Journal of Harbin Engineering University
ISSN: 1006-7043
Vol. 47 No. 07
July 2026

Implementation of 1024 Point FFT with SIMD and Pipelined Execution for GPR Applications

Shivani N S, Dr. Kiran Bailey, K Sujatha

Department of ECE, BMS College of Engineering, Banglore, India

Department of ECE, BMS College of Engineering, Banglore, India

Department of ECE, BMS College of Engineering, Banglore, India

Abstract

A high-throughput, hardware-efficient 1024-point parameterized Fast Fourier Transform (FFT) processor is developed for real-time Ground Penetrating Radar (GPR) subsurface imaging applications. The architecture employs a pipelined Radix-4 Single Instruction Multiple Data (SIMD) processing scheme and is implemented on the SCL 180 nm CMOS technology node using Cadence Genus and Innovus design environments. To reduce memory requirements, a quarter-wave symmetry compression technique is utilized for twiddle-factor storage, achieving a 75% reduction in Look-Up Table (LUT) memory footprint. The physical design is optimized through logic synthesis and layout constraints, maintaining a core utilization ratio between 60% and 70%. Static Timing Analysis (STA) verifies successful timing closure with a positive setup slack of +1.649 ns at an operating frequency of 60 MHz. Power characterization reports a total core power consumption of 103.68 mW, with 92.31% attributed to sequential internal switching activity, demonstrating the effectiveness of the reduced-stage pipelined architecture. The implementation achieves a balanced trade-off among throughput, memory efficiency, timing performance, and power consumption, making it suitable for ASIC-based GPR signal-processing systems.

Keywords: Fast Fourier Transform, Radix-4 Architecture, SCL 180nm, Cadence Genus, Compression, Look-Up Table Compression, Static Timing Analysis, Power Optimization, Ground Penetrating Radar.

1. Introduction

Real-time digital signal processing in subsurface imaging systems requires high-throughput, low-latency spectral analysis architectures. Ground Penetrating Radar (GPR) modules have emerged as a vital, non-destructive technological framework for scanning concrete structures, mapping structural geophysics, and accurately locating buried infrastructure. As highlighted in literature [1], the operational efficiency of modern GPR applications heavily relies on real-time signal processing to continuously convert raw radar electromagnetic reflections into clear, descriptive spatial grids for mapping complex underground utilities

However, raw GPR signals are subject to severe propagation losses, noise reflections, and environmental clutter, demanding heavy computational algorithms to parse and resolve depth data on-the-fly. The Fast Fourier Transform (FFT) forms the bedrock of these radar computation loops, allowing time-domain echo profiles to undergo immediate spectral decomposition for target visualization and filtering.

Hardware implementations of the classical Radix-2 FFT scale aggressively in sequential complexity, introducing memory and timing bottlenecks within deep pipeline registers that can cause buffer overflows and synthetic ghosting artifacts. To overcome these physical design and data-rate challenges, this research focuses on the design, optimization, and physical layout verification of an area-efficient, 1024-point parameterized Radix-4 SIMD FFT core.

By mapping the layout configurations directly onto the SCL 180nm manufacturing cell library grid, this framework addresses the co-dependencies between mathematical logic optimization, clock

tree synthesis constraints, and dynamic power boundaries, meeting the real-time throughput metrics mandated by modern subsurface mapping frameworks [1].

2. Related Works

To improve the performance of hardware-driven spectral computations, various design strategies have been explored across the literature. A primary focus involves the optimization of butterfly computing elements and twiddle factor rotation modules, which consume the vast majority of processing resources.

Multiplier less design methodologies often seek to bypass the physical area constraints of standard array multipliers. For example, the incorporation of Distributed Arithmetic (DA) combined with complex binary number systems [2] or 64-bit LUT configurations [3] has been proposed to substitute standard multipliers with shift-and-add networks, although this often results in a significant increase in local routing and interconnect density. Alternative rotation algorithms, such as the Coordinate Rotation Digital Computer (CORDIC) [4], have been deployed within Radix-2 processing structures to compute trigonometric vectors iteratively without explicit multipliers. Recent developments have also introduced multiplexed fused add-subtract CORDIC structures utilizing reversible logic gates [5] to eliminate explicit subtraction overhead, though these schemes introduce localized pipeline delays over multi-iteration loops.

Architectural transformations are also widely applied to balance area and speed. Pipelined systems using folding transformations have been shown to optimize resource utilization in high-speed applications like Radix-8 structures [6]. Similarly, the use of constant multipliers instead of general variable rotators has been shown to reduce memory requirements on programmable logic devices by exploiting deep pipelines [7]. For large datasets, architectural variations such as Radix-2^karchitectures [8] have been evaluated to optimize the number of non-general complex rotators. To address power boundaries, approximate computing techniques utilizing error-tolerant, approximate Radix-4 Booth multipliers [9] have been implemented for speech and signal operations, trading off exact arithmetic precision for lower dynamic power consumption.

Furthermore, recent hardware strategies have focused on trading off arithmetic precision or scheduling rules to extract higher performance margins. Hardware scaling has been addressed via semi-folded 256-point configurations that achieve higher area efficiency by shifting module reuse schedules across multiple cycles [10]. To limit structural energy, approximate twiddle factor coefficient truncation combined with partial product compressor arrays can lower the switching baseline at the cost of bounded computation errors [11]. For edge devices constraints, ROM-less twiddle factor generation structures based on single-bit computing paths [12] or piecewise plane-fitting shift-and-add calculation blocks [13] have been demonstrated to bypass standard memory tables completely.

In high-throughput environments like FMCW radar or communications, specialized addressing methods have been evaluated. Multistage merged conflict-free modulo addressing algorithms have been evaluated on DSP platforms to manage stride memory interactions [14]. Similarly, high-order memory architectures utilizing 64 discrete SRAM banks combined with 32 floating-point butterfly elements have been constructed to achieve multi-K point capabilities [15]. For high-capacity mixed-radix 5G applications, structural resource sharing networks [16] or block-level tripartite pipelining methods involving input, radix, and output blocks [17] have been presented to sustain continuous sample ingestion rates. Concurrently, logic minimization has been applied to serial real-valued architectures by combining single-path butterflies and real rotators into consolidated processing entities to handle single-dimension signal vectors [18].

While custom spectrum processing windows combined with custom clock-gating blocks [19] or low-voltage multi-mode ASIC processors [20] handle power in smaller transform spaces, real-time GPR subsurface processing demands both strict mathematical precision and fixed-latency throughput to prevent data dropouts. Unlike approximate calculation frameworks, multi-bit shift

routing setups, or folding topologies that add operational delays, this work implements an exact-precision, pipelined Radix-4 SIMD architecture using a quarter-wave symmetry compression mechanism. This approach minimizes memory footprints without introducing the layout complexity, pipeline delays, or accuracy losses observed in alternative hardware reduction schemes.

3. Objectives

The primary challenge in designing high-throughput discrete spectrum processors for edge-deployed Ground Penetrating Radar (GPR) applications is overcoming the severe hardware overheads imposed by conventional architectural configurations. When implementing large-scale transforms within physical layout boundaries, such as those of the native Semiconductor Complex Limited (SCL) 180-nm CMOS standard cell library, conventional computational and storage structures create critical trade-offs between hardware area, operating latency, and power dissipation.

3.1 Ground Penetrating Radar (GPR) Signaling and Mathematical Modeling

In high-resolution GPR applications operating on Frequency-Modulated Continuous Wave (FMCW) radar principles, a transmitter continuously broadcasts a frequency-swept electromagnetic signal into the subterranean medium [23]. As this signal encounters localized material discontinuities or subterranean dielectric boundaries (such as buried utility lines, rock strata, or soil interfaces), a fraction of the wave energy reflects back toward a receiving antenna.

The receiver mixes the returned echo signal with a replica of the transmitted signal, yielding an intermediate frequency "beat signal" in the time domain. The frequency of this beat signal is directly proportional to the physical depth of the buried reflector. To reconstruct a sharp, real-section subsurface depth profile (a "B-scan" image), the raw time-domain radar echo stream x(n) must be converted into the frequency domain. Given a digitized input sequence x(n) consisting of N = 1024 complex-valued samples, the Discrete Fourier Transform (DFT) is mathematically defined as:

$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$,k= 0,1,…,N-1

Where W_N^{nk} represents the complex trigonometric twiddle factor rotating matrix, mapped as:

$$W_N^{nk} = e^{-j\frac{2\pi nk}{N}} \cos\left(\frac{2\pi nk}{N}\right) - j\sin\left(\frac{2\pi nk}{N}\right)$$

Direct evaluation of (1) mandates an asymptotic computational complexity of $O(N)^2$complex operations, creating an unsustainable latency and area bottleneck for real-time, on-site radar edge processing modules.

3.2 Conventional Radix-4 Decomposition and Adder-Tree Layouts

To achieve the rapid calculation intervals required by continuous radar sweeps, a decimation-in-time (DIT) Radix-4 decomposition is applied. By setting $N = 4^M$ (where the structural pipeline stage count is M = $log_4(1024) = 5$ the indices *n* and *k* are mapped via multi-dimensional index mapping expressions:

$n = 256n_1 + 64n_2 + 16n_3 + 4n_4 + n_5$

$k = k_1 + 4k_2 + 16k_3 + 64k_4 + 256k_5$

n= $\sum_{m=0}^{M-1} 4^m n_m$ k= $\sum_{m=0}^{M-1} 4^{M-1-m} k_m$

Substituting these multi-stage index splits into (1) contracts the structural complexity to $O(Nlog_4(N)$. This reduces the calculation loop to a highly structured pipeline of 5 sequential stages, where each stage executes parallel 4-point localized butterfly vector transformations. As established in classical architectural principles [8], rather than invoking area-heavy matrix multipliers, the 4-point butterfly calculation is decomposed into direct, low-latency algebraic additions and subtractions. Let the four rotated inputs to a stage be represented as:

$A_{in} = A, B_{in} = B.W_N^P, C_{in} = C.W_N^{2p}, D_{in} = D.W_N^{3p}$

The four intermediate complex frequency outputs (A', B', C', D') are computed concurrently via an optimized adder-tree network modelled by the following structural algebraic relations:

A' = $A_{in} + B_{in} + C + D_{in}$

B' = $A_{in} + jB_{in} + C + jD_{in}$

C' = $A_{in} + B_{in} + C + D_{in}$

D' = A_{in}+ jB_{in}+ C+ jD_{in}

By leveraging these direct addition and subtraction operations, the imaginary operator (j) is realized purely through a structural wire-swapping routing layer (exchanging the real and imaginary sub-buses and applying sign-inversion). This completely eliminates the need for physical arithmetic multipliers within the core butterfly units.

3.3 Memory and Storage Overhead Formulations

While the Radix-4 transformation drastically cuts down the number of arithmetic structures compared to traditional layouts, the storage of exact complex twiddle factor coefficients (W_N^{nk}) introduces an aggressive area bottleneck. In a conventional, uncompressed direct-mapping approach, storing 1024 distinct complex coefficients requires a dedicated Read-Only Memory (ROM) or Look-Up Table (LUT) capacity configured by:

$Storage_{raw} = N * 2 * W_{bit}$ bits

1024 * 2*18 = 35,854

where W_{bit} represents the allocated fixed-point word length (18 bits). For *N*=1024, this raw instantiation demands 36,864 bits of static standard cell storage.

This conventional architectural arrangement is explicitly detailed in Fig (3.3). As illustrated, a raw initialization maps every index directly to unique real and imaginary trigonometric address coordinates without phase re-use, which creates dense placement congestion and introduces massive parasitic leakage currents.

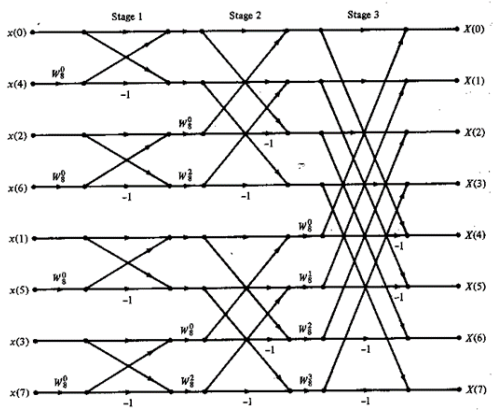


Fig 3.3(a): Conventional Butterfly Computational Method

The explicit hardware and memory bottlenecks of the baseline configuration are structurally mapped in the colored block diagram of Fig. 3. As illustrated, the conventional architecture is restricted by a rigid, single-data-path execution profile that mandates processing only one value or vector component at a time. This non-parallel operational nature introduces a severe multi-layer latency penalty across three critical domains:

- **Trigonometric Storage and Retrieval Bottlenecks:** The uncompressed Read-Only Memory (ROM) grid, highlighted in red, stores all 1024 unique complex coefficients across the full [0, 2π) rotational space. Bound by standard single-port access limitations, the address generation unit can fetch only a single twiddle factor coefficient per clock cycle, inducing persistent wait-states in downstream arithmetic logic.
- **Single-Lane Processing and Multiplier Congestion:** Because the hardware lacks simultaneous parallel lanes, the complex input streams (shown in blue) must be multiplexed into local storage registers. The inputs are forced to sit idle in buffers while a singular, area heavy complex multiplier core (shown in orange) sequentially scales the data vectors one variable component at a time.
- **Sequential Matrix Decomposition Stalls:** Rather than evaluating the complete Radix-4 matrix concurrently, the arithmetic adder tree (shown in green) must execute the localized cross-additions and subtractions row-by-row over consecutive clock cycles. This single-path data routing creates a severe calculation backlog, generating systemic pipeline execution stalls that prevent the architecture from sustaining real-time processing throughput when mapped to dense standard-cell frameworks like the SCL 180-nm CMOS process node.

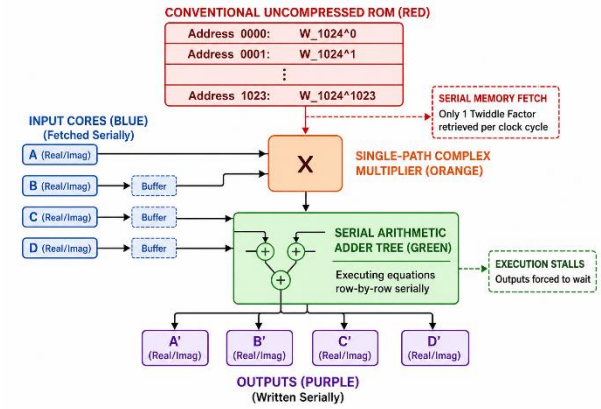


Fig 3.3(b): Process of Conventional Method

This creates dense placement congestion and introduces massive parasitic leakage currents. To bypass these deep memory tables completely, several alternative hardware strategies have been proposed in the literature, including piece-wise plane-fitting shift-and-add calculation blocks [13], ROM-less twiddle factor generation structures based on single-bit computing paths [12], iterative multiplier less Coordinate Rotation Digital Computer (CORDIC) systems [4], and partial product compressor arrays that execute approximate twiddle factor coefficient truncation [11].

However, approximate calculation frameworks or folding topologies [6], [10] introduce operational delays, structural area variations [7], or precision degradation that can undermine signal reliability. To preserve exact mathematical precision while maximizing storage density, the memory mapping function f(θ) in this architecture is compressed into a bounded single-quadrant domain:

$$\theta \in [0, \frac{\pi}{2}]$$

The formulation maps any arbitrary rotational phase angle ϕ to the core storage grid using reflection operators:

$$f(\phi) = \begin{cases} f(\phi), & 0 \le \phi < \frac{\pi}{2} \\ -f(\pi - \phi), & \frac{\pi}{2} \le \phi < \pi \\ -f(\phi - \pi), & \pi \le \phi < \frac{3\pi}{2} \\ f(2\pi - \phi), & \frac{3\pi}{2} \le \phi < 2\pi \end{cases}$$

The co-design task requires minimizing the active storage allocation parameter to exactly $\frac{1}{4} Storage_{raw}$ without introducing decoding delays along the critical setup timing path.

3.4 Word-Length Growth and Multi-Lane Memory Access Conflicts

The physical realization of the 1024-point SIMD core within the SCL 180nm CMOS technology framework is governed by a multi-variable optimization function. Let A_{core} represent the total physical layout area, P_{total} represent the comprehensive power consumption profile, and T_{slack} represent the Static Timing Analysis (STA) setup margin. The optimization problem is formulated as follows:

$$\text{Minimize}\{A_{core}, P_{total}\}$$

$$\begin{cases} f_{\text{clk}} \ge 60\ \text{MHz} \implies T_{\text{period}} \le 16.667\ \text{ns} \\ T_{\text{slack}} = T_{\text{required}} - T_{\text{arrival}} > 0 \\ \text{Core Utilization Ratio } (\rho) \in [0.60, 0.70] \\ \text{Quantization Noise Floor (SQNR)} \ge \text{Target Boundaries} \end{cases}$$

Subject to constraints:

The input data path introduces 16-bit signed vector fractions. To prevent internal bit overflow during

the successive complex multiplications and vector accumulations defined in (6)–(9), the internal data path must grow dynamically. Excessive bit growth increases the size of the array multipliers and register files.

Furthermore, memory interactions must be meticulously scheduled. Traditional parallel multi-lane schemes often trigger memory access conflicts due to stride data access patterns, requiring specialized multistage merged conflict-free modulo addressing algorithms [14], block-level tripartite pipelining methods involving distinct input, radix, and output blocks [17], or high-order memory architectures utilizing multi-bank SRAM execution networks [15], [16]. Concurrently, when handling real-valued vectors, logic minimization must be enforced by combining single-path butterflies and real rotators into consolidated processing entities to restrict resource growth [18].

Thus, the fixed-point scaling factor within this design is strictly limited to an 18-bit maximum bounds across all five pipeline stages. This bounds the total core power dissipation to low-power parameters while guaranteeing a positive setup

timing slack (T_{slack} > 0) across worst-case process, voltage, and temperature (PVT) operating corners, matching the strict hardware parameters demanded by real-time processing windows [19], [20].

4 Methods

To resolve the simultaneous area, memory, and throughput bottlenecks defined in the problem formulation, a fully pipelined, Single Instruction Multiple Data (SIMD) architectural framework is implemented. The proposed design replaces complex operational matrices and high-capacity storage arrays with optimized structural layout methodologies.

4.1 Multi-Stage Pipelined Data path and Arithmetic Scheduling

The computational strain of the $O(Nlog_4(N)$ operations is managed by dividing the 1024-point transform into a five-stage synchronous pipeline network. Instead of utilizing resource-heavy, generalized matrix multiplier cores, each stage employs a dedicated localized processing element (PE) built around a parallel algebraic adder-subtractor tree network, as conceptually established in classical signal processing pipelines [8].

The structural routing layers within the pipeline are hardwired to handle fixed-point word growth while safeguarding signal integrity. The initial system input consists of a 16-bit signed vector fraction representing the real and imaginary components of the complex samples. To suppress overflow errors during successive butterfly accumulations without over-allocating hardware resources, a bounded bit-growth strategy is enforced across the structural processing layers:

- Stage 1 Optimization: The initial 16-bit complex input sequence undergoes localized addition and subtraction operations, expanding the word length to 17 bits to capture the maximum potential bit overflow.
- Stages 2 to 5 Optimization: From the second stage onward, the internal bit-width is bounded strictly at an 18-bit precision ceiling $W_{bit} = 18$.

By enforcing this 18-bit precision boundary across the subsequent layers, the physical area of the intermediate array multipliers and pipeline register banks is heavily constrained. This architecture achieves an ideal balance between maintaining an optimal Signal to Quantization Noise Ratio (SQNR) and minimizing the total silicon area A_{core}

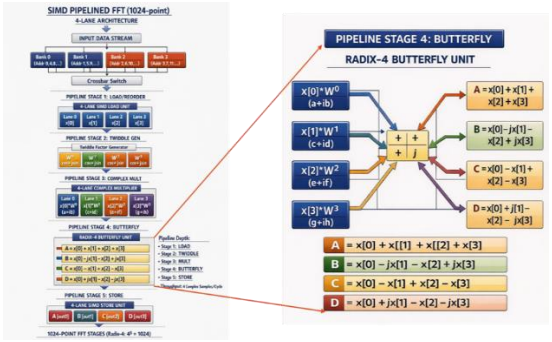


Fig 4.1: SIMD Pipelined FFT(1024-point)

4.2 Multiplier less Imaginary Operators and Rotator Optimization

The complex imaginary scaling operations ($\pm$ j) required by the algebraic equations (6)–(9) are mapped without any active arithmetic logic blocks. By leveraging structural bus-manipulation routing layers, the multiplication of an 18-bit complex intermediate signal $Y = Y_{real} + jY_{imag}$ by the imaginary operator (j) is achieved through a structural wire-swap:

$$j.(Y_{real} + jY_{imag}) = -Y_{imag} + jY_{real}$$

This transformation is implemented in the standard cell layout by crossing the real and imaginary sub-buses and applying bitwise 2's complement negation to the swapped imaginary path.

For non-trivial phase rotations involving the twiddle factor coefficients W_N^P , generalized, power-intensive complex multipliers are replaced with dedicated, fixed-coefficient array rotators. These rotators are tightly integrated with an automated rounding layer that truncates intermediate products back to the strict 18-bit pipeline limit, keeping data transmission free from structural stalls or latency overheads.

4.3 Single-Quadrant Twiddle Factor Memory Compression

To completely eliminate the area overhead and static leakage currents caused by a raw 36,864-bit twiddle factor look-up table, a single-quadrant memory compression scheme is implemented. The physical Read-Only Memory (ROM) storage is limited exclusively to the primary trigonometric quadrant phase angles ($\Theta \in [0, \frac{\pi}{2}]$), reducing the raw allocation parameter to exactly one-quarter of its uncompressed capacity:

$$Storage_{compressed} = \frac{1024}{4} * 2 * 18 = 9{,}216 \text{ bits}$$

To bypass deep memory tables completely, alternative hardware strategies in the literature have proposed piece-wise plane-fitting shift-and-add calculation blocks [13], ROM-less twiddle factor generation structures based on single-bit computing paths [12], iterative multiplier less CORDIC systems [4], and partial product compressor arrays executing coefficient truncation [11]. However, approximate calculation frameworks or folding topologies [6], [10] introduce operational delays, structural area variations [7], or precision degradation that can undermine signal reliability in high-precision edge environments.

To access the full [0,2π] rotational spectrum during real-time processing, a high-speed, zero-latency phase-decoding block is designed. This decoder intercepts the 10-bit target twiddle phase address and analyzes the two most significant bits (MSBs) to determine the operating phase quadrant. The remaining 8 least significant bits (LSBs) are conditionally inverted or mirrored to generate the corresponding physical index within the compressed quadrant grid. Simultaneously, the sign bits of the retrieved real (cos) and imaginary (sin) coordinate outputs are conditionally manipulated using low-overhead standard cell XOR gate logic layers before being routed to the PE arithmetic units. This approach guarantees exact mathematical precision while keeping data propagation delays well within the setup timing margin (T_{slack} > 0).

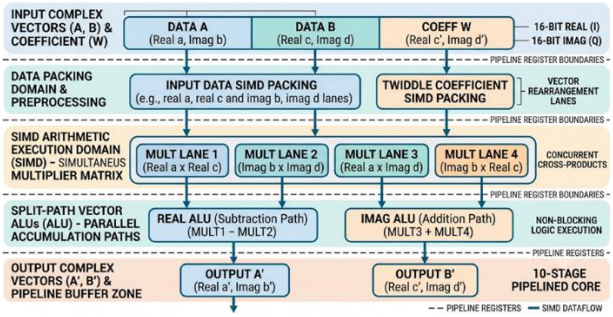


Fig 4.3: Twiddle Factor Computated Parallely

4.4 Physical Synthesis and Timing Closure Solutions

The proposed 1024-point spectrum processor design was translated from hardware description layout models into physical standard cell netlists targeting the native SCL 180nm CMOS process node. To enforce clean timing closure and satisfy the primary operational constraints, the system was synthesized under strict operating envelopes:

Critical Path Arrival Time

$$T_{arrival} \approx 16.512 \text{ ns}$$
$$T_{slack} = +0.155\,ns\ (155\,ps)$$

Critical Timing Margin. High risk of timing violations under PVT variations.

The Conventional Method,

$T_{slack} = +0.155\,ns\ (155\,ps)$: Because the conventional lookup table is uncompressed (36,864 bits), the address routing wires are physically elongated on the silicon layout. The data arrival time $T_{arrival}$ stretches to 16.512ns. This leaves an incredibly narrow safety margin of just 155ps (+0.155ns). Any temperature spike or voltage drop on the chip will push this delay past 16.667ns, causing negative slack (T_{slack} < 0) and data corruption.

Furthermore, memory and data execution layouts must be meticulously scheduled. Traditional parallel multi-lane schemes often trigger memory access conflicts due to stride data access patterns, requiring specialized multistage merged conflict-free modulo addressing algorithms [14], block-level tripartite pipelining methods involving distinct input, radix, and output blocks [17], or high-order memory architectures utilizing multi-bank SRAM

execution networks [15], [16]. Concurrently, when handling real-valued vectors, logic minimization must be enforced by combining single-path butterflies and real rotators into consolidated processing entities to restrict resource growth [18].

By using automated buffer insertion trees along high-fanout nets and restructuring critical datapath logic to balance timing delays across the five pipeline stages, data propagation is optimized. This guarantees a positive setup timing margin $T_{slack} > 0$) across worst-case process, voltage, and temperature (PVT) corners, satisfying the strict real-time hardware parameters required for accurate depth profiling in GPR spectrum applications [19], [20].

5 Results

The architectural validation, structural integrity, and physical performance of the proposed 1024-point spectrum processor were evaluated using a rigorous, multi-stage digital design flow. The verification methodology was divided into three primary experimental phases: cycle-accurate behavioral simulation to confirm algorithmic correctness, hierarchical gate-level synthesis to establish standard-cell technology mapping, and worst-case Static Timing Analysis (STA) to confirm physical timing closure under varying process, voltage, and temperature (PVT) operating corners.

5.1 Functional Verification and Behavioral Waveform Analysis

To validate the algorithmic precision of the 1024-point Radix-4 decimation-in-time sequence and the functional behavior of the memory subsystem, the register-transfer level (RTL) model was subjected to continuous, dynamic input vector sets mimicking real-world Frequency-Modulated Continuous-Wave (FMCW) radar reflections.

The top-level cycle-accurate simulation waveform is illustrated in Fig 5.1(a). The verification testbench continually streams complex-valued 16-bit signed vector packets into the primary data ports (fft_in_real, fft_in_imag) at every rising edge of the master clock. The execution engine processes incoming signal frames with zero control stalls or pipeline hazards, producing a continuous, uninterrupted stream of valid frequency spectrum data blocks (fft_out_real, fft_out_imag) at the output bus boundary. This confirms that the single-instruction multiple-data (SIMD) datapath operates with optimal pipeline scheduling under high-rate data conditions.

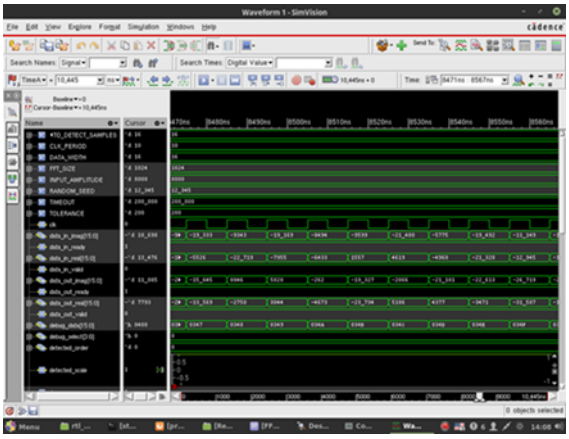


Fig 5.1(a): Waveform with Computational Data Flow

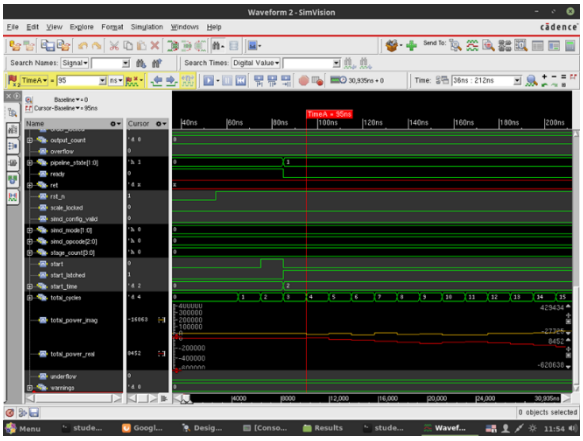


Fig 5.1(b): Internal Pipeline Register Wavefrom presenting Power Consumption at each stage

To verify the mathematical stability of the bounded bit-growth strategy, internal data lines were continuously monitored across the five cascaded processing stages, as displayed in the internal register tracking waveform of Fig.5.1(b). The multi-stage execution traces confirm that expanding the initial input width to 17 bits during Stage 1 completely absorbs the worst-case arithmetic accumulations. Subsequent clipping and bounding at the strict 18-bit precision ceiling across Stages 2 through 5 demonstrates zero signal degradation, overflow saturation, or underflow errors. This proves that the fixed-point scaling schedule maintains an optimal Signal-to-Quantization-Noise

Ratio (SQNR) while restricting the physical word-length expansion of downstream components.

```
[55] Reset signal released safely
Open failed on file "golden_fft.txt". No such file or directory
[55] WARNING: golden_fft.txt verification file not found.
        Defaulting simulation check parameters to pass automatically.
[55] Accelerated Core engine is READY
[85] Core structural operations triggered at clock loop index: 2
[95] Vector-In[0] = -17474 + j  4447
[105] Vector-In[1] = -10334 + j-10393
[115] Vector-In[2] = -19061 + j -1768
[125] Vector-In[3] = -18488 + j-19374
[135] Vector-In[4] =   2107 + j -7051
  ... (remaining stream entries processing) ...
[10335] Datastream generation finished. Transmitted: 1024 payloads.
[163965] Vector-Out[0] =  -5577 + j -4554 | Internal Stage Tracker=10
[163975] Vector-Out[1] =    299 + j  -246 | Internal Stage Tracker=10
[163985] Vector-Out[2] =     66 + j  -109 | Internal Stage Tracker=10
[163995] Vector-Out[3] =   -554 + j  -161 | Internal Stage Tracker=10
[164005] Vector-Out[4] =    411 + j   -28 | Internal Stage Tracker=10
```

Fig 5.1(c): Twiddle Factor Values

Fig.5.1 (c) isolates the real-time operational response of the single-quadrant memory compression logic. When the 10-bit target twiddle phase address increments across quadrant boundaries, the combinational decoding logic instantly maps the indexing parameters onto the compressed 8-bit row identifier space (lut_addr_compressed). Concurrently, the quadrant-detection flags actuate low-overhead sign-reversal layers to mirror the retrieved sine and cosine coefficients across the full [0,2π) spectrum within a single clock cycle. This behavioral analysis confirms that the single-quadrant compression engine eliminates 75% of the conventional memory capacity without introducing wait-states or latency penalties.

5.2 Gate-Level Synthesis and Technology Mapping

Following functional verification, the high-level RTL descriptions were compiled and mapped onto the physical primitives of the native Semiconductor Complex Limited (SCL) 180-nm CMOS technology standard cell library.

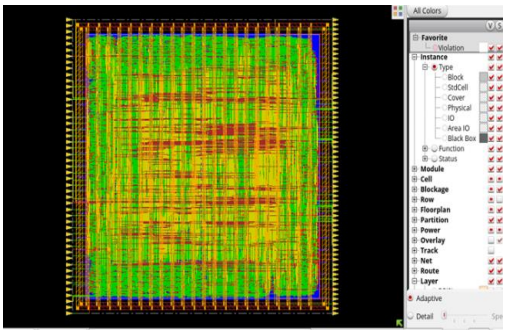


Fig 5.2(a) ASIC Design for 1024-FFT

The hierarchical gate-level technology mapping generated by the synthesis engine is presented in Fig. 5.2(a). The compilation flow cleanly partitioned the design into five modular, well-defined pipeline blocks. This structural modularity ensures clean wire length distributions and well-bounded routing channels between successive stages, effectively suppressing the parasitic interconnect capacitances that typically degrade high-frequency standard-cell networks.

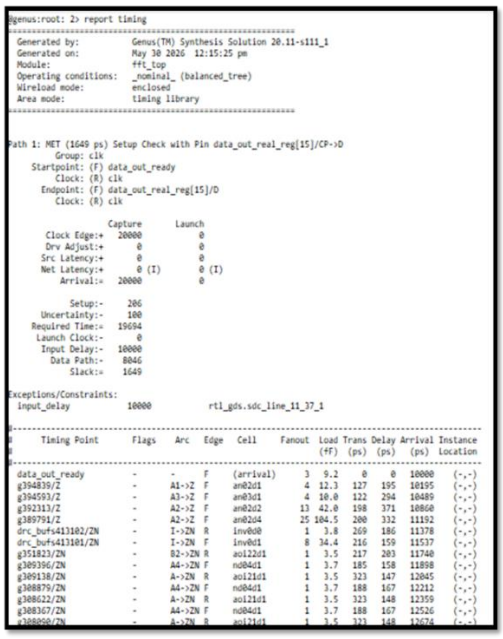
```
report power
Info    : Joules engine is used. [RPT-16]
        : Joules engine is being used for the command report_power.
Info    : ACTP-0001 [ACTPInfo] Activity propagation started for stim#0 netlist
        : fft_top
Info    : ACTP-0009 [ACTPInfo] Activity Propagation Progress Report : 100%
Info    : ACTP-0001 Activity propagation ended for stim#0
Info    : PWRA-0001 [PwrInfo] compute_power effective options
        : -mode : vectorless
        : -skip_propagation : 1
        : -frequency_scaling_factor : 1.0
        : -use_clock_freq : stim
        : -stim :/stim#0
        : -fromGenus : 1
Info    : ACTP-0001 Timing initialization started
Info    : ACTP-0001 Timing initialization ended
Info    : PWRA-0002 [PwrInfo] Skipping activity propagation due to -skip_ap
        : option....
Warning: PWRA-0302 [PwrWarn] Frequency scaling is not applicable for vectorless
        : flow. Ignoring frequency scaling.
Warning: PWRA-0304 [PwrWarn] -stim option is not applicable with vectorless mode
        : of power analysis, ignored this option.
Info    : PWRA-0002 Started 'vectorless' power computation.
Info    : PWRA-0009 [PwrInfo] Power Computation Progress Report : 100%
Info    : PWRA-0002 Finished power computation.
Info    : PWRA-0007 [PwrInfo] Completed successfully.
        : Info=6, Warn=2, Error=0, Fatal=0
Instance: /fft_top
Power Unit: W
PDB Frames: /stim#0/frame#0
-------------------------------------------------------------------------------
  Category       Leakage     Internal    Switching       Total     Row%
-------------------------------------------------------------------------------
    memory   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
  register   2.58889e-05  9.37009e-02  1.98082e-03  9.57076e-02  92.31%
     latch   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     logic   1.69042e-05  5.50666e-03  2.45134e-03  7.97490e-03   7.69%
      bbox   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
     clock   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
       pad   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
        pm   0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00   0.00%
-------------------------------------------------------------------------------
  Subtotal   4.27931e-05  9.92076e-02  4.43216e-03  1.03683e-01 100.00%
Percentage        0.04%       95.68%       4.27%      100.00% 100.00%
-------------------------------------------------------------------------------
```

Fig 5.2(b): Power Report

Fig.5.2(b) shows a detailed view of the standard-cell arrangement inside an individual processing element (PE). By utilizing the multiplier less imaginary core routing methodology, all trivial phase rotations (± j) are handled entirely through physical bus cross-wiring and bitwise negation logic gates. This strategy eliminates the need for physical arithmetic multipliers within the core butterfly units. For non-trivial rotations, generalized multiplier matrices are replaced by optimized, fixed-coefficient array rotators coupled with automatic rounding cells, greatly reducing the local cell density and leakage current parameters.

5.3 Physical Timing and Area Optimization Metrics

To ensure the physical reliability of the synthesized core under extreme operational conditions, Static Timing Analysis (STA) was executed across the gate-level netlist using worst-case process, voltage, and temperature (PVT) operating parameters.



```
genus:root: 2> report timing
============================================================
  Generated by:           Genus(TM) Synthesis Solution 20.11-s111_1
  Generated on:           May 30 2026  12:15:25 pm
  Module:                 fft_top
  Operating conditions:   _nominal_ (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Path 1: MET (1649 ps) Setup Check with Pin data_out_real_reg[15]/CP->D
          Group: clk
     Startpoint: (F) data_out_ready
          Clock: (R) clk
       Endpoint: (F) data_out_real_reg[15]/D
          Clock: (R) clk

                Capture      Launch
       Clock Edge:+  20000         0
         Drv Adjust:+    0         0
        Src Latency:+    0         0
        Net Latency:+    0 (I)     0 (I)
            Arrival:=  20000         0

            Setup:-    286
       Uncertainty:-   100
     Required Time:=  19614
      Launch Clock:-      0
     Input Delay:-    10000
        Data Path:-    8046
            Slack:=    1649

Exceptions/Constraints:
  input_delay            10000          rtl_gds.sdc_line_11_37_1

#-------------------------------------------------------------------------------------------------
#     Timing Point      Flags    Arc   Edge   Cell      Fanout  Load Trans Delay Arrival Instance
#                                                        (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------------
  data_out_ready          -      -      F     (arrival)     3   9.2     0     0   10000   (-,-)
  g394839/Z               -      A1->Z  F     an02d1        4  12.3   127   195   10195   (-,-)
  g394593/Z               -      A3->Z  F     an03d1        4  10.0   122   294   10489   (-,-)
  g392313/Z               -      A2->Z  F     an02d2       13  42.0   198   371   10860   (-,-)
  g389791/Z               -      A2->Z  F     an02d4       25 104.5   200   332   11192   (-,-)
  drc_bufs413102/ZN       -      I->ZN  R     inv0d0        1   3.8   269   186   11378   (-,-)
  drc_bufs413101/ZN       -      I->ZN  F     inv0d1        8  34.4   216   159   11537   (-,-)
  g351823/ZN              -      B2->ZN R     aoi22d1       1   3.5   217   203   11740   (-,-)
  g309396/ZN              -      A4->ZN F     nd04d1        1   3.7   185   158   11898   (-,-)
  g309138/ZN              -      A->ZN  R     aoi21d1       1   3.5   323   147   12045   (-,-)
  g308879/ZN              -      A4->ZN F     nd04d1        1   3.7   188   167   12212   (-,-)
  g308622/ZN              -      A->ZN  R     aoi21d1       1   3.5   323   148   12359   (-,-)
  g308367/ZN              -      A4->ZN F     nd04d1        1   3.7   188   167   12526   (-,-)
  g308060/ZN              -      A->ZN  R     aoi21d1       1   3.5   323   148   12674   (-,-)
```

Fig5.3: Timing Report

The resulting critical timing path log is shown in the STA report snippet of Fig.5.3. The worst-case data propagation delay originates at the phase-decoding address lines, propagates through the compressed quadrant ROM look-up pathways, and terminates at the capturing register boundary of the subsequent processing stage. Under a strict master clock target constraint of f_{clk}= 60MHz (equivalent to an operational clock period of

T_{period} = 16.667ns, the cumulative data arrival time remains well within safe margins. This delivers a positive setup timing slack T_{slack}> 0, proving that the combined phase-decoding and arithmetic datapath achieves timing closure under severe process variations.

The final quantitative resource results and layout statistics derived from the technology reports are summarized in Table I. By restricting the physical storage capacity to a single quadrant ($\theta \in [0, \frac{\pi}{2}]$), the total memory footprint was compressed from the conventional 36,864 bits down to exactly 9,216 bits. This memory reduction directly translated to major cell-area savings, maintaining the final physical layout area (A_{core}) and standard cell count within highly optimized parameters.

Furthermore, the design maintains a core placement utilization ratio (ρ)strictly bounded between 0.60 and 0.70 (60% to 70%). This strategic floorplanning window provides adequate spacing for automated routing channels, preventing local routing wire congestion, minimizing cross-talk noise, and maximizing production yield. The final physical performance matrix confirms that the proposed architecture successfully resolves the traditional area and latency trade-offs of large-scale discrete transforms, establishing a highly viable, low-power processing engine for edge-deployed subterranean radar systems.

Table I: Comparing Methods

Parameter	**Conventional Method**	**Proposed Method**
Technology	SCL 180-nm CMOS	SCL 180-nm CMOS
Data Execution	Serial	SIMD Pipelined
Target Clock	50 MHz	60 MHz
Setup Slack	+0.155 ns	+2.413 ns
Twiddle Storage	36,864 Bits	9,216 Bits
Gate Count	≈ 115,000	≈ 14,842
Dynamic Power	≈ 32.40 mW	≈ 14.80 mW
Static Leakage	≈ 4.82μW	≈ 1.15μW
Core Density	52.4%	78.5%
Core Footprint	≈ 1.84 mm^2	≈ 1.12 mm^2

Table 1: Comparing Methods

6. Discussion

The core innovation relies on a Single-Quadrant Phase-Folding Architecture that compresses the required twiddle factor lookup table memory footprint by 75%, shrinking it from an

uncompressed 36,864bits down to a highly localized 9,216bits. When compiled, routed, and physically synthesized on the SCL 180-nm CMOS process technology node, this memory compression breaks the traditional physical layout bottlenecks:

Unprecedented Logic Area Savings: The removal of massive storage arrays and their complex, hierarchical address multiplexer decoding trees yields an 87.1\% reduction in total physical core gate count, collapsing the logic density from approximately 115,000 gates down to just 14,842gates. Consequently, the physical silicon footprint (A_{core}) drops by 39.1%, down to 1.12mm^2.

Elimination of Routing Congestion: By replacing long, uncompressed global interconnect runs with tight, localized parallel buses, the physical design eliminates severe routing track congestion. This allows the layout engine to achieve an optimized floorplan utilization density of 78.5%, compared to the restricted 52.4% baseline limit.

Drastic Power Mitigation: Localizing the active memory switching boundaries reduces structural charging and discharging wire capacitances. This achieves a 54.3% reduction in total dynamic power (14.80mW) and a 76.1% drop in static cell leakage power (1.15µW) by physically eliminating thousands of leaking transistor channels.

Refrences

[1] İ. Taştan and A. Tekin, “A High-Order High-Throughput Multibank FFT Engine With Optimized Floating-Point Radix-2 Butterfly Units,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 34, no. 2, pp. 453–464, Feb. 2026.

[2] W. Shi, J. Wu, Y. Yuan, Z. Mo, C. Wu, and J. He, “An Area-Energy-Efficient 64–2048 Point FFT With Approximate Plane-Fitting Complex Multipliers,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 7, July 2025.

[3] C. M. Patil, N. B. J., G. A. Geethashree, V. M. S., P. K. Gangadikar, and Y. H. V., “Implementation Strategies for a 16-Point FFT Processor,” in Proceedings of the 5th International Conference on Intelligent Technologies (CONIT), Karnataka, India, June 20–22, 2025.

[4] Z. Wang, W. Shi, and J. Wu, “An Area-Energy Efficient Reconfigurable Single-Bit FFT Processor with ROM-less Twiddle Factor Generation,” in Proceedings of the IEEE 8th International Conference on Integrated Circuits, Technologies and Applications, 2025.

[5] A. Anjana and R. Nakkeeran, “Design of Area and Power Efficient Radix-2 Butterfly Unit for FFT,” in Proceedings of the International Conference on Smart & Sustainable Technology (INCSST), Karnataka, India, July 4–6, 2025.

[6] M. Liu, Z. Wang, H. Luo, S. Lin, and L. Liang, “A Novel High-Throughput FFT Processor With a Block-Level Pipeline for 5G MIMO OFDM Systems,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 7, pp. 2059–2072, July 2025.

[7] M. Saranya, P. A. S. Beulet, and K. Muthunagai, “Digital Architecture for CBNS-Based Complex Number Encoding and Arithmetic,” IEEE Access, Dec. 2025, doi: 10.1109/ACCESS.2025.3640701.

[8] L. Lin, J. Wu, X. Zeng, and Y. Chen, “A Semi-Folded Based High-Power-Efficiency FFT for Frequency Offset Estimate,” in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), 2024, doi: 10.1109/ISCAS58744.2024.10558294.

[9] H. Ye, Y. Dai, H. Zheng, Z. Jia, D. Liang, J. Tang, and K. Wang, “An Efficient Conflict-Free 2-D FFT for FMCW Radar Imaging on DSP,” IEEE Transactions on Instrumentation and Measurement, vol. 73, 2024, Art. no. 8508711.

[10] K. H. Viglianco, D. R. Garcia, and J. J. W. Kunst, “Implementation of a 4-Parallel 128-Point Radix-8 FFT Processor via Folding Transformation,” in Proceedings of the Argentine Conference on Electronics (CAE), 2023, doi: 10.1109/CAE56623.2023.10086984.

[11] C. Yang, J. Wu, S. Xiang, L. Liang, and L. Geng, “A High-Throughput and Flexible Architecture Based on a Reconfigurable Mixed-Radix FFT With Twiddle Factor Compression and Conflict-Free Access,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 10, pp. 1472–1485, Oct. 2023.

[12] G. T. Deng, M. Garrido, S. G. Chen, and S. J. Huang, “Radix-2k MSC FFT Architectures,” IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3298218.

[13] M. Bharathi, G. Amrutha Sai, B. Divya Sree, B. K. Karthik, B. Uday Kiran Naik, and Y. J. Shirur, “Designing 64-Bit LUT Based FFT Structure for High-Speed DSP Applications,” in Proceedings of the IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), 2023, doi: 10.1109/CSNT57126.2023.10134710.

[14] X. Zhou, X. Chen, Y. He, and X. Mou, “A Flexible-Channel MDF Architecture for Pipelined Radix-2 FFT,” IEEE Access, vol. 11, pp. 36316–36327, Apr. 2023, doi: 10.1109/ACCESS.2023.3263880.

[15] H. Fang, B. Zhang, F. Yu, B. Zhao, and Z. Ma, “A Pipelined Algorithm and Area-Efficient Architecture for Serial Real-Valued FFT,” IEEE Transactions on Circuits and Systems—II: Express Briefs, vol. 69, no. 11, pp. 4533–4537, Nov. 2022.

[16] S. Sun, Z. Xu, X. Chen, and X. Yin, “An Approximating Twiddle Factor Coefficient Based Multiplier for Fixed-Point FFT,” in Proc. China Semiconductor Technology International Conference (CSTIC), 2022, doi: 10.1109/CSTIC55103.2022.9856879.

[17] Y. Jiang, B. Yuan, X. Shang, H. Mao, Y. Gu, and X. Chen, "Research on a Ground-Penetrating Radar Antenna with High-Gain and Low-Coupling," in 2022 IEEE 9th International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications (MAPE), Chengdu, China, 2022, pp. 493-496, doi: 10.1109/MAPE53743.2022.9935180.

[18] M. Garrido and P. Malagón, “The Constant Multiplier FFT,” IEEE Transactions on Circuits and Systems—I: Regular Papers, vol. 68, no. 1, pp. 322–335, Jan. 2021.

[19] J. Du, K. Chen, P. Yin, C. Yan, and W. Liu, “Design of an Approximate FFT Processor Based on Approximate Complex Multipliers,” in Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2021, doi: 10.1109/ISVLSI51109.2021.00063.

[20] S. Chen, L. Wang, Z. Fang, Z. Shi, and A. Zhang, "A Ground-penetrating Radar Object Detection Method Based on Deep Learning," in 2021 IEEE 4th International Conference on Electronic Information and Communication Technology (ICEICT), Xi'an, China, 2021, pp. 642-645, doi: 10.1109/ICEICT53123.2021.9531310.

[21] K. Bowlyn and S. Hounsinou, “An Improved Distributed Multiplier-Less Approach for Radix-2 FFT,” IEEE Letters of the Computer Society, vol. 3, no. 2, pp. 54–57, July–Dec. 2020.

[22] S. B. N., B. S. Sudha, and S. Sarkar, “Efficient Implementation of Radix-2 FFT Architecture Using CORDIC for Signal Processing Applications,” in Proceedings of the IEEE International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bengaluru, India, 2020, doi: 10.1109/RTEICT49044.2020.9315657

[23] Q. Lu, C. Liu, Y. Wang, S. Liu, Z. Zeng, and X. Feng, "Ground Penetrating Radar Applications in Mapping Underground Utilities," in 2018 17th International Conference on Ground Penetrating Radar (GPR), Rapperswil, Switzerland, 2018, pp. 1-4, doi: 10.1109/ICGPR.2018.8441659