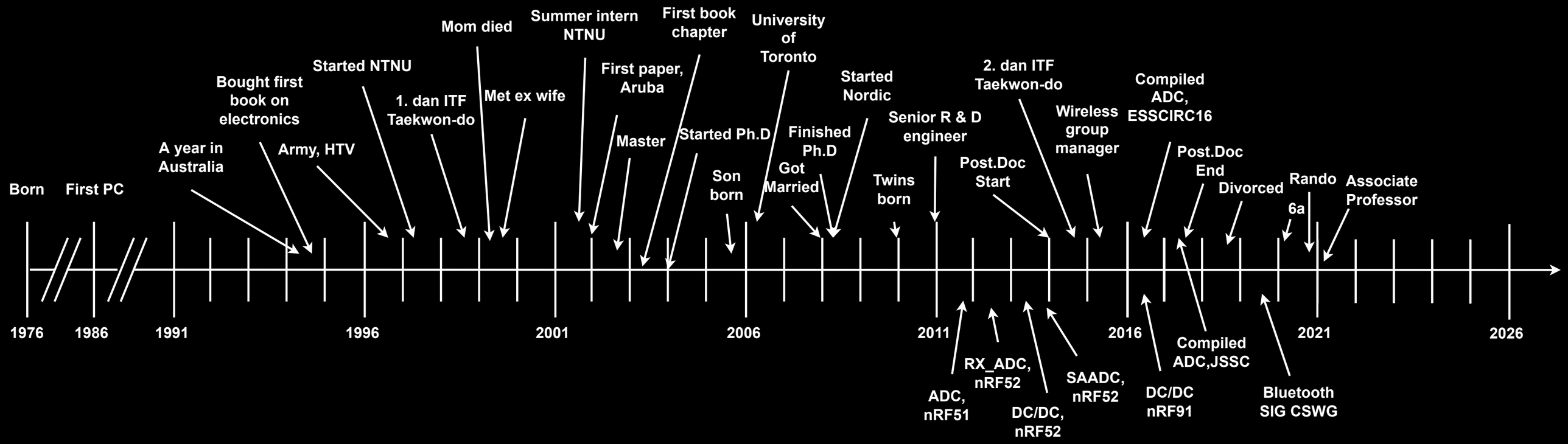
My thoughts on analog design


Born
First PC
A year in Australia
Army, HTV
Bought first book on electronics
Started NTNU
1. dan ITF Taekwon-do
Mom died
Met ex wife
Summer intern NTNU
First paper, Aruba
Master
First book chapter
Started Ph.D
Son born
University of Toronto
Got Married
Finished Ph.D
Started Nordic
Twins born
Senior R & D engineer
ADC, nRF51
RX_ADC, nRF52
DC/DC, nRF52
SAADC, nRF52
Post.Doc Start
2. dan ITF Taekwon-do
Wireless group manager
DC/DC nRF91
Compiled ADC,JSSC
Compiled ADC, ESSCIRC16
Post.Doc End
Divorced
Rando 6a
Bluetooth SIG CSWG
Associate Professor
1976
1986
1991
1996
2001
2006
2011
2016
2021
2026

How I see our roles

Professors: Guide students on what is impossible, possible, and hints on what might be possible

Ph.D students: Venture into the unknown and make something (more) possible

Master students: Learn all that is currently possible

Bachelor students: Learn how to make complicated into easy

Industry: Take what is possible, and/or complicated, and make it easy

The world is analog. The laws of behavior are written in the mathematics of calculus [1]

$$\oint_{\partial\Omega} \mathbf{E} \cdot d\mathbf{S} = \frac{1}{\epsilon_0} \iiint_V \rho \cdot dV$$

Relates net electric flux to net enclosed electric charge

$$\oint_{\partial\Omega} \mathbf{B} \cdot d\mathbf{S} = 0$$

Relates net magnetic flux to net enclosed magnetic charge

$$\oint_{\partial\Sigma} \mathbf{E} \cdot d\ell = -\frac{d}{dt} \iint_\Sigma \mathbf{B} \cdot d\mathbf{S}$$

Relates induced electric field to changing magnetic flux

$$\oint_{\partial\Sigma} \mathbf{B} \cdot d\ell = \mu_0 \left(\iint_\Sigma \mathbf{J} \cdot d\mathbf{S} + \epsilon_0 \frac{d}{dt} \iint_\Sigma \mathbf{E} \cdot d\mathbf{S} \right)$$

Relates induced magnetic field to changing electric flux and to current density

---

[1] Maxwell's equations

The behavior of electrons is written in quantum mechanics

$$\psi(x,t) = Ae^{j(kx-\omega t)}$$

Probability amplitude of an electron

$$\frac{1}{2m}\frac{\hbar}{j^2}\frac{\partial^2}{\partial^2 x}\psi(x,t) + U(x)\psi(x,t) = -\frac{\hbar}{j}\frac{\partial}{\partial t}\psi(x,t)$$

Space and time evolution of an electron (Schrödinger equation)

$$n = \int_{E_1}^{E_2} N(E)f(E)dE$$

Electron density (Density of states)

$$f(E) = \frac{1}{e^{(E_i - E_F)/kT} + 1}$$

Relates the average number of fermions in thermal equlilibrium to the energy of a single-particle state (Fermi-Dirac statistics)

The abstract digital world is written in the mathematics of boolean algebra[4]

$$1 = \mathrm{True},\ 0 = \mathrm{False}$$

A	B	NOT(A AND B)
0	0	1
0	1	1
1	0	1
1	1	0

All digital processing can be made with the NOT(A AND B) function!

[4] Boolean algebra

People that design digital circuits can reuse the work of others

People that design analog circuits can learn from others, but need to deal with the real world on their own

Should we do as much as possible in the abstract digital world?

Worlds first commercial ADC

19" × 15" × 26"

150 lbs

$8,500.00

Courtesy,
Analogic Corporation
8 Centennial Drive
Peabody, MA 01960

http://www.analogic.com

Figure 5: 1954 "DATRAC" 11-Bit, 50-kSPS SAR ADC
Designed by Bernard M. Gordon at EPSCO

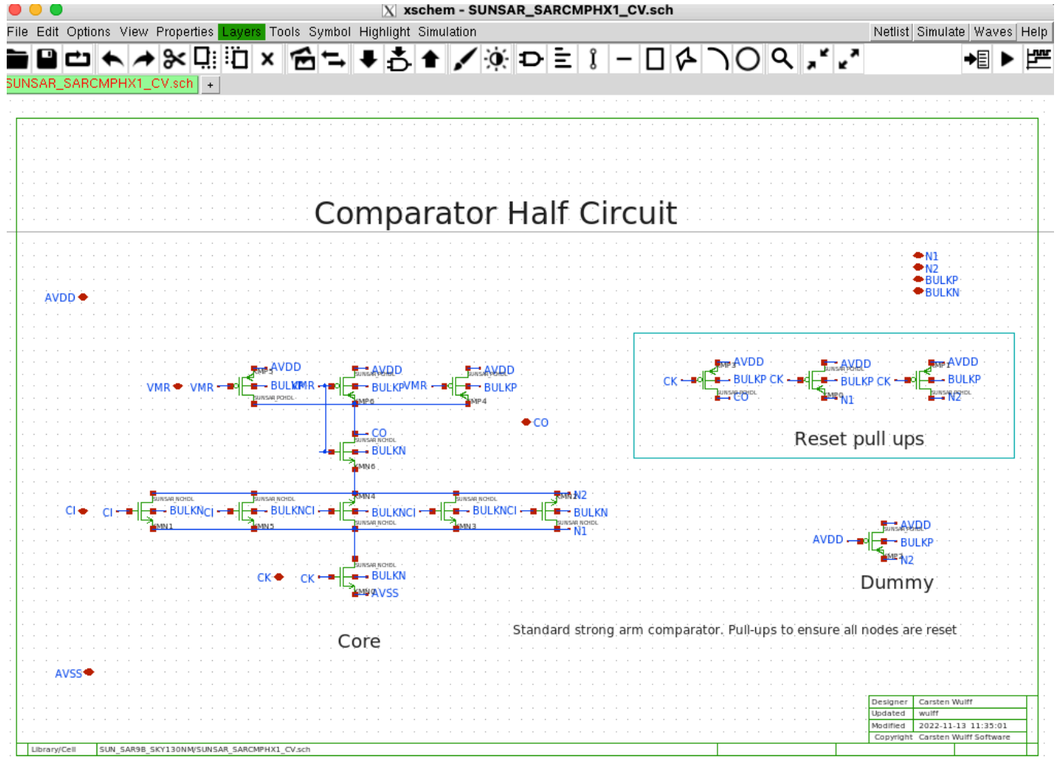
One of the first commerical offerings of a successive approximation analog-to-digital converter [5]

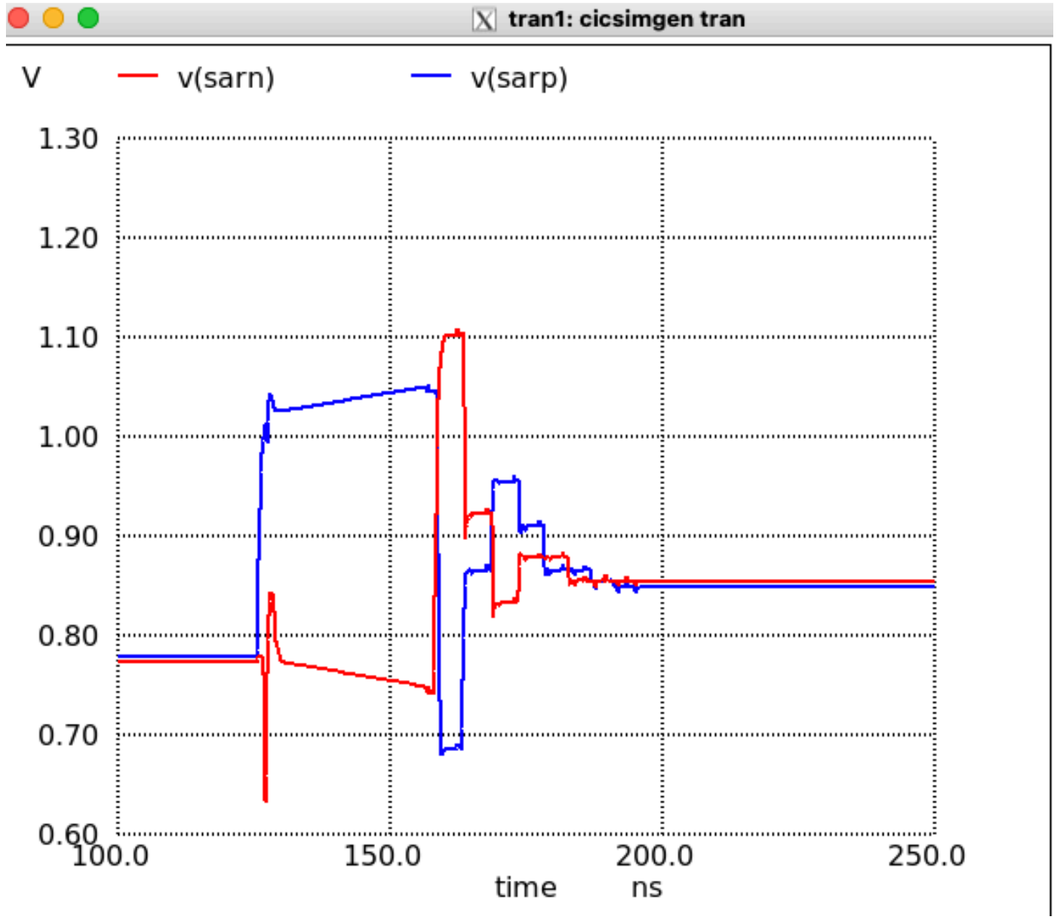
[5] https://www.analog.com/media/en/training-seminars/tutorials/mt-021.pdf

There will always be analog circuits, because the real world is analog

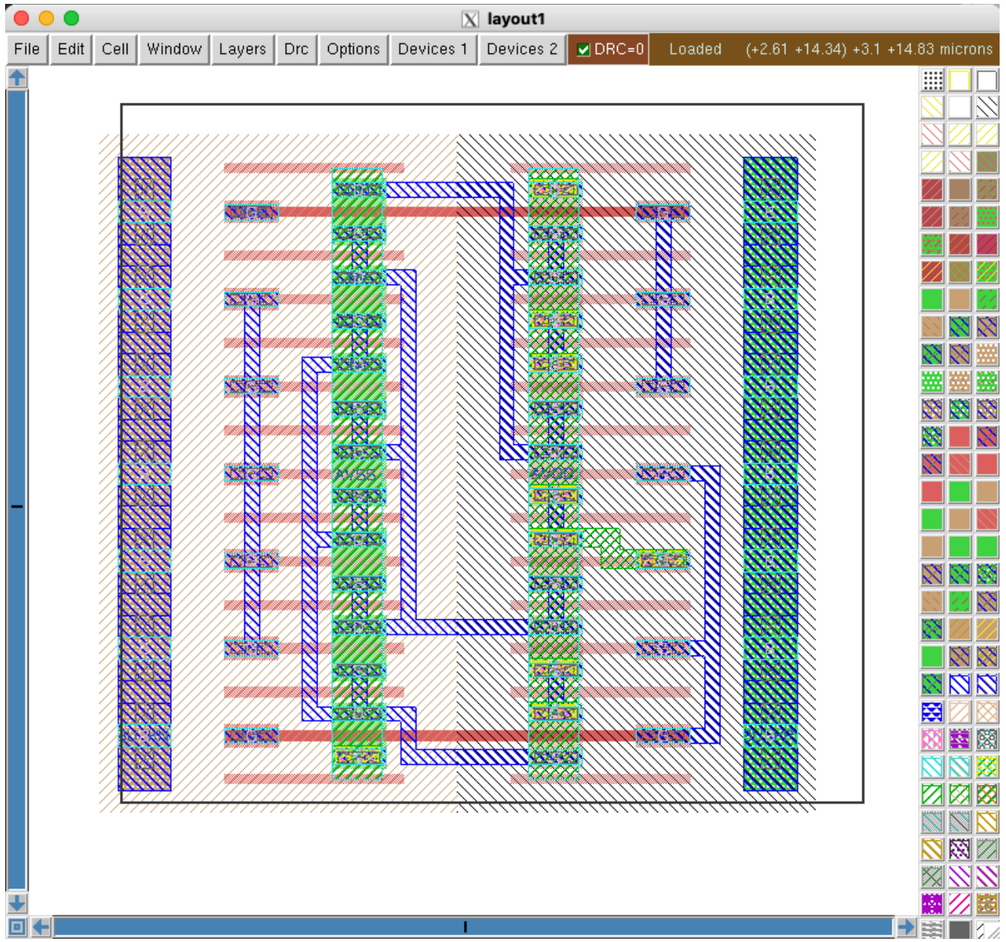
Why is reuse of analog circuits hard?

Life of an analog designer: Schematic Design





Life of an analog designer: Layout Design



My journey on "How can I simplify analog design?"

Trigger

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*

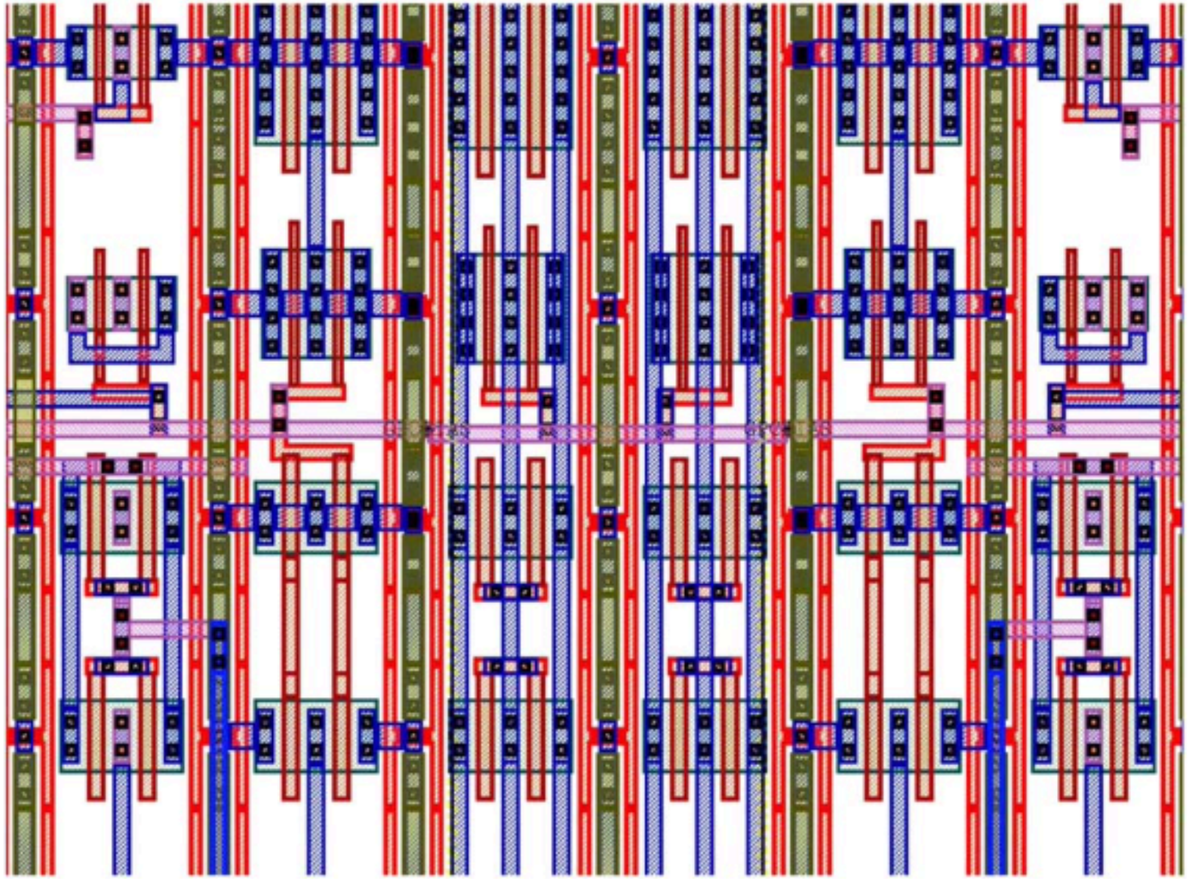
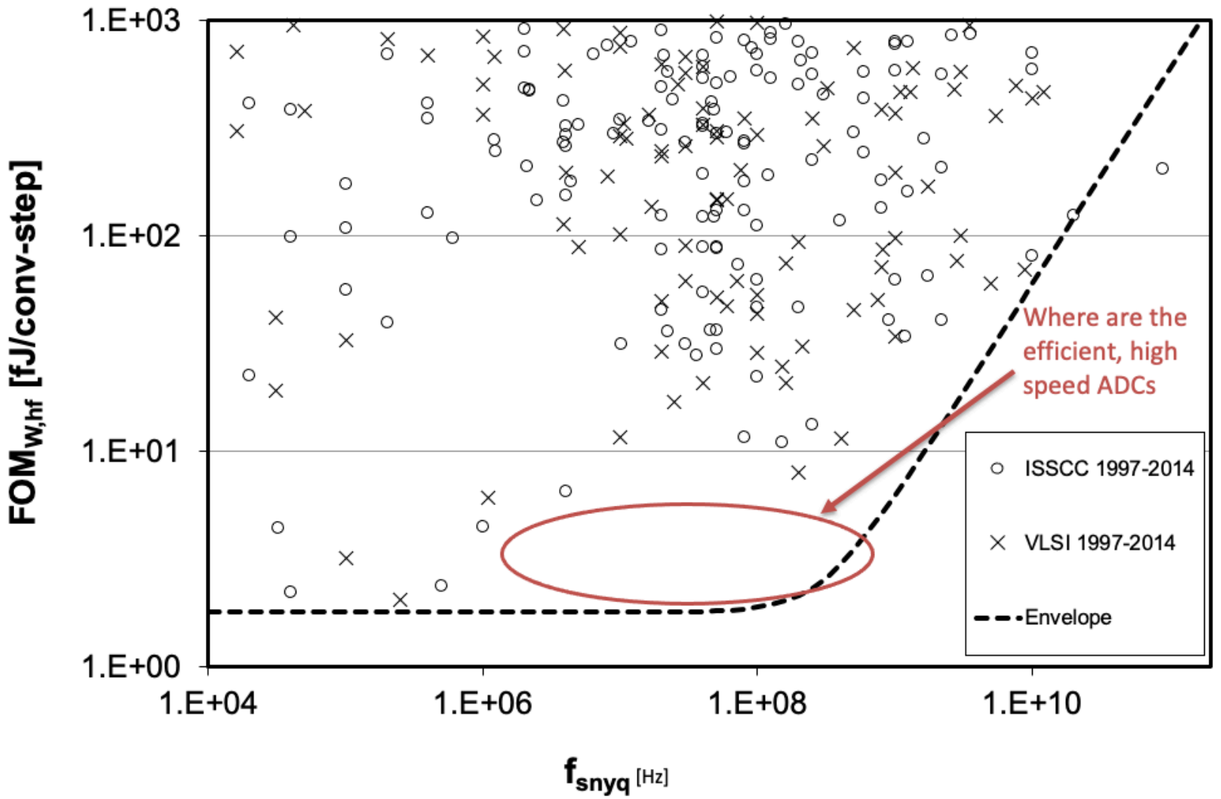
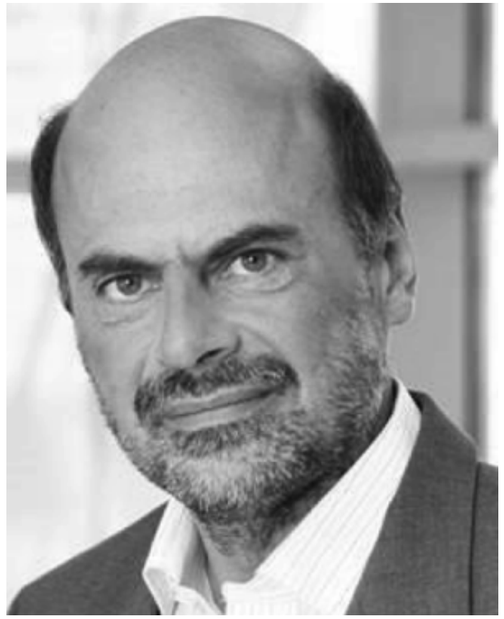


Fig. 6. *A portion of an amplifier cell with regular device pitch in both X and Y directions (upper metal layers removed for clarity). For best HF performance, all devices' substrate ties are placed on either side of two-finger gate patterns. Grounded stripes of poly are interposed between device active 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


1.E+03
1.E+02
1.E+01
1.E+00
1.E+04
1.E+06
1.E+08
1.E+10
FOM$_{W,hf}$ [fJ/conv-step]
f_{snyq} [Hz]
Where are the efficient, high speed ADCs
ISSCC 1997-2014
VLSI 1997-2014
Envelope

Architecture



14.7 A 6b 600MHz 10mW ADC Array in Digital 90nm CMOS

Dieter Draxelmayr

Infineon Design Centers, Villach, Austria

Low power A/D conversion is the key to many applications, especially in portable equipment. Therefore much effort has been put into the creation of low-power architectures and low-power designs [4,5]. In the last few years, we also saw the advent of parallel ADC structures designed to achieve higher speed than possible before. However, using parallelism we also can exploit the power efficiency of simple ADC structures in order to get high performance A/D conversion at low power. In this paper, eight successive approximation ADCs have been put in parallel to get high throughput at low power. We have achieved a sampling rate of 600MHz at a power consumption of 10mW, which to our knowledge is the lowest power reported for high speed ADCs.

which are the buffer RAM. The nine smaller blocks next to them are the nine converters. In principle the converter array consists of eight converters, but there is also a nineth converter for evaluation purposes.

The chip has been fabricated in a "digital" 90nm CMOS process with 6 metal layers. The capacitors are formed with regular metallization layers, so no MIM-cap has been used. Due to the low analog requirements of the charge-redistribution architecture, only regular threshold transistors have been used. The chip operates at a supply from 1 to 1.2V. Current consumption is 8.5mA in the analog portion, as predicted.

Converter arrays are known to suffer from several mismatch effects. In principle, any kind of mismatch between the converters may generate additional error components. Figure 14.7.4 shows a spectrum of the array taken at 600MHz clock and 329MHz input frequency. We see several spurious tones which take the SINAD down to 24.6dB. The dominant tones come from the individual offset values. We can remove the offsets by sub-

Plan

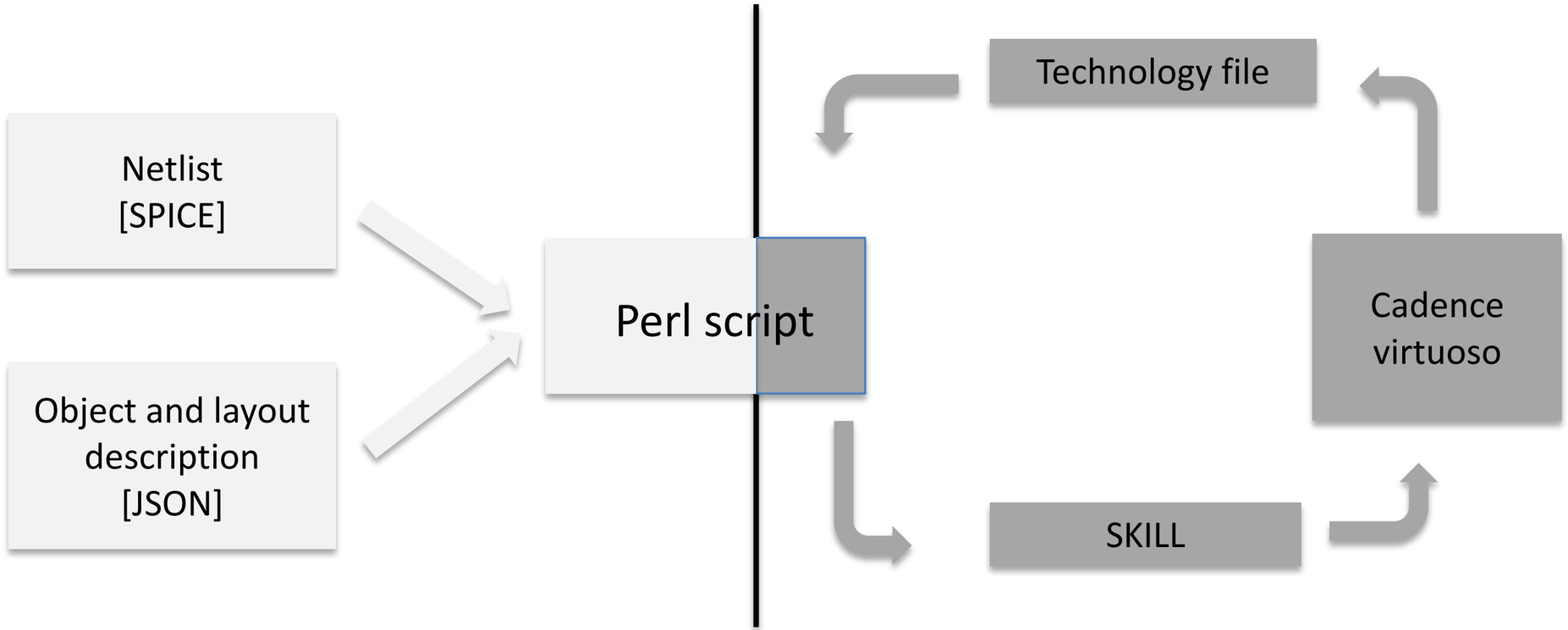
9-bit SAR ADC with 28 nm FDSOI transistors

9-bit SAR ADC with IO voltage (180 nm) FDSOI transistors

How to make multiple SAR ADCs with limited time?

Spend 50% of time for 6 months to **develop** a tool to make SAR ADCs

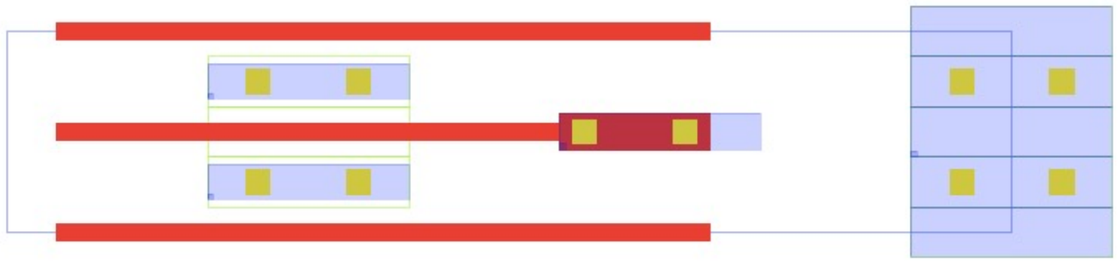
Spend 50% of time for 6 months to **make** the SAR ADCs



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



```
{ "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

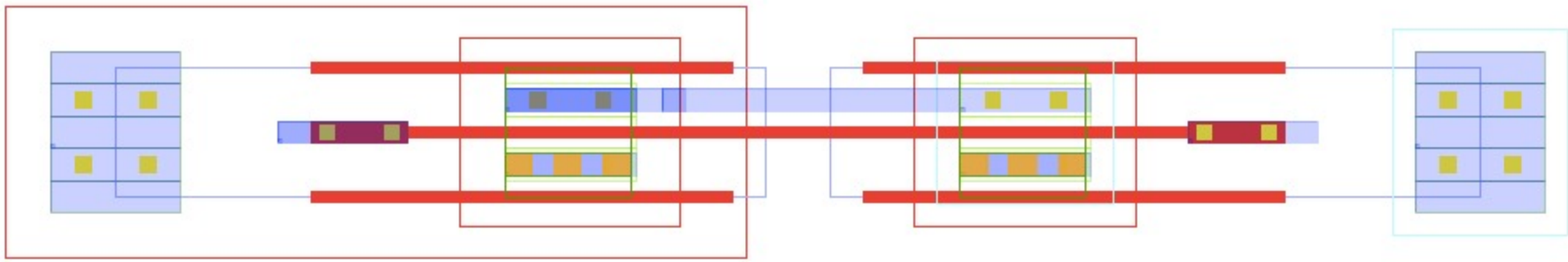
LayoutDigitalCell has extra functions for digital cells, and will add power rails.

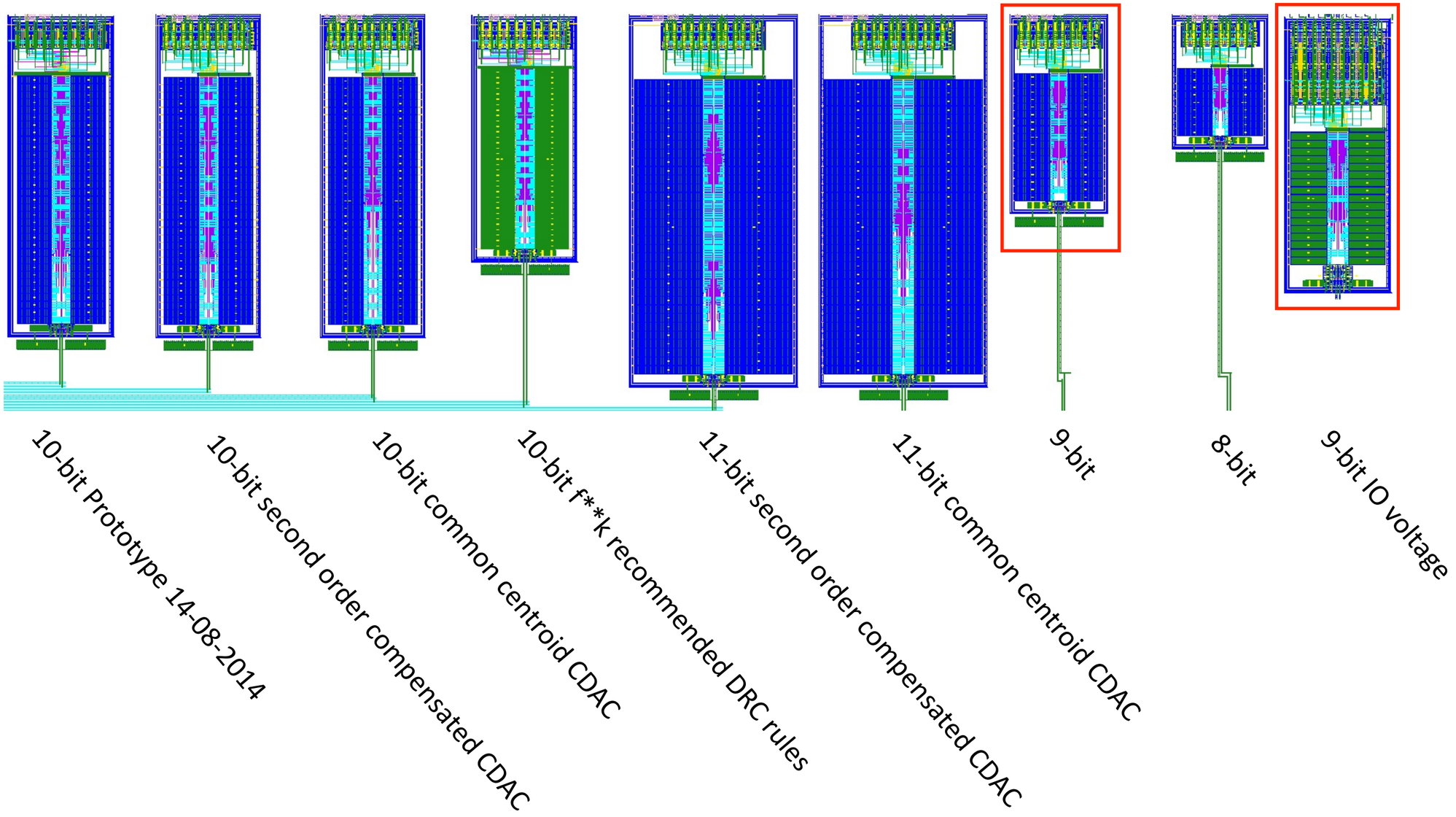
Connectivity defined by SPICE. SPICE subcircuit can be read from a separate file

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




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

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

	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

14-2 (7055)

IEEE Asian Solid-State Circuits Conference
November 4 – 6, 2019
The Parisian Macao, Macao SAR, China

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

Harald Garvik*, Carsten Wulff† and Trond Ytterdal*
Email: harald.garvik@ntnu.no
*Dept. of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
†Nordic Semiconductor, Trondheim, Norway

***Abstract*—This paper[1] presents a noise-shaping SAR ADC with an on-chip, foreground capacitive DAC (CDAC) calibration system. At start-up, the ADC uses the LSBs in the CDAC to measure and digitize the errors of the MSBs. A synthesized digital module accumulates the noise-shaped measurements, computes calibration coefficients, and corrects ADC codes at run-time. The loop filter implements two optimal zeros and two poles, and achieves 27.8 dB in-band attenuation at an oversampling rate (OSR) of 4. The prototype is implemented in 28 nm FDSOI, and achieves 68.2 dB SNDR at 5 MHz bandwidth, while consuming 108.7 µW from a 0.8 V supply. The Walden FOM is 5.2 fJ/conv.-step. The layout of the ADC is compiled from a netlist, a rule file, and an object definition file.**

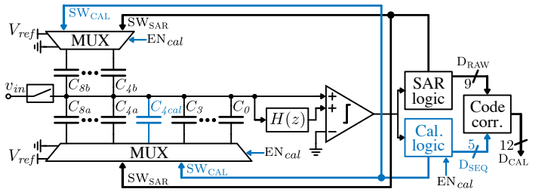
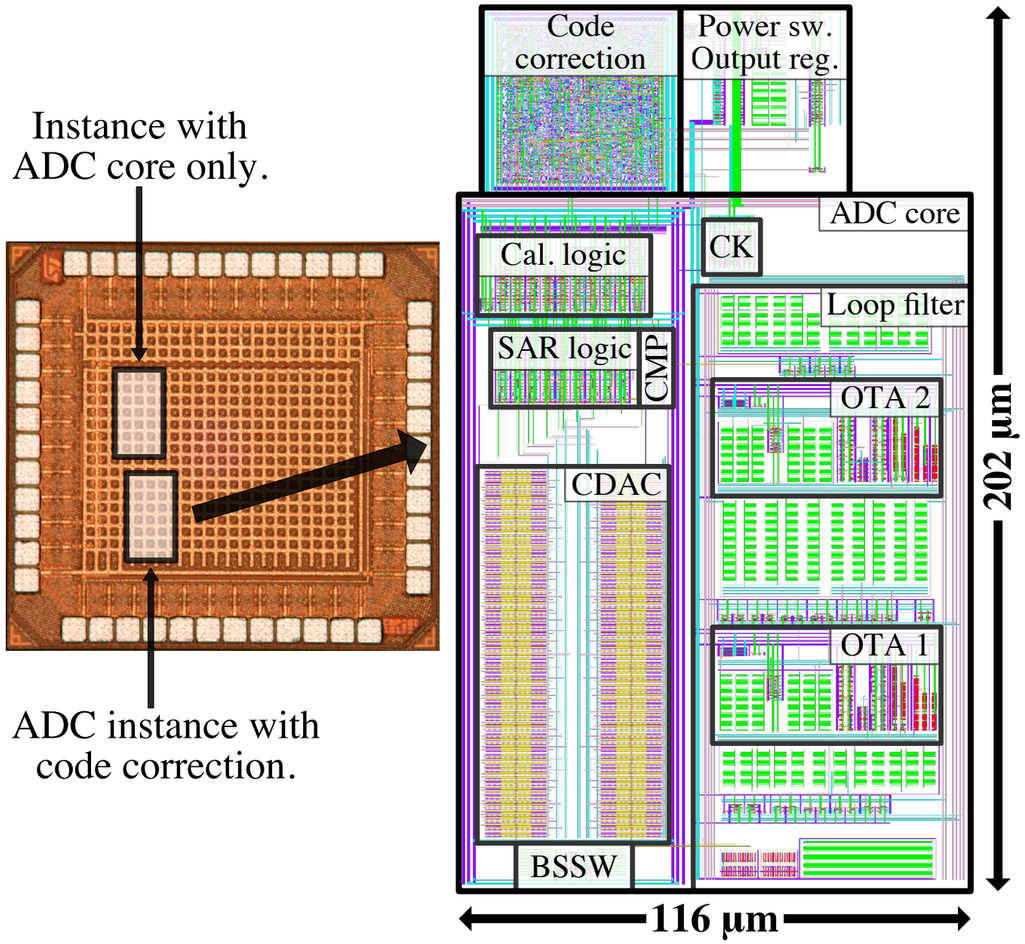


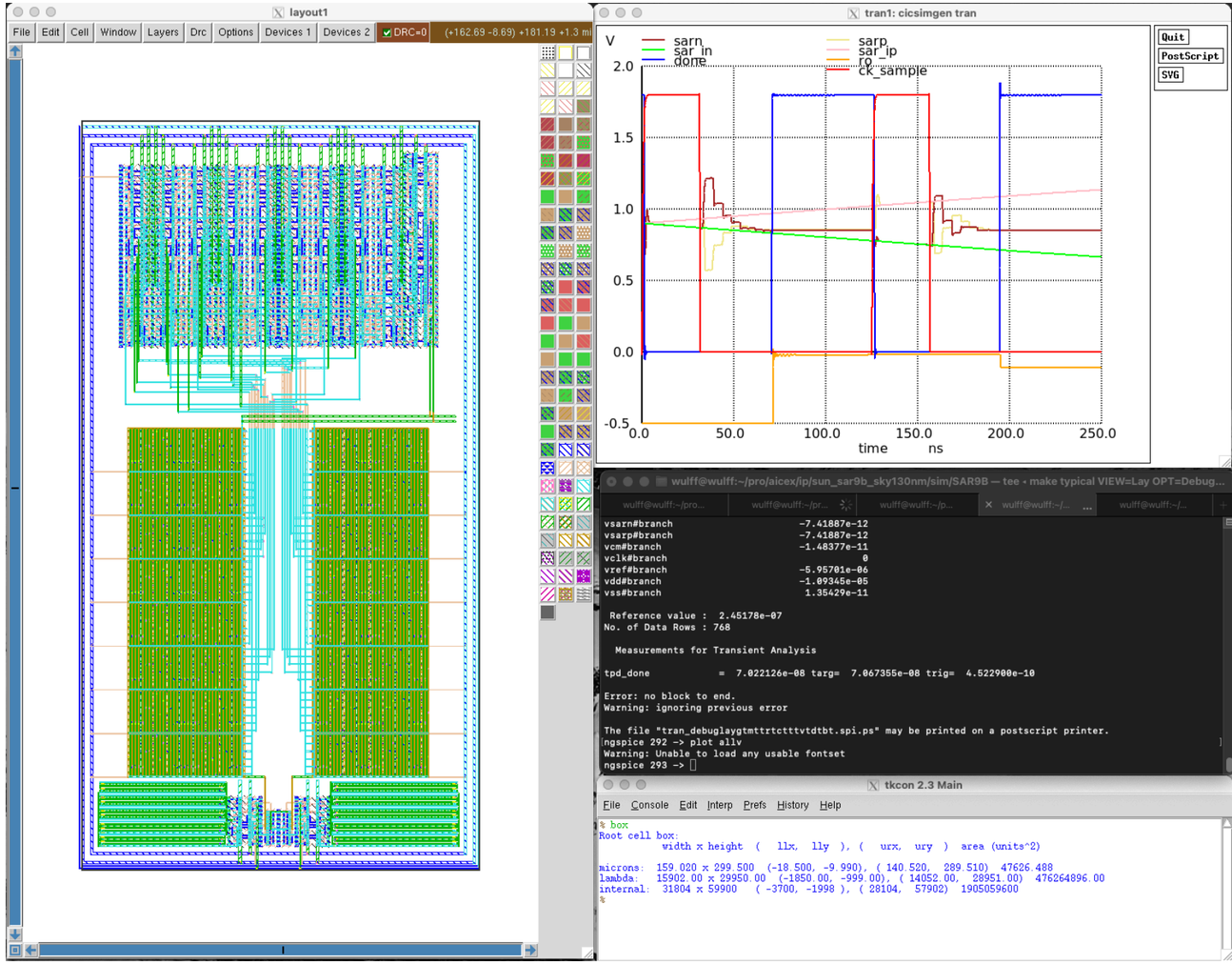
Fig. 1. Proposed noise-shaping SAR architecture. Blue blocks and paths are only active in calibration mode.



Since then

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

Finally, there is an open source port to skywater 130nm!
wulffern/sun*sar9b*sky130nm



Key learnings

Super simple transistor was a good choice for portability

```
{ "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-------------",
           "--------------------"
       ]
  ]
}
```

```
{ "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

wulffern/aicex

Thanks!

Most pictures by DALL-E