Int. J. Sci. R. Tech., 2026 3(11)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

Low-Power and Multiplier-Less 2-D DCT Architectures for Image Compression: A Comprehensive Survey

V. P. Krishnammal*, Ashvath S., Bhuvanesh S., Gokul Raj K.

Department of Electronics and Communication Engineering, Er. Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu, India.

ABSTRACT

In transform-based image and video compression systems, the Discrete Cosine Transform (DCT) serves as a vital mathematical kernel responsible for compacting spatial correlation into a concentrated set of decorrelated transform coefficients. While mathematically elegant, direct hardware realization of the 2-D 8×8 DCT creates substantial architectural bottlenecks in power-constrained embedded systems, edge-vision sensors, and multimedia terminals. Conventional direct matrix evaluation requires extensive arrays of multi-bit multipliers and adders, causing severe dynamic power dissipation, silicon area bloat, and long propagation delays. This survey provides an exhaustive comparative review of low-power and multiplier-less architectural strategies designed to streamline arithmetic complexity. We examine fast orthogonal algorithmic factorizations (Chen and Loeffler formulations), custom circuit topologies (such as Gate Diffusion Input and Pass Transistor Logic Wallace Tree multipliers), Canonical Signed Digit (CSD) number systems, Common Subexpression Elimination (CSE) graph sharing, integer dyadic lifting architectures (BinDCT), and pruned approximate transforms. Particular analytical focus is directed toward two state-of-the-art implementations: a hybrid GDI-PTL Wallace Tree multiplier architecture tailored for DCT processing and an eight-stage pipelined Loeffler transform utilizing modified Booth arithmetic. These works are compared against earlier multiplier-less CSD-CSE topologies to evaluate critical design trade-offs across operator counts, critical-path timing, pipeline balancing, register allocation, word-length precision truncation, and transpose buffer storage overhead. Furthermore, we establish standardized benchmarking criteria across FPGA resources, ASIC cell area, power components, throughput, latency, and reconstructed visual fidelity (PSNR and SSIM) to establish a rigorous foundation for developing optimized, fully multiplier-less 8×8 2-D DCT accelerators.

Keywords: Canonical Signed Digit (CSD), Common Subexpression Elimination (CSE), Discrete Cosine Transform (DCT), FPGA Architecture, Image Compression, Loeffler Algorithm, Low-Power VLSI, Multiplier-Less Hardware, Transpose Memory Buffer

INTRODUCTION

I. INTRODUCTION AND ARCHITECTURAL MOTIVATION:

The widespread deployment of battery-powered embedded vision systems for unmanned aerial vehicles, autonomous automotive platforms, and edge IoT camera nodes has intensified the demand for energy-efficient digital image and video processing accelerators. In these operating environments, physical communication bandwidth, on-chip thermal dissipation, and battery endurance represent fundamental design constraints. Consequently, baseline transform-based image compression standards—such as JPEG and the block-based transformation stages of hybrid video codecs including H.264/AVC and HEVC—remain critical processing elements within modern system-on-chip (SoC) architectures [1,2].

At the foundation of block-based transform coding lies the two-dimensional Discrete Cosine Transform (2-D DCT), introduced by Ahmed, Natarajan, and Rao in 1974[1].

The DCT is renowned for its exceptional energy compaction capability. For typical natural imagery possessing strong spatial correlation among neighboring pixels (often characterized by Markov-1 correlation parameters exceeding 0.90), the DCT closely approximates the decorrelation performance of the statistically optimal, data-dependent Karhunen-Loève Transform (KLT). By converting spatial domain pixel

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

intensity blocks into uncorrelated frequency coefficients, the transform concentrates the vast majority of image energy into a compact cluster of low-frequency coefficients located near the upper-left corner of the transform matrix[1,3].

Despite its exceptional compaction efficacy, mapping the 2-D DCT directly into dedicated hardware presents severe VLSI implementation bottlenecks. When realized via direct matrix multiplication, an 8×8 block transform necessitates hundreds of multi-bit arithmetic operations. Each multiplication between variable 8-bit input pixels and irrational transcendental cosine basis coefficients requires substantial silicon area, consumes high dynamic switching power, and generates deep carry-propagation delay chains. While software implementations running on general-purpose digital signal processors (DSPs) can leverage multi-GHz clock rates and wide SIMD vector instructions, embedded hardware accelerators must achieve multi-megapixel real-time throughput under strict milli-watt power envelopes[4,5].

To overcome these physical limitations, the VLSI design and signal processing communities have explored multiple complementary optimization pathways across distinct levels of the abstraction hierarchy:

- **Algorithmic Orthogonal Factorizations:** These methods decompose the dense transform matrix into sparse orthogonal matrices, symmetry matrices, and butterfly flowgraphs, substantially reducing the baseline count of mathematical operations. Prominent examples include Chen's fast algorithm[3], Lee's recursive decimation-in-time approach[6], and the canonical Loeffler-Ligtenberg-Moschytz algorithm[7], which evaluates an 8-point 1-D DCT using merely 11 multiplications and 29 additions.
- **Advanced Circuit-Level Arithmetic Architectures:** Rather than avoiding multiplication, this paradigm re-engineers the physical transistor topologies of multi-bit multipliers and adders. Techniques such as Modified Booth encoding, Wallace Tree partial-product reduction networks, and novel logic styles—such as Gate Diffusion Input (GDI) and Pass Transistor Logic (PTL)—minimize transistor switching activity, parasitic wiring capacitance, and static leakage[4,8].
- **Multiplier-Less Algebraic Formulations:** Observing that DCT basis coefficients are fixed, time-invariant constants, this approach replaces multi-bit hardware multipliers with hardwired binary shifts and additions. By representing coefficients in signed-digit representations—specifically Canonical Signed Digit (CSD) formatting—the number of non-zero operations is minimized. Furthermore, Common Subexpression Elimination (CSE) identifies redundant algebraic expressions across multiple output paths, sharing intermediate sums to drastically diminish the required adder count[9,10,5].
- **Integer Lifting and Approximate Transforms:** For applications with flexible visual fidelity tolerances, approximate transforms replace transcendental cosine values with dyadic rational approximations or integer lifting steps. Architectures like BinDCT[11] and pruned approximate transforms[12,13,14] completely eliminate fractional multiplications, requiring only coarse integer additions and bit-shifts, albeit at the cost of controlled distortion in reconstructed imagery.

A rigorous understanding of these architectural alternatives is imperative. A common misconception in digital VLSI literature is equating 'low operation count' directly with 'low power' or 'low silicon area.' In practical physical implementations, overall efficiency is governed by a multi-dimensional matrix of design variables: combinational critical-path delay, pipeline balancing registers, routing congestion, intermediate bus bit-widths, memory bandwidth in corner-turn transpose buffers, and target technology library characteristics[5,11]. This survey delivers a deep comparative exploration contextualizing two recent base papers—the hybrid GDI-PTL Wallace Tree multiplier by Bhavya and Vijaya Prakash[4] and the optimized 8-stage pipelined Loeffler transform by Chinmayi and Veena[8]—alongside earlier multiplier-less CSD-CSE topologies[9,10,5] and lifting frameworks[11,15].

II. THEORETICAL BACKGROUND AND MATHEMATICAL FOUNDATIONS:

A. Mathematical Formulation Of 1-D And 2-D DCT

The Discrete Cosine Transform family comprises eight distinct mathematical variants (DCT-I through

DCT-VIII). In image compression hardware, the orthonormal DCT-II formulation is universally adopted due to its superior boundary condition handling and energy compaction properties. For an N-point real spatial input sequence x [n], where n = 0, 1, ..., N-1, the forward 1-D DCT-II coefficient sequence X [k], for k = 0, 1, ..., N-1, is defined by:

$$X[k] = \alpha(k) \cdot \Sigma [n=0 \text{ to } N-1] \{x[n] \cdot \cos[(\pi \cdot (2n + 1) \cdot k) / (2 \cdot N)]\}$$

where the orthogonal normalization scale factor α(k) is defined as $\alpha(0) = \sqrt{(1 / N)}$ for k = 0 (the DC component), and $\alpha(k) = \sqrt{(2 / N)}$ for k = 1, 2, ..., N-1 (the AC components). The DC coefficient at k = 0 represents the mean pixel intensity across the input block, capturing average luminance. The AC coefficients (k > 0) correspond to sinusoidal basis functions of increasing spatial frequencies, capturing localized edges, textures, and rapid spatial transitions[1,3].

Because natural images are two-dimensional signals, compression systems process square two-dimensional spatial blocks of size N×N (typically 8×8 blocks, corresponding to N = 8). For an N×N spatial intensity matrix A, where A (i, j) represents the pixel intensity at row i and column j, the 2-D DCT coefficient matrix Y (u, v) is defined by the double summation:

$$Y[u, v] = \alpha(u) \cdot \alpha(v) \cdot \Sigma [i=0 \text{ to } N-1] \Sigma [j=0 \text{ to } N-1] \{A[i, j] \cdot \cos[(\pi(2i + 1)u) / (2N)] \cdot \cos[(\pi(2j + 1)v) / (2N)]\}$$

Direct evaluation of this 2-D transform requires N^4 mathematical multiplications and $N^2(N^2 - 1)$ additions. For an 8×8 block, a naive direct computation requires 4,096 multiplications and 4,032 additions per block, presenting an intolerable computational burden for dedicated hardware architectures.

B. Separability And Row-Column Decomposition

A decisive mathematical characteristic of the 2-D DCT kernel is its separability. Because the 2-D cosine basis function factors into the direct product of independent orthogonal one-dimensional horizontal and vertical basis functions, the complete two-dimensional matrix transformation evaluates as two successive one-dimensional matrix transformations. Let C denote the N×N orthogonal 1-D DCT transform matrix. The 2-D transform of the spatial image block A can be compactly written in matrix notation as:

$$Y = C \cdot A \cdot C^T$$

where C^T represents the matrix transpose of C. Under this row-column decomposition, the evaluation proceeds in two distinct phases: (i) an intermediate matrix $Z = A \cdot C^T$ is evaluated by performing N 1-D DCT operations along the rows of A, and (ii) the final coefficient matrix $Y = C \cdot Z$ is evaluated by performing N 1-D DCT operations along the columns of Z. This separability property immediately reduces total computational complexity from $O(N^4)$ to $O(2N^3)$. For an 8×8 block, the multiplication count collapses from 4,096 down to 1,024—a fourfold mathematical reduction before applying any fast algorithmic factorization[3,6].

C. Block Processing, Transpose Buffering, And Memory Bottlenecks

While row-column decomposition delivers substantial algorithmic savings, its physical realization in pipelined VLSI hardware introduces a major structural challenge: intermediate matrix storage and data reordering, commonly designated as the corner-turn or transpose buffer[11,3,7]. In a pipelined streaming hardware datapath, the row-processing 1-D DCT engine produces transformed intermediate samples in row-major raster order: Z (0, 0), Z (0, 1), ..., Z (0, 7), followed by Z (1, 0), Z (1, 1), etc.

However, the subsequent column-processing 1-D DCT engine requires its inputs presented strictly in column-major raster order: Z (0, 0), Z (1, 0), ..., Z (7, 0), followed by Z (0, 1), Z (1, 1), ..., Z (7, 1).

Consequently, the hardware must buffer the entire 64-word intermediate matrix before the second transform stage can initiate processing on column vectors without data hazard. In continuous, real-time streaming architectures where new spatial blocks arrive seamlessly every cycle, a single 64-word buffer causes catastrophic pipeline stalls. Designers are therefore compelled to implement ping-pong buffering architectures consisting of dual 64-word memory banks (totaling 128 storage locations).

While one memory bank is being written with freshly evaluated row coefficients from the current block, the alternate bank is concurrently read out along columns to feed the second-stage column engine.

In deep submicron CMOS and modern FPGA fabrics, the silicon area, static leakage, and dynamic clocking power consumed by this transpose buffer frequently match or even surpass the area and power of the

arithmetic operators themselves. A profound finding highlighted in the BinDCT hardware exploration by Abdelali et al.[11] is that the memory interface timing, address decoding logic, and interconnect routing capacitance of the transpose memory often determine the maximum global clock frequency (F_max) of the entire system, even when isolated arithmetic processing stages can operate at significantly higher frequencies. Hence, optimizing arithmetic logic without addressing transpose storage efficiency produces suboptimal system-level results.

D. Canonical Signed Digit (CSD) Representation

In dedicated hardware transforms, the elements of the transform matrix C are known, fixed trigonometric constants. Dedicated multi-bit multipliers are designed to multiply two arbitrary, time-varying variables. Applying such generalized hardware to compute products with static constants introduces massive structural redundancy. If a constant is represented in conventional binary notation, multiplication reduces to an accumulation of hardwired shifts. However, standard two's complement binary representation often contains extensive consecutive sequences of '1' bits, requiring numerous addition stages.

The Canonical Signed Digit (CSD) representation is a unique radix-2 signed-digit numeral system that minimizes the number of non-zero digits for any given numeric constant[9,5,15]. A number in CSD utilizes the ternary digit alphabet {-1, 0, +1}, typically denoted as {1̄, 0, 1}, where 1̄ represents -1. The CSD representation is governed by two strict mathematical properties: (i) no two consecutive digits are non-zero (i.e., d_i · d_{i-1} = 0 for all bit positions i), and (ii) it achieves the absolute minimum Hamming weight (minimum number of non-zero digits) across all possible signed-digit representations of that value.

Statistically, while a standard N-bit binary integer contains an average of N/2 non-zero bits, an N-bit CSD number contains an average of only N/3 non-zero digits—yielding an immediate 33.3% theoretical reduction in addition/subtraction operations. For example, consider multiplying an arbitrary variable x by the integer constant 15. In standard binary representation, $15 = 1111_2$, requiring three additions:

$15 \cdot x = (x << 3) + (x << 2) + (x << 1) + x$

In CSD representation, 15 is encoded as 10001_2(since 16 - 1 = 15). The Hamming weight drops from 4 down to 2, and the multiplication is evaluated using a single hardwired subtraction:

$15 \cdot x = (x << 4) - x$

In digital hardware, binary shifts by static powers of two (e.g., << 4) do not require physical logic gates, multiplexers, or clock cycles; they are realized entirely 'for free' through hardwired routing connections. Consequently, CSD converts complex multi-bit multiplication directly into minimal networks of primitive adders and subtractors[9,10].

E. Common Subexpression Elimination (CSE)

Although CSD representation substantially reduces the number of non-zero digits within individual constant multiplications, applying CSD independently to each matrix coefficient frequently results in redundant, duplicated arithmetic operations across the overall system. In an 8-point 1-D DCT, multiple output coefficients share related linear combinations of the input samples.

Common Subexpression Elimination (CSE) is an algebraic graph optimization technique that identifies identical sub-expressions recurring across multiple arithmetic paths and computes them exactly once, sharing the intermediate result among all dependent pathways[10,5].

To illustrate the power of CSE, consider an idealized datapath that must concurrently evaluate two separate outputs, y_1 and y_2, from a common variable input x:

$y_1 = 5 \cdot x$, $y_2 = 13 \cdot x$

In CSD representation, 5 is represented as 101_2 (4 + 1), and 13 is represented as 10101_2 (16 - 4 + 1), or equivalently (8 + 4 + 1). Evaluated independently:

$y_1 = (x << 2) + x$ (requires 1 adder)

$y_2 = (x << 3) + (x << 2) + x$ (requires 2 adders)

Independent evaluation consumes a total of 3 adder units. By applying CSE, we observe that the term ((x << 2) + x) = 5x is entirely embedded within the evaluation of y_2. By defining a shared intermediate variable:

$t_1 = (x << 2) + x$ (1 adder instantiated)

the two outputs are evaluated as:

$y_1 = t_1$ (0 additional adders)

$y_2 = t_1 + (x << 3)$ (1 additional adder)

The total hardware cost drops from 3 adders to 2 adders—a 33.3% hardware reduction. In an 8×8 DCT datapath containing dozens of interrelated fixed-coefficient operations, the systematic application of CSE across horizontal and vertical sub-graphs eliminates massive amounts of redundant silicon logic[10,5].

F. Numerical Precision, Truncation, And Approximation Mechanics

In hardware design, a critical distinction must be drawn between representation transformation and mathematical approximation. Converting an irrational cosine coefficient into a finite-word-length fixed-point constant inherently introduces quantization error. However, once a fixed-point numerical word length is established, CSD encoding and CSE algebraic graph manipulation are mathematically exact; they introduce zero additional error beyond the established fixed-point quantization[9,5].

In stark contrast, approximate and lifting-based transforms—such as BinDCT[11] and pruned transforms[12,13,14]—intentionally modify the core mathematical transformation matrix. They replace transcendental cosine basis values with integer fractions (e.g., 3/8, 1/2) that can be computed with minimal shifts, or they intentionally zero-out specific high-frequency transform outputs. While this radical structural modification achieves significant silicon area and power reductions, it degrades orthogonality, introduces residual reconstruction distortion, and can cause catastrophic error accumulation if paired with non-matched inverse transforms. A robust literature review must strictly separate exact fixed-point multiplier-less transforms from approximate, lossy computational transforms.

III. REVIEW OF DCT HARDWARE METHODOLOGIES:

A. Conventional Multiplier-Based Topologies

Direct implementations preserve original matrix equations via parallel or time-multiplexed multiplier-accumulator (MAC) blocks. While structurally straightforward and providing an ideal baseline for checking functional correctness and precision, direct parallel arrays consume excessive silicon area and dynamic switching power[4,5,1].

Consequently, purely direct multiplier-based implementations are obsolete in modern embedded SoCs, preserved only as verification baselines or in specialized DSPs possessing dedicated hard-macro multiplier blocks[4,8,5].

B. Fast DCT Algorithmic Factorizations: Chen, Lee, And Loeffler

Fast algorithms exploit symmetry, periodicity, and matrix factorization to prune operational graphs. Chen's algorithm decomposes the 8-point 1-D DCT into even and odd sub-matrices, requiring 16 multiplications and 26 additions[3], while Lee's recursive decimation-in-time approach maps power-of-two point DCTs into regular butterfly graphs[6]. The canonical Loeffler algorithm achieves the theoretical minimum operational bound: exactly 11 multiplications and 29 additions across four cascaded stages[7]. However, the Loeffler graph is highly irregular and asymmetric; data paths spanning different output coefficients traverse vastly different numbers of arithmetic stages, introducing severe path-delay imbalances that require careful pipeline balancing registers[8,10,7].

C. Advanced Circuit-Level Multiplier And Adder Optimization

Circuit-level optimization focuses on re-engineering the physical transistor topologies of multi-bit arithmetic units. Bhavya and Vijaya Prakash (2026) introduced a Wallace Tree reduction multiplier utilizing hybrid full adders that combine Gate Diffusion Input (GDI) for carry generation and Pass Transistor Logic (PTL) for XOR sum paths[4]. While achieving substantial power savings in custom ASIC simulations (90-nm/45-nm), custom GDI-PTL topologies cannot be synthesized directly onto commercial FPGAs, where synthesizers map RTL operators into pre-fabricated lookup tables[4]. Concurrently, Chinmayi and Veena (2025) coupled the Loeffler factorization with Modified Booth multipliers and Modified Carry-Select Adders (CSLA) utilizing binary-to-excess-1 converters, implementing an 8-stage pipeline across 45-nm CMOS and Virtex-7 FPGA targets[8].

D. Canonical Signed Digit (CSD) Constant-Coefficient Arithmetic

Jridi and Alfalou[9] demonstrated that expressing irrational cosine constants in CSD prunes partial-product trees to their minimum theoretical Hamming weight. Crucially, they observed that standalone CSD replacement does not automatically guarantee optimal

silicon efficiency. Isolated CSD networks can duplicate arithmetic sub-terms across adjacent parallel paths, create deep cascaded adder chains, and inflate register bit-widths to prevent overflow. Direct standalone CSD replacement can actually increase total register and logic requirements compared to embedded-multiplier baselines, underscoring that CSD must be combined with global expression sharing[9,10].

E. Multiplier-Less Architectures Via Common Subexpression Elimination (CSE)

Merging CSD with Common Subexpression Elimination (CSE) yields fully multiplier-less datapaths. Jridi, Alfalou, and Meher[10] applied subexpression elimination directly to the 8-point Loeffler factorization, replacing its 11 multiplications with shared addition and shift primitives. Siddiqui et al.[5] subsequently developed a pipelined CSD-CSE 8×8 core requiring only 22 unified adder/subtractor and selectable units across several FPGA families. Importantly, CSE (spatial optimization sharing identical intermediate values within the same cycle) must be distinguished from Time-Multiplexed Resource Sharing (temporal optimization re-using a physical operator across multiple cycles), as each imposes different routing, latency, and control trade-offs[10,5,11].

F. Integer Dyadic Lifting And Pruned Approximate DCT Topologies

Approximate transforms deliberately sacrifice strict precision to eliminate hardware complexity. BinDCT[15] approximates rotation lifting coefficients using dyadic rational fractions ($k / 2^m$), evaluated strictly via binary shifts and additions[11,15]. Cintra, Bayer, and Potluri developed integer approximate transforms requiring only additions (e.g., 14-addition 8-point transforms[13,16]), while Coutinho et al. introduced a pruned 8-point transform requiring merely 10 additions by zeroing high-frequency coefficients[14]. However, approximate transforms introduce high-frequency reconstruction distortion and non-zero rounding bias, making exact CSD-CSE architectures preferable when numerical fidelity must be preserved[5,14,17].

G. Pipelining Strategies, Register Allocation, And Balancing

Pipelining partitions combinational logic paths into smaller segments separated by flip-flops, elevating operating frequencies. However, in 2-D DCT datapaths, each additional pipeline stage introduces multi-bit registers and clock-distribution overhead. While Siddiqui et al.[5] selected an elegant 5-stage pipeline for their CSD-CSE core, Chinmayi and Veena[8] instantiated an 8-stage pipeline to achieve timing closure for their complex Booth/CSLA Loeffler datapath. Determining the optimal pipeline depth is a primary architectural design variable trading off throughput, register area, and initialization latency[8,5].

H. Silicon Implementation Realities: FPGA Vs. Standard-Cell ASIC

A fundamental distinction exists between FPGA and ASIC silicon platforms[4,5,11]. On commercial FPGAs containing pre-fabricated DSP48 blocks, multiplier-less CSD-CSE designs consume logic LUTs to eliminate DSP blocks—valuable primarily in low-cost FPGA families (e.g., Spartan or Artix) where DSP slices are scarce[5,11]. In standard-cell ASICs, where every multiplier must be synthesized from logic gates, eliminating multipliers via CSD-CSE yields direct, massive reductions in total silicon area, static leakage, and dynamic switching power[4,16,14].

I. End-To-End System-Level Compression Integration

Evaluating the DCT within an end-to-end image compression pipeline (alongside quantization, zigzag scanning, and entropy coding) enables transformative optimizations. Jridi and Alfalou[9] demonstrated that by co-designing fixed-point CSD scaling factors directly with quantization divisor matrices, intermediate normalization multiplications can be completely absorbed into quantizer lookup tables, eliminating an entire computational stage from the hardware pipeline[9,2].

IV. DETAILED COMPARATIVE ANALYSIS

The two primary base papers address transform efficiency through complementary design paradigms: Bhavya and Vijaya Prakash[4] focus on transistor-level circuit optimization of Wallace Tree multipliers, whereas Chinmayi and Veena[8] address architectural datapath optimization of a pipelined Loeffler transform engine. Both retain dedicated multi-bit multiplication hardware. In contrast, earlier works by Siddiqui et al.[5] and Jridi et al.[10] eliminate multipliers entirely using algebraic CSD-CSE shift-add networks.

Table 1: Provides an Exhaustive Multi-Variable Comparative Matrix Contrasting Both Base Papers with Prior Csd-Cse Literature and the Planned Multiplier-Less Core Across Ten Fundamental Architectural Dimensions.

Architectural Dimension	Bhavya & Vijaya Prakash (2026)[4]	Chinmayi & Veena (2025)[8]	Prior CSD-CSE: Siddiqui et al. (2014)[5]	Planned Multiplier-Less CSD-CSE Core
Primary Design Paradigm	Transistor-level circuit optimization of multiplier cells	Architectural datapath optimization of Loeffler engine	Algebraic multiplier-less CSD-CSE FPGA datapath	Fully synthesizable, portable RTL multiplier-less architecture
Arithmetic Multiplier Units	Retained: Hybrid GDI-PTL Wallace Tree multipliers	Retained: Modified Booth Radix-4 multipliers (11 units)	Completely Eliminated: Zero dedicated multipliers	Completely Eliminated: Zero dedicated multipliers or DSP blocks
Adder / Subtractor Logic	GDI-PTL hybrid full adders (carry/sum split)	Modified Carry-Select Adders (CSLA) with BEC	Unified Adder/Subtractor and selectable units (22 units)	Common subexpression shared adder/subtractor tree
Underlying Transform Model	Direct DCT constant-coefficient multiplication	Loeffler 8-point fast factorization algorithm[7]	Separable 2-D DCT with matrix symmetry exploitation	Orthogonal 8×8 row-column separable 2-D DCT[1]
Pipelining Depth & Control	Arithmetic-cell level (unpipelined macro cell)	8 balanced pipeline stages across butterfly stages	5 pipeline stages with balanced register insertion	Optimal 5-to-6 stage balanced register pipeline
Corner-Turn Transpose Buffer	Not addressed (isolated multiplier cell focus)	Standard register-based transpose buffer array	Distributed internal register-based buffering	Resource-efficient ping-pong dual-port SRAM / register buffer
Target Silicon Technology	ASIC: 90-nm and 45-nm CMOS (Cadence Spectre)	FPGA: Virtex-7 (xc7vx485t) & ASIC: 45-nm (Cadence)	FPGA: Spartan-3, Virtex-4, Virtex-5 (Xilinx ISE)	FPGA: Artix-7 / Zynq-7000 / Kintex-7 & synthesizable ASIC
Reported Power Reductions	Up to ~90% power cut vs. static CMOS Wallace baseline	25% lower leakage power vs. conventional DCT baseline	Substantial dynamic power cut across FPGA families	Targeted ~50-55% dynamic power cut vs. multiplier core
Reported Area & Delay Gains	Significant reduction in transistor count and layout area	24% lower area, 8.8% lower critical-path delay	Only 22 arithmetic operators; compact logic footprint	Targeted ~40-45% logic area reduction; low latency
Design Portability & Reproducibility	Low: Custom transistor-level layouts; non-synthesizable in RTL	Moderate: Standard synthesizable HDL with CSLA macros	High: Synthesizable Verilog targeting commercial FPGAs	Very High: 100% vendor-agnostic, synthesizable Verilog HDL

V. STANDARDIZED PERFORMANCE EVALUATION CRITERIA AND BENCHMARKING FRAMEWORK:

To establish an authoritative evaluation foundation, hardware transform architectures must be evaluated across five rigorous, orthogonal performance axes[4,5,11,17]:

- **Silicon Footprint Accounting:** FPGA reporting must delineate CLB slices, 6-input LUTs (logic vs. memory), flip-flops, BRAM tiles, and DSP48 slices[5,11]. ASIC reporting must specify process node, cell library corner, standard cell area (µm² or 2-input NAND gates), and routing die area[4,8,16].
- **Power Breakdown Dynamics:** Power reporting must separate dynamic switching power ($P_{dyn} = \alpha \cdot C_L \cdot V_{dd}^2 \cdot f_{clk}$) from static leakage ($P_{stat} = I_{leak} \cdot V_{dd}$) under stated supply voltages, operating frequencies, temperatures, and verified switching activity (e.g., VCD files from real images)[4,8]. Energy per processed 8×8 block ($E_{block} = P_{total} / Throughput$) serves as the primary metric[4,5].
- **Timing, Throughput, and Latency:** Reporting must isolate critical-path combinational delay (t_{crit}, establishing maximum clock frequency F_{max}), pipeline latency (cycles to first block output), and sustained throughput (blocks or megapixels processed per second)[8,5,11].

- **Arithmetic Logic Depth and Fan-Out:** Evaluations must quantify cascaded adder logic depth, maximum electrical fan-out load on shared intermediate nets, and interconnect routing congestion[10,5].
- **Objective Visual Quality Metrics:** Transform accuracy must be verified by passing benchmark test images (e.g., Lena, Barbara, Mandrill, Cameraman, Boat, 512×512) through forward DCT, quantization, and inverse DCT, reporting Mean Squared Error (MSE), Peak Signal-to-Noise Ratio ($PSNR = 10 \cdot \log_{10}(255^2 / MSE)$), and Structural Similarity Index (SSIM)[9,17].

VI. CRITICAL RESEARCH GAPS AND ARCHITECTURAL DESIGN OPPORTUNITIES:

A thorough synthesis of the reviewed literature reveals four major unaddressed research bottlenecks[4,8,5,11]:

A. Subexpression Sharing Under Physical Timing Constraints: Traditional CSE algorithms maximize eliminated additions without accounting for physical fan-out loads. Sharing an intermediate sum across 16 downstream destinations creates massive capacitive loading, degrading slew rates and inducing timing violations. A major opportunity lies in developing timing-aware and fan-out-constrained CSE algorithms[9,10,5].

B. Precision-Aware Intermediate Word-Length Optimization: Uniform worst-case bit-widths across all internal nodes introduce substantial hardware waste. Profiling mathematical noise propagation along individual transform paths allows targeted word-length pruning on high-frequency paths, cutting adder sizes and switching capacitance while maintaining PSNR > 38.5 dB[9,5,11,17].

C. High-Throughput, Low-Overhead Transpose Memory: Corner-turn transpose buffering represents the primary memory bottleneck in pipelined 2-D engines. Implementing optimized interleaved multi-bank transpose buffers or compact ping-pong shift arrays enables conflict-free simultaneous read/write cycles with minimal silicon area[11,3].

D. Normalized Comparative Benchmarking Suite: Published literature frequently contrasts dissimilar platforms and non-standard baselines. There is an urgent need for an open-source, vendor-agnostic RTL benchmarking suite implementing direct, Loeffler-Booth, and CSD-CSE cores under identical precision, constraints, and toolchains[5,11].

VII. FUTURE RESEARCH DIRECTIONS IN TRANSFORM HARDWARE:

The architectural principles of low-power and multiplier-less transform processing extend into several emerging computational paradigms:

- **Dynamic Reconfigurability:** Designing adaptive CSD-CSE datapaths capable of morphing shift-add networks in real time across multiple block sizes (4×4 to 32×32) and transform types (DCT-II, DCT-VIII, DST-VII) required in modern video codecs (HEVC/VVC)[14,2].
- **Machine-Learning Assisted Synthesis:** Utilizing reinforcement learning and graph neural networks to explore the vast combinatorial design space of CSD permutations and pipeline balancing registers, replacing classic greedy CSE heuristics[10,5].
- **Approximate Edge-AI Transforms:** Embedding multiplier-less transform engines directly into deep neural network accelerators performing frequency-domain convolutions, exploiting inherent algorithmic fault tolerance for extreme energy efficiency[11,14].

CONCLUSION:

This survey has presented an exhaustive, multi-dimensional architectural analysis of low-power and multiplier-less Discrete Cosine Transform (DCT) hardware architectures for embedded image compression. We have systematically explored the full spectrum of hardware optimization strategies, ranging from mathematical orthogonal factorizations (Chen, Lee, and Loeffler algorithms) to transistor-level CMOS innovations (GDI-PTL Wallace Tree multipliers), algebraic constant-coefficient transformations (Canonical Signed Digit and Common Subexpression Elimination), and lossy integer lifting schemes (BinDCT and pruned approximate transforms).

A rigorous examination of two recent state-of-the-art studies—the hybrid GDI-PTL Wallace Tree multiplier introduced by Bhavya and Vijaya Prakash

(2026)[4] and the 8-stage pipelined Booth-Loeffler architecture developed by Chinmayi and Veena (2025)[8]—demonstrates that while circuit-level and pipelining techniques yield valuable delay and power reductions, they decisively retain dedicated multi-bit multiplication hardware. In contrast, multiplier-less CSD-CSE architectures fundamentally eliminate physical multipliers, mapping constant trigonometric multiplications directly into minimal networks of hardwired shifts and shared addition/subtraction operators[9,10,5].

Crucially, our analytical synthesis demonstrates that minimizing arithmetic operator counts alone does not guarantee superior silicon efficiency. Optimal real-world VLSI performance is determined by an interdependent balance among critical-path propagation delays, pipeline register balancing, high-fanout wiring constraints, intermediate word-length pruning, and corner-turn transpose memory buffering. By addressing the critical research gaps identified in this survey—specifically timing-constrained subexpression sharing, non-uniform word-length truncation, and conflict-free transpose memory organization—hardware designers can establish robust foundations for engineering next-generation, high-throughput, ultra-low-power 8×8 2-D DCT accelerators capable of meeting the stringent energy budgets of future embedded vision platforms.

REFERENCES

1. Ahmed N, Natarajan T, Rao KR. Discrete cosine transform. IEEE Transactions on Computers. 1974 Jan;C-23 (1):90-93. doi:10.1109/T-C.1974.223784
2. Wallace GK. The JPEG still picture compression standard. Communications of the ACM. 1991 Apr;34 (4):30-44. doi:10.1145/103085.103089
3. Chen WH, Smith CH, Fralick SC. A fast computational algorithm for the discrete cosine transform. IEEE Transactions on Communications. 1977 Sep;25 (9):1004-1009. doi:10.1109/TCOM.1977.1093941
4. Bhavya KB, Vijaya Prakash AM. Design and analysis of hybrid GDI-PTL based Wallace Tree multipliers for low-power DCT architecture in image compression. EAI Endorsed Transactions on Internet of Things. 2026;11. doi:10.4108/ee-tiot.10362
5. Siddiqui MF, Reza AW, Kanesan J, Ramiah H. Investigation of a novel common subexpression elimination method for low power and area efficient DCT architecture. The Scientific World Journal. 2014;2014:Art. no. 620868. doi:10.1155/2014/620868
6. Lee BG. A new algorithm to compute the discrete cosine transform. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1984 Dec;32 (6):1243-1245. doi:10.1109/TASSP.1984.1164443
7. Loeffler C, Ligtenberg A, Moschytz GS. Practical fast 1-D DCT algorithms with 11 multiplications. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Vol. 2. 1989. p. 988-991. doi:10.1109/ICASSP.1989.266596
8. Chinmayi SV, Veena MB. Design of optimized hardware architecture for discrete cosine transform using Loeffler's algorithm. EMITTER International Journal of Engineering Technology. 2025;13 (2):333-351. doi:10.24003/emitter.v13i2.819
9. Jridi M, Alfalou A. Joint optimization of low-power DCT architecture and efficient quantization technique for embedded image compression. In: VLSI-SoC: Forward-Looking Trends in IC and System Design. Berlin, Germany: Springer; 2012. p. 155-181. doi:10.1007/978-3-642-28566-0_7
10. Jridi M, Alfalou A, Meher PK. Optimized architecture using a novel subexpression elimination on Loeffler algorithm for DCT-based image compression. VLSI Design. 2012;2012:Art. no. 209208. doi:10.1155/2012/209208
11. Ben Abdelali A, Chatti I, Hannachi M, Mtibaa A. Efficient BinDCT hardware architecture exploration and implementation on FPGA. Journal of Advanced Research. 2016;7 (6):909-922. doi:10.1016/j.jare.2016.09.002
12. Cintra RJ, Bayer FM. A DCT approximation for image compression. IEEE Signal Processing Letters. 2011 Oct;18 (10):579-582. doi:10.1109/LSP.2011.2163394
13. Bayer FM, Cintra RJ. DCT-like transform for image compression requires 14 additions only. Electronics Letters. 2012;48 (15):919-921. doi:10.1049/el.2012.1148

14. Coutinho VA, Cintra RJ, Bayer FM, Kulasekera S, Madanayake A. A multiplierless pruned DCT-like transformation for image and video compression that requires 10 additions only. Journal of Real-Time Image Processing. 2016;12 (2):247-255. doi:10.1007/s11554-015-0492-8
15. Liang J, Tran TD. Fast multiplierless approximations of the DCT with the lifting scheme. IEEE Transactions on Signal Processing. 2001 Dec;49 (12):3032-3044. doi:10.1109/78.969511
16. Potluri US, Madanayake A, Cintra RJ, Bayer FM, Kulasekera S, Edirisuriya A. Improved 8-point approximate DCT for image and video compression requiring only 14 additions. IEEE Transactions on Circuits and Systems I: Regular Papers. 2014 Jun;61 (6):1727-1740. doi:10.1109/TCSI.2013.2295022
17. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing. 2004 Apr;13 (4):600-612. doi:10.1109/TIP.2003.819861
18. Bayer FM, Cintra RJ, Madanayake A, Potluri US. Multiplierless approximate 4-point DCT VLSI architectures for transform block coding. Electronics Letters. 2013;49 (24):1532-1534. doi:10.1049/el.2013.1352

HOW TO CITE: V. P. Krishnammal, Ashvath S., Bhuvanesh S., Gokul Raj K., Low-Power and Multiplier-Less 2-D DCT Architectures for Image Compression: A Comprehensive Survey, Int. J. Sci. R. Tech., 2026, 3 (11), 934-943. https://doi.org/10.5281/zenodo.24389709