Controllable Hardware Approximation with Canonical Signed Digit Multipliers

Temperature conversion and Gaussian blur case studies

Luk036

2026

Abstract

Constant multiplication by 9/5 is central to Celsius-to-Fahrenheit conversion, and separable Gaussian blur is a multi-coefficient FIR, yet a general multiplier is expensive in embedded and custom hardware. Canonical signed digit (CSD) encoding replaces each multiplier with shifts and additions; the number of non-zero digits (NNZ) directly controls the adder count. This paper treats NNZ as an accuracy knob across three designs. For an air-conditioner panel (-9 to 30 degrees Celsius, integer Fahrenheit), four NNZ suffice for exact rounding: 115/64 implements $F = (115C + 2080)/64$ with four adders, whereas three NNZ misrounds 7 of 40 readings. For an extreme range (-60 to 300 degrees Celsius, 0.001 degrees Fahrenheit), a direct binary approximation needs ten NNZ; nine NNZ misrounds 34 of 361 readings. Scaling by the display unit, $1000F = 1800C + 32000$, is exact with three NNZ because $1800 = 2^{11} - 2^8 + 2^3$. For a separable seven-tap Gaussian blur at scale 1024, the exact kernel needs 21 NNZ; 18 NNZ keep the worst-case pre-round error below 0.5 LSB, and 16 below 1 LSB. The 18-NNZ 1D filter uses 13 adders per pass (26 for the separable 2D blur) and is verified in Icarus Verilog and Yosys. The common lesson is to choose the representation before minimizing NNZ and to optimize the maximum error, not an average.

Keywords: canonical signed digit; multiplierless arithmetic; hardware approximation; fixed-point arithmetic; temperature conversion; Gaussian blur; image filtering; RTL verification.

1 Introduction

Multiplication by a constant is ubiquitous in embedded control, digital signal processing, and sensor interfaces. A general-purpose multiplier occupies substantial area and consumes significant power, so hardware designers often replace constant multiplication with a network of shifts, additions, and subtractions.

The canonical signed digit (CSD) representation is a classical tool for this transformation (Avizienis 1961). A CSD number uses the digit set $\{-1, 0, +1\}$ and forbids adjacent non-zero digits; among all signed-digit representations of an integer, the canonical form has the minimum number of non-zero digits. Each non-zero digit becomes a shifted copy of the input, and an N-digit coefficient requires $N - 1$ adders or subtractors.

The number of non-zero digits, denoted NNZ, is therefore a direct hardware cost knob. A small NNZ yields a cheap datapath but a coarse coefficient; a large NNZ yields a precise coefficient but more adders. The central design question is not whether a coefficient can be approximated, but how to control the approximation so that the worst-case output remains correct. This distinction is particularly visible in display applications: a reading is either right or wrong, and a rare worst case determines whether the panel shows the correct last digit.

This paper studies that trade-off through a concrete and familiar function, the conversion from Celsius to Fahrenheit:

$$F(C) = \frac{9}{5}C + 32 = 1.8C + 32. \tag{1}$$

The coefficient 1.8 is not a dyadic rational, so it cannot be represented exactly by a finite CSD string. A hardware implementation must therefore choose a fixed-point approximation, and the acceptable approximation error depends on the display resolution and on the temperature range. We examine two temperature regimes:

1. An air-conditioner panel with $C \in [-9, 30]$ and an integer Fahrenheit display.
2. An extreme environment with $C \in [-60, 300]$ and a display resolution of 0.001 degrees Fahrenheit.

The first regime is deliberately modest: the range is short and the display has only one significant decimal digit. The second regime tightens the resolution by three orders of magnitude and increases the range by a factor of ten, which stresses both the error budget and the datapath width.

The same methodology applies beyond a single constant. A Gaussian blur is a separable multi-coefficient FIR: a seven-tap kernel is applied horizontally and vertically, and the approximation error is the worst-case convolution error over 8-bit pixels rather than the error of one coefficient. The third design shows how the NNZ budget, kernel symmetry, and separability interact.

The paper makes five contributions. First, it formulates the problem in terms of the maximum error over the input set, and shows that the relevant rounding margin is not half an output unit but the distance from the ideal values to the nearest quantization boundary; for integer Fahrenheit this margin is 0.1, not 0.5, because $1.8C + 32$ has fractional parts in $\{0, 0.2, 0.4, 0.6, 0.8\}$. Second, it performs an NNZ-constrained search for the air-conditioner case and proves that four NNZ are sufficient and three are not. Third, it shows that a direct binary

approximation of 1.8 needs ten NNZ for the extreme case, but that a decimal-scaled formulation, $1000F = 1800C + 32000$, is exact with only three NNZ. Fourth, it extends the framework to a separable seven-tap Gaussian blur, derives a tight worst-case convolution error bound, and shows that 18 NNZ suffice for a pre-round error below 0.5 LSB and 16 for below 1 LSB. Fifth, it generates and verifies the RTL with an open-source toolchain, including exhaustive simulation over the temperature inputs and a synthesis cell count.

The remainder of the paper is organized as follows. Section 2 reviews CSD arithmetic and related work. Section 3 formulates the conversion and error model, including the multi-coefficient extension. Section 4 describes the search, the exact scaling, and the verification flow. Section 5 and Section 6 present the two temperature case studies. Section 7 extends the method to Gaussian blur. Section 8 discusses metrics, representation choice, and limitations. Section 9 concludes.

2 Background

2.1 Canonical Signed Digit Representation

A signed-digit representation expresses an integer a as

$$a = \sum_p d_p 2^p, \qquad d_p \in \{-1, 0, +1\}. \tag{2}$$

The canonical signed digit form is the unique representation with no two adjacent non-zero digits. It has the minimum possible number of non-zero digits among all signed-digit representations of a (Avizienis 1961). The CSD string is written most-significant digit first, using `+`, `-`, and `0`; for example, the integer 115 becomes

$$115 = 2^7 - 2^4 + 2^2 - 2^0, \tag{3}$$

whose CSD string is `+00-0+0-`. The integer 1800 becomes

$$1800 = 2^{11} - 2^8 + 2^3, \tag{4}$$

whose CSD string is `+00-0000+000`.

Table 1 compares binary and CSD encodings for three values used in this paper. The adder count is one less than the number of non-zero digits, because the first term is a shift and each remaining term contributes one addition or subtraction. In this accounting, shifts are free; on an FPGA they are wiring, and in an ASIC they are typically absorbed into the adder tree.

Table 1: Binary versus CSD encodings. The adder count is NNZ minus one.

Value	Binary	Ones	CSD	NNZ	Adders
28.5	`11100.1`	4	`+00-00.+`	3	2
115	`1110011`	5	`+00-0+0-`	4	3
1800	`11100001000`	4	`+00-0000+000`	3	2

2.2 Shift-Add Constant Multiplication

For an input x, a CSD coefficient a is realized as

$$ax = \sum_p d_p(x \ll p), \tag{5}$$

where $\ll$ denotes a left shift. Each non-zero digit selects a shifted copy of x, and the signs select addition or subtraction. The datapath contains no general multiplier; its critical path is the depth of the adder tree, and its area is proportional to the number of adders and to the word length.

The same idea underlies multiplierless FIR filters and common-subexpression elimination. Hartley showed how to share repeated CSD patterns across coefficients (Hartley 1996), and Dempster and Macleod formulated the minimum-adder multiplier-block problem for FIR filters (Dempster and Macleod 1995). Samueli developed search algorithms for powers-of-two filter coefficients (Samueli 1989). The textbook by Parhi provides a broad treatment of multiplierless and low-power datapath design (Parhi 1999). Gaussian smoothing is a standard separable image filter (Gonzalez and Woods 2018); separability reduces the 2D blur to two 1D passes. Exact multiplierless design is often posed as a decision problem for a fixed adder bound, solved by integer linear programming or satisfiability, and wrapped in a search over the bound (Kumm 2018; Garcia and Volkova 2023). CSD weight is a useful proxy for adder cost, but it is not adder-optimal in general (Li 1995), and exact single-constant multiplication is NP-complete (Kumm 2019). Approximate arithmetic explores a complementary direction: rather than encoding an exact constant compactly, it simplifies the arithmetic itself (Jiang et al. 2017; Liu et al. 2017; Mitchell 1962). The present study stays in the exact-arithmetic regime and uses approximation only in the coefficient value.

2.3 The csdigit Toolchain

The experiments use `csdigit`, an open-source Python library for CSD conversion and Verilog generation (Luk036 2026). The relevant functions are:

- `to_csd(value, places)` and `to_csd_i(value)` convert a decimal value to its CSD string.
- `to_csdnnz(value, nnz)` and `to_csdnnz_i(value, nnz)` return a CSD string with at most `nnz` non-zero digits.

- `to_decimal(csd)` converts a CSD string back to a number.
- `generate_csd_multiplier(csd, input_width, max_power)` returns a synthesizable Verilog module that computes the constant multiplication.

The single-coefficient generator accepts a CSD string containing only `+`, `-`, and `0`. The argument `max_power` must equal `len(csd) - 1`, and the generated output has width `input_width + max_power`. The generator emits signed shift wires and a single `assign` expression, and it factors repeated CSD patterns into a `_pat` wire when the longest repeated substring has at least two non-zero digits. A multi-coefficient variant supports FIR-style coefficient banks and cross-coefficient common-subexpression elimination. This paper uses the single-coefficient path because the application has one constant.

3 Problem Formulation

3.1 Conversion and Display Quantization

Let C be an integer temperature in degrees Celsius and let $F(C)$ be the exact Fahrenheit value in Equation 1. The display reports F at resolution q: $q = 1$ for an integer panel and $q = 0.001$ for a three-decimal panel. The ideal displayed value is the quantization of $F(C)$ to the nearest multiple of q. Equivalently, for a decimal display with d fractional digits, it is convenient to work with the integer $10^d F(C)$.

A hardware datapath computes a fixed-point approximation

$$\hat{F}(C) = \frac{aC + b}{2^k}, \tag{6}$$

where a is an integer CSD coefficient, k is the number of fractional bits, and b is a constant offset. The coefficient approximates 1.8 through the ratio $a/2^k$, and the offset implements the $+32$ term together with a rounding constant. For round-to-nearest,

$$b = 32 \cdot 2^k + 2^{k-1}. \tag{7}$$

The first term places 32 at the correct binary point; the second adds half of one least-significant bit before the final right shift.

The approximation error is

$$\varepsilon(C) = \hat{F}(C) - F(C) = \left(\frac{a}{2^k} - 1.8\right) C + \left(\frac{b}{2^k} - 32\right). \tag{8}$$

Writing $\delta = a/2^k - 1.8$ and $e = b/2^k - 32$, the error is the affine function $\varepsilon(C) = \delta C + e$. The offset e can be tuned by choosing b; this is useful because it centers the error over the input interval instead of leaving a one-sided bias.

3.2 Error Metric

The design requirement is that every displayed reading be correct. Define the maximum error over the input set $\mathcal{C}$ as

$$E_{\max} = \max_{C \in \mathcal{C}} |\varepsilon(C)|. \tag{9}$$

The admissible maximum error is not $q/2$ in general. It is the distance from the ideal values to the nearest quantization boundary. For integer Fahrenheit, $1.8C + 32$ has a fractional part in $\{0, 0.2, 0.4, 0.6, 0.8\}$, so the nearest half-integer boundary is at least 0.1 away. Hence the margin is

$$m_1 = 0.1 \quad \text{for } q = 1. \tag{10}$$

For three decimal places, the ideal values are multiples of 0.2, which are exact at the third decimal; the nearest boundary is 0.0005 away, so

$$m_{0.001} = 0.0005. \tag{11}$$

The requirement is $E_{\max} < m_q$. This is a hard constraint on the worst case, not on an average.

Mean absolute error (MAE) and mean squared error (MSE) are not sufficient for this purpose. A rare worst case can be diluted by many good readings: a design with one large error and many tiny errors can have a lower MAE and MSE than a design whose errors are uniformly moderate, even though only the first design produces a wrong reading. Section 8.1 makes this concrete.

3.3 Multi-Coefficient Extension

The Gaussian-blur case replaces the single multiplier by a 1D FIR. For kernel coefficients h_i and an integer scale $S = 2^k$, the hardware computes

$$y = \left\lfloor \frac{\sum_i a_i x_i + S/2}{S} \right\rfloor, \tag{12}$$

where a_i is the CSD approximation of Sh_i. If the input pixels are independent and lie in $[0, 255]$, the worst-case pre-round error is

$$E_{\max} = \frac{255 \max(P, N)}{S}, \qquad P = \sum_i \max(a_i - c_i, 0), \quad N = \sum_i \max(c_i - a_i, 0), \tag{13}$$

where $c_i = Sh_i$ is the target integer coefficient. The L1 bound $255 \sum_i |a_i - c_i|/S$ is conservative when the coefficient errors have mixed signs, because positive and negative errors cannot be made simultaneously worst-case by the same input. For a symmetric kernel, only half of the coefficients are independent; a pair pre-add followed by one CSD multiplier implements each symmetric pair. The feasible set of coefficient vectors with total CSD weight at most N is nested, so the monotone threshold search of Section 4.2 extends to this multi-coefficient setting.

4 Methodology

4.1 Error Budget

Because $\varepsilon(C) = \delta C + e$ is affine, its extrema over an interval occur at the endpoints. If the offset is fixed to the nominal rounding value, the budget is

$$|\delta| \max_{C \in \mathcal{C}} |C| < m_q. \tag{14}$$

If the offset is allowed to be tuned, the condition becomes

$$|\delta| \, (C_{\max} - C_{\min}) < 2m_q. \tag{15}$$

The tuned budget is looser because the offset can center the error, but it still requires the coefficient error to be small enough that the vertical span of $\varepsilon(C)$ fits inside the admissible band.

For the air-conditioner case, $C \in [-9, 30]$, $m_1 = 0.1$, and the untuned budget gives

$$|\delta| < \frac{0.1}{30} = \frac{1}{300} \approx 3.33 \times 10^{-3}. \tag{16}$$

For the extreme case, $C \in [-60, 300]$, $m_{0.001} = 0.0005$, and the untuned budget gives

$$|\delta| < \frac{0.0005}{300} \approx 1.67 \times 10^{-6}. \tag{17}$$

With offset tuning, the bound relaxes to $|\delta| < 0.001/360 \approx 2.78 \times 10^{-6}$.

4.2 NNZ-Constrained Coefficient Search

Given a budget on δ, the design problem is to find an integer a with small CSD weight and a scale k such that $a/2^k$ is close enough to 1.8. Define the candidate set and the feasibility predicate as

$$S_N = \{(k, a) : \mathrm{NNZ}(a) \le N\}, \qquad P(N) = \exists (k, a) \in S_N : E_{\max}(k, a) < m_q. \tag{18}$$

Because $S_N \subseteq S_{N+1}$, the predicate is monotone: if a design exists at N, the same design witnesses feasibility at $N+1$. This is the property that licenses a threshold search over NNZ. The formulation must use "at most N"; an exact-weight version with $\mathrm{NNZ}(a) = N$ is not monotone. Monotonicity of approximation error with the number of terms or adders has also been observed in multiplierless transform design and in the analysis of signed-power-of-two quantization (Chen et al. 2002; Yu and Lim 2007).

The implementation used in this paper enumerates all budgets because the design space is small and the per-budget table is informative:

1. Choose the input set $\mathcal{C}$ and the display margin m_q.
2. For each NNZ budget N from 1 to a small maximum, each scale k from 0 to K, and each integer a in a window around $1.8 \cdot 2^k$:

- Compute the canonical CSD string of a with `to_csd_i`.
- Skip candidates whose CSD weight exceeds N.
- Evaluate the fixed-point datapath over every $C \in \mathcal{C}$.
- Record the maximum error and the number of wrong readings.

3. Select the smallest N with zero wrong readings; break ties by maximum error and then by scale.

The window is chosen to contain every coefficient that could satisfy the error bound, so the inner search is exhaustive for the chosen K. The cost of this per-budget enumeration is $O(NKW|\mathcal{C}|)$, where W is the window size; for the cases in this paper it is a few hundred thousand input evaluations and runs in seconds.

Two alternatives follow from the same monotone predicate. If each $P(N)$ evaluation is an expensive black-box oracle, such as an integer linear program or a satisfiability solve, binary search finds the minimum N in $O(\log N)$ predicate evaluations; if no feasible upper bound is known, exponential search brackets the answer first. This is the structure used by exact multiplierless design methods, which solve a decision problem for a fixed adder bound and wrap it in a search over the bound (Kumm 2018; Garcia and Volkova 2023). At the scale of this paper, however, the candidates are already enumerated and the budget is only a filter. A single pass that records each candidate's actual CSD weight and maximum error, followed by a prefix minimum over weights, yields both the minimum N and every budget's best in $O(KW|\mathcal{C}|)$; it dominates both the per-budget sweep and bisection. We retain the per-budget enumeration for transparency and table generation, and note that binary search is preferable when the feasibility oracle is expensive and the NNZ range is large.

The search uses the canonical CSD of each candidate because the canonical form minimizes the number of non-zero digits. A non-canonical representation of the same integer would cost at least as many adders, so it cannot improve the hardware cost.

4.3 Exact Decimal-Scaled Formulation

The approximation is necessary only because $1.8 = 9/5$ has a factor of 5 in the denominator. If the display works in decimal units, that denominator can be removed by scaling the entire equation:

$$10^d F(C) = 1.8 \cdot 10^d C + 32 \cdot 10^d. \tag{19}$$

For $d = 3$, this becomes

$$1000F(C) = 1800C + 32000. \tag{20}$$

The right-hand side is an integer for every integer C. The multiplier constant is 1800, whose CSD form is `+00-0000+000` with three non-zero digits; the offset is 32000, whose CSD form has three non-zero digits but is a constant addition.

The datapath therefore uses two adders for the multiplier and one adder for the offset, for a total of three adders. No approximation error is introduced.

The same trick applies at lower resolution. For a one-decimal Fahrenheit display ($d = 1$), $10F = 18C + 320$, and $18 = 2^4 + 2^1$ has two non-zero digits. The integer panel in Section 5 cannot use this trick directly because it displays units, not tenths: dividing $18C + 320$ by 10 would reintroduce a division. The three-decimal panel, by contrast, consumes the scaled integer directly.

The scaled formulation is exact when the Celsius input is an integer and the display accepts integer units of 10^{-d} degrees Fahrenheit. If the input is itself fractional, or if the output must be a binary fixed-point value whose real-valued accuracy is judged independently of a decimal display, the scaling argument does not apply and the direct approximation of Section 4.2 is required.

4.4 RTL Generation and Verification

For a chosen coefficient, the RTL is generated by `generate_csd_multiplier`. The generator emits a module with signed input and output, one shift wire per distinct non-zero power, and a single assignment. A top-level module instantiates the multiplier, adds the offset, and performs the final right shift.

Verification has three layers. First, the Python model evaluates the datapath over the entire input set and counts wrong readings. Second, a self-checking Verilog testbench sweeps the same input set and compares the RTL output against the exact expected value. Third, Yosys is used to count the arithmetic cells after elaboration, which confirms that the synthesized structure matches the intended NNZ cost. The testbench and the Python model use the same integer formula for the expected value, so the simulation checks the RTL implementation of the datapath rather than re-checking the arithmetic model.

5 Case Study I: Air-Conditioner Display

5.1 Specification and Baseline

The panel reports integer Fahrenheit for $C \in [-9, 30]$. The exact conversion is Equation 1, and the display rounds to the nearest integer. A straightforward integer-only implementation uses a scaled constant:

```
#define K 117965   /* round(1.8 * 2^16) */
int c2f(int c) {
    return (int)(((int64_t)c * K + (1 << 15)) >> 16) + 32;
}
```

The constant $K = 117965$ has nine non-zero CSD digits, so a CSD realization costs eight adders for the multiplier plus one adder for the offset, for a total of nine adders. It is exact for the air-conditioner range: $K/2^{16} = 1.800003052$, and

the maximum error over the range is 9.2×10^{-5} degrees Fahrenheit, far below the 0.1 margin. The baseline is correct but over-engineered for the task.

5.2 NNZ Sweep

Table 2 reports the best approximation found at each NNZ budget. At one and two NNZ the coefficient is far from 1.8 and the datapath misrounds most readings. At three NNZ the best candidate is $29/16 = 1.8125$. Its coefficient error is $\delta = +0.0125$, which exceeds the budget $1/300 \approx 0.00333$ from Equation 16; the maximum error is 0.375 degrees Fahrenheit, and 7 of 40 readings are wrong. Tuning the offset reduces the count to 5 of 40 but cannot eliminate the failures because the error span is too large.

Table 2: NNZ sweep for the air-conditioner range −9 to 30 degrees Celsius with integer Fahrenheit display.

NNZ	$a/2^k$	Value	$\max_C \lvert\varepsilon(C)\rvert$	Wrong / 40
1	2 / 1	2.0	6.0	35
2	7 / 4	1.75	1.5	21
3	29 / 16	1.8125	0.375	7
4	**115 / 64**	**1.796875**	**0.09375**	**0**
5	461 / 256	1.80078125	0.0234	0
6	1843 / 1024	1.79980469	0.00586	0

At four NNZ the search finds $115/64 = 1.796875$. Its coefficient error is $\delta = -0.003125$, the maximum error is 0.09375 degrees Fahrenheit, and all 40 readings are correct. The value 115 factors as

$$115 = 2^7 - 2^4 + 2^2 - 2^0, \tag{21}$$

with CSD string `+00-0+0-` and four non-zero digits.

Four NNZ is minimal for this case. The closest three-term dyadic approximations to 1.8 are $29/16 = 1.8125$ and $57/32 = 1.78125$; their coefficient errors are $+0.0125$ and -0.01875, both far above the $1/300$ budget. No three-term coefficient can therefore satisfy the margin, while Equation 21 provides a four-term coefficient that does. The search over $k \le 12$ and a ± 64 coefficient window confirms that no smaller-NNZ candidate passes.

5.3 Datapath and RTL

The selected datapath is

$$F(C) = \left\lfloor \frac{115C + 2080}{64} \right\rfloor, \qquad 2080 = 32 \cdot 2^6 + 2^5. \tag{22}$$

The constant 2080 folds the +32 offset and the rounding term 2^5 into one addition. The multiplier contributes three adders (four non-zero digits), and the offset contributes one, for a total of four arithmetic cells. The final shift is free.

The generated multiplier is:

```
module csd_multiplier (
    input  signed [7:0]  x,
    output signed [14:0] result
);
    wire signed [14:0] x_shift7 = x <<< 7;
    wire signed [14:0] x_shift4 = x <<< 4;
    wire signed [14:0] x_shift2 = x <<< 2;
    wire signed [14:0] x_shift0 = x <<< 0;

    assign result = x_shift7 - x_shift4 + x_shift2 - x_shift0;
endmodule
```

The top level adds the offset and shifts:

```
wire signed [14:0] product;
csd_multiplier mul (.x(celsius), .result(product));
localparam signed [15:0] OFFSET = 16'sd2080;
assign fahrenheit = (product + OFFSET) >>> 6;
```

The input range -9 to 30 fits in an 8-bit signed word, and the product width is $8 + 7 = 15$ bits. A self-checking testbench sweeps all 40 inputs and compares the output against $(18C + 325)/10$, the integer form of the correctly rounded value. Icarus Verilog reports zero mismatches. Yosys reports one adder and two subtractors in the multiplier and one additional adder in the top level, for four arithmetic cells, matching the NNZ accounting.

6 Case Study II: Extreme Range and Three Decimal Places

6.1 Specification and Direct Approximation

The second panel covers $C \in [-60, 300]$ and reports three decimal places, so $q = 0.001$ and $m_q = 0.0005$. The range is ten times wider than the air-conditioner range, and the resolution is three orders of magnitude finer. The untuned budget is $|\delta| < 1.67 \times 10^{-6}$ from Equation 17; with offset tuning it relaxes to 2.78×10^{-6}.

The direct binary approximation uses the alternating CSD expansion of 1.8:

$$1.8 = 2 - 2^{-2} + 2^{-4} - 2^{-6} + \cdots . \tag{23}$$

The nine-term partial sum is 1.800003052, with $\delta = 3.05 \times 10^{-6}$. This is just above the tuned budget 2.78×10^{-6}; even after centering the offset, the maximum

error is 5.49×10^{-4} degrees Fahrenheit, and 34 of 361 readings are wrong. The ten-term partial sum is 1.799999237, with $\delta = -7.63 \times 10^{-7}$; its maximum error is 1.37×10^{-4}, and all 361 readings are correct. Table 3 compares the two.

Table 3: Direct binary approximation of 1.8 for the extreme range. The maximum error is measured after tuning the offset.

NNZ	Partial sum	δ	Max error at 300 C (F)	Wrong / 361
9	1.800003052	3.05×10^{-6}	5.49×10^{-4}	34
10	**1.799999237**	7.63×10^{-7}	1.37×10^{-4}	**0**

Within the searched design space ($k \leq 24$ and all coefficients in the feasibility window), no nine-NNZ candidate satisfies the bound; the best one is the nine-term partial sum of Equation 23. Ten NNZ is the smallest design found that meets the requirement. The ten-NNZ coefficient is $471859/2^{18}$; it has nine adders for the multiplier plus one for the offset, for ten arithmetic cells. The product width is $10 + 18 = 28$ bits, and the output retains 18 fractional bits before conversion to decimal.

6.2 Exact Decimal-Scaled Design

The extreme case admits a much cheaper exact solution. Scaling by 1000 gives Equation 20:

$$1000F(C) = 1800C + 32000.$$

The multiplier constant is $1800 = 2^{11} - 2^8 + 2^3$, with CSD string `+00-0000+000` and three non-zero digits. The offset $32000 = 2^{15} - 2^9 - 2^8$ is a constant addition. The datapath is

```
assign result = x_shift11 - x_shift8 + x_shift3;   /* 1800 */
assign f_milli = product + 21'sd32000;             /* thousandths */
```

with three arithmetic cells in total: two for the multiplier and one for the offset. The input range -60 to 300 fits in a 10-bit signed word; the product width is $10 + 11 = 21$ bits; and the output spans -76000 to 572000 thousandths, which is -76.000 to 572.000 degrees Fahrenheit. A testbench over all 361 inputs reports zero mismatches.

The reason the scaling works is representation-theoretic rather than arithmetic: the display unit is a power of ten, and 10^3 cancels the factor of 5 in 9/5. The binary approximation, by contrast, must approximate the infinite CSD expansion of 1.8 and pays for the irrational-in-base-two denominator with additional non-zero digits. The scaled design is exact, uses three adders instead of ten, and has a smaller multiplier coefficient.

6.3 Comparison

Table 4 summarizes the designs. The air-conditioner case improves from nine adders in the baseline to four adders with the same exact behavior. The extreme direct approximation needs ten adders; the exact scaled design needs three. The exact scaled design is therefore more than three times cheaper than the ten-NNZ approximation while also being exact.

Table 4: Design comparison. Adders include one constant adder for the offset.

Case	Design	NNZ	Adders	Max error (F)	Correct
AC	baseline K	9	9	9.2×10^{-5}	yes
AC	CSD 115/64	4	4	0.094	yes
Extreme	binary 9	9	9	5.49×10^{-4}	no (34/361)
Extreme	binary 10	10	10	1.37×10^{-4}	yes
Extreme	scaled 1800	**3**	**3**	**0**	yes

7 Case Study III: Gaussian Blur

7.1 Kernel and Error Metric

Gaussian smoothing is a standard separable image filter (Gonzalez and Woods 2018). We use a seven-tap Gaussian with $\sigma = 2$ and scale $S = 1024$; the integer kernel is

$$(c_0, \ldots, c_6) = (72, 134, 195, 222, 195, 134, 72), \qquad \sum_i c_i = 1024, \tag{24}$$

where the center coefficient is 222. A separable 2D blur applies this 1D kernel horizontally and vertically, so the 1D arithmetic cost is paid twice.

The exact kernel has CSD weights $(2, 3, 4, 3, 4, 3, 2)$ and 21 NNZ. Using the multi-coefficient error metric of Section 3.3, we search for a symmetric approximation with total CSD weight below 21. The independent coefficients are (a_0, a_1, a_2, a_3) for the center and the $\pm1, \pm2, \pm3$ taps; the full kernel is $(a_3, a_2, a_1, a_0, a_1, a_2, a_3)$, and the normalization constraint is $a_0 + 2(a_1 + a_2 + a_3) = 1024$. The search is exhaustive over a ±32 window and solves the last coefficient from the normalization constraint. Table 5 reports the exact kernel, the design with a pre-round error below 0.5 LSB, and the design below 1 LSB. The table lists the four independent weights; a weight w at a non-center tap contributes $2w$ to the total NNZ.

Table 5: Gaussian-blur CSD designs. The pre-round error is $255\max(P,N)/1024$; the 1D adder count includes three pair pre-adds, the coefficient adders, three product adds, and one rounding add.

Design	Coefficients	Weights	NNZ	Pre-round error	1D/2D adders
exact	(222,195,134,72)	(3,4,3,2)	21	0	15/30
< 0.5	(224,194,134,72)	(2,3,3,2)	18	0.498	13/26
< 1	(224,196,132,72)	(2,3,2,2)	16	0.996	12/24

The 18-NNZ design keeps the positive and negative error sums at $P = N = 2$, for a bound of $255 \cdot 2/1024 = 0.498$ LSB. The 16-NNZ design has $P = N = 4$ and a bound of 0.996 LSB. The exact kernel needs 21 NNZ. The 0.5 LSB bound is pre-round; near a rounding boundary the rounded output can differ by one LSB, which is acceptable for an 8-bit blur but should not be described as exact integer equality.

7.2 RTL and Verification

The 18-NNZ design is emitted as a symmetric 1D FIR:

```
wire signed [10:0] s3 = x0 + x6;
wire signed [18:0] total = p3 + p2 + p1 + p0;
wire signed [18:0] rounded = (total + 19'sd512) >>> 10;
```

The three pair pre-adds, six coefficient adders, three product additions, and one rounding addition total 13 arithmetic cells per 1D pass. The separable 2D filter uses two passes, for 26 adders, excluding line buffers and registers. The non-symmetric direct form would use 18 adders per pass and 36 for 2D.

The RTL is generated from four `generate_csd_multiplier` modules, renamed for the four coefficients, and a top-level wrapper. A self-checking testbench covers constant images, impulses, alternating patterns, and deterministic random vectors; Icarus Verilog reports 32/32 matches against the CSD kernel. Yosys reports 10 adders and 3 subtractors, for 13 arithmetic cells, matching the structural count. The Python model also compares the rounded output with the exact Gaussian kernel on the test vectors; the observed difference is at most one LSB, consistent with the 0.498 LSB pre-round bound.

8 Discussion

8.1 Maximum Error versus Average Error

The choice of error metric is not cosmetic. Table 6 gives an illustrative pair of designs with the same 40 readings. Design A has 39 readings at an error of 0.001 degrees Fahrenheit and one reading at 0.20; design B has every reading at 0.05.

Design A has the lower MAE and the lower MSE, yet it produces one wrong reading because its maximum error exceeds the 0.1 margin. Design B has the higher average error but no wrong readings.

Table 6: Illustrative example: lower MAE and MSE do not imply correct rounding.

Design	MAE	MSE	Max error	Wrong
A	0.006	0.0010	0.20	1
B	0.050	0.0025	0.05	0

Mean-based metrics reward a design that is usually good and occasionally wrong. For a display, "occasionally wrong" is a correctness failure. The maximum error is the quantity that determines whether any reading crosses a quantization boundary, so it is the appropriate constraint for this application. The same argument applies to any hardware whose output is compared against an exact specification: the worst case, not the average, is the binding constraint.

8.2 Representation, Range, and Cost

The three case studies suggest a design procedure. First, determine the display unit and the input domain. Second, compute the quantization margin m_q, remembering that it is the distance from the ideal values to a boundary, not automatically $q/2$. Third, check whether a decimal-scaled formulation makes the problem exact; if it does, the multiplier constant is the scaled integer and the NNZ is determined by that integer rather than by an approximation. Fourth, if scaling is unavailable, choose the smallest NNZ that satisfies the maximum-error budget, and verify the choice over the entire input set. For a multi-coefficient FIR, replace the single margin by the worst-case convolution error of Equation 13, and share the NNZ budget across the coefficients before applying the same monotone threshold search.

The range enters the direct approximation linearly: the admissible coefficient error is inversely proportional to the span $C_{\max} - C_{\min}$. The display resolution enters through m_q. For the alternating CSD expansion, each additional pair of terms reduces the coefficient error by a factor of four, so the required NNZ grows logarithmically with the precision-to-range ratio. The exact scaled design avoids this growth entirely when the display unit is a power of ten. In the Gaussian case, the pixel range enters through the maximum value in Equation 13, and symmetry and separability determine the arithmetic cost rather than the coefficient count alone.

The hardware cost model used here counts adders as NNZ minus one, plus one for a constant offset, and treats shifts as free. This is the standard first-order model for multiplierless arithmetic. It ignores adder width, carry-chain delays, and the cost of the final decimal conversion. The product width is `input_width`

`+ max_power`, so a wider range or a larger coefficient increases the width even when the NNZ is unchanged. For a multi-coefficient FIR, the coefficient adders are summed, and pair pre-adds, product accumulation, rounding, and line buffers are additional costs; the Gaussian case uses this extended model. A full ASIC or FPGA evaluation would need timing and area reports; the Yosys cell count in this paper is a structural check, not a placed-and-routed cost.

The method has several limitations. The Celsius input is assumed to be an integer. If the input has fractional bits, the scaled formulation must be revisited: the factor 10^d no longer removes the denominator for every input, and a rounding step may reappear. The study includes one multi-coefficient FIR; larger filter banks can share subexpressions across coefficients, which changes the cost model. The search for the ten-NNZ design is exhaustive only within the stated bounds; a larger search could in principle find a different nine-NNZ representation, although the alternating expansion's error is already above the tuned bound and the search found no better candidate. The Gaussian case fixes a seven-tap kernel, an 8-bit input range, and a bounded ± 32 search window; image-boundary handling and line-buffer registers are not modeled, and the 2D adder count is an arithmetic estimate rather than a placed-and-routed cost. Moreover, NNZ is a proxy for adder count: adder-graph topologies can realize a coefficient with fewer adders than the CSD weight suggests (Li 1995), and exact single-constant multiplication is NP-complete (Kumm 2019). The monotone threshold result in Section 4.2 applies to NNZ, not to the true minimum adder count. Finally, the verification is functional; it does not prove timing closure or metastability behavior.

8.3 Threats to Validity

The main internal threat is the choice of the display model. The margin analysis assumes that the ideal values are quantized to the nearest multiple of q and that the panel shows that quantized value. A panel with truncation instead of rounding would shift the boundary and change the required precision. The external threat is the hardware cost model: NNZ is a proxy for area and power, and the proxy ignores fan-out, wiring, and the decimal conversion logic. The constructive designs (the four-NNZ, three-NNZ, and 18-NNZ filters) are independent of these modeling choices; the negative result for three NNZ in the air-conditioner case depends on the rounding model. The multi-coefficient error bound assumes independent pixels; boundary pixels and image-border normalization are not modeled.

9 Conclusion

This paper used two temperature-conversion designs and a separable Gaussian blur as case studies in controllable hardware approximation with CSD multipliers. For an air-conditioner panel over -9 to 30 degrees Celsius with integer Fahrenheit output, four NNZ are sufficient and minimal: the coefficient $115/64$ yields the datapath $F = (115C + 2080) \gg 6$ with four adders and zero wrong readings,

while three NNZ misrounds 7 of 40 readings. For an extreme panel over -60 to 300 degrees Celsius with three decimal places, a direct binary approximation needs ten NNZ and ten adders; nine NNZ misrounds 34 of 361 readings. Scaling the equation by the display unit, $1000F = 1800C + 32000$, makes the problem exact with only three NNZ and three adders. For a seven-tap Gaussian blur at scale 1024, the exact kernel needs 21 NNZ; 18 NNZ keep the worst-case pre-round error below 0.5 LSB and 16 below 1 LSB, and the 18-NNZ 1D filter uses 13 adders per pass (26 for the separable 2D blur).

The broader lesson is that the representation should be chosen before the approximation is optimized. Decimal scaling can remove a rational denominator and turn an approximation problem into an exact integer problem; when it is unavailable, the maximum error over the input set, not MAE or MSE, is the correct design constraint. The same reasoning extends to multi-coefficient FIRs when the error is measured by the worst-case convolution, not by a per-coefficient average. The generated RTL was verified exhaustively in Icarus Verilog and checked structurally in Yosys, so the reported adder counts correspond to the actual generated hardware rather than to a spreadsheet model.

Future work includes extending the search to larger multi-coefficient FIR banks with LCSRe and cross-coefficient common-subexpression elimination, adding line-buffer and memory costs for 2D image filters, incorporating timing and area from a placed design, and handling fractional Celsius inputs where the decimal scaling argument no longer applies.

10 Reproducibility

The air-conditioner design is reproduced by the demo script in the `csdigit` repository:

```
python examples/temperature_c_to_f.py
```

The script prints the NNZ sweep, writes `temperature_c_to_f.v` and its testbench, and runs Icarus Verilog when `iverilog` and `vvp` are available. The Gaussian-blur design is reproduced by

```
python examples/gaussian_blur_csd.py
```

which prints the NNZ and adder sweep, writes `gaussian_blur_1d.v` and its testbench, and runs Icarus Verilog when available. The exact extreme-range multiplier is generated by

```python
from csdigit.csd import to_csd_i
from csdigit.csd_multiplier import generate_csd_multiplier

csd = to_csd_i(1800)                  # '+00-0000+000'
verilog = generate_csd_multiplier(csd, input_width=10, max_power=len(csd) - 1)
```

and the top level adds `32000` before the display interprets the result as thousandths. The experiments used Python 3.14, `csdigit` at the v0.5 development revision, Icarus Verilog, and Yosys. The paper itself is built with Pandoc 3.11, `pandoc-crossref`, and the default Pandoc CSL; the build is offline.

References

Avizienis, Algirdas. 1961. "Signed-Digit Number Representations for Fast Parallel Arithmetic." *IRE Transactions on Electronic Computers* EC-10 (3): 389–400.

Chen, Yuan, Soontorn Oraintara, Trac D. Tran, Kevin Amaratunga, and Truong Q. Nguyen. 2002. "Multiplierless Approximation of Transforms with Adder Constraint." *IEEE Signal Processing Letters* 9 (11): 344–47. https://doi.org/10.1109/LSP.2002.804419.

Dempster, Andrew G., and Malcolm D. Macleod. 1995. "Use of Minimum-Adder Multiplier Blocks in FIR Digital Filters." *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 42 (9): 569–77.

Garcia, Remi, and Anastasia Volkova. 2023. "Toward the Multiple Constant Multiplication at Minimal Hardware Cost." *IEEE Transactions on Circuits and Systems I: Regular Papers* 70 (5): 1976–88. https://doi.org/10.1109/TCSI.2023.3241859.

Gonzalez, Rafael C., and Richard E. Woods. 2018. *Digital Image Processing*. 4th ed. Pearson.

Hartley, Richard I. 1996. "Subexpression Sharing in Filters Using Canonic Signed Digit Multipliers." *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 43 (10): 677–88.

Jiang, Honglan, Chenghua Liu, Leibo Liu, Fabrizio Lombardi, and Jie Han. 2017. "A Review, Classification and Comparative Evaluation of Approximate Arithmetic Circuits." *ACM Journal on Emerging Technologies in Computing Systems* 13 (4): 1–37.

Kumm, Martin. 2018. "Optimal Constant Multiplication Using Integer Linear Programming." *IEEE Transactions on Circuits and Systems II: Express Briefs* 65 (5): 567–71. https://doi.org/10.1109/TCSII.2018.2823780.

Kumm, Martin. 2019. *Design of Optimal Multiplierless FIR Filters*. https://arxiv.org/abs/1912.04210.

Li, H. 1995. "Minimum Number of Adders for Implementing a Multiplier

and Its Application to the Design of Multiplierless Digital Filters." *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 42 (7): 453–60. https://doi.org/10.1109/82.401168.

Liu, Weiqiang, Liang Qian, Chenghua Wang, Honglan Jiang, Jie Han, and Fabrizio Lombardi. 2017. "Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing." *IEEE Transactions on Computers* 66 (8): 1435–41.

Luk036. 2026. *csdigit: Canonical Signed Digit Conversion and CSD Multiplier Generation*. https://github.com/luk036/csdigit.

Mitchell, J. N. 1962. "Computer Multiplication and Division Using Binary Logarithms." *IRE Transactions on Electronic Computers* EC-11 (4): 512–17.

Parhi, Keshab K. 1999. *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley.

Samueli, Henry. 1989. "An Improved Search Algorithm for the Design of Multiplierless FIR Filters with Powers-of-Two Coefficients." *IEEE Transactions on Circuits and Systems* 36 (7): 1044–47.

Yu, Y. J., and Y. C. Lim. 2007. "Roundoff Noise Analysis of Signals Represented Using Signed Power-of-Two Terms." *IEEE Transactions on Signal Processing* 55 (9): 4567–73. https://doi.org/10.1109/TSP.2007.893216.