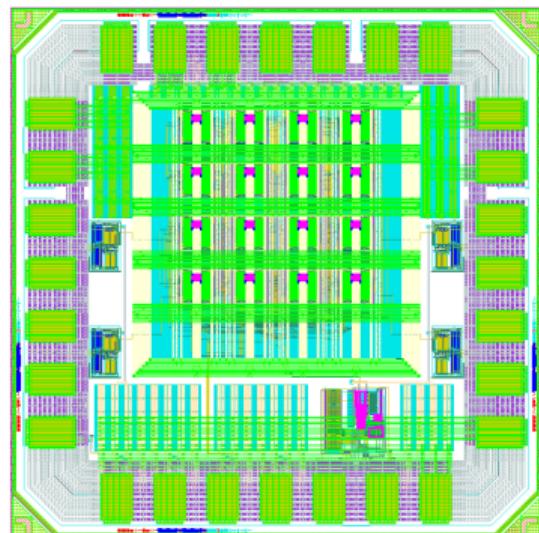
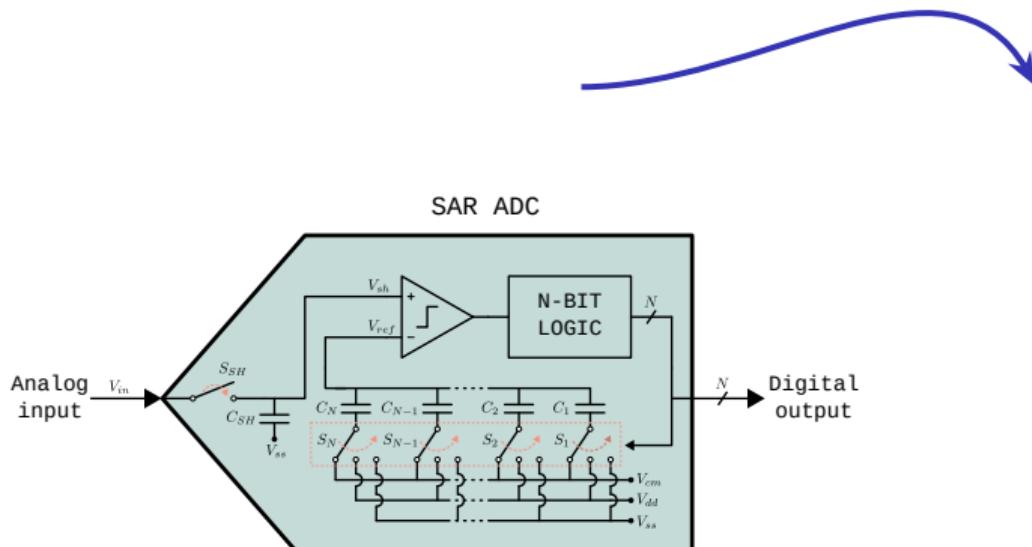


FRIDA Design Review

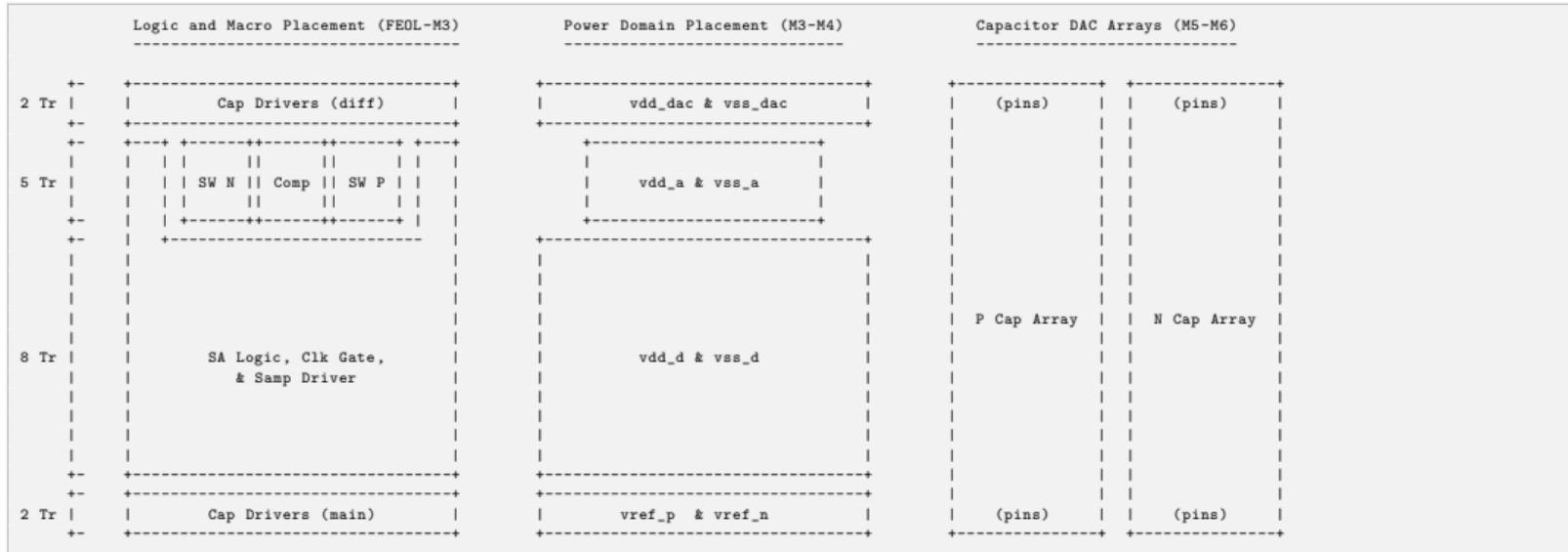
Kennedy Caisley

Wednesday, 28 January 2026

FRIDA Chip Overview

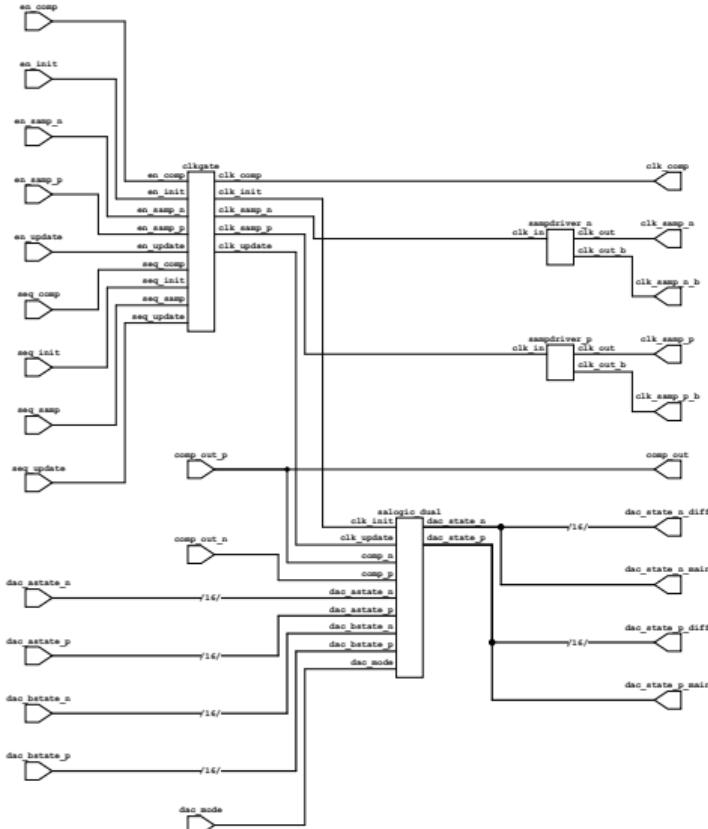


Sketch of ADC Floorplan



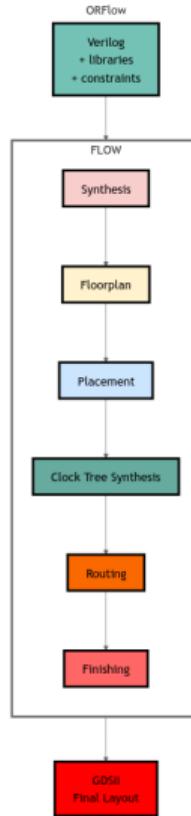
ADC Digital Block Netlist

- ▶ **salogic**: SAR sequencing FSM
- ▶ **clkgate**: Clock gating for power
- ▶ **sampdriver_p/n**: Sampling switch drivers
- ▶ Symmetric P/N halves for differential operation
- ▶ **salogic** and **clkgate** shared between halves



OpenROAD Flow Scripts (ORFS)

- ▶ **Yosys:** RTL synthesis
 - ▶ Verilog parsing and elaboration
 - ▶ Logic optimization (ABC)
 - ▶ Technology mapping to standard cells
- ▶ **OpenROAD:** Physical design
 - ▶ Floorplanning and PDN
 - ▶ Placement (global + detailed)
 - ▶ Clock tree synthesis
 - ▶ Routing (global + detailed)
 - ▶ Timing analysis (OpenSTA)
- ▶ **KLayout:** Final verification
 - ▶ DRC and LVS checks
 - ▶ GDS export



Yosys: Step 1 - RTL Synthesis

Input Files

- ▶ adc_digital.v (top module)
- ▶ clkgate.v, salogic.v, sampdriver.v
- ▶ cells_tsmc65.v (cell wrappers)

Synthesis Statistics

- ▶ Total cells: 350
- ▶ Total area: $893.88 \mu\text{m}^2$
- ▶ Sequential elements: 49% of area
- ▶ 48 flip-flops (DFQD1, EDFD2)
- ▶ 5 clock gates (CKLNQD1LVT)

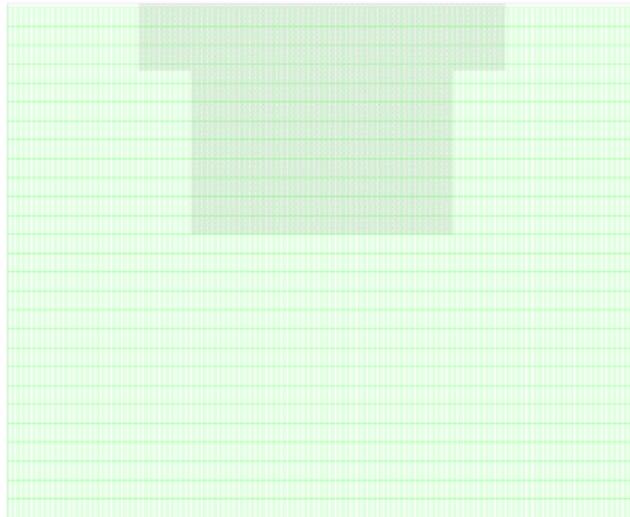
Technology Mapping (ABC)

Module	Cells	Area
adc_digital	38	54.36
clkgate	6	33.48
salogic	302	798.84
sampdriver	2	3.60
Total	350	893.88

Key Cell Types

EDFD2LVT (edge DFF)	32	334.08
DFQD1LVT (DFF)	16	109.44
NR2D0LVT (NOR2)	93	133.92
BUFFD0LVT (buffer)	69	99.36
CKLNQD1LVT (clkgate)	5	32.40

OpenROAD Step 2.1: Floorplan



- ▶ Die and core area: $60\mu\text{m} \times 49\mu\text{m}$
- ▶ Row height: $2.8\mu\text{m}$ (17 rows)
- ▶ Blockages reserve space for analog macros
- ▶ I/O pin and routing grid ($0.2\mu\text{m}$ pitch)

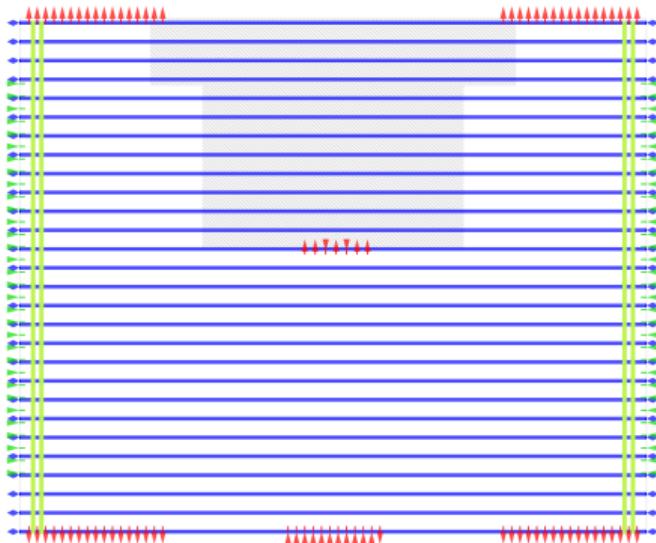
config.mk

```
export DIE_AREA = 0 0 60 49
export CORE_AREA = 0 0 60 49
export CREATE_BLOCKAGES = .../create_blockages.tcl
```

create_blockages.tcl

```
# Reserve area for comparator
create_blockage -region {17.5 27.0 42.5 49}
# Reserve areas for sampling switches
create_blockage -region {12.5 42.6 21 49}
create_blockage -region {39 42.6 47.5 49}
```

OpenROAD Step 2.2: Power Distribution Network



- ▶ M4 vertical stripes at edges for vdd_d/vss_d
- ▶ M1 horizontal stripes follow cell rows
- ▶ M1-M4 connections for power delivery

pdn.tcl

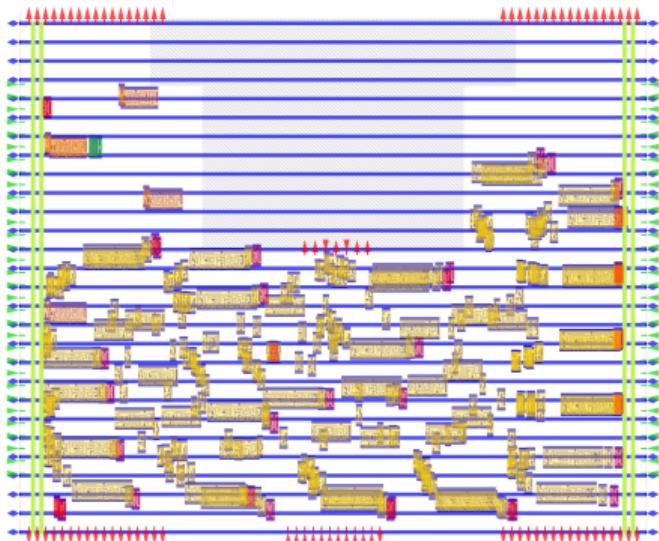
```
add_global_connection -net {vdd_d} \
    -inst_pattern {*} -pin_pattern VDD -power
add_global_connection -net {vss_d} \
    -inst_pattern {*} -pin_pattern VSS -ground

define_pdn_grid -name "Core" -pins {M4}

add_pdn_stripe -grid "Core" -layer M4 \
    -width 0.4 -offset 1.3 -nets {vdd_d}
add_pdn_stripe -grid "Core" -layer M4 \
    -width 0.4 -offset 2.1 -nets {vss_d}

add_pdn_stripe -grid "Core" -layer M1 \
    -followpins -width 0.33
add_pdn_connect -grid "Core" -layers {M1 M4}
```

OpenROAD Step 3.1: Global Placement



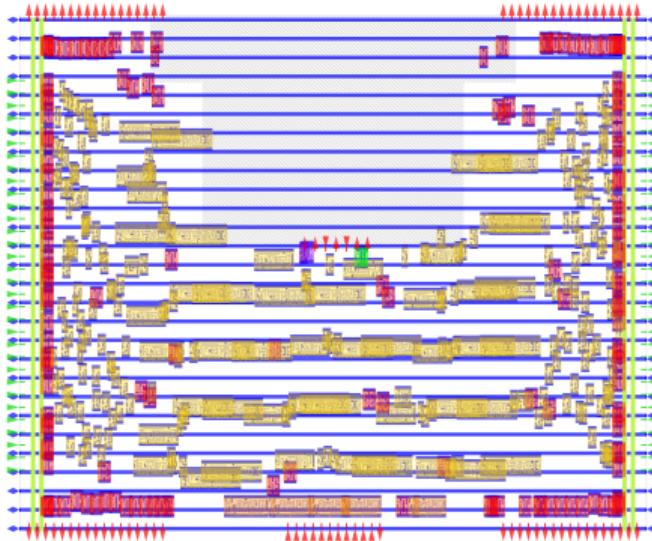
- ▶ Initial cell location doesn't consider I/O
- ▶ Instead routability-driven placement
- ▶ 60% target placement density
- ▶ Placement avoids blockages

config.mk

```
# Standard cell placement density
export PLACE_DENSITY = 0.50

# I/O pin layers
export IO_PLACER_H = M3
export IO_PLACER_V = M2
```

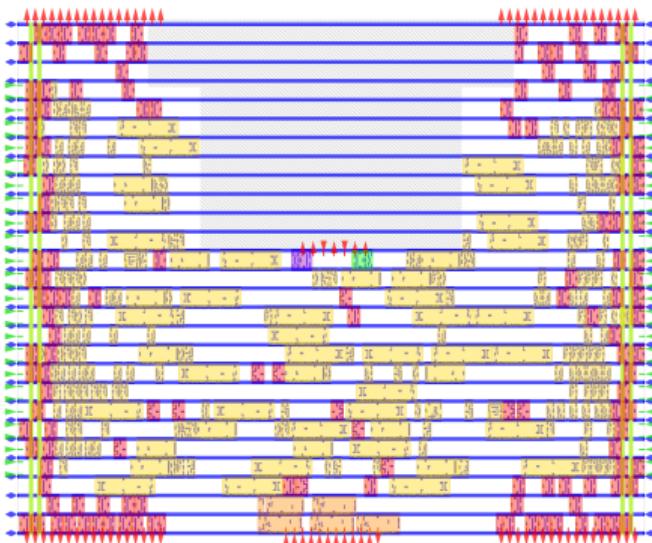
OpenROAD Step 3.2: Global Placement (I/O Aware)



- ▶ Cells migrated toward connected IO pins
- ▶ 445 movable cells, 457 total instances
- ▶ Timing-driven with Nesterov solver
- ▶ 335 iterations to converge

Metric	Value
Core area	2916 μm^2
Cell area	1102 μm^2
Utilization	49%
Final HPWL	5125 μm
Worst slack	4.49 ns
fmax achievable	1975 MHz

OpenROAD Step 3.3: Detailed Placement



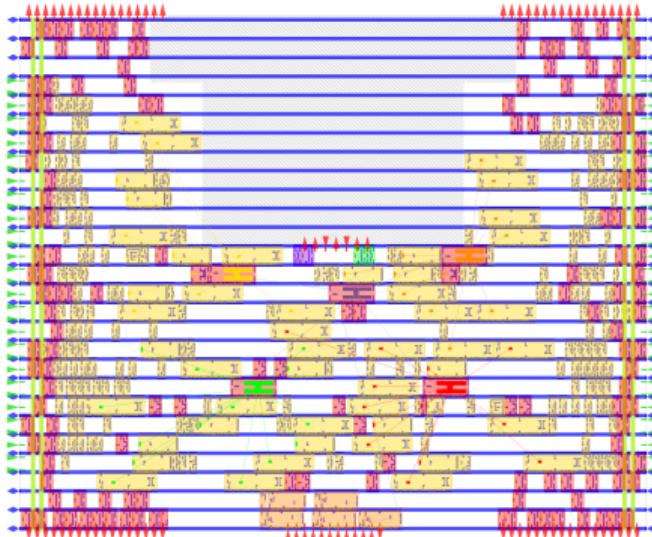
- ▶ 424 cells legalized to row sites
- ▶ Legalization adds 4% HPWL
- ▶ Optimization recovers 9.8% HPWL

Optimization Passes

Independent set matching	0.35%
Global swaps	2.36%
Vertical swaps	0.34%
Reordering	0.96%
Random improvement	3.09%
Cell flipping	3.10%

Metric	Value
Final HPWL	4796 μm
Final area	1102 μm^2
Utilization	38%

OpenROAD Step 4.1: Clock Tree Synthesis



- ▶ System clock: 10 MHz (100ns period)
- ▶ Samp/comp clock: 200 MHz (5ns period)
- ▶ 50ps clock uncertainty
- ▶ Balanced tree for minimal skew

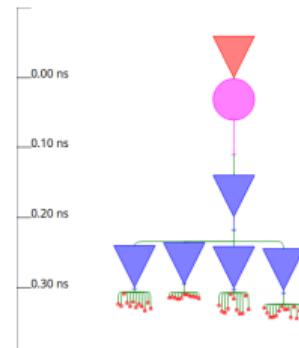
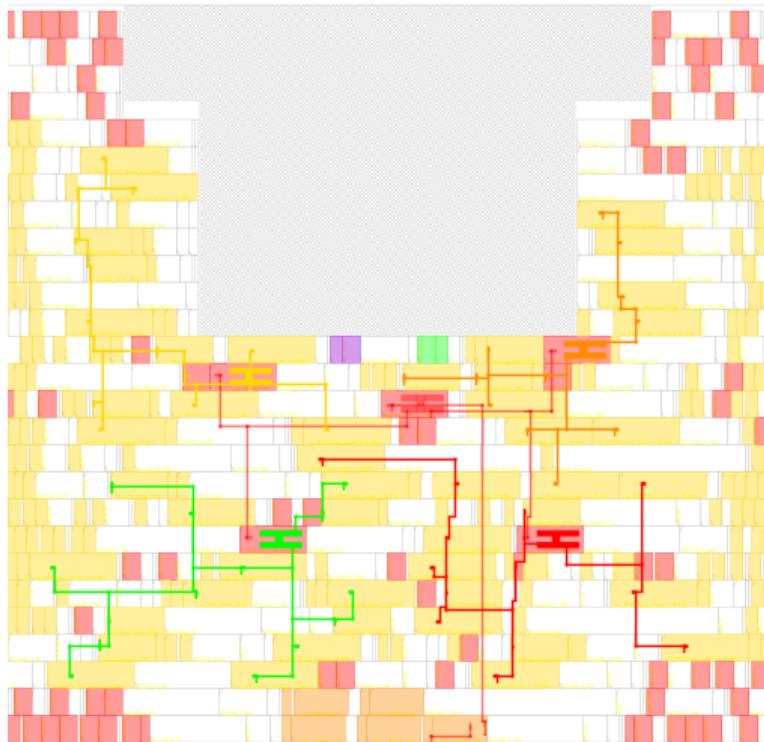
constraint.sdc

```
current_design adc_digital

# Create clock for sequencing (200 MHz)
create_clock -name seq_update \
-period 5 [get_ports seq_update]

# Set clock uncertainties
set_clock_uncertainty 0.05 [all_clocks]
```

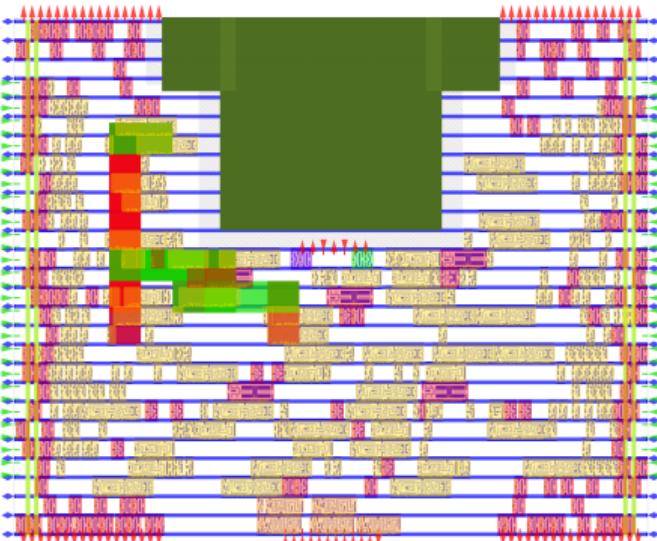
OpenROAD Step 4.1: Clock Tree Views



CTS Results:

- ▶ 31 BUFD2LVT cells across 4 clock paths
- ▶ Domains unsynchronized within ADC
(relies on upstream sequencer)
- ▶ Clock power: 41% of total
- ▶ Setup skew: 0.07 ns

OpenROAD Step 5.1: Global Routing



- ▶ Route guides shown for global routing
- ▶ M2-M3 routing layers only
- ▶ Routing obstructions for analog areas

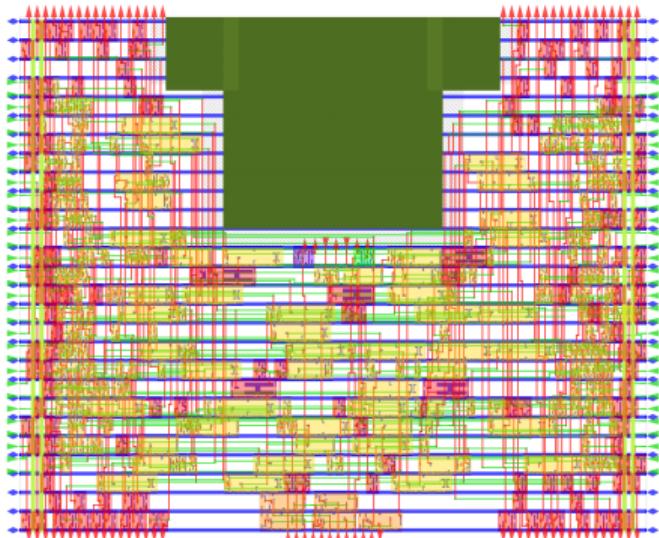
config.mk

```
export MIN_ROUTING_LAYER = M2
export MAX_ROUTING_LAYER = M3
export PRE_GLOBAL_ROUTE_TCL = \
.../routing_blockages.tcl
```

routing_blockages.tcl

```
# Create obstructions on M1-M4 for analog
odb::dbObstruction_create $block $layer_M1 \
$comp_llx $comp_lly $comp_urx $comp_ury
odb::dbObstruction_create $block $layer_M2 \
$comp_llx $comp_lly $comp_urx $comp_ury
```

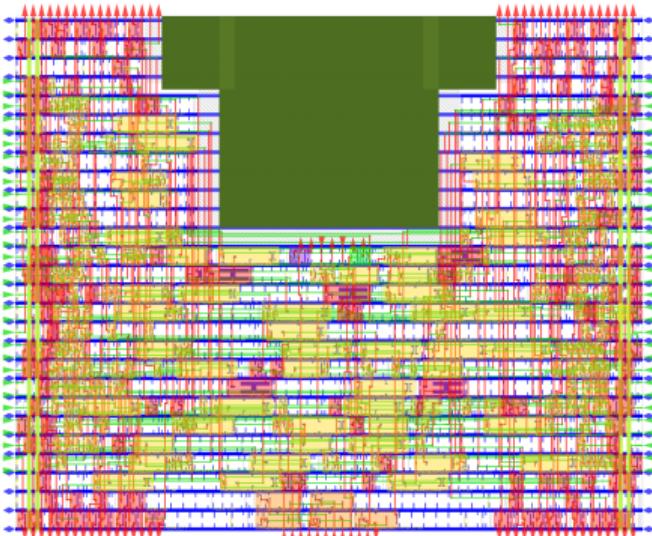
OpenROAD Step 5.2: Detailed Routing



- ▶ Final metal routing on M2/M3
- ▶ All signals routed to I/O pins
- ▶ DRC-clean routing around blockages
- ▶ **Issue:** Multicut vias on large devices

Metric	Value
Nets routed	506
Total wire length	5717 μm
M2 wire length	3183 μm
M3 wire length	2625 μm
Total vias	2058
Clock skew	0.07 ns
Worst slack	4.37 ns

OpenROAD Step 5.3: Fill Cells

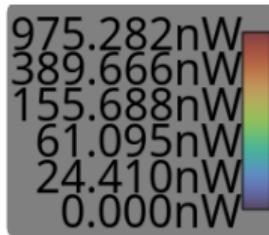
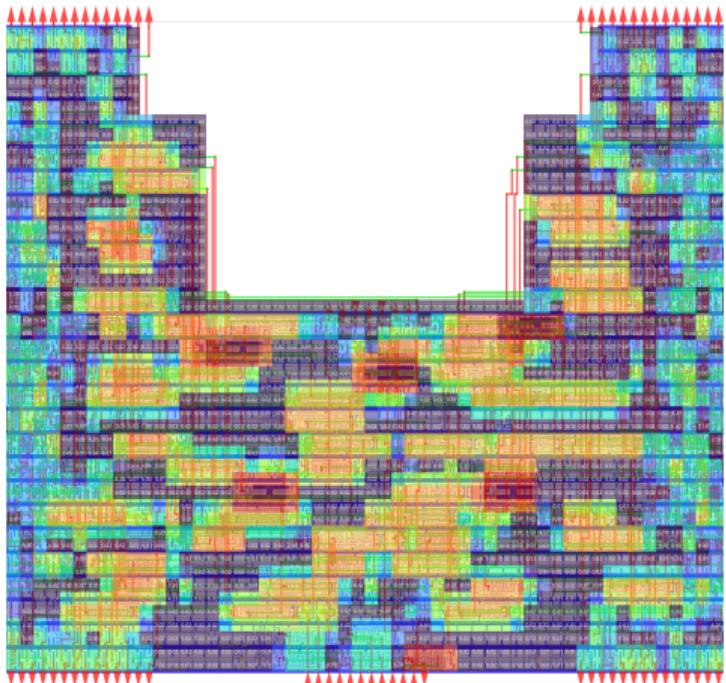


- ▶ Decap fill cells added (FEOL)
- ▶ Metal fill disabled (BEOL)
- ▶ Fills gaps in standard cell rows

config.mk

```
# Only disables metal fill
# FILL_CELLS in routing step still
# adds FEOL decap fill cells
export USE_FILL = 0
```

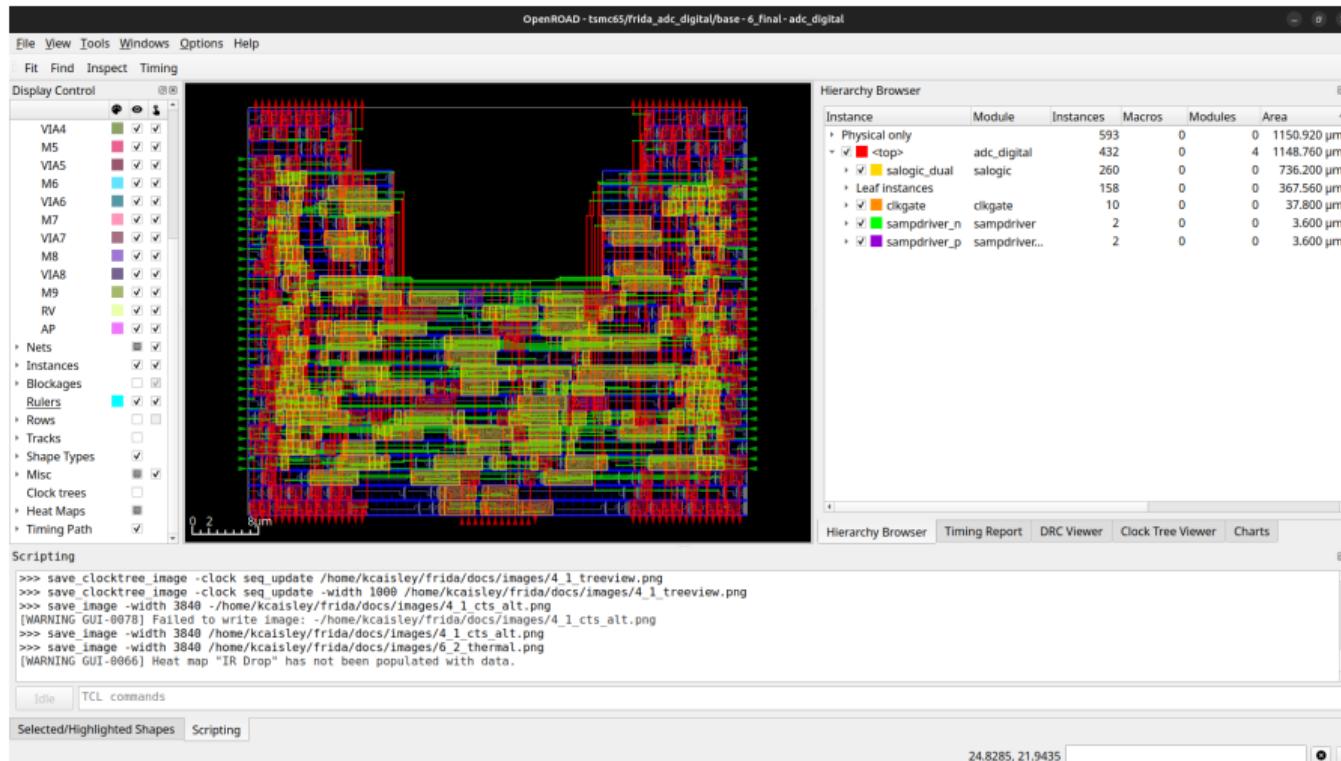
OpenROAD Step 6.1: Heat Maps



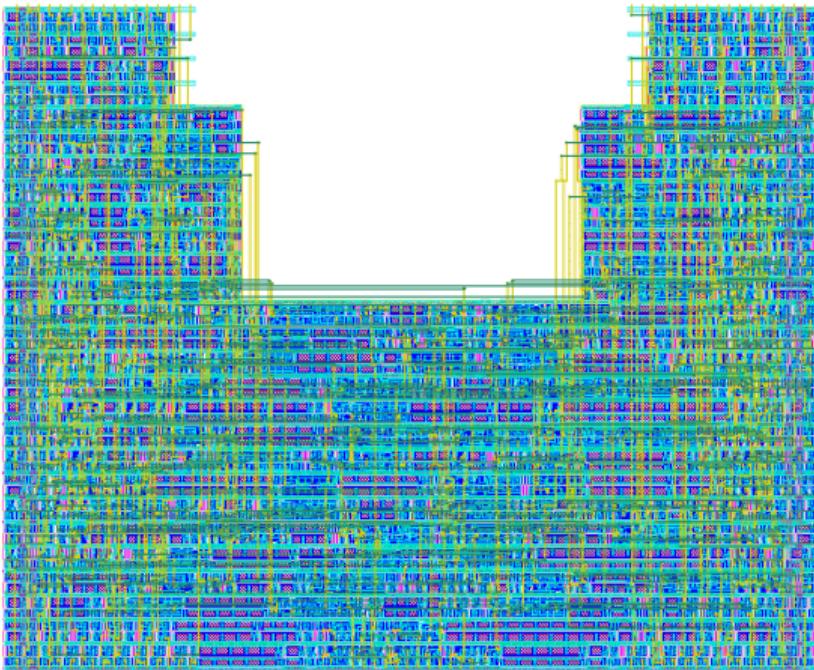
- ▶ Spatial visualizations for IR drop, thermal dissipation, placement density, pin density, and power consumption
- ▶ DRC and antenna violation views for debugging

OpenROAD GUI

- ▶ Interactive GUI, net tracing, hierarchical cell grouping, timing reports, DRC viewer, heatmaps

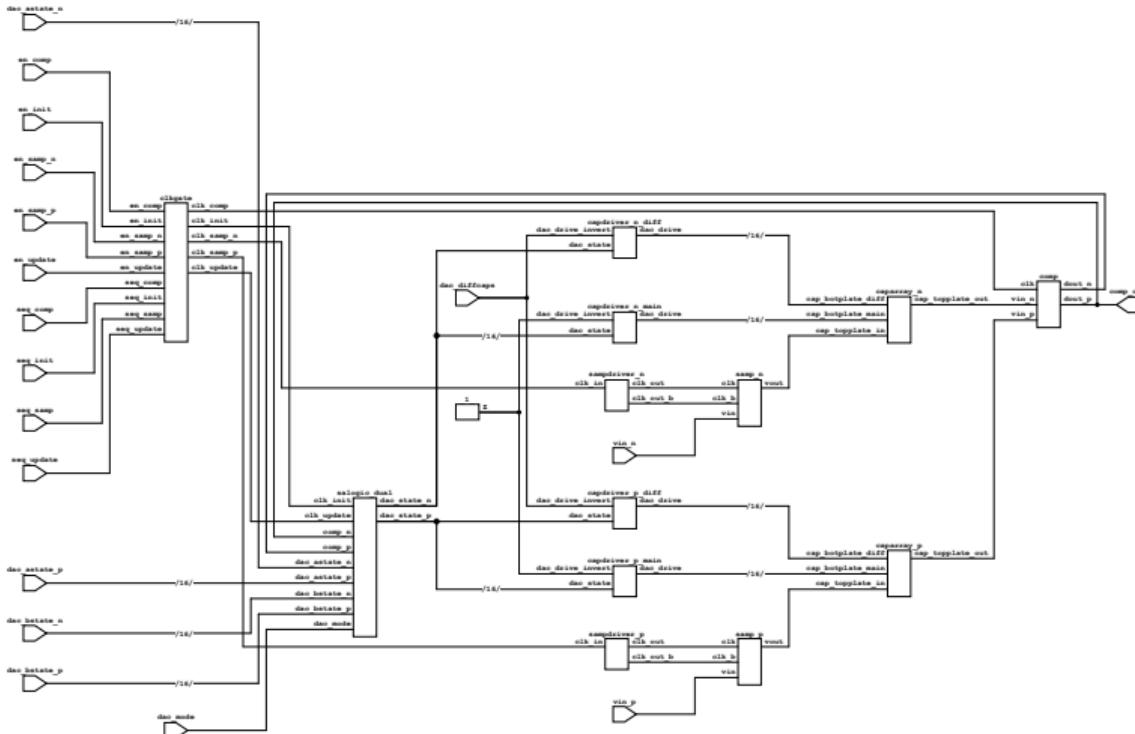


ADC Digital Block Layout



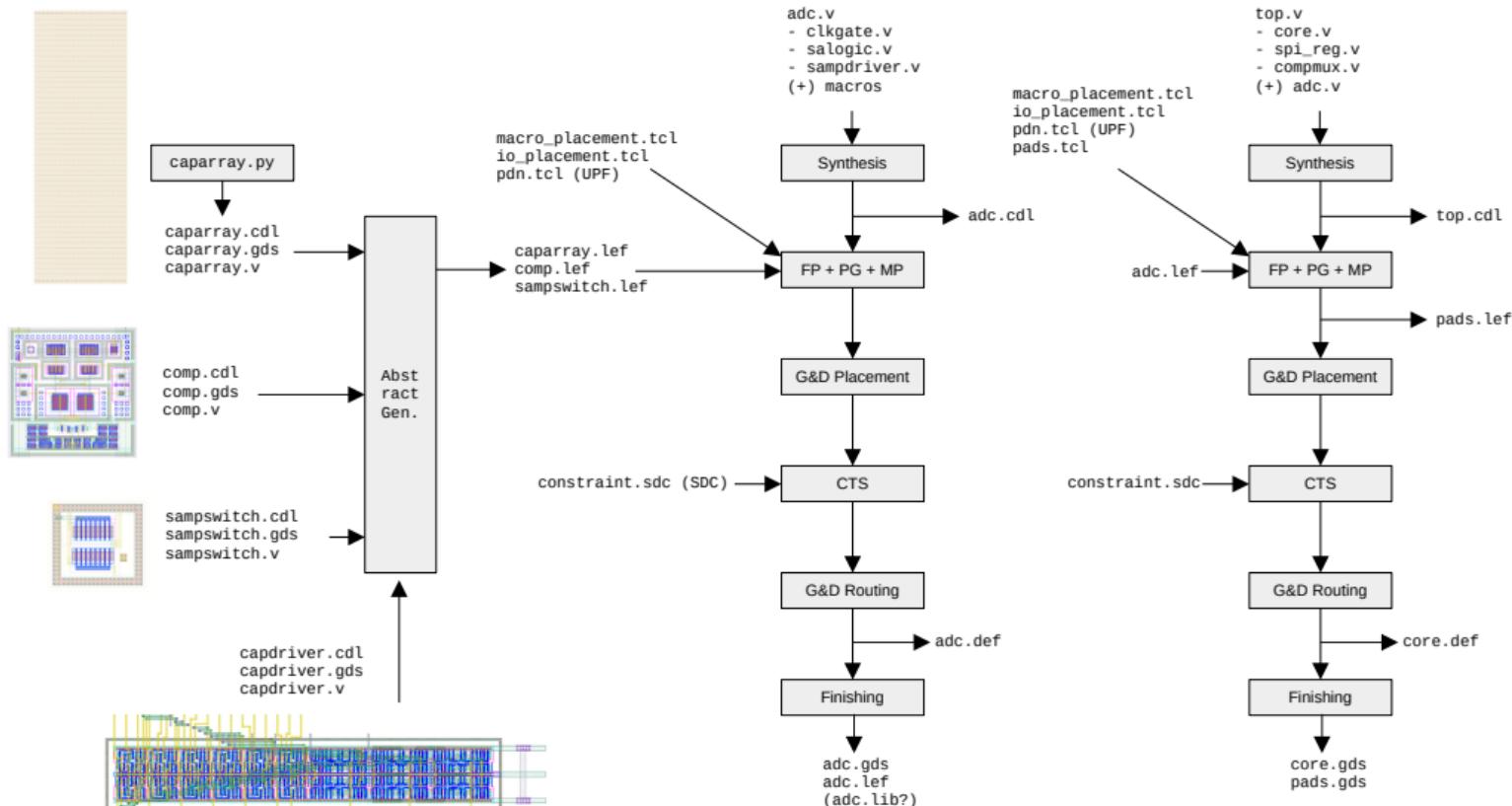
Metric	Value
Dimensions	$60 \times 49 \mu\text{m}$
Cell count	350
Area	$894 \mu\text{m}^2$
Flip-flops	48
Clock gates	5
Wire length	$5717 \mu\text{m}$
Vias	2058
Power	$18 \mu\text{W}$

ADC Top-Level Assembly



- ▶ ADC assembled using hierarchical Verilog as well, with protected routing for sensitive analog nets

ADC Level Integration



Unit Fringe Capacitor

Capacitance Density:

1 layer: $0.31 \text{ fF}/\mu\text{m}^2$

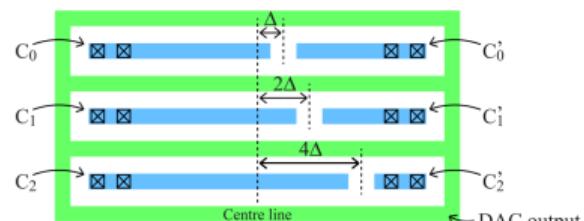
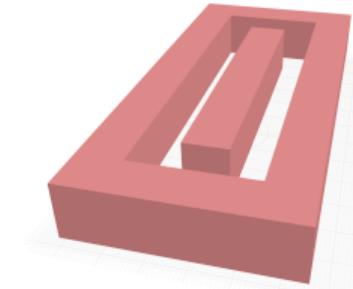
2 layers: $0.62 \text{ fF}/\mu\text{m}^2$

3 layers: $0.93 \text{ fF}/\mu\text{m}^2$

Matching: $\sigma(\Delta C/C) = 0.85\%/\sqrt{C \text{ [fF]}}$

Per-Bit Mismatch:

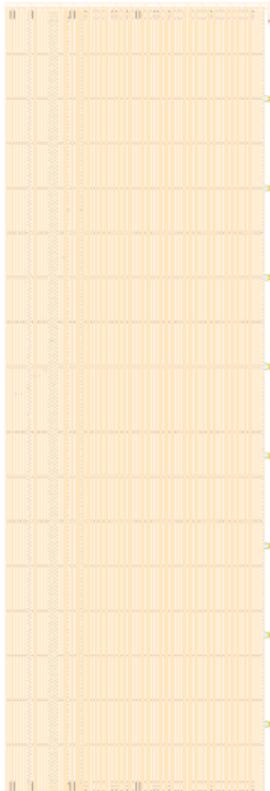
Bit	C_{main}	C_{diff}	uncorr	$\rho=0.9$
15 (768)	619 fF	4.8 fF	0.04%	0.03%
10 (64)	51.6 fF	0.4 fF	0.12%	0.11%
1 (1)	26.4 fF	25.6 fF	7.7%	2.4%



$$\begin{aligned} b_0 &= \parallel \quad C_0 = C_m + C_\Delta \\ \overline{b}_0 &= \parallel \quad C_0^* = C_m - C_\Delta \quad \} \quad C_{0,\text{eff}} = 2C_\Delta \\ b_1 &= \parallel \quad C_1 = C_m + 2C_\Delta \\ \overline{b}_1 &= \parallel \quad C_1^* = C_m - 2C_\Delta \quad \} \quad C_{1,\text{eff}} = 4C_\Delta \\ b_2 &= \parallel \quad C_2 = C_m + 4C_\Delta \\ \overline{b}_2 &= \parallel \quad C_2^* = C_m - 4C_\Delta \quad \} \quad C_{2,\text{eff}} = 8C_\Delta \end{aligned}$$

[P. Harpe, ISSCC 2018]

Capacitor Array Layout



Parameter	Value
Dimensions	$17 \times 50 \mu\text{m}$
C_{tot}	1.88 pF
C_{eff}	1.64 pF
Unit cap (C_u)	0.8 fF
Layers	M5–M6–M7 fringe

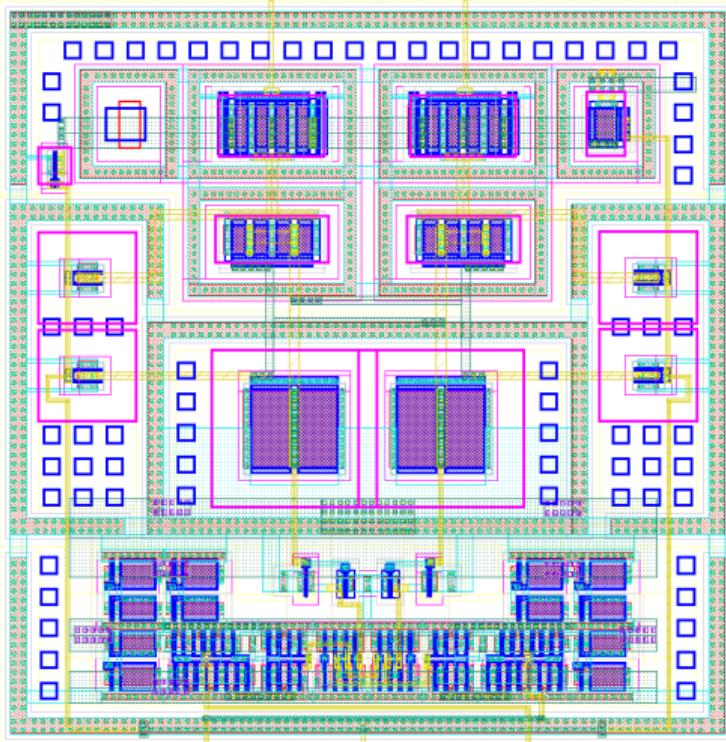
Weights: 768, 512, 320, 192, 96, 64, 32, 24, 12, 10, 5, 4, 4, 2, 1, 1

Harpe Penalty Factor:

$$\sigma_{\text{penalty}} = \sqrt{\frac{C_{\text{tot}}}{C_{\text{eff}}}} = \sqrt{\frac{1880}{1638}} \approx 1.07$$

Only 7% mismatch penalty — MSBs have $C_{\text{diff}} \ll C_{\text{main}}$

Comparator Layout



Parameter	Pre-Extract
Dimensions	$18 \times 18 \mu\text{m}$
Topology	StrongARM latch
Input noise	$\sim 177 \mu\text{V}$
Avg current	$\sim 4 \mu\text{A}$
Peak current	$\sim 500 \mu\text{A}$

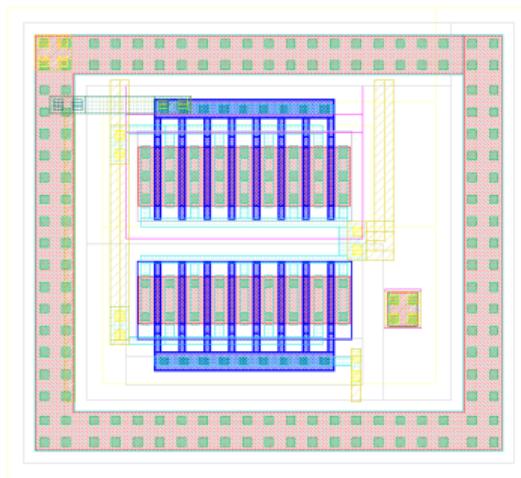
Noise contribution to ENOB:

$$V_{\text{LSB}} = \frac{2V_{\text{ref}}}{2^N} = \frac{1.2 \text{ V}}{4096} \approx 293 \mu\text{V}$$

$$\text{Comparator noise} \approx \frac{77 \mu\text{V}}{293 \mu\text{V}} \approx 0.26 \text{ LSB}$$

- ▶ Dynamic (no static power)
- ▶ Regenerative latch output

Sampling Switch Layout



Parameter	Value
Dimensions	$12 \times 12 \mu\text{m}$
Device size	$W \approx 3 \mu\text{m}$, L_{\min}
R_{on}	$\sim 1 \text{ k}\Omega$
C_{samp}	1.64 pF

1. Settling Time Requirement

Time constant: $\tau = R_{\text{on}} \cdot C = 1.64 \text{ ns}$

For 12-bit accuracy, settling error $< 0.5 \text{ LSB}$:

$$e^{-n} < \frac{0.5}{4096} \Rightarrow n > 9$$

$$\therefore t_{\text{sample}} > 9 \times 1.64 \text{ ns} \approx 15 \text{ ns}$$

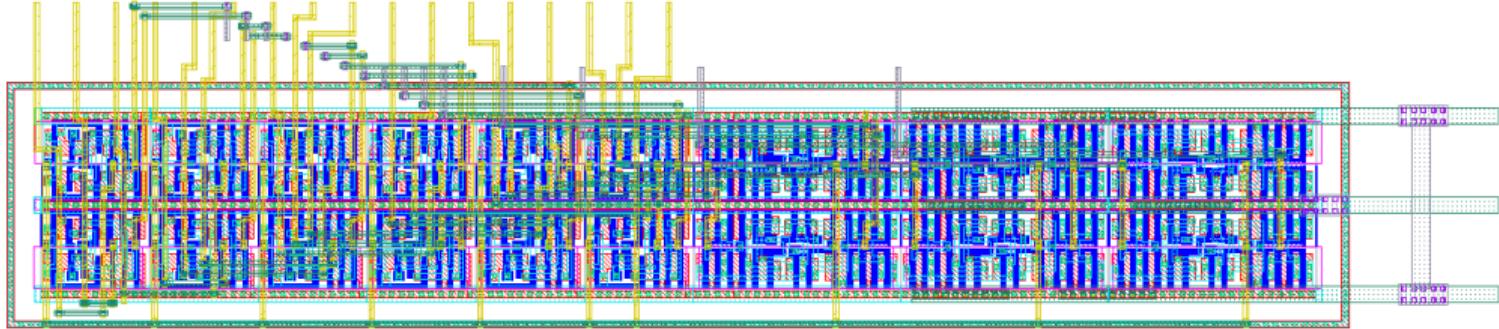
2. kT/C Sampling Noise

$$\sigma_{\text{samp}} = \sqrt{\frac{kT}{C}} = \sqrt{\frac{4.14 \times 10^{-21}}{1.64 \times 10^{-12}}} \approx 50 \mu\text{V}$$

With $\text{LSB} = 293 \mu\text{V}$:

$$\text{Noise} \approx \frac{50 \mu\text{V}}{293 \mu\text{V}/\text{LSB}} \approx 0.17 \text{ LSB} \quad \checkmark$$

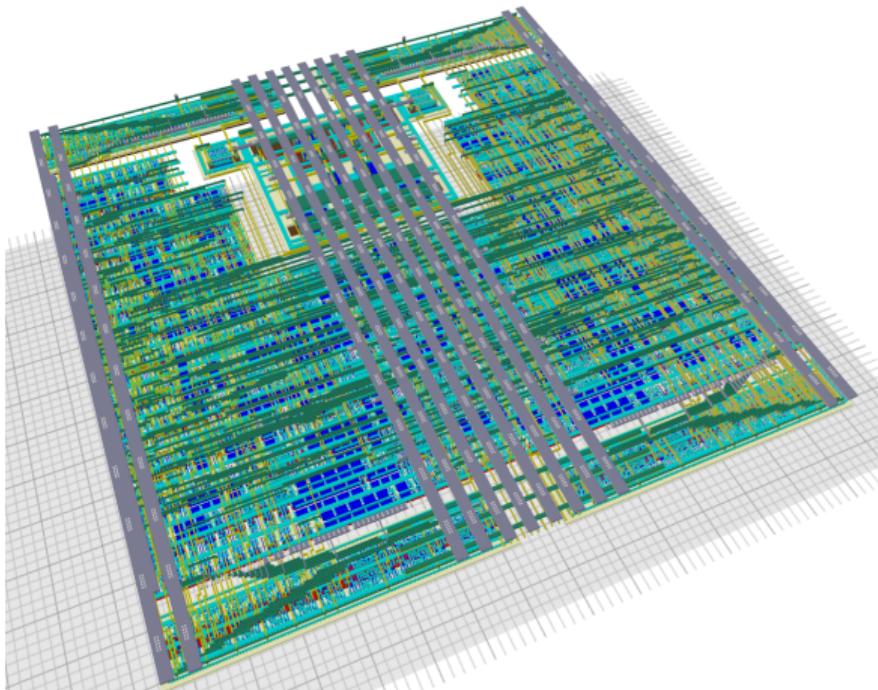
Capacitor Driver Layout



Drivers scaled in width to match corresponding capacitor size:

Bits	Output W (NMOS)	Output W (PMOS)
15–14 (MSB)	3.12 μm	4.16 μm
13–12	1.56 μm	2.08 μm
11–0 (LSBs)	0.78 μm	1.04 μm

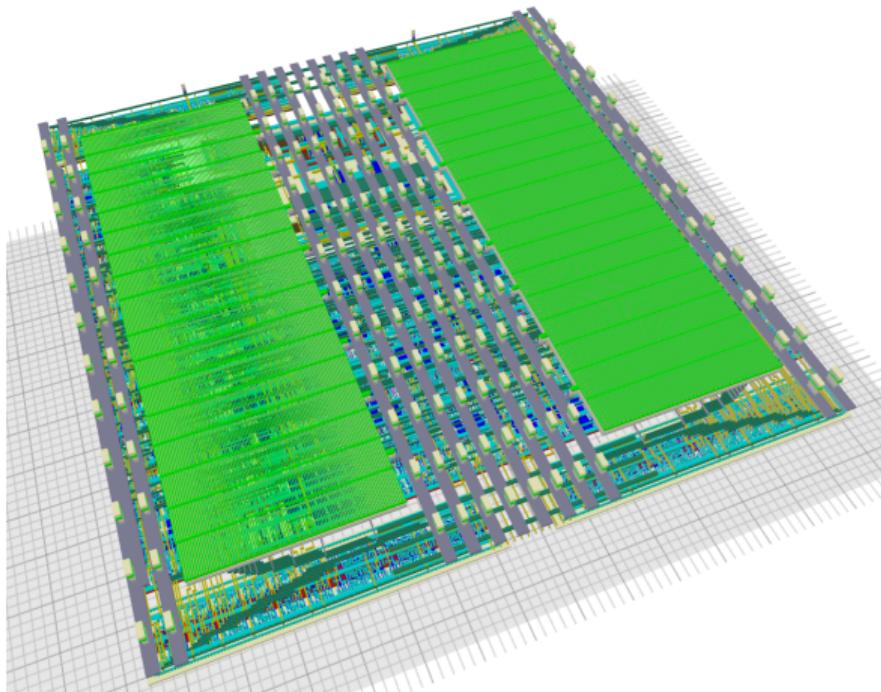
ADC Layout: Lower Metals (3D)



Layer	Usage
M1	Standard cell routing
M2–M3	Signal routing
M4	Power distribution

- ▶ Digital logic in center/bottom
- ▶ Comparator at top center
- ▶ Sampling switches on top sides
- ▶ Cap drivers at top/bottom edges

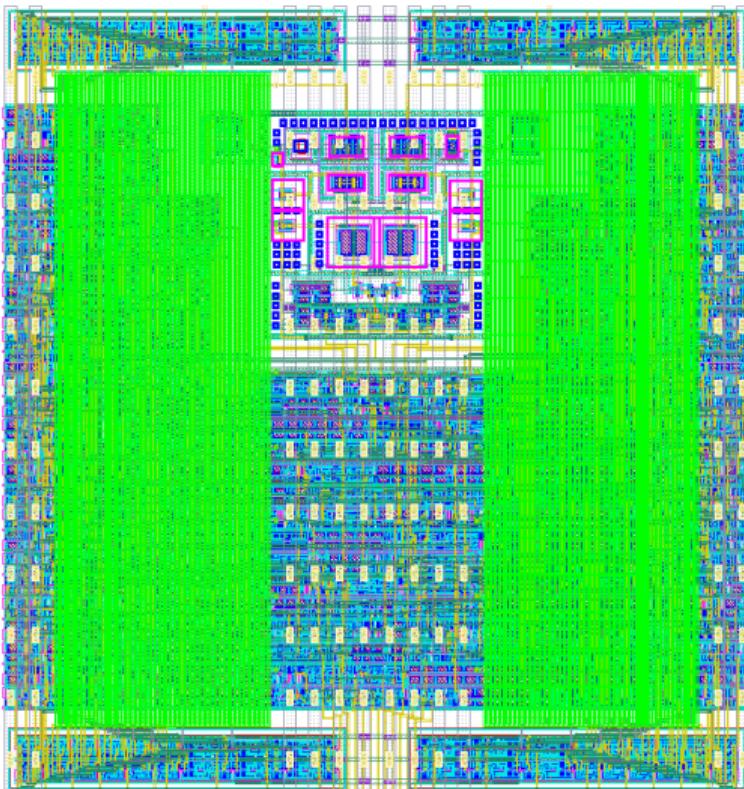
ADC Layout: With Cap Arrays (3D)



Layer	Usage
M5–M6	CDAC fringe caps

- ▶ P and N cap arrays on left/right
- ▶ Stacked above digital/analog, with shield
- ▶ Direct connection to sampling nodes
- ▶ Vias from top power grid between

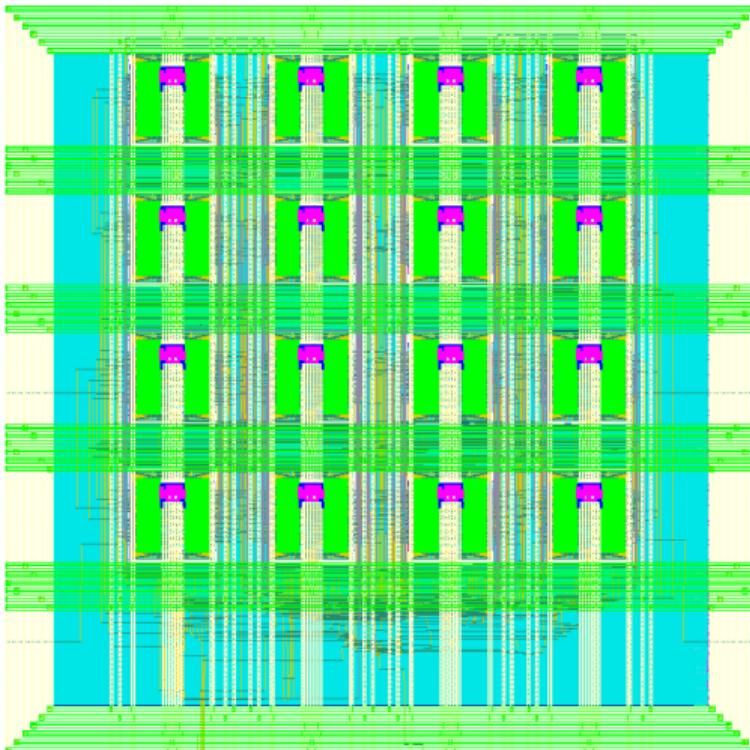
Full ADC Layout



Parameter	Value
Dimensions	$60 \times 60 \mu\text{m}$
Area	0.0036 mm^2
Resolution	12-bit
Sample rate	10 MS/s
Power	$\sim 200 \mu\text{W}$

- ▶ Fully differential SAR ADC
- ▶ Mixed-signal integration
- ▶ Column-parallel compatible

ADC Core Array (4×4)



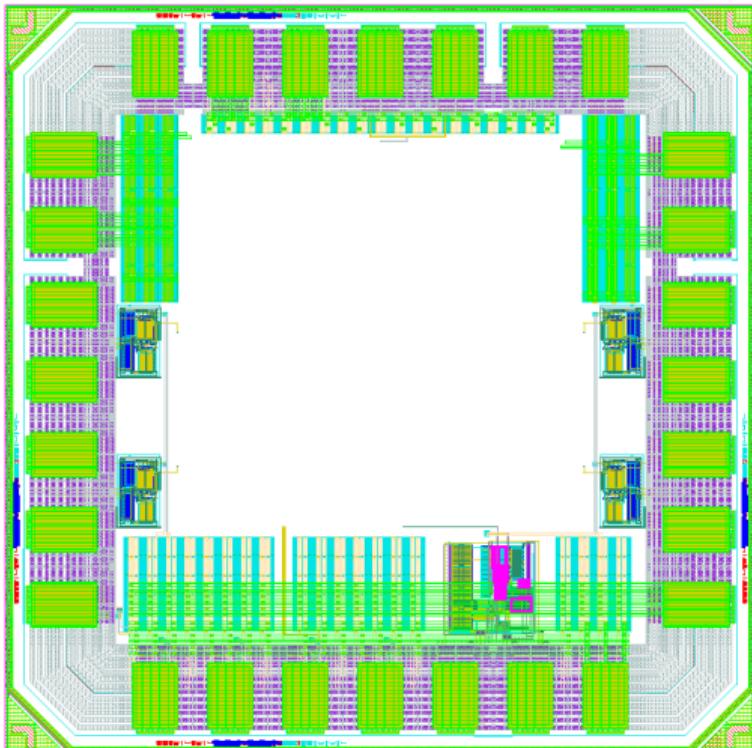
Parameter	Value
Array size	4×4 ADCs
Dimensions	$240 \times 240 \mu\text{m}$
Power Grid	M8-M9
Total power	$\sim 3.2 \text{ mW}$

- ▶ Multiplexed inputs and outputs
- ▶ Shared clock distribution

SPI Register (180 bits):

Bits	Function
[179:176]	Mux select (4b)
[175:64]	Per-ADC ctrl (7b \times 16)
[63:0]	Shared DAC states (64b)

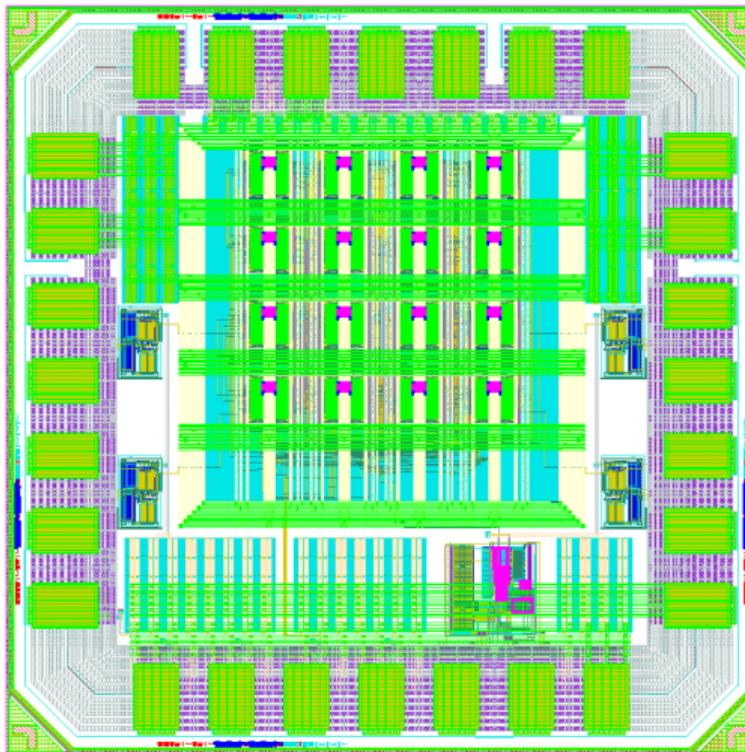
Pad Ring Layout



Pad Type	Count	Purpose
LVDS_RX	4	Sequencing clocks
LVDS_TX	1	Comparator output
CMOS_IO	5	SPI interface + reset
PASSIVE	2	Analog input (diff)
POWER	4	VDD (A, D, IO, DAC)
GROUND	4+1	VSS (A, D, IO, DAC)
Total	21	7 per side

- ▶ Separate analog/digital supplies
- ▶ ESD protection omitted on specific pads
- ▶ ~100 pF MOSCAP decoupling/rail
- ▶ Die size 1×1 mm

Full Chip Layout (FRIDA 65A)



Analog Automation: Sampling Switch Definition

Netlist struct (blocks/samp.py):

```
subckt = {
    "cellname": "samp",
    "ports": {}, # Computed by gen_topo_subckt()
    "instances": {}, # Computed by gen_topo_subckt()
    "tech": ["tsmc65", "tsmc28", "tower180"],
    "topo_params": {
        "switch_type": ["nmos", "pmos", "tgate"]
    },
    "inst_params": [
        {"instances": {"nmos": [{"MN": 1, "MP": 1}], "w": [5, 10, 20, 40], "l": [1, 2]}, "type": "lvt", "w": 1, "l": 1, "nf": 1},
        {"instances": {"nmos": "all", "pmos": "all"}, "type": "lvt", "w": 1, "l": 1, "nf": 1}
    ],
}
```

Topology generator:

```
def gen_topo_subckt(switch_type: str):
    ports = {"in": "I", "out": "O", "clk": "I",
             "clk_b": "I", "vdd": "B", "vss": "B"}
    instances = {}
    if switch_type == "nmos":
        instances["MN"] = {"dev": "nmos",
                           "pins": {"d": "out", "g": "clk", "s": "in", "b": "vss"}}
    elif switch_type == "tgate":
        instances["MN"] = ...
        instances["MP"] = ...
    return ports, instances
```

Flow Commands:

```
# Generate subcircuit netlists
make subckt cell=samp

# Generate testbench netlists
make tb cell=samp

# Run simulations
make sim cell=samp mode=all

# Extract measurements
make meas cell=samp
```

Combinatorial Expansion:

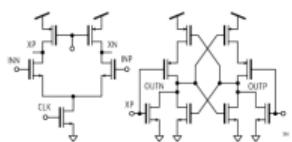
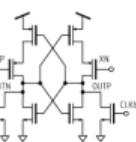
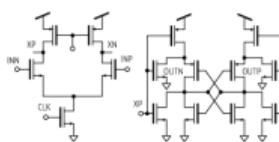
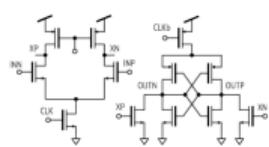
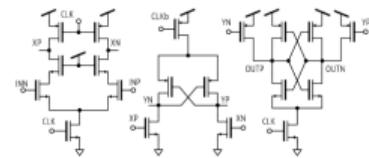
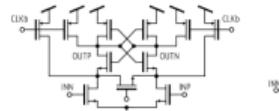
- ▶ 3 technologies × 3 topologies
- ▶ 4 widths × 2 lengths
- ▶ = 72 netlist variants

Analog Automation: Subcircuit Generation

```
$ make subckt cell=samp
=====
          SUBCIRCUIT NETLISTING
=====
Cell:      samp
Script:    blocks/samp.py
Step:     subckt
OutDir:   results/samp/subckt
-----
tech      switch_type  MN.1        MN.w        MP.1        MP.w
-----
tsmc65    3x          2x          4x          2x          4x
tsmc28    3x          2x          4x          2x          4x
tower180  3x          2x          4x          2x          4x
-----
Result:   72 SPICE netlists generated
          72 Verilog netlists generated
Wall Time: 16.9ms
Errors:   [none]
```

- ▶ Cross-technology sweep: same design across foundries
- ▶ Device parameter sweep: automatic design space exploration
- ▶ Parallel generation: fast iteration on analog IP blocks
- ▶ Dual output: SPICE for simulation, Verilog for integration

Analog Automation: Comparator Topologies



Topology Parameters:

Parameter	Options
preamp_diffpair	nmos, pmos
preamp_bias	std, dyn
comp_stages	1, 2
latch_pwrgate_ctl	clk, sig
latch_pwrgate_node	ext, int
latch_RST_ext_ctl	clk, sig, none
latch_RST_int_ctl	clk, sig

- ▶ Auto netlist from `blocks/comp.py`
- ▶ Sweep device params & corners
- ▶ Can also vary topology & tech node
- ▶ Share testbench, simulation, and analysis code

Analog Automation: Generated SPICE Netlists

180nm Double-Stage (Tower):

```
.subckt comp in+ in- out+ out- clk clkb vdd vss

M_preamp_diff+ out- in+ tail vss n18lvt W=880n L=180n
M_preamp_diff- out+ in- tail vss n18lvt W=880n L=180n
M_preamp_tail tail clk vss vss n18lvt W=440n L=360n
M_preamp_rst+ out- clk vdd vdd p18lvt W=440n L=180n
M_preamp_rst- out+ clk vdd vdd p18lvt W=440n L=180n
M_preamp_latch+ latch- latch+ latch_vdd vdd p18lvt W=220n L=180n
M_preamp_latch- latch+ latch- latch_vdd vdd p18lvt W=220n L=180n
M_preamp_latch+ latch- latch+ latch_vss vss n18lvt W=220n L=180n
M_preamp_latch- latch+ latch- latch_vss vss n18lvt W=220n L=180n
M_preamp_to_latch+ latch- out- vss vss n18lvt W=220n L=180n
M_preamp_to_latch- latch+ out+ vss vss n18lvt W=220n L=180n
M_latch_ext_powergate+ latch_vdd clk vdd vdd p18lvt W=220n
M_latch_ext_powergate- latch_vdd clk vdd vdd p18lvt W=220n
M_latch_ext_rst+ latch- clk vdd vdd p18lvt W=220n
M_latch_ext_rst- latch+ clk vdd vdd p18lvt W=220n
M_latch_vss_conn+ latch_vss vdd vss vss n18lvt W=220n
M_latch_vss_conn- latch_vss vdd vss vss n18lvt W=220n
M_latch_int_rst+ latch- clk vdd vdd p18lvt W=220n
M_latch_int_rst- latch+ clk vdd vdd p18lvt W=220n
M_latch_out+ out- latch- vdd vdd p18lvt W=220n L=180n
M_latch_out- out+ latch+ vdd vdd p18lvt W=220n L=180n
M_b_latch_out+ out- latch- vss vss n18lvt W=220n L=180n
M_b_latch_out- out+ latch+ vss vss n18lvt W=220n L=180n

.ends comp
```

65nm Single-Stage (TSMC):

```
.subckt comp in+ in- out+ out- clk clkb vdd vss

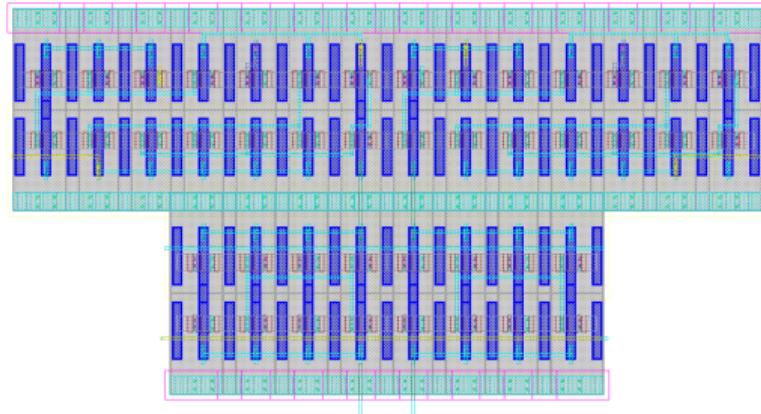
M_preamp_diff+ out- in+ tail vss nch_lvt W=480n L=60n
M_preamp_diff- out+ in- tail vss nch_lvt W=480n L=60n
M_preamp_tail tail clk vss vss nch_lvt W=240n L=120n
M_preamp_rst+ out- clk vdd vdd pch_lvt W=240n L=60n
M_preamp_rst- out+ clk vdd vdd pch_lvt W=240n L=60n
M_latch+ latch- out- vdd vdd pch_lvt W=120n L=60n
M_latch- out+ out- vdd vdd pch_lvt W=120n L=60n
M_b_latch+ latch- out+ vss vss nch_lvt W=120n L=60n
M_b_latch- out+ out- vss vss nch_lvt W=120n L=60n
M_latch_int_rst+ out- clk vdd vdd pch_lvt W=120n L=60n
M_latch_int_rst- out+ clk vdd vdd pch_lvt W=120n L=60n

.ends comp
```

Auto-Generated Verilog:

```
module comp (in+, in-, out+, out-, clk, clkb, vdd, vss);
    input in+, in-;
    output out+, out-;
    input clk, clkb;
    inout vdd, vss;
    wire tail;
    nch_lvt M_preamp_diff+ (.d(out-), .g(in+), .s(tail));
    nch_lvt M_preamp_diff- (.d(out+), .g(in-), .s(tail));
    // ... (structural Verilog for LVS)
endmodule
```

Analog Automation: Layout with OpenROAD



1. **Parse SPICE → Verilog**
Structural netlist with transistor primitives
2. **Map to primitive cells**
NMOS/PMOS → stdcell-like wrapper LEFs
3. **Partition differential halves**
Split circuit at symmetry axis (P vs N)
4. **Explore with net weights**
Vary placement via `specifyNetWeight` to generate layout candidates
5. **Reassemble with symmetry**
Mirror placement coordinates across axis; lock cells with `LOCKED` status
6. **Route with guides**
Extract guides, mirror for paired nets

Analog Automation: CDAC Weight Strategies

Five strategies in blocks/cdac.py:

Strategy	Description	Radix
rdx2	Standard binary	2.0
subrdx2	Sub-radix-2 rounded	<2.0
subrdx2lim	Sub-radix-2 w/ floor	<2.0
subrdx2rdst	MSB redistribution	~2.0
rdx2rpt	Binary w/ repeated	2.0

Weight calculation:

$$\beta = 2^{N/M} \quad W_i = \max(1, \lfloor \beta^{M-1-i} \rfloor)$$

Coarse-Fine Partitioning (threshold $T = 64$):

$$W = q \cdot T + r \quad \text{where } q = \lfloor W/T \rfloor$$

Difference Capacitor Split:

- ▶ Coarse: $C_{\text{main}} = T \cdot C_u, m = q$
- ▶ Coarse diff: $C_{\text{diff}} = 1 \cdot C_u, m = q$
- ▶ Fine: $C_{\text{main}} = (T + 1 + r),$
 $C_{\text{diff}} = (T + 1 - r)$

Effective: $C_{\text{eff}} = C_{\text{main}} - C_{\text{diff}}$

ADC Target Specifications

Parameter	CD v1	CoRDIA	M	H	F (target)
Resolution	8-bit	10-bit	8-bit	10-bit	12-bit
ENOB	8.3	8.8	8.0	9.5?	11.0?
Conversion rate	6.25 MHz	2.5 MHz	4.5 MHz	10 MHz	10 MHz
ADC dimensions	$40 \times 55 \mu\text{m}$	$80 \times 330 \mu\text{m}$	$60 \times 800 \mu\text{m}$	$15 \times 100 \mu\text{m}$	$50 \times 50 \mu\text{m}$
ADC area	0.002 mm^2	0.026 mm^2	0.048 mm^2	0.0015 mm^2	0.0025 mm^2
Power per ADC	$960 \mu\text{W}$	$30 \mu\text{W}$	$700 \mu\text{W}$	$100 \mu\text{W}$	$200 \mu\text{W?}$
$\text{FOM}_{\text{csa}} (\text{Hz}/\mu\text{m}^2)$	3125	95	105	5000	5000
$\text{FOM}_{\text{wal}} (\text{fJ}/\text{conv-step})$	487	26	608	14	10

- ▶ F = FRIDA target: 12-bit, 10 MHz, $50 \times 50 \mu\text{m}$ footprint
- ▶ FOM_{csa} = conversion rate / area; FOM_{wal} = energy per conversion step