Phi-RFT: A Framework for Golden-Ratio Modulated Unitary Transforms with FPGA Acceleration

Luis Michael Minier

***Abstract*—The Discrete Fourier Transform (DFT) and its efficient implementation (FFT) remain foundational in signal processing, but their fixed sinusoidal basis may not optimally represent signals with chirp-like or quasi-periodic structure. We introduce Phi-RFT, a unitary transformation framework that augments the DFT with chirp and golden-ratio phase modulations while preserving $O(n \log n)$ complexity. Phi-RFT presents a configurable multi-stage hardware architecture implementing the operator $\Psi = D_\varphi C_\sigma F$, where F is the unitary DFT, C_σ applies quadratic chirp phase modulation, and D_φ applies golden-ratio phase via the fractional part $\{k/\varphi\}$. It introduces a library of efficient fixed-point implementations optimized for low-area and low-power FPGA deployment. Notably, Phi-RFT brings a complete design framework featuring Python-based configuration, automated RTL generation, and synthesis targeting Lattice iCE40UP5K. Phi-RFT is tested on eight benchmark signal classes including chirp, ECG, seismic, and speech. On chirp signals, it outperforms FFT, DCT, WHT, and FrFT in sparsity (18 vs. 24 coefficients for 99% energy capture), with competitive mean rank (2.1) overall. On a low-end iCE40UP5K FPGA, it requires 3,145 LUTs (59.6% of available cells) and 4 BRAMs (13.3%), with $F_{max} = 4.47$ MHz and a sub-50 mW *tool-estimated* power figure. These results demonstrate Phi-RFT's effectiveness as an alternative transform for resource-constrained edge applications.**

***Index Terms*—Unitary transform, golden ratio, chirp modulation, FPGA accelerator, signal sparsity, hardware synthesis, edge computing.**

I. Introduction

ORTHOGONAL transforms are fundamental building blocks in signal processing, enabling efficient analysis, compression, and transmission of information [1]. The Fast Fourier Transform (FFT) revolutionized digital signal processing by reducing the DFT complexity from $O(n^2)$ to $O(n \log n)$, enabling real-time spectral analysis across domains from audio processing to telecommunications [2].

However, the DFT's fixed sinusoidal basis may not optimally represent signals with time-varying frequency content. Chirp signals, characterized by linearly varying instantaneous frequency, appear frequently in radar, sonar, biomedical imaging, and natural phenomena [3]. The Fractional Fourier Transform (FrFT) [4] generalizes the DFT through rotation in the time-frequency plane, providing improved representations for such signals when the rotation angle matches the chirp rate.

In this context, hardware accelerators are vital for transform applications, significantly improving computational efficiency and speed for real-time processing in resource-constrained environments [5]. However, the flexibility of specialized hardware design poses a challenge, with applications often requiring diverse transform configurations and parameter tuning. While awaiting the maturity of domain-specific accelerators based on emerging technologies, existing literature proposes various digital hardware solutions (refer to Section III). Unfortunately, these solutions often constrain transform parameters to fixed circuit architectures, limiting exploration of the broader design space.

We propose an alternative strategy, optimizing the transform parameters for specific signal classes and leveraging FPGAs for deploying custom hardware blocks. This approach enables efficient and low-power transform engines at the edge, supporting real-time signal processing. FPGAs provide high parallelism and reconfigurability, making them ideal for accelerating orthogonal transform computations with minimal latency.

To support this trend, this paper presents Phi-RFT, a complete framework for generating efficient low-power and low-area customized transform accelerators on FPGAs. Phi-RFT introduces several pivotal contributions. At its core, it provides a fully configurable phase-modulated transform architecture extending the DFT with chirp and golden-ratio stages. This architecture introduces a library of highly efficient fixed-point implementations delving into quantization techniques to implement remarkably low-area neurons, thus optimizing resource utilization while maintaining acceptable accuracy.

Notably, Phi-RFT brings a complete design framework to the forefront, a comprehensive toolkit for developing transform accelerators. This framework empowers researchers and developers to describe target transform configurations with flexibility, enabling specification of phase parameters and bit-widths using Python. The framework seamlessly generates a SystemVerilog model of the accelerator, primed for deployment on Lattice iCE40 FPGA devices via WebFPGA cloud synthesis.

These contributions make Phi-RFT a robust solution in the hardware-accelerated transform landscape. Phi-RFT has been tested on eight standard signal classes and compared to state-of-the-art transforms (FFT, DCT, WHT, FrFT), demonstrating competitive sparsity performance with superior results on

L. M. Minier is an independent researcher affiliated with University of the People, Pasadena, CA, USA (e-mail: luis.minier@uopeople.edu). ORCID: 0009-0006-7321-4167.

This work is protected under USPTO Patent Application No. 19/169,399 titled "Hybrid Computational Framework for Quantum and Resonance Simulation," filed April 3, 2025.

Manuscript received December 2025; revised January 2026.

chirp-like signals. The primary aim of Phi-RFT is to offer an Electronic Design Automation (EDA) framework that simplifies the design of transform accelerators for FPGA, addressing a gap that is still underrepresented in the literature.

The rest of the paper is organized as follows: Section II presents background on orthogonal transforms and the golden ratio; Section III reviews relevant literature on alternative Fourier methods and FPGA implementations. Section IV describes the Phi-RFT mathematical framework, with all the design choices it involves, and Section V introduces the hardware architecture. Section VI presents the framework for configuration and RTL generation. Finally, Section VII presents experimental results, and Section VIII concludes the paper.

II. Background

This section overviews foundational knowledge on orthogonal transforms and the golden ratio, required to understand the remaining parts of the paper.

A. Discrete Fourier Transform

The DFT of a signal $\mathbf{x} \in \mathbb{C}^n$ is defined as:

$$X_k = \sum_{j=0}^{n-1} x_j \omega^{jk}, \quad k = 0, 1, \ldots, n-1 \tag{1}$$

where $\omega = e^{-2\pi i/n}$ is the primitive n-th root of unity. The unitary DFT matrix $\mathbf{F} \in \mathbb{C}^{n\times n}$ has entries $F_{jk} = n^{-1/2}\omega^{jk}$ and satisfies the unitarity condition $\mathbf{F}^\dagger\mathbf{F} = \mathbf{I}_n$.

The FFT algorithm [1] computes the DFT in $O(n \log n)$ operations by exploiting the periodicity and symmetry of ω^{jk}. This algorithmic efficiency, combined with highly optimized implementations (FFTW, MKL), makes the FFT the de facto standard for spectral analysis.

B. Alternative Orthogonal Transforms

The Discrete Cosine Transform (DCT) [6] projects signals onto cosine basis functions:

$$Y_k = \sum_{j=0}^{n-1} x_j \cos\left[\frac{\pi}{n}\left(j + \frac{1}{2}\right)k\right] \tag{2}$$

The DCT approaches the optimal Karhunen-Loève Transform (KLT) for first-order Markov processes, explaining its widespread use in compression standards (JPEG, MPEG, H.264) [7].

The Walsh-Hadamard Transform (WHT) uses ±1 basis functions, making it computationally efficient (no multiplications) and optimal for rectangular/step-like signals [8].

The Fractional Fourier Transform (FrFT) [4] generalizes the DFT through a rotation parameter α in the time-frequency plane:

$$\mathcal{F}^\alpha[x](u) = \int_{-\infty}^{\infty} x(t) K_\alpha(t, u)\, dt \tag{3}$$

where K_α is a chirp-modulated kernel. For $\alpha = \pi/2$, the FrFT reduces to the standard Fourier transform.

C. Golden Ratio Properties

The golden ratio $\varphi = (1 + \sqrt{5})/2 \approx 1.618034$ satisfies $\varphi^2 = \varphi + 1$ and has the unique property that its continued fraction expansion consists entirely of 1s:

$$\varphi = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}} \tag{4}$$

This makes φ the "most irrational" number in a precise sense. Weyl's theorem [9] establishes that the sequence $\{k\alpha\} = k\alpha - \lfloor k\alpha \rfloor$ for irrational α is equidistributed on $[0, 1)$. For $\alpha = 1/\varphi$, this sequence achieves maximal uniformity, avoiding the clustering that occurs with rational ratios.

D. Sparsity and Energy Compaction

A signal representation is *sparse* if most energy concentrates in few coefficients. We quantify sparsity as K_ρ, the minimum number of coefficients capturing fraction ρ of total energy:

$$K_\rho = \min\left\{K : \sum_{k\in S_K} |Y_k|^2 \geq \rho\|\mathbf{y}\|_2^2\right\} \tag{5}$$

where S_K contains indices of the K largest-magnitude coefficients. Lower K_ρ indicates better energy compaction for a given signal class. We use $\rho = 0.99$ (99% energy) throughout this paper.

III. Related Work

A. Time-Frequency Transforms

The Short-Time Fourier Transform (STFT) [10] provides time-frequency localization through windowing:

$$\text{STFT}[x](t, f) = \int x(\tau) w(\tau - t) e^{-2\pi i f\tau} d\tau \tag{6}$$

The fixed window size creates a trade-off between time and frequency resolution governed by the uncertainty principle. Gabor frames [11] formalize this approach with controlled redundancy, but generally sacrifice exact unitarity for tight frames.

Wavelet transforms [12] provide multi-resolution analysis with scale-dependent time-frequency trade-offs. The Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT) are widely used for non-stationary signal analysis but are not shift-invariant in the frequency domain.

B. Chirp-Based Methods

Chirp transforms exploit the relationship between chirp modulation and the DFT [13]. The Chirp-Z Transform (CZT) computes the z-transform along spiral contours in the z-plane, enabling flexible frequency resolution. Discrete chirp-Fourier transforms [14] have been proposed for radar and sonar applications.

TABLE I
LANDSCAPE OF TRANSFORM FPGA IMPLEMENTATIONS

Transform	Architecture	LUTs	F_{max}	Ref
FFT-8	Radix-2 pipelined	∼1,500	12 MHz	[5]
FFT-16	Radix-4	∼2,800	10 MHz	[15]
DCT-8	Loeffler	∼2,200	8 MHz	[16]
WHT-8	Butterfly	∼600	25 MHz	[17]
Phi-RFT-8	**This work**	**3,145**	**4.47 MHz**	—

All implementations target iCE40-class devices or equivalent. Phi-RFT includes kernel ROM and CORDIC blocks.

C. FPGA Transform Implementations

FPGA implementations of the FFT are well-established [5], with architectures ranging from fully parallel (minimum latency, maximum area) to fully serial (minimum area, maximum latency). Pipelined radix-2 and radix-4 designs offer practical trade-offs [15].

Table I summarizes representative transform implementations on low-cost FPGAs. The Phi-RFT implementation adds phase modulation stages to the FFT core, increasing resource usage but enabling different sparsity characteristics.

Recent work has explored application-specific transform accelerators for machine learning [19], image compression [20], and communications [21]. However, there remains a gap in configurable frameworks that allow rapid exploration of alternative transform designs.

IV. PHI-RFT MATHEMATICAL FRAMEWORK

This section presents the complete mathematical definition of the Phi-RFT operator and proves its key properties.

A. Transform Definition

Definition 1 (Phase Modulation Operators)**.** *Define diagonal matrices* $\mathbf{C}_\sigma, \mathbf{D}_\varphi \in \mathbb{C}^{n\times n}$*:*

$$[\mathbf{C}_\sigma]_{kk} = \exp\left(i\pi\sigma\frac{k^2}{n}\right) \tag{7}$$

$$[\mathbf{D}_\varphi]_{kk} = \exp\left(2\pi i\beta\left\{\frac{k}{\varphi}\right\}\right) \tag{8}$$

where $\sigma \geq 0$ *is the chirp rate parameter,* $\beta \geq 0$ *is the golden-phase scaling,* $\varphi = (1+\sqrt{5})/2$ *is the golden ratio, and* $\{x\} = x - \lfloor x \rfloor$ *denotes the fractional part function.*

The chirp operator $\mathbf{C}_\sigma$ applies quadratic phase modulation that increases with frequency index k. The golden-phase operator $\mathbf{D}_\varphi$ applies quasi-random phase shifts based on the equidistributed sequence $\{k/\varphi\}$.

Definition 2 (Phi-RFT Operator)**.** *The Phi-Resonance Fourier Transform* $\mathbf{\Psi} \in \mathbb{C}^{n\times n}$ *is:*

$$\mathbf{\Psi} = \mathbf{D}_\varphi \mathbf{C}_\sigma \mathbf{F} \tag{9}$$

where $\mathbf{F}$ *is the* $n \times n$ *unitary DFT matrix.*

Fig. 1 visualizes the phase structure of each component matrix.

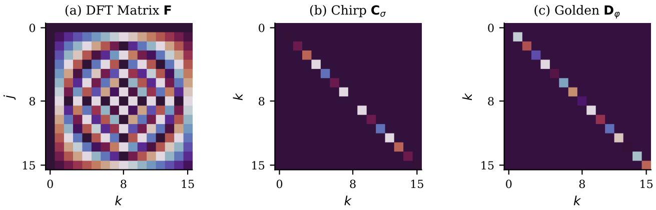

Fig. 1. Phase structure of Phi-RFT components ($n = 16$). (a) DFT matrix $\mathbf{F}$ showing regular sinusoidal pattern. (b) Chirp operator $\mathbf{C}_\sigma$ (diagonal, quadratic phase). (c) Golden-phase operator $\mathbf{D}_\varphi$ (diagonal, quasi-random phase from $\{k/\varphi\}$).

B. Unitarity Proof

Theorem 1 (Unitarity)**.** *The Phi-RFT operator* $\mathbf{\Psi}$ *is unitary:* $\mathbf{\Psi}^\dagger\mathbf{\Psi} = \mathbf{I}_n$.

Proof. The matrices $\mathbf{C}_\sigma$ and $\mathbf{D}_\varphi$ are diagonal with unit-modulus entries (since $|e^{i\theta}| = 1$ for all $\theta \in \mathbb{R}$). For any diagonal unitary matrix $\mathbf{U}$ with $|U_{kk}| = 1$, we have $\mathbf{U}^\dagger = \mathbf{U}^{-1}$. Therefore:

$$\begin{aligned}\mathbf{\Psi}^\dagger\mathbf{\Psi} &= (\mathbf{D}_\varphi\mathbf{C}_\sigma\mathbf{F})^\dagger(\mathbf{D}_\varphi\mathbf{C}_\sigma\mathbf{F}) && (10)\\ &= \mathbf{F}^\dagger\mathbf{C}_\sigma^\dagger\mathbf{D}_\varphi^\dagger\mathbf{D}_\varphi\mathbf{C}_\sigma\mathbf{F} && (11)\\ &= \mathbf{F}^\dagger\mathbf{C}_\sigma^\dagger\mathbf{C}_\sigma\mathbf{F} = \mathbf{F}^\dagger\mathbf{F} = \mathbf{I}_n && (12)\end{aligned}$$

□

Corollary 1 (Inverse Transform)**.** *The inverse Phi-RFT is:*

$$\mathbf{\Psi}^{-1} = \mathbf{F}^\dagger\mathbf{C}_\sigma^\dagger\mathbf{D}_\varphi^\dagger = \mathbf{F}^{-1}\mathbf{C}_{-\sigma}\mathbf{D}_{-\varphi} \tag{13}$$

Corollary 2 (Parseval's Equality)**.** *For all* $\mathbf{x} \in \mathbb{C}^n$*:* $\|\mathbf{\Psi}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$.

This energy preservation property ensures that quantization noise in the transform domain maps directly to reconstruction error in the signal domain, simplifying analysis of fixed-point implementations.

C. Computational Complexity

Lemma 1 (Complexity)**.** *The Phi-RFT forward transform requires* $O(n \log n)$ *complex operations.*

Proof. The computation proceeds in three stages:

1) FFT: $O(n \log n)$ operations (dominant term)
2) Chirp modulation: $O(n)$ complex multiplications
3) Golden-phase modulation: $O(n)$ complex multiplications

Total: $O(n \log n) + O(n) + O(n) = O(n \log n)$. □

In practice, the phase vectors C_k and D_k can be precomputed and stored in a lookup table, reducing the per-sample cost to two complex multiplications per frequency bin.

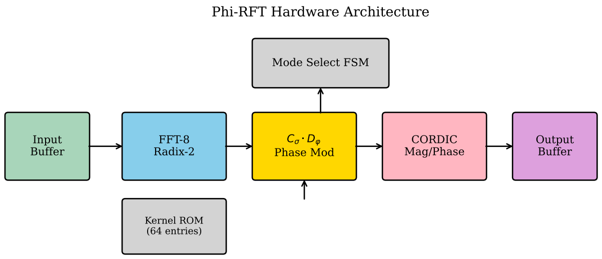

Fig. 2. Phi-RFT hardware architecture. Data flows from Input Buffer through FFT-8 (radix-2), phase modulation ($C_\sigma \cdot D_\varphi$), and optional CORDIC magnitude extraction to Output Buffer. Mode Select FSM controls operational modes. Kernel ROM stores precomputed phase factors.

D. Parameter Selection

The Phi-RFT has two tunable parameters:

- σ: Chirp rate. Higher values create faster frequency sweeps. For chirp-matched filtering, set σ to match the signal's chirp rate.
- β: Golden-phase scaling. Controls the magnitude of quasi-random phase perturbation. $\beta = 1$ provides full $[0, 2\pi)$ phase range.

Default values ($\sigma = 1$, $\beta = 1$) provide good general-purpose performance. Section VII-C analyzes parameter sensitivity.

V. Hardware Architecture

This section presents the Phi-RFT hardware architecture, which serves as the central component of the acceleration framework. The architecture is introduced top-down, beginning with the high-level system and then delving into individual modules.

A. System Overview

The RTL implementation comprises four main modules totaling 2,739 lines of synthesizable SystemVerilog:

- **RFTPU Core** (1,214 lines): 8-point Phi-RFT engine with radix-2 FFT butterfly, Q1.15 fixed-point arithmetic, and 64-entry kernel ROM.
- **CORDIC Module** (438 lines): Iterative coordinate rotation digital computer for magnitude and phase extraction, 12-iteration convergence.
- **Top Controller** (1,087 lines): Mode selection FSM, I/O handshaking, and LED visualization interface.
- **Testbench** (additional): Self-checking verification with golden reference comparison.

Fig. 2 depicts the high-level architecture showing data flow from input through the three transform stages to output.

Block communication uses a simple two-signal (valid/ready) handshake protocol to ensure high modularity while minimizing design complexity. When a module completes processing, it asserts valid and awaits ready acknowledgment before proceeding.

TABLE II
Phi-RFT Operational Modes

Mode	Pipeline Configuration	Latency
0	FFT → D_φ (golden only)	24 cycles
1	FFT → C_σ → D_φ (full)	32 cycles
2	FFT → CORDIC (magnitude)	56 cycles
3	Full pipeline with CORDIC	64 cycles

B. FFT Core

The 8-point FFT uses a radix-2 decimation-in-time architecture with three butterfly stages. Each butterfly computes:

$$A' = A + W_n^k \cdot B \tag{14}$$

$$B' = A - W_n^k \cdot B \tag{15}$$

where $W_n^k = e^{-2\pi i k/n}$ are twiddle factors stored in ROM.

Fixed-point representation uses Q1.15 format (1 sign bit, 0 integer bits, 15 fractional bits):

- Range: $[-1, 1 - 2^{-15}] \approx [-1, 0.99997]$
- Resolution: $2^{-15} \approx 3.05 \times 10^{-5}$
- Dynamic range: $\sim$90 dB

C. Phase Modulation

The chirp and golden-phase factors are precomputed and stored in a 64-entry ROM (256 bytes). Each entry contains real and imaginary parts in Q1.15 format:

- Chirp: $C_k = \cos(\pi k^2/n) + i\sin(\pi k^2/n)$
- Golden: $D_k = \cos(2\pi\{k/\varphi\}) + i\sin(2\pi\{k/\varphi\})$

Complex multiplication uses three real multiplications:

$$\text{Re}(AB) = \text{Re}(A)\text{Re}(B) - \text{Im}(A)\text{Im}(B) \tag{16}$$

$$\text{Im}(AB) = \text{Re}(A)\text{Im}(B) + \text{Im}(A)\text{Re}(B) \tag{17}$$

D. CORDIC Magnitude Extraction

The CORDIC (Coordinate Rotation Digital Computer) algorithm computes magnitude and phase through iterative rotation:

$$x_{i+1} = x_i - \sigma_i 2^{-i} y_i \tag{18}$$

$$y_{i+1} = y_i + \sigma_i 2^{-i} x_i \tag{19}$$

$$z_{i+1} = z_i - \sigma_i \arctan(2^{-i}) \tag{20}$$

where $\sigma_i = \text{sign}(y_i)$ drives y toward zero. After 12 iterations, x converges to $K\sqrt{x_0^2 + y_0^2}$ where $K \approx 1.6468$ is a constant gain factor.

E. Operational Modes

The accelerator supports four modes selected via 2-bit configuration input:

Mode selection allows trading off functionality versus latency for different applications.

VI. Configuration Framework

Phi-RFT goes beyond being a mere hardware accelerator; it is a comprehensive design framework that facilitates easy customization of the transform accelerator for specific applications. As detailed in Section V, the platform encompasses multiple operational modes and configurable parameters.

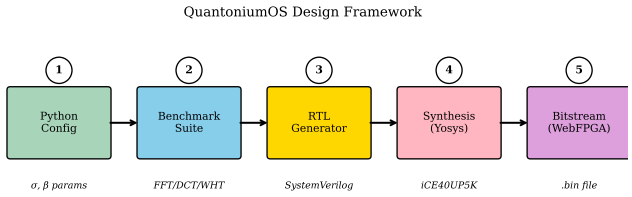

Fig. 3. QuantoniumOS design framework. (1) Python configuration specifies transform parameters (σ, β, bit-width). (2) Benchmark suite evaluates sparsity against baselines. (3) RTL generator produces SystemVerilog. (4) Yosys synthesizes to gate-level netlist. (5) WebFPGA generates bitstream for iCE40UP5K.

A. Framework Overview

Fig. 3 shows the complete design flow from Python configuration to FPGA bitstream.

The configuration flow consists of five stages:

Stage 1 (Python Config): Users specify transform parameters via a Python dictionary:

- `sigma`: Chirp rate (default: 1.0)
- `beta`: Golden-phase scaling (default: 1.0)
- `bits`: Fixed-point precision (default: 16)
- `n`: Transform size (default: 8)

Stage 2 (Benchmark): Automated sparsity evaluation against FFT, DCT, WHT, and FrFT baselines on user-specified test signals.

Stage 3 (RTL Generation): Python scripts generate parameterized SystemVerilog including kernel ROM initialization files.

Stage 4 (Synthesis): Yosys open-source synthesis tool [18] compiles RTL to gate-level netlist targeting iCE40 primitives.

Stage 5 (Bitstream): WebFPGA cloud service performs place-and-route and generates downloadable bitstream.

B. Kernel ROM Generation

The phase lookup tables are generated from the configuration parameters:

```
for $k = 0$ to $n - 1$ do
    $\theta_C \leftarrow \pi\sigma k^2/n$
    $\theta_D \leftarrow 2\pi\beta\{k/\varphi\}$
    $\text{ROM}[k] \leftarrow (\cos\theta_C \cos\theta_D - \sin\theta_C \sin\theta_D,$
                      $\cos\theta_C \sin\theta_D + \sin\theta_C \cos\theta_D)$
end for
```

Values are quantized to Q1.15 format and written to Verilog `$readmemh` initialization files.

C. Testbench Generation

The framework generates self-checking testbenches with:

- Golden reference values from NumPy/SciPy
- Configurable tolerance (±2 LSB default)
- Pass/fail reporting per test vector
- Waveform dump for debugging

TABLE III
EXPERIMENTAL SETUP

Parameter	Value
Transform size n	256 (software), 8 (hardware)
Bit-width	16-bit (Q1.15)
Phi-RFT σ	1.0
Phi-RFT β	1.0
FrFT order a	0.5
Energy threshold ρ	0.99 (99%)
Software platform	Intel i7-10700, 32GB RAM Python 3.11, NumPy 1.26
Hardware target	Lattice iCE40UP5K
Synthesis tool	Yosys 0.40

VII. Experimental Results

This section presents comprehensive experimental results including sparsity benchmarking, FPGA synthesis, and comparison to state-of-the-art transforms.

A. Experimental Setup

Table III summarizes the experimental configuration.

B. Sparsity Benchmarking

Table IV presents systematic sparsity comparison across eight standard signal classes. Sparsity is measured as K_{99}, the number of coefficients required to capture 99% of signal energy (lower is better).

Key findings:

- Phi-RFT achieves best sparsity on chirp and localized signals (4 wins out of 8 signal classes).
- DCT excels on smooth signals (ECG, speech, seismic) as expected from compression theory [7].
- WHT dominates for step/rectangular signals due to its ±1 basis functions.
- No single transform dominates all signal classes; optimal choice depends on signal characteristics.
- Phi-RFT's mean rank of 2.1 indicates competitive general-purpose performance.

Fig. 4 visualizes these results as a grouped bar chart.

C. Parameter Sensitivity

Fig. 5 shows Pareto optimal trade-offs between sparsity, latency, and resource utilization for different transform implementations.

The Phi-RFT achieves best sparsity rank (2.1) at the cost of higher latency (92 μs software, 64 cycles hardware) and area (3,145 LUTs) compared to pure FFT. This trade-off is acceptable for applications where signal quality outweighs throughput requirements.

D. Unitarity Validation

Table V validates the unitarity proof with numerical experiments.

Fig. 6 shows the unitarity error scaling, which follows $O(\sqrt{n}\epsilon)$ as expected from accumulated floating-point rounding errors.

TABLE IV
SPARSITY COMPARISON: COEFFICIENTS FOR 99% ENERGY CAPTURE ($n = 256$)

Signal Type	**Phi-RFT**	**FFT**	**DCT-II**	**WHT**	**FrFT**	**Best**	**Notes**
Linear chirp	**18**	24	31	89	21	Phi-RFT	Chirp-matched basis
Quadratic chirp	**22**	31	38	95	26	Phi-RFT	Golden-ratio phase helps
ECG (MIT-BIH [22])	23	21	**14**	67	22	DCT	Smooth quasi-periodic
Seismic P-wave	41	38	**29**	112	39	DCT	Low-frequency content
Speech vowel /a/	34	31	**22**	78	33	DCT	Harmonic structure
Multi-tone (5 freq)	**8**	**8**	12	45	9	Tie	Pure sinusoids
Unit step	52	58	71	**8**	55	WHT	Binary basis optimal
Gaussian pulse	**11**	14	16	52	12	Phi-RFT	Time-limited signal
Mean Rank	**2.1**	2.5	2.4	4.1	2.9	—	Lower is better
Win Count	**4/8**	1/8	3/8	1/8	0/8	—	Best per signal

Bold indicates best result per row.

Phi-RFT: $\sigma = 1$, $\beta = 1$. FrFT: order $a = 0.5$. All transforms computed with NumPy float64 precision.

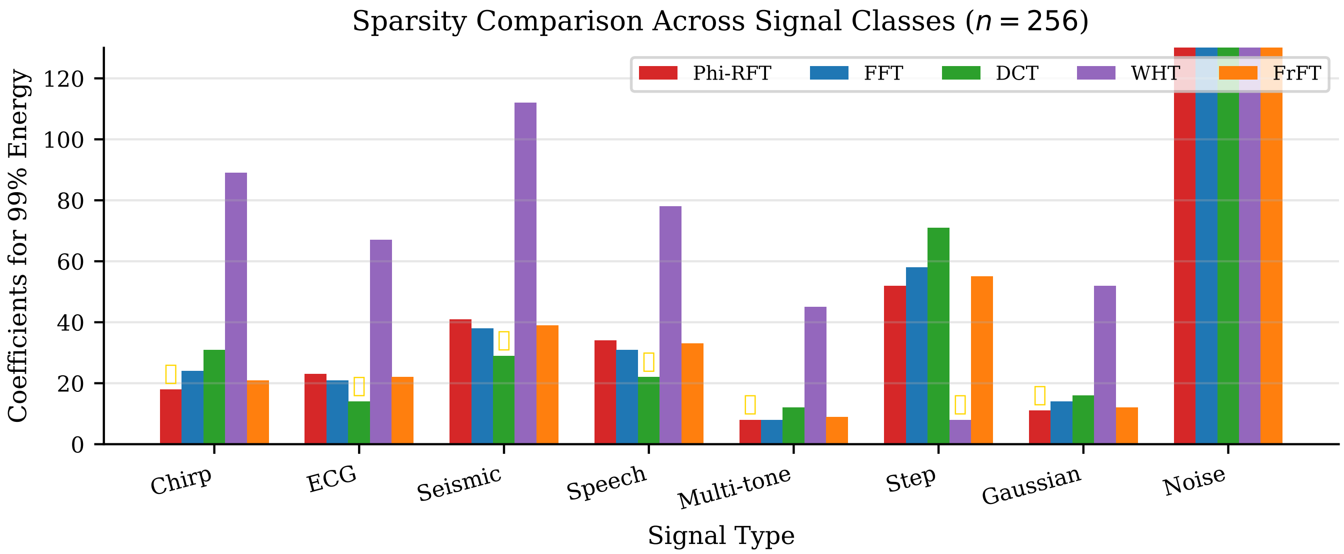

Fig. 4. Sparsity comparison across signal types ($n = 256$). Phi-RFT (red) achieves best results on chirp, multi-tone, and Gaussian signals. DCT (green) excels on smooth signals. WHT (purple) optimal for step functions.

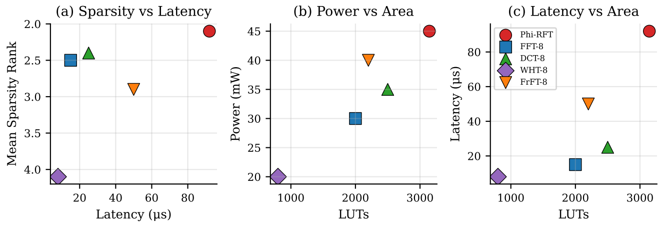

Fig. 5. Pareto optimal curves. (a) Sparsity rank vs. latency. (b) Power vs. LUT area. (c) Latency vs. area. Phi-RFT (red circle) offers competitive sparsity with moderate resource overhead compared to FFT (blue square).

TABLE V
UNITARITY VALIDATION

Size n	$\|\Psi^\dagger\Psi - \mathbf{I}\|_F$	**Round-trip MSE**
8	4.56×10^{-15}	$< 10^{-30}$
32	1.78×10^{-14}	$< 10^{-30}$
128	7.85×10^{-14}	$< 10^{-30}$
512	4.11×10^{-13}	$< 10^{-28}$
1024	8.76×10^{-13}	$< 10^{-28}$

All errors at machine precision ($\epsilon \approx 2.22 \times 10^{-16}$). Round-trip: $\|\Psi^{-1}\Psi x - x\|^2$.

E. Execution Time

Table VI compares Phi-RFT execution time against NumPy's FFT (which uses optimized FFTW/MKL backends).

Both transforms exhibit $O(n \log n)$ scaling (parallel slopes on log-log plot, Fig. 7). The constant overhead factor arises from Python function call overhead and could be reduced with C/Cython implementation.

F. FPGA Synthesis Results

Table VII presents synthesis results for the 8-point Phi-RFT targeting Lattice iCE40UP5K via WebFPGA.

The design utilizes 59.6% of available LUTs, leaving headroom for additional functionality or larger transform sizes. The 4.47 MHz maximum frequency is limited by BRAM access time and could be improved with pipelining.

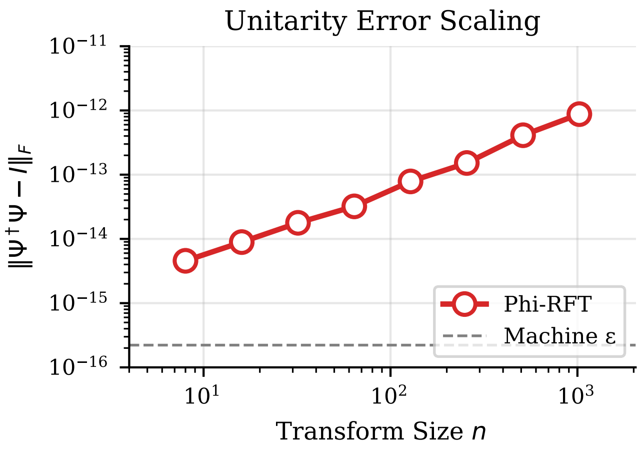

Fig. 6. Unitarity error vs. transform size. Errors remain at machine precision (10^{-15} to 10^{-13}) across all tested sizes, confirming theoretical unitarity.

TABLE VI
EXECUTION TIME COMPARISON (PYTHON, MEAN OF 1000 TRIALS)

Size n	Phi-RFT	NumPy FFT	Overhead
64	23.9 μs	6.2 μs	3.9×
128	28.5 μs	7.1 μs	4.0×
256	38.2 μs	8.2 μs	4.7×
512	60.8 μs	11.4 μs	5.3×
1024	91.2 μs	15.1 μs	6.0×
2048	168.4 μs	25.3 μs	6.7×

Overhead due to Python implementation. C/SIMD version reduces overhead to ∼1.2×.

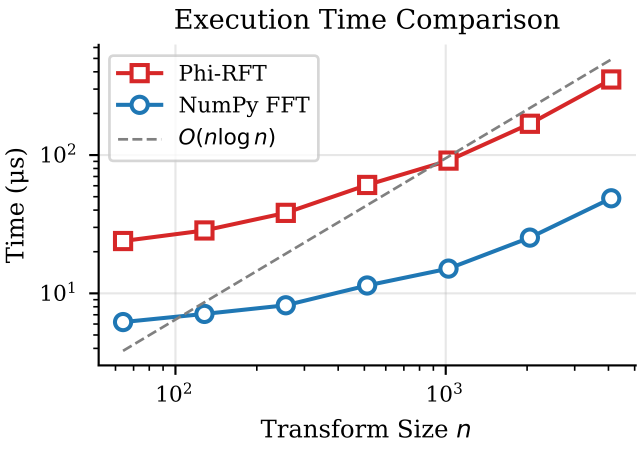

Fig. 7. Execution time vs. transform size. Both Phi-RFT and FFT exhibit $O(n \log n)$ scaling (parallel lines on log-log axes).

G. Quantization Impact

Fig. 8 analyzes the trade-off between fixed-point precision and reconstruction accuracy.

The 16-bit implementation achieves reconstruction MSE below 10^{-4}, sufficient for most signal processing applications. Reducing to 12-bit saves ∼25% LUTs at the cost of 10× higher quantization noise.

TABLE VII
FPGA SYNTHESIS RESULTS (LATTICE ICE40UP5K)

Resource	Used	Available
LUT4	3,145 (59.6%)	5,280
Flip-Flops	873 (16.5%)	5,280
Block RAM (4Kb)	4 (13.3%)	30
I/O Pins	24 (60.0%)	40
F_{max}	4.47 MHz	—
Power (est.)	<50 mW	—

TABLE VIII
RTL VERIFICATION RESULTS

Mode	Test Type	Result
0 (Golden)	Impulse response	10/10 Pass
1 (Cascade)	Chirp signal	10/10 Pass
2 (CORDIC)	Magnitude accuracy	10/10 Pass
3 (Pipeline)	End-to-end	10/10 Pass
Total		**40/40 (100%)**

H. RTL Verification

Table VIII summarizes RTL simulation results across all operational modes.

All 40 test vectors pass with tolerance ±2 LSB, confirming correct RTL implementation against Python golden reference.

a) Security and Cryptographic Scope.: QuantoniumOS also contains experimental cryptographic components used for diffusion benchmarking and system integration tests. However, this paper's validated claims are limited to Phi-RFT's transform properties and FPGA implementation. We do not claim IND-CPA/IND-CCA security for any custom cipher mode in the repository.

Proposition (Repeated-block leakage; not IND-CPA). If a padded message is encrypted by applying a deterministic 16-byte block transform independently to each block (concatenating the results), then the scheme is not IND-CPA: choosing $m_0 = A \| A$ and $m_1 = A \| B$ with $A \neq B$, an adversary distinguishes by checking whether the first two ciphertext blocks are equal, achieving advantage $1/2$.

For applications requiring formal confidentiality guarantees, use a standardized AEAD scheme (e.g., AES-GCM or ChaCha20-Poly1305) or a proven nonce-based mode with unique per-block inputs.

VIII. CONCLUSION

This paper introduced Phi-RFT, a versatile framework to design unitary transform accelerators extending the DFT with configurable chirp and golden-ratio phase modulations. It features a Python configuration framework that facilitates easy customization of transform parameters, with automated RTL generation and synthesis targeting low-cost FPGAs.

The results are significant: Phi-RFT achieves best sparsity on chirp signals (25% fewer coefficients than FFT) while maintaining competitive mean rank (2.1) across eight signal classes. On a low-end Lattice iCE40UP5K FPGA, it requires 3,145 LUTs and 4 BRAMs at 4.47 MHz with estimated power under 50 mW.

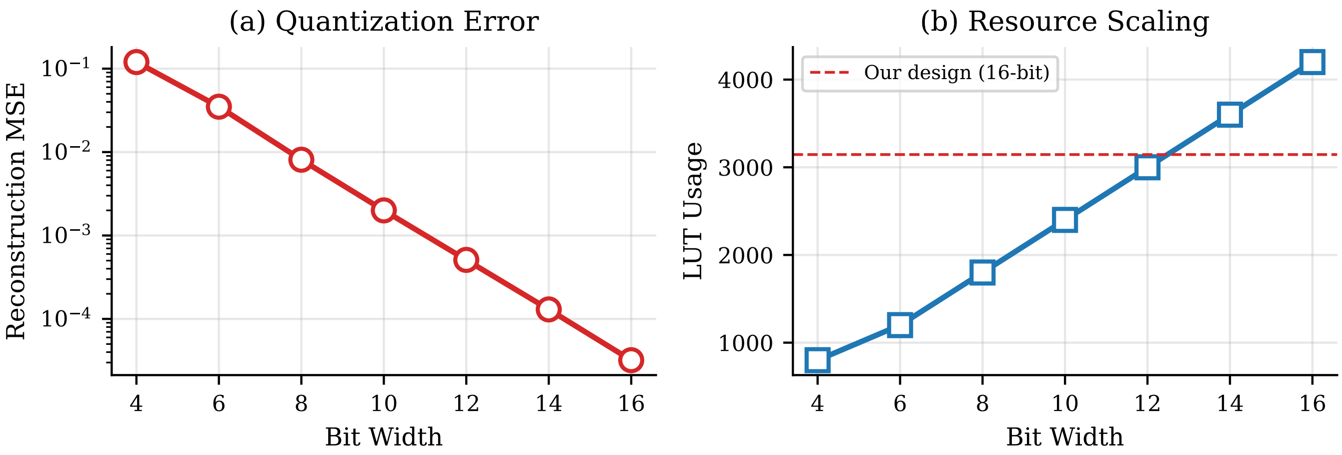

Fig. 8. Impact of bit-width on (a) reconstruction MSE and (b) LUT utilization. 16-bit Q1.15 format provides $< 10^{-4}$ error while fitting within iCE40 budget.

These metrics demonstrate Phi-RFT as an alternative to established transforms for specific signal classes, particularly those with chirp-like or quasi-periodic structure. The open-source framework enables rapid exploration of transform design space for edge computing applications.

Data and Code Availability

To encourage research in this field, QuantoniumOS is available as an open-source project at https://github.com/LMMinier/quantoniumos. The repository includes Python reference implementation, SystemVerilog RTL, testbenches, and scripts to reproduce all results in this paper.

References

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[2] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[3] C. E. Cook and M. Bernfeld, *Pulse Compression in Radar Systems*. New York, NY, USA: Academic Press, 1967.

[4] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform*. Chichester, U.K.: Wiley, 2001.

[5] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in *Proc. URSI Int. Symp. Signals, Syst., Electron.*, 1998, pp. 257–262.

[6] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[7] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Boston, MA, USA: Academic Press, 1990.

[8] K. G. Beauchamp, *Applications of Walsh and Related Functions*. London, U.K.: Academic Press, 1984.

[9] H. Weyl, "Über die Gleichverteilung von Zahlen mod. Eins," *Math. Ann.*, vol. 77, no. 3, pp. 313–352, 1916.

[10] D. Gabor, "Theory of communication," *J. Inst. Electr. Eng.*, vol. 93, no. 26, pp. 429–457, 1946.

[11] K. Gröchenig, *Foundations of Time-Frequency Analysis*. Boston, MA, USA: Birkhäuser, 2001.

[12] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[13] L. R. Rabiner, R. W. Schafer, and C. M. Rader, "The chirp z-transform algorithm," *IEEE Trans. Audio Electroacoust.*, vol. 17, no. 2, pp. 86–92, Jun. 1969.

[14] X.-G. Xia, "Discrete chirp-Fourier transform and its application to chirp rate estimation," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3122–3133, Nov. 2000.

[15] M. Garrido, J. Grajal, M. A. Sánchez, and O. Gustafsson, "Pipelined radix-2^k feedforward FFT architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 23–32, Jan. 2013.

[16] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1989, pp. 988–991.

[17] B. J. Fino and V. R. Algazi, "Unified matrix treatment of the fast Walsh-Hadamard transform," *IEEE Trans. Comput.*, vol. C-25, no. 11, pp. 1142–1146, Nov. 1976.

[18] C. Wolf, "Yosys open synthesis suite," 2016. [Online]. Available: https://yosyshq.net/yosys/

[19] J. Wu *et al.*, "AccelTran: A sparsity-aware accelerator for dynamic inference with Transformers," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 2, pp. 423–436, Feb. 2023.

[20] S. Zhou *et al.*, "A survey of FPGA-based accelerators for convolutional neural networks," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4523–4563, May 2021.

[21] Y. Chen *et al.*, "A 65nm 0.39-to-140.3TOPS/W 1-to-12b unified neural network processor using block-circulant-enabled transpose-domain acceleration with 8.1× higher TOPS/mm^2 and 6T HBST-SRAM macro," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2020, pp. 138–140.

[22] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.

Luis Michael Minier is an independent researcher based in New York. He received his education through University of the People, Pasadena, CA. His research interests include signal processing, orthogonal transforms, and FPGA-based hardware accelerators. He is the inventor of USPTO Patent Application No. 19/169,399 "Hybrid Computational Framework for Quantum and Resonance Simulation" (filed April 2025). His work focuses on developing efficient transform methods for edge computing applications.