A Deterministic SIMD Accelerator for On-Chip Near-Field RIS Phase Computation

Sohan Bag

***Abstract*—Near-field reconfigurable intelligent surface (RIS) beam steering requires a range-dependent phase for every element, refreshed within the channel coherence time. Embedded RIS controllers largely bypass this computation through stored codebooks or learned models. This letter presents a deterministic 4-lane SIMD accelerator that computes the near-field range and 2-bit phase of each element directly from target coordinates, with no stored phase states or trained models. Each lane combines fixed-point distance computation, a 16-stage non-restoring square-root unit, and Binary Angular Measurement (BAM) quantization in a 22-cycle pipeline. A 256-element update takes 87 cycles (580 ns at 150 MHz) independent of target position, a $3.21\times$ reduction over one lane. Post-route on a Zynq-7020, the design uses 2642 LUTs, 3086 FFs and 20 DSP48s, with an estimated 126 mW dynamic power and a $+0.075$-ns timing margin. Bit-exact verification over 212,992 phase states, semantic array-factor checks and closed-loop RTL co-simulation confirm correct operation. The update is 0.35% of a 167-μs coherence bound at 30 GHz.**

***Index Terms*—Reconfigurable intelligent surface, hardware accelerator, FPGA, near-field beamforming, RTL co-simulation.**

I. Introduction

NEAR-FIELD reconfigurable intelligent surfaces (RIS) require an element-dependent phase profile because the propagation distance from a steering target differs across the aperture. Unlike far-field steering, where a common angular gradient sets the profile, near-field focusing requires a distance computation per element [1]–[3]. This workload grows with aperture and must finish within the channel coherence time (Sec. V-E); 2-bit states incur only a bounded quantization loss [4], [5]. Existing embedded RIS controllers largely avoid the computation, applying stored phase states or codebooks [6]–[8] or learned configuration models [9] (Sec. II). Both tie the update to stored configurations or model inference rather than direct evaluation of the target geometry, motivating a datapath that computes the profile itself with predictable latency.

This letter presents a 4-lane SIMD accelerator for direct near-field RIS phase computation. It exploits the independence of element-wise range calculations and combines fixed-point distance accumulation, a data-independent non-restoring square-root pipeline, and BAM quantization. Element geometry is held in ROM; phase states are generated at run time. Four lanes process the 256 elements of a 16×16 RIS in 64 issue cycles plus pipeline drain. *Update latency* denotes the datapath time from dispatch through `done`, excluding host register writes.

The design is evaluated post-route on a Xilinx Zynq-7020 at 150 MHz, where four lanes complete a frame in 87 cycles versus 279 for one lane. Fixed-point accuracy is measured against an ideal floating-point reference, the RTL is verified bit-exactly and semantically, and RTL co-simulation places the accelerator in a closed-loop tracking and blockage-recovery system as a representative embedded control workload; all results derive from post-route implementation and RTL simulation. The contributions are: (i) a 4-lane SIMD datapath for codebook-free near-field phase computation, with an update latency set by array size and lane count rather than target position; (ii) a Q4.12 fixed-point datapath with a non-restoring square-root pipeline that meets the phase-accuracy requirement, chosen against a coarser alternative; and (iii) post-route, bit-exact, semantic and closed-loop evidence of the accelerator's cost, timing, correctness and system-level utility.

S. Bag is with the Department of Electronics and Communication Engineering, MCKV Institute of Engineering, Howrah 711204, India (e-mail: bag.sohan2006@gmail.com).

II. Related Work

Embedded RIS controllers differ in how they obtain the element phase configuration. Prototypes use stored or pre-configured phase states, including FPGA-based beam tracking at 200 MHz [6] and an MCU hierarchy whose near-field spot-beam scheme uses a phase-compensation codebook [7]; near-field implementations use precomputed phase contours [8]. These trade computation for storage. Learned approaches select configurations on edge processors with latency–rate trade-offs [9], so execution time depends on the model and inference path. This work evaluates the near-field phase directly in a deterministic, element-parallel datapath whose latency is set by array size and lane count. Unlike far-field steering, which can use an angular phase gradient [10], it evaluates the spherical-wavefront range required for near-field focusing. RIS tracking [11], [12] is typically evaluated algorithmically; here it serves as a validation workload.

III. Architecture

A. Near-Field Phase Formulation

The RIS is a planar 16×16 grid with $\lambda/2$ (5 mm) spacing at 30 GHz ($\lambda = 10$ mm). For a target at $\mathbf{p}_t$, element i at $\mathbf{p}_i$ requires

$$\theta_i = \frac{2\pi}{\lambda} R_i, \qquad R_i = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2} = \|\mathbf{p}_t - \mathbf{p}_i\|_2, \tag{1}$$

which is computed independently for each element, so the $N = 256$ evaluations map naturally onto P parallel lanes.

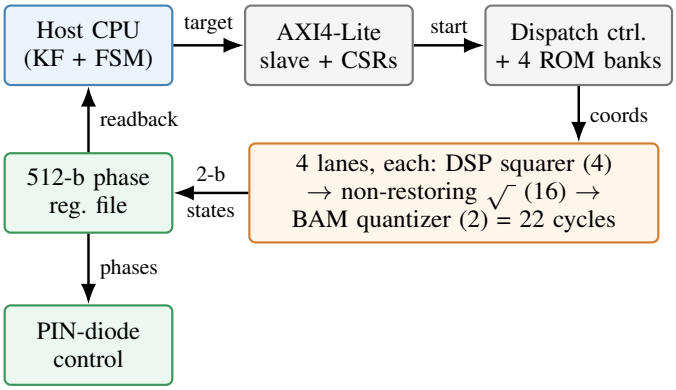


Fig. 1. Accelerator architecture: host interface, dispatch, four 22-cycle lanes, and 512-bit phase register file.

B. Fixed-Point Datapath

Phase is held in a 16-bit BAM register (wrap-around by unsigned overflow) and quantized to $B = 2$ bits (four states). Coordinates are Q4.12 signed metres (LSB 0.24 mm $= \lambda/41$, range ± 8 m), the squared range is Q8.24 unsigned (32-bit), and the extracted range is Q4.12 unsigned (16-bit). Range truncation gives 8.8° of phase per range LSB (error quantified in Sec. IV-B).

Each lane is a fixed three-stage pipeline (Fig. 1): (i) a DSP48-cascade squarer computes $D = \Delta x^2 + \Delta y^2 + \Delta z^2$ (4 cycles); (ii) a 16-stage non-restoring square-root recurrence, consuming two radicand bits and producing one root bit per stage, gives $R = \sqrt{D}$ (16 cycles, data-independent, initiation interval 1, zero DSPs); (iii) BAM scaling with a half-LSB rounding bias and extraction of the top B bits (2 cycles), so $L_{\text{pipe}} = 4 + 16 + 2 = 22$ cycles per element. Non-restoring extraction avoids the DSP48s consumed by Newton–Raphson and LUT/Taylor schemes and the scale-factor correction of CORDIC.

C. SIMD Organization and Latency Model

Four lanes process the 256 elements as four streams of 64, fed from one ROM bank per lane (conflict-free single-cycle fetch) by a dispatch controller issuing one tuple per lane per cycle; the banks hold element geometry, not phase states. Because the recurrence is data-independent, update latency for N elements and P lanes is a deterministic function of array size and lane count:

$$L(N, P) = \left\lceil \frac{N}{P} \right\rceil + L_{\text{pipe}} + 1, \tag{2}$$

where the final cycle asserts `done`. For $N = 256$, $P = 4$ this gives $64 + 22 + 1 = 87$ cycles (580 ns at 150 MHz), and for $P = 1$ it gives $256 + 22 + 1 = 279$ cycles.

D. Interface and Control

An output packer writes each 2-bit state to a 512-bit register file exposed as sixteen 32-bit registers over an AXI4-Lite slave. The host writes the target and the BAM constant, sets the start bit, polls busy, and reads back the phases; a two-state (`IDLE`/`COMPUTE`) FSM sequences the core. Read-back serves verification only; the PIN diodes are driven directly from the phase register file.

TABLE I
SEQUENTIAL (P=1) VS. 4-WAY SIMD (P=4), POST-ROUTE AT 150 MHZ, $N = 256$

Metric	P=1	P=4	Ratio
Frame cycles (to `done`)	279	87	3.21×
Frame latency	1860 ns	580 ns	3.21×
LUTs / FFs	1212 / 1196	2642 / 3086	2.18× / 2.58×
DSP48 / BRAM	5 / 0	20 / 0	4.00× / –
WNS	+0.208 ns	+0.075 ns	–

TABLE II
COORDINATE FORMAT VS. PHASE ACCURACY (GOLDEN MODEL, 400 TARGETS PER REGIME)

Metric	Q4.12	Q8.8
Range LSB / phase per LSB	0.24 mm / 8.8°	3.9 mm / 140.6°
Max. phase error (front / beside)	7° / 9°	122° / 165°
Mean gain loss (front / beside)	0.01 dB / 0.01 dB	2.4 dB / 3.0 dB
Worst-target gain loss	0.3 dB	4.8 dB

IV. IMPLEMENTATION AND VERIFICATION

A. FPGA Implementation

Single-lane (P=1) and 4-lane (P=4) designs were implemented for the Zynq-7020 (`xc7z020clg400-1`) with a 150 MHz (6.667 ns) clock constraint (Table I). DSP48 usage scales 4× (per-lane multipliers) and LUTs and FFs sub-linearly (2.18×, 2.58×; dispatch, addressing and drain logic are fixed overhead). Frame latency improves 3.21× rather than 4× (279 vs. 87 cycles) because the 23-cycle fill in (2) (L_{pipe} plus the `done` cycle) does not scale with lane count. P=4 uses 2642 LUTs (≈5.0%), 3086 FFs (≈2.9%), 20 DSP48s (≈9.1%) and no BRAM; the square-root core accounts for 585–587 LUTs and 630–647 FFs per lane (over 80% of a lane). Vivado's post-route power estimate for P=4 (vectorless, confidence Medium) is 0.230 W on-chip, of which 0.126 W is dynamic and 0.104 W static: about 73 nJ of dynamic energy per 580-ns update.

Timing at 200 MHz does not close for P=4 (WNS $= -1.259$ ns, 222 failing endpoints). Registering each recurrence stage (L_{pipe} 22→37, frame 87→102 cycles) closes 200 MHz with WNS $= +0.084$ ns and cuts frame latency from 580 ns to 510 ns; all closed-loop experiments use the 22-cycle, 150 MHz core.

B. Numerical Accuracy

At fixed datapath width, precision trades phase error against beamforming quality. Against the ideal float-range 2-bit profile (golden model, 400 targets per regime), the per-element phase error with the global offset removed has a median of 2° and a maximum of 7° for targets 0.5–1 m in front of the array and 9° for targets 0.04–0.08 m beside it at 0.05–0.2 m height, with a mean gain loss of 0.01 dB (worst target 0.3 dB). A Q8.8 format at the same width is too coarse at 30 GHz (3.9 mm LSB, 140.6° per range LSB; maxima of 122° and 165°; mean losses of 2.4 and 3.0 dB; Table II). Only the ROM contents and K_λ differ between formats; the RTL is identical.

C. Bit-Exact and Semantic Verification

A Python golden model implements every fixed-point primitive bit-exactly (Q4.12 saturation, the 16-stage recurrence, BAM

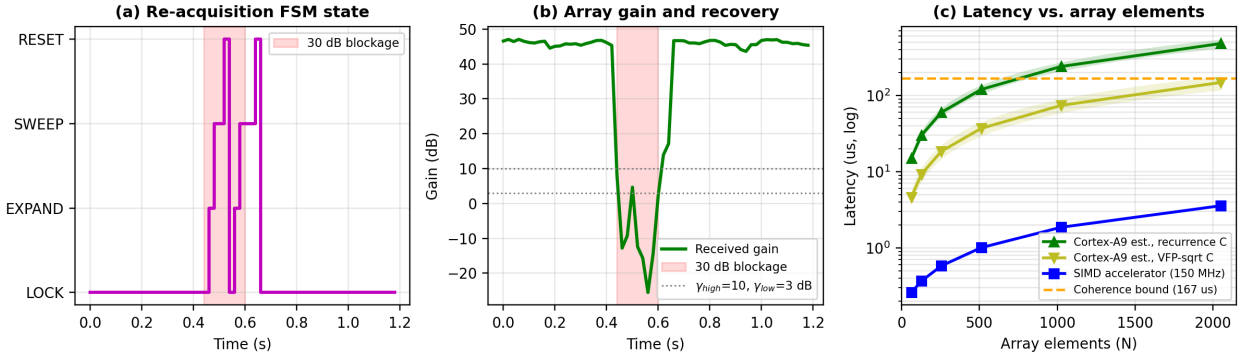


Fig. 2. System-level validation: (a) FSM state and (b) array gain under a 30 dB obstruction; (c) update latency vs. the coherence bound. Cortex-A9 curves (667 MHz) are idealized simulation estimates for two scalar C variants (shaded: range); the accelerator curve is $L(N, 4) = N/4 + 23$ cycles at 150 MHz, verified at $N = 256$ and projected elsewhere.

scaling and quantization) and generates the testbench vectors, ROM contents and C reference outputs. A SystemVerilog testbench drives the core through the AXI4-Lite interface for three reference targets: all 48 registers (768 two-bit states) match, and assertions confirm a 22-cycle lane latency and `done` on cycle 87. The core also matched the golden model on 832 further targets with varied K_λ (216 saturation/corner, 16 on-element, 600 random; 212,992 states), and the square-root unit matched exact integer roots on 2.13M radicands (all perfect squares and predecessors, extremes and random values, with and without input gaps). These checks establish equivalence to the golden model; a beam-pattern-level semantic check addresses the specification itself.

Because bit-exact checks share their specification with the model, two common-mode faults in an earlier Q8.8 implementation passed them. First, an early BAM formulation, $\theta = (R_{Q8.8} K_\lambda) \gg 8 \bmod 2^{16}$, repeated a shift already absorbed into $K_\lambda = 65536/(256\lambda)$, attenuating phase by $256\times$; model and RTL agreed, co-simulation reported a 100% pass, and the fault surfaced only because every element quantized to the same state. Second, the closed-loop bridge computed K_λ as $(65536/\lambda)\,\&\,0\text{xFFFF} = 0$ instead of the golden 25600; the RTL applied the value it was given and the regression, which uses the golden constant, still passed. The flat profile gave a plausible gain near boresight and was exposed by 3–22 dB off boresight, against about 44 dB after correction. The bridge now checks every returned phase matrix against a golden model with its own K_λ, and the array-factor metric of Sec. V-C serves as the semantic check.

V. System-Level Validation

A. RTL Co-Simulation Bridge

A Python server (`ipc_hw_runner.py`) receives a packet with the frame ID, Q4.12 target coordinates and BAM constant over a lockstep TCP socket, runs the compiled RTL testbench under Icarus Verilog [13], and returns the 16 phase registers from `hdl_output.hex` with the verified 87-cycle latency. The tracker blocks on each reply, synchronizing the 60-frame trajectory with the RTL frame by frame.

B. Tracking and Blockage Scenario

A smooth 3-D trajectory spans 60 epochs over a $44 \times 32 \times 20$ cm volume at $z \approx 0.85$ m (peak 2.5 m/s). A constant-velocity Kalman filter [14] observes $\mathbf{z}_k = \mathbf{p}_k + \mathbf{n}_k$, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \sigma_r^2 \mathbf{I}_3)$, $\sigma_r = 12$ mm, without access to the true position, and writes its one-step prediction $\hat{\mathbf{p}}_{k+1} = (\mathbf{F}\mathbf{x}_{k|k})_{1:3}$ to the accelerator each epoch; its error is 29 mm RMS outside the blockage window.

A four-state FSM (TRACKING_LOCK, COVARIANCE_EXPAND, CODEBOOK_SWEEP, LINK_RESET) compares the RTL-evaluated gain G_{k-1} of the previous epoch with $\gamma_{\text{high}} = 10$ dB and $\gamma_{\text{low}} = 3$ dB: lock is kept above γ_{high}, the process covariance $\mathbf{Q}$ is inflated between the thresholds, tracking is suspended for a localized codebook sweep below γ_{low}, and recovery above γ_{low} resets the covariance to $\mathbf{P}_0$ and returns to lock. A 30 dB extrinsic attenuation is injected over frames 22–30, during which the RF position measurement is rejected and the filter coasts on its prediction; the FSM sees only the composite gain.

C. Semantic Beam-Pattern Metric

Link quality is the coherent sum of the quantized phasors,

$$G = 10 \log_{10} \left| \sum_{m,n=1}^{16} w_{m,n}\, e^{j[\Phi_{m,n} - k\Delta d_{m,n}]} \right|^2, \qquad (3)$$

with $\Phi_{m,n} \in \{0, \pi/2, \pi, 3\pi/2\}$, $\Delta d_{m,n}$ the path-length differential to the target, and $w_{m,n} = (\hat{z}/R_{m,n})^q$, $q = 1.2$, an obliquity weight: the field form of the $\cos^{2q}$ element pattern used for low-gain patch-type RIS elements [15]. The empirical value $q = 1.2$ gives a broadside element gain of $2(2q+1) = 6.8$ (8.3 dBi), within the 3–9 dBi quoted for patch elements [15]. The ideal cophasal ceiling is $10 \log_{10} N^2 \approx 48.16$ dB. With the steering target at the true position, the Q4.12 phases give a mean gain of 47.07 dB over 500 targets in the trajectory volume, 0.007 dB below the ideal float-range 2-bit profile and 1.09 dB below the ceiling, against an average 2-bit quantization loss of 0.91 dB ($\eta_q = \text{sinc}^2(\pi/2^B) \approx 0.81$ for $B = 2$ [4], [5]). In the closed loop, steering on the filter's one-step prediction gives a mean gain of 46.0 dB outside the blockage window (43.6–47.1 dB).

D. Closed-Loop Results

Over the 60-frame run (Fig. 2), every phase matrix returned by the RTL matches the golden model, and under the injected 30 dB obstruction (frames 22–30) the FSM executes all four states and returns to lock. A 500-trial Monte Carlo study randomizes trajectory amplitude and phase, noise seed, blockage onset (frames 10–40) and duration (6–12 frames), using golden-model phase states (matched to the RTL on every vector tested). All four states executed in 96.4% of trials; in the other 3.6% the gain stayed above γ_{low} and the FSM did not reach CODEBOOK_SWEEP. In every trial lock returned within five frames of the last blocked frame, with no excursion outside the blockage and recovery interval.

E. Latency Context

Under the Jakes model [16], [17], $T_c \approx 9/(16\pi f_D)$ with $f_D = v f_c/c$; at 30 GHz, $T_c = 167$ μs corresponds to $v \approx 10.7$ m/s, and the evaluated trajectory (peak 2.5 m/s) has $T_c \approx 0.72$ ms. The 580-ns update is therefore 0.35% of the 167-μs window, and (2) projects 1.9 μs at $N = 1024$ and 3.6 μs at $N = 2048$ for the same four lanes, excluding ROM and register-file growth (Fig. 2c). For reference, bit-exact scalar C versions of the computation (`gcc -O2`) were timed for the 667-MHz Cortex-A9 by replaying QEMU instruction traces through an idealized dual-issue pipeline model [18]. These simulated estimates (15–68 μs at $N = 256$) also fit the window at this size. The accelerator's advantage at $N = 256$ is an update time independent of operating-system load, processor cores left free for the tracker, and one added cycle per four additional elements.

VI. Scope and Conclusion

Scope. Results derive from post-route implementation and RTL co-simulation; the 126 mW figure is a vectorless estimate. The gain metric models aperture phase coherence, excluding path loss, mutual coupling, edge diffraction and unit-cell reflection loss. Q4.12 spans ± 8 m. Determinism applies to the phase-update datapath: the 87-cycle figure excludes the host's AXI4-Lite coordinate writes, which were not timed. The closed loop uses injected attenuation and rejected position measurements; Cortex-A9 times are simulation estimates. The design targets 2-bit quantization, a 16×16 grid at 30 GHz and a near-linear-motion estimator, with a $+0.075$ ns margin at 150 MHz.

The accelerator computes near-field RIS phase states directly from target coordinates instead of selecting them from a codebook or learned model. A data-independent non-restoring square-root pipeline and 2-bit quantization deliver a complete 256-element update in 87 cycles (580 ns) at 150 MHz on a Zynq-7020, 0.35% of the 167-μs coherence bound at 30 GHz. Q4.12 coordinates hold the phase error to 7–9° with 0.01 dB mean gain loss relative to an ideal floating-point range, and bit-exact and semantic verification with closed-loop RTL co-simulation confirm operation in a tracking and blockage-recovery workload. Next steps are board-level evaluation with a physical RIS, a smaller square-root core, and closed-loop evaluation of the 200-MHz deeper-pipeline variant.

References

[1] Q. Yu, L. Dai, and M. Jian, "Near-field wideband beamforming for RIS based on Fresnel zone," in *Proc. IEEE GLOBECOM*, 2024, pp. 3980–3985.

[2] G. C. Alexandropoulos, V. Jamali, R. Schober, and H. V. Poor, "Near-field hierarchical beam management for RIS-enabled millimeter wave multi-antenna systems," in *Proc. IEEE SAM*, Trondheim, Norway, 2022, pp. 460–464.

[3] X. Zheng, W. Cheng, J. Wang, and W. Zhang, "Location-driven beamforming for RIS-assisted near-field communications," *IEEE Communications Magazine*, vol. 63, no. 1, pp. 44–50, Jan. 2025.

[4] Q. Wu and R. Zhang, "Beamforming optimization for intelligent reflecting surface with discrete phase shifts," in *Proc. IEEE ICASSP*, 2019, pp. 7830–7833.

[5] A. Haskou and A. Khaleghi, "On the effect of RIS phase quantization on communications system performances," in *Proc. IEEE IWCMC*, 2023, pp. 1406–1411.

[6] M. Ouyang, F. Gao, Y. Wang, S. Zhang, P. Li, and J. Ren, "Computer vision-aided reconfigurable intelligent surface-based beam tracking: Prototyping and experimental results," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 8681–8693, 2023.

[7] Y. Zhao, Y. Feng, A. M. Ismail, Z. Wang, Y. L. Guan, Y.-X. Guo, and C. Yuen, "2-bit RIS prototyping enhancing rapid-response space-time wavefront manipulation for wireless communication: Experimental studies," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 4885–4901, 2024.

[8] V. Tapio, M. Juntti, I. Hemadeh, and A. Mourad, "Near-field beamforming at intelligent re-configurable surface in uplink transmission," in *Proc. IEEE ICC Workshops*, 2021, pp. 1–5.

[9] L. Urbinati, N. Decarli, F. Guidi, A. Guerra, B. M. Masini, and A. Zanella, "On-device deep learning for RIS beamforming: Design and latency-rate trade-offs," *IEEE Open J. Commun. Soc.*, vol. 7, pp. 4605–4622, 2026.

[10] G. A. Works, "Distributed beam steering computer," US Patent 4,445,119, Raytheon Company, Apr. 1984.

[11] S. E. Zegrar, L. Afeef, and H. Arslan, "A general framework for RIS-aided mmWave communication networks: Channel estimation and mobile user tracking," arXiv:2009.01180, 2020.

[12] D. Yu, G. Zheng, A. Shojaeifard, S. Lambotharan, and Y. Liu, "Kalman filter based channel tracking for RIS-assisted multi-user networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3856–3869, Apr. 2024.

[13] "Icarus verilog: Open-source Verilog simulation engine," https://steveicarus.github.io/iverilog/, accessed 2026.

[14] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[15] S. W. Ellingson, "Path loss in reconfigurable intelligent surface-enabled channels," arXiv:1912.06759, 2019.

[16] W. C. Jakes, *Microwave Mobile Communications*. New York, NY, USA: Wiley, 1974.

[17] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, 2002.

[18] "GCC Cortex-A9 pipeline description, `cortex-a9.md`," https://gcc.gnu.org/, accessed 2026.