8-bit SAR ADC on Tiny Tapeout

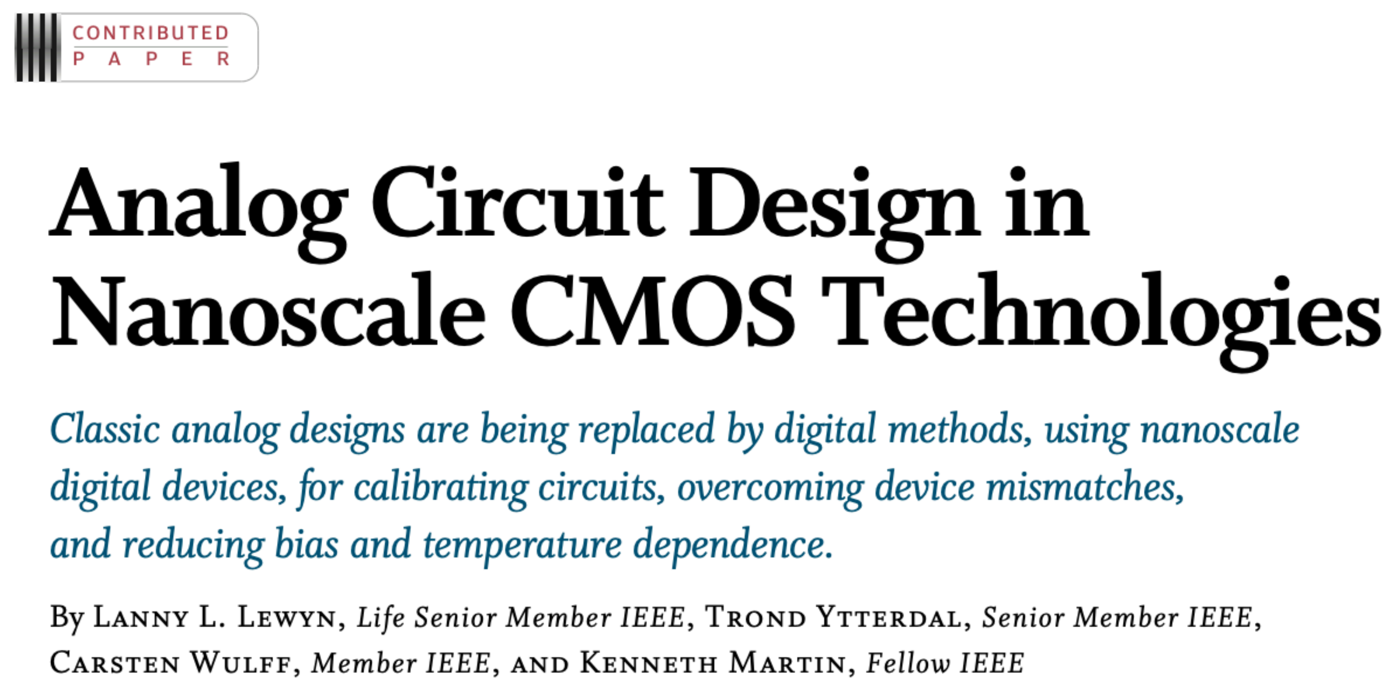
tt06-sar

Goal

Walk you through **tt06-sar**, and the steps necessary to **tapeout** on **TinyTapeout**

Idea

Trigger (2009)



CONTRIBUTED PAPER

Analog Circuit Design in Nanoscale CMOS Technologies

Classic analog designs are being replaced by digital methods, using nanoscale digital devices, for calibrating circuits, overcoming device mismatches, and reducing bias and temperature dependence.

By Lanny L. Lewyn, *Life Senior Member IEEE*, Trond Ytterdal, *Senior Member IEEE*, Carsten Wulff, *Member IEEE*, and Kenneth Martin, *Fellow IEEE*

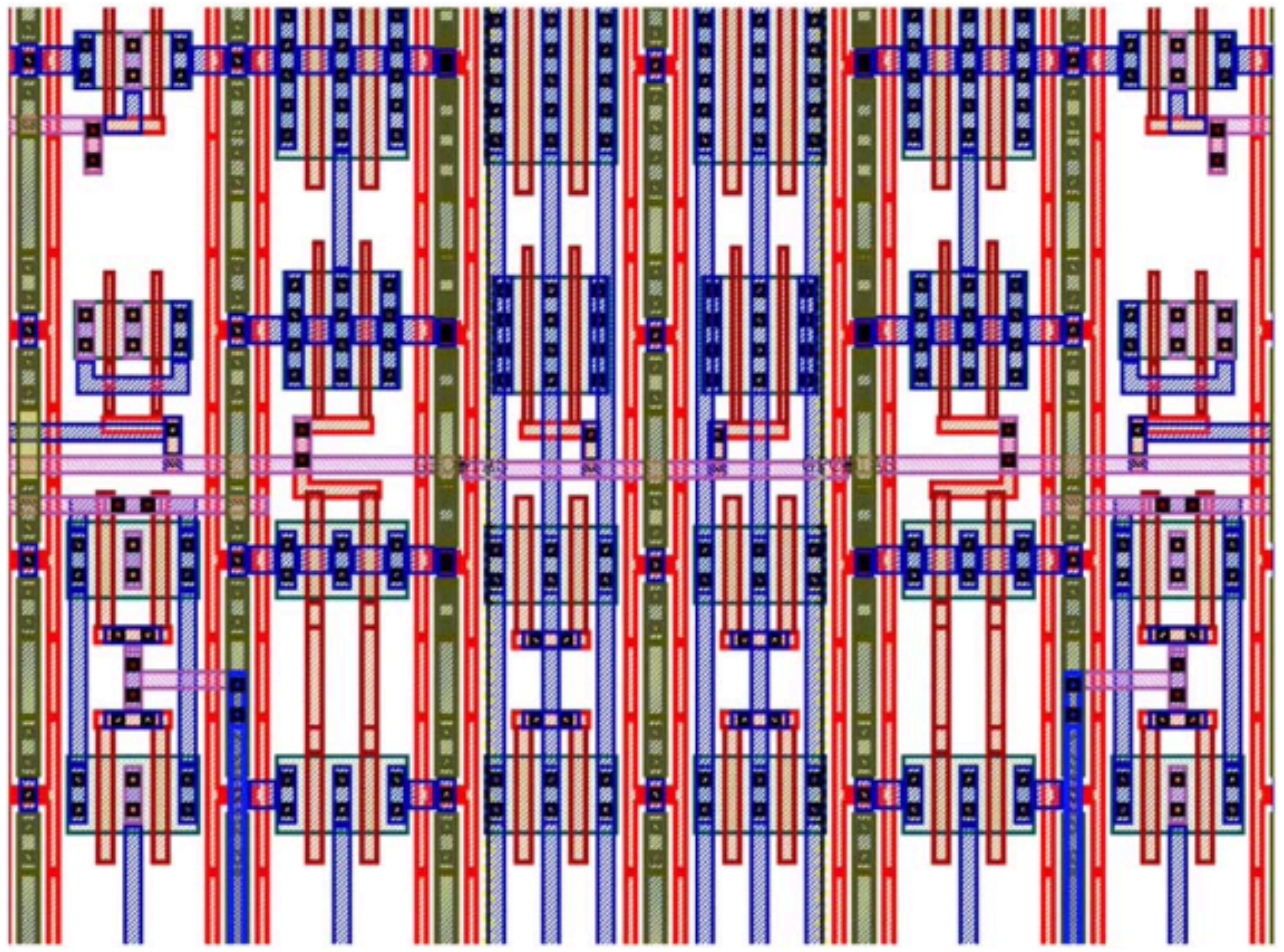
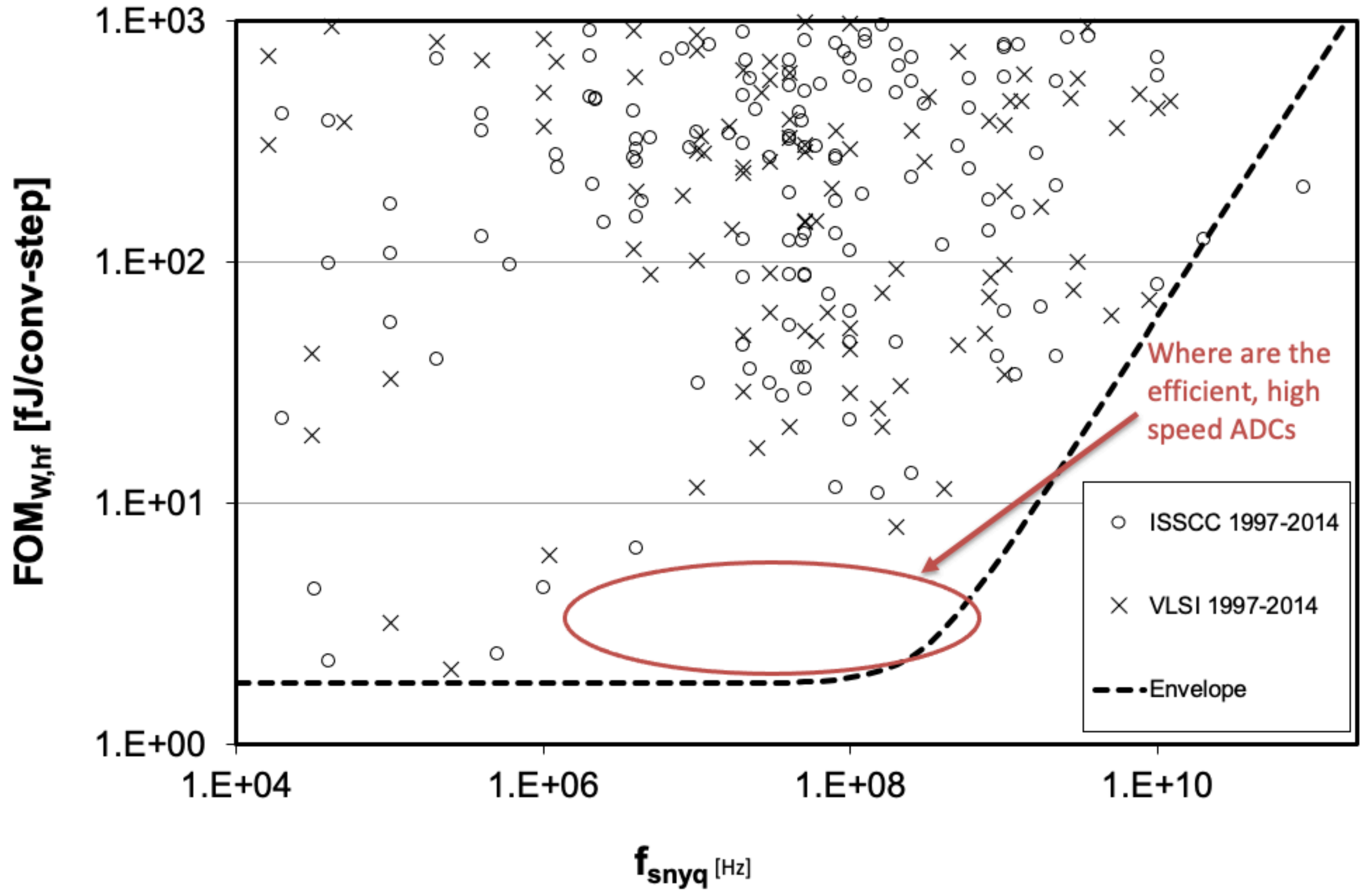
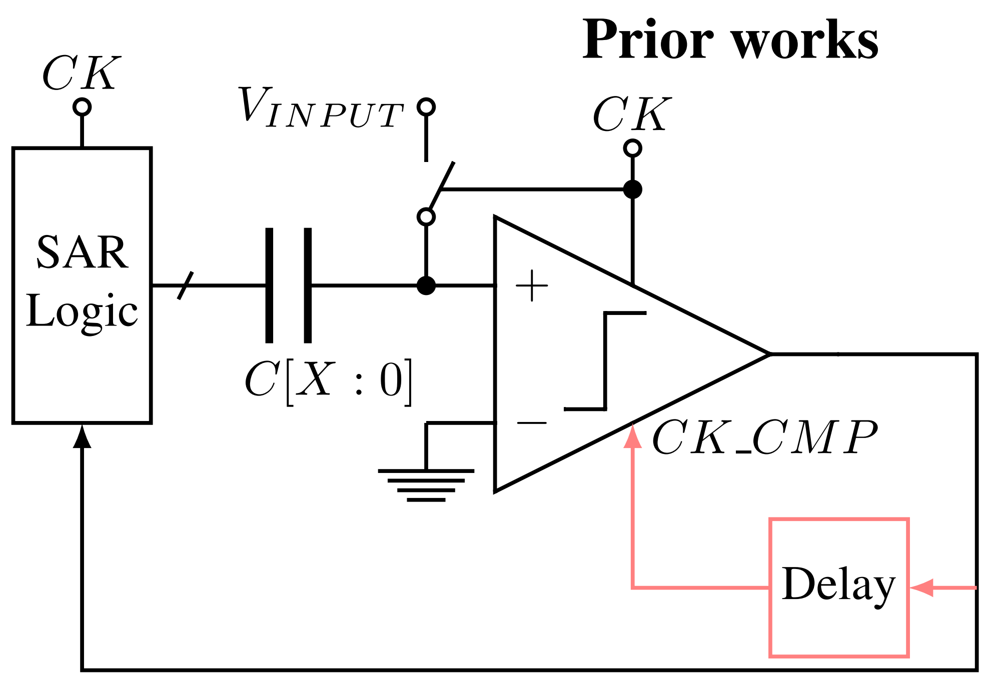


Fig. 6. ***A portion of an amplifier cell with regular device pitch in both X and Y directions (upper metal layers removed for clarity). For best HF performance, all devices' substrate ties are placed on either side of two-finger gate patterns. Grounded stripes of poly are interposed between device active area and all substrate ties to minimize the need for reticle compensation (OPC) and also reduce poly etch loading to achieve good CD accuracy.***

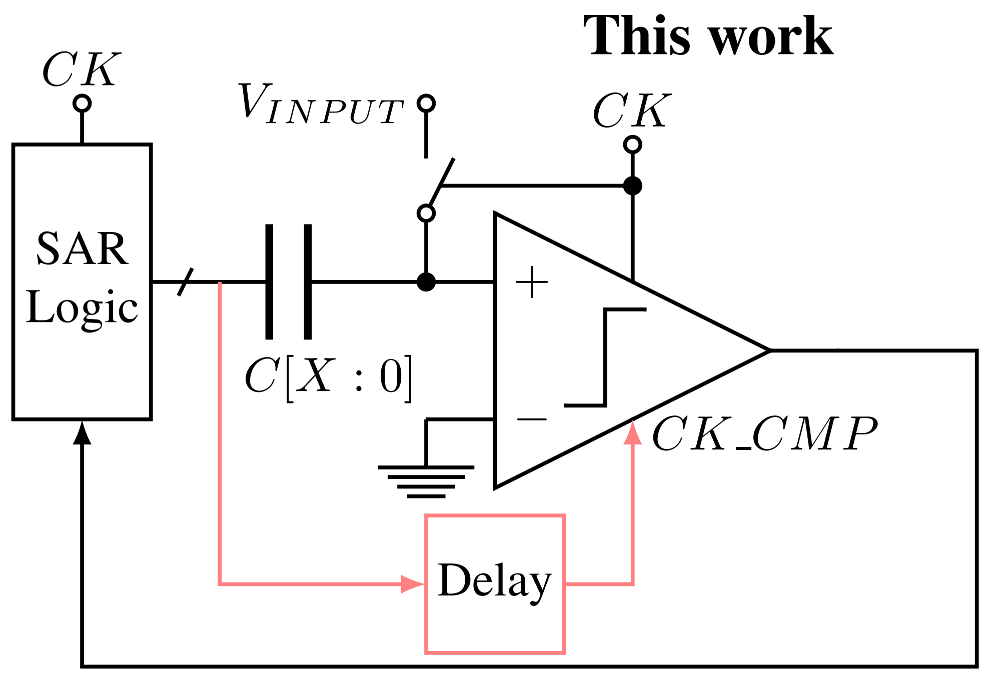
Problem (2014)



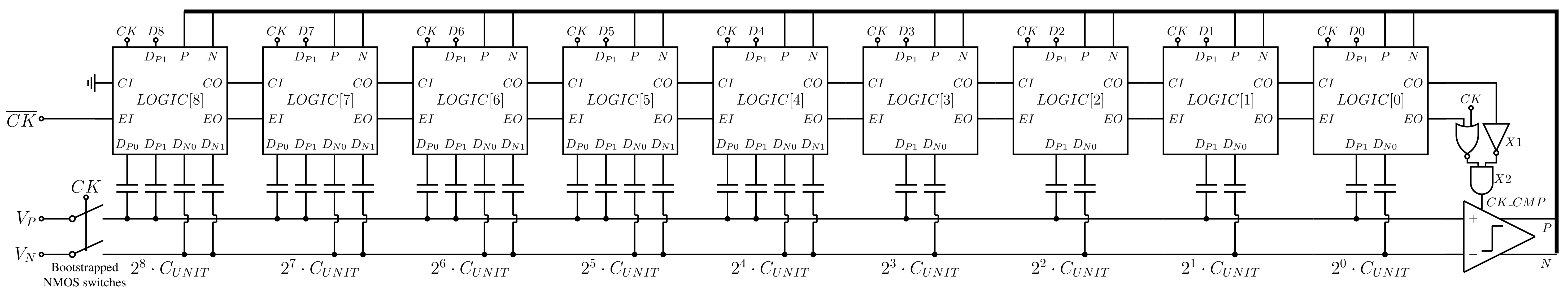
Circuit

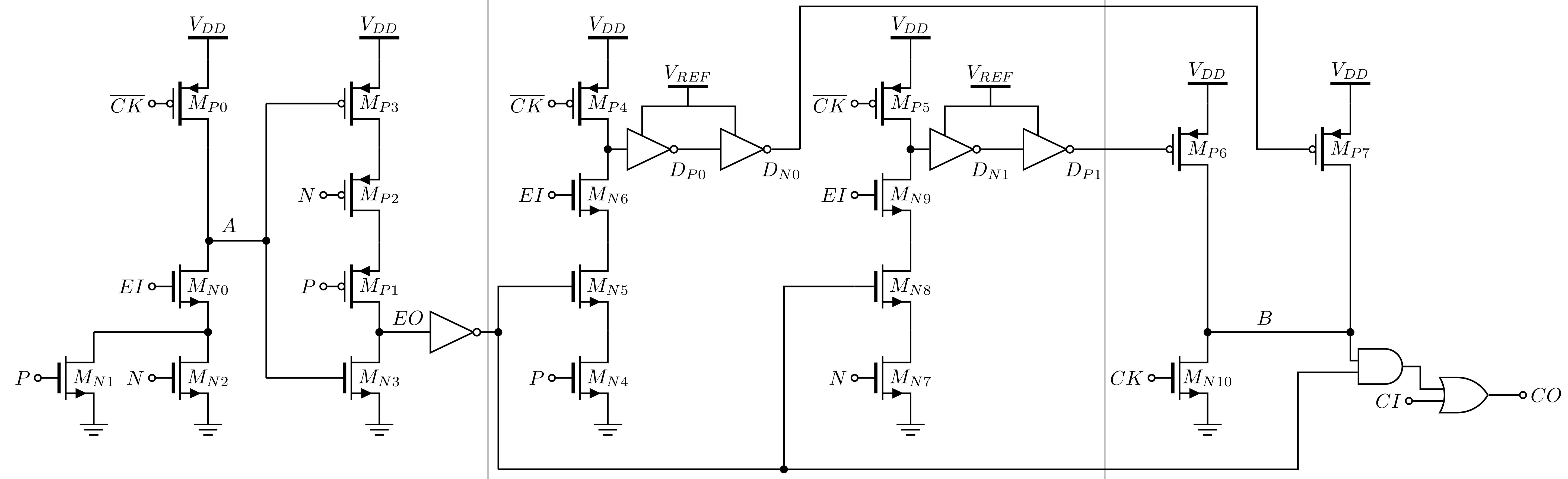


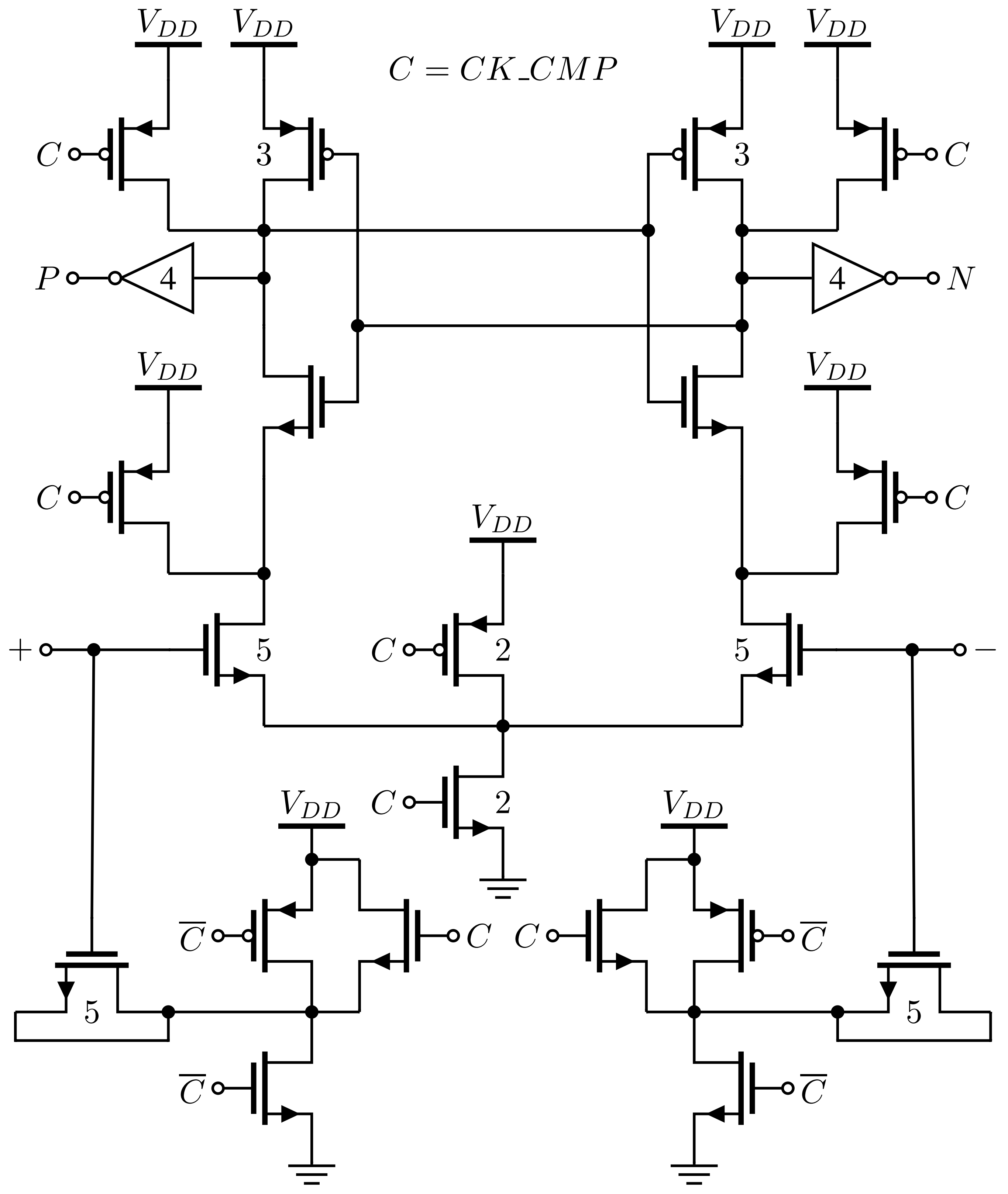
(a) Clock generation separate from CDAC

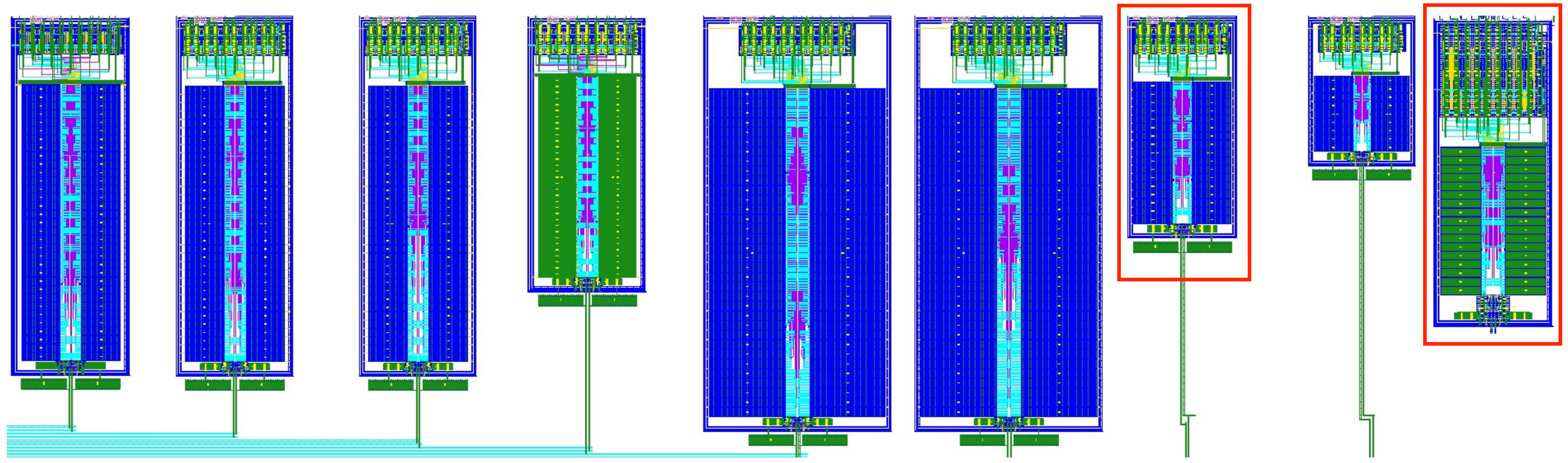


(b) Clock generation including CDAC


CK $D8$ ... CK $D0$
D_{P1} P N
CI CO
$LOGIC[8]$ $LOGIC[7]$ $LOGIC[6]$ $LOGIC[5]$ $LOGIC[4]$ $LOGIC[3]$ $LOGIC[2]$ $LOGIC[1]$ $LOGIC[0]$
EI EO
D_{P0} D_{P1} D_{N0} D_{N1}
$\overline{CK}$
CK
V_P
V_N
Bootstrapped NMOS switches
$2^8 \cdot C_{UNIT}$ $2^7 \cdot C_{UNIT}$ $2^6 \cdot C_{UNIT}$ $2^5 \cdot C_{UNIT}$ $2^4 \cdot C_{UNIT}$ $2^3 \cdot C_{UNIT}$ $2^2 \cdot C_{UNIT}$ $2^1 \cdot C_{UNIT}$ $2^0 \cdot C_{UNIT}$
$X1$ $X2$ CK_CMP
P N
(a)

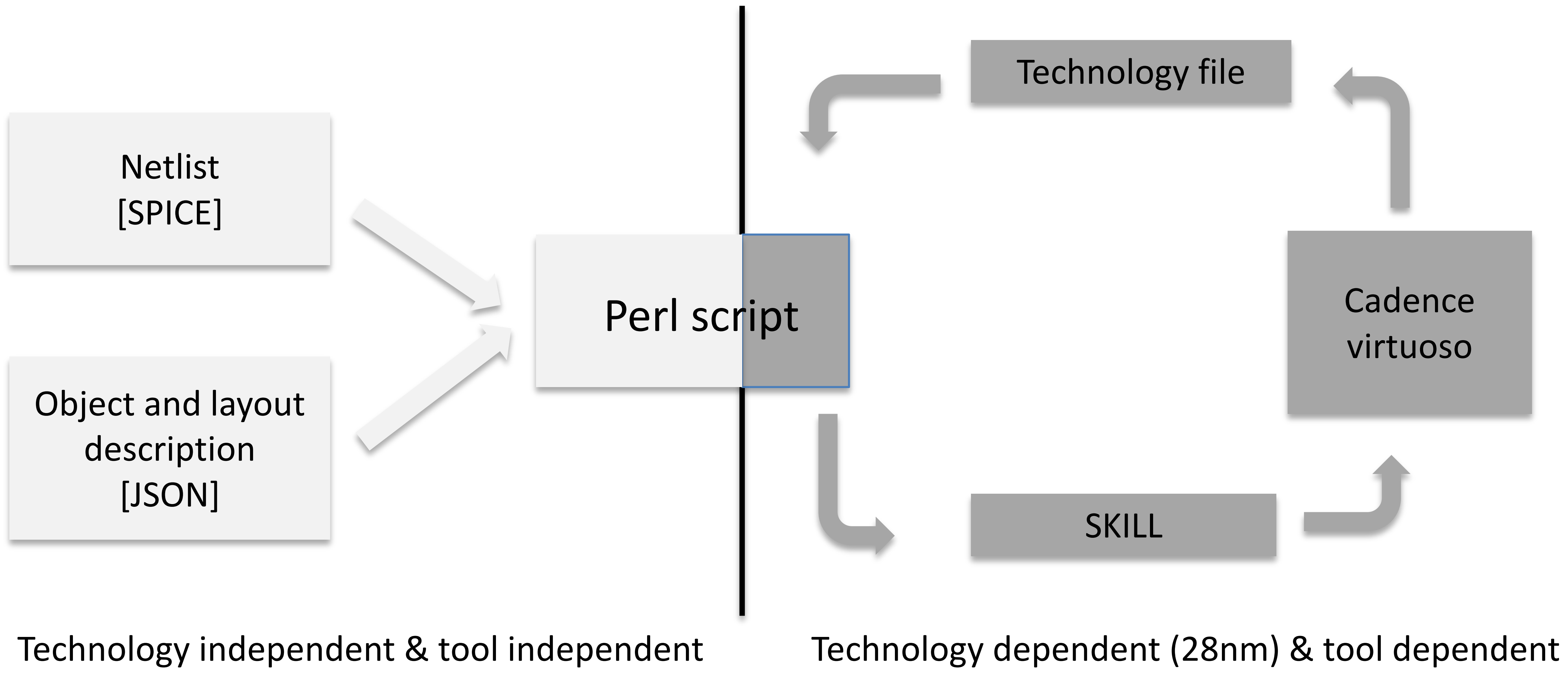

V_{DD}
$\overline{CK}$ M_{P0} M_{P3} N M_{P2} P M_{P1}
A
EI M_{N0}
P M_{N1} N M_{N2} M_{N3}
EO
V_{REF}
M_{P4} D_{P0} D_{N0} EI M_{N6} M_{N5} P M_{N4}
M_{P5} D_{N1} D_{P1} EI M_{N9} M_{N8} N M_{N7}
M_{P6} M_{P7}
B
CK M_{N10}
CI CO
(b) (c) (d)


$C = CK\_CMP$
V_{DD}
C
3
P
4
N
5
2
$+$
$-$
$\overline{C}$


10-bit Prototype 14-08-2014
10-bit second order compensated CDAC
10-bit common centroid CDAC
10-bit f**k recommended DRC rules
11-bit second order compensated CDAC
11-bit common centroid CDAC
9-bit
8-bit
9-bit IO voltage

	Weaver [5]	Harpe [9]	Patil [10]	Liu [11]	This work	
Technology (nm)	90	90	28 FDSOI	28	28 FDSOI	
Fsample (MS/s)	21	2	No sampling	100	2	20
Core area (mm^2)	0.18	0.047	0.0032	0.0047	0.00312	
SNDR (dB)	34.61	57.79	40	64.43	46.43	48.84
SFDR (dBc)	40.81	72.33	30	75.42	61.72	63.11
ENOB (bits)	5.45	6.7 - 9.4	6.35	10.41	7.42	7.82
Supply (V)	0.7	0.7	0.65	0.9	0.47	0.69
Pwr (μW)	1110	1.64 -3.56	24	350	0.94	15.87
Compiled	Yes	No	No	No	Yes	
FoM (fJ/c.step)	838	2.8 - 6.6	3.7	2.6	2.7	3.5

Compilation

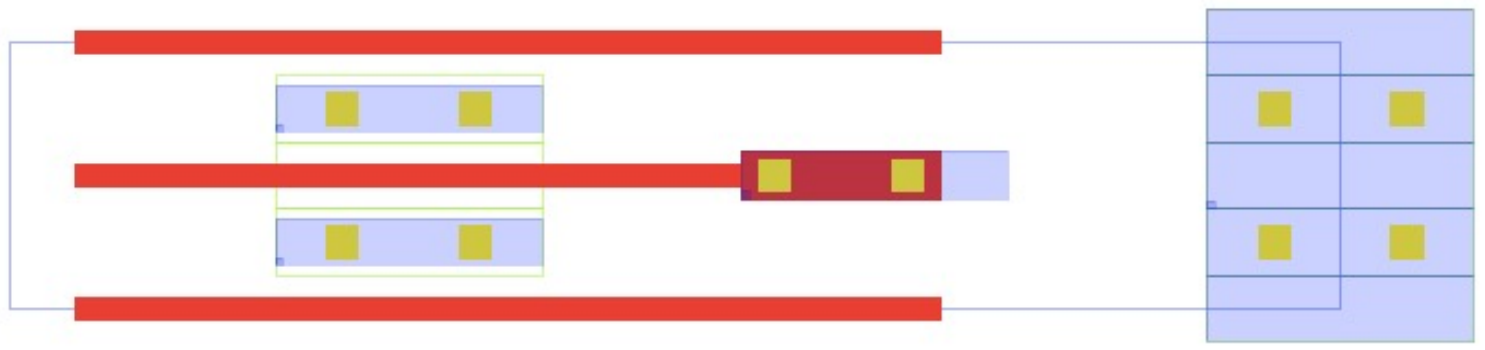

Netlist
[SPICE]
Object and layout description
[JSON]
Perl script
Technology file
Cadence virtuoso
SKILL
Technology independent & tool independent
Technology dependent (28nm) & tool dependent

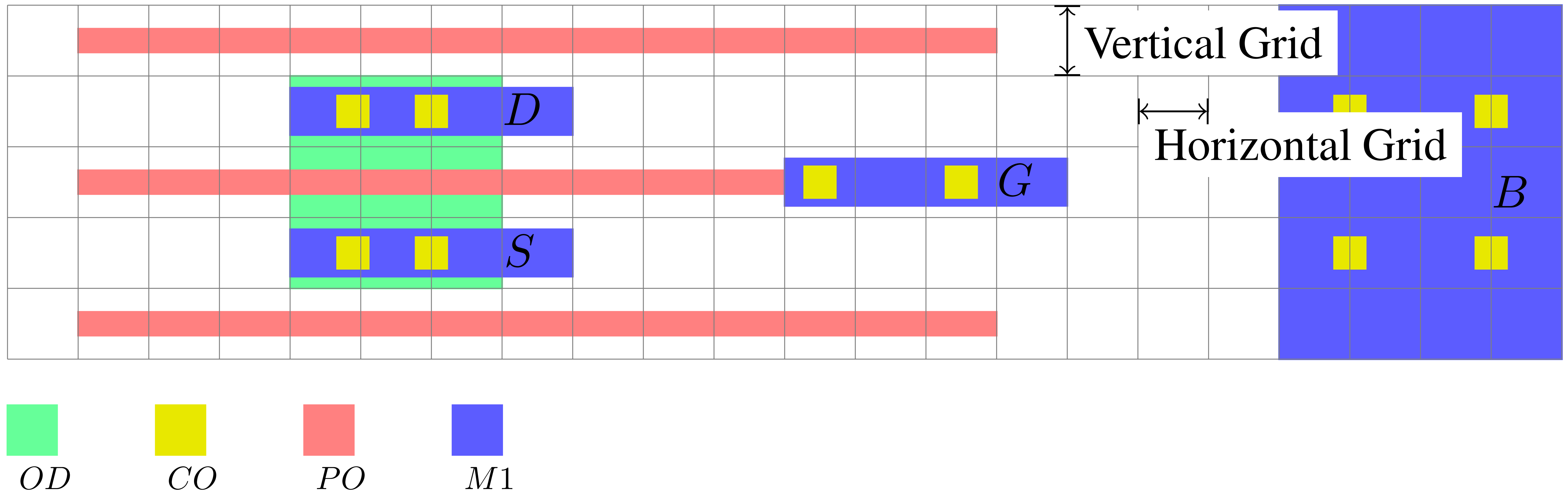
16 k Perl lines. Ported to C++ for speed ⇒ ciccreator

```
{ "name" : "DMOS" ,
  "class" : "Gds::GdsPatternTransistor",
  "yoffset": -0.5,
  "widthoffset" : -1,
  "fillCoordinatesFromStrings" : [
             [   "OD",
                 "------------------xxxx",
                 "----xCxC----------xCxC",
                 "----xxxx----------xxxx",
                 "----xCxC----------xCxC",
                 "------------------xxxx"
             ],
             [   "PO",
                 "-mmmmmmmmmmmmm--------",
                 "----------------------",
                 "-mmmmmmmmmmcxc--------",
                 "----------------------",
                 "-mmmmmmmmmmmmm--------"
             ],
             [   "M1",
                 "------------------xxxx",
                 "----wDww----------xxxx",
                 "-----------wGww---xBxx",
                 "----wSww----------xxxx",
                 "------------------xxxx"

             ]
       ]
}
```

- Structure and any other property is described in JSON (JavaScript Object Notation)
- "name" is the name of the cell
- "class" defines which object to use
- All other classes in the JSON object refer to object methods (there are some special functions, but more on that later)
- Convert a text string into a layout drawing
  - c = contact
  - C = center contact on rectangle left edge
  - x = fill rectangle
  - m = use minimum length poly
  - w = use "width" from techfile
  - DGSB = add ports




Vertical Grid
Horizontal Grid
D
G
S
B
OD
CO
PO
$M1$

```
{ "name": "IVX1_CV" ,
  "symbol" : "inv",
  "class" : "Layout::LayoutDigitalCell",
  "spice" : [
      ".subckt IVX1_CV A Y AVDD AVSS" ,
      "MN0 Y A AVSS AVSS NCHDL",
      "MP0 Y A AVDD AVSS PCHDL",
      ".ends IVX1_CV”],
  "addSchematicCoordinates" : {
      "MN0" : [ 0.25, 0, "R0"],
      "MP0" : [0.25, 0.5, "R0”]
  },
  "beforeRoute" : {
      "addDirectedRoutes" : [ ["M1","Y","MN:D-|--MP:D"], ["PO","A","MN:G-MP:G"] ]
  },
  "afterRoute"  : {
      "addPortOnRects" : [  ["A","M1", "MN0:G"] , ["Y", "M1", "MN0:D"]]
  }
}
```

What symbol to use, defaults to templates/skill/<name>.il

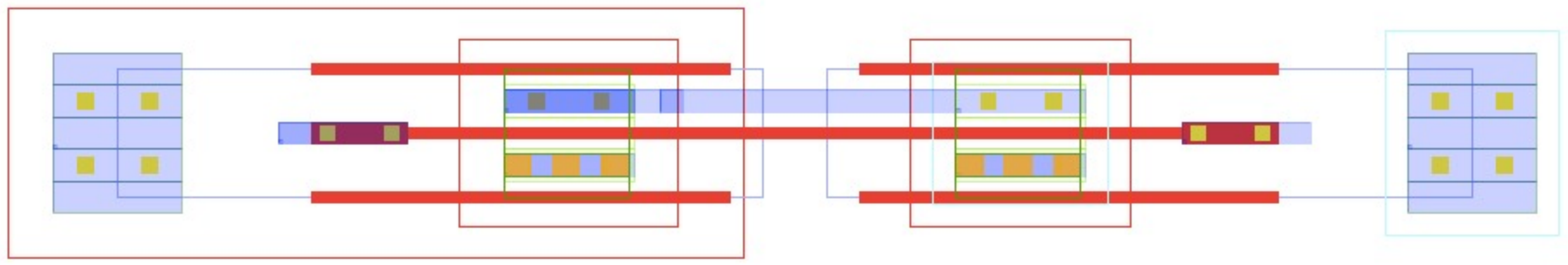
LayoutDigitalCell has extra functions for digital cells, and will add power rails.

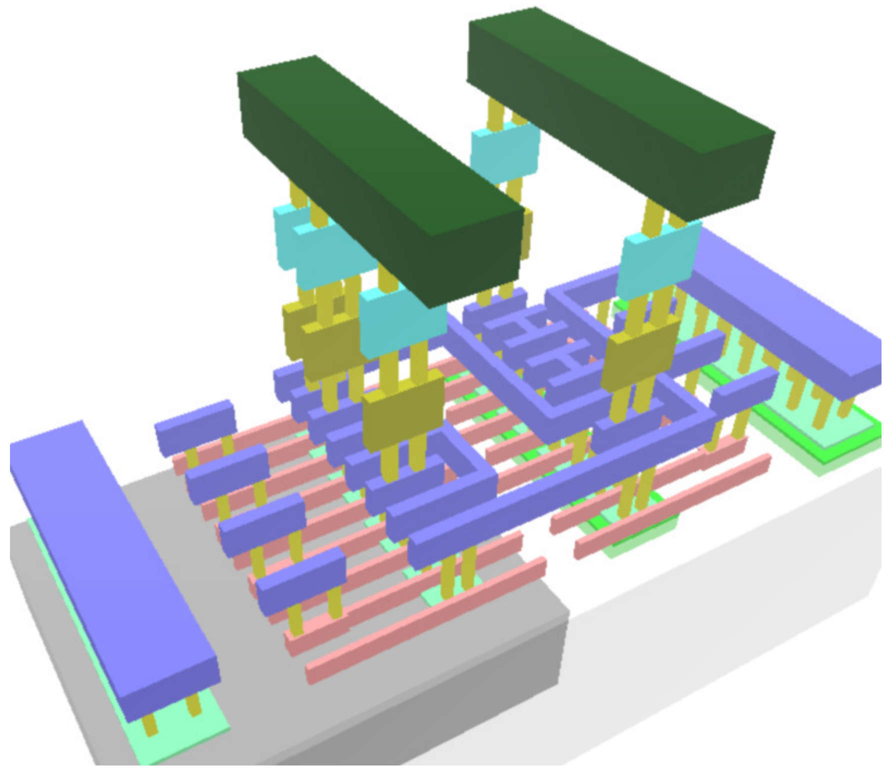
Connectivity defined by SPICE. SPICE subcircuit can be read from a separate file

Help the schematic generator to place transistors so it’s easier to read schematics

Find rectangle on device MN:D, and route in M1 to rectangle MP:D using a left, up or down, left pattern.

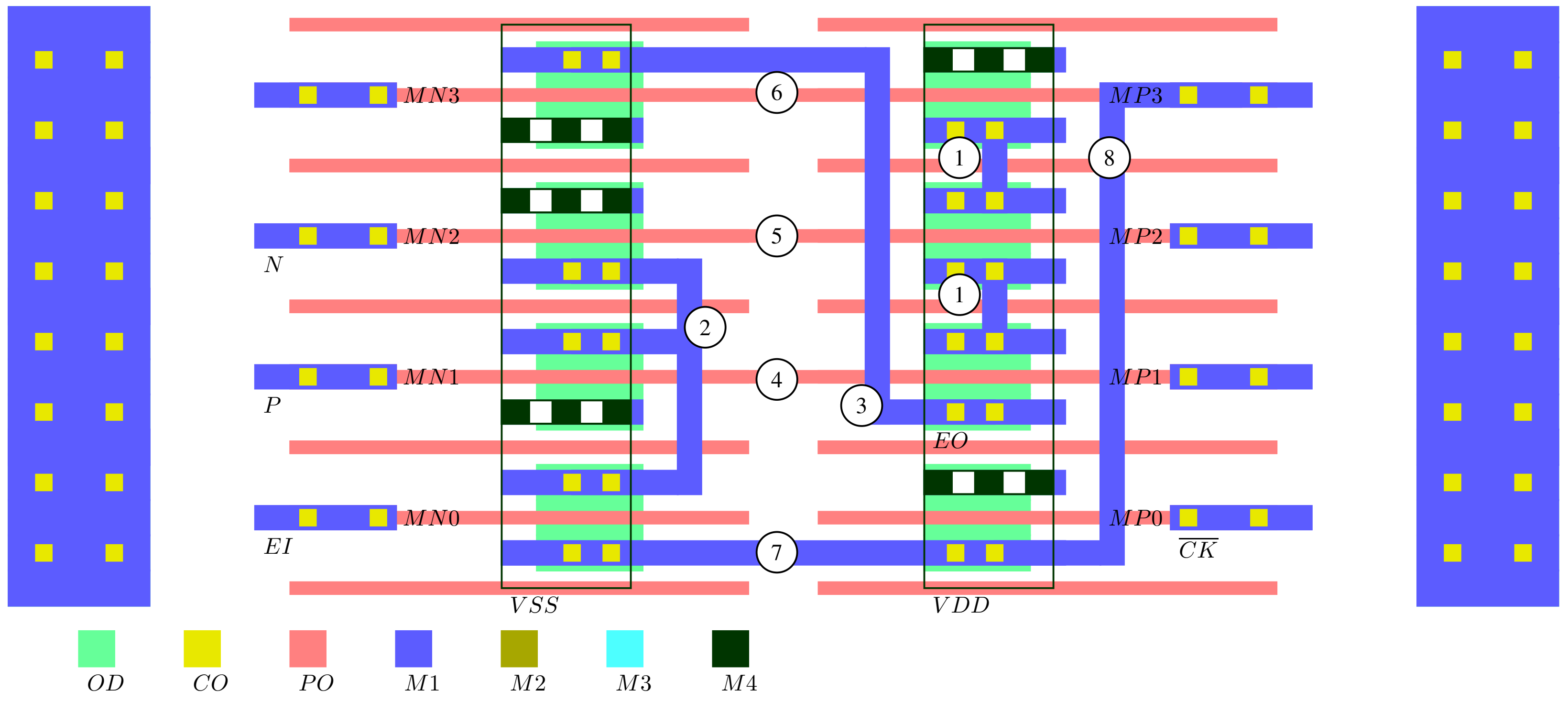
Add port for A on the gate of MN0




(a)

```
.SUBCKT SAREMX1_CV P  N EI EO CK_N AVDD AVSS
  MN0 N3   EI   A    AVSS NCHDL
  MN1 N3   P    AVSS AVSS NCHDL
  MN2 AVSS N    N3   AVSS NCHDL
  MN3 EO   A    AVSS AVSS NCHDL
  MP0 AVDD CK_N A    AVSS PCHDL
  MP1 N2   P    EO   AVSS PCHDL
  MP2 N1   N    N2   AVSS PCHDL
  MP3 AVDD A    N1   AVSS PCHDL
.ENDS
```
(b)

```
{ "name": "SAREMX1_CV",
  "class" : "Layout::LayoutDigitalCell",
  "addConnectivityRoutes": [
      ["M1","N1|N2","||",""],        1
      ["M1","N3","-|",""],           2
      ["M1","EO","--|-","onTopR"]    3
        ],
    "addDirectedRoutes" : [
      ["PO","P","MN1:G-MP1:G"],      4
      ["PO","N","MN2:G-MP2:G"],      5
      ["PO","A","MN3:G-MP3:G"],      6
      ["M1","A","MN0:S-MP0:S"],      7
      ["M1","A","MP0:S-|--MP3:G"]    8
      ]
    }
}
```
(c)

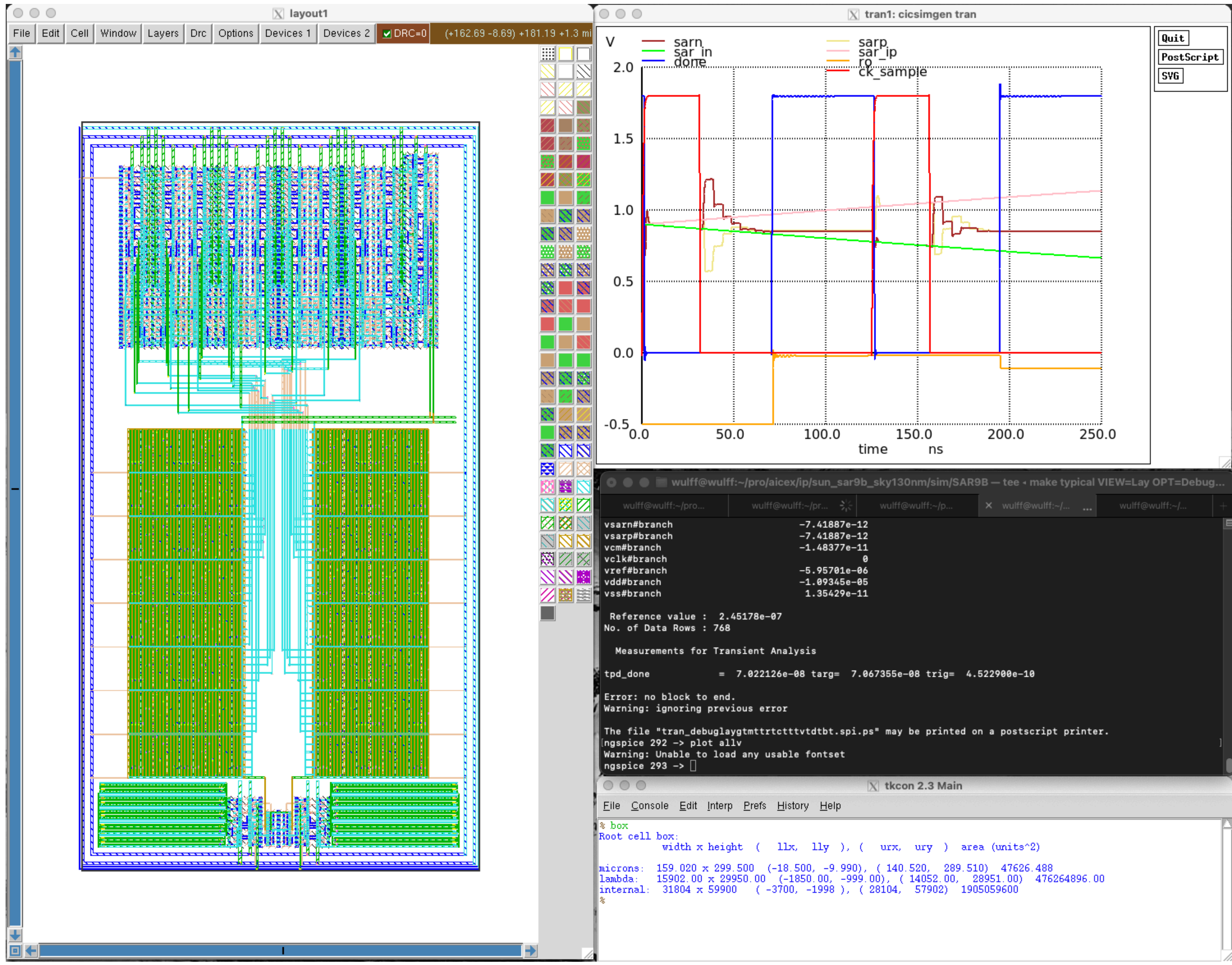


(d)

Since then

Measured: 28 nm FDSOI, 55 nm
Ported: 22 nm FDSOI, 22 nm, 28 nm, 65 nm, 130 nm

2022: There is an open source port to skywater 130nm!
[wulffern/sun_sar9b_sky130nm]



```
vsarn#branch                    -7.41887e-12
vsarp#branch                    -7.41887e-12
vcm#branch                      -1.48377e-11
vclk#branch                                0
vref#branch                     -5.95701e-06
vdd#branch                      -1.09345e-05
vss#branch                       1.35429e-11

 Reference value :  2.45178e-07
No. of Data Rows : 768

  Measurements for Transient Analysis

tpd_done            =  7.022126e-08 targ=  7.067355e-08 trig=  4.522900e-10

Error: no block to end.
Warning: ignoring previous error

The file "tran_debuglaygtmttrtctttvtdtbt.spi.ps" may be printed on a postscript printer.
ngspice 292 -> plot allv
Warning: Unable to load any usable fontset
ngspice 293 -> 
```

```
% box
Root cell box:
           width x height  (   llx,  lly  ), (   urx,  ury  )  area (units^2)

microns:  159.020 x 299.500  (-18.500, -9.990), ( 140.520,  289.510)  47626.488
lambda:   15902.00 x 29950.00  (-1850.00, -999.00), ( 14052.00,  28951.00)  476264896.00
internal:  31804 x 59900   ( -3700, -1998 ), ( 28104,  57902)  1905059600
%
```

Super simple transistor was a good choice for portability

```
//sky130
{ "name" : "DMOS_BULKN" ,
  "class" : "Gds::GdsPatternTransistor",
  "abstract" : 1,
  "yoffset": -0.5,
  "widthoffset" : -0.5,
   "fillCoordinatesFromStrings" : [
       [   "OD",
          "-------------------",
          "----xxx------------",
          "----xxx------------",
          "----xxx------------",
          "-------------------"
      ],
      ...
      [   "M1",
          "----------------xxx",
          "----wDw---------xxx",
          "----------wGw---xBx",
          "----wSw---------xxx",
          "----------------xxx"
      ],
      ...
      [   "NDIFFC",
          "-------------------",
          "----LTR------------",
          "-------------------",
          "----LTR------------",
          "-------------------"
      ]
  ]
}
```

```
//28nm FDSOI
{ "name" : "DMOS" ,
  "class" : "Gds::GdsPatternTransistor",
  "yoffset": -0.5,
  "type": "pch",
  "widthoffset" : -1,
  "fillCoordinatesFromStrings" : [
   [  "OD",
      "------------------xxxx",
      "----xxK-----------xCxC",
      "----xxx-----------xxxx",
      "----xxK-----------xCxC",
      "------------------xxxx"
   ],
   [  "PO",
      "-mmmmmmmmmmmmm--------",
      "----------------------",
      "-mmmmmmmmmmcxc--------",
      "----------------------",
      "-mmmmmmmmmmmmm--------"
   ],
   [  "M1",
      "------------------xxxx",
      "----wDww----------xxxx",
      "-----------wGww---xBxx",
      "----wSww----------xxxx",
      "------------------xxxx"
   ]
  ],
  "afterNew" : {
    "copyColumns" :[
      { "count" : 0, "offset" : 4,"length" : 4}
    ]
  }
}
```

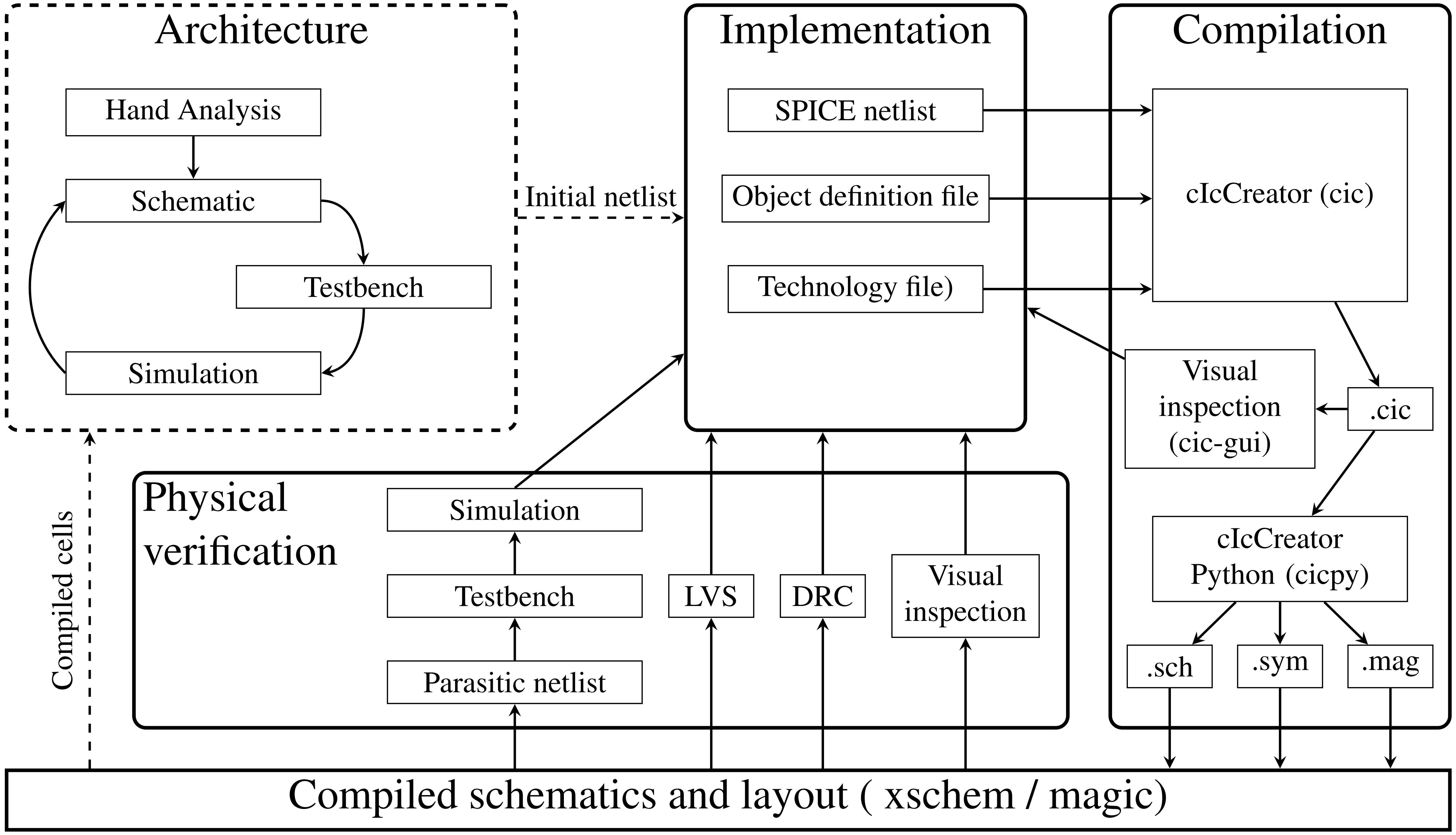
2016 (Perl compiler)

```
{ "name": "SARCMPHX1_CV",
  "description" : "Half a strong-arm comparator",
  "class" : "Layout::LayoutDigitalCell",
  "setYoffsetHalf" :  "" ,
  "rows" : 7,
  "beforeRoute" : {
    "addDirectedRoutes" : [ ["PO","VMR","MN6:G-MP6:G"],
                            ["M1","VMR","MP4:G||MP6:G"],
                            ["M1","CI","MN1:G||MN5:G"],
                            ["M1","N2","MN1:D,MN3:D,MN5:D-|--MP1:D"],
                            ["M1","N1","MN0:D,MN2:D|-MN4:D"],
                            ["M1","N1","MN0:D-|--MP0:S"],
                            ["M1","CO","MP3:D,MP5:D--|-MN6:D"],
                            ["PO","CK","MN0:G-MP0:G"],
                            ["M1","CK","MP0:G,MP1:G-|MP3:G"],
                            ["M4","NC","MP2$:D--|--MP2:G"]
                          ]
    },
    "afterRoute" : {
    "addPortOnRects" : [ ["AVDD","M4" ],
       ["N1","M1","MN4:D"],
       ["N2","M1","MN5:D" ]]
 }
}
```

2022 (C++ compiler)

```
{ "name": "SARCMPHX1_CV",
  "description" : "Half a strong-arm comparator",
  "class" : "Layout::LayoutDigitalCell",
  "setYoffsetHalf" :  1 ,
  "rows" : 7,
  "meta" : {
       "noSchematic" : true
  },
  "decorator" : [
     {"ConnectSourceDrain" : ["M1","||",""]}
  ],
  "beforeRoute" : {
    "addDirectedRoutes" : [ ["PO","VMR","MN6:G-MP6:G"],
                            ["M1","VMR","MP4:G||MP6:G"],
                            ["M1","CI","MN1:G||MN5:G"],
                            ["M1","N2","MN1:D,MN3:D,MN5:D-|--MP1:D"],
                            ["M1","N1","MN0:D,MN2:D|-MN4:D"],
                            ["M1","N1","MN0:D-|--MP0:S"],
                            ["M1","CO","MP3:D,MP5:D--|-MN6:D"],
                            ["PO","CK","MN0:G-MP0:G"],
                            ["M1","CK","MP0:G,MP1:G-|MP3:G"],
                            ["M4","NC","MP2$:D-|--MP2:G"]
                          ]
    },
    "afterRoute" : {
              "addPortOnRects" : [["BULKP","M1"],
                  ["BULKN","M1"],
                  ["AVDD","M4" ],
                  ["N1","M1","MN4:D"],
                  ["N2","M1","MN5:D" ]]
    }
}
```

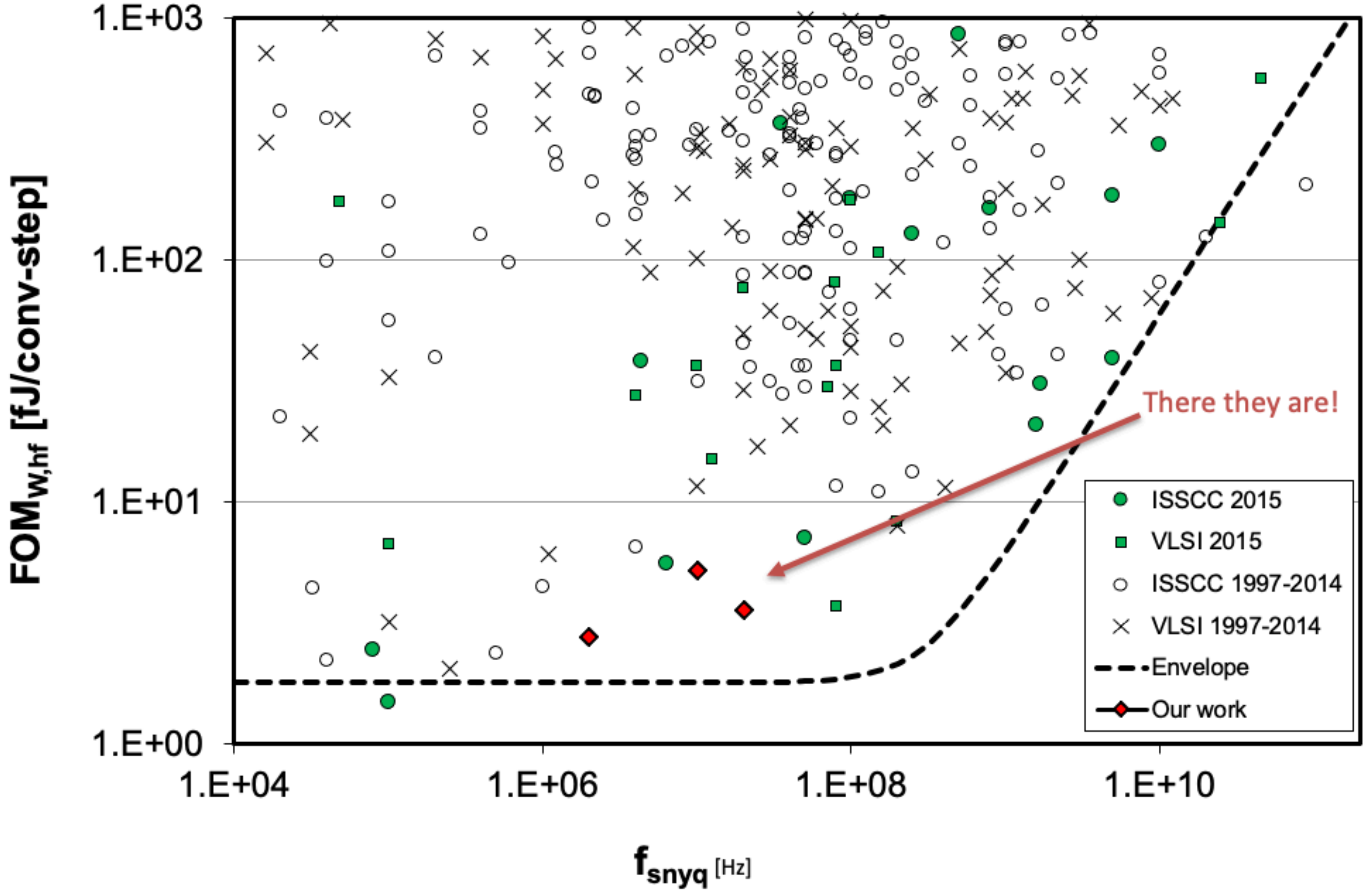
Usage is hard, requires a new type of analog designer/programmer


Architecture
Hand Analysis
Schematic
Testbench
Simulation
Initial netlist
Implementation
SPICE netlist
Object definition file
Technology file)
Compilation
cIcCreator (cic)
Visual inspection (cic-gui)
.cic
cIcCreator Python (cicpy)
.sch
.sym
.mag
Physical verification
Simulation
Testbench
Parasitic netlist
LVS
DRC
Visual inspection
Compiled cells
Compiled schematics and layout (xschem / magic)

ciccreator

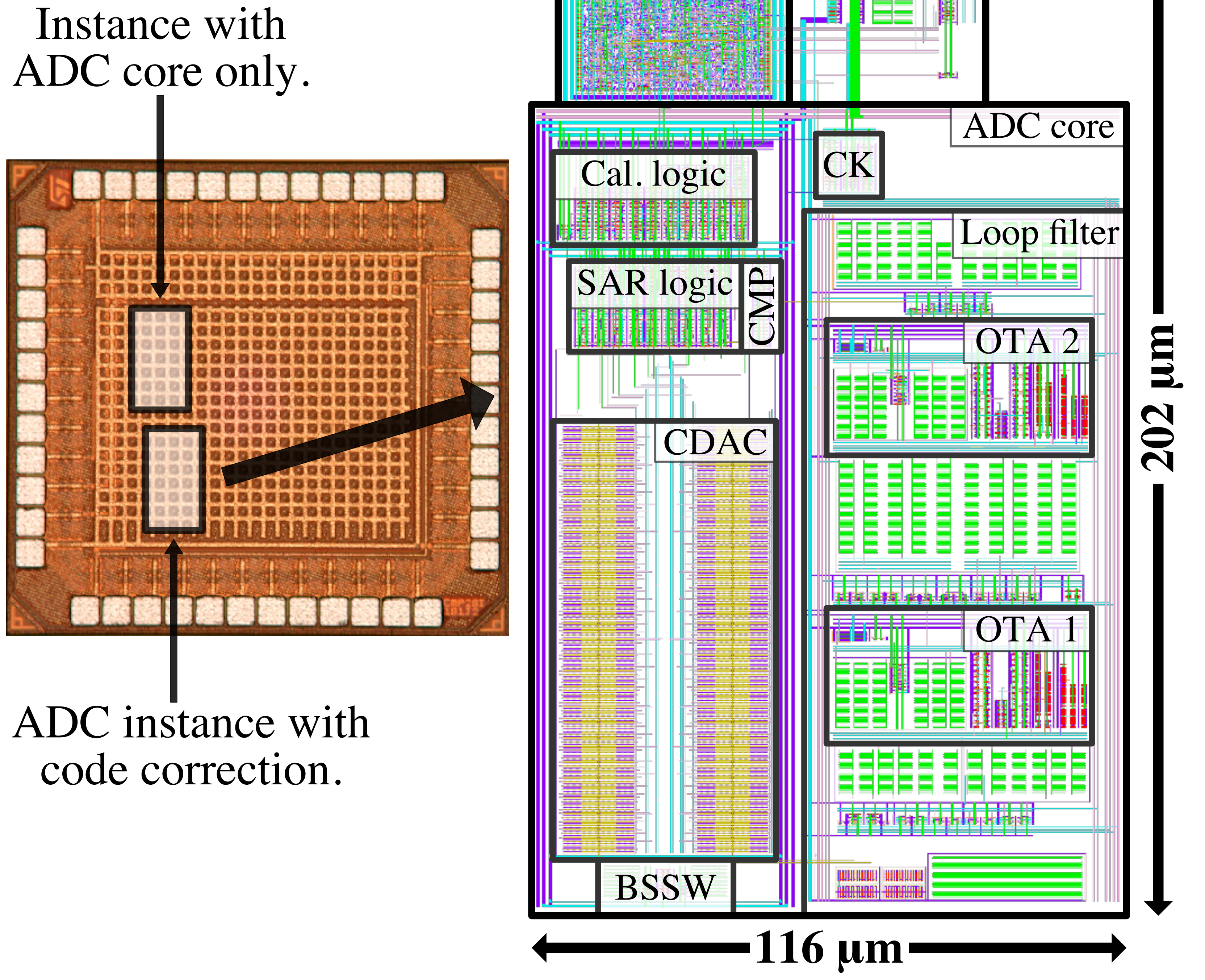
ciccreator docs

cicpy



[1] A Compiled 9-bit 20-MS/s 3.5-fJ/conv.step SAR ADC in 28-nm FDSOI for Bluetooth Low Energy Receivers

[2] A 68 dB SNDR Compiled Noise-Shaping SAR ADC With On-Chip CDAC Calibration


Instance with ADC core only.
ADC instance with code correction.
Code correction
Power sw. Output reg.
ADC core
CK
Cal. logic
SAR logic
CMP
Loop filter
OTA 2
CDAC
OTA 1
BSSW
116 µm
202 µm

Things I want to show

- Tiny Tapeout Process
- Schematics
- Layout
- DRC/LVS
- Parasitic extraction
- Verification plan and simulations
- Delivery
- Iteration

Thanks!