Controllable Hardware Approximation with CSD Multipliers

Temperature conversion and Gaussian blur case studies

Luk036

2026

Section 1

Overview

Roadmap

Part I: Background
- Constant multiplication by $9/5$
- Canonical signed digit (CSD)
- Shift-add cost

Part II: Error budget
- Fixed-point model
- Maximum error
- Monotone NNZ search

Part III: Case studies
- Air-conditioner display
- Extreme range, three decimals
- Exact decimal scaling
- Gaussian blur, multi-coefficient

Part IV: Lessons
- Design flow
- NNZ as a proxy
- Verification

Section 2

Motivation

The constant-multiplication problem

- Celsius to Fahrenheit:

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

- 1.8 is not dyadic: no finite CSD string represents it exactly.
- A general multiplier is expensive; shifts and adds are cheap.

Panel	Range	Resolution
Air conditioner	-9 to 30 C	integer F
Extreme	-60 to 300 C	0.001 F

*The problem is not accuracy, but how to **control** it.*

Three ways to control the approximation

Integer-only arithmetic

- $K = \mathrm{round}(1.8 \cdot 2^{16}) = 117965$
- 9 NNZ, 9 adders
- Exact for the AC range

Approximate multiplier

- Gate removal; design-space exploration
- Empirical maximum error
- Difficult to bound

CSD multiplier

- NNZ is a direct hardware knob
- Adders $= \mathrm{NNZ} - 1$
- Exhaustive or monotone search
- Provable maximum error

The design question is **how many NNZ** are sufficient.

Error metric: max, not mean

- MAE and MSE average the worst case away.
- A rare outlier can hide behind many good readings.
- The display fails only when the worst case crosses a rounding boundary.

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

- Design A has lower MAE and MSE, yet produces a wrong reading.
- Optimize $\max_C |\varepsilon(C)|$, not an average.

Section 3

CSD Background

Canonical signed digit

- Digits $\{-1, 0, +1\}$; no two adjacent non-zero digits.
- The canonical form is unique and has minimal weight.
- NNZ is the number of non-zero digits; adders $= \mathrm{NNZ} - 1$.

$$28.5 = 2^5 - 2^2 + 2^{-1} = 32 - 4 + 0.5$$

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

Shift-add cost

$$ax = \sum_p d_p(x \ll p).$$

- $115 = 2^7 - 2^4 + 2^2 - 2^0$: 4 NNZ, 3 adders.
- $1800 = 2^{11} - 2^8 + 2^3$: 3 NNZ, 2 adders.
- CSD weight is a proxy: adder graphs can use fewer adders (Li 1995; Kumm 2019).
- Sharing across coefficients: LCSRe and common-subexpression elimination (Hartley 1996; Dempster and Macleod 1995).
- Exact decision procedures: ILP and SAT (Kumm 2018; Garcia and Volkova 2023).

Section 4

Error Budget and Search

Fixed-point model

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

- Round-to-nearest: $b = 32 \cdot 2^k + 2^{k-1}$.
- Worst-case requirement: $\max_C |\varepsilon(C)| < m$.
- Integer F: $m = 0.1$, because the fractional parts are $0, 0.2, \ldots, 0.8$.
- 0.001 F: $m = 0.0005$.
- Error budget: $|\delta| \max|C| < m$; a tuned offset relaxes it.

Monotone feasibility in NNZ

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

- $S_N \subseteq S_{N+1}$ implies P is monotone: use **at most** N, not exactly N.
- Binary search: $O(\log N)$ evaluations; exponential search if no upper bound.
- Expensive oracle (ILP/SAT): bisection is the right wrapper.
- Enumerated candidates: one pass, bucket by actual NNZ, prefix minimum; $O(KW|C|)$.

Method	Predicate calls	Per-budget table
Exhaustive per budget	N	yes
Binary search	$\log N$	no
One pass	1	yes

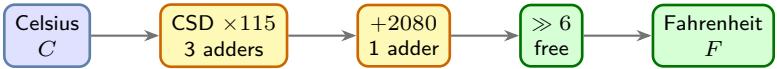
AC example: $4{,}727 \to 3{,}019 \to 1{,}401$ candidate evaluations.

AC case: NNZ sweep

NNZ	$a/2^k$	Value	Max error	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	1.800781	0.0234	0
6	1843 / 1024	1.799805	0.00586	0

- 4 NNZ is minimal: the best three-term dyadics are $29/16$ and $57/32$, both with $|\delta| \geq 0.0125$.
- $115 = 2^7 - 2^4 + 2^2 - 2^0$.

AC case: the 4-NNZ datapath



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

- Three adders for 115, one for the offset: four arithmetic cells.

AC case: RTL and verification

```
assign result = x_shift7 - x_shift4 + x_shift2 - x_shift0;
assign fahrenheit = (product + 16'sd2080) >>> 6;
```

- Icarus Verilog: all 40 readings correct.
- Yosys: two adders and two subtractors, for four arithmetic cells.
- The Python exhaustive sweep and the RTL agree.

Section 5

Extreme Range

Direct binary approximation

- -60 to 300 C, 0.001 F; $|\delta| < 1.67 \times 10^{-6}$ (2.78×10^{-6} with offset tuning).
- $1.8 = 2 - 2^{-2} + 2^{-4} - 2^{-6} + \cdots$.

NNZ	δ	Max error	Wrong / 361
9	3.05×10^{-6}	5.49×10^{-4}	34
10	7.63×10^{-7}	1.37×10^{-4}	**0**

- Ten NNZ: nine adders plus one offset, for ten arithmetic cells.
- Nine NNZ is just above the tuned budget.

Exact decimal scaling

- Work in thousandths of a degree:

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

- $1800 = 2^{11} - 2^8 + 2^3$: 3 NNZ, 2 adders.
- Three adders total: two for 1800 and one for 32000.
- Exact for every integer C; no approximation error.
- 10-bit input, 21-bit product, output -76000 to 572000 thousandths.
- Icarus Verilog: all 361 readings correct.

Temperature trade-off summary

Case	Design	NNZ	Adders	Max error	Correct
AC	baseline	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}	34/361
Extreme	binary 10	10	10	1.37×10^{-4}	yes
Extreme	scaled 1800	**3**	**3**	**0**	yes

- Exact scaling: three adders instead of ten.
- AC: four adders instead of nine.

Section 6

Gaussian Blur

Gaussian blur: kernel and error bound

- Seven-tap Gaussian, $\sigma = 2$, scale $S = 1024$.
- Exact integer kernel: $(72, 134, 195, 222, 195, 134, 72)$, sum 1024.
- Separable 2D blur = horizontal + vertical 1D pass.
- Worst-case pre-round error:

$$E_{\max} = \frac{255 \max(P, N)}{S},$$

  where P and N are the positive and negative coefficient-error sums.
- The L1 bound is conservative when coefficient errors have mixed signs.

Gaussian blur: NNZ results

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

- 18 NNZ for a pre-round error below 0.5 LSB; 16 NNZ below 1 LSB; exact kernel 21.
- 0.498 LSB is pre-round; rounded outputs can differ by 1 LSB near a boundary.

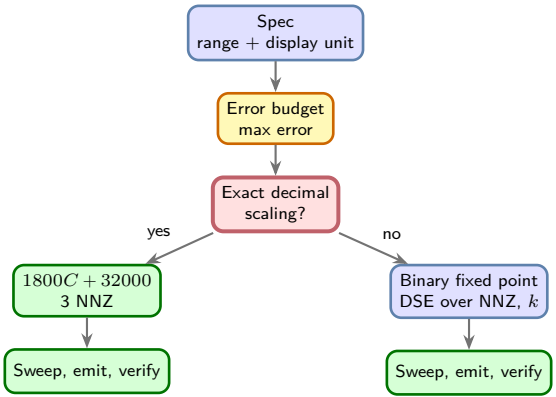
Gaussian blur: hardware

- Symmetric 1D FIR: 3 pair pre-adds + 6 coefficient adders + 3 product adds + 1 round = **13 adders per pass**.
- Separable 2D: 26 adders, excluding line buffers; direct form 18/36.
- Icarus Verilog: 32/32 vectors match the CSD kernel.
- Yosys: 10 adders and 3 subtractors = 13 cells.
- Coefficients: 72, 134, 194, 224.

Section 7

Lessons

Design flow


Spec
range + display unit
Error budget
max error
Exact decimal
scaling?
yes
no
$1800C + 32000$
3 NNZ
Binary fixed point
DSE over NNZ, k
Sweep, emit, verify
Sweep, emit, verify

Lessons

- Control, not raw accuracy: choose the error budget first.
- Maximum error, not MAE or MSE.
- Choose the representation before optimizing NNZ: decimal scaling can be exact.
- NNZ is a proxy; adder graphs can use fewer adders.
- Multi-coefficient kernels: use $\max(P, N)$; symmetry and separability change the adder count.
- Feasibility is monotone in NNZ: bisection or one pass.
- Verify the generated RTL with a simulator and synthesis.

Reproducibility

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

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

csd = to_csd_i(1800)
verilog = generate_csd_multiplier(csd, input_width=10, max_power=len(csd) - 1)
```

- csdigit v0.5, Icarus Verilog, Yosys, Python 3.14.

Key references

- Avizienis (1961): signed-digit representations.
- Hartley (1996); Dempster and Macleod (1995): subexpression sharing and minimum-adder blocks.
- Kumm (2018); Garcia and Volkova (2023): exact MCM via ILP and SAT.
- Chen et al. (2002); Yu and Lim (2007): monotonicity of error with term or adder budget.
- Li (1995); Kumm (2019): CSD weight is not adder-optimal; exact SCM is NP-complete.
- Gonzalez and Woods (2018): Gaussian smoothing and separable image filtering.
- csdigit: github.com/luk036/csdigit.

Questions

Questions?