Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Error Correction and Feedforward Timing in a Simulated Dual-Rail Photonic Architecture

Aarti Singla
Thapar Institute of Engineering & Technology
Angel Garg
Thapar Institute of Engineering & Technology
Anil Singh
anils.rawat@thapar.edu

Thapar Institute of Engineering & Technology
Manu Bansal
Thapar Institute of Engineering & Technology
Vibhu Srivastava
Shastrah.AI
Sunny Sharma
Indian Institute of Information Technology Allahabad

Research Article

Keywords:

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11538004/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Error Correction and Feedforward Timing in a Simulated Dual-Rail Photonic Architecture

Aarti Singla[1], Angel Garg[1], Anil Singh[1*], Manu Bansal[1], Vibhu Srivastava[2], Sunny Sharma[3]

[1*]Department of Electronics and Communication Engineering, Thapar Institute of Engineering & Technology (TIET), Patiala, India.
[2]Shastrah.AI.
[3]Department of Electronics and Communication Engineering, Indian Institute of Information Technology Allahabad, Prayagraj, India.

*Corresponding author(s). E-mail(s): anils.rawat@thapar.edu;
Contributing authors: asingla7_be24@thapar.edu;
agarg8_be23@thapar.edu; mbansal@thapar.edu;
vibhu.09305155@gmail.com; sunnymeharwal@gmail.com;

Abstract

In a linear-optical quantum computer, error correction must finish before the encoded photon leaves the optical delay line. We present a 33-mode dual-rail architecture for single-error detection and evaluate repeated-syndrome $d = 3$ correction using an extended 55-mode circuit-level simulation model. We co-design the timing budget and an FPGA decoder. On a Basys3 board, the decoder was validated at $d = 3, 5, 7$ using 40,000 Stim-generated syndrome vectors per distance (120,000 total), with no mismatches. Under a reference two-term circuit-level Pauli noise model, finite-size scaling gives a threshold estimate of $p_{\text{th}} = 0.194 \pm 0.001$, with statistical uncertainty only; the fitted suppression exponents are consistent with $\lceil d/2 \rceil$. Post-place-and-route timing gives a conservative decoder cycle of 8.79 ns (113.8 MHz). With assumed 50-ns detection and 10-ns switching latencies, decoding uses about 9% of the 99.8-ns optical-buffer interval, identifying detector readout as the main target for latency reduction. The stated source-quality model also yields a photon-indistinguishability bound of $V > 0.982$; this is a model-dependent estimate, not an experimental measurement.

Keywords: Photonic quantum computing, dual-rail encoding, quantum error correction, minimum-weight perfect matching, FPGA decoder, feedforward control

1 Introduction

Photonic quantum computing offers room-temperature operation, compatibility with complementary metal–oxide–semiconductor (CMOS) processes, and potential integration with quantum communication networks [1, 2]. In linear-optical quantum computing (LOQC), dual-rail encoding represents a logical qubit as one photon occupying one of two modes. Combined with linear-optical elements and feedforward, this encoding supports universal computation in the KLM scheme [3–5].

Error correction in this setting has a physical deadline. While the syndrome is measured and decoded, the encoded photon travels through a fibre delay line; the correction signal must reach the switching element before the photon arrives. The fibre length therefore sets the time available to the classical electronics and constrains the decoder implementations that can be used. Photon-source quality is also relevant because imperfect indistinguishability between interfering photons contributes to gate error.

These issues are often studied separately. Photonic fault-tolerance proposals specify codes, thresholds, and loss budgets [11, 12], whereas decoder studies typically report latency and resource use for superconducting hardware [16, 19]. A joint demonstration of code, decoder, and feedback has been reported on another platform: Aurora combines a distance-2 repetition code with FPGA decoding and single-cycle feedback in a networked, continuous-variable (CV) cluster-state system [17]. Aurora encodes information in squeezed light and measures it by homodyne detection. By contrast, the discrete-variable (DV) dual-rail encoding studied here represents information in single-photon rail occupation; photon loss can, in principle, be heralded as an erasure at the detection event. We bring the architecture, decoder, and feedback budget together for this DV setting. The contribution is not a new repetition code or matching algorithm, but a quantitative co-design that links a mode-resolved photonic architecture, circuit-level QEC simulations, an FPGA decoder, and a system-level latency budget under an explicit optical-delay constraint.

At $d = 3$, the decoder's conservative 8.79-ns cycle occupies 8.8% of the 99.8-ns feedforward window, compared with the assumed 50-ns detection latency.

1.1 Contributions

1. A mode-resolved 33-mode architecture for a dual-rail repetition-code module, with an explicit rationale for the mode count and a demonstration of logical-state preservation under the adopted circuit-level model (Section 3).
2. A feedforward timing budget that distinguishes assumed, timing-derived, and literature-informed quantities (Section 4).
3. A closed-form $d = 3$ decoder implemented in RTL and carried through FPGA place-and-route, together with a reference-generated lookup decoder whose single-round scaling is additionally evaluated through standard-cell synthesis. Both implementations are verified over their respective complete syndrome spaces against a reference matching decoder (Section 5). The closed-form decoder's underlying code-capacity graph is further extended, as a table-driven implementation, to $d = 5, 7$ and

hardware-validated on the same Basys3 board used for $d = 3$, with zero mismatches across 120,000 syndrome vectors total.

4. A finite-size-scaling threshold estimate and sensitivity analysis using the extended circuit-level simulation configuration, together with a quantitative photon-source requirement and ablation study (Sections 6–7).

The paper is organised as follows. Section 2 reviews the background and related work; Section 3 describes the architecture; and Section 4 develops the feedforward latency budget. Section 5 presents the decoder implementations, and Section 6 details the simulation methods. Section 7 reports the results, followed by a discussion of code-family trade-offs and scaling in Section 8. Sections 9 and 10 cover limitations and reproducibility, respectively, before the conclusion in Section 11.

2 Background and Related Work

2.1 Dual-Rail Encoding

Dual-rail encoding [3, 9] represents a logical qubit using a single photon occupying one of two optical modes, or rails:

$$|0_L\rangle = |1, 0\rangle, \qquad |1_L\rangle = |0, 1\rangle, \tag{1}$$

where the notation $|n_0, n_1\rangle$ denotes photon occupation number in rails 0 and 1. A general logical state is the superposition $|\psi\rangle = \alpha|1, 0\rangle + \beta|0, 1\rangle$, realised experimentally by distributing a single photon across the two rails with a beamsplitter or directional coupler, with α and β set by the coupling ratio and relative phase. Unlike matter-qubit encodings, the dual-rail code space has a fixed, physically observable photon number. If a photon is lost from either rail, the state $|\psi\rangle$ maps deterministically to the vacuum state $|0, 0\rangle$, which contains no photon at all, lies outside the logical code space, and is therefore an event distinguishable in principle from any valid logical state.

This distinguishability is the reason loss can be treated as an erasure [8], a location-known error, rather than as an ordinary depolarizing error. On platforms where loss is not separately observable (for instance, a photon that survives but undergoes an unheralded phase error), the decoder must infer both the presence and the location of an error from the syndrome alone. Here, loss events can in principle be flagged by a coincidence check at the point of detection, before the syndrome is even decoded. The present architecture does not yet exploit this heralding advantage in its decoding logic; the implications are discussed in Section 9. The erasure-channel results in Section 7 depend on this distinction between photonic and matter-qubit encodings.

2.2 Code Distance and Noise-Model Regime

Two common noise-model regimes are relevant for repetition-code simulations: *code-capacity* noise, in which syndrome measurements are assumed perfect, and *repeated-syndrome* noise, in which measurement errors are included. Under the code-capacity regime, a distance-d repetition code uses exactly d data qubits and a single, perfectly reliable syndrome measurement round. This regime isolates the combinatorics

of the decoding problem from the physical noise process, but does not model errors introduced during syndrome extraction itself.

The *repeated-syndrome* regime, adopted throughout this work, accounts for that idealisation directly. A distance-d memory under this regime uses d data qubits and $d-1$ measurement ancillas, $2d-1$ physical qubits in total. We use d measurement rounds in the simulations; the underlying circuit construction permits an arbitrary number of rounds. The additional $d-1$ ancillas and repeated measurement rounds allow the decoder to distinguish data errors from measurement errors. This is the circuit-level regime used by Stim [26] throughout the present work, and it is the regime under which every threshold and suppression exponent reported in Section 7 should be read.

Under the repeated-syndrome regime, the 33-mode architecture, built from three dual-rail data qubits and a single syndrome round, provides single-error detection but is not the circuit-level $d = 3$ memory used for the threshold simulations below, since that requires repeated syndrome rounds to distinguish a genuine data error from a faulty measurement. Achieving full $d = 3$ correction under this regime requires three data qubits and two measurement ancillas (five physical qubits total, following the $2d-1$ accounting above), evaluated over three measurement rounds, which is why the $d = 3$ correction results in this paper are reported for an extended 55-mode simulation rather than the 33-mode architecture. The RTL decoder (Section 5) is verified for functional correctness against the closed-form reference decoder over the complete syndrome space. The threshold and suppression results are obtained from software simulation using the same noise model. The scope and validation basis of the principal results are summarised in Table 1.

Table 1 Scope of Reported Results

Claim	Status	Basis
33-mode module, single-error detection	Architectural target	Architecture spec (this paper)
Threshold, ν, suppression (d=3, 5, 7)	Simulated	Extended circuit-level repetition-code simulations, not the 33-mode architecture
RTL decoder correctness	Verified	Exhaustive check plus 40,000-sample co-simulation
RTL decoder latency (8.79 ns)	Timing-derived	Post-place-and-route, Xilinx Artix-7
Decoder hardware validation	Hardware-validated	Basys3 board, 40,000 Stim vectors, 0 mismatches (Section 5)
SNSPD[a] / Pockels timing (50/10 ns)	Assumed	50 ns assumed; 10 ns literature-informed
HOM[b] visibility bound (V>0.982)	Analytical + Monte Carlo	From stated relations; not bootstrapped (Section 7)

[a]Superconducting nanowire single-photon detector. [b]Hong–Ou–Mandel.

2.3 Stabilizers and Syndrome Extraction

Error detection in a stabilizer code [22] proceeds by measuring a set of operators, the stabilizer generators, chosen so that they commute with the logical operators of the code but anti-commute with the errors the code is designed to catch. For the three-qubit bit-flip repetition code, the generators are the adjacent-pair parity checks $\{ZZ_{01}, ZZ_{12}\}$, where ZZ_{ij} measures whether qubits i and j agree in the computational basis. A single bit-flip on qubit q_1, for instance, flips the outcome of both ZZ_{01} and ZZ_{12}, while a bit-flip on q_0 flips only ZZ_{01}; the resulting syndrome pattern is therefore diagnostic of both the presence and the location of the error. These generators are measured non-destructively through ancilla-assisted beamsplitter interactions (Section 3), in which each data qubit is coupled to an ancilla mode that acquires the parity information without being measured itself until the coupling is complete. Experimental precedents include a linear-optical quantum parity check [31] and a two-photon parity code built from a data photon interfered with an ancilla photon at a polarizing beamsplitter [30]. A deterministic, general-purpose single-ancilla nondestructive ZZ measurement on an arbitrary dual-rail qubit has not been experimentally demonstrated in these cited linear-optical implementations. This interaction is therefore treated as an architectural target rather than an experimentally established primitive. The conjugate generator set $\{XX_{01}, XX_{12}\}$, sensitive to phase-flip rather than bit-flip errors, follows by conjugating the same circuit with Hadamard-equivalent MZI (Mach-Zehnder interferometer) operations [10]; this is the structural basis for the dual phase-flip configuration discussed below. The complete single-round syndrome-to-error mapping is $00\rightarrow$ no error, $10\rightarrow q_0$, $11\rightarrow q_1$, $01\rightarrow q_2$, and Algorithm 2 in Section 5 gives the closed-form decision rule that realises this mapping in hardware.

Which generator set is actually measured is not a property of the circuit alone but depends on how the data qubits are initialised. Preparing the data rails in the computational basis ($|0\rangle$ or $|1\rangle$) produces a state that is a $+1$ eigenstate of every ZZ generator, so measuring ZZ yields deterministic, noise-free outcomes in the absence of errors, and the module functions as a bit-flip memory. Preparing instead in the conjugate (Hadamard) basis produces a $+1$ eigenstate of the XX generators, giving the dual phase-flip memory. Both configurations were simulated independently and are shown in Section 7 to overlay within point-wise statistical error, a qualitative consistency check rather than an independent threshold fit for the X-memory configuration, which is expected because the two circuits are related by a basis change that leaves the underlying decoding graph unchanged. The two configurations are, however, mutually exclusive on a single execution of the module: a state cannot simultaneously be a $+1$ eigenstate of both the ZZ and XX generator sets on this code, so a single run of the repetition code protects against one error type or the other. Achieving simultaneous protection against both bit-flip and phase-flip errors is precisely the problem CSS (Calderbank-Shor-Steane) codes [23, 24] solve, at the cost of a more demanding stabilizer structure; that extension is identified as future work in Section 11.

2.4 Related Work

Photonic fault-tolerance architectures. Fusion-based quantum computation [12] and manufacturable photonic blueprints [11] specify complete architectures at the level of resource states, fusion networks and per-photon loss budgets. Here we instantiate a mode-level specification for a single small-distance code and add the classical conditional timing-budget analysis those proposals leave to future engineering.

Integrated hardware demonstrations. Aurora, a networked continuous-variable cluster-state photonic system, has demonstrated joint real-time decoding and single-clock-cycle feedforward for a distance-2 repetition code in physical hardware [17], alongside a detailed loss-tolerance analysis, providing a fabricated, operating device. The present work instead considers a discrete-variable dual-rail architecture and combines mode-level design, circuit-level QEC simulation, and FPGA decoder hardware implementation, without a fabricated photonic device. The differing encodings lead to distinct loss models and syndrome representations: Aurora encodes on continuous quadratures of squeezed light, this design on which-rail occupation of single photons, so a lost photon here is a direct, per-event erasure, whereas loss and excess noise in a continuous-variable system manifest as a shift in a homodyne-measured quadrature. Because the two systems use different physical encodings and loss models, we compare them qualitatively. Here, the integration occurs at the architectural and simulation levels, through a shared mode budget, noise model, and timing analysis (Section 9).

Real-time decoding. Hardware decoders for surface codes have been demonstrated on FPGA and ASIC at distances far beyond the one considered here [16], establishing that classical decoding need not limit a feedback loop. Those results are obtained on superconducting platforms, where the deadline arises from qubit coherence and readout rather than from an optical delay line, and where the decoding graph is orders of magnitude larger. Closer in technique to the present decoder, LILLIPUT demonstrates that a lookup-table decoder can achieve tens-of-nanoseconds latency and under 7% FPGA logic utilisation for surface-code decoding by translating syndromes directly into table indices [18]. The decoder described in Section 5 uses the same lookup principle at a much smaller scale: a two-detector single-round decision and a multi-round table generated from the reference decoder, appropriate to the repetition code's small syndrome space. We evaluate this decoder through place-and-route closure and standard-cell synthesis against a stated latency budget.

Erasure-aware decoding. Optimal linear-time maximum-likelihood decoding over the quantum erasure channel is established [28], and erasure conversion has been shown to raise thresholds substantially when the dominant error is detectable [29]. The erasure results reported below confirm this behaviour for the dual-rail repetition code rather than extending it.

Feedforward LOQC. Feedforward-corrected linear-optical error correction was demonstrated in the laboratory two decades ago, using a two-qubit encoding and Pockels-cell feedback [30]. The present work extends this approach to a mode-resolved dual-rail architecture with an implemented decoder and timing analysis.

3 33-Mode Architecture

3.1 Mode Budget and Design Rationale

The 33-mode count follows from the module's functional partitioning. The three dual-rail data qubits use six modes. The remaining modes support correction-relevant syndrome extraction, post-CZ readout, the delay buffer, the post-selected CZ gate and its ancillas, heralded injection, and inter-block routing (Table 2). Reducing the allocation below the uniform six-mode-per-stage width would require removing a syndrome stage or reusing modes within a logical cycle, thereby coupling syndrome extraction to the data path. A further modest reduction may be possible if the uniform allocation is relaxed, as noted in the table footnote.

Table 2 Physical Mode Assignment for the 33-Mode Architecture

Mode	Ct.	Functional Role	Active Layers
M01–M06	6	Data rails (3 dual-rail qubits)	L01, L02, L06, L08, L11
M07	1	Inter-block routing switch	L02, L11
M08–M13	6	Correction-relevant syndrome ancilla[a]	L03, L10
M14–M19	6	Buffer delay lines	L05
M20–M25	6	Post-selected CZ ancilla	L08
M26–M31	6	Post-CZ readout syndrome ancilla	L07, L09
M32–M33	2	Heralded-injection verification	L01
Total	**33**		

[a] 4 of these 6 modes are logically occupied by the named ancilla variables a_{01}, a_{12}; the remaining 2 are provisioned for control-electronics symmetry, not logically required (see text below).

The six-mode allocation preserves a uniform physical width across the eleven pipeline stages for scheduling and control-electronics symmetry. Within the syndrome-ancilla group (M08–M13), four modes are required for the two dual-rail ancillas; the other two are provisioned for symmetry rather than logical necessity.

The extended circuit-level simulation uses the $d = 3$ configuration described in Section 2: three data qubits, two measurement ancillas, and three measurement rounds. This gives the 55-mode model used for the correction results in Section 7, following the same functional-partitioning principle as Table 2. The 55-mode configuration is a simulation-level representation, not a fully routed optical design. The $d = 5$ and $d = 7$ circuits are specified programmatically in the simulation scripts described in Section 10.

3.2 Eleven-Layer Pipeline

Figure 1 shows the pipeline. The layer order was designed manually to satisfy the feedforward deadline; it was not generated by an automated scheduler. Preparation and encoding (L01–L02) precede syndrome extraction (L03–L04), so that parity information is available as early as possible. The delay-line duration follows directly from its physical length: for a fibre of length L and group index n_g, the transit time is

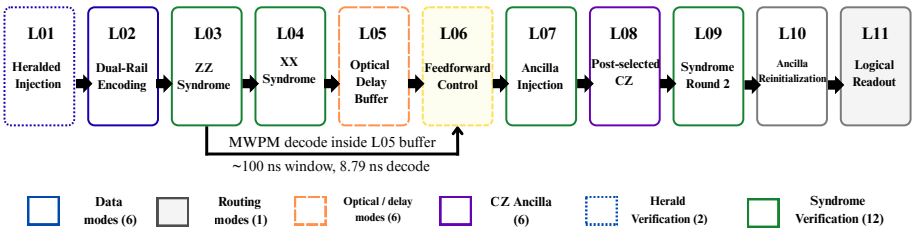


Fig. 1 Eleven-layer pipeline of the 33-mode module, showing the syndrome extraction (L03–L04), optical delay buffer (L05), and correction (L06) stages, coloured by functional mode group

$t = Ln_g/c$, so the chosen $L = 20.5\,\text{m}$ and $n_g = 1.46$ give $t = (20.5)(1.46)/(3{\times}10^8) \approx 99.8\,\text{ns}$, the $\sim$100 ns figure used throughout. This delay is a design choice: a longer fibre widens the correction window at the cost of additional switching-window overhead. The SMF (single-mode fibre) delay itself uses standard low-loss telecom fibre rather than the lithographic silicon photonic waveguides used for on-chip routing. At the standard telecom-fibre attenuation figure of $\sim$0.2 dB/km at 1550 nm, this SMF delay contributes approximately 0.004 dB of propagation loss over its 20.5 m length, negligible over this length. The 0.15 dB/cm figure used elsewhere in this paper's noise model (Section 6) characterises the on-chip waveguides, not this delay fibre, and should not be applied to it. The 20.5 m fibre delay (L05, n_g=1.46) holds the data modes for $\sim$100 ns while the syndrome is decoded, and the correction is applied at L06 before the photon leaves the buffer. Entangling and readout operations follow: the post-selected CZ gate (L08) has an assumed $\sim$75% discard rate, comparable to the 1/4 success probability reported for a nondestructive linear-optical CNOT (controlled-NOT) construction using an entangled ancilla pair [31]. A post-CZ readout syndrome (L09), ancilla reset (L10) and logical readout (L11) follow; the L09 syndrome serves entangling-operation readout only, and does not contribute to the L03/L04 correction decision. Placing the buffer between extraction and correction is what makes the loop closable; any other ordering would require the correction before the syndrome is known.

3.3 Logical-State Preservation and Architecture-Level Equivalence

The stabilizer structure provides the analytical basis for logical-state preservation. We assessed architecture-level equivalence separately through circuit-level simulation.

Analytically, non-interference follows from the code rather than from this implementation. The measured operators are stabilizers, which commute with the logical operators by construction; their measurement therefore projects onto a stabilizer eigenspace without acting on the encoded information. Only one property must therefore be guaranteed by the layout: that no unintended coupling occurs outside the designated interaction windows. Data and ancilla modes are routed on physically separate waveguides with spacing exceeding the evanescent coupling threshold ($> 10\,\mu$m, an assumed conservative bound rather than a figure taken from a specific fabrication PDK), and interact only through controlled couplers during the L03 and L04 time

slots. Following interaction, ancilla modes are routed directly to localised detectors and reset at L10, preventing residual optical power from re-entering the data path.

We compared a syndrome circuit built from the specified layer schedule and mode assignment with a generic code generator (Fig. 2). Across the swept range of p, their logical-error rates agree within statistical uncertainty. This supports equivalence under the adopted noise model, although it does not establish that the physical partitioning introduces no additional failure mechanisms. The comparison does not validate fabrication-level optical crosstalk, because the $> 10\,\mu$m spacing is a design assumption. The comparison is limited to $d = 3$; the architecture-specific circuit was not generalised to $d = 5, 7$, since the 33-mode budget in Table 2 is specific to the three-data-qubit layout. At $2{\times}10^5$ shots per point, the binomial standard error of both curves was combined in quadrature at each swept point ($\sigma_i = \sqrt{p_i(1-p_i)/\text{shots}}$ for each curve, summed in quadrature across the pair), and the absolute difference between the two curves was divided by this combined standard error. The largest disagreement between the two circuits is approximately 2σ, at the most sensitive point in the sweep; every other point agrees more closely than this.

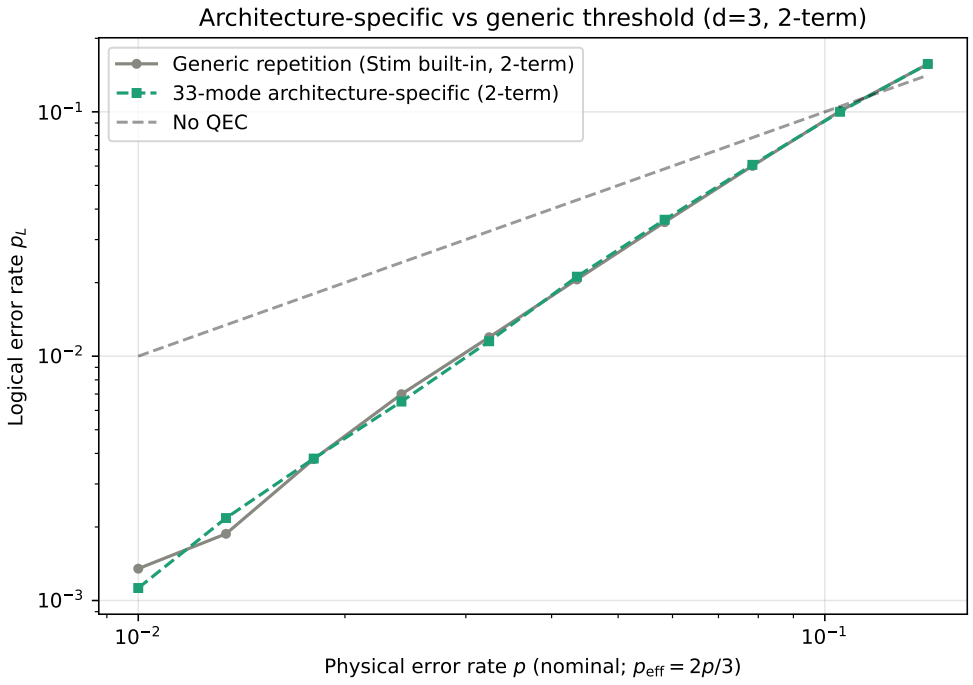


Fig. 2 Logical error rate versus physical error rate: architecture-specific circuit against the generic repetition-code circuit

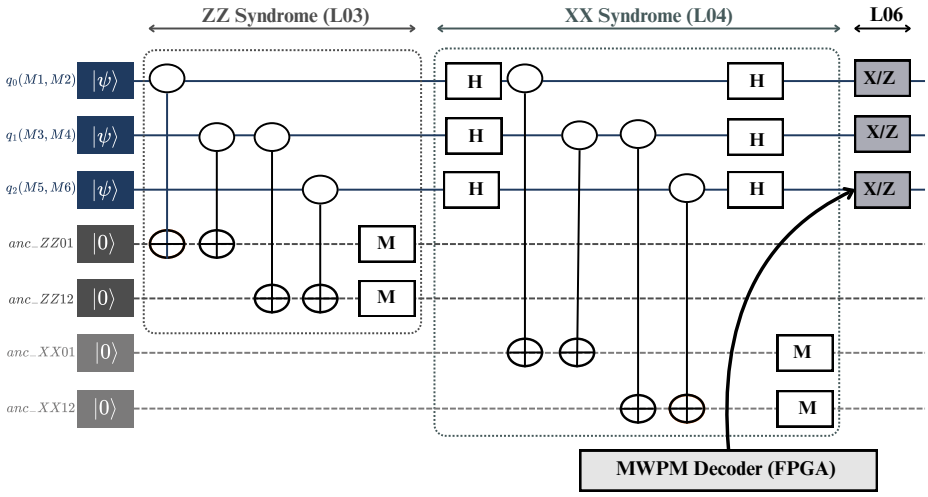


Fig. 3 Syndrome extraction for L03 (ZZ, bit-flip) and L04 (XX, structural, shown greyed). Ancilla outcomes feed the FPGA MWPM decoder, whose conditional correction is applied at L06. The XX path is instantiated structurally but not measured in the bit-flip configuration

Algorithm 1: Syndrome extraction round (bit-flip basis). ParityCouple denotes the logical effect of the ancilla-assisted beamsplitter interaction described in the surrounding text (Section 2): a beamsplitter-mediated parity coupling between each data mode and its ancilla, not a gate applied to abstract qubits and not a deterministic linear-optical CNOT primitive

Input: Data qubits q_0, q_1, q_2; ancillas a_{01}, a_{12}
Output: Syndrome bits s_{01}, s_{12}

Reset $a_{01} \leftarrow |0\rangle$, $a_{12} \leftarrow |0\rangle$;
`// parity of adjacent data pairs onto ancillas, via beamsplitter-mediated coupling (Section 2)`
ParityCouple(q_0, a_{01}); ParityCouple(q_1, a_{01});
ParityCouple(q_1, a_{12}); ParityCouple(q_2, a_{12});
$s_{01} \leftarrow$ measure(a_{01});
$s_{12} \leftarrow$ measure(a_{12});
Route a_{01}, a_{12} to local detectors; reset at L10;
return (s_{01}, s_{12});

3.4 Syndrome Extraction Circuit

Fig. 3 shows the extraction circuits for L03 (ZZ) and L04 (XX). Algorithm 1 states the round in full.

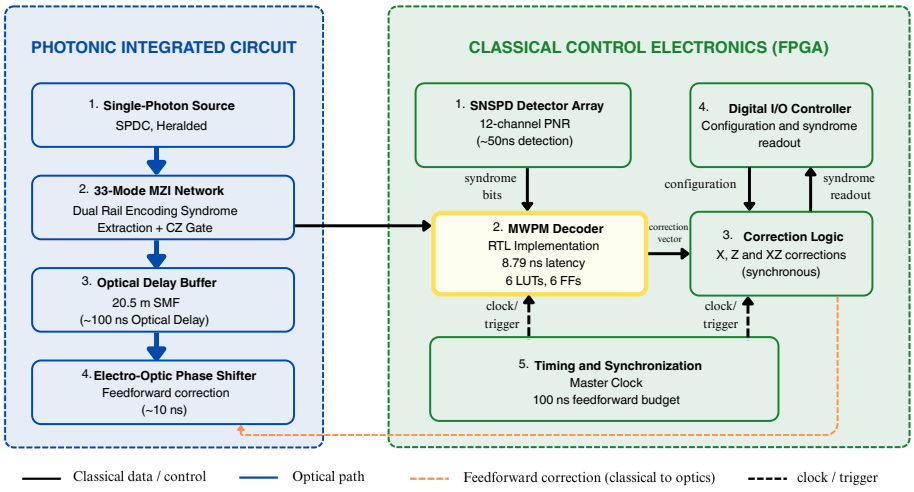


Fig. 4 Photonic–classical co-design and feedforward path. The latency bar at the bottom is drawn to scale: detection (50 ns, assumed), decoder (8.79 ns, timing-derived), and switching (10 ns, literature-informed assumption), against the 99.8 ns optical buffer

4 Feedforward Control and Latency Analysis

4.1 System Architecture

Fig. 4 illustrates the complete timing path. A heralded source [32, 33] announces each photon, ensuring that only successfully generated photons enter the circuit. The MZI network performs the mode transformations of the pipeline: dual-rail encoding, followed by the couplings that write parity information onto the ancilla modes. Data modes then enter the fibre delay, which holds them for approximately 100 ns and defines the available correction window.

All classical processing must finish within this interval. An array of superconducting nanowire single-photon detectors (SNSPDs) reads the ancilla modes to produce syndrome bits. The decoder processes those bits and drives the electro-optic switch, which applies the correction at L06 while the photon remains in the delay line. Two additional blocks support the loop without contributing to its latency: the timing and synchronisation unit distributes the master clock and gates the detectors against the layer schedule, and the digital I/O controller manages configuration and syndrome readout. The latency contribution of each stage is quantified next.

4.2 Latency Budget

The feedforward path has three components: detection, decoding, and switching.

$$t_{\mathrm{ff}} = \underbrace{50}_{\text{SNSPD}} + \underbrace{8.79}_{\text{decoder}} + \underbrace{10}_{\text{Pockels}} = 68.79 \text{ ns}, \tag{2}$$

against the 99.8 ns permitted by the optical buffer, leaving 31.01 ns of margin under the assumed component-level timing budget (Table 3): only the 8.79 ns decoder figure is a post-place-and-route timing-derived result. This sum aggregates each stage's internal processing time only; it does not separately itemise interconnect propagation delay between the physical stages (detector array to FPGA, FPGA to Pockels-cell driver). At a typical signal velocity of $\sim 0.66c$ in coaxial cable or PCB trace, this delay is approximately 5 ns per metre of cable run — for component separations of order 1–2 m, roughly 5–10 ns, a non-negligible fraction of the 31.01 ns margin not included in the total above. The 50 ns SNSPD figure is a design assumption, while the 10 ns Pockels-cell figure is a conservative value informed by published device-level timing; neither was measured for the present architecture. The Pockels-cell figure is reasonably well matched by a modern driver circuit built for laser Q-switching, reporting 7.3 ns rise and 7.8 ns fall times [35], close to though somewhat faster than the 10 ns assumed here. Here, the 50 ns value denotes the assumed end-to-end readout-chain latency from photon detection to a usable syndrome bit; no published measurement of this specific quantity was found, so it remains a design assumption rather than a literature-derived value. The closest available benchmark, a complete cryogenic feedforward chain co-located below 4 K, measures 23 ± 3 ns end to end [34], of which 20 ± 2 ns is attributed to trigger and discrimination logic alone — faster than the 50 ns assumed here, but achieved through cryogenic co-location rather than the separately-staged architecture assumed in this work. Under these assumptions, detection—not decoding—is the largest contributor to feedforward latency. Of the 68.79 ns actually spent, detection accounts for 72.7%, switching for 14.5%, and the decoder for 12.8%; separately, the decoder occupies only 8.8% of the full 99.8 ns budget. Detector latency is therefore the dominant target for further optimisation. For assumed detector latencies of 30–70 ns, the resulting feedforward margin ranges from 51 ns to 11 ns (Fig. 5).

Table 3 Feedforward Events and Latency Budget

Event	Trigger	Latency	Status	Action
FF-1	L03/L04 parity	<99.8 ns	Conditional[b]	Correct at L06
FF-2	L08 coincidence	<50 ns	—	Accept / retry
FF-3	L11 PNR[a]	Classical	—	Assign logical bit
FF-1 breakdown				
Detection	SNSPD array	50 ns	Assumed	Syndrome
Decode	FPGA	8.79 ns	Timing-derived	Correction
Switching	Pockels	10 ns	Literature-informed	Apply
Total		**68.79 ns**		
Budget		99.8 ns		
Margin		**31.01 ns**		

[a]Photon-number-resolving (detection). [b]Conditional on the assumed/derived component latencies in the FF-1 breakdown below, not itself a measured end-to-end result.

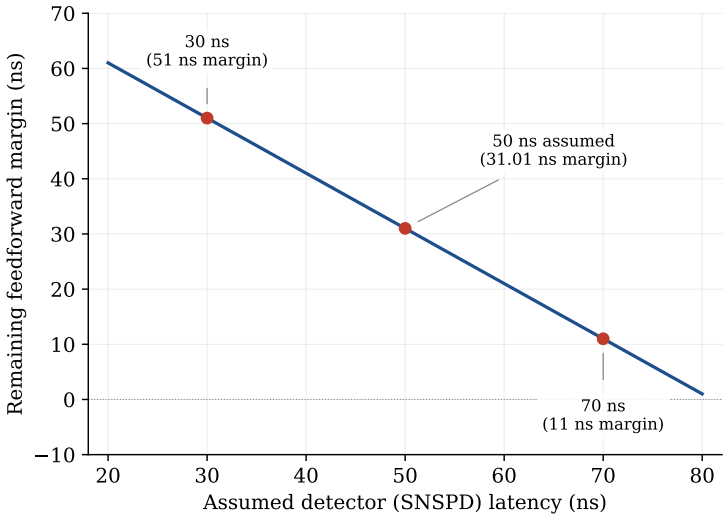


Fig. 5 Remaining feedforward margin as a function of the assumed SNSPD detector latency, at fixed decoder (8.79 ns) and switching (10 ns) contributions. The margin depends linearly on the detector-latency assumption; the 50 ns value used throughout this paper is marked alongside the 30–70 ns sensitivity range

FF-2 and FF-3 are bounded by separate constraints from FF-1's optical-delay-line budget analysed above: FF-2's <50 ns applies to the classical accept/retry decision at the post-selected CZ gate (L08), which discards roughly 75% of attempts (Section 3), meaning a retry is the common outcome rather than an edge case; FF-3 is a classical readout assignment with no comparable optical deadline. The 99.8 ns timing budget discussed throughout this work applies to the FF-1 syndrome-correction path only; the cumulative timing of repeated L08 retries, and any required retry bound, are outside the scope of the present analysis.

5 Decoder Design and Implementation

We use minimum-weight perfect matching (MWPM) to decode syndromes represented as graph-matching problems, following the approach established for topological quantum memories [21].

Two decoder implementations are evaluated: a closed-form $d = 3$ decoder carried through FPGA place-and-route and further validated on the Basys3 board, detailed below; and a reference-generated ROM lookup decoder, whose single-round scaling at $d = 3, 5, 7$ is separately evaluated through standard-cell synthesis (Section 8.2). They share the same syndrome definition but target different implementation regimes.

5.1 Closed-Form Matching at $d = 3$

For the single-round $d = 3$ repetition syndrome, the decoding problem reduces to two detector bits, d_0 and d_1, corresponding to the ZZ_{01} and ZZ_{12} parity checks. With only two detector bits there are four possible input patterns: three resolve directly

Algorithm 2: Closed-form minimum-weight matching, $d = 3$: the single-round, code-capacity-style decision rule for the two-detector syndrome

Input: Detector bits d_0 (ZZ_{01}), d_1 (ZZ_{12})
Output: Correction (c_0, c_1, c_2) on data qubits q_0, q_1, q_2
`// graph: BL --`q_0`-- D0 --`q_1`-- D1 --`q_2`-- BR, unit weights`

if $d_0 = 0 \wedge d_1 = 0$ **then**
  return $(0, 0, 0)$; `// no fired detector`
else if $d_0 = 1 \wedge d_1 = 0$ **then**
  return $(1, 0, 0)$; `// D0 to left boundary`
else if $d_0 = 0 \wedge d_1 = 1$ **then**
  return $(0, 0, 1)$; `// D1 to right boundary`
else
  `// both fired: the internal edge has weight 1, while the two boundary edges have total weight 2, so the minimum-weight decision is the central-qubit correction`
  $w_{\text{pair}} \leftarrow w(D_0 - D_1) = 1$;
  $w_{\text{bnd}} \leftarrow w(D_0 - \text{BL}) + w(D_1 - \text{BR}) = 2$;
  if $w_{pair} \leq w_{bnd}$ **then return** $(0, 1, 0)$;
  else return $(1, 0, 1)$;
end

with no comparison, and the fourth (both bits fired) requires comparing two candidate matchings, an $O(1)$ operation. For the unit-weight graph used here, the central correction is the minimum-weight choice when both detectors fire. The two weights compared in Algorithm 2 ($w_{\text{pair}} = 1$, $w_{\text{bnd}} = 2$) are fixed constants for this specific $d = 3$ unit-weight graph, not values that vary at runtime, so the comparison always resolves in favour of the central-qubit correction and the alternative branch is unreachable for this graph; the comparison is retained explicitly to show the underlying minimum-weight matching rule, rather than simply asserting the outcome, and would become runtime-dependent on a graph with non-uniform edge weights. This two-detector decoder is distinct from the four-detector, six-edge multi-round circuit-level graph obtained at $d = 3, r = 1$ under the $(d-1)(r+1)$ convention above; the latter is handled instead by the reference-generated lookup decoder of Section 5.2. Thus, the closed-form decoder addresses the single-round $d = 3$ syndrome used for the hardware implementation, whereas the reference-generated lookup decoder addresses the repeated-syndrome circuit-level model used for fault-tolerant simulations. This closed-form treatment is specific to $d = 3$: the number of syndrome patterns requiring exhaustive case coverage grows combinatorially with distance, making a hand-derived closed form impractical beyond this smallest nontrivial case, which is why the lookup-table approach of the next section is used for the $d = 5, 7$ scaling study rather than extending Algorithm 2 directly.

Algorithm 3: Reference-generated decode table

Input: Code distance d, rounds r, reference decoder $\mathcal{D}$
Output: Table T indexed by syndrome

Construct circuit $C \leftarrow \textsc{RepetitionMemory}(d, r)$;
$M \leftarrow \textsc{DetectorErrorModel}(C)$; $n \leftarrow |\text{detectors}(M)|$;
if 2^n *exceeds the resource budget* **then return** INFEASIBLE;
for $s \leftarrow 0$ **to** $2^n - 1$ **do**
  $\sigma \leftarrow$ binary expansion of s over n detectors;
  $T[s] \leftarrow \mathcal{D}(\sigma)$; `// reference MWPM prediction`
end
return T;

5.2 Multi-Round Decoding by Reference-Generated Lookup

Multi-round decoding, required in the presence of measurement error, operates on the full syndrome graph. Rather than reimplementing a matching engine, the decoder is realised as a lookup whose contents are generated by the reference matching decoder itself (Algorithm 3), making its output exactly equivalent to the reference decoder over the entire syndrome space.

The closed-form $d = 3$ decoder evaluates directly in $O(1)$ time, whereas reference-generated lookup decoding also provides $O(1)$ runtime evaluation after an offline construction whose size scales as 2^n, with $n = (d-1)(r+1)$. The latter scaling motivates the single-round synthesis study below and limits direct extension of the lookup approach to larger multi-round circuits (Section 8.2).

5.3 Implementation

An FPGA target was chosen because the architecture is still being iterated; standard-cell synthesis is reported alongside it to bound the eventual integrated cost. The decoder was hand-written directly in Verilog and implemented through place-and-route on a Xilinx Artix-7 (xc7a35t-csg324-2). Both implementations were verified exhaustively against the reference decoder over their respective complete syndrome spaces, matching its output exactly on every syndrome. They were also evaluated on 40,000 Stim-generated syndrome samples drawn from four physical error rates spanning the range used for the threshold fits; all 40,000 decoded outputs matched the reference decoder's output exactly, for each implementation independently.

Place-and-route maps the decoder to six FPGA LUTs and six flip-flops, 0.03% of the device, with no DSP (digital signal processing) or block-RAM resources (Table 5). The decoder uses a negligible fraction of the FPGA resources. The standalone implementation does not close at a 2.5 ns (400 MHz) target, with a worst negative slack of −6.29 ns; the 4.023 ns register-to-pad data path is dominated by the I/O output buffer rather than decoder logic (register clock-to-output 0.379 ns, pad routing 1.258 ns, output buffer 2.386 ns). Retargeting to the achieved path delay gives a conservative cycle time of 8.79 ns (113.8 MHz) at the slow process corner, confirmed as the conservative choice against the fast corner (−2.169 ns vs. −6.290 ns); five independent placement

runs produced the same slack to three decimal places. This 8.79 ns figure is used throughout this work as the conservative decoder contribution to the system-level feedforward budget.

Beyond synthesis and place-and-route, the decoder was also implemented on a Digilent Basys3 board (Xilinx xc7a35tcpg236-1), a separate physical board from the xc7a35t-csg324-2 device targeted for the synthesis results above, for functional validation on physical hardware. It was driven with a separate set of 40,000 syndrome vectors, sampled at a single noise point ($p = 0.05$) under the paper's reference noise model rather than the four-error-rate sweep used for the software co-simulation above, streamed over a UART (universal asynchronous receiver-transmitter) link with the correction result and a hardware-observed response latency, at the board's 100 MHz clock resolution, returned for each vector. All 40,000 hardware responses matched an independent software reference decoder exactly, with zero mismatches and zero timeouts, and were observed within 0–1 clock cycles of the board's 100 MHz clock. Post-route static timing analysis on this implementation, which includes the UART interface logic alongside the decoder core and so is not directly comparable in resource count to the 6-LUT figure above, reports 5.261 ns of positive slack against the 10 ns clock constraint, implying a critical-path delay of 4.739 ns, below the 8.79 ns standalone estimate.

The same Basys3 hardware-validation methodology was extended to $d = 5$ and $d = 7$. Since a hand-derived closed form is impractical beyond $d = 3$ (Section 5), the $d = 5, 7$ decoders are a table-driven implementation of the same code-capacity graph and syndrome convention as Algorithm 2, verified exhaustively against the reference matching decoder over their complete syndrome spaces before hardware implementation; this is a separate decoder variant from the repeated-syndrome reference-generated lookup decoder of Section 5.2 used for the standard-cell scaling study below. Each was implemented and placed-and-routed on the same board and driven with its own set of 40,000 syndrome vectors, sampled at the same single noise point ($p = 0.05$) and streamed over the same UART link, matching the $d = 3$ protocol exactly. All 80,000 hardware responses (40,000 per distance) matched an independent software reference decoder exactly, with zero mismatches and zero timeouts at both distances, and were observed within 0–1 clock cycles of the board's 100 MHz clock at both distances, the same pattern reported for $d = 3$. Table 4 reports the full-implementation (decoder plus UART interface) resource utilisation and post-route timing for all three validated distances, with the same trend shown graphically in Fig. 6.

6 Simulation Methodology

We use Stim [26] to sample circuits and PyMatching [14] for MWPM decoding in the circuit-level simulations reported below, including the threshold, suppression, ablation, and photon-source analyses.

6.1 Noise Model

We represent photon loss by amplitude damping with transmissivity $\eta = 1 - p_{\text{loss}}$, where $p_{\text{loss}} \in [10^{-3}, 2\times10^{-2}]$. This probability range is motivated by reported integrated-photonic propagation-loss scales of 1–3 dB/cm; it is not a direct conversion

Table 4 Basys3 Hardware Validation Across Code Distances

Metric	$d = 3$	$d = 5$	$d = 7$
Test vectors	40,000	40,000	40,000
Mismatches / Timeouts	0 / 0	0 / 0	0 / 0
Measured latency (cycles)	0–1	0–1	0–1
Post-route critical path	4.739 ns	4.583 ns	4.633 ns
Clock	100 MHz	100 MHz	100 MHz
Full-impl. LUTs	71	78	90
Full-impl. registers	146	158	178
BRAM / DSP	0 / 0	0 / 0	0 / 0

Full-implementation figures include the decoder core plus UART interface logic together, and so are not directly comparable to the standalone decoder-core-only synthesis reported in Table 5 (6 LUTs, 6 registers at $d = 3$); the difference between the two is attributable to the UART receiver, transmitter, latency counter, and response-sequencing logic shared across all three distances.

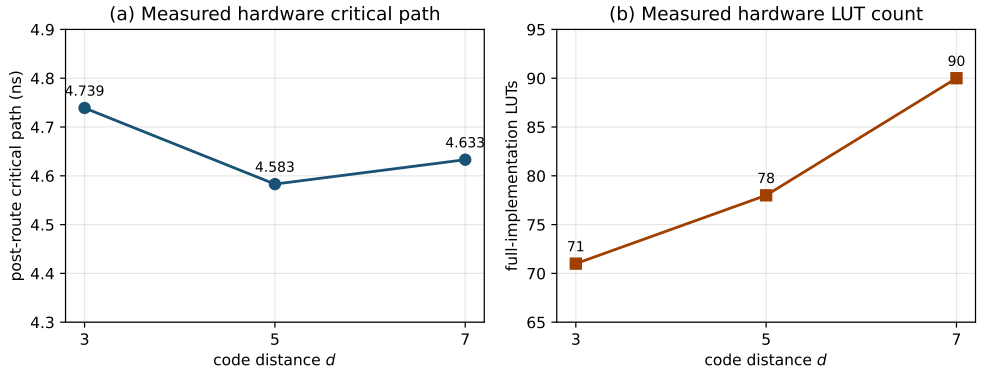


Fig. 6 Measured Basys3 hardware results across the three validated distances, both panels measured on physical hardware. (a) Post-route critical path, $d = 3, 5, 7$. (b) Full-implementation LUT count (decoder plus UART interface), $d = 3, 5, 7$, distinct from the standalone-decoder-core synthesis result reported separately (6 LUTs at $d = 3$, Table 5)

of those loss figures. In the model, gate errors arise from beamsplitter reflectivity deviations, parameterised by ϵ, and phase noise accumulated across the delay, consistent with silicon photonic process-design-kit specifications [15]. The specific on-chip loss used in simulation is 0.15 dB/cm (Table 6), an optimistic design target. This parameter and the broader 1–3 dB/cm literature range serve different purposes and should not be conflated.

The reference circuit-level model has two terms: data-qubit depolarizing noise of strength p before each round and a measurement-flip probability of $p/2$. The threshold, suppression-exponent, and finite-size-scaling results all use this model. It is a single-parameter abstraction of the physical error sources described above; loss, beamsplitter error, and phase noise are not added as simultaneous independent channels. The reported threshold is therefore a threshold for this Pauli model, not a prediction

Table 5 Decoder Implementation Results

Metric	Value
Slice LUTs	6
Slice registers	6
BRAM / DSP	0 / 0
Conservative standalone cycle time	8.79 ns (1 cyc., 113.8 MHz)
Standalone register-to-pad data path	4.023 ns
Device utilisation (LUT)	0.03%
Estimated on-chip power (vectorless)	0.081 W
Basys3 hardware validation (Section 5)	
Board	Digilent Basys3 (xc7a35tcpg236-1)
Test vectors	40,000 (Stim-generated)
Mismatches / Timeouts	0 / 0
Post-route critical path	4.739 ns (100 MHz clock)

of device-level performance. We use it as the reference because it captures data and measurement errors with one parameter and supports comparison with studies using the same convention. Section 7 examines the effect of adding a third, after-Clifford noise term. Table 7 maps each reported result to the model used to obtain it.

Table 6 Simulation Configuration Parameters

Parameter	Target Value	Tolerance Range
Propagation loss (on-chip)	0.15 dB/cm	±0.02 dB/cm
Beamsplitter error (ϵ)	0.01	0.001–0.027
Fiber coupling delay	100 ns	±5 ns
Phase noise (std. dev.)	1.5°	0.5°–3.0°

Under a depolarizing channel of total strength p, an X, Y, or Z error each occurs with probability $p/3$. In the computational basis used to define the bit-flip syndrome, both X and Y flip the logical bit value, while Z, a pure phase error in this basis, does not; the effective bit-flip probability is therefore $p_{\text{eff}} = P(X) + P(Y) = p/3 + p/3 = 2p/3$. Both nominal ($p$) and effective ($p_{\text{eff}}$) values are quoted throughout, since thresholds are not comparable across studies unless the convention is stated, and the factor of 2/3 is precisely what converts between them for this noise model.

6.2 Statistical Protocol

All logical error rates use 2×10^5 shots per point with an independent random seed per point, and are reported with 95% Wilson confidence intervals [38]. Points with fewer than 20 logical failures are excluded from fits. Suppression exponents are fitted

Table 7 Result-to-Model Correspondence

Result	Underlying model
Primary threshold ($p_{\text{th}} = 0.194$)	Two-term circuit-level Pauli model
Erasure decoder comparison	Separate heralded-erasure channel model
CZ gate fidelity	Beamsplitter fabrication-error model ($c = 7$–14)
HOM visibility bound ($V > 0.982$)	Visibility/gate-error relation composed with the primary Pauli model
Architecture-vs-generic equivalence	Primary circuit-level Pauli model
Basys3 hardware validation	Real hardware; syndrome vectors sampled from the primary Pauli model

on the asymptotic window $p \leq p_{\text{th}}/3$; uncertainties on the threshold and correlation-length exponent are obtained by binomial bootstrap resampling [7] with replacement, using 1000 resamples per point. Independent deterministic seeds are used for each point to avoid correlating the bootstrap noise across points. The supplementary shot-count-sensitivity and 3-term ablation checks described later use a lighter version of the same procedure, 30 resamples with a fixed seed, since their purpose is a consistency check rather than the primary uncertainty estimate. Every reported uncertainty in this protocol reflects finite-sample noise at 2×10^5 shots: the bootstrap carries that same shot noise through to the fitted threshold and exponent. Shot-count dependence is evaluated separately in the sensitivity analysis below.

An independent, narrower 8-point sweep over $d = 5, 7$ (Fig. 7) gives $p_{\text{th}} = 0.196 \pm 0.003$ (5×10^4 shots), 0.193 ± 0.001 (2×10^5), and 0.1929 ± 0.0006 (10^6), consistent with the main-text value (0.194 ± 0.001), with uncertainty shrinking as $1/\sqrt{\text{shots}}$; χ^2/ndof rises from 0.71 to 3.09 over the same range, consistent with the $d = 3$ corrections-to-scaling behaviour noted above.

The same fit-window check applied to the 3-term ablation (Section 7) was also run on the primary 2-term result: an independent 10^5-shot sweep (a separate run, not the production pipeline) over $d = 3, 5, 7$, $p \in [0.10, 0.28]$ gives $p_{\text{th}} = 0.201$ (residual 27, full range), 0.209 (residual 3.0, low-p), 0.199 (residual 1.4, high-p), and 0.200 (residual 1.7, narrow) across four fit windows. The resulting 0.199–0.209 spread (∼5% relative variation) is small against the 3-term ablation's 0.132–0.148 spread (∼11%), indicating modest fit-window sensitivity relative to that ablation.

7 Results

7.1 Threshold and Error Suppression

Importantly, the threshold and suppression results below characterise the extended circuit-level repetition-code model rather than the 33-mode single-round architectural module; the latter is evaluated for architectural consistency and single-error detection (Section 3), not for the threshold itself.

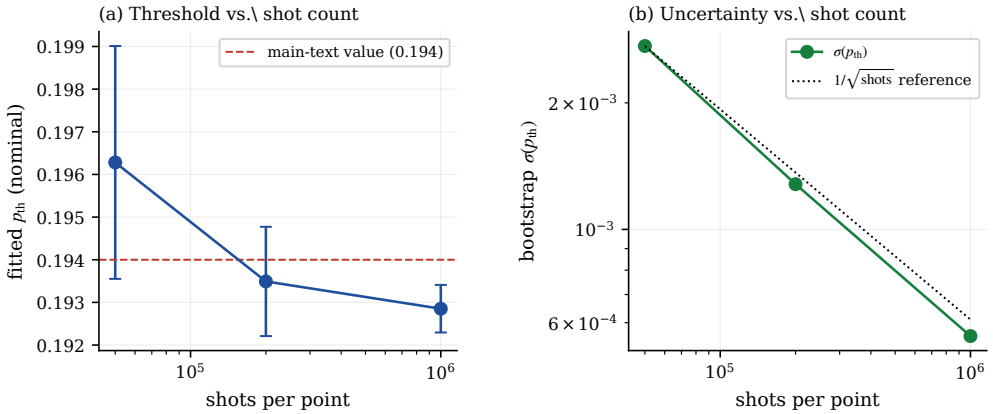


Fig. 7 Shot-count sensitivity of the finite-size-scaling threshold extraction, using an independent 8-point sweep over $d = 5, 7$ narrower in scope than the main campaign. (a) Fitted threshold approaches the main-text value as shot count increases. (b) Bootstrap uncertainty shrinks consistent with binomial sampling noise ($1/\sqrt{\text{shots}}$ reference)

Threshold characterisation follows the standard finite-size scaling procedure [37]. Logical error rates for $d = 3, 5, 7$ were fitted to the scaling form $p_L = f\big((p - p_{\mathrm{th}})\, d^{1/\nu}\big)$, yielding a reference-model threshold estimate, where the quoted uncertainty is statistical and conditional on the stated fit window and scaling ansatz (an independent fit-window sweep gives 0.199–0.209, Section 6.2), of

$$p_{\mathrm{th}} = 0.194 \pm 0.001 \text{ (finite-shot statistical uncertainty only)}, \quad p_{\mathrm{th}}^{\mathrm{eff}} = 0.129 \pm 0.001, \tag{3}$$

with $\chi^2/\mathrm{ndof} = 0.94$ for the collapse over $d \geq 5$ (Fig. 8). Given the simplified two-term noise model underlying this estimate (Section 6), p_{th} is best used as a reference value for comparing noise-model and circuit-design variants within this paper (Section 7.5) and against other threshold estimates obtained under comparable simplified models, rather than interpreted as a prediction of physical device performance. This is substantially higher than the equal-rate ($p = q$) fault-tolerant threshold of the toric code under independent bit-flip and measurement error, reported at $p_c \approx 0.03$ [39], although the code family and noise model differ (phenomenological rather than circuit-level), so the values are not directly comparable. The same fit returns a correlation-length exponent of $\nu = 1.56$; this value is quoted separately from Eq. (3) because a window-sensitivity check (below) found it varying between roughly 1.7 and 3.3 depending on the fit range, well beyond the fit's own quoted ± 0.05 statistical uncertainty. The threshold is stable across the stated fit, whereas ν varies substantially with the fit window and is therefore not used for extrapolation.

Including $d = 3$ increases ν to 1.64 and degrades the fit quality to 3.65, the expected signature of corrections to scaling at small distance; the extracted threshold is unchanged, and the $d \geq 5$ fit is therefore reported.

For reference, the Nishimori-point fixed point of the 2D $\pm J$ random-bond Ising model, the universality class expected for this kind of disordered syndrome-extraction problem, has been numerically determined at $\nu = 1.33 \pm 0.03$ [36]. The extracted ν

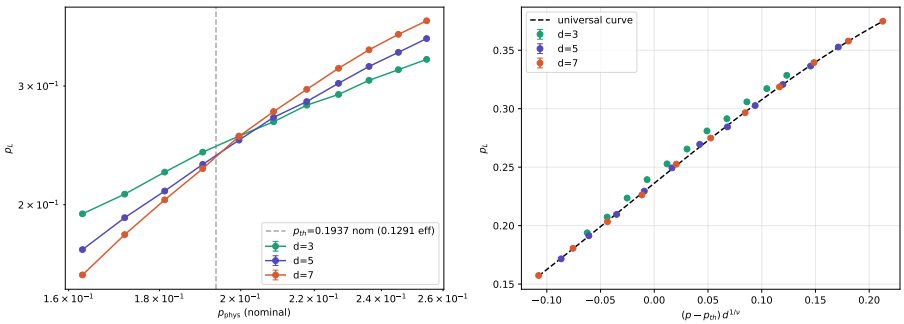


Fig. 8 (a) Logical error rate for $d = 3, 5, 7$ under the two-term circuit-level noise model ($2{\times}10^5$ shots per point). (b) Collapse onto the universal curve using the fitted p_{th} and ν of Eq. (3). The fit uses $d \geq 5$; $d = 3$ is shown for comparison but excluded from the fit

here differs from that value, and its strong fit-window dependence prevents a reliable universality-class comparison; ν is therefore not used to draw conclusions about extrapolation to larger distances.

The 33-mode detection architecture and the 55-mode $d = 3$ simulation configuration used for correction results (Section 2) remain distinct, as established earlier. Sub-threshold suppression follows $p_L \propto p^{\lceil d/2 \rceil}$, with fitted exponents 1.82 ± 0.01, 2.86 ± 0.02 and 3.98 ± 0.03 for $d = 3, 5, 7$.

7.2 Gate Fidelity and Beamsplitter Tolerance

Beamsplitter fabrication error degrades gate fidelity quadratically, $F \approx 1 - c\,\epsilon^2$. The coefficient c depends on which of the three beamsplitters carries the deviation: deviation confined to the central beamsplitter, where the two-photon interference defining the gate outcome occurs, gives $c \approx 7$–10, while coherent deviation of all three gives $c \approx 14$. The band shown below and used in downstream calculations brackets these two scenarios by their extremes, $c = 7$ to $c = 14$; the coefficient is therefore reported as a range, not a single value.

At the nominal operating point $\epsilon = 0.01$ the fidelity is at least 0.9986 even under the least favourable model, above the $F = 0.99$ target, and the target continues to be met out to $\epsilon \approx 0.027$ (Fig. 9). That value sets the tolerance quoted in Table 6.

7.3 Model-Derived Photon-Source Visibility Requirement

Photon indistinguishability, characterised by Hong–Ou–Mandel visibility [6] V, determines CZ gate fidelity. When two intended-identical photons interfere at a beamsplitter with visibility $V < 1$, a fraction $(1 - V)/2$ of attempted gate operations behaves as if the photons were distinguishable, contributing an error with probability one-half; the net gate error is therefore $p_{\text{gate}} \approx (1 - V)/2$. This is the relation used throughout this work for the post-selected CZ gate; an alternative fusion-based construction admits a milder $p_{\text{gate}} \approx (1 - V)/4$ dependence, shown alongside it in Fig. 10(b), since a fusion measurement can sometimes recover partial information even when the interfering photons are not perfectly indistinguishable. Fig. 10 shows the two analytic relations

that define the source-quality chain: panel (a) the visibility as a function of spectral mismatch, and panel (b) the resulting CZ gate error as a function of visibility.

The spectral-overlap conversion in panel (a) is analytic and independently verified: for equal-bandwidth Gaussian wavepackets, $V = \exp(-\Delta\omega^2/4\sigma^2)$. This closed form agrees with direct numerical integration of the mode-overlap integral to machine precision, and is self-consistent under inversion for any target V. The linear relation $p_{\text{gate}} \approx (1 - V)/2$ in panel (b) is a stated modelling choice rather than a simulation output, consistent with a post-selected CZ gate whose dominant error channel is central-beamsplitter distinguishability.

The critical visibility is obtained by directly composing the stated $p_{\text{gate}}(V) = (1 - V)/2$ relation with the paper's own reference two-term noise model: the resulting $p_{\text{gate}}(V)$ value is substituted directly for the parameter p in that model (before-round data depolarization p, measurement flip probability $p/2$, Section 6), the same substitution and the same two-term formula used for the main threshold result, evaluated on the d=3 repetition Monte Carlo at 2×10^5 shots per point. This places the $p_L = 10^{-3}$ crossing at

$$V > 0.982, \quad \text{equivalently} \quad \Delta\omega/\sigma < 0.270. \tag{4}$$

Including beamsplitter fabrication error shifts the bound to $V > 0.983$–0.985 for $c = 7$–14 (combined additively, $p_{\text{total}} = p_{\text{gate}}(V) + c\epsilon^2$), while the 3-term noise model used in

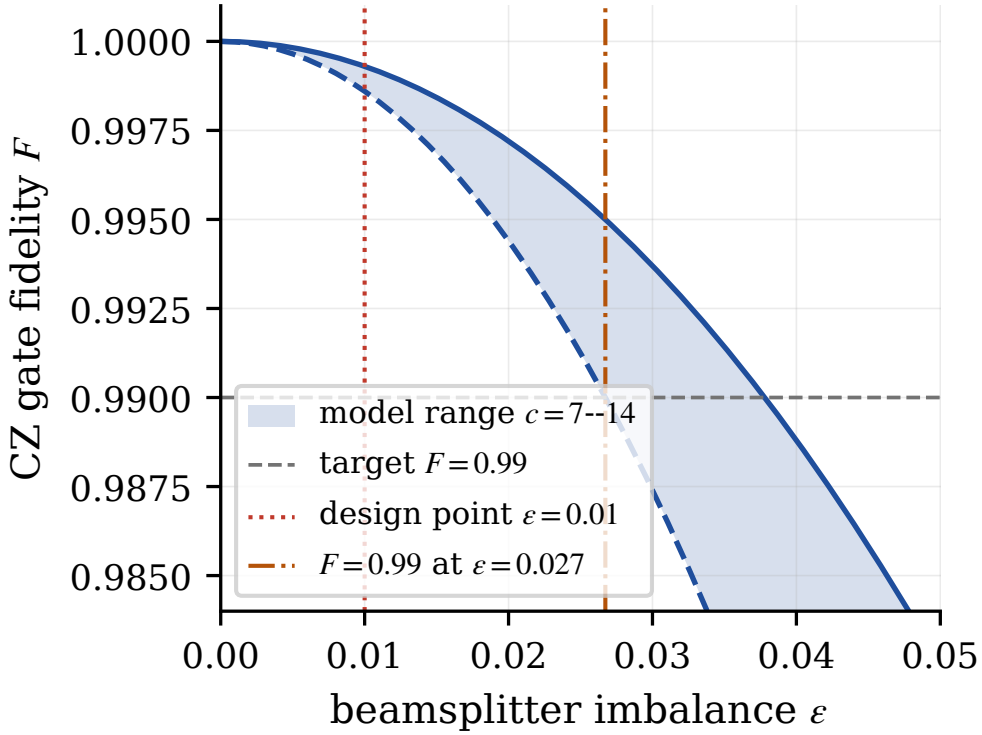


Fig. 9 Post-selected CZ gate fidelity versus beamsplitter imbalance ϵ, for the model range $c = 7$–14

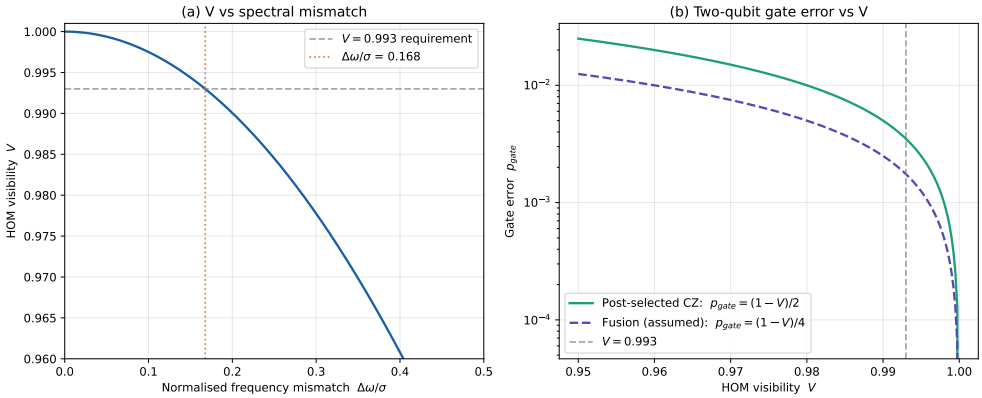


Fig. 10 (a) Hong–Ou–Mandel visibility versus normalised spectral mismatch. (b) Two-qubit gate error versus visibility, for the post-selected CZ and fusion gate models

the ablation study (Section 7.5) independently gives $V > 0.985$. The bound is model-dependent and is given as a point estimate without a bootstrap confidence interval, unlike the threshold and ν results of Section 7.1.

State-of-the-art integrated photonic sources report HOM visibilities approaching 0.995 [27], above the model-dependent $V > 0.982$ bound derived above; the constraint is demanding relative to typical reported values, achieved only by current best sources, and it identifies photon indistinguishability as an important near-term constraint on this architecture.

7.4 Basis Duality

Preparing in the conjugate basis yields the dual phase-flip memory. Its logical error rate was evaluated at $d = 3$ across the same physical-error-rate range as the bit-flip configuration, and the resulting curve overlays that of the generic repetition code within the point-wise statistical error at every sampled value of p (Fig. 11); no divergence between the two curves is observed across the swept range. This is a point-wise consistency check rather than an independent finite-size scaling fit, since a separate threshold and ν extraction for the X-memory configuration was not performed. The two circuits are related by the Hadamard conjugation described in Section 2 and are expected on that structural basis to share a threshold. That the same hardware supports either basis, but not both simultaneously, is precisely the gap a CSS code closes, and this gap can be quantified rather than only stated: an explicit Bacon-Shor $[[9, 1, 3]]$ subsystem code [13], built from the same dual-rail building blocks and confirmed to protect both bases simultaneously at the expected code distance, requires 21 physical qubits (9 data, 12 syndrome-measurement ancillas), compared with 5 (3 data, 2 ancillas) for the single-basis $d = 3$ repetition code, a factor of about 4.2× in qubit count. We do not convert this into a mode-count ratio: the mode budget depends on the allocation convention for routing, buffering and post-selected gates, and a like-for-like count would require a full mode-level design of the Bacon-Shor circuit, which we have

not carried out. The qubit count alone illustrates the connectivity and overhead cost of simultaneous X- and Z-error protection.

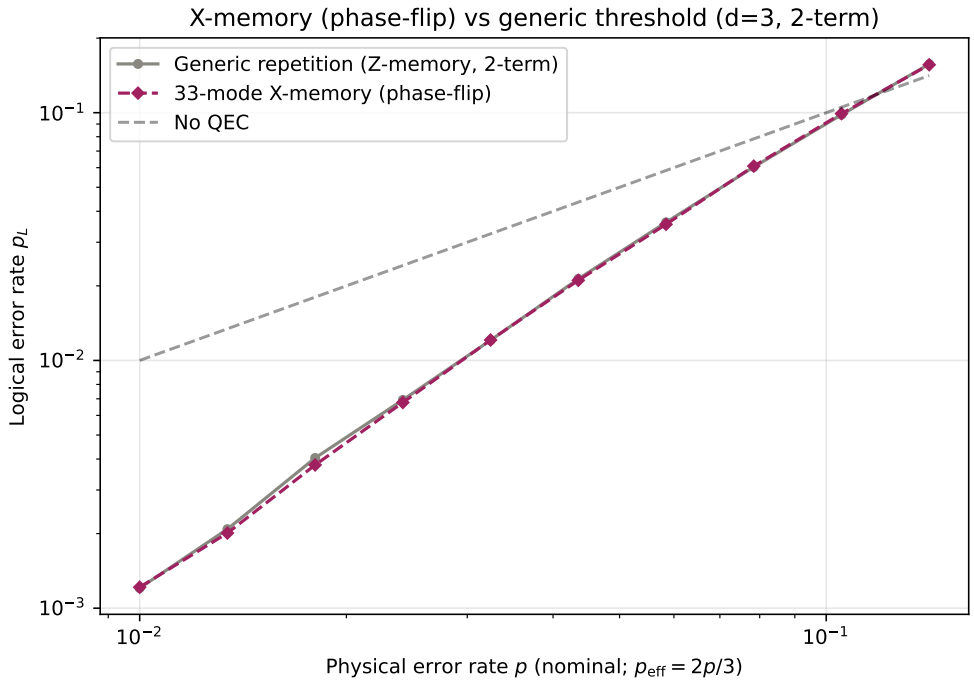


Fig. 11 Logical error rate versus physical error rate: phase-flip (X-memory) configuration overlaid on the generic repetition-code curve

The two configurations are mutually exclusive: the repetition-code implementation protects against one error type at a time.

7.5 Ablation of Design Choices

Table 8 isolates three design choices, varying one element while holding the remainder fixed. Including an after-Clifford depolarizing term reduces the apparent threshold substantially. The finite-size collapse ansatz used throughout this paper, $p_L = f((p - p_{\text{th}})d^{1/\nu})$, is expected to perform best near the critical point, with corrections to scaling increasing away from it. We therefore treat the near-threshold fit window as the primary estimate. The 3-term model gives $p_{\text{th}} = 0.136$ in the original campaign ($p \in [0.05, 0.14]$, residual 522, reduced $\approx$52) and $p_{\text{th}} = 0.148 \pm 0.002$ in an independent high-p re-evaluation ($p \in [0.16, 0.26]$, residual 34, reduced $\approx$3.4); the corresponding 2-term re-evaluation (0.193 ± 0.001) matches the main-text result, so the discrepancy is specific to the 3-term case.

Table 8 Ablation of Design Choices

Element	Config. A	Config. B	Effect
Noise model	3-term	2-term (ref.)	p_{th}: 0.148 adopted (23% reduction; ±0.002 is within-window statistical uncertainty only, systematic spread is 0.132–0.148); 0.136/0.132 retained, poorly-resolved fits
Erasure decoder	MWPM	Max. likelihood	2.74–2.86× lower p_L
Circuit model	Generic	Architecture-specific	Agreement within 2σ (worst point)

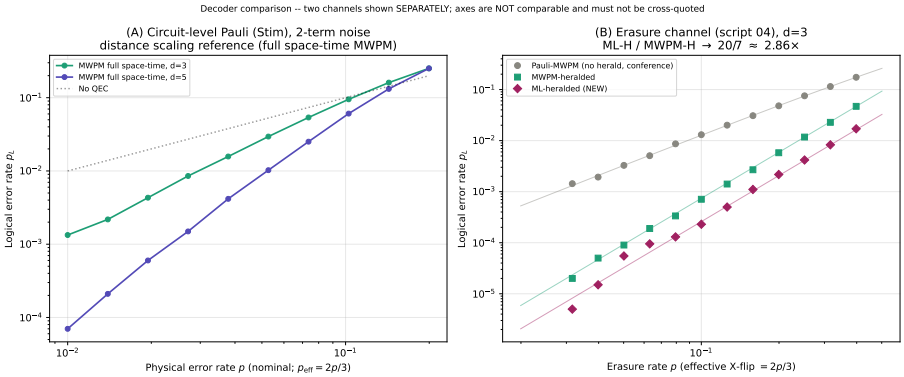


Fig. 12 Decoder comparison across two channels with non-comparable axes. (a) Circuit-level Pauli distance scaling. (b) Heralded erasure channel, maximum-likelihood decoding versus matching. Both retain $O(p^3)$ scaling; the Pauli baseline in (a) is $O(p^2)$

A unified sweep over the full combined range ($p \in [0.05, 0.26]$, 16 points) gives $p_{\text{th}} = 0.132$, with a reduced residual of ≈27, substantially poorer than the high-p window. The high-p window gives $p_{\text{th}} = 0.148 \pm 0.002$ (statistical fit uncertainty within this window only) and satisfies the stated collapse-quality criterion (23% reduction relative to the 2-term reference), so it is adopted as the 3-term estimate. Across fit windows (Table 8), the estimate ranges from 0.132 to 0.148, treated as systematic fit-window sensitivity, the dominant source of uncertainty on this estimate. For comparison, the equivalent fit-window sweep on the primary two-term model (Section 6.2) gives a substantially tighter 0.199–0.209 spread (∼5% relative variation, against ∼11% here), indicating the primary result is markedly more stable to fit-window choice than this ablation. On the heralded erasure channel, maximum-likelihood decoding lowers the logical error rate relative to matching by a factor approaching $20/7 \approx 2.86$ asymptotically, with 2.74 obtained at finite error rate (Fig. 12); both retain $O(p^3)$ scaling, so the gain is a constant factor rather than an improvement in suppression order. The architecture-specific circuit also reproduces the generic repetition-code behaviour within statistical error (Section 3).

8 Discussion

8.1 Code Family Trade-offs

Under the near-term linear-photonic constraints considered here, repetition codes offer advantages in connectivity, latency and implementation complexity, at the cost of protecting against a single error type, a trade-off that would not favour repetition codes in a setting where full fault tolerance, rather than near-term implementability, is the primary requirement. Repetition codes require degree-2 connectivity, mapping directly to linear waveguides without crossings, while surface-code implementations [20] generally impose higher local connectivity and routing demands, which can increase waveguide-crossing overhead and associated optical loss in a photonic realisation; QLDPC (quantum low-density parity-check) constructions [25] typically require higher connectivity still, which is precisely the trade-off that motivates linear architectures for near-term photonic implementations [12].

The latency consequence of this connectivity difference is visible in implemented decoder numbers, not only in the asymptotic scaling. Unlike the larger surface-code decoding graphs discussed in Section 2, the present $d = 3$ repetition decoder fits within a single FPGA clock cycle (8.79 ns, 113.8 MHz, post-place-and-route timing-derived) because its syndrome graph is small enough for the same lookup-table principle to resolve without the pipelined or clustering architectures larger graphs demand. The comparison is therefore one of implementation regime rather than absolute decoder performance; Table 10 below summarises the factual comparison across platforms.

8.2 Scaling Limits of the Constant-Time Decoder

Lookup decoding has $O(1)$ evaluation time once the table is constructed, but the table size grows as $2^{(d-1)(r+1)}$ with the number of rounds r. For single-round decoding, the table remains modest through $d = 9$, where 16 detectors require 65,536 entries; this is a size projection, not a synthesised implementation (only $d = 3, 5, 7$ were carried through standard-cell synthesis, Table 9). In the multi-round, fault-tolerant regime, the table size grows sharply: $d = 5$ requires 2^{24} entries and $d = 7$ requires 2^{48}. The $O(1)$ evaluation result therefore concerns runtime lookup after table construction. Table 9 and Fig. 13, by contrast, report only the tractable single-round synthesis that was actually run at $d = 3, 5, 7$: a separate, distinct regime from the fault-tolerant multi-round table above.

The single-round ROM decoder was synthesised to standard cells at $d = 3, 5, 7$ using a 45 nm open-cell library. This ROM-based decoder is a different implementation from the closed-form decoder reported in Table 5, which was synthesised to FPGA fabric (6 LUTs) rather than standard cells: the closed-form decoder's matching branch needs actual comparator logic, while a ROM lookup performs no computation at all. Table 9 and Fig. 13 report the ROM decoder's scaling: cell count grows substantially slower than the underlying ROM table size, indicating substantial logic minimisation for these three synthesized instances. These synthesis results characterise the single-round ROM implementation; the fault-tolerant multi-round decoder has the exponential table-size scaling given above. A third, independently-measured scaling

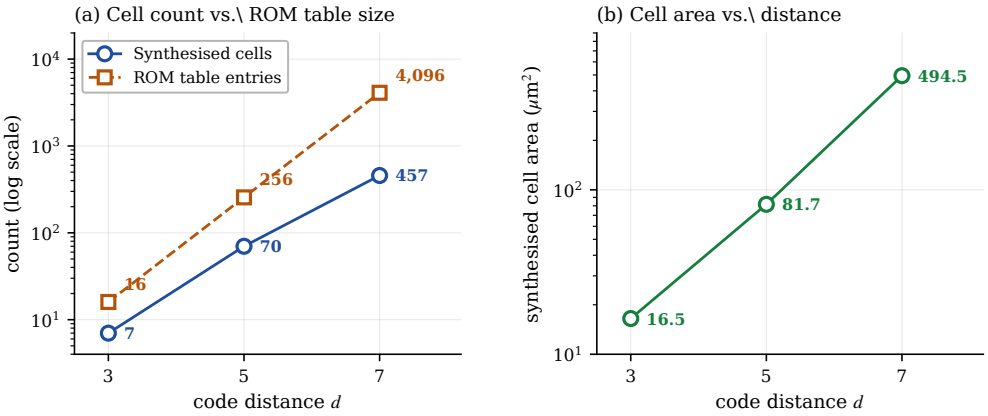


Fig. 13 (a) Standard-cell count and ROM table size versus code distance for the single-round decoder, both on a logarithmic scale. (b) Synthesised cell area versus distance

trend exists for the table-driven extension of the closed-form decoder itself, hardware-validated on FPGA at $d = 3, 5, 7$ (Table 4, Section 5); its LUT count and critical-path latency both grow modestly and near-monotonically over this range, consistent with the logic-minimisation pattern seen in the ROM decoder above, though the two target different fabrics (FPGA versus standard cells) and so are not directly comparable figure-for-figure.

Table 9 Single-Round ROM Decoder Synthesis vs. Code Distance

d	Detectors	ROM entries	Cells	Area (μm^2)
3	4	16	7	16.49
5	8	256	70	81.66
7	12	4,096	457	494.49

At sufficiently larger d or repeated-syndrome depth, where the exponential table size (Section 8.2) makes direct lookup increasingly impractical, a general matching or clustering engine is required, and the relevant prior work is the existing hardware decoder literature [16] rather than anything proposed here.

Two further bounds apply. Weight-two error patterns are mis-corrected, producing a logical failure. This is the expected $d = 3$ correction capacity, but it also means no protection remains once two data errors coincide. Increasing d improves bit-flip suppression without conferring any phase-flip protection, since every stabilizer measured here is invariant under a phase error.

8.3 Cross-Platform Context

Real-time decoding has been studied in far greater depth on superconducting hardware, where implemented decoders operate on surface-code syndrome graphs that

are orders of magnitude larger than the one considered here [16, 19]. Those results establish that classical decoding need not limit a feedback loop. Here, the deadline instead arises from the fibre delay line, and the $d = 3$ syndrome graph is small enough for single-cycle decoding. Direct quantitative comparison between the two settings is therefore inappropriate: surface and QLDPC codes correct both bit- and phase-flip errors, while the code considered here corrects one.

Table 10 compares the reported validation levels, decoder latency, and resource usage across the closest experimental and implementation precedents. Because the platforms and code families differ substantially in encoding, error coverage, and validation level, many cells are marked not applicable rather than populated with an estimated number: entries are reported only where a direct comparison is meaningful.

9 Limitations

Modelling and loss assumptions. The models used in this work are valid only within a bounded range. The two-term noise model (Section 6) is fitted and reported over $p \in [10^{-2.5}, 10^{-0.5}]$; extrapolation outside this range is not supported by the data. The gate-fidelity model (Section 7) is quoted as a coefficient range $c \approx 7$–14 rather than a single value because it depends on an unstated distribution of fabrication error across the three beamsplitters, and should be interpreted as a model range rather than a device-specific value. The finite-size scaling ansatz assumes the system is close enough to the critical point that leading-order scaling holds; the $d = 3$ deviation discussed in Section 7 is direct evidence that this assumption is already marginal at the smallest distance simulated, which is why the $d \geq 5$ fit is reported as the primary result rather than the full $d = 3, 5, 7$ fit. Photon loss is modelled through erasure conversion, with a heralded loss event mapped to a Pauli error of probability $p_{\text{loss}}/2$, accurate while loss events are independent and sparse; at higher loss rates the approximation degrades because correlations between successive layers become significant, and the mapping systematically underestimates the resulting logical error rate. The present study does not establish this crossover quantitatively; the modelled range $p_{\text{loss}} \in [10^{-3}, 2\times10^{-2}]$ (Section 6) sits at the lower end of loss rates reported for near-term integrated photonics, where sparsity is more plausible, and results near the upper end of this range should therefore be interpreted as model-dependent estimates.

Timing characterisation limitation. The 8.79 ns decoder figure anchoring the feedforward-margin argument (Eq. 2) was checked for placement stability (five placement runs, identical -6.290 ns slack) and corner direction (fast corner -2.169 ns, confirming slow corner is conservative), both in Section 5, but remains a single-tool-run characterisation: no formal voltage/temperature derating, and no synchroniser on reset deassertion (clock-domain-crossing analysis is not required internally, since the decoder is designed as a single clock domain). Positive slack was consistent across three independent post-route hardware runs at $d = 3, 5, 7$ (Table 4), though formal derating and cross-validation against an independent timing tool remain future work. This treatment does not extend across the SNSPD-to-FPGA boundary: the detector array and the FPGA are described throughout this work as separate physical subsystems, and the detector outputs are assumed to be synchronised to the FPGA clock

Table 10 Cross-Platform Positioning

Work	Encoding	Code / distance	Decoder latency	Decoder resources	Validation level
This work	DV dual-rail (single photon)	Repetition, d=3, 5, 7	8.79 ns (1 cycle, 113.8 MHz, conservative post-P&R standalone cycle time, d=3)	6 LUTs (FPGA, d=3 standalone)	Simulation + FPGA P&R + hardware validation (d=3, 5, 7)
LILLIPUT [18]	N/A (classical decoder, code-agnostic)	Surface, d=3–5	29–42 ns	<7% FPGA logic	FPGA, measured
Riverlane [16]	N/A (superconducting)	Surface, d=23	∼1 μs (FPGA); 240 ns (ASIC, 1057 qubits)	0.06 mm^2, 8 mW (ASIC)	FPGA/ASIC, measured
Aurora [17]	CV (squeezed-light cluster state)	Foliated repetition, d=2	Single clock cycle (feedforward)	N/A (not reported at component level)	Physical, networked hardware
FBQC [12]	DV fusion-based	Architecture proposal, not code-specific	N/A	N/A	Theoretical proposal

before entering the decoder, an assumption that has not been independently verified. A full worst-case timing sign-off was not performed.

Structural limitation. The repetition code protects against bit-flip errors only; increasing the distance does not provide simultaneous phase-flip protection, since every stabilizer measured here is invariant under a phase error by construction. It remains an appropriate starting point for a near-term linear-photonic demonstration, but full fault-tolerant operation would therefore require a code construction capable of protecting against both error types simultaneously; a CSS-type construction is one natural way to achieve this.

Simulation-vs-hardware limitation. The d=3 correction results come from the extended simulation rather than the 33-mode architecture, which provides single-error detection only. The decoder was validated on FPGA hardware at $d = 3, 5, 7$ (Section 5), but no photonic hardware was fabricated or measured. In the feedforward budget, only the decoder latency was timing-derived; the 50 ns SNSPD and 10 ns Pockels-cell values are assumed or literature-informed.

Scale-generalisation limitation. The eleven-layer pipeline (L01–L11) is architecturally generic, but the photonic feedforward margin (Section 4) has only been evaluated for the $d = 3$ architecture. Syndrome extraction at higher d involves proportionally more ancilla and more per-round operations within the same L03/L04 layer slots, and the computed margin is not a large fraction of the total budget; at $d = 5, 7$, this additional optical-side work may require a longer buffer or a revised timing budget, a question that requires photonic hardware not available here. The classical decoder component of this limitation is narrower than it was, given the decoder-level hardware validation now spanning all three distances (see the simulation-vs-hardware limitation above): the $d = 5, 7$ decoders' post-route critical paths, 4.583 ns and 4.633 ns (Table 4), are both below the 8.79 ns standalone timing this work's overall budget already accommodates for the decoder at $d = 3$. The $d = 5, 7$ logical-error-rate results reported in Section 7 therefore establish circuit-level QEC behaviour, but physical feedforward closure of the full 99.8 ns optical budget at those distances remains unverified, since it requires photonic hardware this work does not include.

10 Reproducibility

All simulation, synthesis and implementation artefacts are publicly available at https://github.com/aarti-singla/dual-rail-qec-fpga. The repository contains the Stim circuit constructions and PyMatching decoding scripts for every reported figure; the RTL decoder sources and self-checking testbenches for $d = 3, 5, 7$; the lookup-table generator (Algorithm 3); the Vivado implementation script and the standard-cell synthesis script; the Basys3 hardware validation sources for all three validated distances (UART interface RTL, board constraints, and the host-side streaming and verification script); and the noise-model conventions used throughout.

Software versions: Stim 1.16, PyMatching 2.4, Vivado 2025.2, Yosys 0.33 with the Nangate 45 nm open cell library. Every reported logical error rate uses 2×10^5 shots; the scripts fix the random seeds, and the software versions above are those used to produce the reported curves.

11 Conclusion

Under the assumed timing model, detector latency—not decoder latency—is the largest contribution to feedforward operation in this small-distance dual-rail photonic memory. The decoder occupies about 9% of the 99.8-ns budget, making detector readout the main target for system-level latency reduction.

The work combines a mode-resolved 33-mode dual-rail architecture with an implemented feedforward decoder. Under the reference two-term model, finite-size scaling gives $p_{\mathrm{th}} = 0.194 \pm 0.001$ (statistical uncertainty only; effective value 0.129), and the suppression exponents for $d = 3, 5, 7$ are consistent with $\lceil d/2 \rceil$. Applying the stated source-quality relation to the same noise model gives the model-dependent bound $V > 0.982$ (Eq. 4). The standalone $d = 3$ decoder uses six LUTs and has a conservative post-place-and-route cycle time of 8.79 ns (113.8 MHz). Hardware validation on the Basys3 board covered $d = 3, 5, 7$ and 120,000 syndrome vectors in total, with no mismatches.

The main contribution is a reference design that links the 33-mode detection architecture, the extended 55-mode $d = 3$ correction model, and the FPGA decoder through a shared specification of modes, noise, and timing. The present results establish decoder functionality and estimate its contribution to the feedforward budget; they do not demonstrate a fabricated photonic device or a closed-loop photonic experiment. Future work includes using heralded loss information directly in the decoder and extending the architecture to a CSS code, which would protect against both bit- and phase-flip errors but impose a larger resource and timing burden.

Acknowledgments. The authors gratefully acknowledge the support of the C2S programme under MeitY, Government of India, for computational resources.

Statements and Declarations

Funding: This work was supported in part by the C2S programme under MeitY, Government of India.
Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.
Ethics approval and consent to participate: Not applicable.
Consent for publication: Not applicable.
Data availability: All simulation, synthesis, and implementation artefacts (Stim circuit constructions and PyMatching decoding scripts for every reported figure; RTL decoder sources and self-checking testbenches; the lookup-table generator; the Vivado implementation script and the standard-cell synthesis script; the Basys3 hardware validation sources; and the noise-model conventions used throughout) are publicly available at https://github.com/aarti-singla/dual-rail-qec-fpga.
Materials availability: Not applicable.
Code availability: See Data availability above.
Author contributions: Aarti Singla and Angel Garg contributed to the conceptualisation, design and development of the work, including the architecture, simulations, FPGA implementation and hardware validation. Aarti Singla prepared the original

draft of the manuscript. Anil Singh and Manu Bansal reviewed and verified the work and the manuscript. Vibhu Srivastava and Sunny Sharma critically reviewed the manuscript and contributed to its revision. All authors read and approved the final manuscript.

References

[1] Kok, P., Munro, W.J., Nemoto, K., Ralph, T.C., Dowling, J.P., Milburn, G.J.: Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.* 79, 135–174 (2007). https://doi.org/10.1103/RevModPhys.79.135.

[2] O'Brien, J.L., Furusawa, A., Vučković, J.: Photonic quantum technologies. *Nat. Photon.* 3, 687–695 (2009). https://doi.org/10.1038/nphoton.2009.229.

[3] Knill, E., Laflamme, R., Milburn, G.J.: A scheme for efficient quantum computation with linear optics. *Nature* 409, 46–52 (2001). https://doi.org/10.1038/35051009.

[4] Reck, M., Zeilinger, A., Bernstein, H.J., Bertani, P.: Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* 73, 58–61 (1994). https://doi.org/10.1103/PhysRevLett.73.58.

[5] Clements, W.R., Humphreys, P.C., Metcalf, B.J., Kolthammer, W.S., Walmsley, I.A.: Optimal design for universal multiport interferometers. *Optica* 3, 1460–1465 (2016). https://doi.org/10.1364/OPTICA.3.001460.

[6] Hong, C.K., Ou, Z.Y., Mandel, L.: Measurement of subpicosecond time intervals between two photons by interference. *Phys. Rev. Lett.* 59, 2044–2046 (1987). https://doi.org/10.1103/PhysRevLett.59.2044.

[7] Efron, B.: Bootstrap methods: another look at the jackknife. *Ann. Stat.* 7, 1–26 (1979). https://doi.org/10.1214/aos/1176344552.

[8] Grassl, M., Beth, T., Pellizzari, T.: Codes for the quantum erasure channel. *Phys. Rev. A* 56, 33–38 (1997). https://doi.org/10.1103/PhysRevA.56.33.

[9] Ralph, T.C., White, A.G., Munro, W.J., Milburn, G.J.: Simple scheme for efficient linear optics quantum gates. *Phys. Rev. A* 65, 012314 (2001). https://doi.org/10.1103/PhysRevA.65.012314.

[10] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10th Anniversary edn. Cambridge University Press, Cambridge (2010).

[11] Bourassa, J.E. et al.: Blueprint for a scalable photonic fault-tolerant quantum computer. *Quantum* 5, 392 (2021). https://doi.org/10.22331/q-2021-02-04-392.

[12] Bartolucci, S. et al.: Fusion-based quantum computation. *Nat. Commun.* 14, 912 (2023). https://doi.org/10.1038/s41467-023-36493-1.

[13] Bacon, D.: Operator quantum error-correcting subsystems for self-correcting quantum memories. *Phys. Rev. A* 73, 012340 (2006). https://doi.org/10.1103/PhysRevA.73.012340.

[14] Higgott, O.: PyMatching: A Python package for decoding quantum codes with minimum-weight perfect matching. *ACM Trans. Quantum Comput.* 3(3) (2022). https://doi.org/10.1145/3505637.

[15] Giewont, K. et al.: 300-mm monolithic silicon photonics foundry technology. *IEEE J. Sel. Top. Quantum Electron.* 25(5), 1–11, Art. no. 8200611 (2019). https://doi.org/10.1109/JSTQE.2019.2908790.

[16] Barber, B. et al.: A real-time, scalable, fast and resource-efficient decoder for a quantum computer. *Nat. Electron.* 8, 84–91 (2025). https://doi.org/10.1038/s41928-024-01319-5.

[17] Aghaee Rad, H., Ainsworth, T., Alexander, R.N. et al.: Scaling and networking a modular photonic quantum computer. *Nature* 638, 912–919 (2025). https://doi.org/10.1038/s41586-024-08406-9.

[18] Das, P., Locharla, A., Jones, C.: LILLIPUT: a lightweight low-latency lookup-table decoder for near-term quantum error correction. In: Proc. 27th ACM Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 541–553 (2022). https://doi.org/10.1145/3503222.3507707.

[19] Google Quantum AI: Suppressing quantum errors by scaling a surface code logical qubit. *Nature* 614, 676–681 (2023). https://doi.org/10.1038/s41586-022-05434-1.

[20] Fowler, A.G., Mariantoni, M., Martinis, J.M., Cleland, A.N.: Surface codes: towards practical large-scale quantum computation. *Phys. Rev. A* 86, 032324 (2012). https://doi.org/10.1103/PhysRevA.86.032324.

[21] Dennis, E., Kitaev, A., Landahl, A., Preskill, J.: Topological quantum memory. *J. Math. Phys.* 43, 4452–4505 (2002). https://doi.org/10.1063/1.1499754.

[22] Gottesman, D.: Stabilizer codes and quantum error correction. Ph.D. thesis, California Institute of Technology (1997). https://doi.org/10.7907/rzr7-dt72.

[23] Calderbank, A.R., Shor, P.W.: Good quantum error-correcting codes exist. *Phys. Rev. A* 54, 1098–1105 (1996). https://doi.org/10.1103/PhysRevA.54.1098.

[24] Steane, A.M.: Multiple-particle interference and quantum error correction. *Proc. R. Soc. Lond. A* 452, 2551–2577 (1996). https://doi.org/10.1098/rspa.1996.0136.

[25] Breuckmann, N.P., Eberhardt, J.N.: Quantum low-density parity-check codes. *PRX Quantum* 2, 040101 (2021). https://doi.org/10.1103/PRXQuantum.2.040101.

[26] Gidney, C.: Stim: A fast stabilizer circuit simulator. *Quantum* 5, 497 (2021). https://doi.org/10.22331/q-2021-07-06-497.

[27] PsiQuantum Team: A manufacturable platform for photonic quantum computing. *Nature* 641(8064), 876–883 (2025). https://doi.org/10.1038/s41586-025-08820-7.

[28] Delfosse, N., Zémor, G.: Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Phys. Rev. Research* 2, 033042 (2020). https://doi.org/10.1103/PhysRevResearch.2.033042.

[29] Wu, Y., Kolkowitz, S., Puri, S., Thompson, J.D.: Erasure conversion for fault-tolerant quantum computing in alkaline earth Rydberg atom arrays. *Nat. Commun.* 13, 4657 (2022). https://doi.org/10.1038/s41467-022-32094-6.

[30] Pittman, T.B., Jacobs, B.C., Franson, J.D.: Demonstration of quantum error correction using linear optics. *Phys. Rev. A* 71, 052332 (2005). https://doi.org/10.1103/PhysRevA.71.052332.

[31] Pittman, T.B., Jacobs, B.C., Franson, J.D.: Demonstration of nondeterministic quantum logic operations using linear optical elements. *Phys. Rev. Lett.* 88, 257902 (2002). https://doi.org/10.1103/PhysRevLett.88.257902.

[32] Pittman, T.B., Jacobs, B.C., Franson, J.D.: Heralding single photons from pulsed parametric down-conversion. *Opt. Commun.* 246, 545–550 (2005). https://doi.org/10.1016/j.optcom.2004.11.027.

[33] U'Ren, A.B., Silberhorn, C., Banaszek, K., Walmsley, I.A., Erdmann, R., Grice, W.P., Raymer, M.G.: Generation of pure-state single-photon wavepackets by conditional preparation based on spontaneous parametric downconversion. *Laser Phys.* 15, 146–161 (2005). https://arxiv.org/abs/quant-ph/0611019.

[34] Thiele, F., Lamberty, N., Hummel, T., Lange, N.A., Procopio, L.M., Barua, A., Lengeling, S., Quiring, V., Eigner, C., Silberhorn, C., Bartley, T.J.: Cryogenic feedforward of a photonic quantum state. *Optica* 12, 720 (2025). https://doi.org/10.1364/OPTICA.551287.

[35] Wu, Y., Liu, W., Chen, J., Sun, X., Zhang, T.: A novel PWM signal-generation strategy for Pockels cell drivers. *Photonics* 10, 866 (2023). https://doi.org/10.3390/photonics10080866.

[36] Honecker, A., Picco, M., Pujol, P.: Universality class of the Nishimori point in the 2D $\pm J$ random-bond Ising model. *Phys. Rev. Lett.* 87, 047201 (2001). https://doi.org/10.1103/PhysRevLett.87.047201.

[37] Wang, C., Harrington, J., Preskill, J.: Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Ann. Phys.* 303, 31–58 (2003). https://doi.org/10.1016/S0003-4916(02)00019-2.

[38] Wilson, E.B.: Probable inference, the law of succession, and statistical inference. *J. Am. Stat. Assoc.* 22, 209–212 (1927). https://doi.org/10.1080/01621459.1927.10502953.

[39] Andrist, R.S., Katzgraber, H.G., Bombin, H., Martin-Delgado, M.A.: Error tolerance of topological codes with independent bit-flip and measurement errors. *Phys. Rev. A* 94, 012318 (2016). https://doi.org/10.1103/PhysRevA.94.012318.