A Real-Time Feed-Forward System for Heralded Quantum Entanglement

Zhizhen Qin, Zhongtao Shen, Ruoyu Yan, Zhaibin Cui, Junpeng She, Hongxuan Zhang, Kaicheng Jin, Zichao Zhou, Changqing Feng, Shubin Liu and Binxiang Qi

***Abstract*—Quantum networks rely on the ability to generate and manipulate entanglement between remote nodes with high precision and low latency. In this work, we present an enhancement to an existing laboratory quantum-computing real-time control platform by integrating a time-to-digital converter (TDC) synchronized with the control and measurement sequence. The system performs real-time timestamp acquisition, flexible phase calculation, and deterministic feedback, enabling high-precision time measurements and phase feedforward. Our implementation achieves a TDC root-mean-square (RMS) timing resolution of 11.5 ps, providing sufficient accuracy for heralded Bell-state generation in ion–photon entanglement experiments. This platform allows flexible, low-latency phase compensation, demonstrating the capability for real-time control of stochastic quantum events with sub-nanosecond precision.**

***Index Terms*—Field-programmable gate arrays(FPGA), High-Precision Timing, Real-Time Measurement,Quantum Networks**

I. Introduction

QUANTUM networks provide a promising architecture for distributing quantum information among spatially separated nodes. By combining stationary matter qubits with flying photonic qubits, quantum networks are expected to support long-distance quantum communication, distributed quantum computing, quantum teleportation, entanglement-enhanced sensing, and modular quantum information processing [1]–[3]. In this context, entanglement distribution between remote nodes is a fundamental primitive, since many quantum-network protocols rely on the generation, storage, and manipulation of nonlocal quantum correlations [4].

Among different physical platforms, trapped ions, neutral atoms, and solid-state emitters have been widely investigated as quantum-network nodes [5]–[7]. A common strategy for remote entanglement generation is the *heralded* scheme, in which photons emitted from different quantum nodes interfere at an intermediate station, and a successful photon detection event projects the remote matter qubits into an entangled state [8]–[10]. Recent experiments on multi-emitter quantum-network nodes have further demonstrated that multiplexed entanglement distribution and multipartite entanglement generation can enhance network scalability [11].

This work was supported by Quantum Science and Technology-National Science and Technology Major Project (2021ZD0301601).(Corresponding author: Zhongtao Shen.)

Zhizhen Qin, Zhongtao Shen, Kaicheng Jin, Changqing Feng and Shubin Liu are with Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China (e-mail: henzt@ustc.edu.cn).

Ruoyu Yan, Zhaibin Cui, Junpeng She, Hongxuan Zhang, Zichao Zhou and Binxiang Qi are with Center for Quantum Information, IIIS, Tsinghua University, Beijing 100084, China

Zichao Zhou is with Hefei National Laboratory, Hefei 230088, China

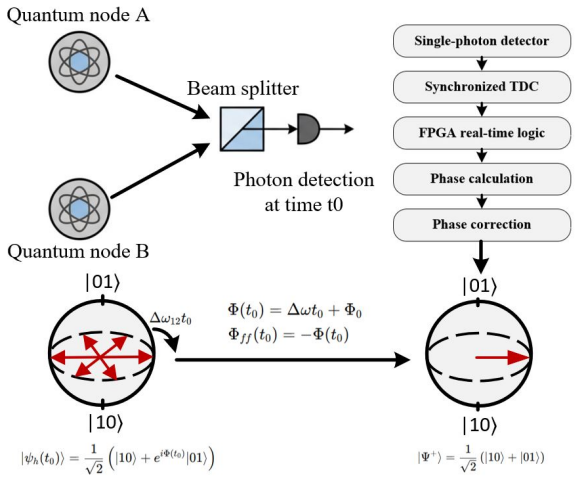


Fig. 1. Overview of the proposed real-time feedforward system for heralded quantum entanglement. A photon detection event heralds remote entanglement and provides a timestamp for phase correction. FPGA-based electronics acquire the timestamp using a synchronized TDC and generate the corresponding feedforward signal. Real-time phase compensation converts a stochastic entangled state into a deterministic Bell state.

Figure 1 illustrates the principle of the proposed real-time feedforward system [12]–[15]. A photon detection event heralds remote entanglement and simultaneously provides timing information. The detected timestamp is processed by FPGA-based electronics to generate the phase correction required for Bell-state stabilization.

In heralded matter–matter entanglement experiments, the photon detection event is not only a success flag, but also carries timing information that determines the phase of the generated entangled state. For two remote emitters labeled A and B, a single-photon detection event erases the which-emitter information and projects the two matter qubits into

$$|\psi_h(t_0)\rangle = \frac{1}{\sqrt{2}}\left(|10\rangle + e^{i\Phi(t_0)}|01\rangle\right), \tag{1}$$

where t_0 is the photon detection time. For frequency-nondegenerate optical transitions, the relative phase can be written as

$$\Phi(t_0) = \Delta\omega t_0 + \Phi_0, \tag{2}$$

where $\Delta\omega$ is the effective angular frequency difference and Φ_0 is a fixed phase offset determined by the optical, microwave, and detection paths [8], [11].

The target Bell state can be chosen as

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}\left(|10\rangle + |01\rangle\right). \tag{3}$$

However, because t_0 is stochastic, $\Phi(t_0)$ varies from shot to shot. Without compensation, the ensemble-averaged density matrix is

$$\rho = \int P(t_0)|\psi_h(t_0)\rangle\langle\psi_h(t_0)|dt_0, \tag{4}$$

and its off-diagonal coherence term is proportional to

$$\rho_{10,01} = \frac{1}{2}\left\langle e^{-i\Phi(t_0)}\right\rangle. \tag{5}$$

A broad phase distribution suppresses the coherence and degrades the heralded state into an incoherent mixture. Therefore, to obtain a deterministic Bell-state phase, a real-time feedforward phase

$$\Phi_{\mathrm{ff}}(t_0) = -\Phi(t_0) = -\Delta\omega t_0 - \Phi_0 \tag{6}$$

must be applied after each successful detection event.

The timing accuracy of the photon detection event directly determines the phase accuracy of the feedforward operation. If the timing uncertainty of the time-to-digital converter (TDC) is δt, the resulting phase error is

$$\delta\Phi = \Delta\omega\delta t. \tag{7}$$

For small phase errors, the Bell-state fidelity after feedforward is approximately

$$F = |\langle\Psi^+|\psi_{\mathrm{out}}\rangle|^2 \approx 1 - \frac{(\Delta\omega)^2\delta t^2}{4}. \tag{8}$$

For the planned heralded entanglement experiment between a $^{137}\mathrm{Ba}^+$ ion and a $^{87}\mathrm{Rb}$ atom in a hybrid quantum repeater system, the effective frequency difference between the two ground-state hyperfine splittings is approximately

$$\Delta f = f_{\mathrm{Ba}} - f_{\mathrm{Rb}} \approx 8.0377 - 6.8347 = 1.203\ \mathrm{GHz}. \tag{9}$$

Using $\Delta\omega = 2\pi\Delta f$, the timing requirement for achieving $F \geq 0.95$ is

$$\delta t \leq \frac{\sqrt{0.2}}{2\pi\Delta f} \approx 59.2\ \mathrm{ps}. \tag{10}$$

This requirement indicates that the TDC must provide picosecond-level timing precision to support high-fidelity phase feedforward in the target heterogeneous quantum-node experiment.

In addition to timing resolution, feedback latency is also a critical parameter. After a photon detection event, the control system must complete event discrimination, timestamp acquisition, phase calculation, and feedback output before the subsequent quantum operation. The total latency can be expressed as

$$T_{\mathrm{lat}} = T_{\mathrm{det}} + T_{\mathrm{TDC}} + T_{\mathrm{FPGA}} + T_{\mathrm{out}} + T_{\mathrm{act}}. \tag{11}$$

For successful real-time feedforward, this latency should be much shorter than the qubit coherence time,

$$T_{\mathrm{lat}} \ll T_2. \tag{12}$$

For neutral-atom and trapped-ion qubits, the coherence time can reach the millisecond scale or longer [16], [17]. Taking a conservative estimate of

$$T_2 \approx 1\ \mathrm{ms}, \tag{13}$$

the feedforward system should therefore operate on a microsecond or sub-microsecond time scale. This motivates a deterministic hardware feedback path rather than a conventional host-CPU feedback loop.

For phase feedforward, deterministic timing is as important as the average feedback latency. A conventional host-CPU feedback loop may suffer from nondeterministic delays caused by operating-system scheduling, interrupt handling, and data-transfer jitter. Such latency fluctuations introduce an additional phase error,

$$\delta\Phi_{\mathrm{jitter}} = \Delta\omega\delta T_{\mathrm{lat}}, \tag{14}$$

which can destroy the fixed phase relationship required for Bell-state synthesis. Therefore, an FPGA-based real-time architecture is preferred for phase feedforward, as it provides deterministic pipeline timing and low-latency response.

To meet these requirements, this work extends an existing laboratory quantum-computing real-time control platform by integrating a synchronized TDC module and flexible phase-feedforward logic [18]–[24]. The TDC records photon detection times while remaining synchronized with the control and measurement sequence. The FPGA performs event validation, timing-window discrimination, phase calculation, and deterministic feedback output. The measured timing resolution and latency performance of the implemented system are presented in Section III.

II. System Architecture

The requirements derived in Section I indicate that a practical feedforward system for heralded entanglement generation must simultaneously provide picosecond-level timestamp acquisition, deterministic low-latency processing, and tight synchronization with the experimental sequence. To satisfy these requirements, we developed a real-time feedforward platform based on a synchronized FPGA controller and an integrated TDC. The overall architecture and operating workflow are described in the following subsections.

A. Overall System Architecture

The proposed real-time feedforward system is implemented on an existing FPGA-based quantum-control platform and consists of a host computer, a Trigger and Clock Module (TCM), TTL modules [25]–[29], and DDS/AWG modules [30]–[35], as illustrated in Fig. 2.

The host computer is responsible for non-real-time tasks, including experimental-sequence compilation, parameter configuration, and data management. Experimental programs are translated into hardware-executable instructions and distributed to the corresponding FPGA modules before execution.

The TCM serves as the central timing and synchronization unit of the system. It distributes the global clock and trigger signals to all modules, ensuring deterministic timing alignment

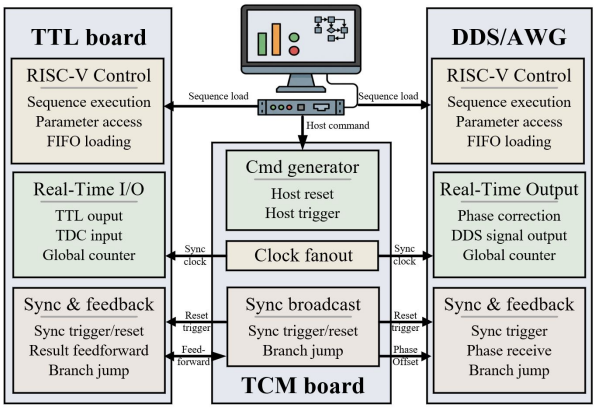


Fig. 2. Architecture of the proposed real-time feedforward system. The host computer configures and controls the experiment through the FPGA-based control platform. The TCM distributes the global clock and trigger signals, while the TTL module performs timestamp acquisition and phase-compensation calculation. The calculated correction is broadcast by the TCM to the DDS module, which updates the corresponding output phase for subsequent quantum operations.

throughout the experiment. The DDS/AWG modules generate the analog control waveforms required for quantum-state preparation and manipulation, while the TTL modules provide digital input/output functions and photon-detection interfaces.

For heralded-entanglement experiments, the TTL module performs the real-time feedforward operation. Photon-detection events from the single-photon detector (SPD) are received by the TTL module, where a synchronized TDC measures the photon arrival timestamp. The timestamp is then processed by the local feedforward logic to calculate the phase compensation associated with the heralding event. The resulting phase-correction information is transmitted to the TCM, which broadcasts the updated phase parameters to the corresponding DDS module. Upon receiving the feedforward information, the target DDS channel updates its output phase and generates the corrected control waveform for the subsequent quantum operation.

By integrating timestamp acquisition, phase calculation, synchronization, and waveform update within the FPGA-based control system, deterministic low-latency feedforward can be achieved without involving the host computer during experiment execution.

B. Real-Time Feedforward Workflow

The timing relationship and conditional operations of the real-time feedforward system are illustrated in Fig. 3, organized in three swimlanes: TCM, TTL, and DDS.

The workflow begins with system initialization, during which the TCM distributes the global clock and trigger signals, DDS modules load the waveform parameters, and TTL modules reset counters and TDCs. After initialization, the entanglement sequence is executed, preparing the quantum state for photon emission.

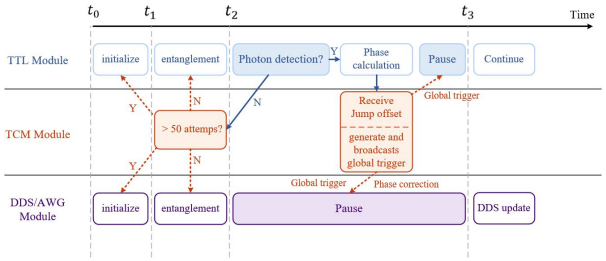


Fig. 3. Swimlane timing diagram of the real-time feedforward workflow. Following each entanglement attempt, photon-detection events trigger timestamp acquisition and phase compensation. Successful events initiate feedforward correction, while unsuccessful attempts return to the entanglement sequence or reinitialization after a predefined number of retries.

Immediately following the entanglement sequence, the TTL module opens the photon-detection window. Upon photon detection, the TTL module acquires the photon arrival timestamp through the synchronized TDC and calculates the corresponding phase correction. The calculated compensation is sent to the TCM, which broadcasts the updated phase information to the target DDS channel. The DDS updates its output phase before the next experimental operation, completing the feedforward loop.

If no photon is detected, the TTL module reports detection failure to the TCM, which triggers another entanglement attempt. This loop continues until a photon is successfully detected or until 50 consecutive attempts have failed. In the latter case, the system returns to initialization to restart the process.

This workflow imposes several stringent requirements on the system: precise timestamp acquisition with picosecond resolution, deterministic low-latency processing for timely phase correction, reliable inter-module synchronization, and support for conditional branching and repeated entanglement attempts within the FPGA-based control architecture.

C. Timestamp Acquisition Module

Accurate photon-arrival timestamp acquisition is a key requirement of the proposed feedforward system, as the measured arrival time can significantly affect the fidelity of subsequent phase-compensated quantum operations.

Figure 4 shows the complete timestamp acquisition chain implemented in the TTL module. The architecture includes a CARRY8-based TDC, encoding logic, clock-domain-crossing circuitry, non-linearity correction, and timestamp aggregation modules.

To provide high-resolution timing information, a carry-chain-based TDC is implemented within the TTL module. The TDC reset is controlled by the running experimental sequence, which dynamically defines the active measurement window. This allows the system to sample photon events only during the relevant entanglement interval, minimizing unnecessary captures and reducing dead time. The reset signal is synchronized with the system timing provided by the TCM

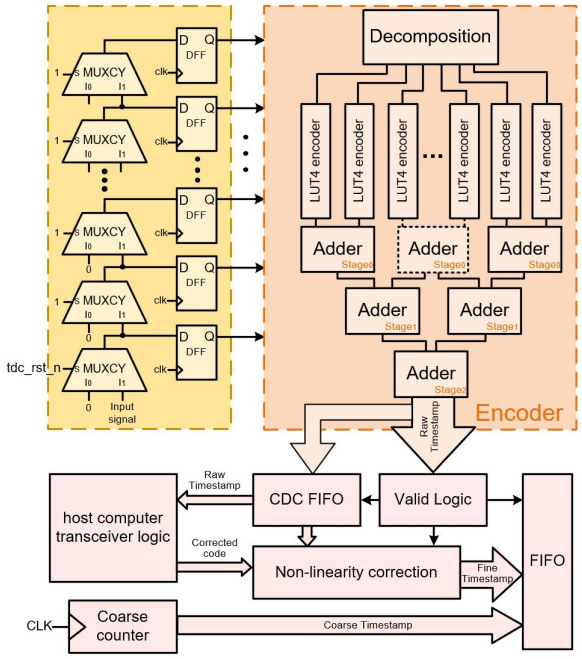


Fig. 4. Architecture of the timestamp acquisition module. Photon-detection events are measured by a CARRY8-based TDC and processed through encoding, validity checking, clock-domain crossing, and non-linearity correction. The calibrated fine timestamp is combined with the coarse counter value and stored for subsequent feedforward processing.

to ensure precise alignment of the TDC window with each entanglement attempt.

The TDC uses FPGA CARRY8 primitives. A 500 MHz sampling clock captures the state of the carry chain taps upon the arrival of a photon-detection event. The sampled thermometer-like pattern is forwarded to an encoder, where it is first divided into multiple groups for independent encoding. The intermediate encoded results are accumulated in three stages to generate the raw TDC code corresponding to the detected transition position.

The raw TDC code is first processed by a valid-logic block to determine whether a valid rising-edge transition exists. Valid samples are then transferred through a clock-domain-crossing FIFO, converting the 500 MHz TDC clock domain to the 250 MHz system-processing domain. The timestamp data follow two parallel paths. One path is sent to the host computer for offline code-density analysis. The resulting calibration data are converted into a lookup table and loaded into the non-linearity correction module. The second path is directly routed to the same module, where raw TDC codes are corrected using the calibration table to compensate for differential and integral nonlinearity.

The corrected fine-time result is combined with the coarse timestamp from a synchronous counter. The resulting timestamp is written into a data-aggregation FIFO, from which it can be accessed by the phase-compensation module in the feedforward engine. This architecture enables calibrated timestamps to be delivered to the feedforward engine with deterministic latency, providing the precise timing information required to maximize the fidelity of real-time phase compensation.

D. Feedforward Processing Module

After timestamp acquisition, the feedforward processing module determines the subsequent execution path of the experiment and generates the phase correction required for the heralded entangled state. As shown in Fig. 5, the module consists of three major components: an embedded RISC-V processor, a microcode arithmetic engine, and a synchronized feedback interface. Together, these modules form a hardware-based feedforward engine capable of performing conditional branching, phase calculation, and distributed synchronization within a deterministic latency.

The embedded RISC-V processor serves as the control core of the feedforward engine. It is responsible for managing the execution flow of the experimental sequence and determining the branch direction according to the photon-detection result. During operation, the processor monitors the timestamp and detection status generated by the timestamp acquisition module. If no valid photon event is detected within the measurement window, the processor generates a branch command that redirects the sequence back to the initialization or entanglement-generation stage, allowing another entanglement attempt to be performed. When a valid photon-detection event is available, the processor transfers execution to the phase-compensation procedure and initiates the subsequent feedforward operation.

Although the phase-compensation algorithm is ultimately derived from the measured timestamp, the computation itself is not executed directly by the RISC-V processor. Implementing multiplication operations within the processor would require additional arithmetic hardware and increase the complexity of the processor datapath, which could negatively affect timing closure and operating frequency. To avoid these issues, all computationally intensive operations are offloaded to a dedicated microcode arithmetic engine.

The microcode arithmetic engine is designed as a lightweight programmable hardware accelerator. It supports addition, subtraction, multiplication, and shift operations, which are sufficient for implementing the phase-calculation algorithm required by the feedforward protocol. Prior to experiment execution, the host computer compiles the phase-calculation procedure into a sequence of microcode instructions and stores them in the instruction RAM of the arithmetic engine. During runtime, the RISC-V processor provides the starting address and instruction count corresponding to the selected calculation routine. The arithmetic engine then sequentially fetches and executes the requested instructions using the measured timestamp as input.

This architecture provides two important advantages. First, the phase-calculation algorithm can be modified through software without requiring changes to the FPGA logic. Second, computational complexity is isolated from the processor core, allowing the control logic to remain compact while maintaining deterministic execution timing. The calculated phase-

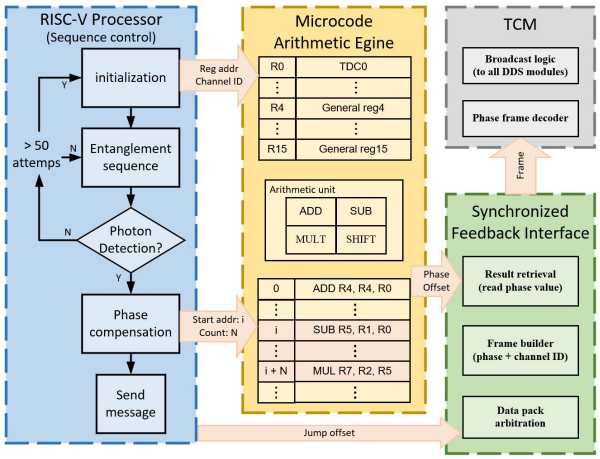


Fig. 5. Feedforward processing module architecture. The RISC-V core controls branch decisions based on photon detection, the microcode engine computes the phase correction, and the synchronized feedback interface broadcasts the result to TCM for DDS phase update.

compensation value is finally stored in the register file of the arithmetic engine for subsequent retrieval.

After the phase calculation is completed, the synchronized feedback interface retrieves the compensation value from the arithmetic-engine register file and packages it together with the destination DDS channel information. The resulting feedback packet is transmitted to the TCM, which serves as the synchronization hub of the distributed control system. Upon receiving the packet, the TCM broadcasts the updated phase information to all participating DDS modules through the dedicated synchronization network.

Each DDS module monitors the broadcast channel and extracts the information corresponding to its assigned output channels. When a matching channel identifier is detected, the DDS updates the associated phase register and applies the calculated compensation to the subsequent waveform output. Since all DDS modules receive the same synchronized broadcast from the TCM, phase updates can be applied in a deterministic and globally synchronized manner.

In addition to phase dissemination, the synchronized feedback interface also generates the global branch-control information required by the distributed execution framework. After the phase-compensation procedure is completed, the RISC-V processor issues a branch command indicating whether the experimental sequence should continue with the compensated operation or return to a previous stage. This branch information is transmitted to the TCM together with the feedback data and subsequently distributed to all participating modules. As a result, phase correction, waveform updates, and sequence transitions are performed under a common timing reference, ensuring consistent system behavior across the entire control platform.

By combining programmable sequence control, hardware-accelerated phase calculation, and synchronized feedback dissemination, the proposed feedforward processing module enables deterministic real-time phase compensation for heralded entanglement experiments while preserving the scalability of the distributed control architecture.

III. Test

A. TTL Module Hardware Platform

The electronic tests were performed using the developed TTL module, whose hardware architecture and photograph are shown in Fig. 6. The module is built around an FPGA device, which serves as the central processing unit of the board. All timestamp acquisition and feedforward-related logic described in Section II are implemented within the FPGA fabric.

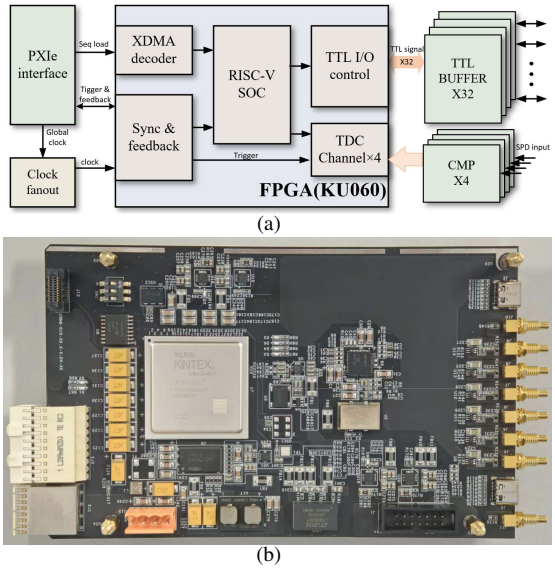


Fig. 6. Hardware architecture and photograph of the developed TTL module. The board integrates FPGA-based control logic, 32 TTL channels, four adjustable comparators, and a PXIe interface for communication and synchronization.

The TTL module interfaces with the PXIe chassis through a PXIe connector. In addition to the PCIe communication link used for configuration and data transfer, the connector also provides global synchronization signals distributed by the TCM, including the system clock, trigger signals, and feedback synchronization signals required for deterministic operation.

The board provides 32 TTL input/output channels for external digital signal control. Four high-speed comparators are integrated to convert photomultiplier tube (PMT) analog signals into FPGA-compatible LVDS pulses with adjustable thresholds. The conditioned LVDS pulses are routed to the FPGA, where timestamp acquisition is performed by the embedded CARRY8-based TDC described in Section II-C. This architecture enables the TTL module to simultaneously provide experiment control, photon-event detection, and high-resolution timestamp acquisition within a single hardware platform.

B. TDC Performance Evaluation

The performance of the implemented TDC was characterized through code-density and timing measurements.

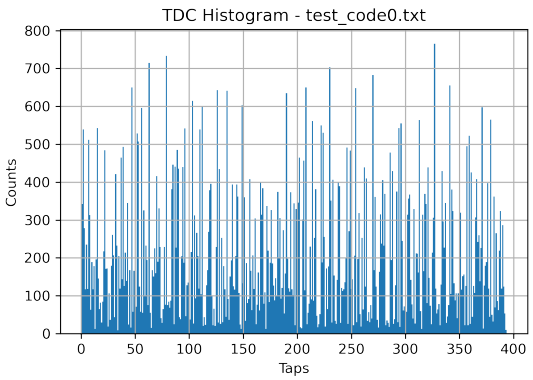
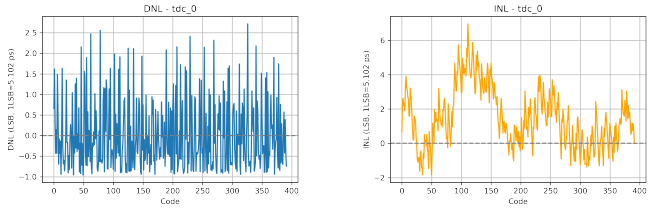


Fig. 7. code density

Code-density analysis was first performed to evaluate the linearity of the TDC. A large number of events were collected under uniformly distributed sampling conditions, and the occurrence frequency of each TDC code was recorded. The resulting code-density histogram is shown in Fig. 7. Based on the measured code density, the differential nonlinearity (DNL) and integral nonlinearity (INL) were calculated, and the results are presented in Fig. 8. The calibration table derived from the code-density analysis was subsequently stored in the FPGA memory and used for online non-linearity correction.



(a) (b)

Fig. 8. Measured differential nonlinearity (DNL) and integral nonlinearity (INL) of the implemented TDC.

After calibration, the timing precision of the TDC was evaluated using a cable-delay measurement setup. A pulse generated by a signal source was divided into two synchronous outputs through a power splitter and routed to two TDC input channels through cables of different lengths. Since the two signals originated from the same source, the measured time difference corresponds to the relative propagation delay introduced by the cable-length difference.

A large number of measurements were collected and the resulting time-difference distribution is shown in Fig. 9. The measured distribution exhibits an RMS timing uncertainty of 11.5 ps, demonstrating the effectiveness of the calibration procedure and the high timing precision achieved by the implemented TDC.

C. System-Level Feedforward Verification

To verify the functionality of the complete feedforward chain, a system-level electronic test was performed. Instead of using experimentally measured timestamps, simulated TDC values were generated by the host computer and injected into the feedforward processing module. The simulated timestamps were processed through the same feedforward path described in Section II, including phase-compensation calculation, synchronized feedback transmission, and DDS phase update.

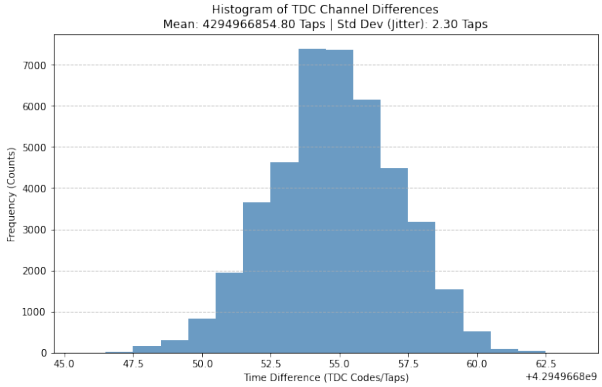
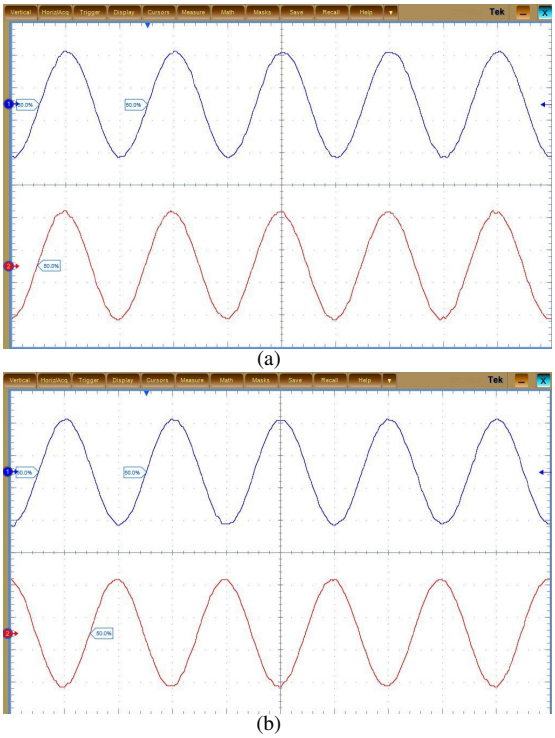


Fig. 9. Measured time-difference distribution between two calibrated TDC channels.

To clearly demonstrate the operation of the feedforward system, the phase-compensation algorithm was configured to generate a phase correction of 180°. Two DDS output channels were selected for verification. One channel served as a reference, while the other acted as the feedforward-controlled channel. Before the simulated timestamp was applied, both DDS channels were configured with identical output phases.



(a)

(b)

Fig. 10. Waveform demonstration of the system-level feedforward function. The phase shift confirms correct operation of the complete feedforward chain, from timestamp processing to DDS phase update.

After the simulated TDC value was injected, the feedforward engine calculated the corresponding phase-compensation value and transmitted the result to the target DDS channel through the synchronized feedback network. As a result, the phase of the controlled DDS channel shifted by 180° relative to the reference channel. The measured waveforms before and after feedforward correction are shown in Fig. 10. The observed phase transition confirms the correct operation of the complete feedforward chain, including timestamp processing, phase calculation, feedback dissemination, and DDS phase updating.

D. System-Level End-to-End Latency Measurement

To evaluate the real-time performance of the proposed feedforward architecture, a system-level closed-loop latency measurement was conducted using a synchronized electronic test setup.

In this experiment, a TTL pulse generated by the FPGA-based TTL module was used to emulate a photomultiplier tube (PMT) detection event. The trigger signal was injected into the TDC measurement window and synchronized with the global timing reference provided by the TCM, ensuring deterministic alignment across all system modules.

Upon detection of the trigger event, the TDC acquired the corresponding timestamp, which was subsequently processed by the FPGA-based feedforward engine to compute the required phase correction. The computed result was transmitted through the synchronized feedback interface to the DDS module, which updated the output phase accordingly.

Two 50 MHz DDS channels were initialized with identical phases prior to the experiment. After the feedforward event was triggered, one channel applied the computed phase correction while the reference channel remained unchanged, enabling direct observation of the phase transition.

The system latency T_{lat} is defined as the time interval between the falling edge of the TTL trigger signal (emulated detection event) and the rising edge of the DDS response signal after phase update. This definition captures the complete end-to-end processing chain, including signal acquisition, timestamp conversion, FPGA computation, inter-module communication, and waveform update.

Experimental measurements using a high-resolution oscilloscope show that the proposed system achieves a deterministic end-to-end latency of approximately 800 ns. The latency remains stable across repeated measurements, indicating that the feedforward process operates within a deterministic timing domain governed by the global clock distribution network.

The measured waveform of the complete feedforward operation is shown in Fig. 11, where a clear and repeatable phase transition is observed after the trigger event.

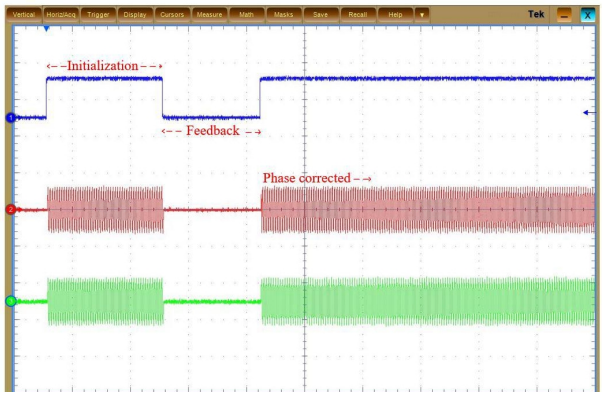


Fig. 11. System-level feedforward latency and phase response measurement. The measured end-to-end latency from trigger to DDS response is approximately 800 ns.

E. System-Level Validation of Real-Time Phase Feedforward via Ramsey Interference

To verify that the implemented real-time control system is capable of supporting heralded Bell-state experiments in quantum networks, we perform a Ramsey-interference-based benchmark of the full timestamp-to-phase feedforward pipeline. The objective of this experiment is to validate the deterministic phase-correction capability required for real-time stabilization of entangled quantum states under stochastic timing conditions. In particular, this benchmark serves as a functional surrogate for heralded entanglement experiments, where photon detection events introduce random arrival times that map directly onto phase uncertainty. The experimental configuration is shown in Fig. 12, which illustrates the integration of an external DDS, synchronized TTL-based timestamp acquisition, FPGA-based real-time phase computation, and deterministic phase update applied to the second Ramsey pulse.

The experiment is implemented using a Ramsey sequence with randomized trigger timing. The first $\pi/2$ pulse is generated by an external DDS together with a synchronized TTL signal that defines the timing reference for each experimental shot. This signal is captured by the system time-to-digital converter (TDC) and converted into a timestamp Δt, which determines the effective evolution time $\tau(\Delta t)$. The second $\pi/2$ pulse is generated by the system DDS and applied at a fixed reference time T, with its phase either kept constant or updated in real time based on the measured timestamp. The resulting Ramsey signal is given by

$$P_e = P_0 + \frac{C}{2}\cos\left(\delta\tau(\Delta t) + \phi_1 - \phi_2 + \phi_{\mathrm{sys}}\right). \tag{15}$$

In the absence of real-time feedback, the second pulse phase is fixed as $\phi_2 = c$. Under this condition, stochastic variations in $\tau(\Delta t)$ lead to uncontrolled phase accumulation across experimental shots. Since each realization corresponds to a different effective evolution time, the Ramsey phase becomes uniformly distributed over the ensemble. As a result, the coherent contribution averages out, yielding a flat response $\langle P_e \rangle \approx P_0$, which indicates complete loss of interference visibility due to timing-induced phase randomization.

In contrast, when real-time phase feedforward is enabled, the control system extracts the timestamp Δt and updates the second pulse phase according to

$$\phi_2(\Delta t, c) = \delta\hat{\tau}(\Delta t) + \phi_{\mathrm{cal}} + c. \tag{16}$$

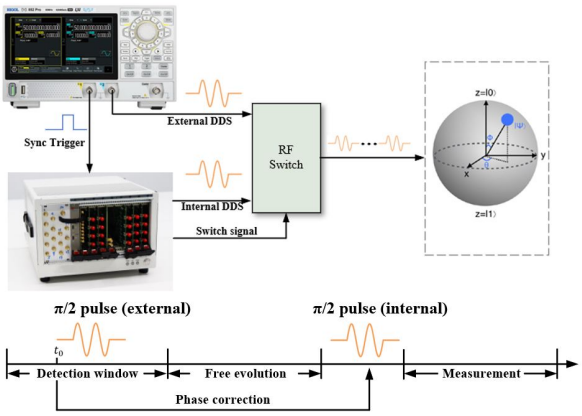


Fig. 12. Experimental setup for real-time phase feedforward Ramsey experiment. An external DDS generates the first Ramsey $\pi/2$ pulse together with a synchronized TTL trigger signal used for timing acquisition. The second Ramsey pulse is generated by the system DDS via RF switching and is applied after real-time phase correction based on the measured timing information. The corrected pulse drives the quantum system, represented on the Bloch sphere.

Here $\hat{\tau}(\Delta t)$ denotes the hardware-estimated evolution time, while ϕ_{cal} accounts for static calibration offsets in the microwave and optical signal paths. When the timing estimation satisfies $\hat{\tau}(\Delta t) \approx \tau(\Delta t)$, the stochastic phase contribution introduced by random triggering is canceled in real time at the hardware level. Consequently, the Ramsey response reduces to

$$P_e \approx P_0 + \frac{C}{2}\cos(c + c_0), \tag{17}$$

recovering a stable high-visibility interference fringe even in the presence of shot-to-shot timing fluctuations.

The experimental comparison between the two regimes is summarized in Fig. 13. Without feedback, the Ramsey signal is fully washed out due to ensemble averaging over stochastic delays, reflecting uncontrolled phase evolution. With real-time phase correction enabled, a clear and stable sinusoidal oscillation is recovered after averaging over random trigger events. This sharp transition demonstrates that the system is capable of deterministically converting stochastic timing information into coherent phase corrections within a single experimental cycle, including timestamp acquisition, phase computation, and synchronized DDS phase update.

From a systems perspective, this result directly validates the real-time feedback architecture required for heralded quantum network operation. In such systems, photon detection events define a stochastic temporal reference frame, and successful entanglement generation requires converting this randomness into a well-defined phase basis. The observed restoration of Ramsey interference therefore confirms that the implemented hardware pipeline can preserve phase coherence under timing uncertainty.

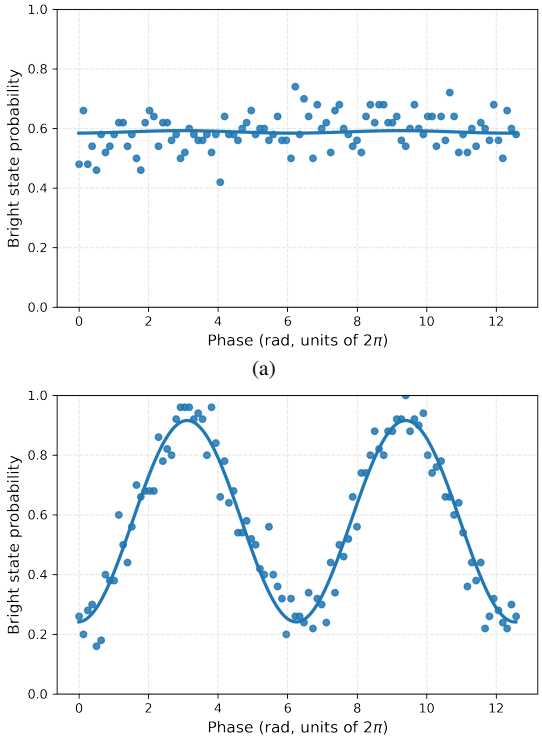


Fig. 13. Experimental results of real-time phase feedforward validation using Ramsey interference.Comparison between feedback-off and feedback-on regimes. Without real-time phase correction, stochastic timing fluctuations lead to complete dephasing after ensemble averaging, resulting in a flat Ramsey response. With real-time feedforward enabled, the system recovers a high-visibility sinusoidal interference fringe, demonstrating deterministic cancellation of timing-induced phase errors through timestamp-based phase compensation.

IV. Conclusion

We have developed and experimentally demonstrated a real-time phase feedforward control system for quantum network applications. The platform integrates high-resolution timestamp acquisition with deterministic FPGA-based processing to enable shot-by-shot phase correction based on stochastic timing events. Using a Ramsey-interference-based benchmark, we experimentally validated the full feedback chain, including time-to-digital conversion, real-time phase computation, and synchronized phase update on microwave control pulses. The results show that the system can fully recover coherent interference under random timing conditions, while coherence is lost without feedback. This confirms the capability of the proposed architecture to perform deterministic phase stabilization, providing a key enabling technology for high-fidelity Bell-state generation in heralded quantum network experiments.

References

[1] H. J. Kimble, "The quantum internet," *Nature*, vol. 453, pp. 1023–1030, 2008.
[2] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, p. eaam9288, 2018.

[3] G. Guccione, T. Darras, H. Le Jeannic, V. B. Verma, S. W. Nam, A. Cavaillès, and J. Laurat, "Connecting heterogeneous quantum networks by hybrid entanglement swapping," *Science advances*, vol. 6, no. 22, p. eaba4508, 2020.

[4] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, "Long-distance quantum communication with atomic ensembles and linear optics," *Nature*, vol. 414, pp. 413–418, 2001.

[5] D. L. M. et al., "Entanglement of single-atom quantum bits at a distance," *Nature*, vol. 449, pp. 68–71, 2007.

[6] L. J. S. et al., "High-rate, high-fidelity entanglement of qubits across an elementary quantum network," *Phys. Rev. Lett.*, vol. 124, p. 110501, 2020.

[7] V. K. et al., "Entanglement of trapped-ion qubits separated by 230 meters," *Phys. Rev. Lett.*, vol. 130, p. 050803, 2023.

[8] C. Cabrillo, J. I. Cirac, P. García-Fernández, and P. Zoller, "Creation of entangled states of distant atoms by interference," *Phys. Rev. A*, vol. 59, pp. 1025–1033, 1999.

[9] F. Xu, M. Wang, C. Qiao, S. Li, H. Wang, and X. Su, "Hybrid entanglement carrying orbital angular momentum," *Science Bulletin*, vol. 70, no. 6, pp. 876–881, 2025.

[10] J. Hofmann, M. Krug, N. Ortegel, L. Gérard, M. Weber, W. Rosenfeld, and H. Weinfurter, "Heralded entanglement between widely separated atoms," *Science*, vol. 337, no. 6090, pp. 72–75, 2012.

[11] A. R. et al., "Multiplexed entanglement of multi-emitter quantum network nodes," *Nature*, vol. 639, pp. 54–59, 2025.

[12] A. J. Stolk, K. L. van der Enden, M.-C. Slater, I. te Raa-Derckx, P. Botma, J. Van Rantwijk, J. B. Biemond, R. A. Hagen, R. W. Herfst, W. D. Koek *et al.*, "Metropolitan-scale heralded entanglement of solid-state qubits," *Science advances*, vol. 10, no. 44, p. eadp6442, 2024.

[13] R. Prevedel, P. Walther, F. Tiefenbacher, P. Böhi, R. Kaltenbaek, T. Jennewein, and A. Zeilinger, "High-speed linear optics quantum computing using active feed-forward," *Nature*, vol. 445, no. 7123, pp. 65–69, 2007.

[14] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. Kim, "Quantum generalisation of feedforward neural networks, npj quantum inf," 2017.

[15] Q.-C. Sun, Y.-L. Mao, S.-J. Chen, W. Zhang, Y.-F. Jiang, Y.-B. Zhang, W.-J. Zhang, S. Miki, T. Yamashita, H. Terai *et al.*, "Quantum teleportation with independent sources and prior entanglement distribution over a network," *Nature Photonics*, vol. 10, no. 10, pp. 671–675, 2016.

[16] C. H. Chow, B. L. Ng, and C. Kurtsiefer, "Coherence of a dynamically decoupled single neutral atom," *Journal of the Optical Society of America B*, vol. 38, no. 2, pp. 621–629, 2021.

[17] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, "High-fidelity preparation, gates, memory and readout of a trapped-ion quantum bit," *Physical Review Letters*, vol. 113, p. 220501, 2014.

[18] J. Wu and Z. Shi, "The 10-ps wave union tdc: Improving fpga tdc resolution beyond its cell delay," in *2008 IEEE Nuclear Science Symposium Conference Record*. IEEE, 2008, pp. 3440–3446.

[19] W. Pan, G. Gong, and J. Li, "A 20-ps time-to-digital converter (tdc) implemented in field-programmable gate array (fpga) with automatic temperature correction," *IEEE Transactions on Nuclear Science*, vol. 61, no. 3, pp. 1468–1473, 2014.

[20] B. Wu, Y. Wang, Q. Cao, Z. Li, X. Li, X. Zhou, Y. Hu, Z. Wang, M. Shao, J. Liu *et al.*, "Design of time-to-digital converters for time-over-threshold measurement in picosecond timing detectors," *IEEE Transactions on Nuclear Science*, vol. 68, no. 4, pp. 470–476, 2021.

[21] S. Tancock, E. Arabul, and N. Dahnoun, "A review of new time-to-digital conversion techniques," *IEEE transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3406–3417, 2019.

[22] J. Song, Q. An, and S. Liu, "A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays," *IEEE transactions on nuclear science*, vol. 53, no. 1, pp. 236–241, 2006.

[23] Q. Shen, S. Liu, B. Qi, Q. An, S. Liao, P. Shang, C. Peng, and W. Liu, "A 1.7 ps equivalent bin size and 4.2 ps rms fpga tdc based on multichain measurements averaging method," *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 947–954, 2015.

[24] Y. Wang, J. Kuang, C. Liu, and Q. Cao, "A 3.9-ps rms precision time-to-digital converter using ones-counter encoding scheme in a kintex-7 fpga," *IEEE Transactions on Nuclear Science*, vol. 64, no. 10, pp. 2713–2718, 2017.

[25] A. Sitaram, G. K. Campbell, and A. Restelli, "Programmable system on chip for controlling an atomic physics experiment," *Review of Scientific Instruments*, vol. 92, no. 5, 2021.

[26] W.-Z. Zhang, X. Qin, L. Wang, Y. Tong, Y. Rui, X. Rong, and J.-F. Du, "A fully-adjustable picosecond resolution arbitrary timing generator based on multi-stage time interpolation," *Review of Scientific Instruments*, vol. 90, no. 11, 2019.

[27] S. Reisenbauer, P. Behal, G. Wachter, and M. Trupke, "Lithpulser: An open-source pulse generator with 1 ns time resolution based on the red pitaya stemlab 125-10 featuring real-time conditional logic for experimental control," *Review of Scientific Instruments*, vol. 93, no. 1, 2022.

[28] A. M. Amiri, A. Khouas, and M. Boukadoum, "Pseudorandom stimuli generation for testing time-to-digital converters on an fpga," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 7, pp. 2209–2215, 2009.

[29] L. Miari, S. Antonioli, I. Labanca, M. Crotti, I. Rech, and M. Ghioni, "Eight-channel fully adjustable pulse generator," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 9, pp. 2399–2408, 2015.

[30] M.-D. Zhu, L. Yan, X. Qin, W.-Z. Zhang, Y. Lin, and J. Du, "Fpga based hardware platform for trapped-ion-based multi-level quantum systems," *Chinese Physics B*, vol. 32, no. 9, p. 090702, 2023.

[31] T. Pruttivarasin and H. Katori, "Compact field programmable gate array-based pulse-sequencer and radio-frequency generator for experiments with trapped atoms," *Review of Scientific Instruments*, vol. 86, no. 11, 2015.

[32] Y. Yang, Z. Shen, X. Zhu, Z. Wang, G. Zhang, J. Zhou, X. Jiang, C. Deng, and S. Liu, "Fpga-based electronic system for the control and readout of superconducting quantum processors," *Review of Scientific Instruments*, vol. 93, no. 7, 2022.

[33] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler *et al.*, "Compact ion-trap quantum computing demonstrator," *PRX quantum*, vol. 2, no. 2, p. 020343, 2021.

[34] D. D. Caviglia, A. De Gloria, G. Donzellini, G. Parodi, and D. Ponta, "Design and construction of an arbitrary waveform generator," *IEEE transactions on instrumentation and measurement*, vol. 32, no. 3, pp. 398–403, 1983.

[35] M. B. Yeary, R. J. Fink, D. Beck, D. W. Guidry, and M. Burns, "A dsp-based mixed-signal waveform generator," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 3, pp. 665–671, 2004.