/simulation tests use guidance and asynchronous chunks. GPU-only means differ from CPU/network-inclusive rollout intervals. Workspace, accumulator formats and maximum latency remain unresolved; overlap alone does not prove schedulability.
VVC search [28]	Sections 3.1–3.3 and 4.	Virtex-7 implementation/timing simulation yields calculated 4K30 for the stated CTU/search window. Affine search and motion-vector bit cost are excluded; this is kernel throughput, not complete-encoder measurement.
Adaptive CFAR [29]	Sections 5.1, 5.2 and 5.5.	MATLAB-vector RTL checks and Stratix-II synthesis/timing use $N = 16$; $N = 36$ is illustrative. Sum/square reuse and sorting require scheduled phases. Complete fixed-point, rounding and overflow semantics remain unverified.
Torque control [30]	Sections 2.1–2.3, 3 and 4.	Physical motor/inverter/FPGA tests reconstruct torque from current/angle. The 64-/16-kHz comparison changes control rate. Arithmetic widths and worst-case latency are unresolved; predictor interval needs clarification.
FAST-LIVO2 [31]	Sections IV-A, V-A and X-A.	CPU and unmanned aerial vehicle (UAV) tests couple estimator state, map and patch storage. The 80.4-ms mean includes ten MPC calls per 100-ms window; it is neither accelerator timing nor a deadline bound.
HPIPM [32]	Sections 3.2–3.5, 4.1 and 5.3.	CPU benchmarks and closed-loop simulation cover structured QP. Timing excludes condensing and SQP overhead; condensing choice, conditioning and solve quality constrain reuse of low-precision inference arithmetic.
OSQP [33]	Sections 3.1, 5.2 and 6–8.	Direct-method factors persist while the KKT matrix is unchanged; updates and polishing alter work. This differs from indirect iteration. Residual tolerance specifies neither precision nor a control deadline.
LMS distributed arithmetic [35]	Sections III-A/C/D, IV-A/C and V-A/B/F.	Four-bit Q1.3 simulation/synthesis compares mutable lookup memories and schedules for 4/8/16 taps. Acoustic convergence, sample deadlines and completed-update throughput are unmeasured; intermediate widths/saturation are incomplete.
SPWM [36]	Sections 3.1, 3.2 and 4.1–4.4.	FPGA/oscilloscope results establish waveform generation. The 413× comparison is pulse-width calculation/interrupt initialization versus register configuration, not continuous waveform service or closed-loop speedup.
VVC multiple transforms [40]	Sections II, III and IV.	Shared shift–add/adder trees support 4–32-point DCT-II/DCT-VIII/DST-VII, excluding 64 points; permutation/sign/transpose state constrains reuse. Reference-vector checks and ASIC/FPGA synthesis are not fabricated-silicon or complete-encoder measurements.
Chirp-scaling SAR [41]	Sections II-A, III and IV.	Reordering shares FFT/phase-compensation multipliers. FPGA timing and image quality are dataset-specific. The 128-bit Advanced eXtensible Interface (AXI) carries four 32-bit words, not every operator format; arbitration, flight deadlines and system energy are unresolved.
Narrowband adaptation [42]	Sections 3–6 and 8.	Physical microphonics tests accompany adaptation/identification. Eighteen-bit firmware, 81.25-MHz clock and 2-kHz output describe separate boundaries. The 0.166-W power report is not a board measurement; piezo feedback remains future work.
Early radar detection [44]	Sections III–VI.	AWR2243/Artix-7 block-onset measurements exclude separately stated alignment delay. The early branch must avoid backpressure. Complete sensor-to-brake response and shared-arbitration bounds remain unestablished.
FMCW ASIC [45]	Sections II, III-A/C and IV.	Measured 40-nm chip uses FPGA-delivered synthetic front-end data, scheduled SRAM and truncated spectra. The calculated 12.67-ms frame time assumes specified chirps/targets; online/full-buffer worst-case timing is unresolved.

Table A5. *Cont.*

Source	Key Sections	Relevance to the Synthesis
NLO-CSM [46]	Sections II, III and IV-A/B.	Coarse/fine matching shares transforms, reads and accumulation; host completes matrix processing. FPGA demonstration differs from 65-nm layout. Eight-bit binary-point/overflow details, full-trajectory accuracy and transfer-inclusive deadlines remain unresolved.
FAST-LIO2 [49]	Sections IV-A/B, V-C/E and VII-B1.	Software sensor tests combine 24-state estimation and mutable indexing. The 756-point comparison changes CPU/thread count. ARM occasionally exceeds the scan period; numerical widths and full working set are unresolved.
Square-root marginalization [52]	Sections 4.2, 4.3 and 5.1.	CPU replay compares priors/numerical variants. The reduced camera solve still uses normal equations and LDL^{T}; runtime improvement also changes precision. No onboard actuation or hardware-sharing experiment is reported.
SR-VINS [54]	Sections II-C, III and V.	Dataset replay supports permuted QR/square-root updates. The FP32 1.1-ms case is MSCKF-only: 15 clones, 200 tracks, no feature tracking. Other configurations and complete VINS timing differ.
Square-root VINS [55]	Sections III-B, IV and VI.	LLT handles a conditioned update; other phases retain QR. Laptop means, Jetson tests and initialization differ. This extends SR-VINS, rather than independently replicating hardware gains or transferring laptop timing.
MCUNet [56]	Sections 3.2 and 4.1.	Measured MCU inference uses generated code/activation reuse; search/training is setup. Default INT8 and separate INT4 tests differ. In-place depthwise writes require expired old-channel uses.
MCUNetV2 [57]	Sections 3.1–3.3 and 4.1.	The analytical 1372-to-172-kB patch example repeats halos and increases MACs. Separate device profiles differ. Network redistribution changes architecture, so its benefit cannot be attributed solely to patching.
Tiny Transformers [59]	Sections 2.2.2, 5.1 and 5.2.1.	Hot starts exclude initial level-3 (L3) to level-2 (L2) weight transfer; measured power includes external random-access memory (RAM) and Flash. Enlarged level-1 (L1) GVSoC kernel simulation excludes movement and is distinct from physical-memory inference measurements.
I-ViT [61]	Sections 3.2, 3.5 and 4.1, 4.2.	INT8 products use wider intermediates/requantization; LayerNorm uses ten square-root steps. Fine-tuned, batch-eight TVM compiler-framework timing on RTX 2080Ti establishes GPU execution, not robotic embedded performance.
SOLE [62]	Sections III-B/C, IV and V-A/C/D.	Buffered nonlinear units are evaluated by RTL synthesis/switching-based power estimates. GPU comparisons scale unit count. Complete softmax/normalization and isolated statistics differ; no chip/robot measurements are reported.
hls4ml attention [64]	Sections IV-A/C and VI-B.	Small engine classifier omits LayerNorm and uses partitioned storage, FIFOs and fixed point. Vivado mappings are not measured board timings; timing inconsistencies and model size limit extrapolation.
MINOTAUR [66]	Sections II–V and VII.	Measured posit-8 silicon includes wider accumulation, on-chip memory and LoRA state. Checkpointing trades memory for recomputation. Power excludes I/O; summed layer timing is not a control deadline.
FINN-T [65]	Sections III–VI.	Runtime operands, hybrid softmax and sequence-dependent buffers constrain attention. Evaluated models alter normalization/activation. Untrained resource sweeps, trained accuracy and throughput differ; streaming throughput is not autoregressive query latency.
ACT [69]	Sections IV-A/B; Appendix C.	Four red–green–blue (RGB) views and joint coordinates feed ensembled action chunks; the conditional variational autoencoder (CVAE) training encoder is removed. GPU latency differs from the motor proportional–integral–derivative (PID) period/worst-case response. Feature-channel descriptions are inconsistent.
OpenVLA-OFT [70]	Sections IV-B/C and VI-A/C/D; Appendices A–C and F.	Simulation/ALOHA configurations differ in images, chunk size and action dimensions. Action throughput differs from observation refresh. L_1 is a fine-tuning loss; peak memory, accumulators and maximum transfer-inclusive latency are unknown.

Table A5. *Cont.*

Source	Key Sections	Relevance to the Synthesis
Diffusion Policy [71]	Sections 3.1–3.4; Appendix A.4.	Ten-step/approximately 0.1-s and sixteen-step real-task configurations differ. Conditioning reuse follows the selected schedule; complete sensor-to-command timing is unspecified. The Robotics: Science and Systems (RSS) antecedent is not an independent experiment.
OpenVLA [72]	Sections 3.2 and 5.3; Appendix D.4.	A5000 precision comparisons expose weight/conversion costs. Blocking-control follow-up supports a timing explanation for nonblocking success differences, not generally higher INT4 accuracy.
VLA-Cache [73]	Sections 3.1–3.4 and 5; Appendices D and E.1.	Latency measured with the CUDA GPU platform and control frequency have different boundaries. Static/adaptive caching differs in task success, requiring invalidation/quality checks. Cache bytes, transfer energy, physical sharing and bitwise equivalence remain unestablished.
acados [76]	Sections 3.4, 4.1, 4.2 and 5.3.	Embedded hardware-in-the-loop (HIL) uses a simulated engine and preallocated workspaces. Observed peaks are not worst-case proofs; platform memory is not solver allocation. Integration, sensitivities, condensing and QP phases differ.
TinyMPC [75]	Sections III-B/C and IV-A1/B1/B2.	Cached factors require model/cost/penalty assumptions. Teensy benchmarks differ from physical Crazyflie tests; retain each horizon/control/motion-capture rate. Observed maxima are not universal bounds; extracted numerical width is unspecified.
MPCGPU [77]	Sections IV and V-A/B.	Measured workstation solvers accompany simulated IIWA tracking. Block structure, PCG reductions and synchronization yield nonuniform timing, including slow outliers. Physical manipulator control and arithmetic format are not established.
GRiD [80]	Sections IV-B/C and V-A/C.	Topology-aware CPU/GPU kernels reuse intermediates across three robot configurations. Single-call comparison excludes GPU I/O; batch throughput has different transfer sensitivity and does not establish one-step or closed-loop benefit.
Dadu-RBD [81]	Sections II, III-B, IV-B2, V and VI-A.	Spatial functions retain FIFO/feedback state and convert reciprocals; morphology changes require reconfiguration. FPGA board execution is ambiguous and power estimated. Batch advantage coexists with a slower single derivative than Robomorphic.
Inverter FCS-MPC [82]	Sections 2.2, 3.1, 3.2 and 4.	Physical inverter/RL-load tests match sampling/current distortion. Calculation time excludes acquisition/full-loop response; synthesized DSP use is not an energy measurement. Switch state persists between samples; binary-point extraction is incomplete.
FOC synchronization [84]	Sections 2–4.	Physical PMSM/FPGA tests vary feedback scheduling and sampling. Operand readiness changes service intervals; arithmetic widths, state bytes and power remain unverified.
Unified factor-graph accelerator [87]	Sections 4–8.	ORIANNA extension covers localization/planning/control with block sparsity and operation-specific precision. FPGA power is estimated post-route; inconsistent width descriptions prevent consolidation. Mean timing does not establish deadline distributions.
HiPER [88]	Sections 3–6.2 and 6.5.	Cycle-accurate simulation/16-nm synthesis evaluates shared neural/model processing elements (PEs). Host setup and on-chip networks precede static runtime schedules, excluding discussed dynamic rapidly exploring random tree (RRT)/SLAM cases.
Shared-core MPC [89]	Sections 3–5.3; artifact appendix.	MPC/DroNet coexist on one processor. RTL comparisons, measured power and simulated plant/rotor dynamics differ; the artifact slows physics for hardware timing. Not independently reproduced; no general shared-arithmetic guarantee follows.
Snitch [90]	Sections 4.1–4.3.	RTL timing/post-layout power concern local kernels. FREP does not improve store-limited AXPY. kNN measures distance only, excluding dominant sorting; local throughput does not establish complete off-chip performance.

Table A5. *Cont.*

Source	Key Sections	Relevance to the Synthesis
Ara2 [93]	Sections II–IV.	Width changes require register rearrangement; stores interact with scalar caching and segmented accesses execute elementwise. Simulated timing retains transition/memory costs; instruction support alone does not establish sustained bandwidth.
Spatz [94]	Sections II, III, V and VI-A.	Local vectors/multibanked storage face reuse-dependent port limits. Physical estimates use synthesis/place-and-route and gate-level switching, not silicon measurements. Port/register changes alter throughput/energy configurations.
Quadrilatero [95]	Sections 3 and 4.	FP32 and integer-to-INT32 modes differ; tested shapes change utilization. Physical estimates are 65-nm synthesis, not silicon. Cross-design inconsistencies limit comparison; within-study shape effects remain informative.
Sparse SSR [96]	Sections 3.2, 3.3 and 4.	Values/indices/row pointers and intersection/union require distinct representations/order. Local-memory RTL timing and separate physical estimates establish sparse-stream mechanisms, not arbitrary dynamic maps or complete robotic performance.
FPnew [97]	Sections II-B and IV.	Mixed-format fused operations distinguish source/accumulation/destination semantics. Core integration includes measured 22FDX silicon; operation energy also uses calibrated post-layout analysis. Parallel/merged slices trade implementation cost.
Vicuna [98]	Sections 3–5.	In-order integer/fixed-point coprocessor implements RVV draft 0.10. Measured timing depends on cache/memory/control flow; configured predictability does not extend automatically to other arbitration policies or complete control runtimes.
Gemmini [99]	Sections III–V.	Array, scratchpad, wider accumulators, tiling and translation are coupled. FireSim timing is FPGA-accelerated simulation; ASIC implementation estimates and reported fabricated integrations are separate evidence categories.

Appendix C. Compatibility Decisions and Acceptance Checks

The six records behind Table 12 apply the decision rules in Section 3.4 and the common protocol in Section 6.3; Appendix B supplies source locators. Records were prepared on 19 September and revised on 20 September 2026. None validates concurrent physical sharing.

Appendix C.1. PC01: Audio and Radar FFT Service

Configuration and verdict: ECHOES [20] supports 32/16/8-bit fixed-point components at maximum lengths 512/1024/2048. Radar FFT [18] supports half precision and 64–4096 points; its timed pipeline is 256 × 256. Adaptation is needed for FFT arithmetic, ordering and memory service.

Adaptation and checks: Match length and quality tolerance; add decomposition/storage for 4096 points or reject that mapping. Validate conversion, scaling, rounding and overflow across reference-vector dynamic range. Account for permutation, twiddles, banks and mode-specific transfers, including windowing/magnitude/phase where within scope. Analyze combined arrivals, buffering and service time.

Unresolved: The matched audio length/tolerance, concurrent arrivals/deadlines, conversion cost and interference are unreported.

Appendix C.2. PC02: Quantized Inference and Structured MPC Linear Algebra

Configuration and verdict: DORY [17] uses MobileNet-v1 width 1.0, 128 × 128 input, signed INT8 weights, unsigned INT8 activations and INT32 accumulation. HPIPM [32]

uses double-precision hanging-chain MPC: 24 states, three controls, 40 stages condensed to five. Arithmetic/scratchpad reuse needs adaptation.

Adaptation and checks: Establish solver precision through conditioning, residual, convergence/failure and solution/control-error tests. Extract actual layer/condensed-block shapes; state dimension alone is not an array shape. Account for dependent solves, state/workspace, competing traffic and complete transactions, including condensing, factorization, refinement and the full loop. Test difficult problems and short-kernel utilization.

Unresolved: Selected neural-layer and condensed-matrix layouts, shared schedule, worst-case iterations and reduced-precision control guarantees.

Appendix C.3. PC03: Video Candidate Scoring and Radar CFAR Statistics

Configuration and verdict: VVC motion estimation [28] uses a 128 × 128 CTU, 64 × 64 PE array, 8-bit pixels and 13–22-bit comparison paths. CFAR [29] times $N = 16$; $N = 36$, $p = 24$ is a different illustration. Reductions/comparisons require adaptation.

Adaptation and checks: Separate SAD differences, CFAR squares/products, sums, extremum/index selection and partial sorting. Specify reduction lengths, widths, rounding/ties and partition/window state. Validate complete scoring/detection outputs bit-exactly or within declared quality bounds, with a cycle/state schedule covering traversal, windows and sorting. Do not use $N = 16$ timing for $N = 36$.

Unresolved: CFAR fixed-point semantics, matched frame/arrival rates, queue bounds and multiplexing overhead.

Appendix C.4. PC04: Configured Q1.3 LMS and Quantized Neural Inference

Configuration and verdict: Ahmad et al. [35] implement 4/8/16-tap, four-bit Q1.3 LMS using mutable distributed-arithmetic memories and shift accumulation. The DORY [17] comparator is the PC02 configuration. Product/reduction reuse needs adaptation.

Adaptation and checks: Specify scaling, intermediate rounding and accumulation. Add coefficient update/writeback or coherent lookup updates when preserving distributed arithmetic; separate coefficient/history and activation/weight contexts. Validate convergence, signal quality, conversion, lookup/state updates, transfers and context cost, then full sample deadlines under a concrete concurrent schedule.

Unresolved: Mapped convergence, complete service time and interference remain unmeasured. The under-specified audio example [16] motivates lifecycle analysis only and is excluded from six-case counts.

Appendix C.5. PC05: Attention Tiles and Robot Dynamics Products

Configuration and verdict: ITA [26] uses 64 × 64 tiles, 16 PEs with 64 MACs each, INT8 inputs and 24-bit accumulation. Robomorphic [19] evaluates seven-joint KUKA iiwa-14 gradients with 6 × 6 spatial matrices and 32-bit fixed point/16 fractional bits. Product reuse needs adaptation.

Adaptation and checks: Establish each product/accumulator format; a scalar format does not specify internal widths. Validate quality/convergence under numeric conversions and measure/simulate utilization, padding/batching/lane partitioning, launch, transfer and context costs. Preserve morphology-specific recursion without assuming generic sparse encoding, alongside attention normalization/rescaling and their data movement. Require a dependency-respecting concurrent schedule.

Unresolved: Shared mapping, utilization, dynamic range, control deadlines, attention arrivals and switching cost.

Appendix C.6. PC06: Video and LiDAR Candidate Value/Index Selection

Configuration and verdict: VVC [28] uses the PC03 configuration to select SAD minima/indices. LiDAR CSM [25] uses at most 512 points, a 320 × 320 map, 32-bit fixed-point coordinates and 6-bit occupancy. Conditional compatibility covers only value/index extrema.

Adaptation and checks: Verify score/index signedness, width, overflow, ties and min/max mode; coordinate/occupancy widths do not establish score width. Keep separate producer/address paths and candidate contexts. Test token semantics against the final publication. Serialize requests or prove tagged/arbitrated deadlines, protecting partial winners and bounding backpressure; measure throughput, buffering and pipeline interference.

Unresolved: LiDAR binary-point/score accumulation and final-publication consistency, arrival rates, concurrent deadlines and arbitration cost.

Appendix D. Dated Citation Observations

Table A6 preserves identity-matched citation observations from database result pages/exports on 19 September 2026: 50 of the initial 103 references, excluding four later studies and their corrigendum. WoS Core/Total denotes Core Collection/Total Times Cited; Scopus denotes Cited By; IEEE denotes Article Citation Count. Patent/view/usage counts are excluded. Counts are not summed across databases or repeated queries, ranked as quality/adoption, or used for inclusion.

NA means unavailable, not zero or database absence. Semicolons retain distinct readings: OSQP's Scopus page showed 1125 and its same-day export 1129; a common snapshot cannot be reconstructed.

Table A6. Database-specific citation observations, 19 September 2026. Reference numbers link to the complete bibliography.

Reference	WoS Core	WoS Total	Scopus	IEEE
[14]	19	23	23	NA
[16]	4	4	3	NA
[17]	105	112	127	124
[18]	26	33	44	NA
[19]	38	46	48	NA
[20]	1	1	4	3
[21]	9	12	14	16
[23]	NA	NA	6	NA
[24]	4	4	6	NA
[25]	20	22	29	27
[26]	19	20	27	24
[27]	NA	NA	3	NA
[28]	8	8	13	NA
[29]	16	17	24	NA
[30]	3	3	4	NA
[31]	141	152	166	193
[32]	204	229	221	NA
[33]	991	1164	1125; 1129	NA
[35]	13	14	20	19
[36]	1	1	1	NA
[41]	8	11	12	NA
[42]	0	0	1	NA
[44]	4	6	6	NA
[45]	10	10	14	12
[46]	8	9	11	NA
[47]	1905	2231	2519	2075

Table A6. *Cont.*

Reference	WoS Core	WoS Total	Scopus	IEEE
[48]	933	1069	1180	1117
[49]	1396	1576	1786	1731
[51]	14	15	17	13
[52]	12	12	26	23
[54]	2	2	8	9
[55]	2	2	2	2
[59]	17	17	21	29
[61]	102	106	145	132
[62]	44	44	58	51
[64]	4	4	4	NA
[77]	20	22	28	23
[78]	33	39	38	40
[80]	NA	NA	NA	17
[81]	9	10	14	NA
[82]	14	15	18	NA
[83]	NA	NA	26	NA
[84]	7	7	8	NA
[88]	2	2	2	NA
[89]	0	0	1	3
[90]	70	71	79	79
[93]	28	30	34	33
[94]	10	10	19	18
[95]	0	0	3	NA
[96]	12	12	18	15

No value was available in the retained observations for the other 53 references in that initial bibliography: [1–13,15,22,34,37–39,43,50,53,56–58,60,63,67–76,79,85,86,91,92,97–107].

References

1. Wu, E.-H.; Liu, Y.-Q.; Xu, T.-C.; Ren, L.-X.; Qin, Y.-M.; Wei, M.-Y.; He, X.-W.; Yuan, D.-Y.; Hou, W.-C.; Ma, Z.-W.; et al. Physical AI: Evolution, Progress, Challenges, and Prospects. *J. Comput. Sci. Technol.* **2026**, *41*, 271–288. [CrossRef]
2. Asanović, K.; Bodik, R.; Catanzaro, B.C.; Gebis, J.J.; Husbands, P.; Keutzer, K.; Patterson, D.A.; Plishker, W.L.; Shalf, J.; Williams, S.W.; et al. The Landscape of Parallel Computing Research: A View from Berkeley. UCB/EECS-2006-183. 2006. Available online: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.pdf (accessed on 19 September 2026).
3. Krishnakumar, A.; Ogras, U.; Marculescu, R.; Kishinevsky, M.; Mudge, T. Domain-Specific Architectures: Research Problems and Promising Approaches. *ACM Trans. Embed. Comput. Syst.* **2023**, *22*, 28. [CrossRef]
4. Melchert, J.; Feng, K.; Donovick, C.; Daly, R.; Sharma, R.; Barrett, C.; Horowitz, M.A.; Hanrahan, P.; Raina, P. APEX: A Framework for Automated Processing Element Design Space Exploration using Frequent Subgraph Analysis. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3 (ASPLOS)*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 33–45. [CrossRef]
5. Vasile, C.-E.; Ulmămei, A.-A.; Bîră, C. Image Processing Hardware Acceleration - A Review of Operations Involved and Current Hardware Approaches. *J. Imaging* **2024**, *10*, 298. [CrossRef] [PubMed]
6. Peccerillo, B.; Mannino, M.; Mondelli, A.; Bartolini, S. A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives. *J. Syst. Archit.* **2022**, *129*, 102561. [CrossRef]
7. Giuffrida, L.; Masera, G.; Martina, M. A Survey of Automotive Radar and Lidar Signal Processing and Architectures. *Chips* **2023**, *2*, 243–261. [CrossRef]
8. Wan, Z.; Yu, B.; Li, T.Y.; Tang, J.; Zhu, Y.; Wang, Y.; Raychowdhury, A.; Liu, S. A Survey of FPGA-Based Robotic Computing. *IEEE Circuits Syst. Mag.* **2021**, *21*, 48–74. [CrossRef]
9. Vishwakarma, S.; Khan, Z.; Lehtomäki, J.J. Impact of MAC Unit Design Architectures and Their Applications in Modern Computing: An In-Depth Review. *IEEE Access* **2026**, *14*, 24339–24360. [CrossRef]
10. Yusuf, A.; Adegbija, T.; Gajaria, D. Domain-Specific STT-MRAM-Based In-Memory Computing: A Survey. *IEEE Access* **2024**, *12*, 28036–28056. [CrossRef]

11. Agosta, G.; Galimberti, A.; Zoni, D. Deep Learning on RISC-V Platforms at the Edge: A Perspective on the Hardware and Software Support. *ACM Comput. Surv.* **2026**, *58*, 121. [CrossRef]
12. Feng, T.; Wang, X.; Jiang, Y.-G.; Zhu, W. Embodied AI: From LLMs to World Models. In *IEEE Circuits and Systems Magazine*; IEEE: Piscataway, NJ, USA, 2025; pp. 14–37. [CrossRef]
13. Parashar, A.; Raina, P.; Shao, Y.S.; Chen, Y.-H.; Ying, V.A.; Mukkara, A.; Venkatesan, R.; Khailany, B.; Keckler, S.W.; Emer, J. Timeloop: A Systematic Approach to DNN Accelerator Evaluation. In Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, Madison, WI, USA, 24–26 March 2019; pp. 304–315. [CrossRef]
14. Hao, Y.; Gan, Y.; Yu, B.; Liu, Q.; Han, Y.; Wan, Z.; Liu, S. Orianna: An Accelerator Generation Framework for Optimization-based Robotic Applications. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS)*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 813–829. [CrossRef]
15. Huang, Y.; Hao, Y.; Yu, B.; Yan, F.; Yang, Y.; Min, F.; Han, Y.; Ma, L.; Liu, S.; Liu, Q.; et al. Dadu-Corki: Algorithm-Architecture Co-Design for Embodied AI-powered Robotic Manipulation. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture (ISCA)*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 327–343. [CrossRef]
16. Yuan, J.; Zhao, Q.; Wang, W.; Meng, X.; Li, J.; Li, Q. Audio Denoising Coprocessor Based on RISC-V Custom Instruction Set Extension. *Acoustics* **2022**, *4*, 538–553. [CrossRef]
17. Burrello, A.; Garofalo, A.; Bruschi, N.; Tagliavini, G.; Rossi, D.; Conti, F. DORY: Automatic End-to-End Deployment of Real-World DNNs on Low-Cost IoT MCUs. *IEEE Trans. Comput.* **2021**, *70*, 1253–1268. [CrossRef]
18. Heo, J.; Jung, Y.; Lee, S.; Jung, Y. FPGA Implementation of an Efficient FFT Processor for FMCW Radar Signal Processing. *Sensors* **2021**, *21*, 6443. [CrossRef] [PubMed]
19. Neuman, S.M.; Plancher, B.; Bourgeat, T.; Tambe, T.; Devadas, S.; Reddi, V.J. Robomorphic Computing: A Design Methodology for Domain-Specific Accelerators Parameterized by Robot Morphology. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 674–686. [CrossRef]
20. Sinigaglia, M.; Bertaccini, L.; Valente, L.; Garofalo, A.; Benatti, S.; Benini, L.; Conti, F.; Rossi, D. ECHOES: A 200 GOPS/W Frequency Domain SoC with FFT Processor and I2S DSP for Flexible Data Acquisition from Microphone Arrays. In *Proceedings of the 2023 IEEE International Symposium on Circuits and Systems (ISCAS)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5. [CrossRef]
21. Zhou, Z.; Pan, Z. Effective Hardware Accelerator for 2D DCT/IDCT Using Improved Loeffler Architecture. *IEEE Access* **2022**, *10*, 11011–11020. [CrossRef]
22. Pertsch, K.; Stachowicz, K.; Ichter, B.; Driess, D.; Nair, S.; Vuong, Q.; Mees, O.; Finn, C.; Levine, S. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In Proceedings of the Robotics: Science and Systems, Los Angeles, CA, USA, 21–25 June 2025. [CrossRef]
23. Kolodziejski, W.; Domanski, R.; Porto, M.; Zatt, B.; Agostini, L. Ultra-High Definition AV1 FME Interpolation Architectures Exploring Approximate Computing. *J. Integr. Circuits Syst.* **2022**, *17*, 1–12. [CrossRef]
24. Jeong, D.; Lee, M.; Lee, W.; Jung, Y. FPGA-Based Acceleration of Polar-Format Algorithm for Video Synthetic-Aperture Radar Imaging. *Electronics* **2024**, *13*, 2401. [CrossRef]
25. Sugiura, K.; Matsutani, H. A Universal LiDAR SLAM Accelerator System on Low-Cost FPGA. *IEEE Access* **2022**, *10*, 26931–26947. [CrossRef]
26. İslamoğlu, G.; Scherer, M.; Paulin, G.; Fischer, T.; Jung, V.J.B.; Garofalo, A.; Benini, L. ITA: An Energy-Efficient Attention and Softmax Accelerator for Quantized Transformers. In Proceedings of the 2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Vienna, Austria, 7–8 August 2023; pp. 1–6. [CrossRef]
27. Black, K.; Galliker, M.Y.; Levine, S. Real-Time Execution of Action Chunking Flow Policies. *Adv. Neural Inf. Process. Syst.* **2025**, *38*, 33383–33407. [CrossRef]
28. Ahmad, W.; Mahdavi, H.; Hamzaoglu, I. An efficient versatile video coding motion estimation hardware. *J. Real-Time Image Process.* **2024**, *21*, 25. [CrossRef]
29. Sim, Y.; Heo, J.; Jung, Y.; Lee, S.; Jung, Y. FPGA Implementation of Efficient CFAR Algorithm for Radar Systems. *Sensors* **2023**, *23*, 954. [CrossRef] [PubMed]
30. Sun, Z.; Xu, Y.; Ma, Z.; Xu, J.; Zhang, T.; Xu, M.; Mei, X. Field Programmable Gate Array Based Torque Predictive Control for Permanent Magnet Servo Motors. *Micromachines* **2022**, *13*, 1055. [CrossRef] [PubMed]
31. Zheng, C.; Xu, W.; Zou, Z.; Hua, T.; Yuan, C.; He, D.; Zhou, B.; Liu, Z.; Lin, J.; Zhu, F.; et al. FAST-LIVO2: Fast, Direct LiDAR-Inertial-Visual Odometry. *IEEE Trans. Robot.* **2025**, *41*, 326–346. [CrossRef]
32. Frison, G.; Diehl, M. HPIPM: A high-performance quadratic programming framework for model predictive control. *IFAC-PapersOnLine* **2020**, *53*, 6563–6569. [CrossRef]
33. Stellato, B.; Banjac, G.; Goulart, P.; Bemporad, A.; Boyd, S. OSQP: An operator splitting solver for quadratic programs. *Math. Program. Comput.* **2020**, *12*, 637–672. [CrossRef]

34. de Rivaz, P.; Haughton, J. AV1 Bitstream and Decoding Process Specification. Alliance for Open Media, 2019, 1.0.0 with Errata 1. Available online: https://aomediacodec.github.io/av1-spec/av1-spec.pdf (accessed on 19 September 2026).
35. Ahmad, S.; Khawaja, S.G.; Amjad, N.; Usman, M. A Novel Multiplier-Less LMS Adaptive Filter Design Based on Offset Binary Coded Distributed Arithmetic. *IEEE Access* **2021**, *9*, 78138–78152. [CrossRef]
36. Yin, Q.; Li, D.; Wu, Z.; Pun, S.H.; Liu, Y. Design and implementation of RISC-V system-on-chip for SPWM generation based on FPGA. *IEICE Electron. Express* **2024**, *21*, 20240603. [CrossRef]
37. Cooley, J.W.; Tukey, J.W. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comput.* **1965**, *19*, 297–301. [CrossRef]
38. ETSI. Digital Enhanced Cordless Telecommunications (DECT); Low Complexity Communication Codec plus (LC3plus). European Telecommunications Standards Institute, 2023, TS 103 634, V1.4.1. Available online: https://www.etsi.org/deliver/etsi_ts/103600_103699/103634/01.04.01_60/ts_103634v010401p.pdf (accessed on 19 September 2026).
39. ITU-T. Versatile Video Coding. International Telecommunication Union, 2020, Recommendation H.266, August 2020. Available online: https://www.itu.int/ITU-T/recommendations/rec.aspx?id=14336 (accessed on 19 September 2026).
40. Hao, Z.; Sun, H.; Xiang, G.; Zhang, P.; Zeng, X.; Fan, Y. A Reconfigurable Multiple Transform Selection Architecture for VVC. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2023**, *31*, 658–669. [CrossRef]
41. Lee, J.; Jeong, D.; Lee, S.; Lee, M.; Lee, W.; Jung, Y. FPGA Implementation of the Chirp-Scaling Algorithm for Real-Time Synthetic Aperture Radar Imaging. *Sensors* **2023**, *23*, 959. [CrossRef] [PubMed]
42. Abdi, F.; Cichalewski, W.; Jałmużna, W.; Butkowski, Ł.; Branlard, J.; Bellandi, A.; Jabłoński, G. FPGA-Based Implementation of an Adaptive Noise Controller for Continuous Wave Superconducting Cavity. *Electronics* **2024**, *13*, 155. [CrossRef]
43. Rohling, H. Radar CFAR Thresholding in Clutter and Multiple Target Situations. *IEEE Trans. Aerosp. Electron. Syst.* **1983**, *AES-19*, 608–621. [CrossRef]
44. Petrović, N.; Petrović, M.; Milovanović, V. Radar Signal Processing Architecture for Early Detection of Automotive Obstacles. *Electronics* **2023**, *12*, 1826. [CrossRef]
45. Guo, M.; Zhao, D.; Wu, Q.; Wu, J.; Li, D.; Zhang, P. An Integrated Real-Time FMCW Radar Baseband Processor in 40-nm CMOS. *IEEE Access* **2023**, *11*, 36041–36051. [CrossRef]
46. Hu, A.; Yu, G.; Wang, Q.; Han, D.; Zhao, S.; Liu, B.; Yu, Y.; Li, Y.; Wang, C.; Zou, X. Efficient Hardware Accelerator Design of Non-Linear Optimization Correlative Scan Matching Algorithm in 2D LiDAR SLAM for Mobile Robots. *Sensors* **2022**, *22*, 8947. [CrossRef] [PubMed]
47. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-Coupled Lidar Inertial Odometry via Smoothing and Mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 5135–5142. [CrossRef]
48. Xu, W.; Zhang, F. FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3317–3324. [CrossRef]
49. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. Robot.* **2022**, *38*, 2053–2073. [CrossRef]
50. Cai, Y.; Xu, W.; Zhang, F. ikd-Tree: An Incremental K-D Tree for Robotic Applications. *arXiv* **2021**. [CrossRef]
51. Hao, Y.; Gan, Y.; Yu, B.; Liu, Q.; Liu, S.-S.; Zhu, Y. BLITZCRANK: Factor Graph Accelerator for Motion Planning. In Proceedings of the ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 9–13 July 2023; pp. 1–6. [CrossRef]
52. Demmel, N.; Schubert, D.; Sommer, C.; Cremers, D.; Usenko, V. Square Root Marginalization for Sliding-Window Bundle Adjustment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 13240–13248. [CrossRef]
53. Geneva, P.; Eckenhoff, K.; Lee, W.; Yang, Y.; Huang, G. OpenVINS: A Research Platform for Visual-Inertial Estimation. In Proceedings of the IEEE International Conference on Robotics and Automation, Paris, France, 31 May–31 August 2020; pp. 4666–4672. [CrossRef]
54. Peng, Y.; Chen, C.; Huang, G. Ultrafast Square-Root Filter-based VINS. In Proceedings of the IEEE International Conference on Robotics and Automation, Yokohama, Japan, 13–17 May 2024; pp. 6966–6972. [CrossRef]
55. Peng, Y.; Chen, C.; Wu, K.; Huang, G. sqrtVINS: Robust and Ultrafast Square-Root Filter-Based 3D Motion Tracking. *IEEE Trans. Robot.* **2025**, *41*, 6570–6589. [CrossRef]
56. Lin, J.; Chen, W.-M.; Lin, Y.; Cohn, J.; Gan, C.; Han, S. MCUNet: Tiny Deep Learning on IoT Devices. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 11711–11722. [CrossRef]
57. Lin, J.; Chen, W.-M.; Cai, H.; Gan, C.; Han, S. Memory-efficient Patch-based Inference for Tiny Deep Learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 2346–2358. Available online: https://proceedings.neurips.cc/paper/2021/hash/1371bccec2447b5aa6d96d2a540fb401-Abstract.html (accessed on 19 September 2026).
58. Lin, J.; Zhu, L.; Chen, W.-M.; Wang, W.-C.; Gan, C.; Han, S. On-Device Training Under 256KB Memory. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 22941–22954. [CrossRef]

59. Jung, V.J.B.; Burrello, A.; Scherer, M.; Conti, F.; Benini, L. Optimizing the Deployment of Tiny Transformers on Low-Power MCUs. *IEEE Trans. Comput.* **2025**, *74*, 526–541. [CrossRef]
60. Kim, S.; Gholami, A.; Yao, Z.; Mahoney, M.W.; Keutzer, K. I-BERT: Integer-only BERT Quantization. In Proceedings of the 38th International Conference on Machine Learning, Online, 18–24 July 2021; Volume 139, pp. 5506–5518. Available online: https://proceedings.mlr.press/v139/kim21d.html (accessed on 19 September 2026).
61. Li, Z.; Gu, Q. I-ViT: Integer-only Quantization for Efficient Vision Transformer Inference. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 17065–17075. [CrossRef]
62. Wang, W.; Zhou, S.; Sun, W.; Sun, P.; Liu, Y. SOLE: Hardware-Software Co-design of Softmax and LayerNorm for Efficient Transformer Inference. In Proceedings of the 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD), San Francisco, CA, USA, 28 October–2 November 2023; pp. 1–9. [CrossRef]
63. Dao, T.; Fu, D.Y.; Ermon, S.; Rudra, A.; Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16344–16359. [CrossRef]
64. Jiang, Z.; Yin, D.; Chen, Y.; Khoda, E.E.; Hauck, S.; Hsu, S.-C.; Govorkova, E.; Harris, P.; Loncar, V.; Moreno, E.A. Low latency transformer inference on FPGAs for physics applications with hls4ml. *J. Instrum.* **2025**, *20*, P04014. [CrossRef]
65. Berganski, C.; Jentzsch, F.; Platzner, M.; Kuhmichel, M.; Giefers, H. FINN-T: Compiling Custom Dataflow Accelerators for Quantized Transformers. In Proceedings of the 2024 International Conference on Field Programmable Technology (ICFPT), Sydney, Australia, 10–12 December 2024; pp. 1–10. [CrossRef]
66. Prabhu, K.; Radway, R.M.; Yu, J.; Bartolone, K.; Giordano, M.; Peddinghaus, F.; Urman, Y.; Khwa, W.-S.; Chih, Y.-D.; Chang, M.-F.; et al. MINOTAUR: A Posit-Based 0.42–0.50-TOPS/W Edge Transformer Inference and Training Accelerator. *IEEE J. Solid-State Circuits* **2025**, *60*, 1311–1323. [CrossRef]
67. Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In Proceedings of the 7th Conference on Robot Learning, Atlanta, GA, USA, 6–9 November 2023; Volume 229, pp. 2165–2183. Available online: https://proceedings.mlr.press/v229/zitkovich23a.html (accessed on 19 September 2026).
68. Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.R.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. In Proceedings of the Robotics: Science and Systems, Los Angeles, CA, USA, 21–25 June 2025. [CrossRef]
69. Zhao, T.Z.; Kumar, V.; Levine, S.; Finn, C. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In Proceedings of the Robotics: Science and Systems, Daegu, Republic of Korea, 10–14 July 2023. [CrossRef]
70. Kim, M.J.; Finn, C.; Liang, P. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In Proceedings of the Robotics: Science and Systems, Los Angeles, CA, USA, 21–25 June 2025; XXI. [CrossRef]
71. Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *Int. J. Robot. Res.* **2025**, *44*, 1684–1704. [CrossRef]
72. Kim, M.J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.P.; Sanketi, P.R.; Vuong, Q.; et al. OpenVLA: An Open-Source Vision-Language-Action Model. In Proceedings of the 8th Conference on Robot Learning, Munich, Germany, 6–9 November 2024; Volume 270, pp. 2679–2713. Available online: https://proceedings.mlr.press/v270/kim25c.html (accessed on 19 September 2026).
73. Xu, S.; Wang, Y.; Xia, C.; Zhu, D.; Huang, T.; Xu, C. VLA-Cache: Efficient Vision-Language-Action Manipulation via Adaptive Token Caching. *Adv. Neural Inf. Process. Syst.* **2025**, *38*, 164448–164473. [CrossRef]
74. OSQP Developers. Code Generation Documentation. Available online: https://osqp.org/docs/codegen/index.html (accessed on 19 September 2026).
75. Nguyen, K.; Schoedel, S.; Alavilli, A.; Plancher, B.; Manchester, Z. TinyMPC: Model-Predictive Control on Resource-Constrained Microcontrollers. In Proceedings of the IEEE International Conference on Robotics and Automation, Yokohama, Japan, 13–17 May 2024; pp. 1–7. [CrossRef]
76. Verschueren, R.; Frison, G.; Kouzoupis, D.; Frey, J.; van Duijkeren, N.; Zanelli, A.; Novoselnik, B.; Albin, T.; Quirynen, R.; Diehl, M. acados: A Modular Open-Source Framework for Fast Embedded Optimal Control. *Math. Program. Comput.* **2022**, *14*, 147–183. [CrossRef]
77. Adabag, E.; Atal, M.D.; Gerard, W.; Plancher, B. MPCGPU: Real-Time Nonlinear Model Predictive Control through Preconditioned Conjugate Gradient on the GPU. In Proceedings of the IEEE International Conference on Robotics and Automation, Yokohama, Japan, 13–17 May 2024; pp. 9787–9794. [CrossRef]
78. Plancher, B.; Neuman, S.M.; Bourgeat, T.; Kuindersma, S.; Devadas, S.; Reddi, V.J. Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2335–2342. [CrossRef]
79. Neuman, S.M.; Ghosal, R.; Bourgeat, T.; Plancher, B.; Reddi, V.J. RoboShape: Using Topology Patterns to Scalably and Flexibly Deploy Accelerators Across Robots. In *Proceedings of the International Symposium on Computer Architecture*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1–13. [CrossRef]

80. Plancher, B.; Neuman, S.M.; Ghosal, R.; Kuindersma, S.; Reddi, V.J. GRiD: GPU-Accelerated Rigid Body Dynamics with Analytical Gradients. In Proceedings of the IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, 23–27 May 2022; pp. 6253–6260. [CrossRef]
81. Yang, Y.; Chen, X.; Han, Y. Dadu-RBD: Robot Rigid Body Dynamics Accelerator with Multifunctional Pipelines. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 297–309. [CrossRef]
82. Zafra, E.; Vazquez, S.; Guzman Miranda, H.; Sanchez, J.A.; Marquez, A.; Leon, J.I.; Franquelo, L.G. Efficient FPSoC Prototyping of FCS-MPC for Three-Phase Voltage Source Inverters. *Energies* **2020**, *13*, 1074. [CrossRef]
83. Zafra, E.; Vazquez, S.; Regalo, C.; Baena Lecuyer, V.; Marquez Alcaide, A.; Leon, J.I.; Franquelo, L.G. Parallel Sphere Decoding Algorithm for Long-Prediction-Horizon FCS-MPC. *IEEE Trans. Power Electron.* **2022**, *37*, 7896–7906. [CrossRef]
84. Mishra, I.; Tripathi, R.N.; Hanamoto, T. Synchronization and Sampling Time Analysis of Feedback Loop for FPGA-Based PMSM Drive System. *Electronics* **2020**, *9*, 1906. [CrossRef]
85. Milik, A.; Rudnicki, T. A Mixed Hardware-Software Implementation of a High-Performance PMSM Controller. *Electronics* **2023**, *12*, 440. [CrossRef]
86. Intel. AN 1000: Drive-on-Chip Design Example: Agilex 5 Devices. Version 24.1, Document 826207, 8 July 2024. Sections 8.3.5 and 8.3.6. Available online: https://www.intel.com/content/www/us/en/docs/programmable/826207/24-1/foc-subsystem.html (accessed on 19 September 2026).
87. Liu, Q.; Hua, Y.; Hao, Y.; Yu, B.; Liu, S.; Gan, Y. Unified and Efficient Factor Graph Accelerator Design for Robotic Optimization. *ACM Trans. Archit. Code Optim.* **2025**, *22*, 153. [CrossRef]
88. Ting, J.; Kim, M.; Zhu, J.; Sheng, H.; Zhang, Z. HiPER: Hierarchically-Composed Processing for Efficient Robot Learning-Based Control. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 313–326. [CrossRef]
89. Dong, S.K.; Nikiforov, D.; Soedarmadji, W.; Nguyen, M.; Jain, V.; Fletcher, C.W.; Shao, Y.S. Characterizing and Optimizing Real-Time Optimal Control for Embedded SoCs. In Proceedings of the 2025 IEEE International Symposium on Workload Characterization (IISWC), Irvine, CA, USA, 12–14 October 2025; pp. 489–503. [CrossRef]
90. Zaruba, F.; Schuiki, F.; Hoefler, T.; Benini, L. Snitch: A Tiny Pseudo Dual-Issue Processor for Area and Energy Efficient Execution of Floating-Point Intensive Workloads. *IEEE Trans. Comput.* **2021**, *70*, 1845–1860. [CrossRef]
91. Banbury, C.; Reddi, V.J.; Torelli, P.; Jeffries, N.; Kiraly, C.; Holleman, J.; Montino, P.; Kanter, D.; Warden, P.; Pau, D.; et al. MLPerf Tiny Benchmark. Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 2021, 1. Available online: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/da4fb5c6e93e74d3df8527599fa62642-Abstract-round1.html (accessed on 19 September 2026).
92. RISC-V International. The RISC-V Instruction Set Manual, Volume I: Unprivileged Architecture. Release 20.01.2026, Vector Chapter, Naming Conventions and Instruction Listings. Available online: https://docs.riscv.org/reference/isa/v20260120; (accessed on 19 September 2026).
93. Perotti, M.; Cavalcante, M.; Andri, R.; Cavigelli, L.; Benini, L. Ara2: Exploring Single- and Multi-Core Vector Processing with an Efficient RVV 1.0 Compliant Open-Source Processor. *IEEE Trans. Comput.* **2024**, *73*, 1822–1836. [CrossRef]
94. Perotti, M.; Riedel, S.; Cavalcante, M.; Benini, L. Spatz: Clustering Compact RISC-V-Based Vector Units to Maximize Computing Efficiency. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2025**, *44*, 2488–2502. [CrossRef]
95. Cammarata, D.; Perotti, M.; Bertuletti, M.; Garofalo, A.; Schiavone, P.D.; Atienza, D.; Benini, L. Quadrilatero: A RISC-V Programmable Matrix Coprocessor for Low-Power Edge Applications. In *Proceedings of the Companion of the 22nd ACM International Conference on Computing Frontiers*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 66–69. [CrossRef]
96. Scheffler, P.; Zaruba, F.; Schuiki, F.; Hoefler, T.; Benini, L. Sparse Stream Semantic Registers: A Lightweight ISA Extension Accelerating General Sparse Linear Algebra. *IEEE Trans. Parallel Distrib. Syst.* **2023**, *34*, 3147–3161. [CrossRef]
97. Mach, S.; Schuiki, F.; Zaruba, F.; Benini, L. FPnew: An Open-Source Multiformat Floating-Point Unit Architecture for Energy-Proportional Transprecision Computing. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2021**, *29*, 774–787. [CrossRef]
98. Platzer, M.; Puschner, P. Vicuna: A Timing-Predictable RISC-V Vector Coprocessor for Scalable Parallel Computation. *Leibniz Int. Proc. Inform.* **2021**, *196*, 1:1–1:18. [CrossRef]
99. Genc, H.; Kim, S.; Amid, A.; Haj-Ali, A.; Iyer, V.; Prakash, P.; Zhao, J.; Grubb, D.; Liew, H.; Mao, H.; et al. Gemmini: Enabling Systematic Deep-Learning Architecture Evaluation via Full-Stack Integration. In Proceedings of the 58th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 5–9 December 2021; pp. 769–774. [CrossRef]
100. *IEEE Std 754-2019*; IEEE Standard for Floating-Point Arithmetic. IEEE: Piscataway, NJ, USA, 2019. Available online: https://standards.ieee.org/ieee/754/6210/ (accessed on 19 September 2026).
101. Arm. CMSIS-DSP Software Library: Overview. Online Documentation. Available online: https://arm-software.github.io/CMSIS-DSP/latest/ (accessed on 20 September 2026).

102. Mei, L.; Houshmand, P.; Jain, V.; Giraldo, S.; Verhelst, M. ZigZag: Enlarging Joint Architecture-Mapping Design Space Exploration for DNN Accelerators. *IEEE Trans. Comput.* **2021**, *70*, 1160–1174. [CrossRef]
103. Wu, Y.N.; Emer, J.S.; Sze, V. Accelergy: An Architecture-Level Energy Estimation Methodology for Accelerator Designs. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, Westminster, CO, USA, 4–7 November 2019; pp. 1–8. [CrossRef]
104. Kao, S.-C.; Kwon, H.; Pellauer, M.; Parashar, A.; Krishna, T. A Formalism of DNN Accelerator Flexibility. *Proc. ACM Meas. Anal. Comput. Syst.* **2022**, *6*, 41. [CrossRef]
105. Williams, S.; Waterman, A.; Patterson, D. Roofline: An Insightful Visual Performance Model for Multicore Architectures. *Commun. ACM* **2009**, *52*, 65–76. [CrossRef]
106. Wang, F.; Wu, Y.N.; Woicik, M.; Emer, J.S.; Sze, V. Architecture-Level Energy Estimation for Heterogeneous Computing Systems. In Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, Stony Brook, NY, USA, 28–30 March 2021; pp. 229–231. [CrossRef]
107. Yates, R.D.; Sun, Y.; Brown, D.R.; Kaul, S.K.; Modiano, E.; Ulukus, S. Age of Information: An Introduction and Survey. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 1183–1210. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.