Low-Power CMOS ADC Design and Calibration Techniques

Yun Chiu

Professor, E. Jonsson Distinguished Professor
University of Texas @ Dallas

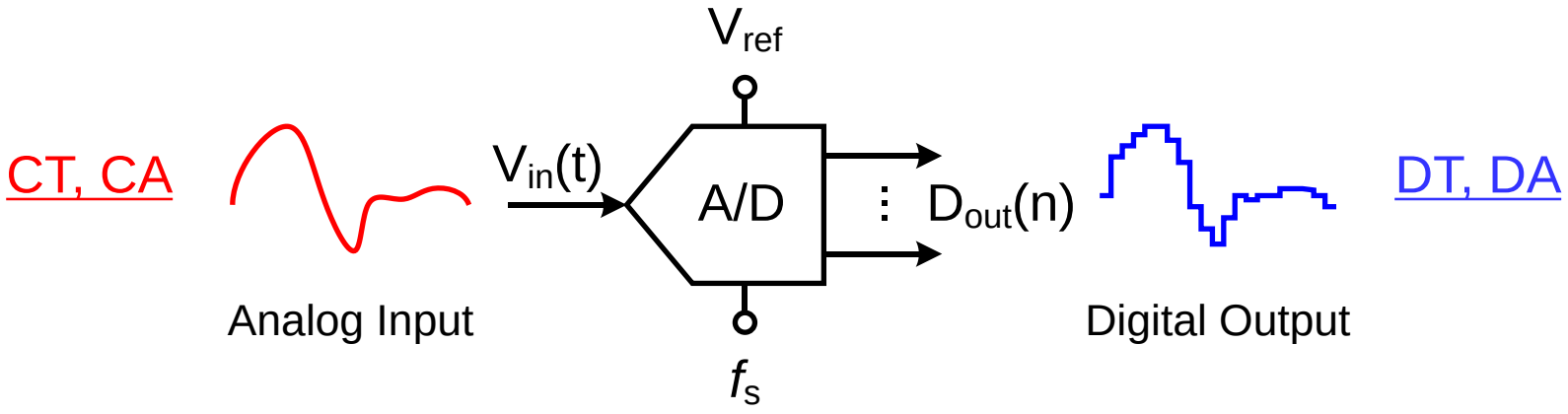
Course Outline

- Principles of Multistep A/D Conversion
- Architectural Redundancy
- Error Mechanisms and Digital-Domain Calibration
- Error-Parameter Identification
  - PRBS Test-Signal Injection (sub-ADC, sub-DAC, input)
  - Two-ADC Equalization (ref.-ADC, split-ADC, ODC)
- Energy Efficiency and Trend
- Summary

Course Outline

- Principles of Multistep A/D Conversion
- Architectural Redundancy
- Error Mechanisms and Digital-Domain Calibration
- Error-Parameter Identification
  - PRBS Test-Signal Injection (sub-ADC, sub-DAC, input)
  - Two-ADC Equalization (ref.-ADC, split-ADC, ODC)
- Energy Efficiency and Trend
- Summary

What is A/D Conversion?

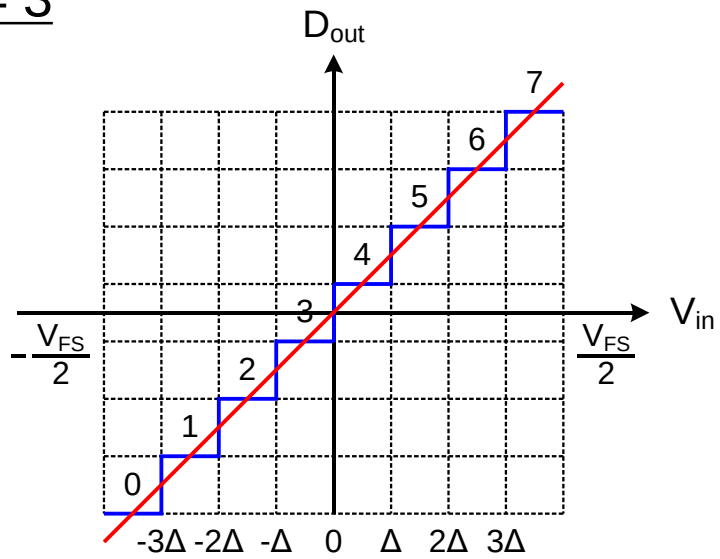


$$LSB = \Delta = \frac{V_{FS}}{2^N}, \quad D_{out}(n) = \left\lfloor \frac{V_{in}(t)}{\Delta} \right\rfloor_{t=nT_s} = \left\lfloor 2^N \cdot \frac{V_{in}(nT_s)}{V_{FS}} \right\rfloor$$

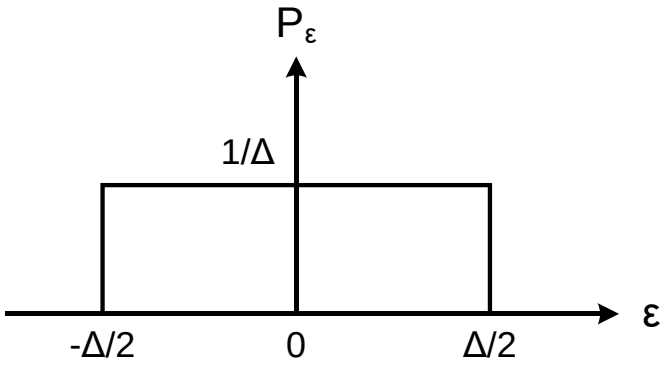
- Quantization = division + normalization + truncation
- V_{FS} is the Full-Scale range of ADC determined by V_{ref}.

Quantization Error (or Noise)

N = 3

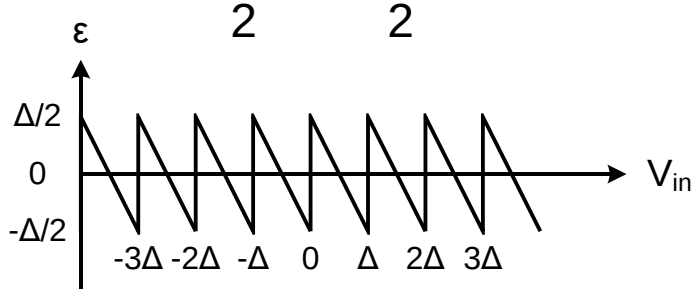


- N is large
- $V_{in} >> \Delta$, V_{in} is active
- ε is <u>uniformly</u> distributed



$$-\frac{\Delta}{2} \le \varepsilon \le \frac{\Delta}{2}$$

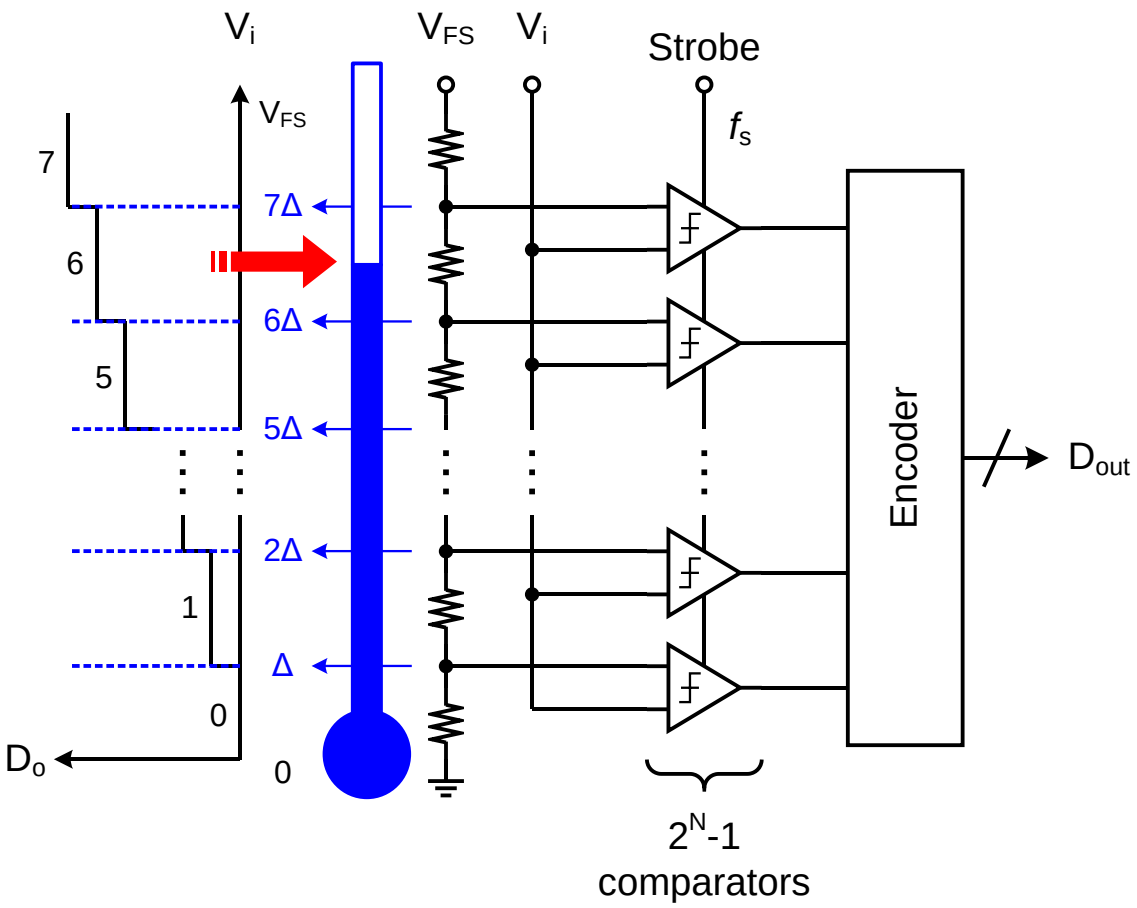
$$\sigma_\varepsilon^2 = \int_{-\Delta/2}^{\Delta/2} \varepsilon^2 \cdot \frac{1}{\Delta} \cdot d\varepsilon = \frac{\Delta^2}{12}$$



"Random" quantization error is usually regarded as noise.

Ref. [1]

Flash ADC – Exhaustive Search



- Massive parallelism
- Very fast
- Reference ladder consists of 2^N equal size resistors
- Input is compared to 2^N-1 reference voltages
- Throughput = f_s
- Complexity = 2^N

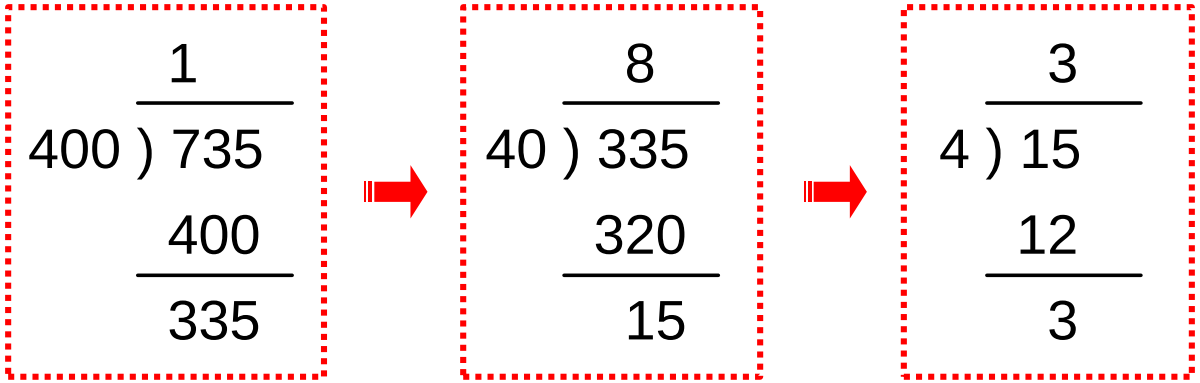
- Flash ADC is rarely used for beyond 6-8 bits due to complexity.

Long Division (Decimal Case)

$$735 \div 4 = 183 \text{ r } 3$$

Dividend ÷ Divisor = Quotient r Remainder

```
  183
4 ) 735
  4        (1×4=4)
  33       (7−4=3)
  32       (8×4=32)
  15       (33−32=1)
  12       (3×4=12)
  3        (15−12=3)
```



Step 1: 1st bit

Step 2: 2nd bit

Step 3: 3rd bit

Quantization (Binary Case)

$N = 3,\ FS = 1000,\ \Delta = 1000/8 = 125,\ V_{in} = 735$

$$D_o = \left\lfloor \frac{V_{in}}{\Delta} \right\rfloor$$

$$\underset{\text{Vin}}{735} \div \underset{\text{LSB}}{125} = \underbrace{[1,0,1]}_{\text{Do}} \ r \ \underset{\text{QN}}{110}$$

$$\begin{array}{r} 1 \\ \hline 500 \,)\, 735 \\ 500 \\ \hline 235 \end{array}$$

Step 1:

1st bit

$$\begin{array}{r} 0 \\ \hline 250 \,)\, 235 \\ 0 \\ \hline 235 \end{array}$$

Step 2:

2nd bit

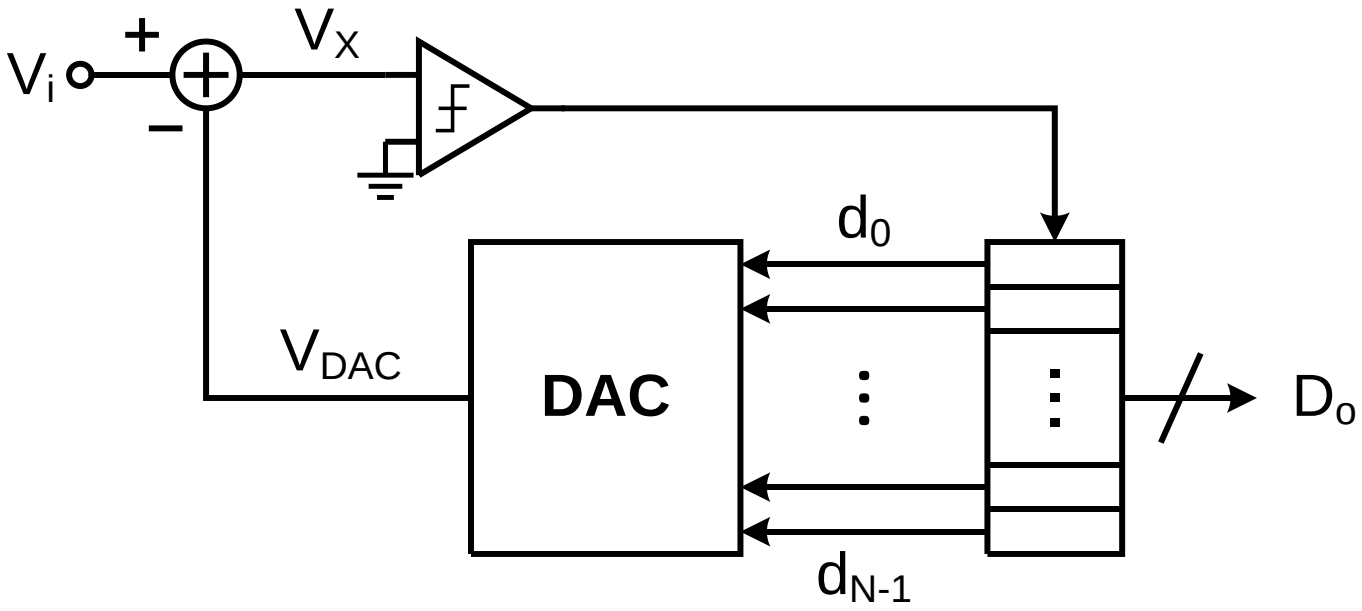
$$\begin{array}{r} 1 \\ \hline 125 \,)\, 235 \\ 125 \\ \hline 110 \end{array}$$

Step 3:

3rd bit

- The procedure is also known as "**binary search**".

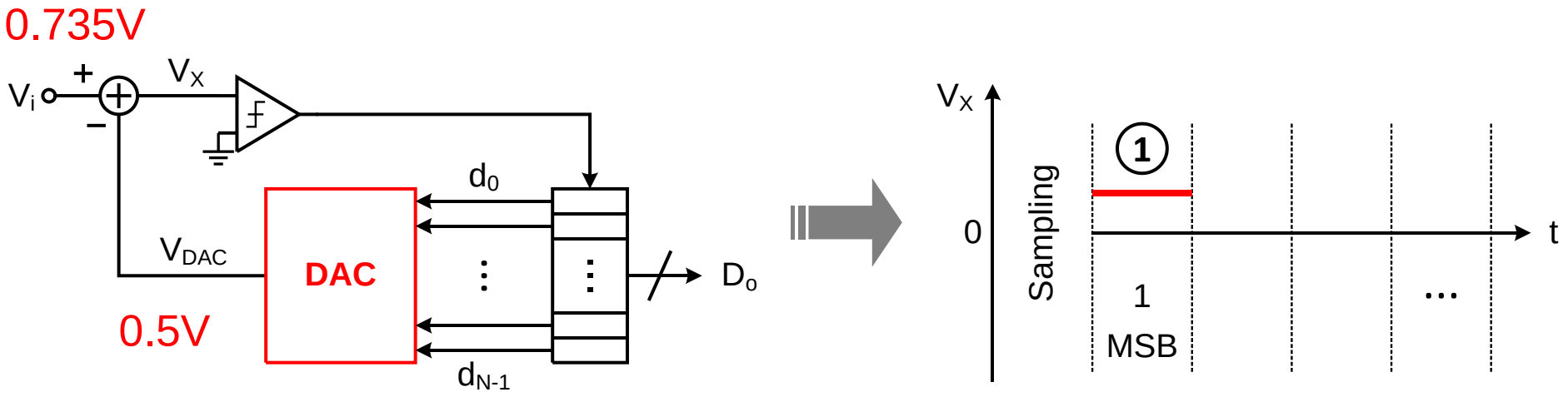
Successive-Approximation (SAR) ADC



SAR = 1 comparator + 1 DAC + digital logic

Binary Search – MSB Cycle

N = 3, FS = 1 V, Δ = 0.125 V, V_{in} = 0.735 V



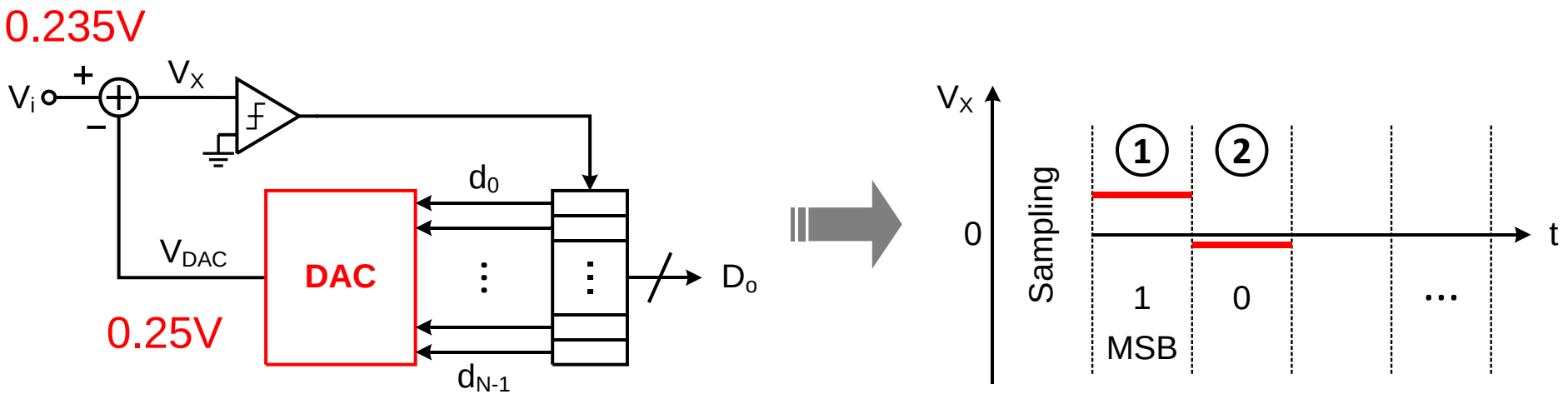
① $V_X = V_i - 0.5V$;

② if $V_X > 0$, MSB = 1, keep current V_X → V_X;

otherwise, MSB = 0, restore V_X → $V_X + 0.5V$;

Binary Search – MSB-1 Cycle

N = 3, FS = 1 V, Δ = 0.125 V, V_{in} = 0.735 V



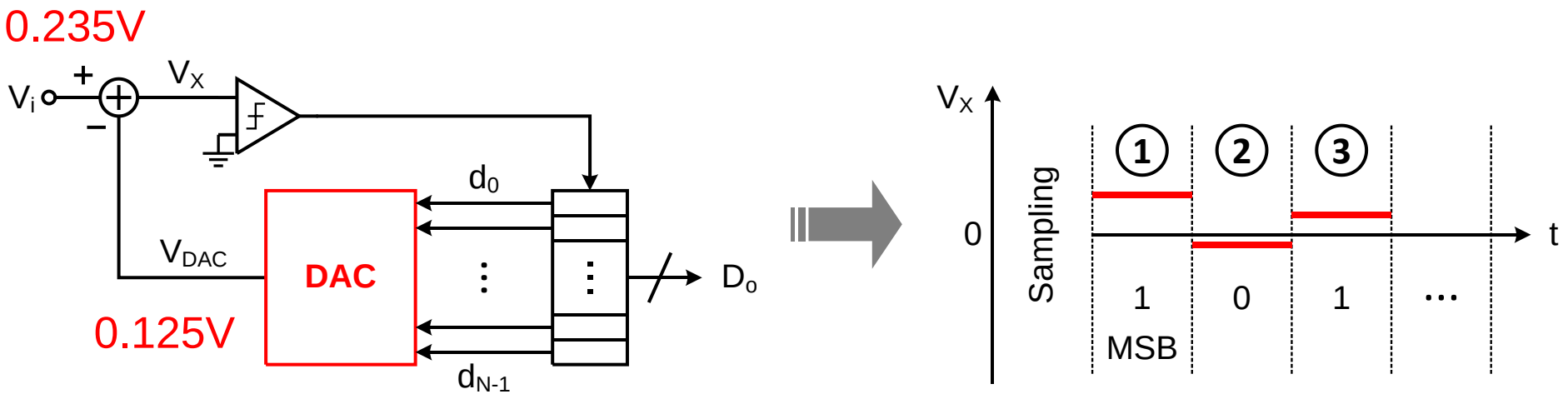
① $V_X = V_X - 0.25V$;

② if $V_X > 0$, MSB-1 = 1, keep current $V_X \rightarrow V_X$;

otherwise, MSB-1 = 0, restore $V_X \rightarrow V_X + 0.25V$;

Binary Search – MSB-2 Cycle

N = 3, FS = 1 V, Δ = 0.125 V, V_{in} = 0.735 V



① $V_X = V_X - 0.125V$;

② if $V_X > 0$, MSB-2 = 1, keep current $V_X \rightarrow V_X$;

otherwise, MSB-2 = 0, restore $V_X \rightarrow V_X + 0.125V$;

Quantization (Binary) Modified…

$N = 3,\ FS = 1000,\ \Delta = 1000/8 = 125,\ V_{in} = 735$

$$D_o = \left\lfloor \frac{V_{in}}{\Delta} \right\rfloor \qquad 735 \ (\text{Vin}) \ \div \ 125 \ (\text{LSB}) \ = \ \underbrace{[1,0,1]}_{\text{Do}} \ \text{r} \ 110 \ (\text{QN})$$

$$\begin{array}{r} 1 \\ \hline 500 \,)\, 735 \\ 500 \\ \hline 235\times 2 \end{array} \qquad \begin{array}{r} 0 \\ \hline 500 \,)\, 470 \\ 0 \\ \hline 470\times 2 \end{array} \qquad \begin{array}{r} 1 \\ \hline 500 \,)\, 940 \\ 500 \\ \hline 440 \end{array}$$

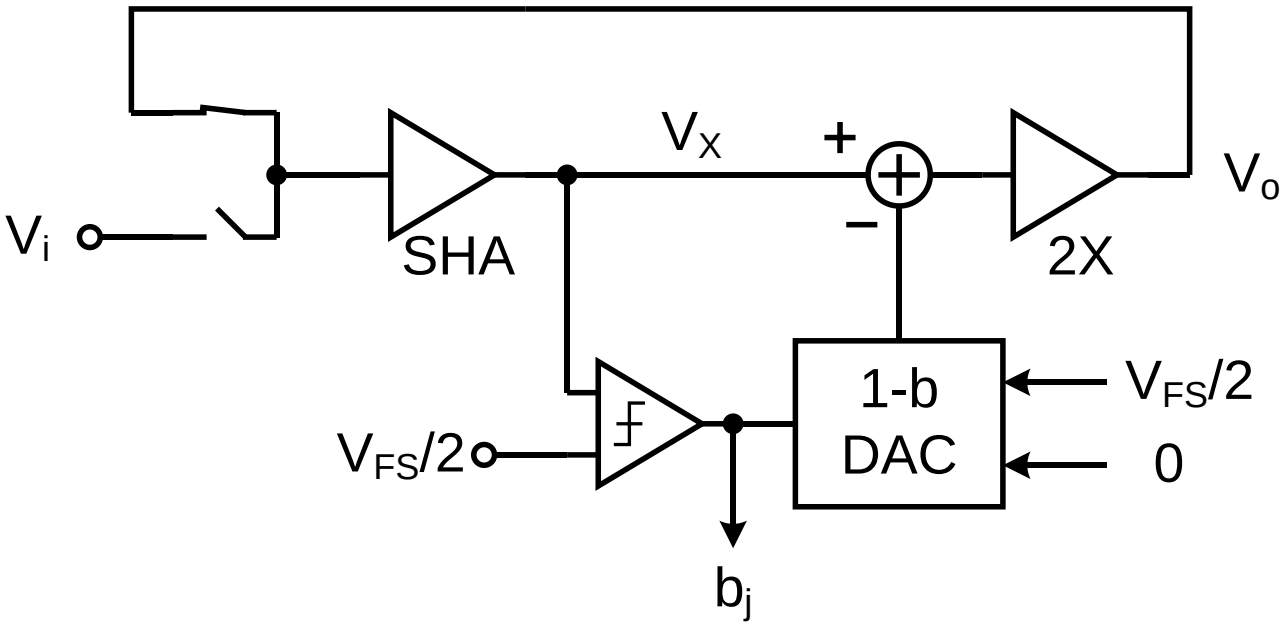
Step 1: 1st bit

Step 2: 2nd bit

Step 3: 3rd bit

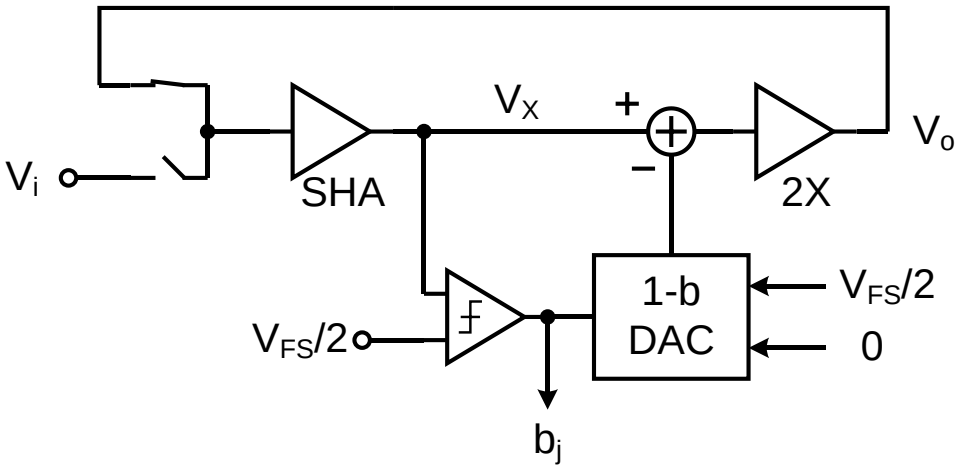
- Always use the same divisor but amplify the residue.

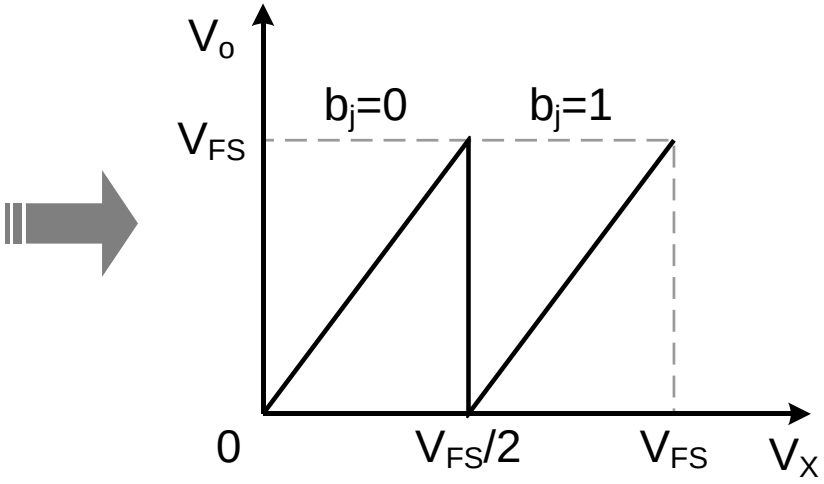
Algorithmic (Cyclic) ADC



- Fixed comparison threshold ($V_{FS}/2$) + 1-b DAC + Residue Amplifier
- Modified "Binary Search"

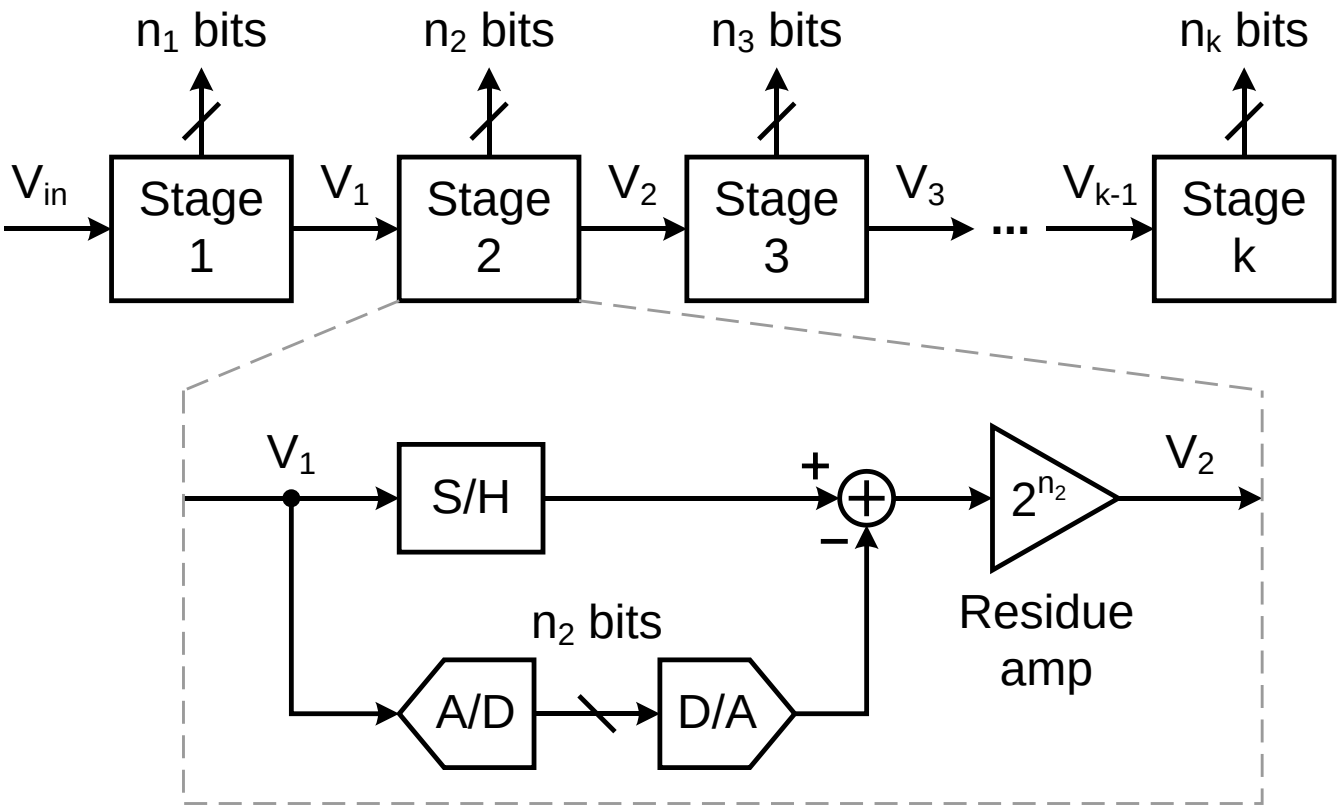
Bit Cycles





- Comparison → if $V_X < V_{FS}/2$, then $b_j = 0$; otherwise, $b_j = 1$
- Residue generation → $V_o = 2 \cdot (V_X - b_j \cdot V_{FS}/2)$

Pipelined ADC

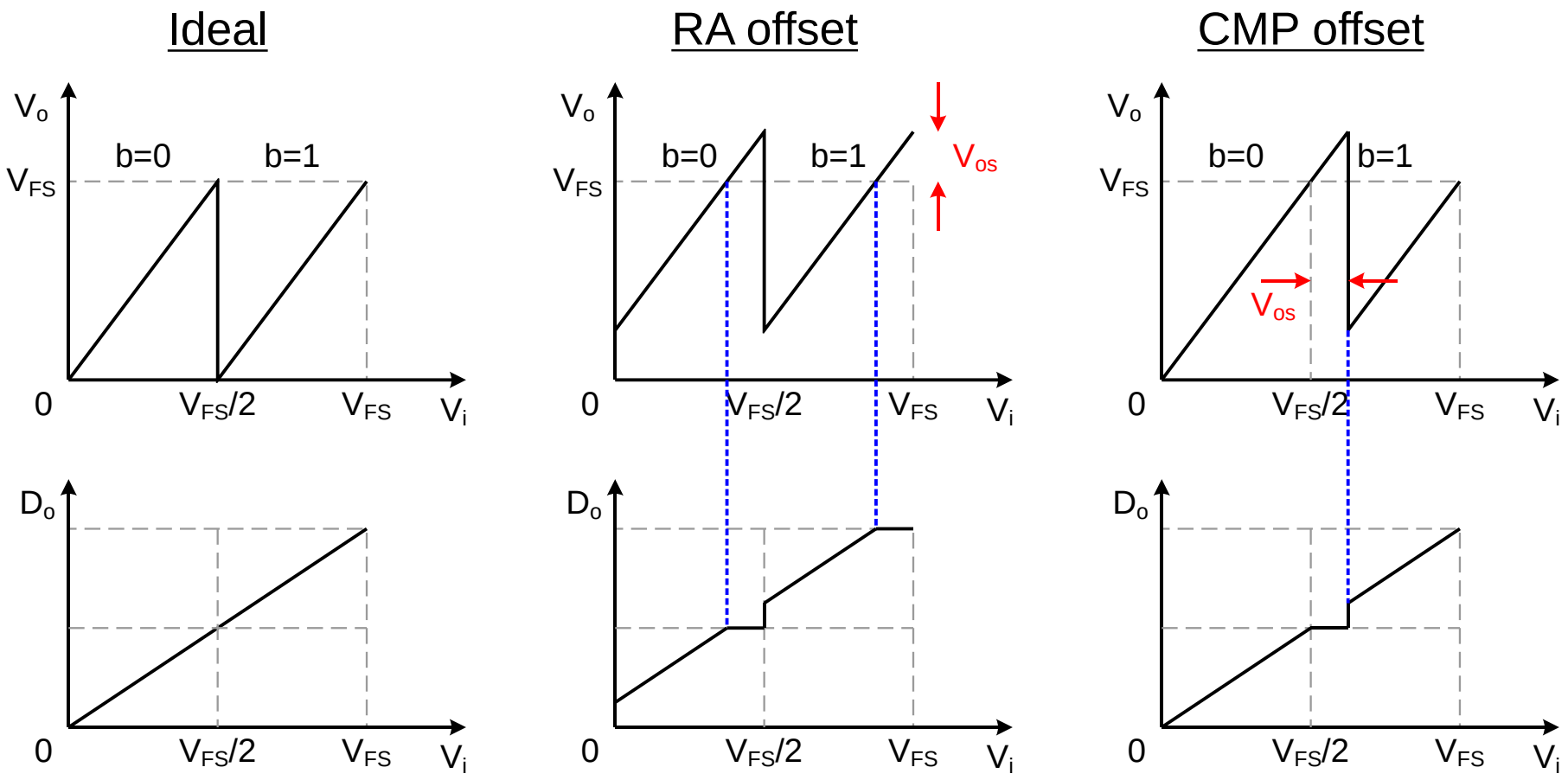


- Algorithmic ADC loop unrolled → pipeline enables high throughput

Presentation Outline

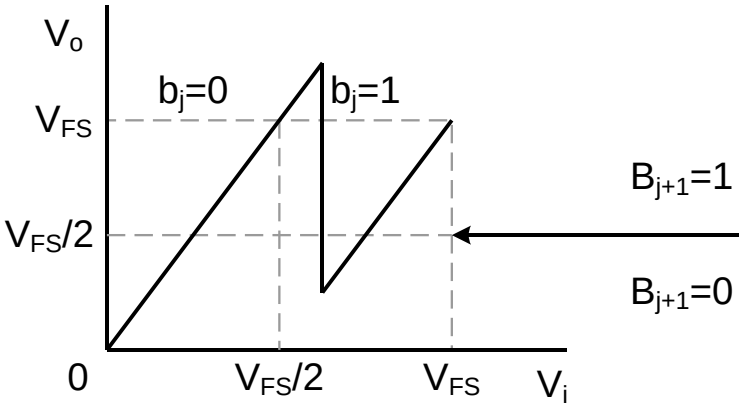
- Principles of Multistep A/D Conversion
- Architectural Redundancy
- Error Mechanisms and Digital-Domain Calibration
- Error-Parameter Identification
  - PRBS Test-Signal Injection (sub-ADC, sub-DAC, input)
  - Two-ADC Equalization (ref.-ADC, split-ADC, ODC)
- Energy Efficiency and Trend
- Summary

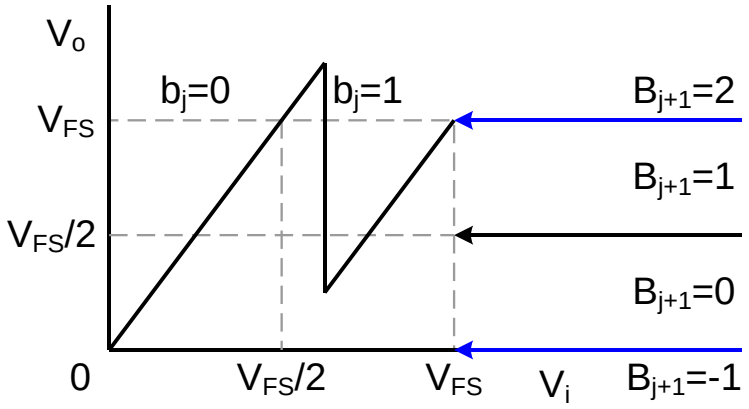
What happens with circuit offsets?

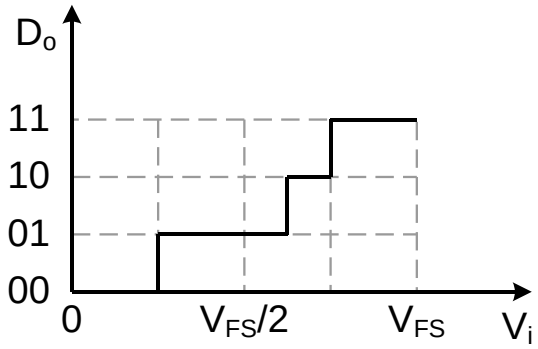


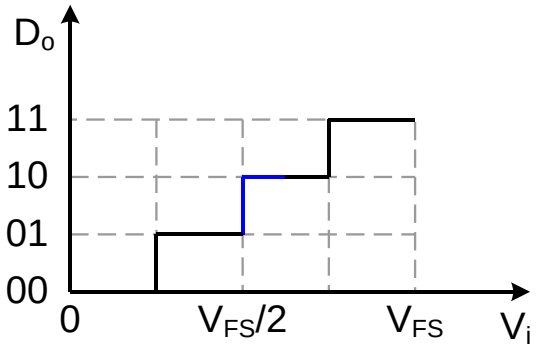
Nearly zero tolerance on circuit offset errors!!

Over-range & Under-range Comparators







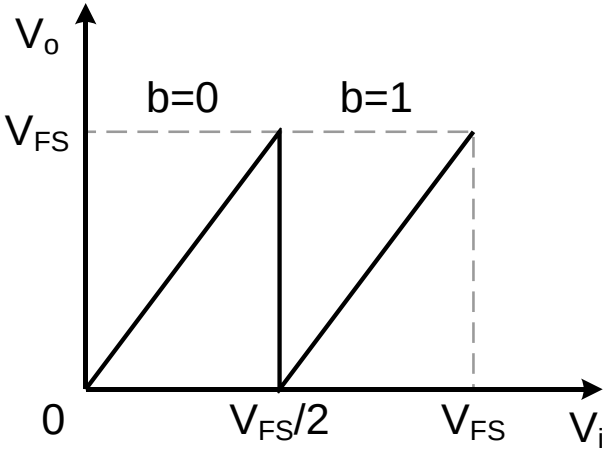


1 CMP

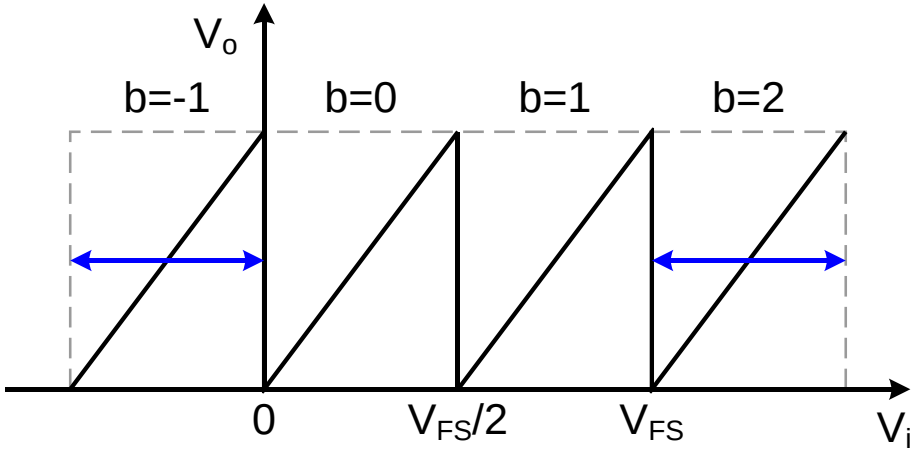
3 CMPs

Redundancy (a.k.a. DEC or RSD)

Original

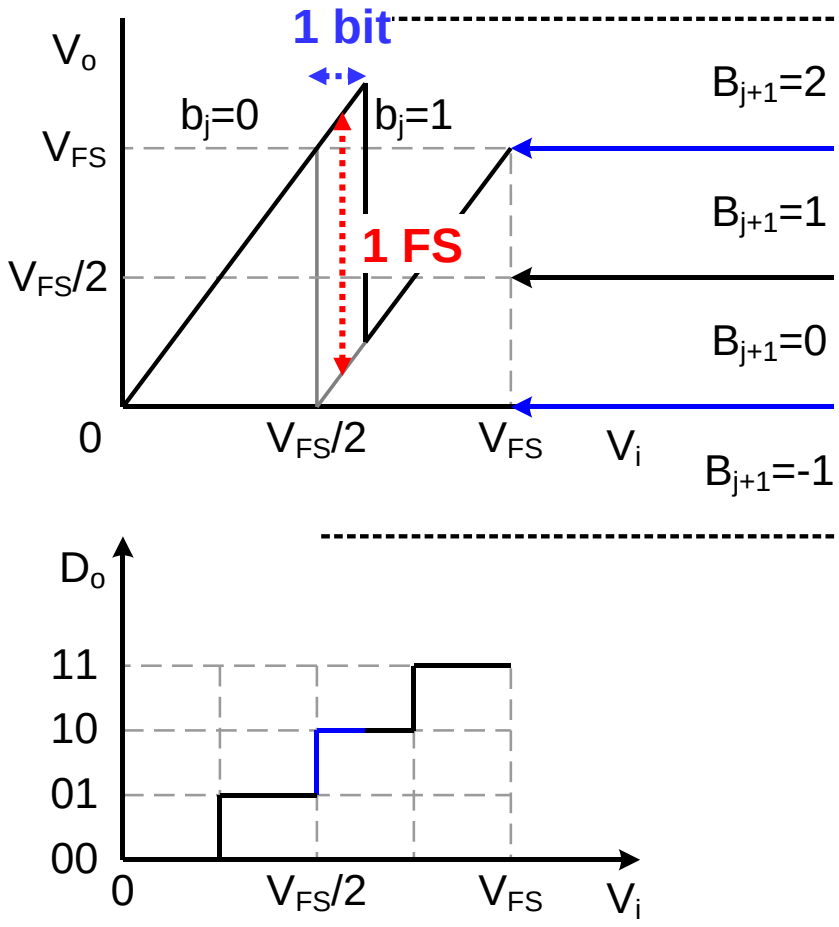


w/ Redundancy



- 4-level (2-bit) DAC required instead of 2-level (1-bit) DAC

Complementary Analog-Digital Information

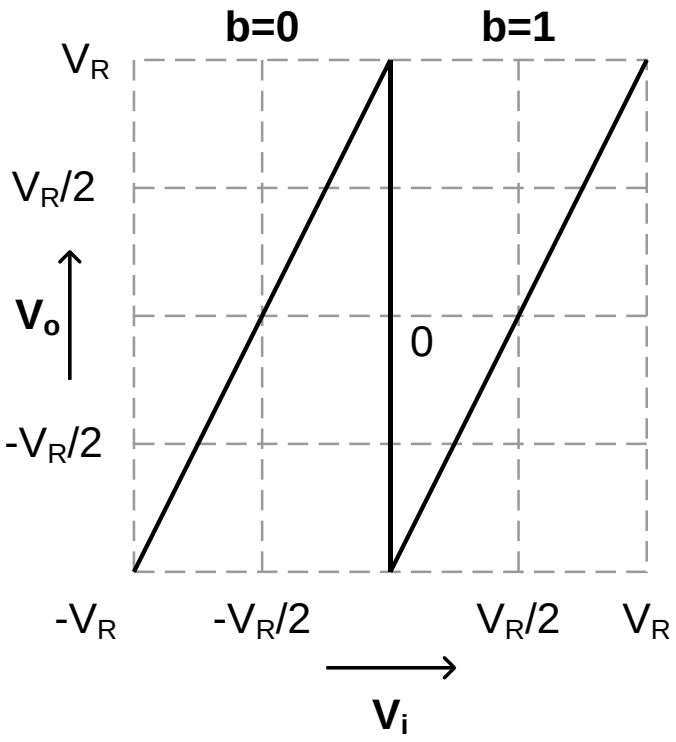


- Max tolerance of comparator offset is $\pm V_{FS}/4$ → simple comparators
- Key to understand redundancy:

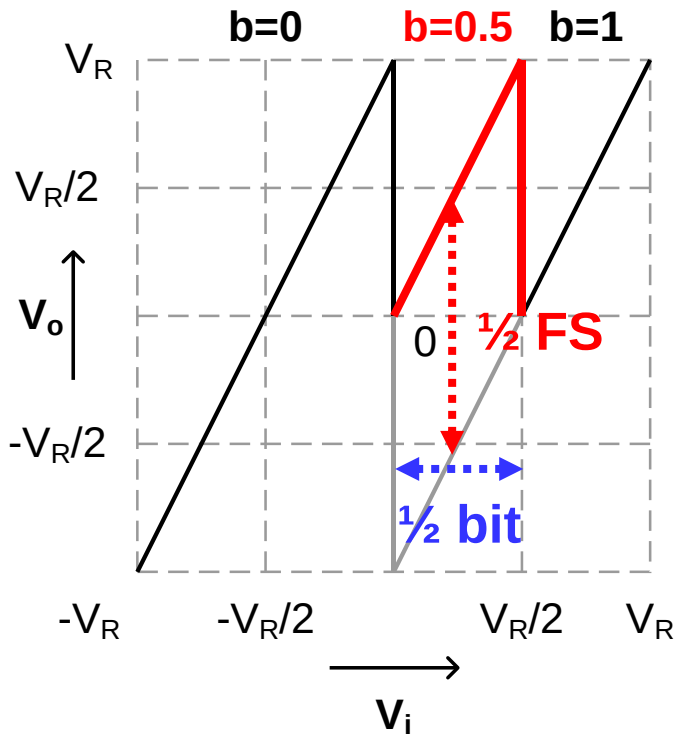
$$\Delta b_j \cdot V_{FS} + \Delta V_o = 0$$

$$V_i = b_j \cdot \frac{V_{FS}}{2} + \frac{V_o}{2}$$

From 1-bit to 1.5-bit Architecture

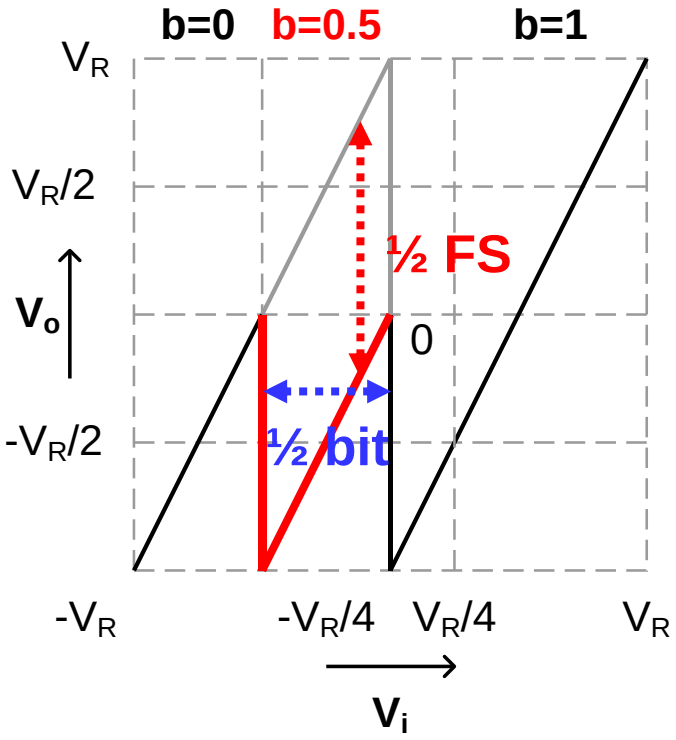


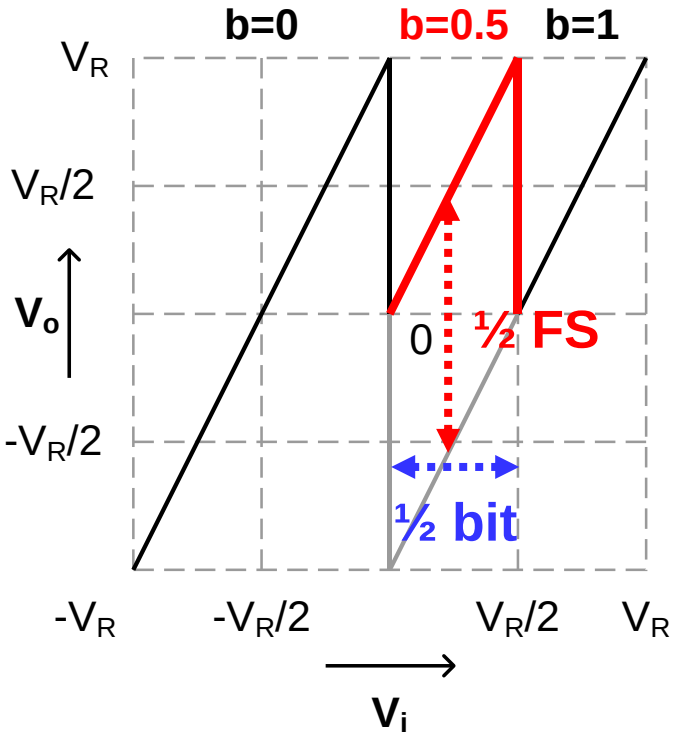
1-bit
No redundancy



$$\Delta b \cdot V_{FS} + \Delta V_o = 0$$

From 1-bit to 1.5-bit Architecture

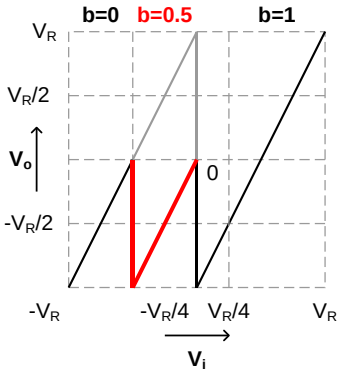


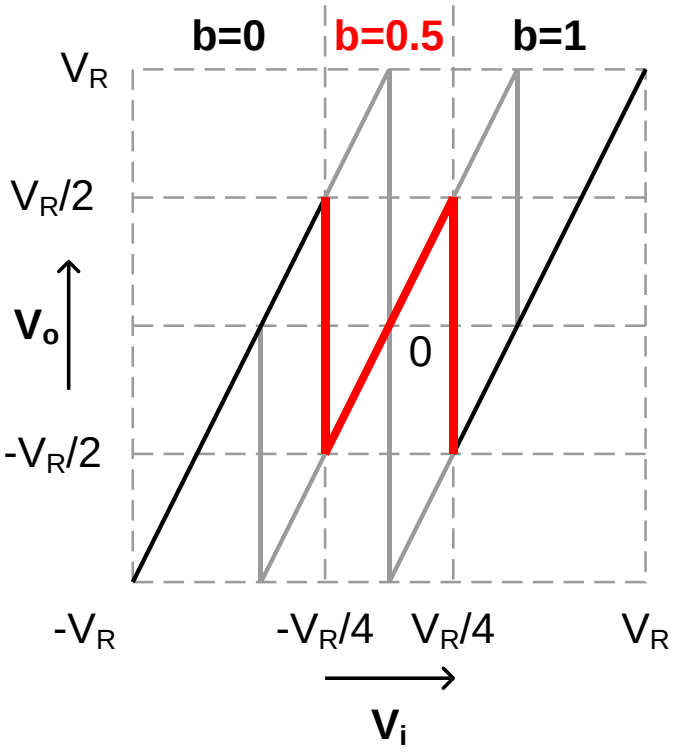


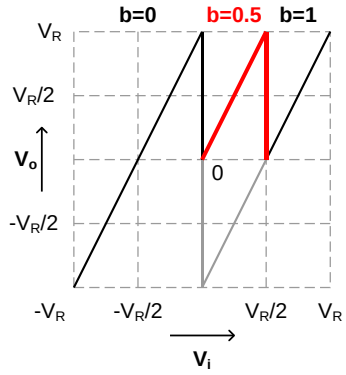
$$\Delta b \cdot V_{FS} + \Delta V_o = 0$$

$$\Delta b \cdot V_{FS} + \Delta V_o = 0$$

From 1-bit to 1.5-bit Architecture





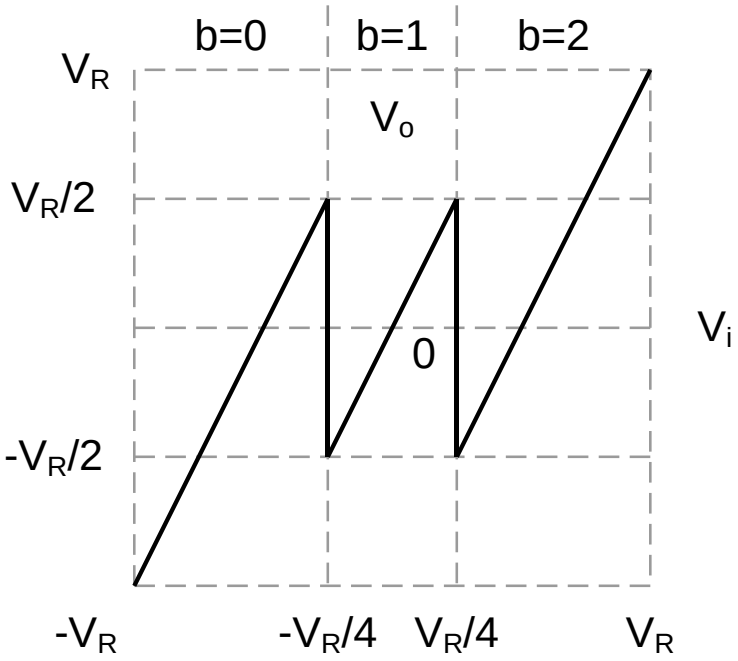


$$\Delta b \cdot V_{FS} + \Delta V_o = 0$$

$$\Delta b \cdot V_{FS} + \Delta V_o = 0$$

- Center the two thresholds → optimal symmetric offset tolerance

The 1.5-bit Architecture



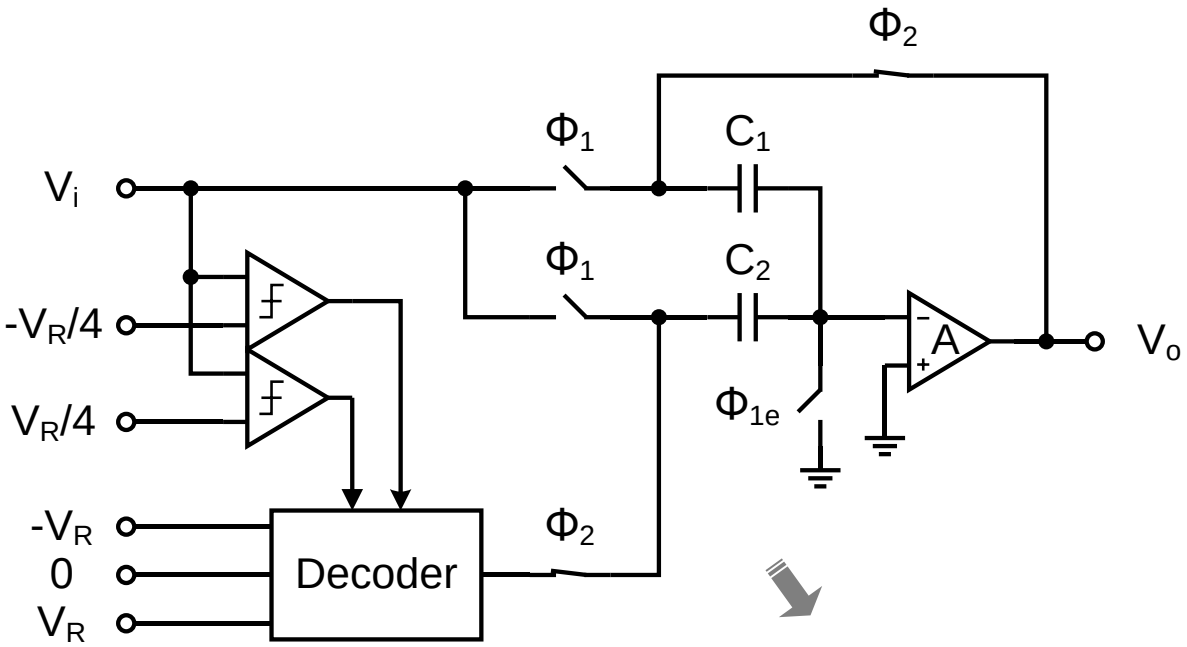
$$V_o = 2 \cdot V_i - (b-1) \cdot V_R$$

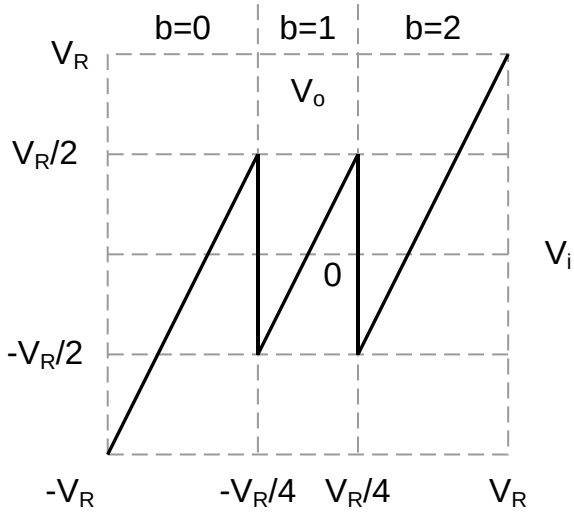
- 3 decision levels
  → ENOB = $\log_2 3 = 1.58$
- Max tolerance of comparator offset is $\pm V_R/4$
- An implementation of the Sweeny-Robertson-Tocher (SRT) division principle
- The conversion accuracy relies on the loop-gain error, i.e., the gain error and nonlinearity
- A 3-level DAC is required

Can the same technique be applied to SAR?

Ref. [2]

1.5-bit Multiplier DAC (MDAC)



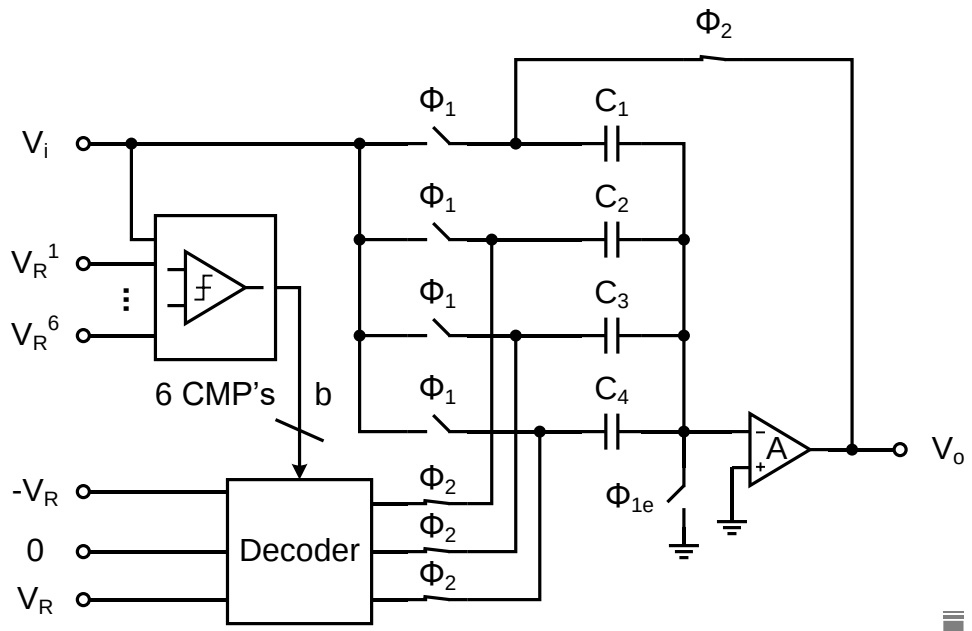


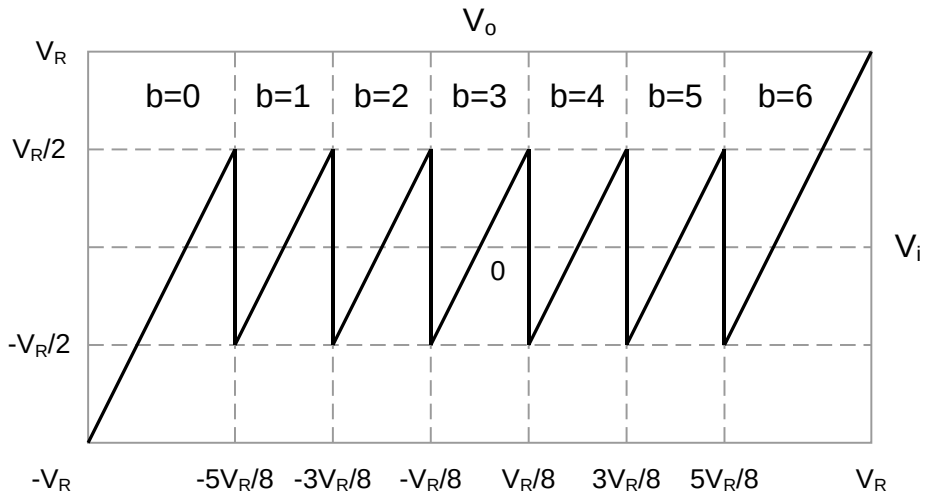
$$V_o = \frac{C_1 + C_2}{C_1} \cdot V_i - (b-1)\frac{C_2}{C_1} \cdot V_R$$

$$V_o = 2 \cdot V_i - (b-1) \cdot V_R$$

- 2X gain + 3-level DAC + subtraction all integrated
- Can be generalized to n.5-bit architectures

2.5-bit Multiplier DAC (MDAC)

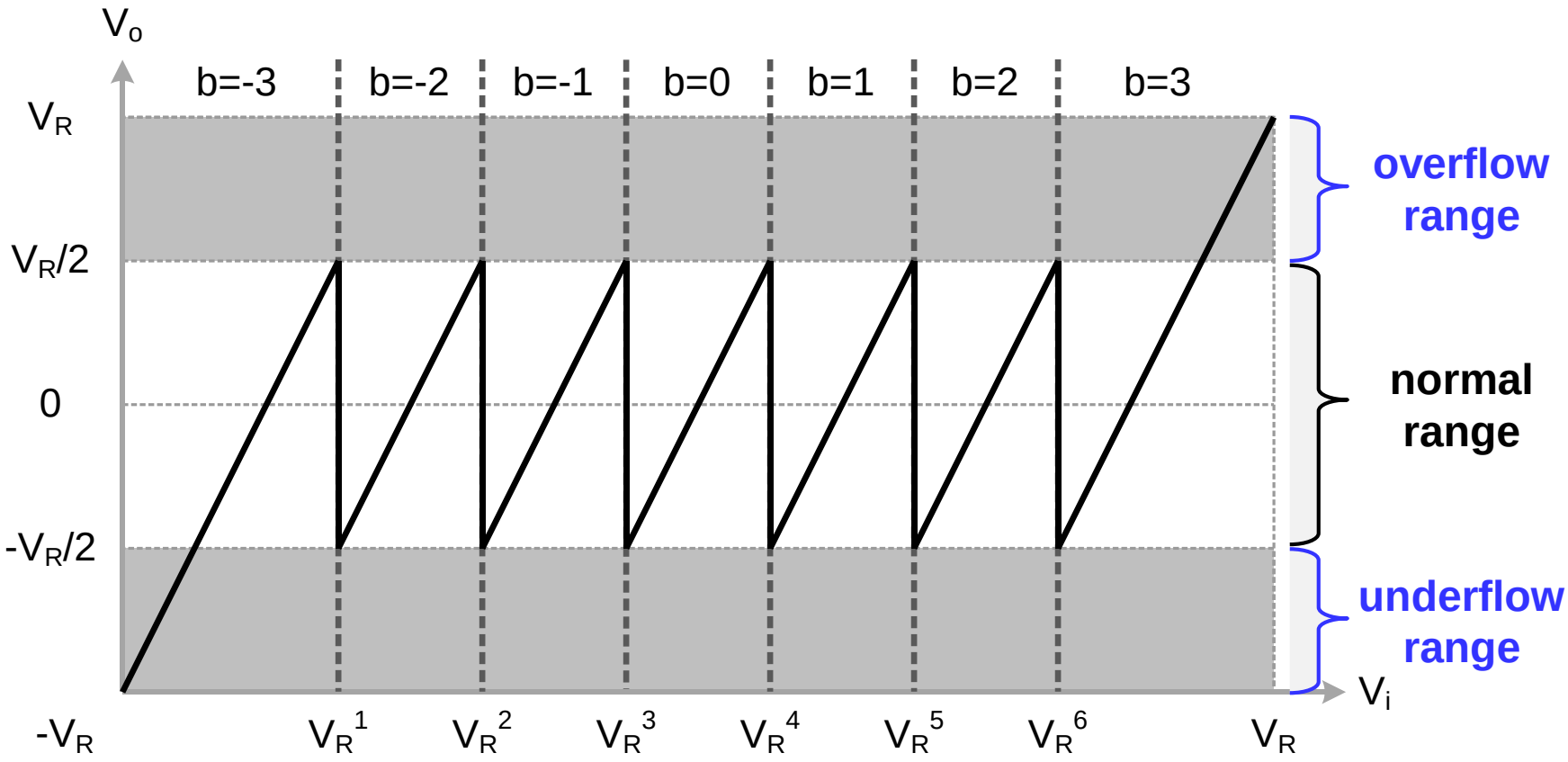




$$V_o = 4 \cdot V_i - (b-3) \cdot V_R$$

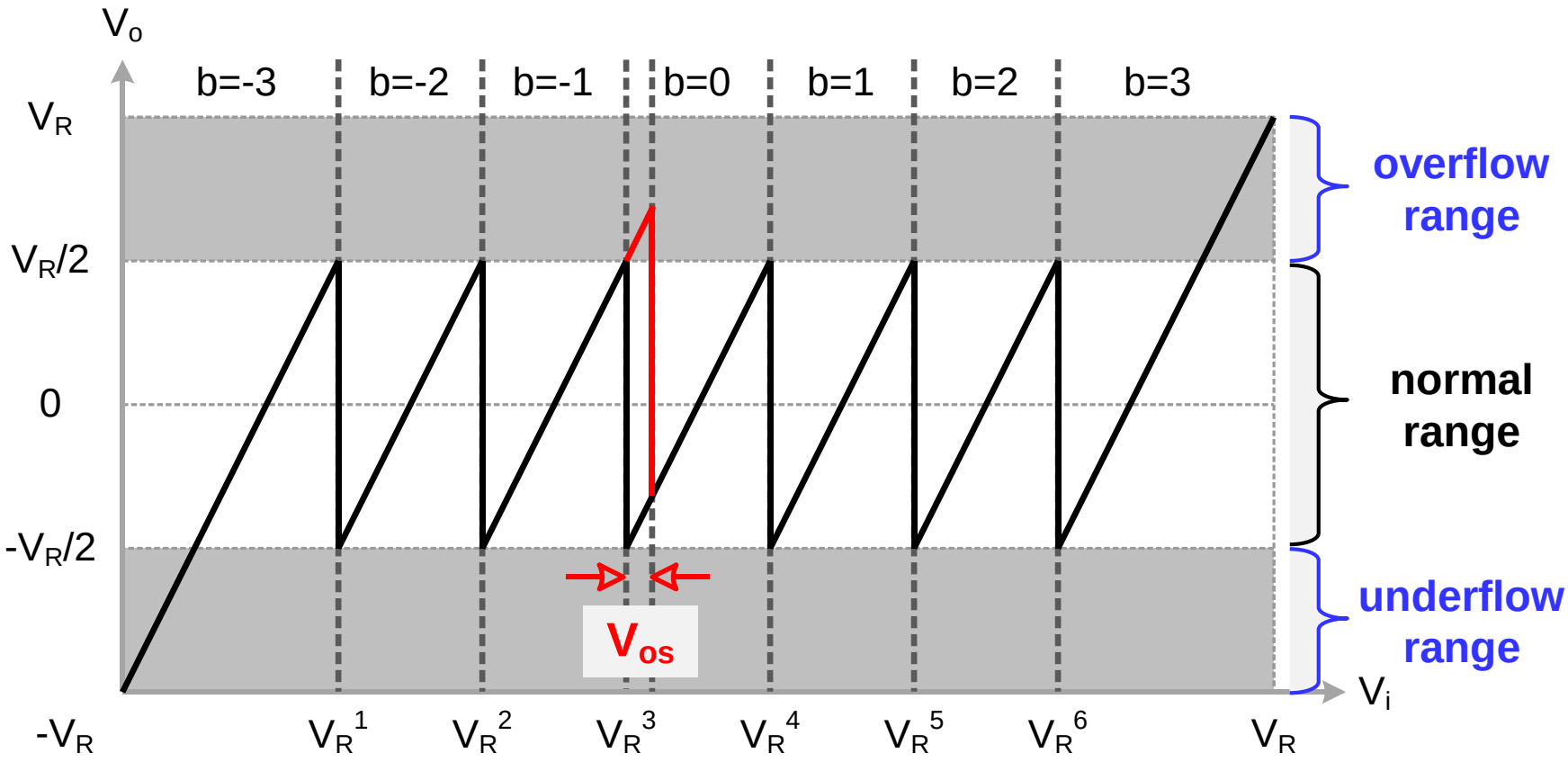
- 4X gain + 7-level DAC + subtraction all integrated

Residue Transfer Function (2.5b MDAC)

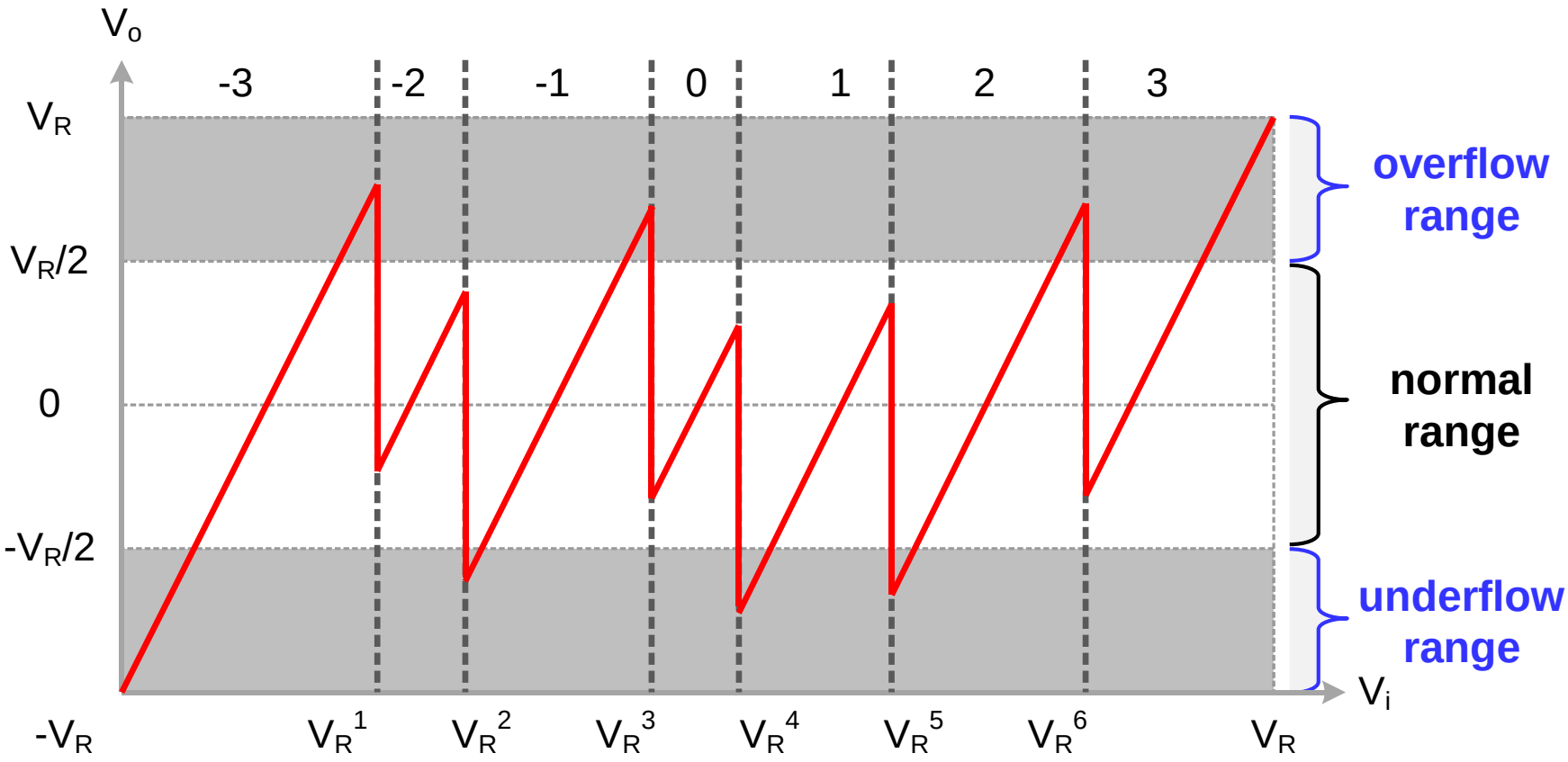


- Only <u>half</u> of the internal dynamic range is used under ideal condition!

With comparator offset

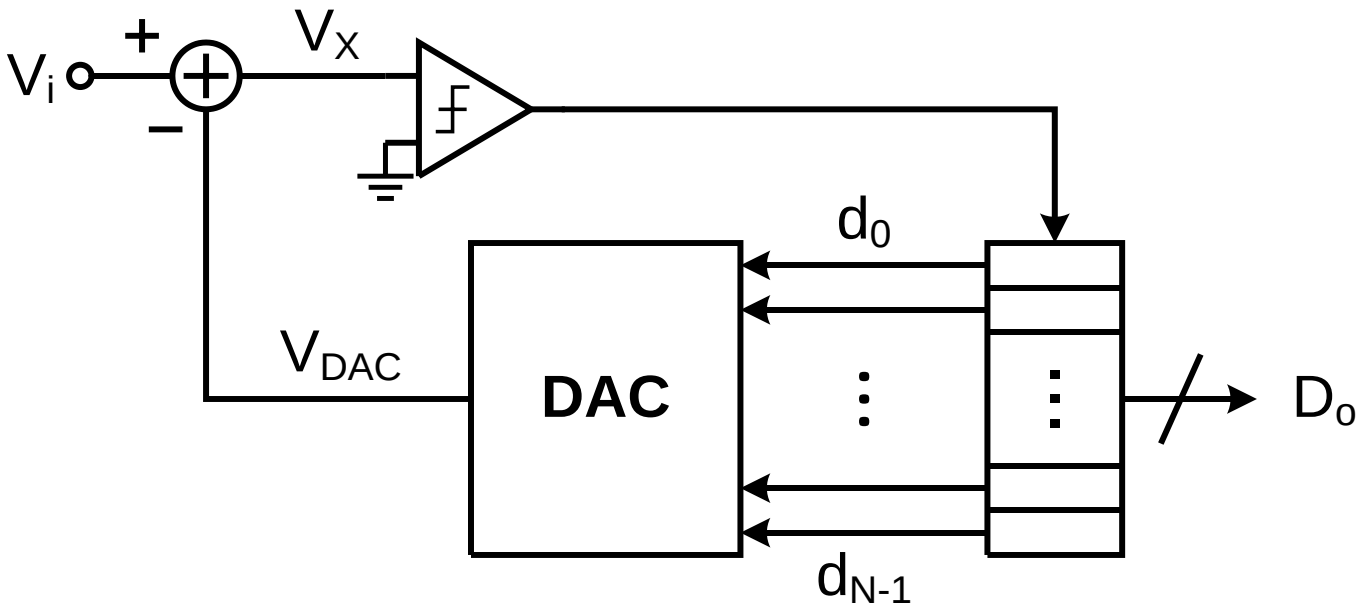


Internal Redundancy



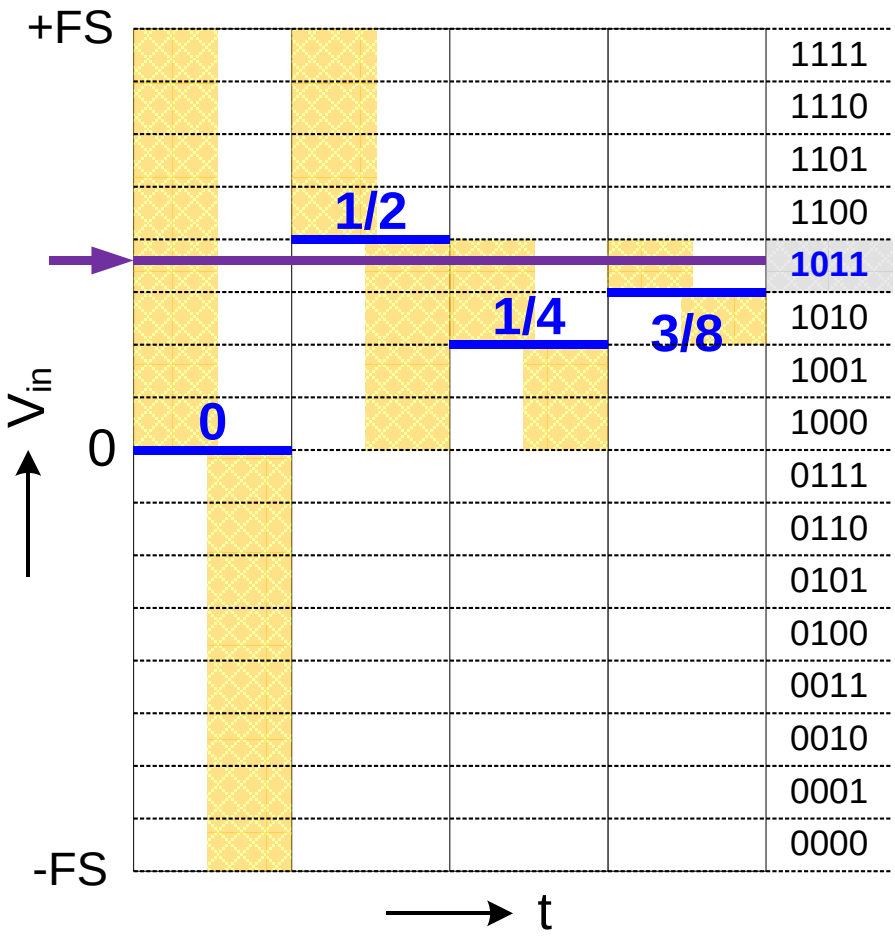
- Comparator and amplifier offsets tolerated by **internal redundancy**.

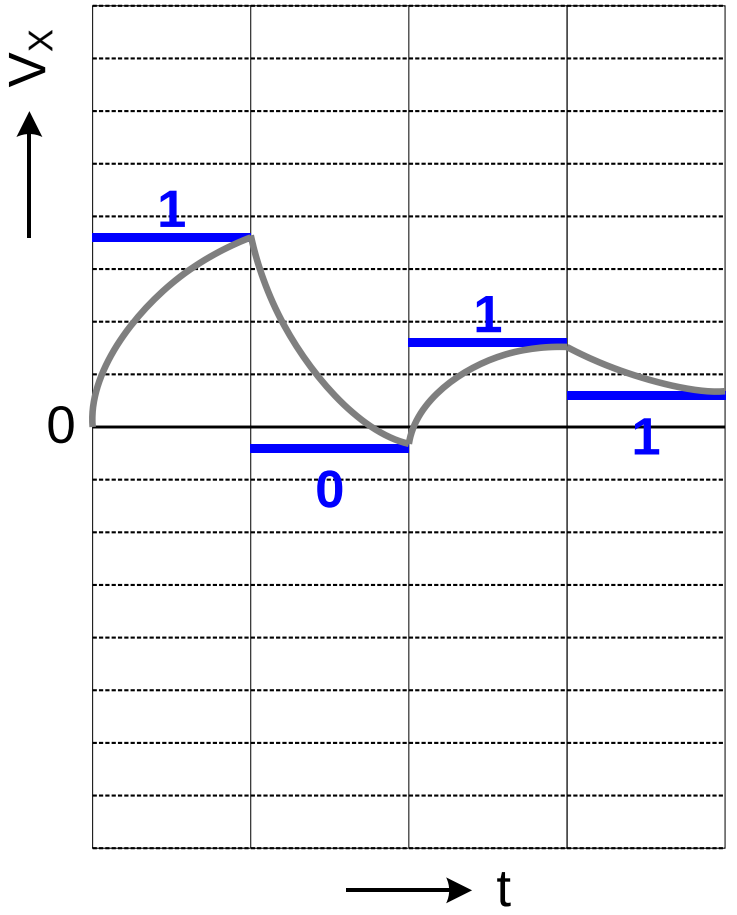
How does Redundancy work in SAR?



- Binary search is efficient, but displays zero error tolerance.

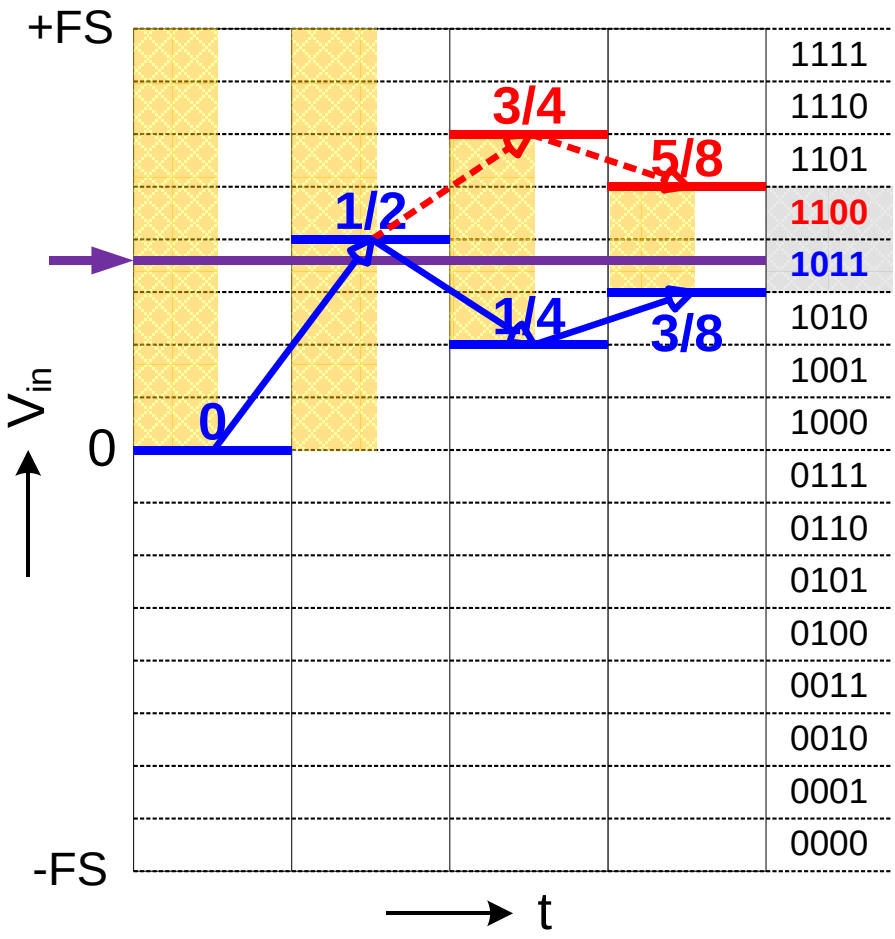
Binary Search Revisited

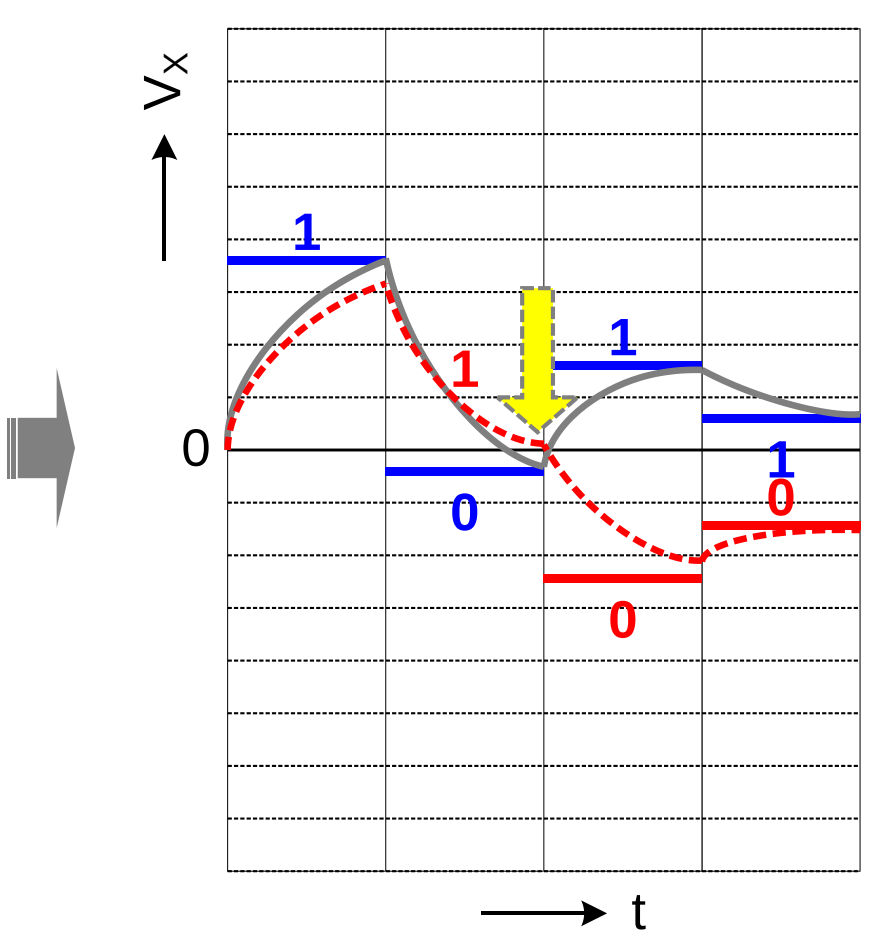




- When everything is ideal…

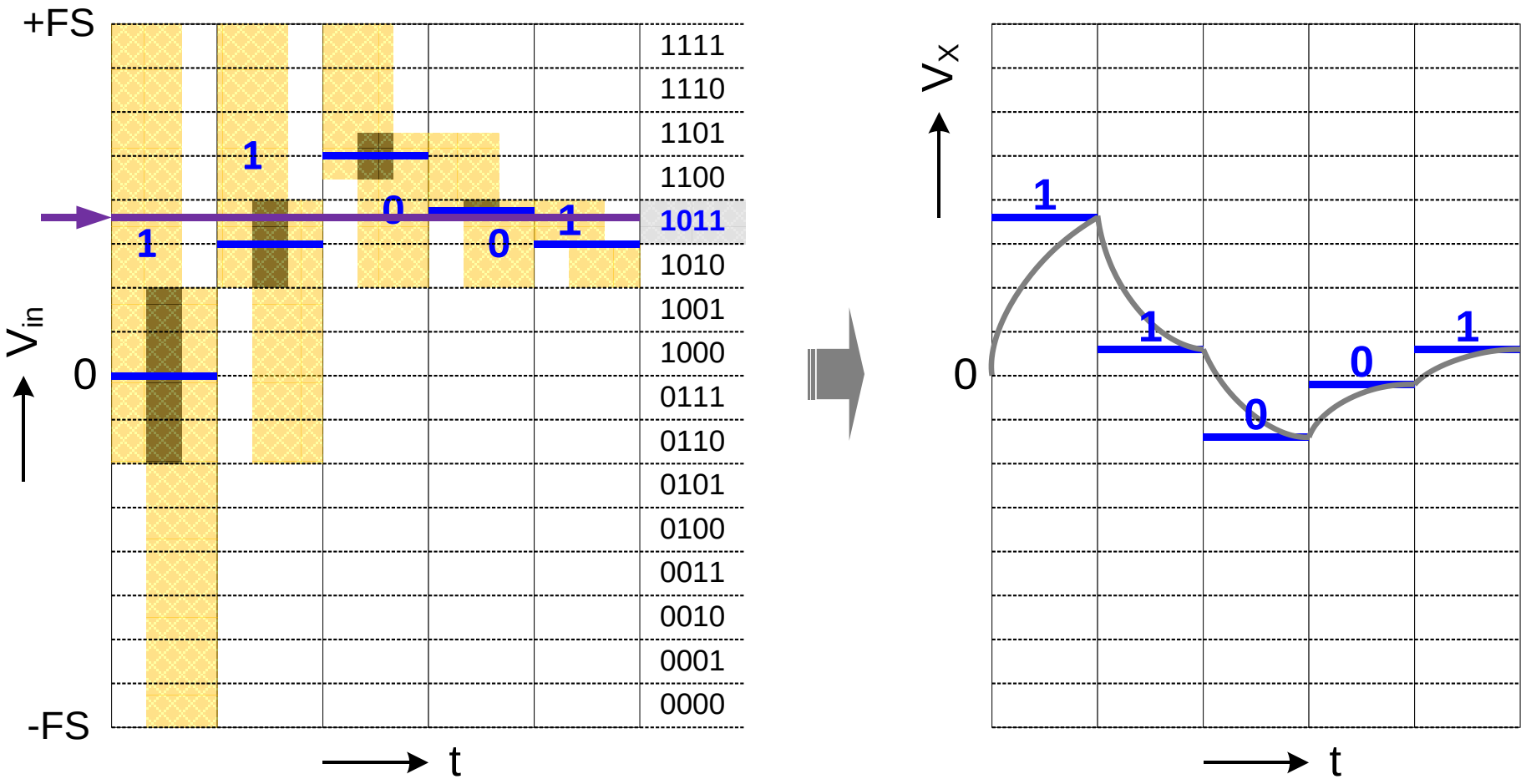
Binary Search w/ Dynamic Error





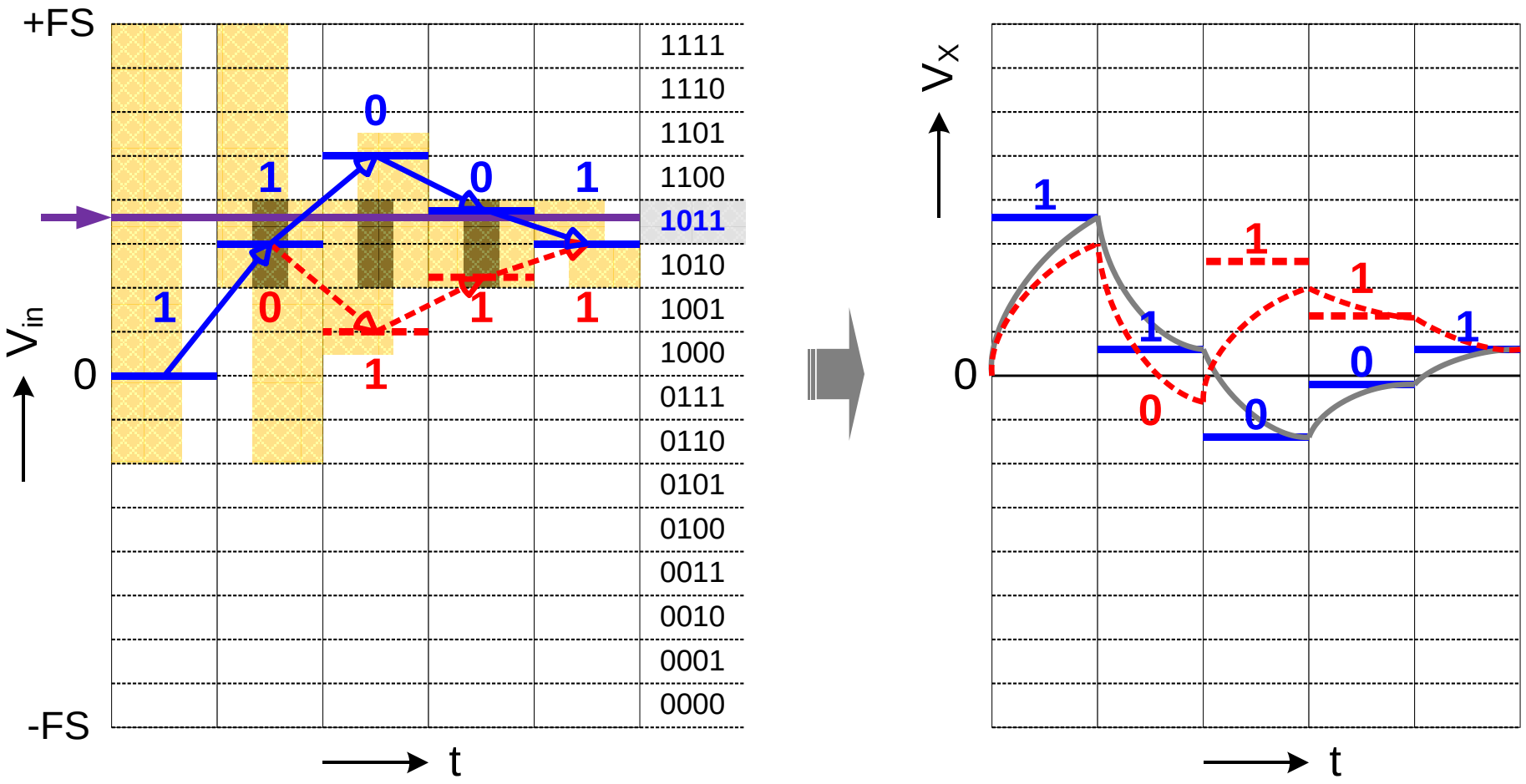
- Settling error, comparator hysteresis etc.

Overlapping Search Ranges



- Results indicate decision trajectory, no longer binary-coded.

Redundancy of Sub-binary Search



- Dynamic errors absorbed by redundancy.

SAR Redundancy

- Redundant conversion consumes more bit cycles, but can recover intermediate decision errors.
- Redundancy can be exploited to expedite conversion progress or to save power.
- **DAC levels (matching) still need to be accurate.**

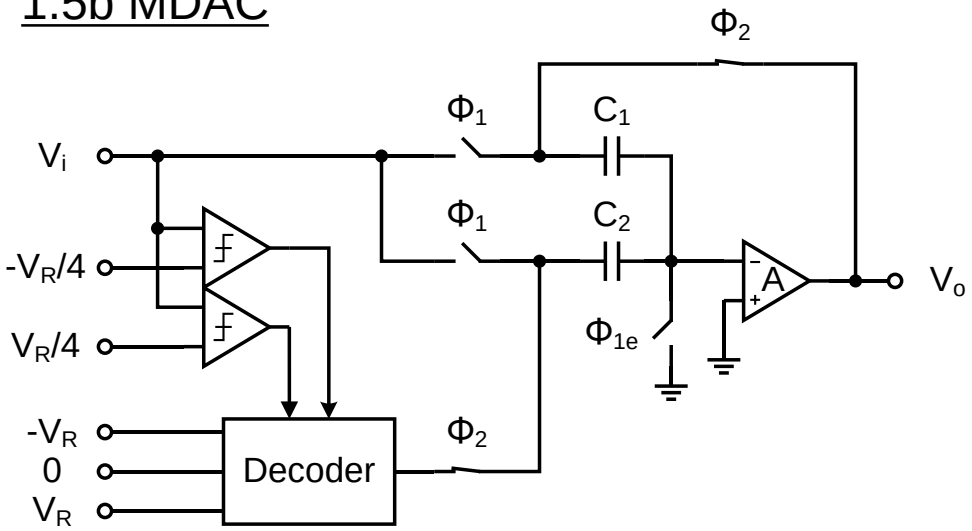
  (will come back to this later…)

Presentation Outline

- Principles of Multistep A/D Conversion
- Architectural Redundancy
- Error Mechanisms and Digital-Domain Calibration
- Error-Parameter Identification
  - PRBS Test-Signal Injection (sub-ADC, sub-DAC, input)
  - Two-ADC Equalization (ref.-ADC, split-ADC, ODC)
- Energy Efficiency and Trend
- Summary

Pipelined ADC Errors (I)

1.5b MDAC



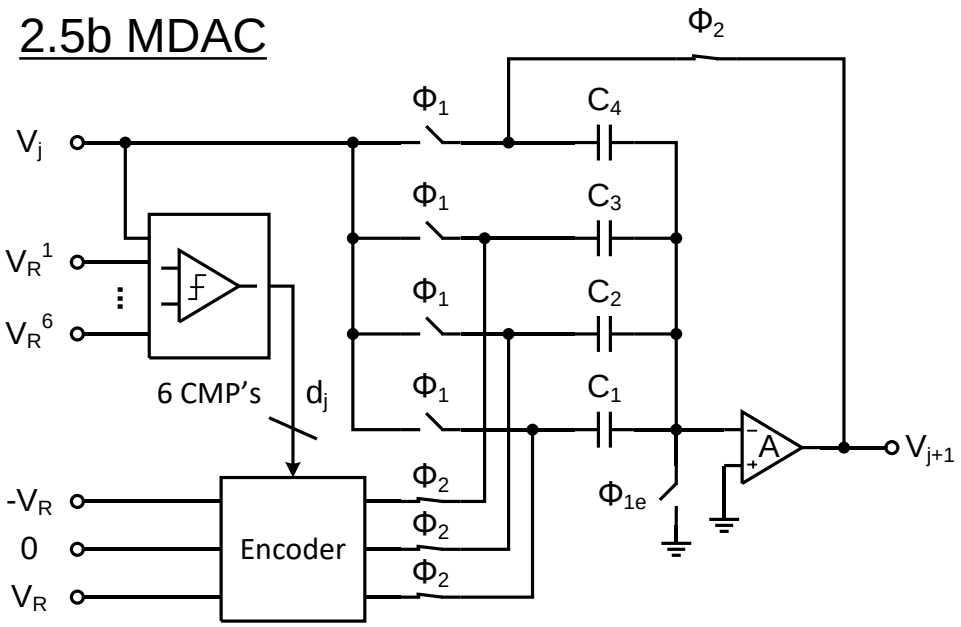
- Capacitor mismatch
- Op-amp finite-gain error and nonlinearity
- Charge injection and clock feed-through (S/H)
- Settling error

$$V_o = 2 \cdot V_i - d \cdot V_R$$

$$V_o(t=\infty) = \frac{C_1 + C_2}{C_1 + \frac{C_1 + C_2}{A(V_o)}} \cdot f_{S/H}(V_i) - \frac{C_2}{C_1 + \frac{C_1 + C_2}{A(V_o)}} \cdot d \cdot V_R$$

Pipelined ADC Errors (II)

2.5b MDAC



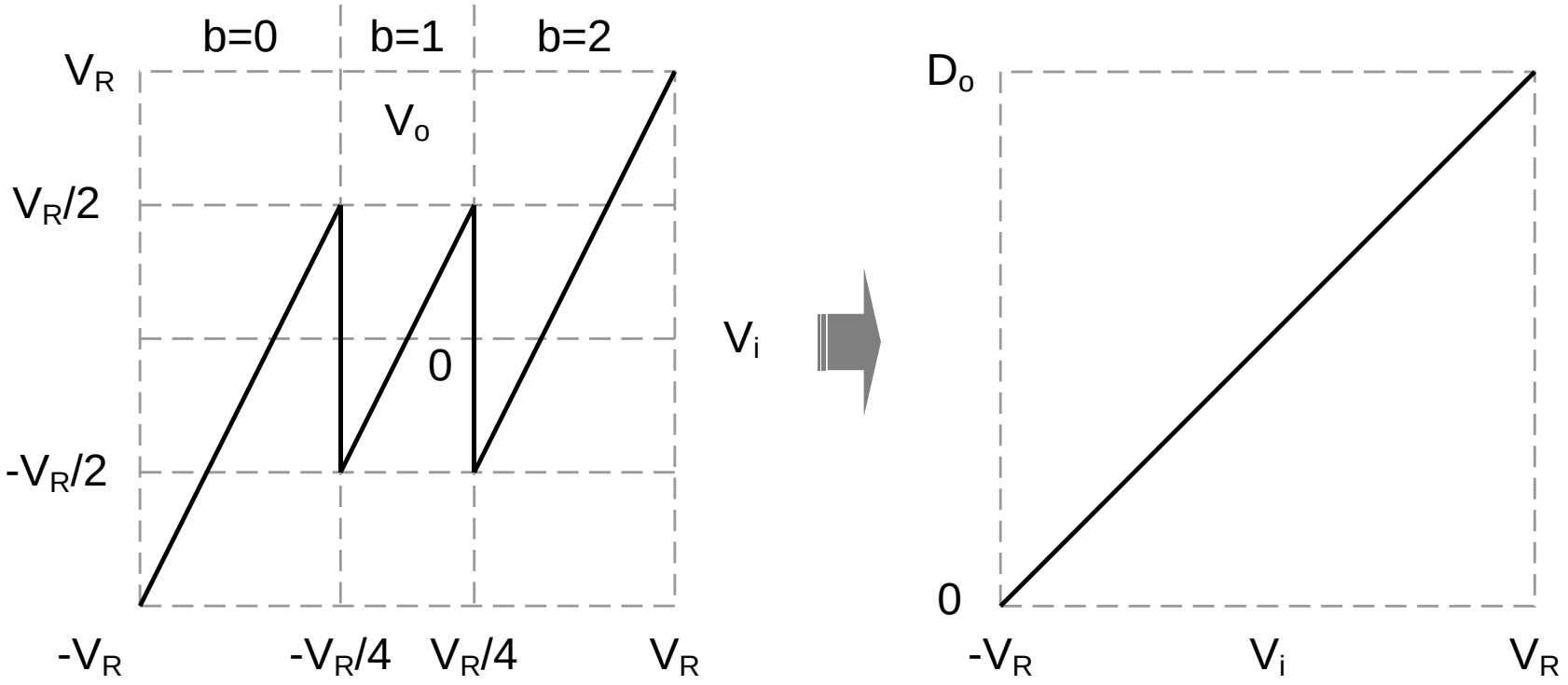
DAC bit-encoding scheme

d_j	-3	-2	-1	0	1	2	3
$d_{j,1}$	-1	-1	-1	-1	-1	0	1
$d_{j,2}$	-1	-1	-1	0	1	1	1
$d_{j,3}$	-1	0	1	1	1	1	1

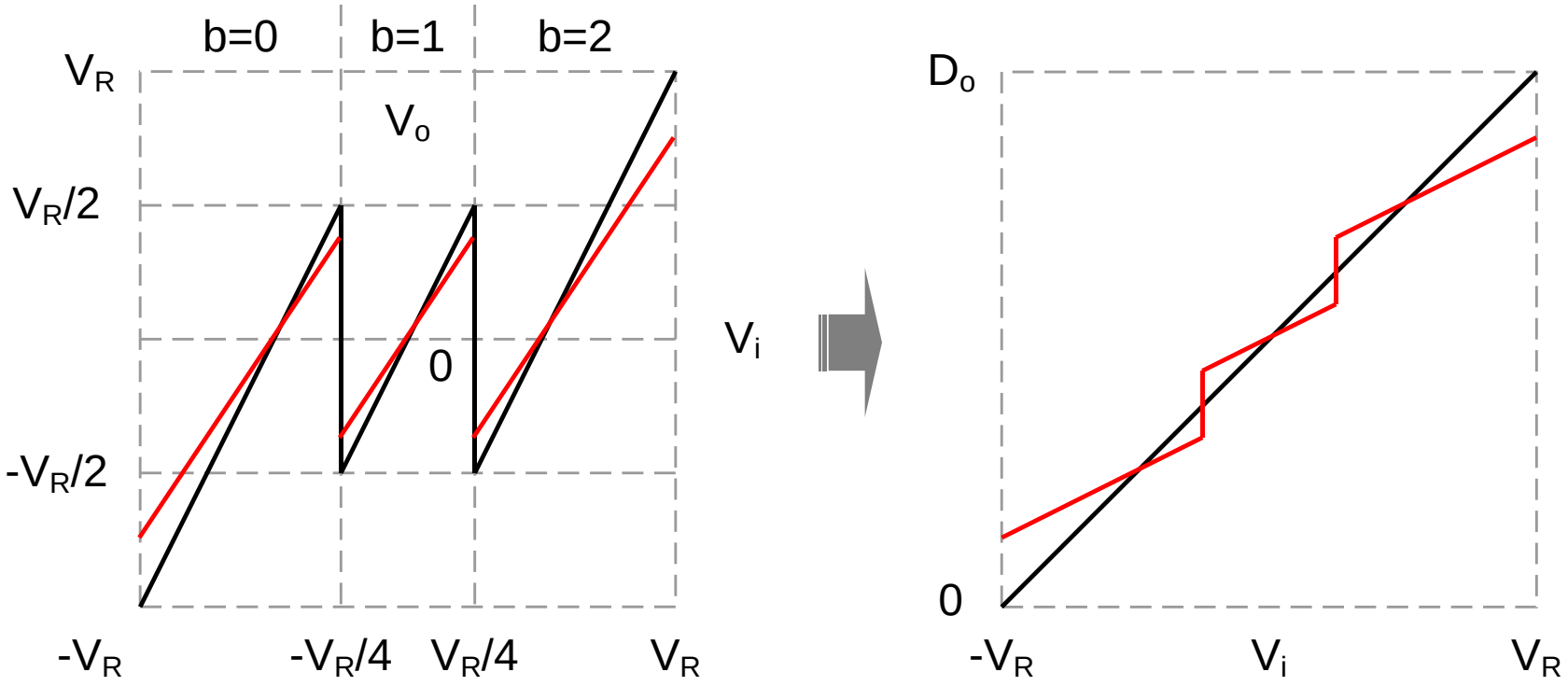
$$\mathbf{d_j} = d_{j,1} + d_{j,2} + d_{j,3}$$

$$V_j = \left(d_{j,1} \cdot \frac{C_1}{\sum C} + d_{j,2} \cdot \frac{C_2}{\sum C} + d_{j,3} \cdot \frac{C_3}{\sum C} \right) \cdot V_r + V_{j+1} \cdot \frac{C_4 + \sum C / A(V_{j+1})}{\sum C}$$

RA Gain Error and Nonlinearity

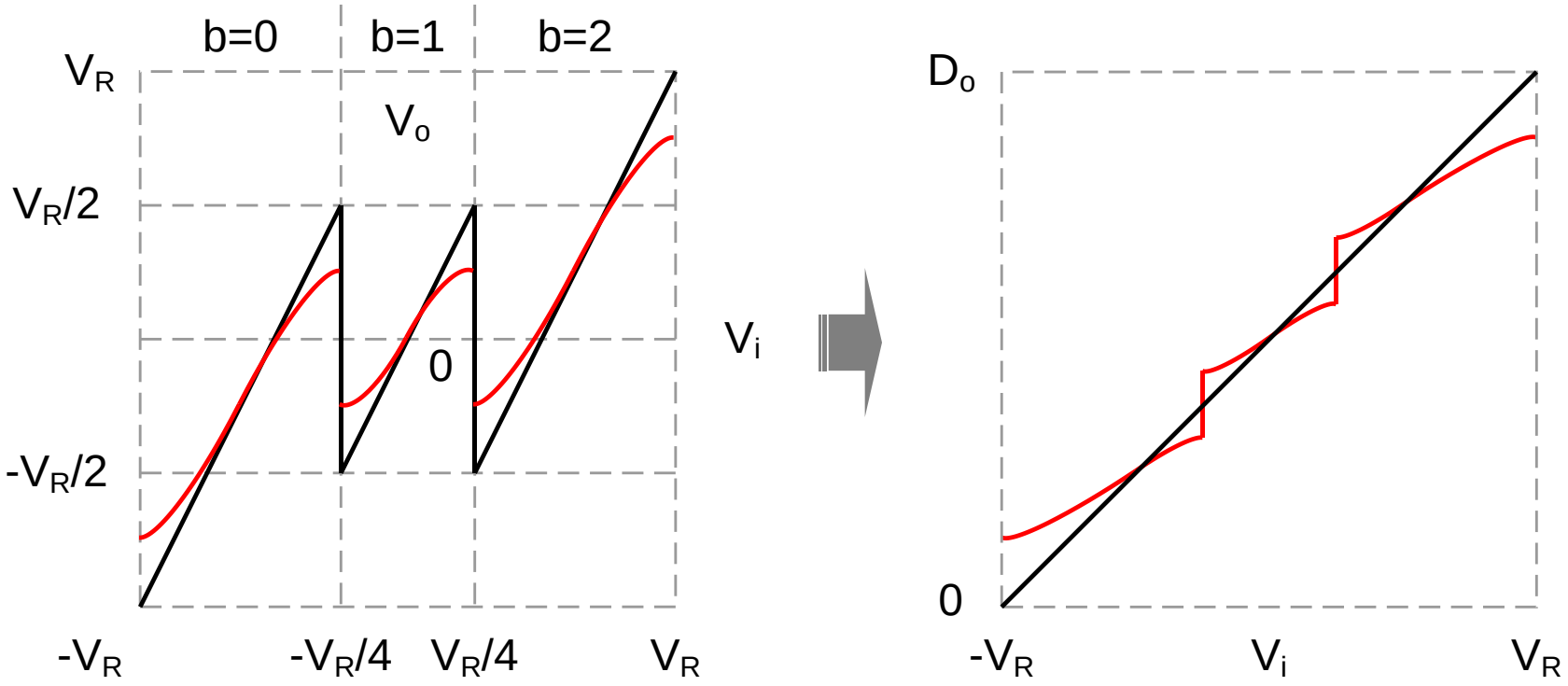


RA Gain Error and Nonlinearity



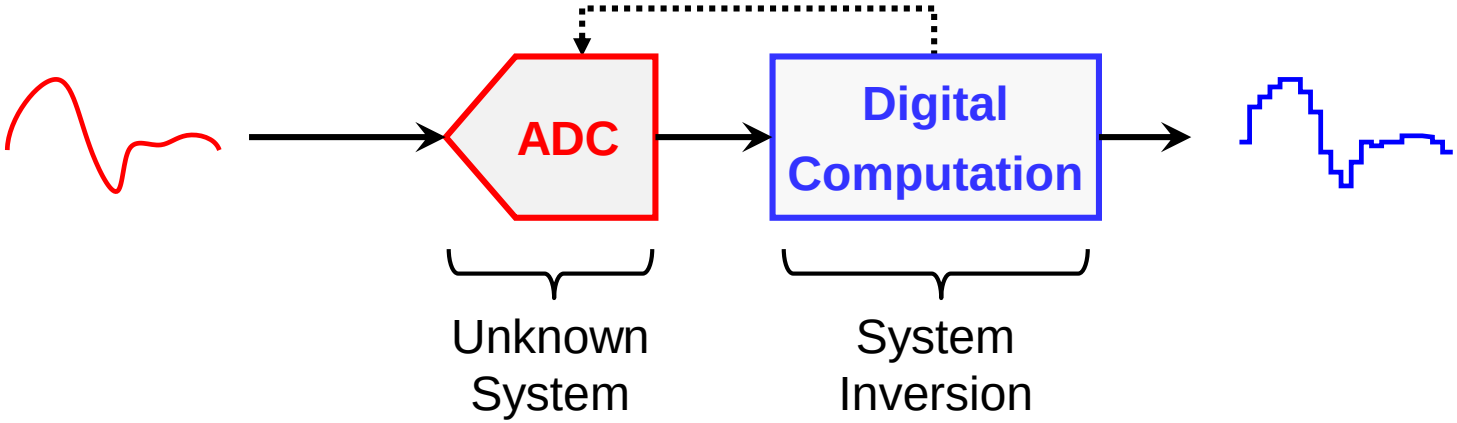
- Raw accuracy is usually limited to 10-12 bits w/o error correction.

RA Gain Error and Nonlinearity



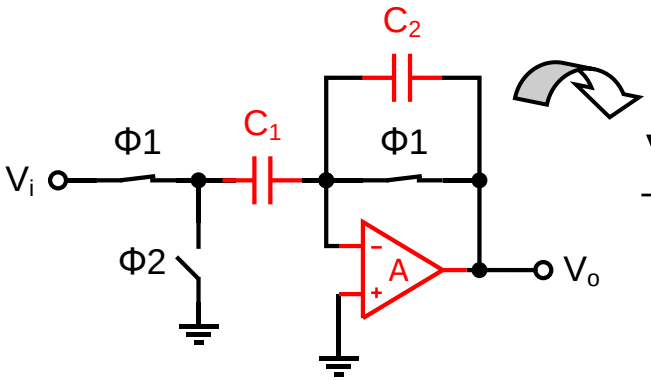
- Raw accuracy is usually limited to 10-12 bits w/o error correction.

The Basic Idea of Digital Calibration



Calibration = <u>efficient</u> digital processing to undo <u>certain</u> analog errors

e.g., SC amplifier:



$$\frac{V_o}{V_i} \approx -\frac{C_1}{C_2}\left(1-\frac{1}{\beta A}\right)$$

<u>Analog soln:</u>

- match C_1 and C_2
- make βA very large

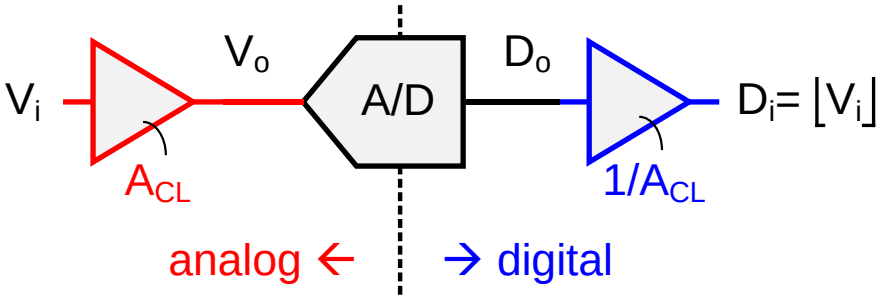
<u>Digital soln:</u>

- any constant C_1 and C_2
- any constant A is fine

Two Essential Components of Dig. Cal.

1. A digital-domain technique (e.g. equation) to recover accurate analog information from raw digital output
   - Treat analog precision or linearity only
   - Neglect small consequence on SNR

$$A_{CL} \approx -\frac{C_1}{C_2}\left(1-\frac{1}{\beta A}\right) = \#$$



2. An algorithm to identify the error parameters
   - Foreground vs. Background techniques

Linear MDAC Correction

Analog residue function:

$$V_j = \left(d_{j,1} \cdot \frac{C_1}{\sum C} + d_{j,2} \cdot \frac{C_2}{\sum C} + d_{j,3} \cdot \frac{C_3}{\sum C} \right) \cdot V_r + V_{j+1} \cdot \frac{C_4 + \sum C / A}{\sum C}$$

Normalized residue function:

$$\frac{V_j}{V_r} = \left(d_{j,1} \cdot \frac{C_1}{\sum C} + d_{j,2} \cdot \frac{C_2}{\sum C} + d_{j,3} \cdot \frac{C_3}{\sum C} \right) + \frac{V_{j+1}}{V_r} \cdot \frac{C_4 + \sum C / A}{\sum C}$$

$$\frac{V_j}{V_r} = \left(d_{j,1} + d_{j,2} + d_{j,3} \right) \cdot \frac{1}{4} + \frac{V_{j+1}}{V_r} \cdot \frac{1}{4} = d_j \cdot \frac{1}{4} + \frac{V_{j+1}}{V_r} \cdot \frac{1}{4}$$

☜ ideal residue function

Digital representation:

$$\Rightarrow \quad D_j = \left(d_{j,1} \cdot \beta_{j,1} + d_{j,2} \cdot \beta_{j,2} + d_{j,3} \cdot \beta_{j,3} \right) + D_{j+1} \cdot \alpha_j$$

$$= \sum_k d_{j,k} \cdot \beta_{j,k} + D_{j+1} \cdot \alpha_j$$

☜ error parameters: { $\boldsymbol{\alpha_j, \beta_{j,k}}$ }

Bit-Weight (Radix) Correction

For 1-b or 1.5-b MDAC:

$$D_1 = D_{in}$$

$$= \ldots + \left(d_j \cdot \beta_j + \left(d_{j+1} \cdot \beta_{j+1} + \left(d_{j+2} \cdot \beta_{j+2} + \ldots \right) \cdot \alpha_{j+1} \right) \cdot \alpha_j \right) \cdot \alpha_{j-1} \ldots$$

$$= \ldots + d_j \cdot \gamma_j + d_{j+1} \cdot \gamma_{j+1} + d_{j+2} \cdot \gamma_{j+2} + \ldots$$

☞ weighted sum of **ALL** bits!
(bit weight or radix error)

Alternatively,

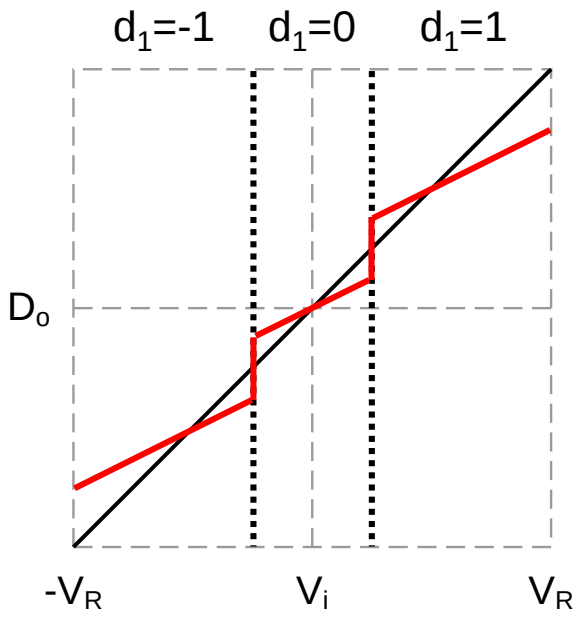
$$D_1 = D_{in}$$

$$= \ldots + \frac{d_j}{2^j} \cdot \left(1 + \Delta_j\right) + \frac{d_{j+1}}{2^{j+1}} \cdot \left(1 + \Delta_{j+1}\right) + \frac{d_{j+2}}{2^{j+2}} \cdot \left(1 + \Delta_{j+2}\right) + \ldots$$

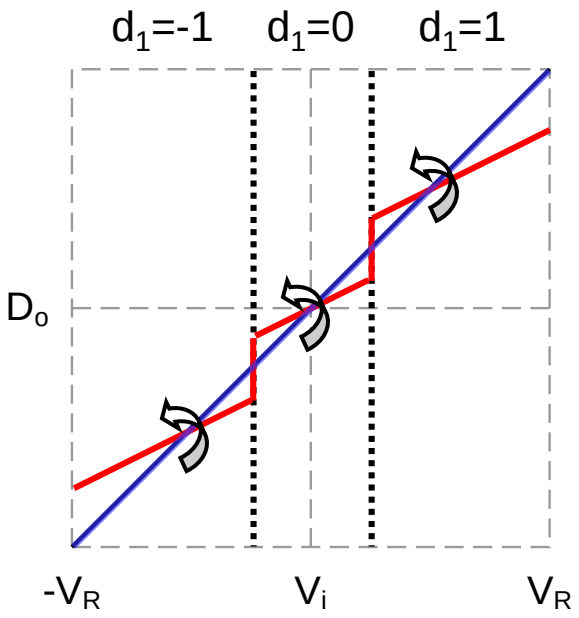
$$= \ldots + \frac{\left(d_j + d_j \Delta_j\right)}{2^j} + \frac{\left(d_{j+1} + d_{j+1} \Delta_{j+1}\right)}{2^{j+1}} + \frac{\left(d_{j+2} + d_{j+2} \Delta_{j+2}\right)}{2^{j+2}} + \ldots$$

☞ segmental offset

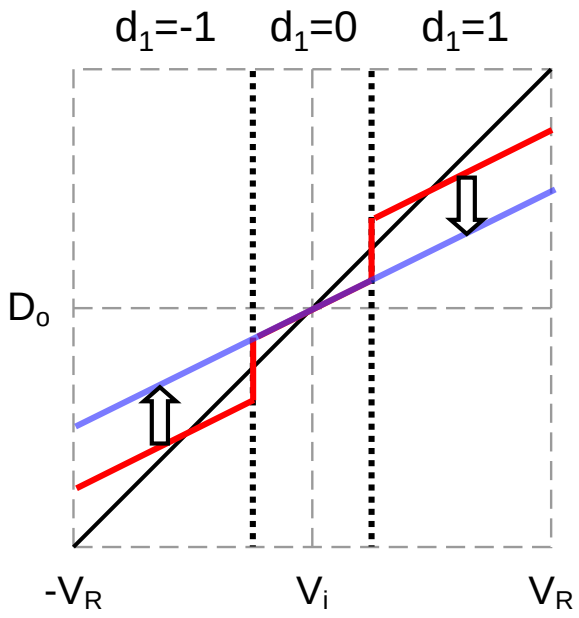
Bit-Weight (Radix) Correction



1.5b MDAC
residue nonlinearity



radix error:
needs multiplication



segmental offset:
addition only

Nonlinear MDAC Correction

Analog representation:

$$V_j = \left(d_{j,1} \cdot \frac{C_1}{\sum C} + d_{j,2} \cdot \frac{C_2}{\sum C} + d_{j,3} \cdot \frac{C_3}{\sum C} \right) \cdot V_r + V_{j+1} \cdot \frac{C_4 + \sum C / A\left(V_{j+1}\right)}{\sum C}$$

Normalized analog representation:

$$\frac{V_j}{V_r} = \left(d_{j,1} \cdot \frac{C_1}{\sum C} + d_{j,2} \cdot \frac{C_2}{\sum C} + d_{j,3} \cdot \frac{C_3}{\sum C} \right) + \frac{V_{j+1}}{V_r} \cdot \frac{C_4 + \sum C / A\left(V_{j+1}\right)}{\sum C}$$

Digital representation:

$$\Rightarrow \quad D_j = \left(d_{j,1} \cdot \beta_{j,1} + d_{j,2} \cdot \beta_{j,2} + d_{j,3} \cdot \beta_{j,3} \right) + f\left(D_{j+1}\right)$$

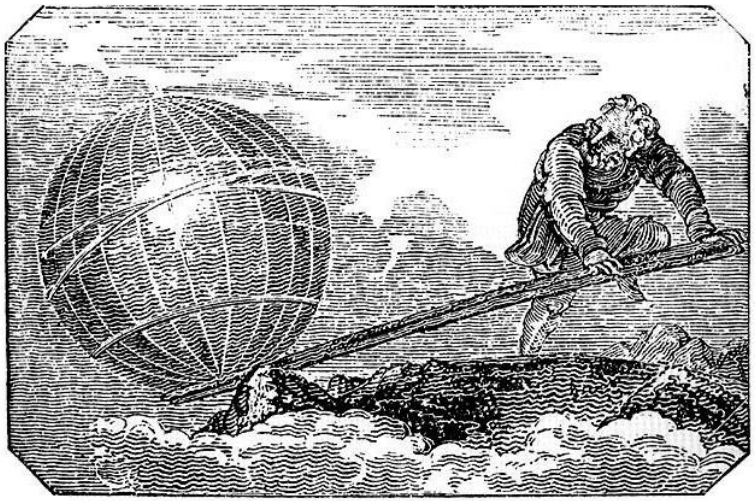
$$\approx \sum_k d_{j,k} \cdot \beta_{j,k} + \sum_m {D_{j+1}}^m \cdot \alpha_{j,m}$$

☜ error parameters: { $\boldsymbol{\alpha_{j,m}}$, $\boldsymbol{\beta_{j,k}}$ }

Next problem: how to determine {$\alpha_{j,m}$, $\beta_{j,k}$} precisely?

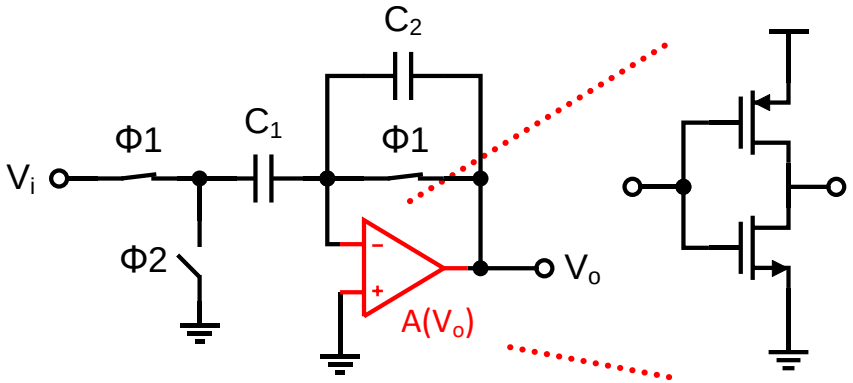
Let's try to push this…

"Give me a place to stand on, and I will move the Earth…"

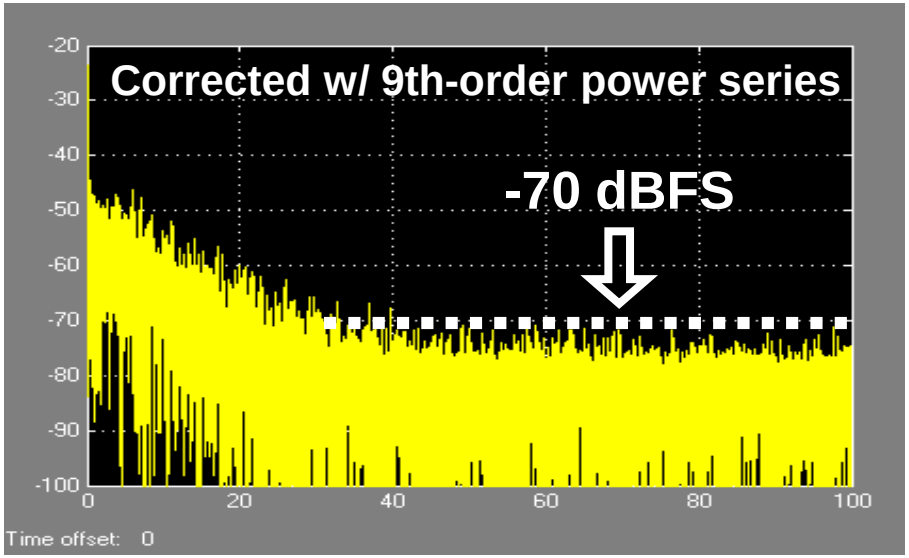


Archimedes, 200 BC

Correcting nonlinearity:



L_{Drawn}
0.15µm
V_{DD}
1.2V

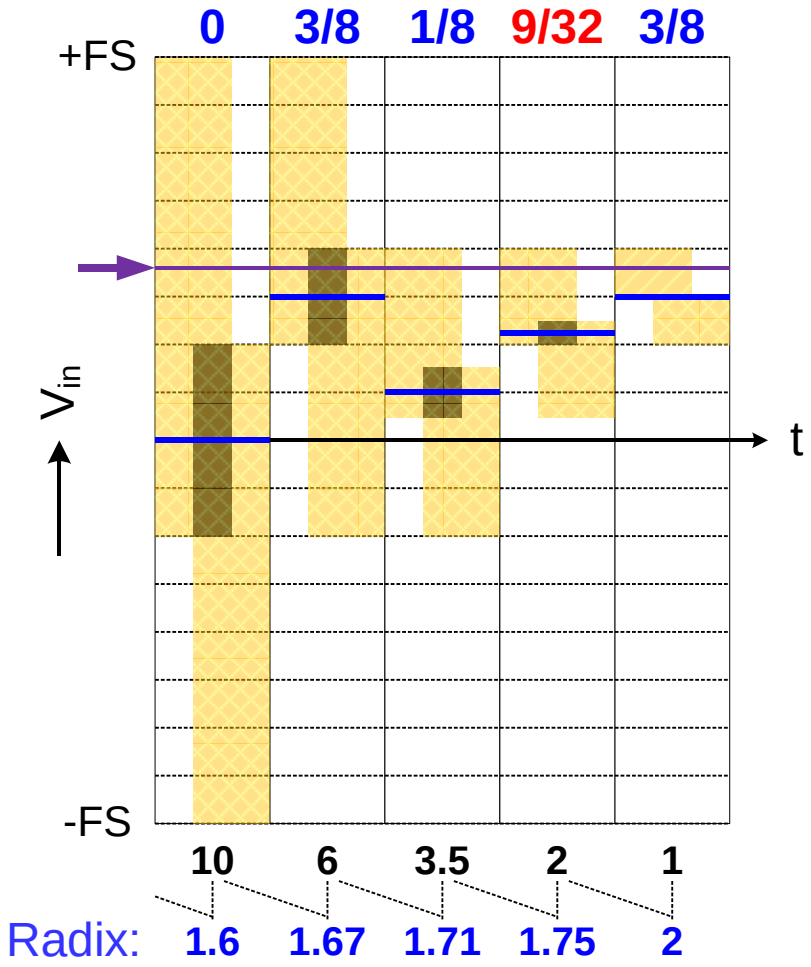


On nonlinear correction

- Memory-less polynomial computation is efficient
  - A few coefficients fits/predicts full-range nonlinearity (requiring digital multipliers and adders mostly)
  - **Caveat 1: coefficients depend on signal statistics!**
  - **Caveat 2: coefficients depend on PVT variations!**
- Piecewise-linear or lookup table can be useful
  - Memory, digital power, and cost
  - Complexity and convergence time (esp. tracking speed in background mode)

Solution needs to be practical after all…

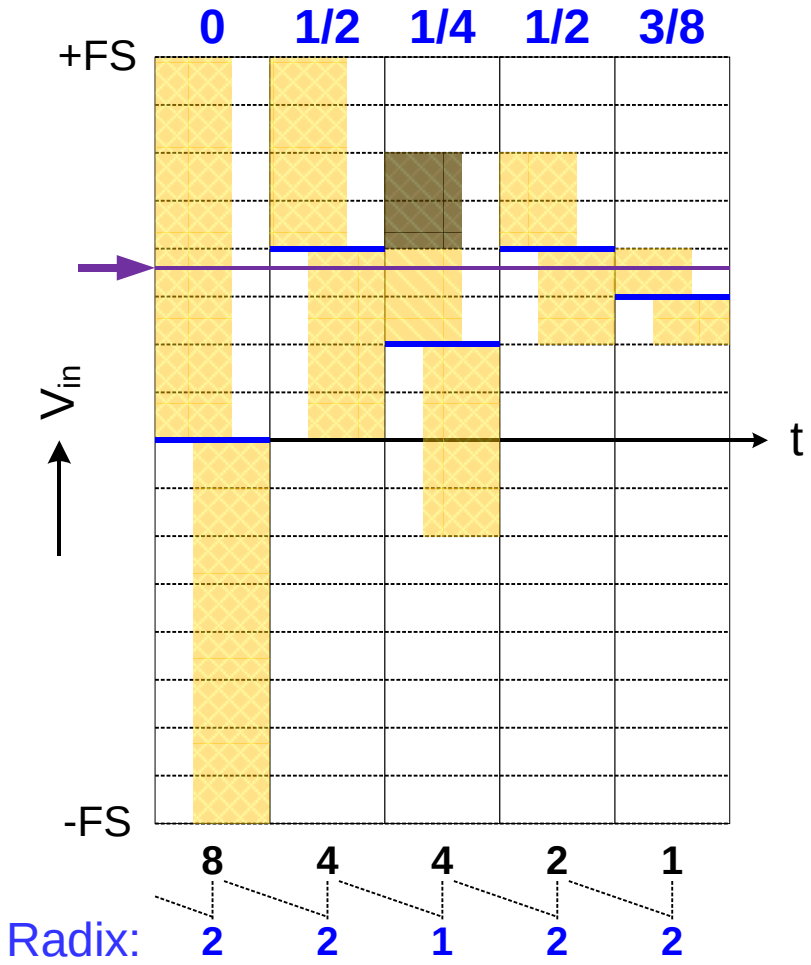
SAR Redundancy Forms (I)



- Unit-Element DAC
- Best matching
- Arbitrary decision threshold → arbitrary radix
- Redundancy @ each bit
- DAC resolution slightly higher
- Binary-to-Thermo encoder (slow)

Ref. [3]

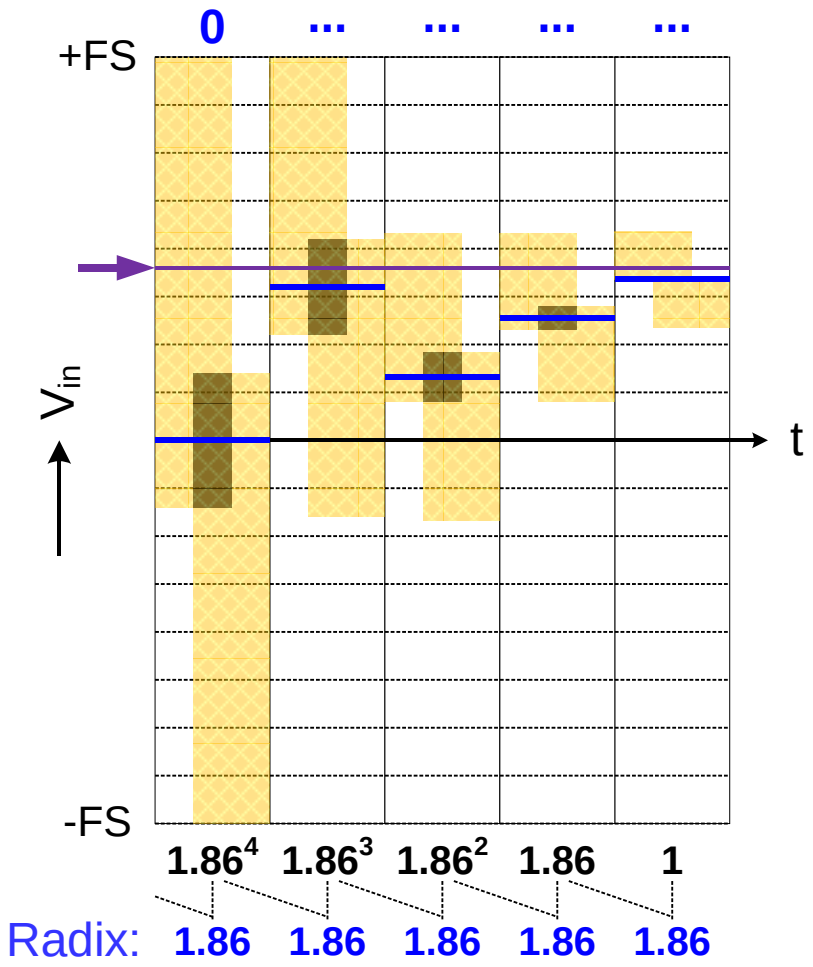
SAR Redundancy Forms (II)



- Binary DAC
- Good matching
- Periodic redundancy → non-uniform radix
- Redundancy @ selective bits
- SAR logic slightly more complex
- Can also use UE DAC

Ref. [4]

SAR Redundancy Forms (III)

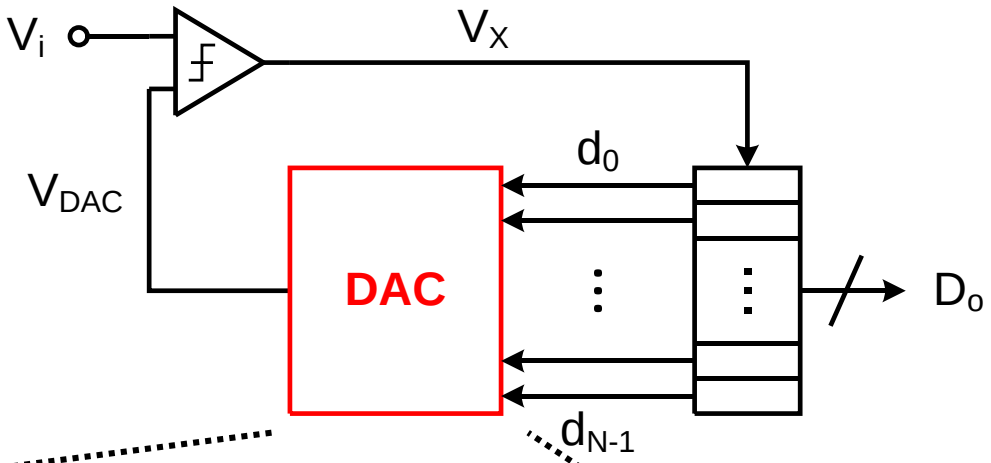


- Sub-binary DAC
- Poor matching
- Uniform redundancy → uniform radix
- Redundancy @ each bit
- Simple layout, simple SAR logic (fast)
- Cannot use UE DAC
- **Must calibrate DAC**

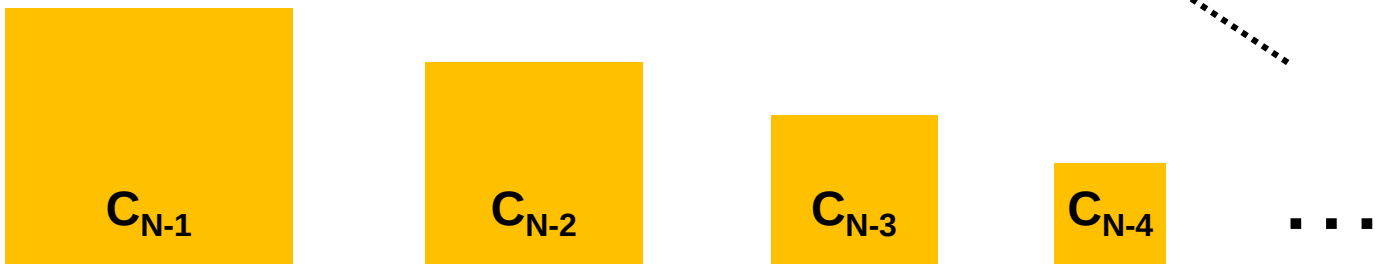
Ref. [5]

Sub-binary DAC – Construction

- Hard-coded in analog form
- No B2T encoder
- Simple layout



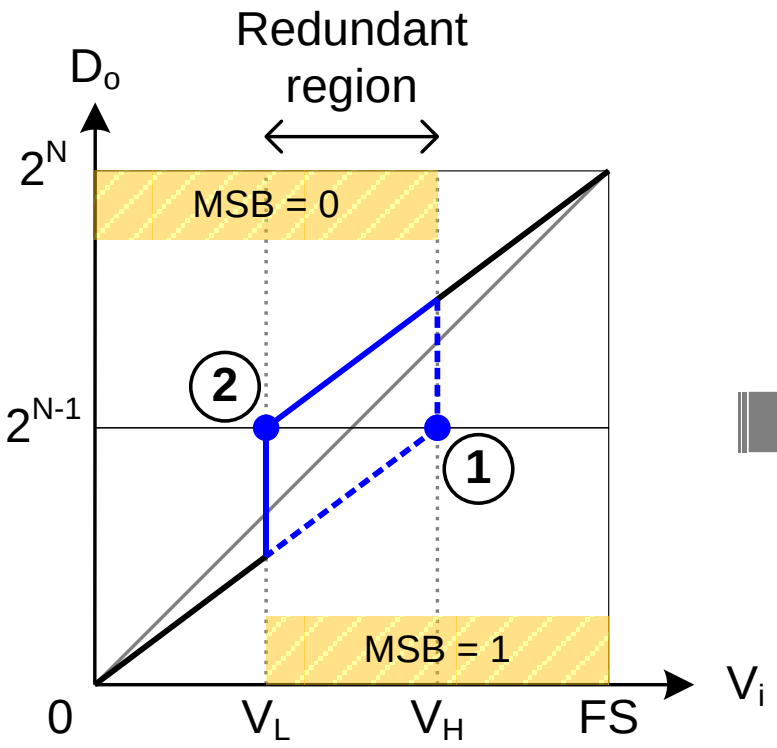
e.g., Radix = 1.86

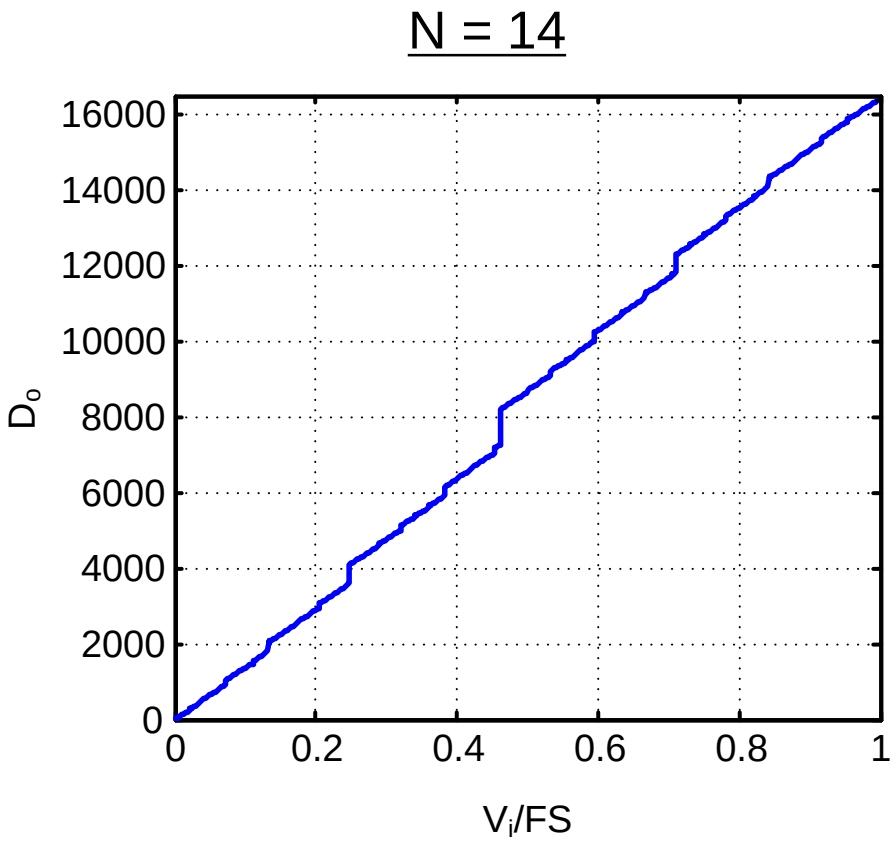


$C_0 \cdot 1.86^{N-1}$ $C_0 \cdot 1.86^{N-2}$ $C_0 \cdot 1.86^{N-3}$ $C_0 \cdot 1.86^{N-4}$

Ref. [5]

Sub-binary DAC – Transfer Function





① = **011…1** → V_H

② = **100…0** → V_L

Note: only one transition edge shows up

Sub-binary DAC – Quantization

Ref. [5]

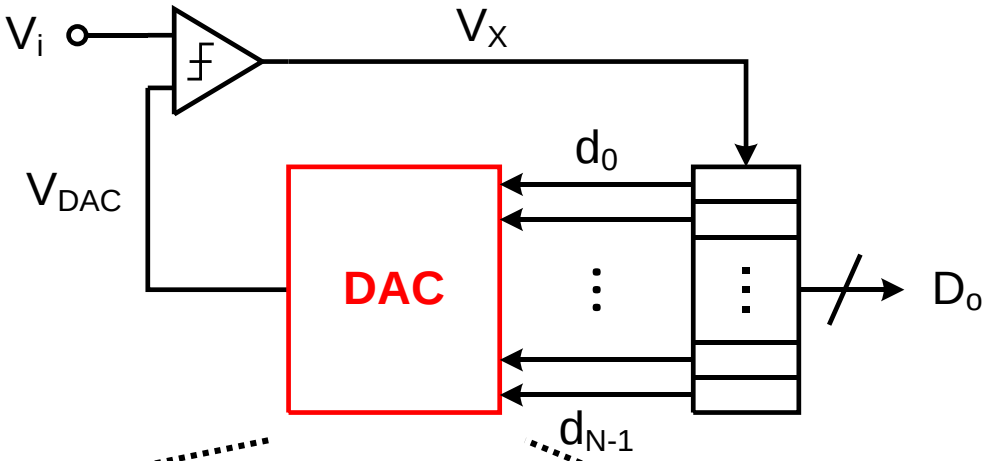
$$d_o = \left\lfloor \frac{V_i}{V_{FS}} \right\rfloor$$

$$\approx \sum_{j=0}^{N-1} \left(\frac{C_j}{\sum C_j} \right) \cdot \left(2d_j - 1 \right)$$

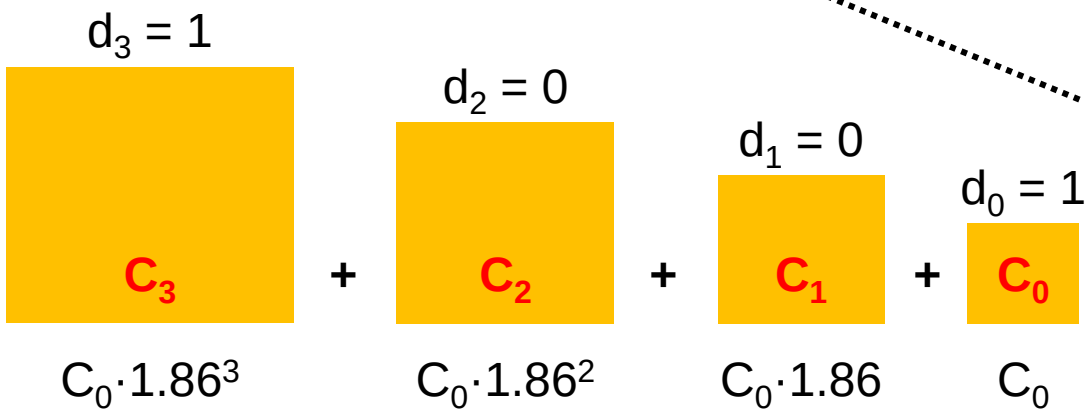
$$= \sum_{j=0}^{N-1} w_j \cdot \left(2d_j - 1 \right)$$

$$Q_{DAC} = V_R \sum_{j=0}^{3} C_j \left(2d_j - 1 \right)$$

$$\approx V_i \sum_{j=0}^{3} C_j$$

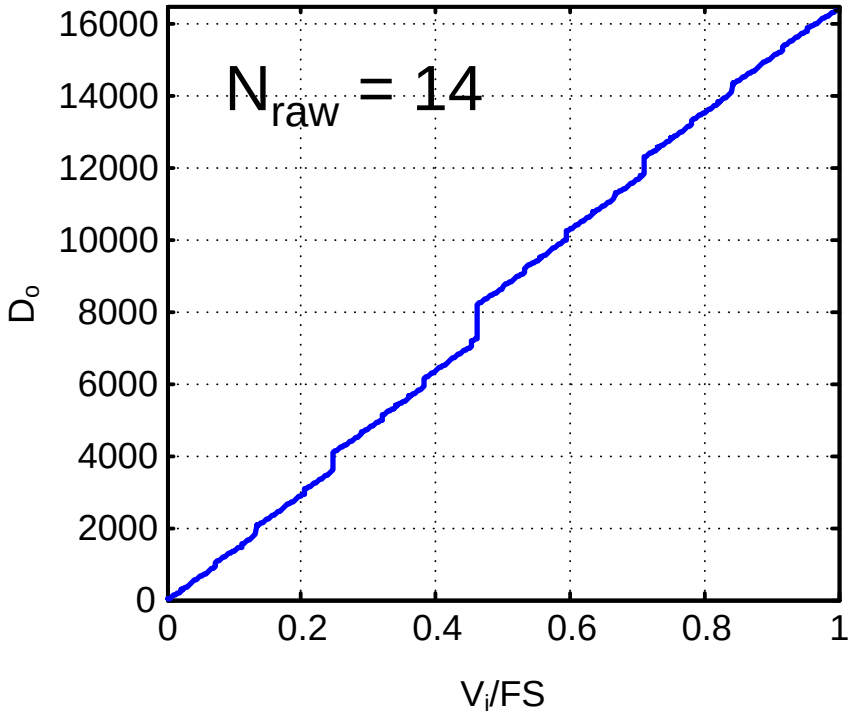


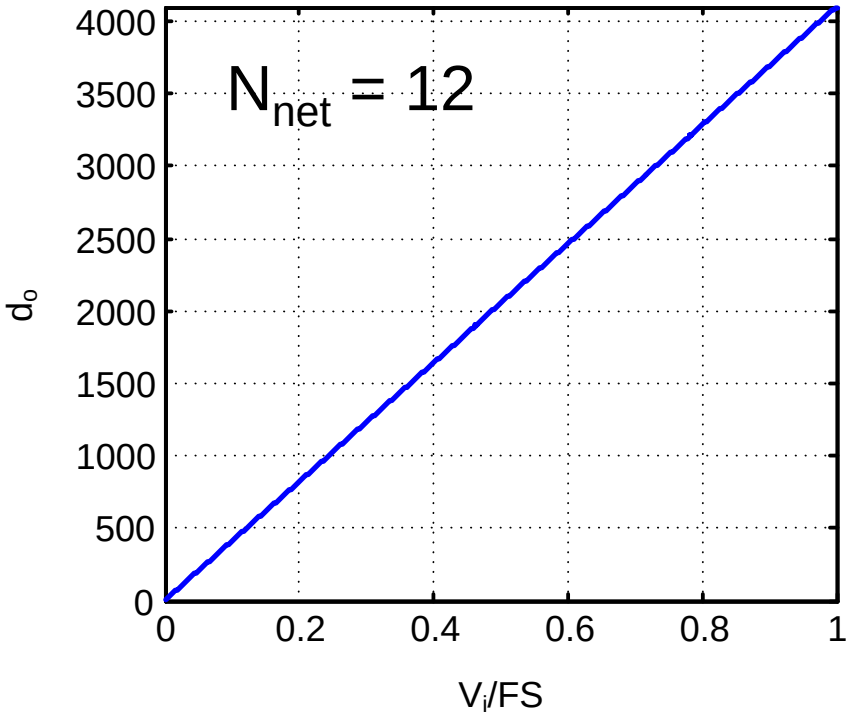
Radix = 1.86



Sub-binary DAC – Digital Correction

$$d_o = \left\lfloor \frac{V_i}{V_{FS}} \right\rfloor = \sum_{j=0}^{N-1} w_j \cdot (2d_j - 1) \qquad \{w_j\} = \text{bit weights}$$





Next problem: how to determine $\{w_j\}$ precisely?

Presentation Outline

- Principles of Multistep A/D Conversion
- Architectural Redundancy
- Error Mechanisms and Digital-Domain Calibration
- Error-Parameter Identification
  - PRBS Test-Signal Injection (sub-ADC, sub-DAC, input)
  - Two-ADC Equalization (ref.-ADC, split-ADC, ODC)
- Energy Efficiency and Trend
- Summary

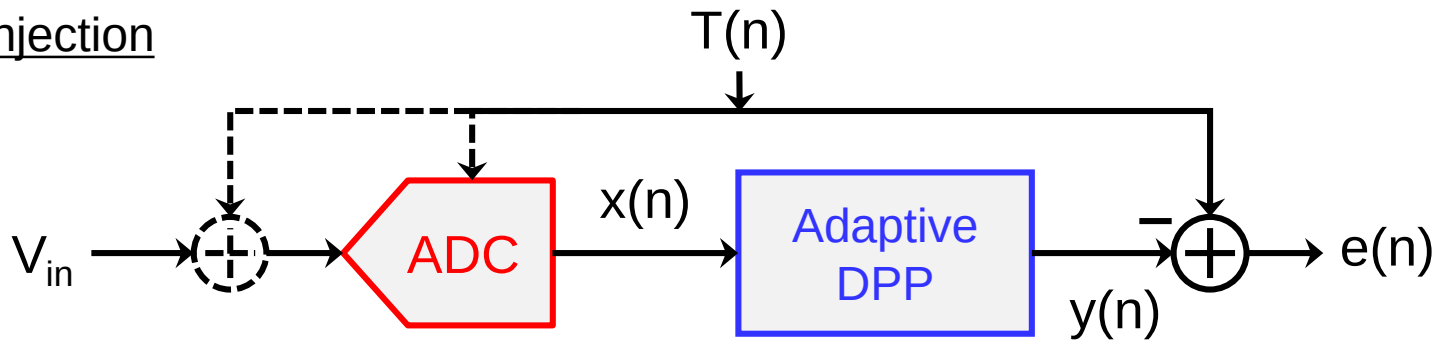
Error-Parameter Identification Techniques (BG)

- **With PRBS test-signal injection (dither)**
  - Sub-ADC injection (comparator dither)
  - Sub-DAC injection (DAC dither)
  - Input injection (Independent Component Analysis)
- **With two-ADC equalization (test signal free)**
  - Reference-ADC equalization (training sequence)
  - Split-ADC equalization (blind)
  - Offset double conversion (ODC) (blind, single ADC)

Parameter extraction is what the game is all about…

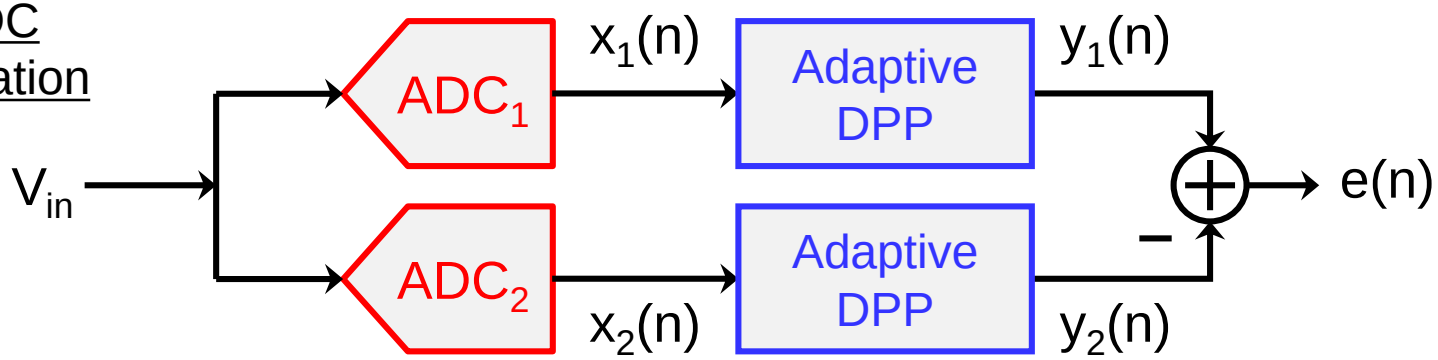
Recent BG Digital Calibration Techniques

PRBS injection (Dither)



Temes [6,7], Lewis [8], Galton [9,10]…

Two-ADC equalization



Lewis [11], Chiu [12], McNeill [13]…

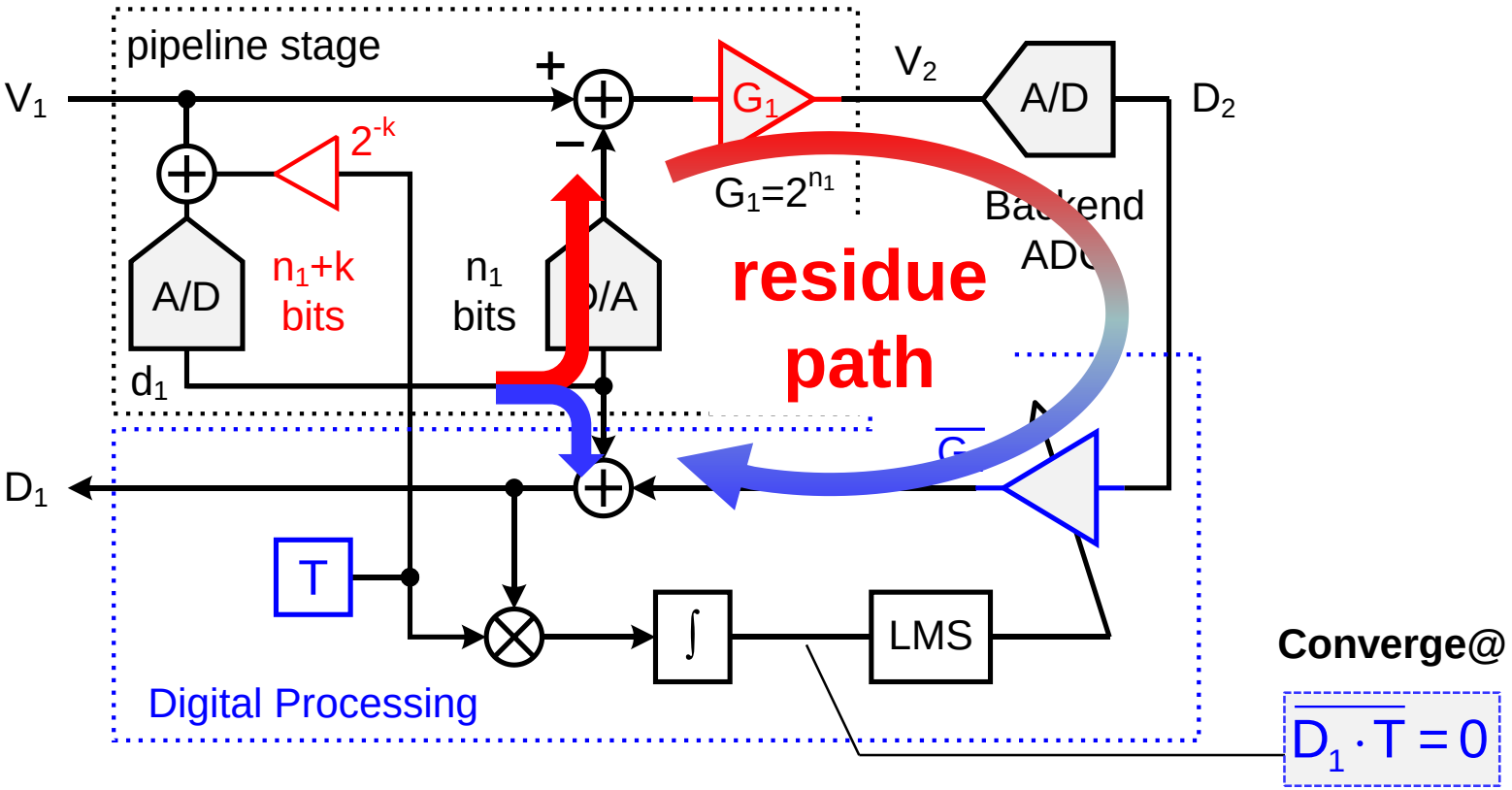
PRBS Test-Signal Injection

Comparison of PRBS Injection Techniques

- **Sub-ADC injection**
  - considered as dynamic comparator offset, no removal needed
  - higher sub-ADC resolution (injection and ADC matching not req'd)
  - works only with busy input
- **Sub-DAC injection**
  - needs to be removed in digital output
  - higher sub-DAC resolution (injection and DAC matching req'd)
  - can work with quiet input
- **Input injection (sub-DAC + sub-ADC)**
  - needs to be removed in digital output
  - No impact on sub-ADC or sub-DAC resolution
  - works only with busy input

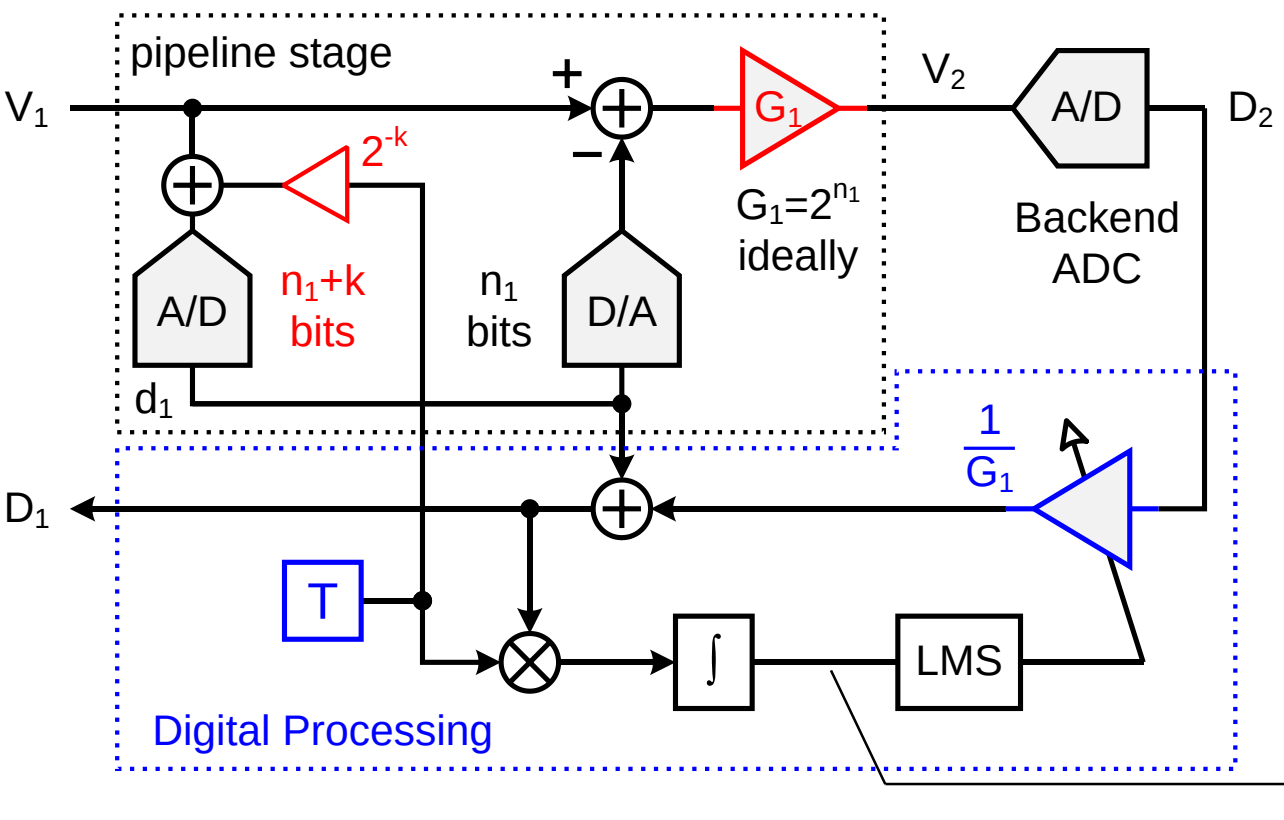
PRBS Test-Signal Injection (Sub-ADC Injection)

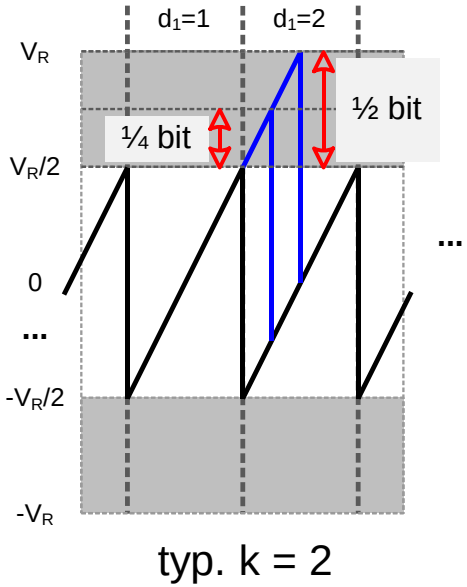
Sub-ADC Injection – Comparator Dither



- In steady state, analog gain (G_1) and digital gain (G_1^{-1}) cancel exactly.
- $2^{-k} \le ¼$ to avoid overflow in residue output.
- No need to match injection scaling factor (2^{-k}) to the sub-ADC thresholds.

Sub-ADC Injection – Comparator Dither





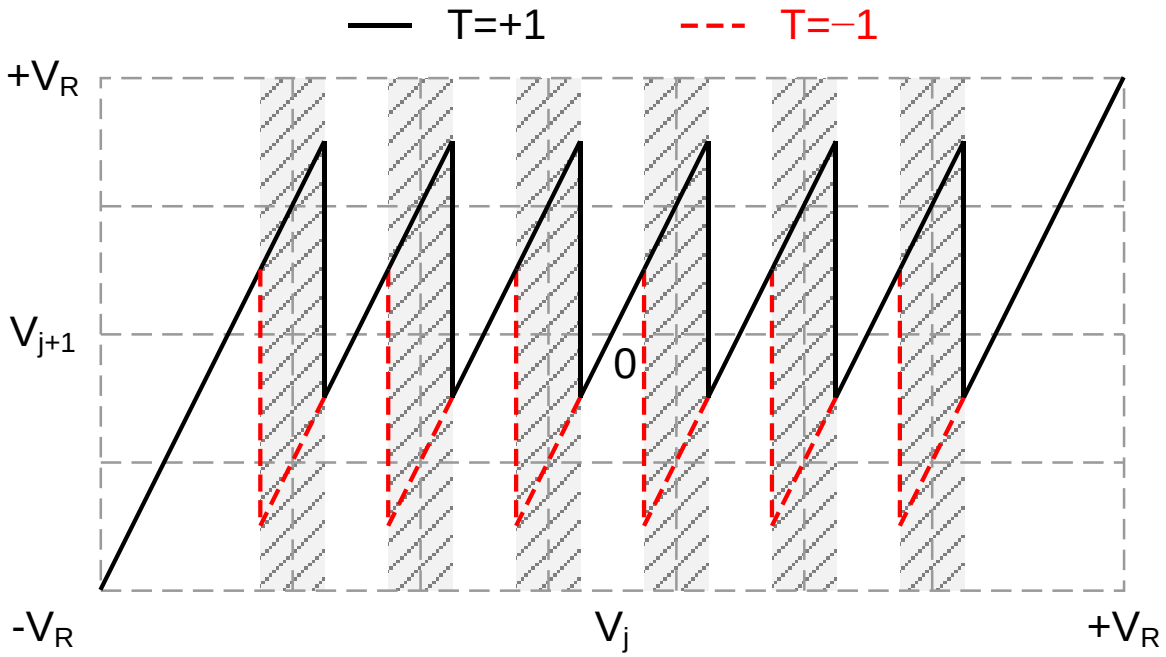
Converge@

$\overline{D_1 \cdot T} = 0$

- In steady state, analog gain (G_1) and digital gain (G_1^{-1}) cancel exactly.
- $2^{-k} \le ¼$ to avoid overflow in residue output.
- No need to match injection scaling factor (2^{-k}) to the sub-ADC thresholds.

Exploiting Internal Redundancy

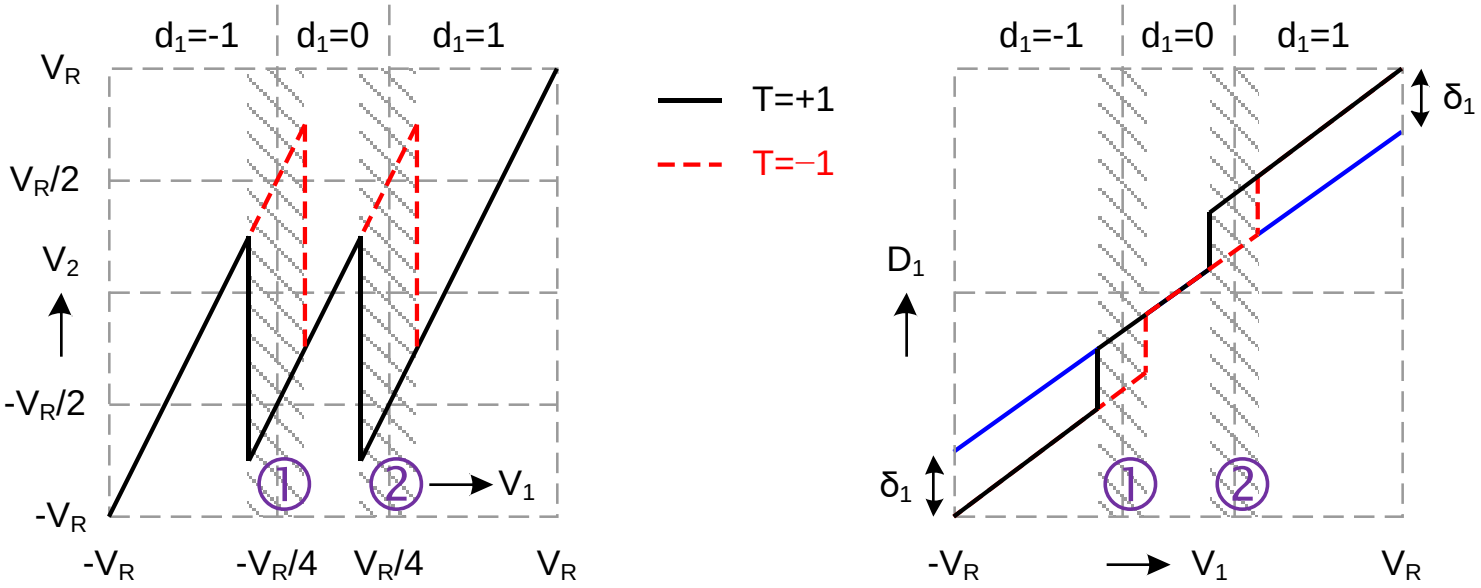
Ref. [14]



- Input falling in shaded region randomly sees one of two RTF's → dithering.
- Decision threshold needs not to be accurate or matched to each other.
- Digitization outcome is independent of PRBS when ADC is ideal !!

Identifying Residue Gain Error

Segmental offset: $D_1 = \frac{1}{4}(d_1 + d_1 \cdot \delta_1) + \frac{1}{8}d_2 + \frac{1}{16}d_3 + \ldots$



If $V_1 \in \{\text{region 1}\}$ and $T = +1$, $D_1 = D_{ideal}$; if $T = -1$, $D_1 = D_{ideal} - \delta_1$

If $V_1 \in \{\text{region 2}\}$ and $T = +1$, $D_1 = D_{ideal} + \delta_1$; if $T = -1$, $D_1 = D_{ideal}$

Identifying Residue Gain Error

Calculating correlation:

$$\overline{D_1 \cdot T} = \frac{1}{2}\left[\overline{D_{ideal}} - \left(\overline{D_{ideal}} - \delta_1\right)\right] \cdot \Pr\left(V_1 \in \{\text{region 1}\}\right) + \frac{1}{2}\left[\left(\overline{D_{ideal}} + \delta_1\right) - \overline{D_{ideal}}\right] \cdot \Pr\left(V_1 \in \{\text{region 2}\}\right)$$

$$= \frac{1}{2}\delta_1 \cdot \Pr\left(V_1 \in \{\text{region 1}\}\right) + \frac{1}{2}\delta_1 \cdot \Pr\left(V_1 \in \{\text{region 2}\}\right)$$

$$= \frac{1}{2}\delta_1 \cdot \Pr\left(V_1 \in \{\text{region 1 or 2}\}\right)$$

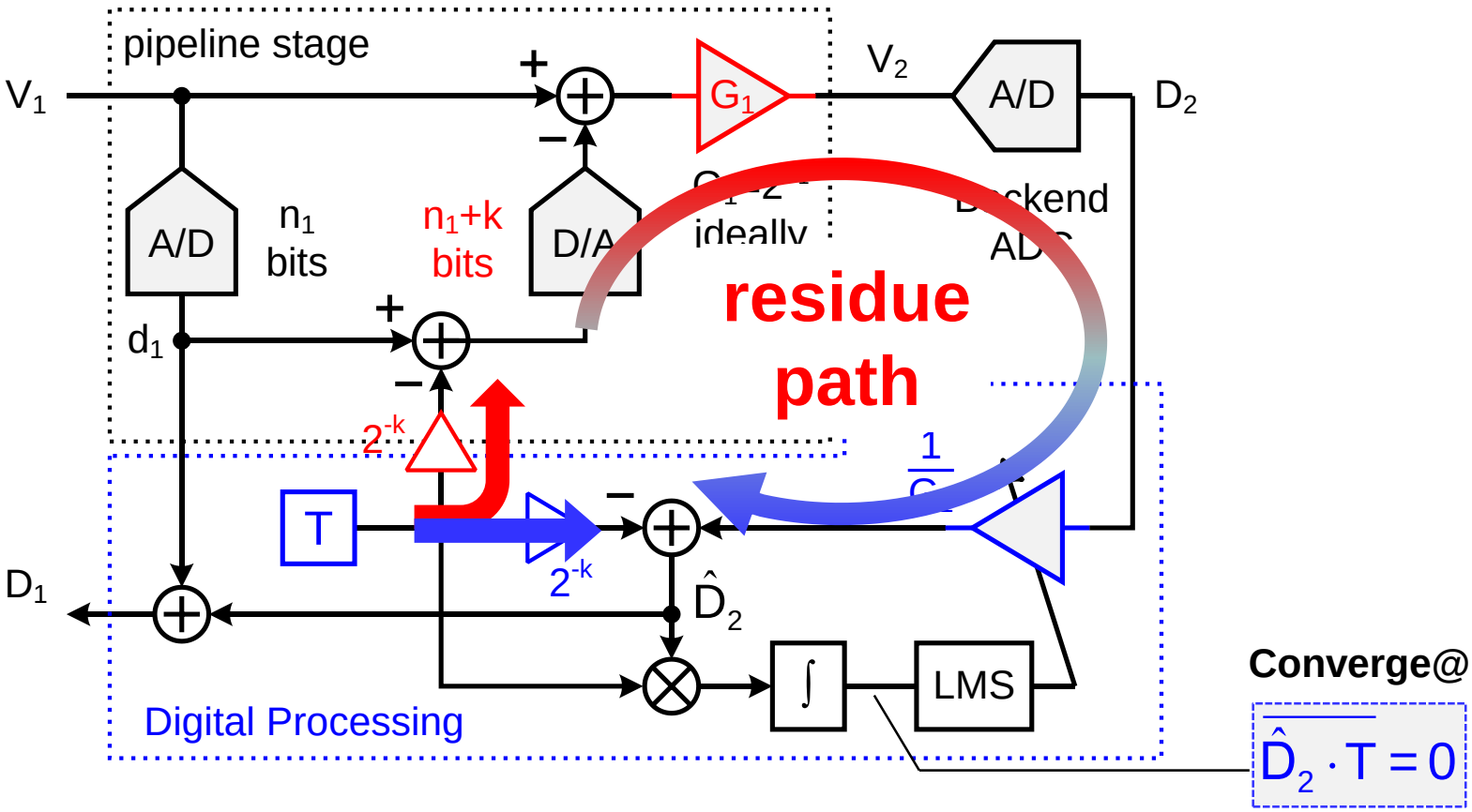
LMS learning:

$$\delta_1(n+1) = \delta_1(n) + \mu \cdot D_1(n)T(n) \quad \Rightarrow \quad \overline{D_1\left(\delta_1 \text{ removed}\right) \cdot T} \to 0$$

- Correlation reveals information about segmental offset.
- Exact size of shaded region is not important (only affects Pr(.)).
- **Key observation:** if ADC is ideal, D_1 must be uncorrelated to T.

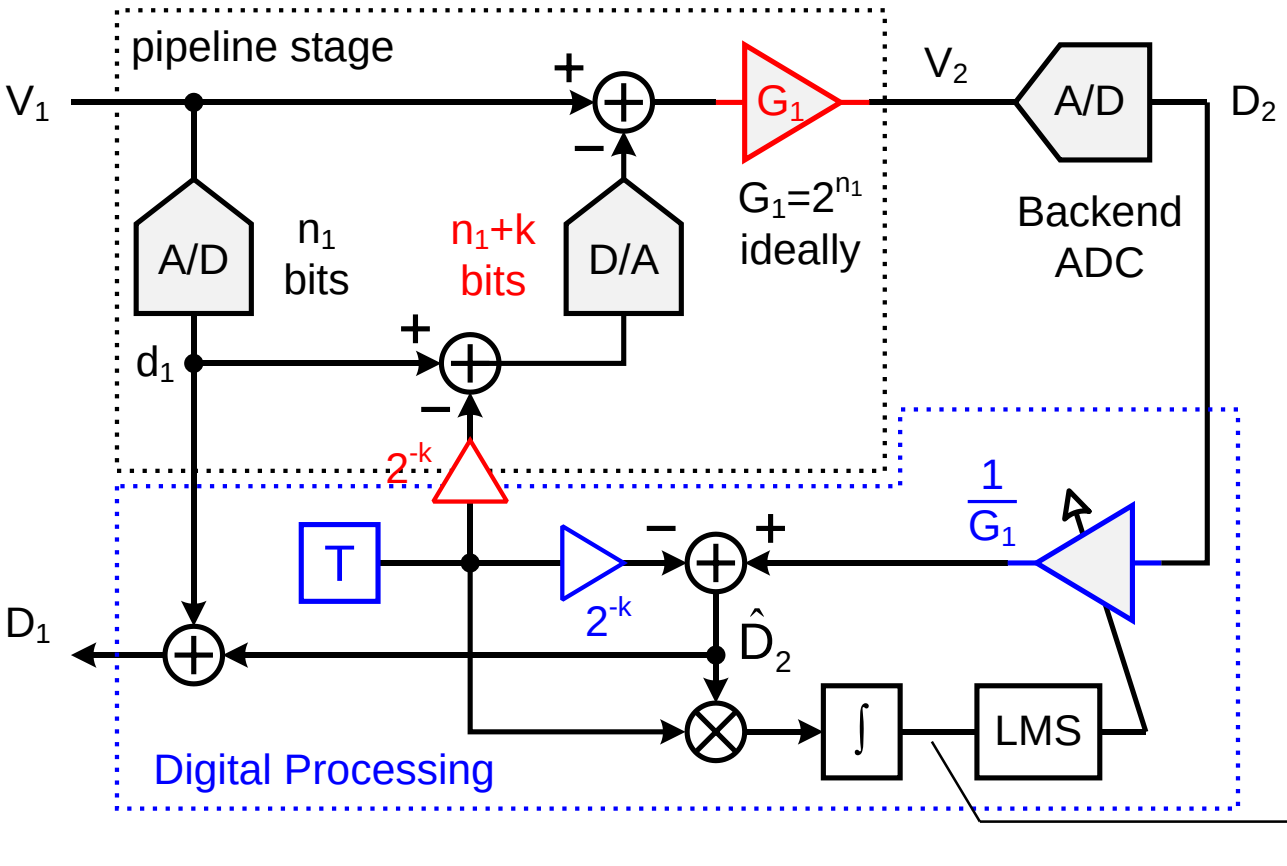
PRBS Test-Signal Injection (Sub-DAC Injection)

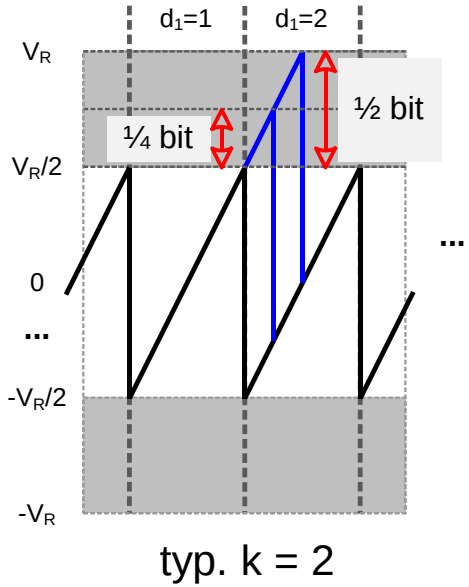
Sub-DAC Injection – DAC Dither



- In steady state, analog gain (G_1) and digital gain (G_1^{-1}) cancel exactly.
- $2^{-k} \le ¼$ to avoid overflow in residue output, DAC adds 2 bits minimum.
- Injection bit scaling factor (2^{-k}) must match to the sub-DAC unit elements.

Sub-DAC Injection – DAC Dither



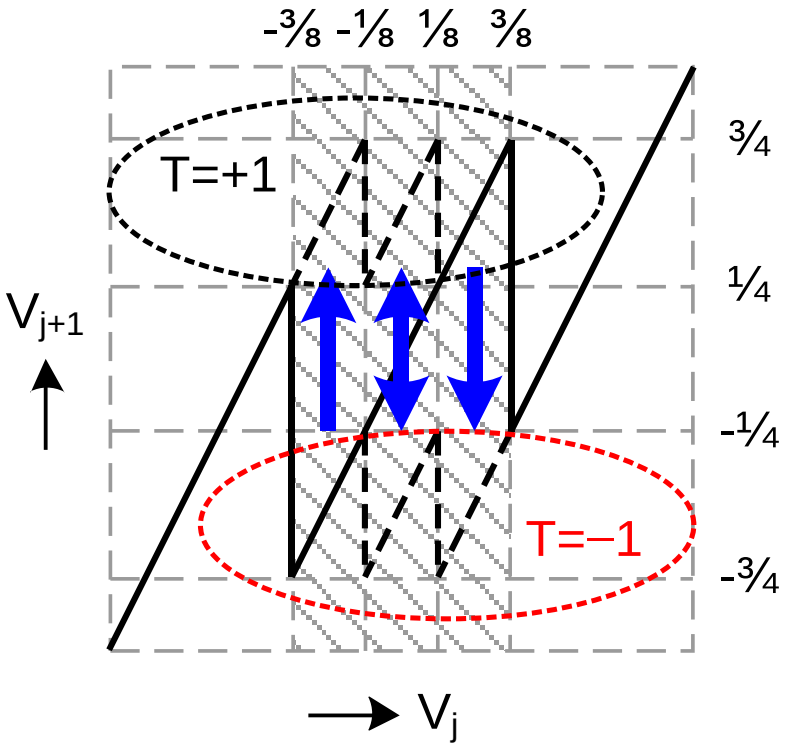


Converge@

$$\overline{\hat{D}_2 \cdot T} = 0$$

- In steady state, analog gain (G_1) and digital gain (G_1^{-1}) cancel exactly.
- $2^{-k} \le ¼$ to avoid overflow in residue output, DAC adds 2 bits minimum.
- Injection bit scaling factor (2^{-k}) must match to the sub-DAC unit elements.

Signal-Dependent DAC Dither



PRBS Injection Table

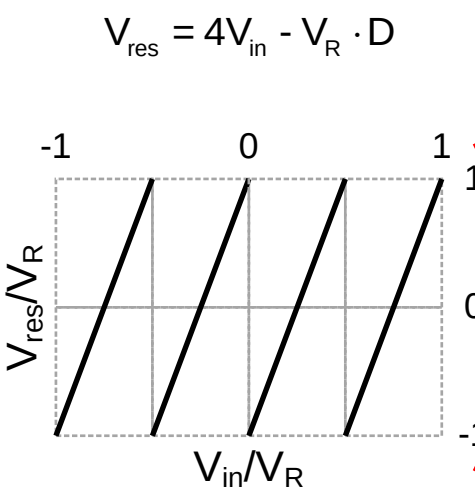
V_j (V_R)	T = +1	T = -1
-1 → -⅜	0	0
-⅜ → -⅛	0	V_R
-⅛ → ⅛	-½ V_R	½ V_R
⅛ → ⅜	-V_R	0
⅜ → 1	0	0

- PRBS only injected when input falls within the shaded region.
- Extra comparators needed to instrument the SD dither.

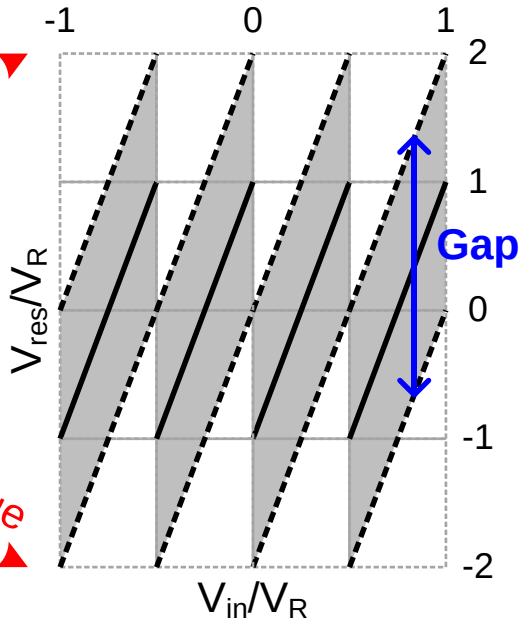
Ref. [15]

Opportunistic DAC Dither

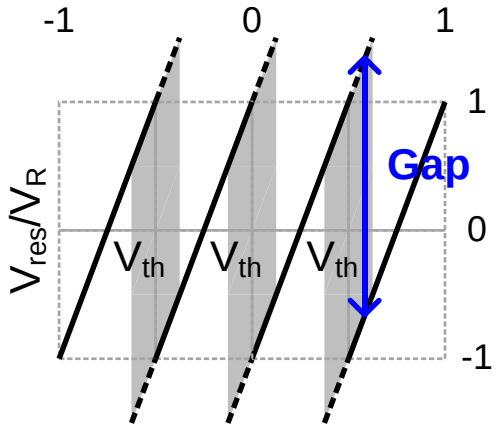
$$V_{res} = 4V_{in} - V_R \cdot D$$



$$V_{res} = 4V_{in} - V_R (D + PN)$$

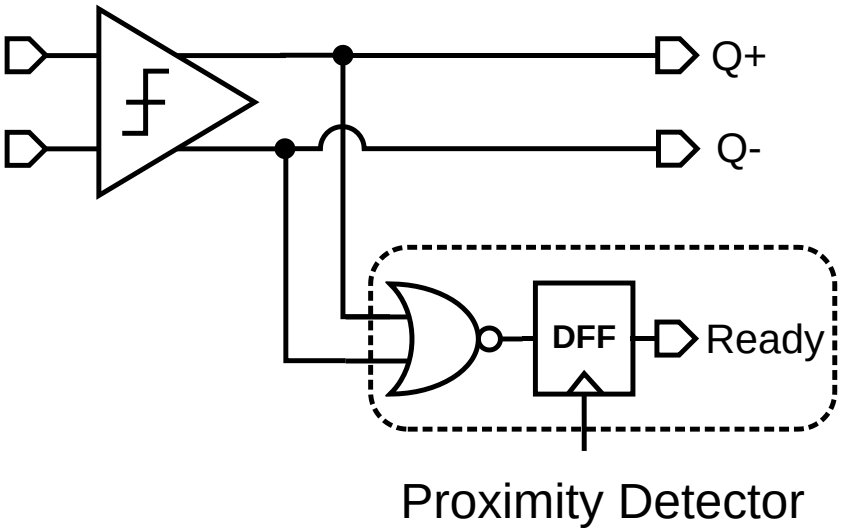


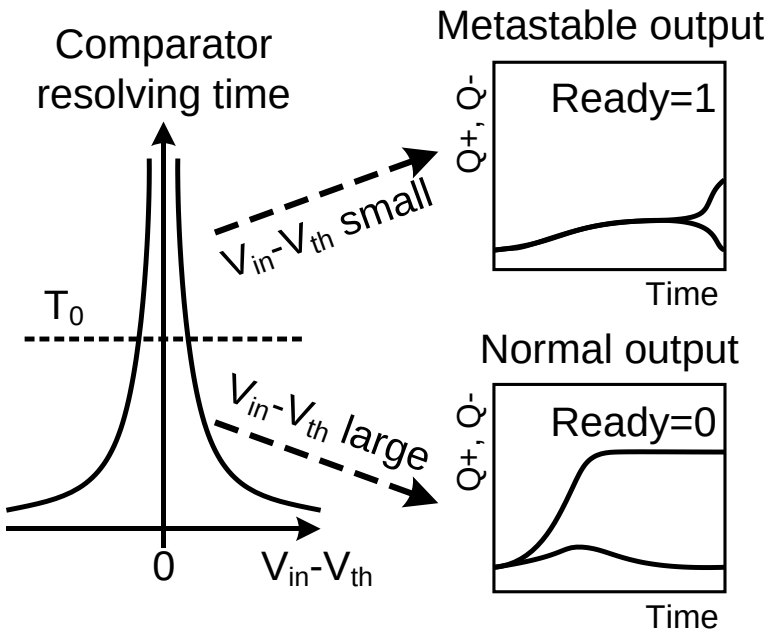
$$V_{res} = \begin{cases} 4V_{in} - V_R \cdot (D' + PN), & V_{in} \approx V_{th} \\ 4V_{in} - V_R \cdot D, & \text{o.w.} \end{cases}$$



- When redundancy is not ample, blind injection requires large DR.
- Without additional comparators, detecting V_{th} vicinity is difficult.

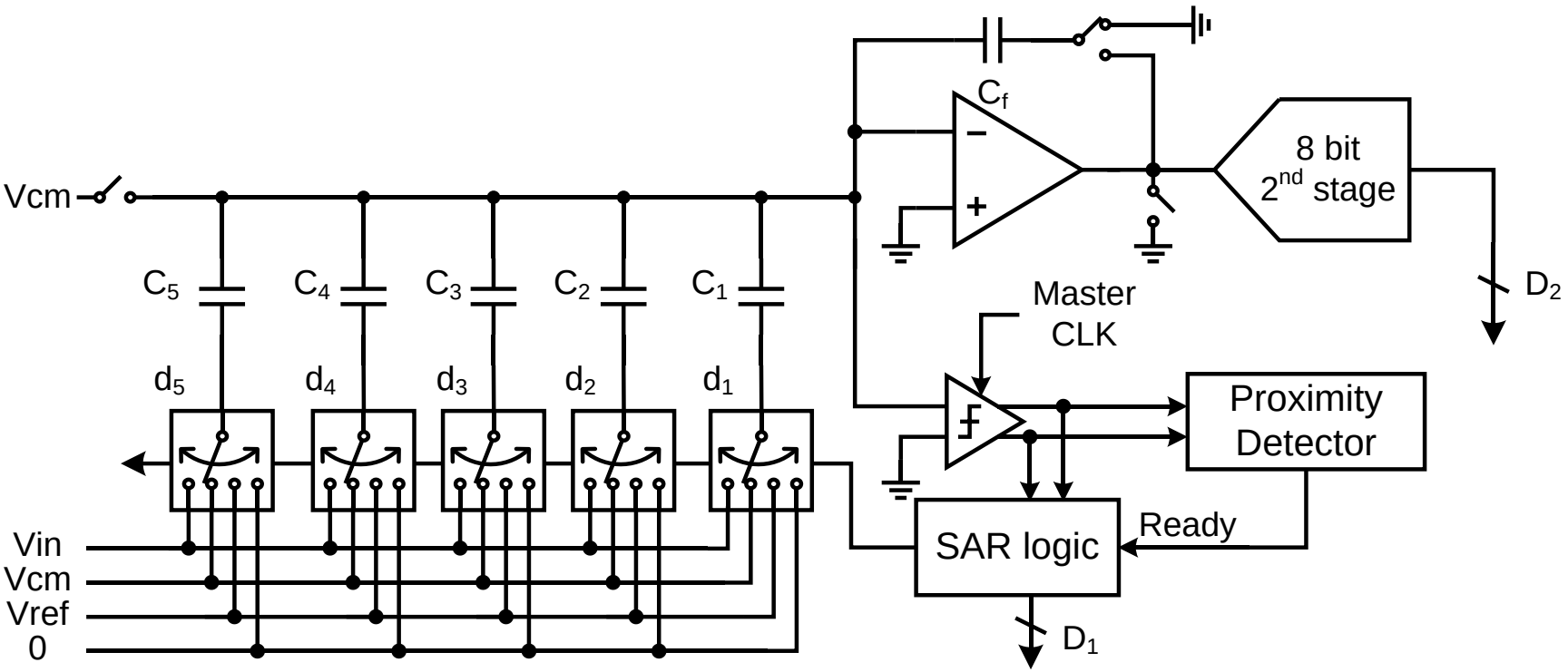
Exploiting Comparator Metastability





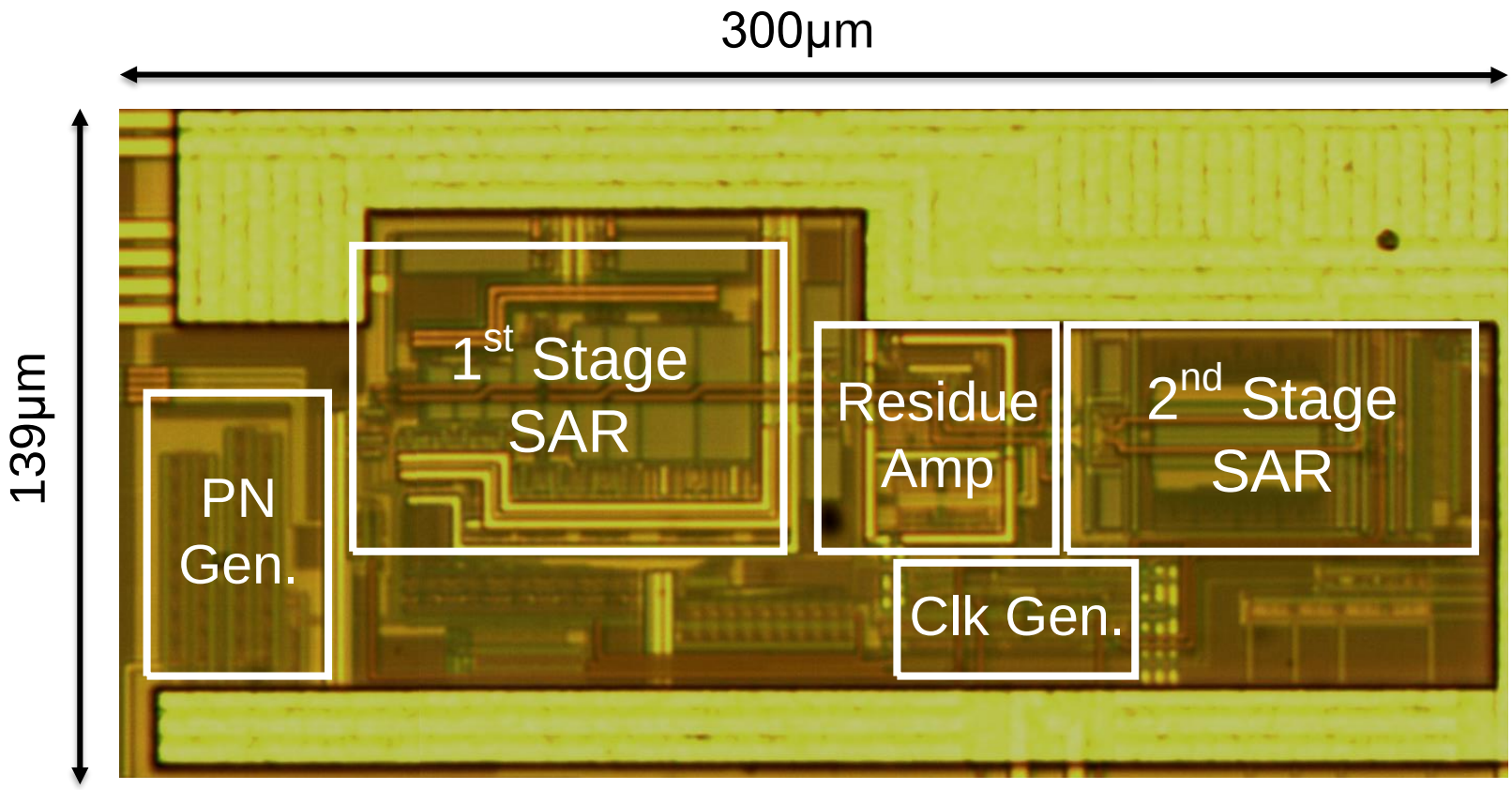
- Comparator resolving time indicates proximity of input.
- Proximity detector also functions as metastabilty detector/resolver.

12b 160MS/s CMOS Prototype (40nm)



- (5b + 8b) synchronous two-step pipelined SAR architecture.
- First-stage capacitor weights identified w/ opportunistic DAC dither.

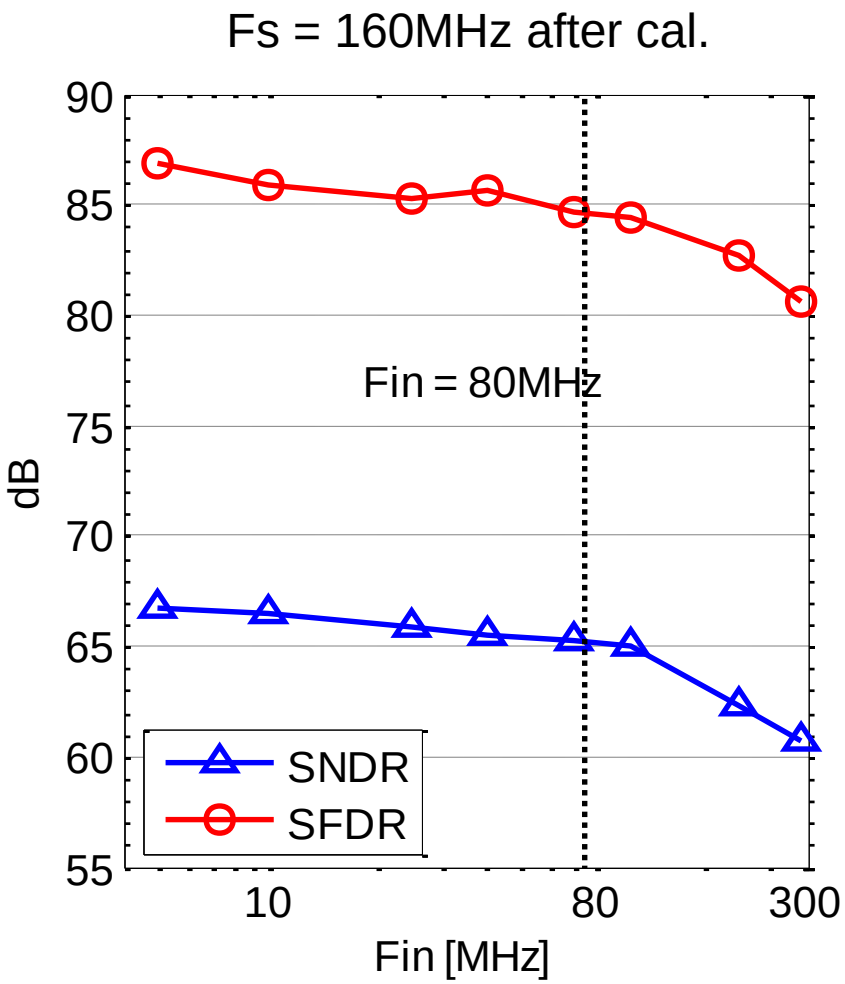
Die Photo

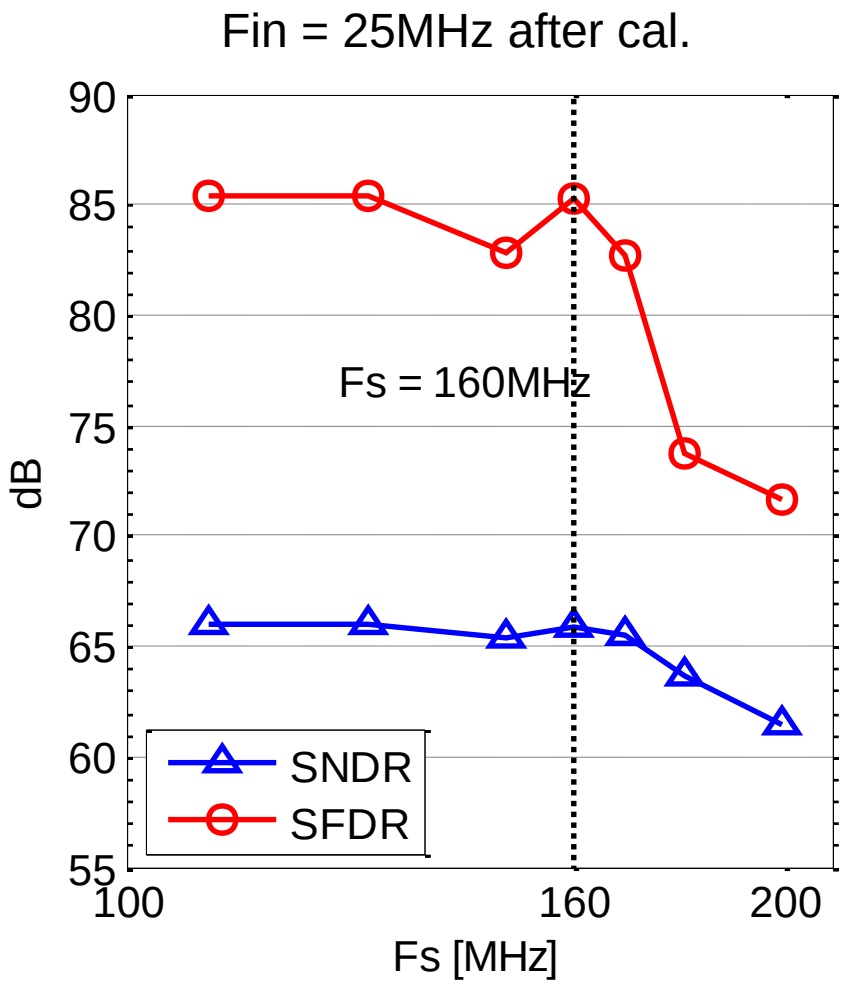


40nm low-leakage CMOS process
(active area = 0.042mm^2)

Ref. [16]

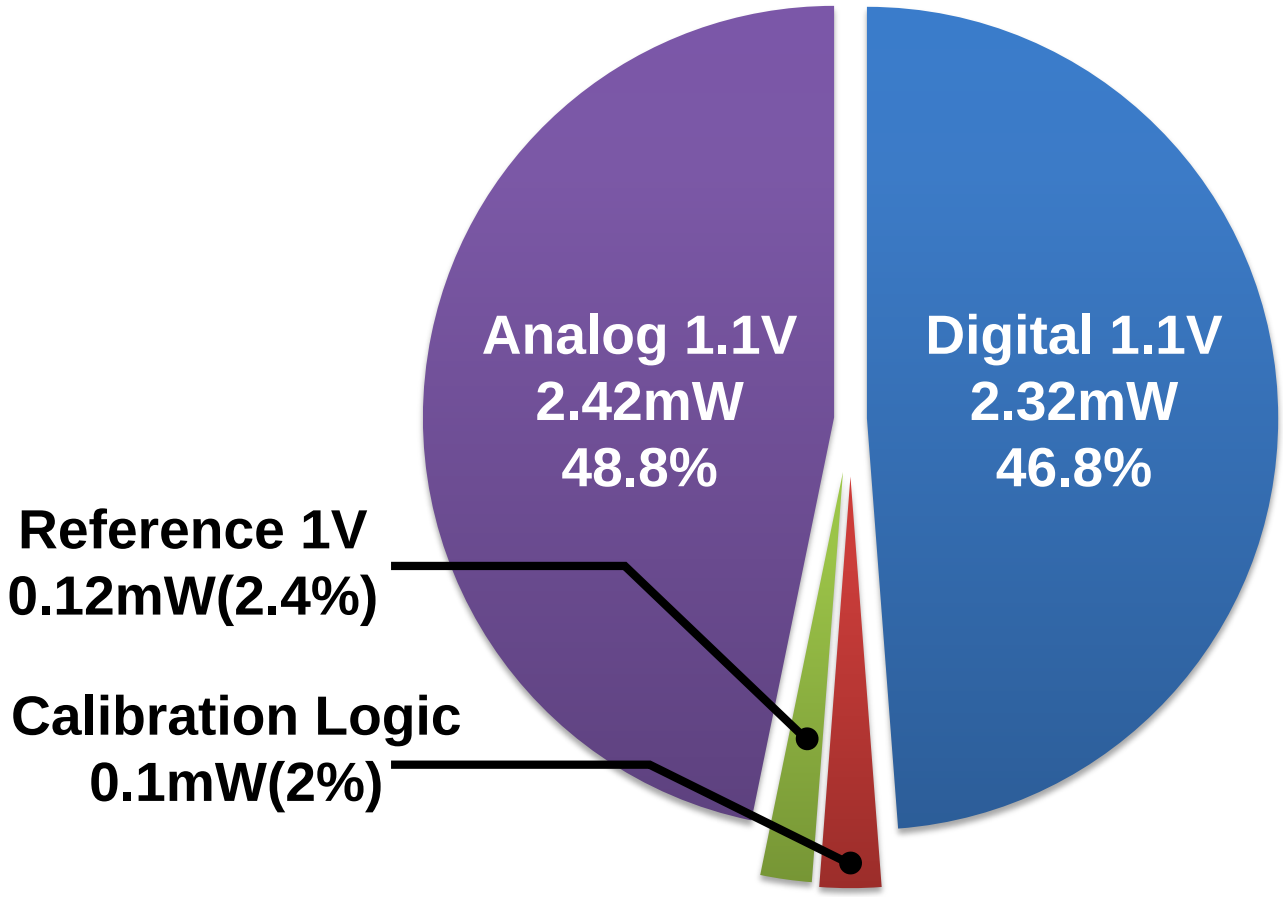
Measured ADC Dynamic Performance





Power Consumption

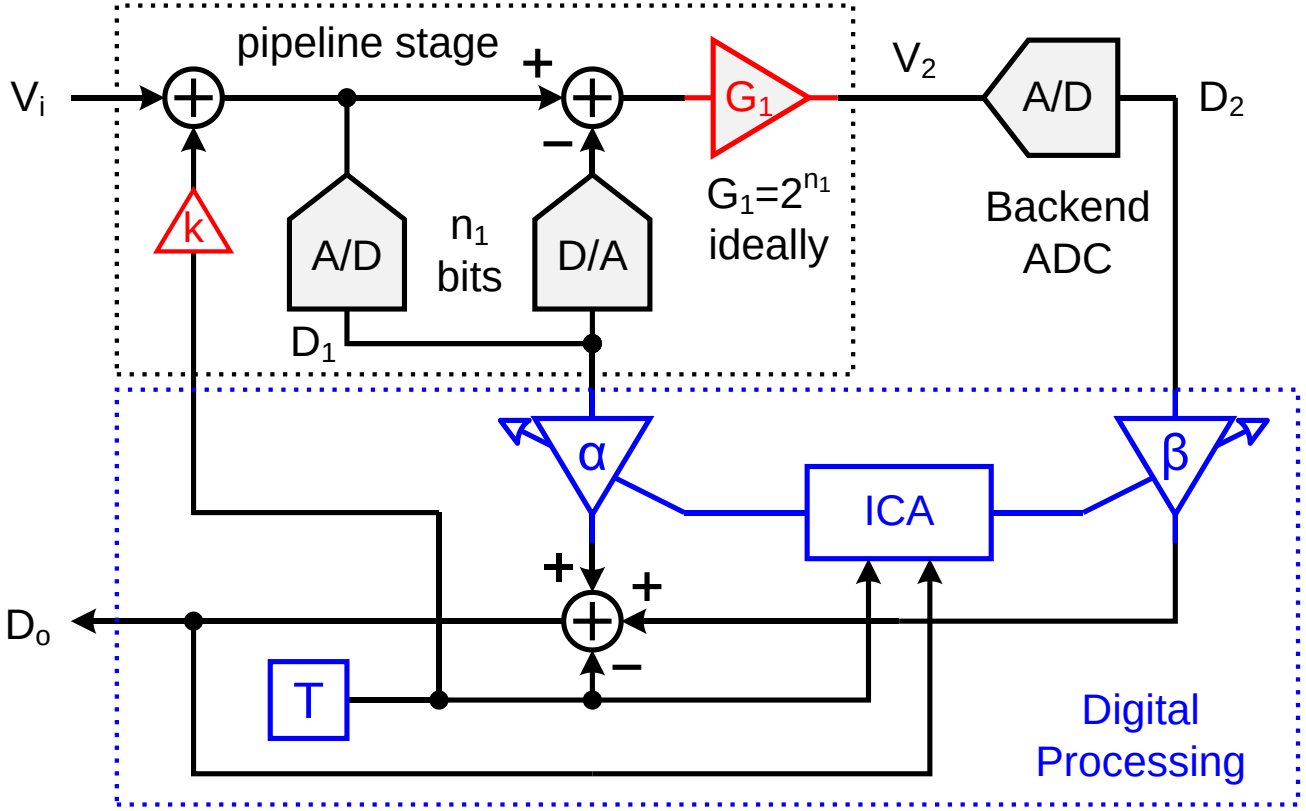
Ref. [16]



- **Total power is 4.96mW at 160MS/s operation.**

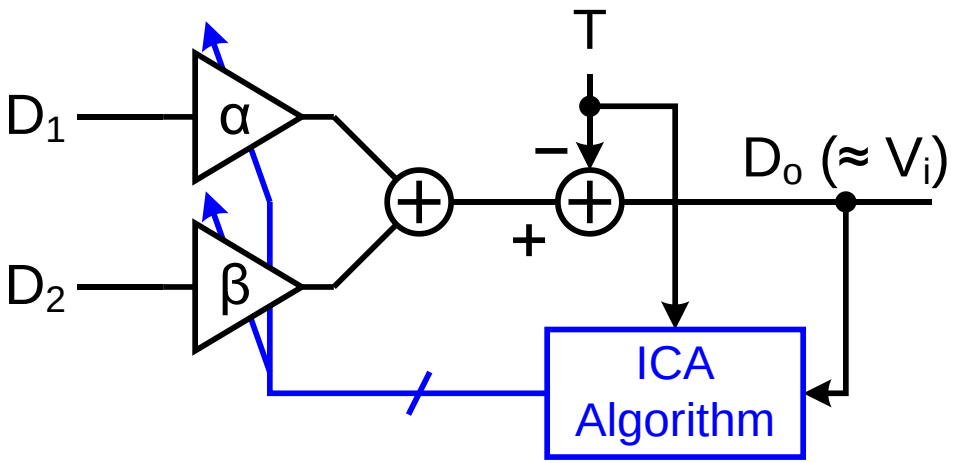
PRBS Test-Signal Injection (Input Injection)

Direct Input Injection



- Algorithm works reliant on the independence b/t input and T.
- Multi-parameter extraction is possible by Independent Component Analysis.

Independent Component Analysis (ICA)



$$D_o = \alpha \cdot D_1 + \beta \cdot D_2 - k \cdot T$$

Only two parameters need to be identified.

Hérault-Jutten (HJ) stochastic de-correlation:

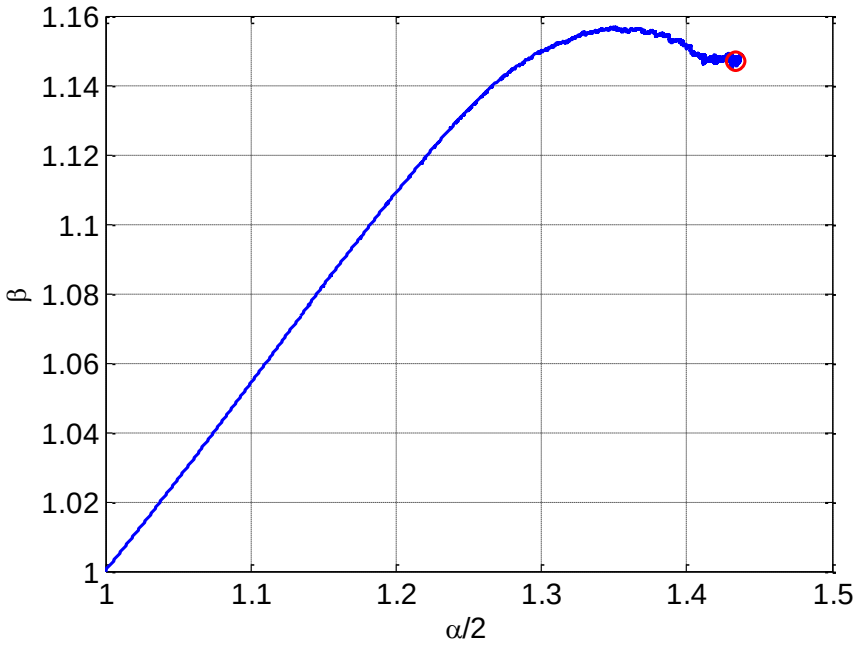
$$\alpha_{n+1} = \alpha_n - \mu_\alpha \cdot g_1(D_o) \cdot g_2(T)$$
$$\beta_{n+1} = \beta_n - \mu_\beta \cdot g_2(D_o) \cdot g_1(T)$$

In our simulation, we picked $g_1(x) = x$ and $g_2(x) = x^3$.

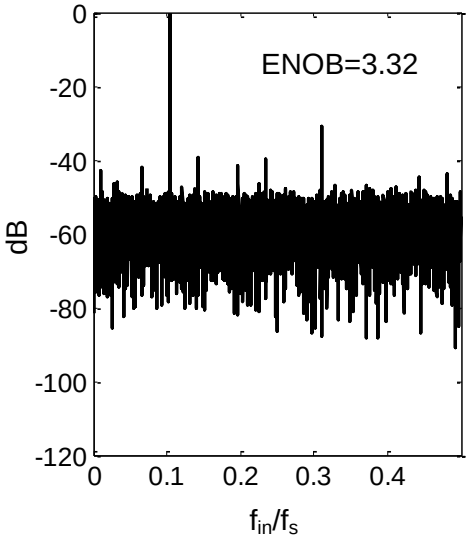
Ref. [17]

Simulation Results

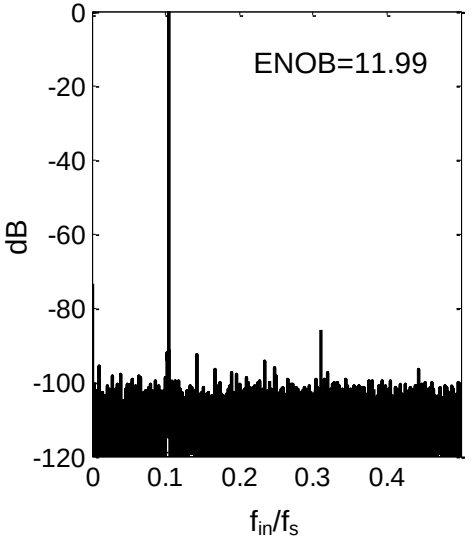
H-J Learning Trajectory



Before

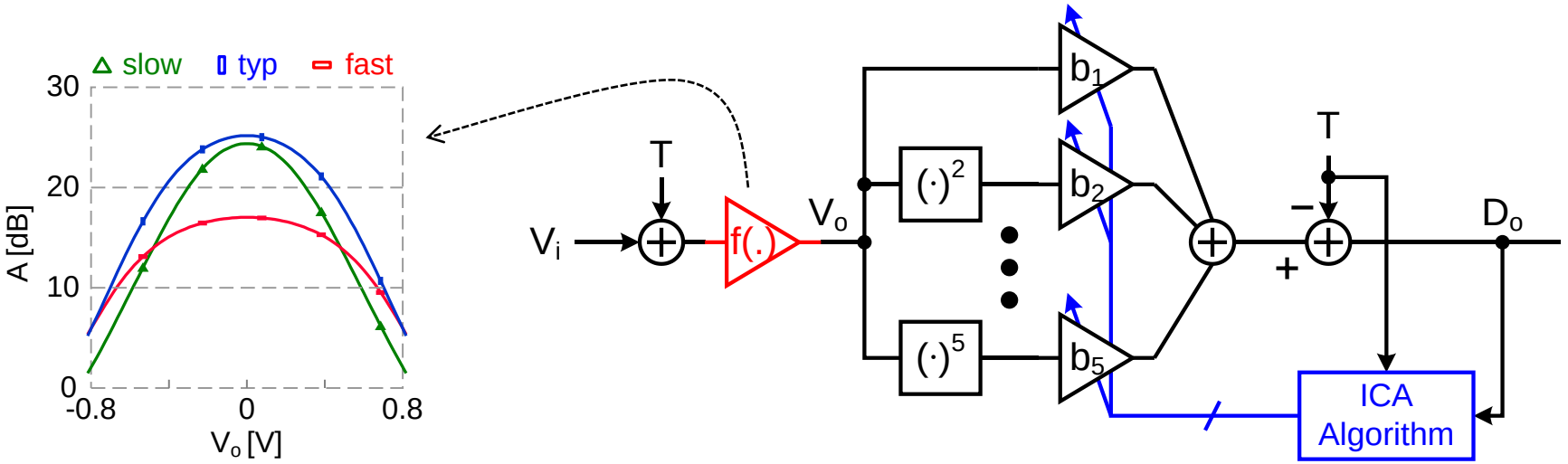


After



- Typical learning pattern of the H-J algorithm
- Steady-state coefficient fluctuation causing low-level spurs

ICA extended to nonlinear treatment



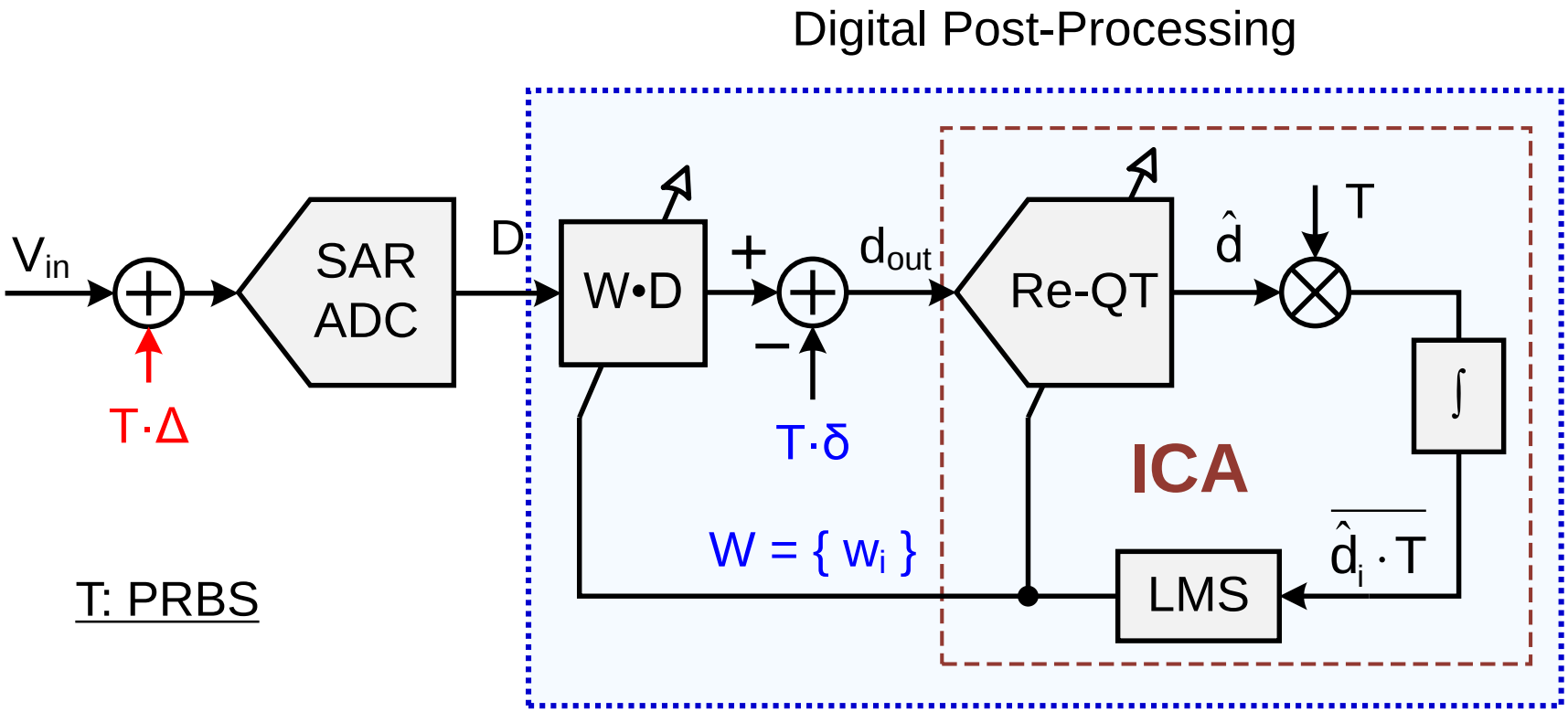
Error model: $V_o = f(V_i) \approx a_0 + a_1 V_i + a_2 V_i^2 + a_3 V_i^3 + \ldots$

Update equations:

$$b_j(n+1) = b_j(n) - \mu_j \cdot D_o{}^j(n) \cdot T(n) \qquad j = 1,\ldots,5$$

Ref. [17]

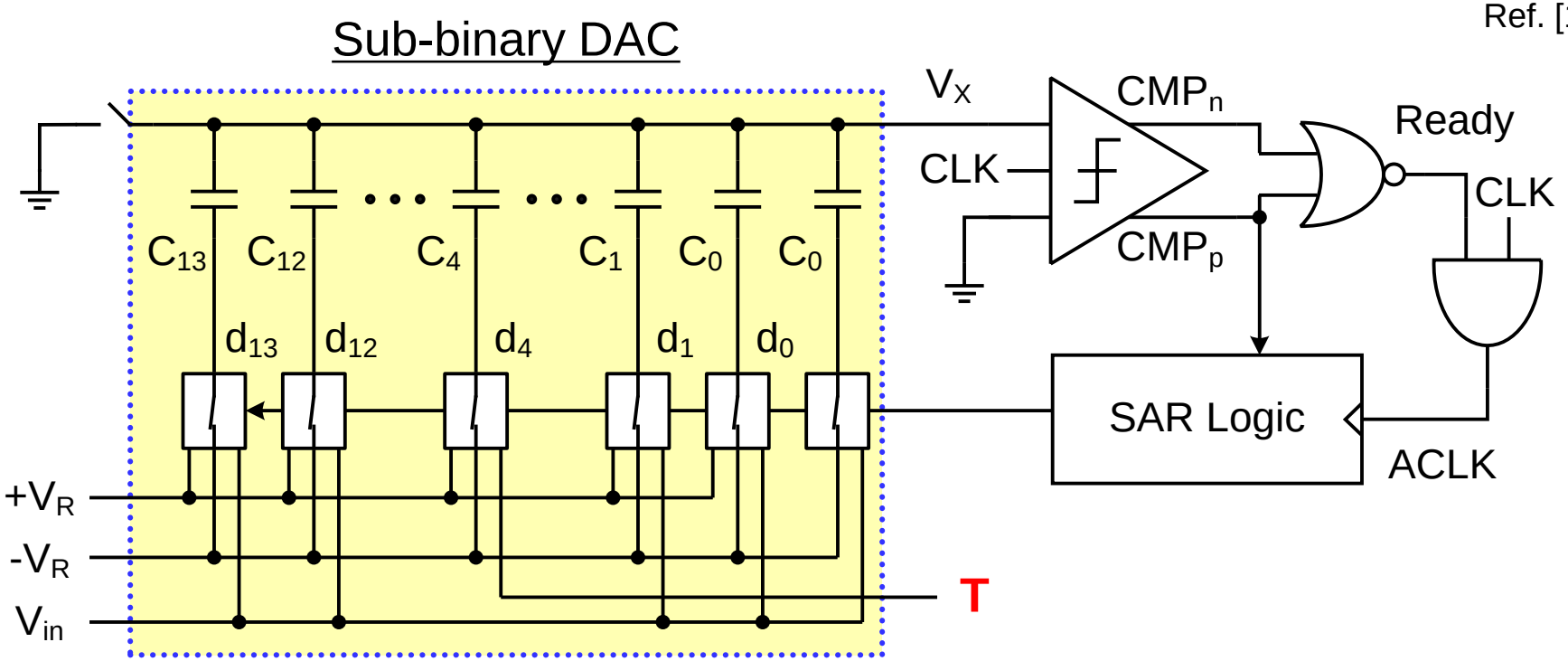
An ICA Approach for SAR Calibration



- ICA recovers 2X speed at the cost of slower convergence.
- ALL bit weights $\{w_j\}$ are learned simultaneously!

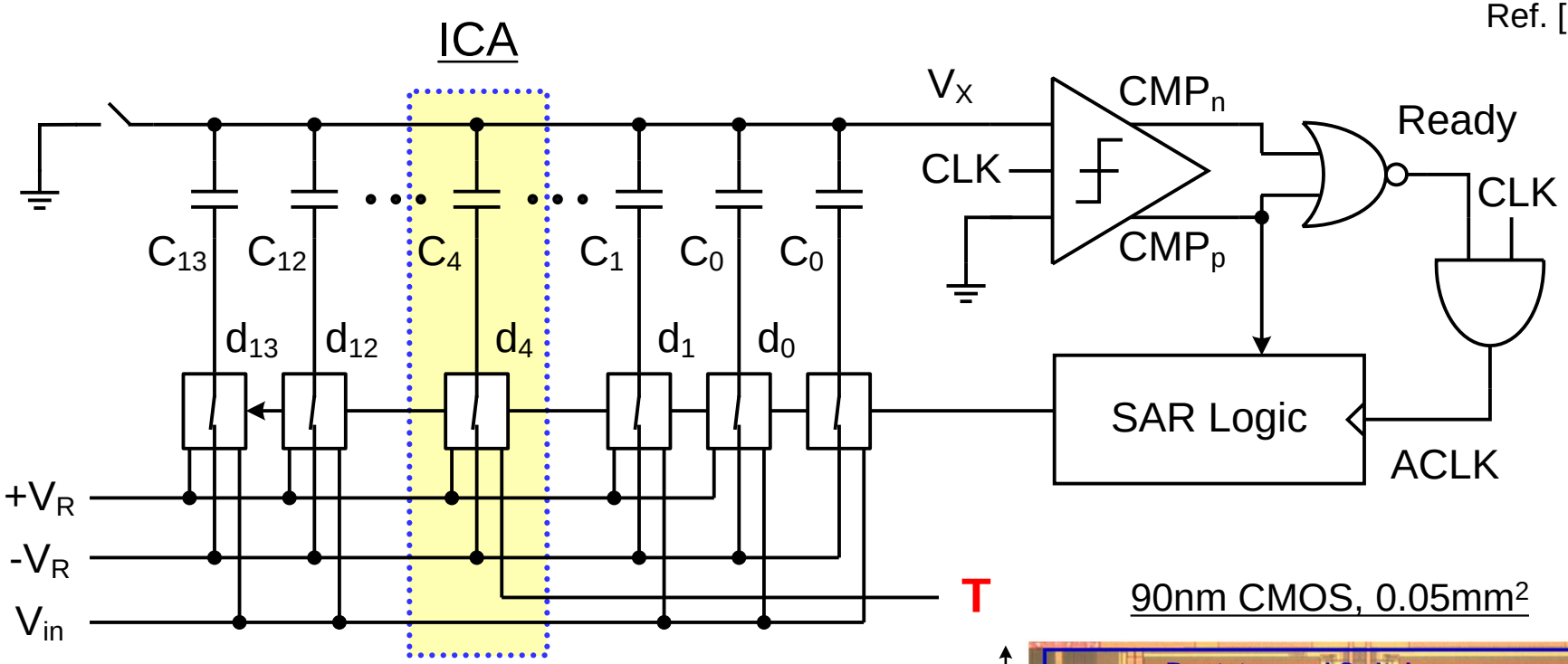
Prototype SAR ADC w/ ICA

Ref. [18]

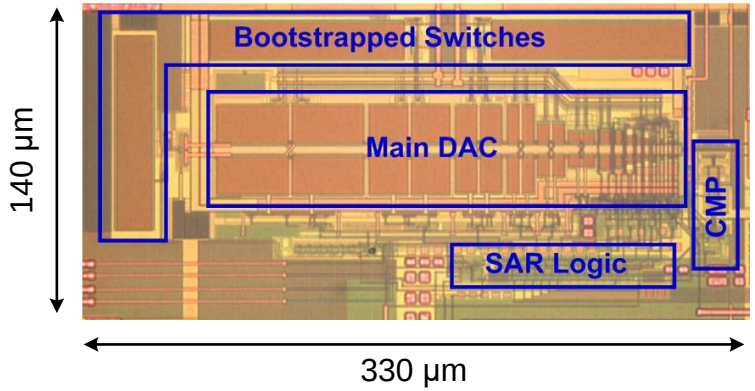


Prototype SAR ADC w/ ICA

Ref. [18]



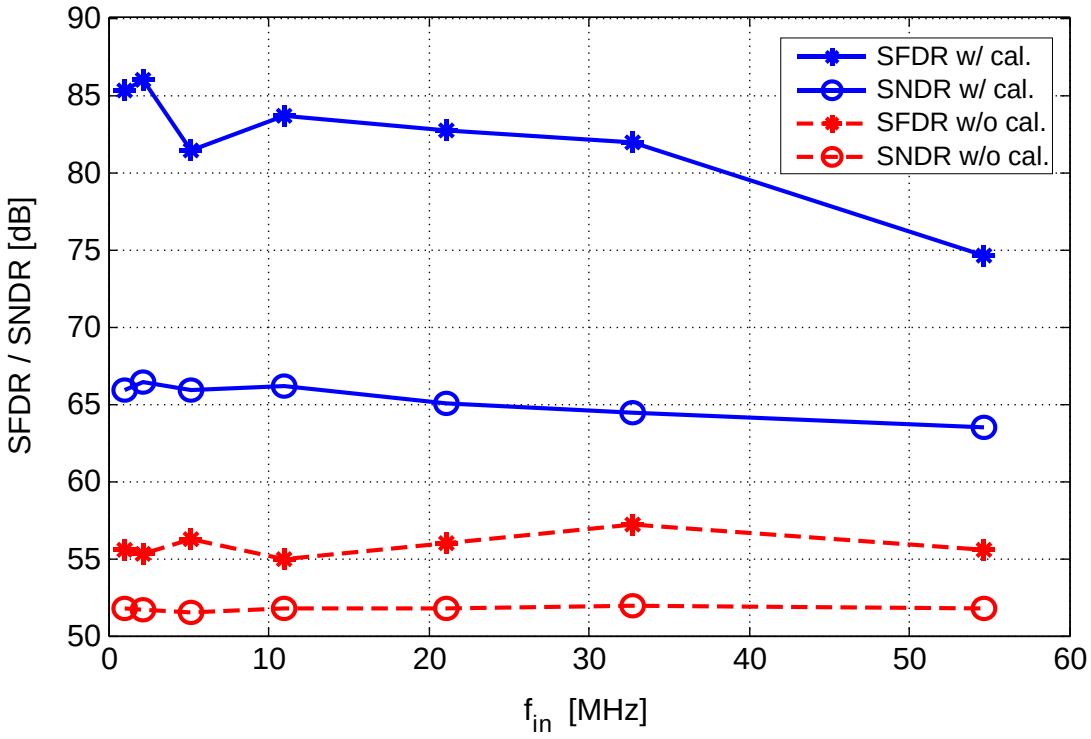
90nm CMOS, 0.05mm^2

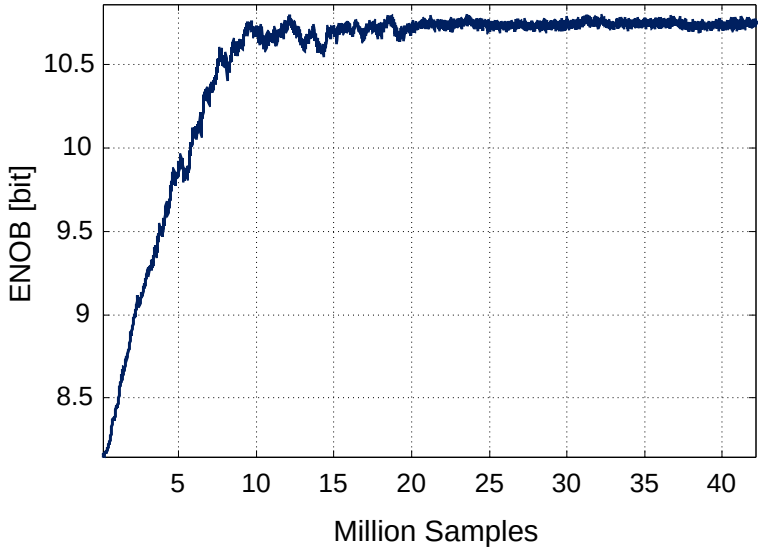


- 12b, 50MS/s in BG mode
- (3.3+1.4)mW power (45fJ/step)
- **CICC Best Paper Award**

Measurement Results

Dynamic Performance





Learning Curve

- Gear shifting helps stabilize the steady-state fluctuations.

Two-ADC Equalization

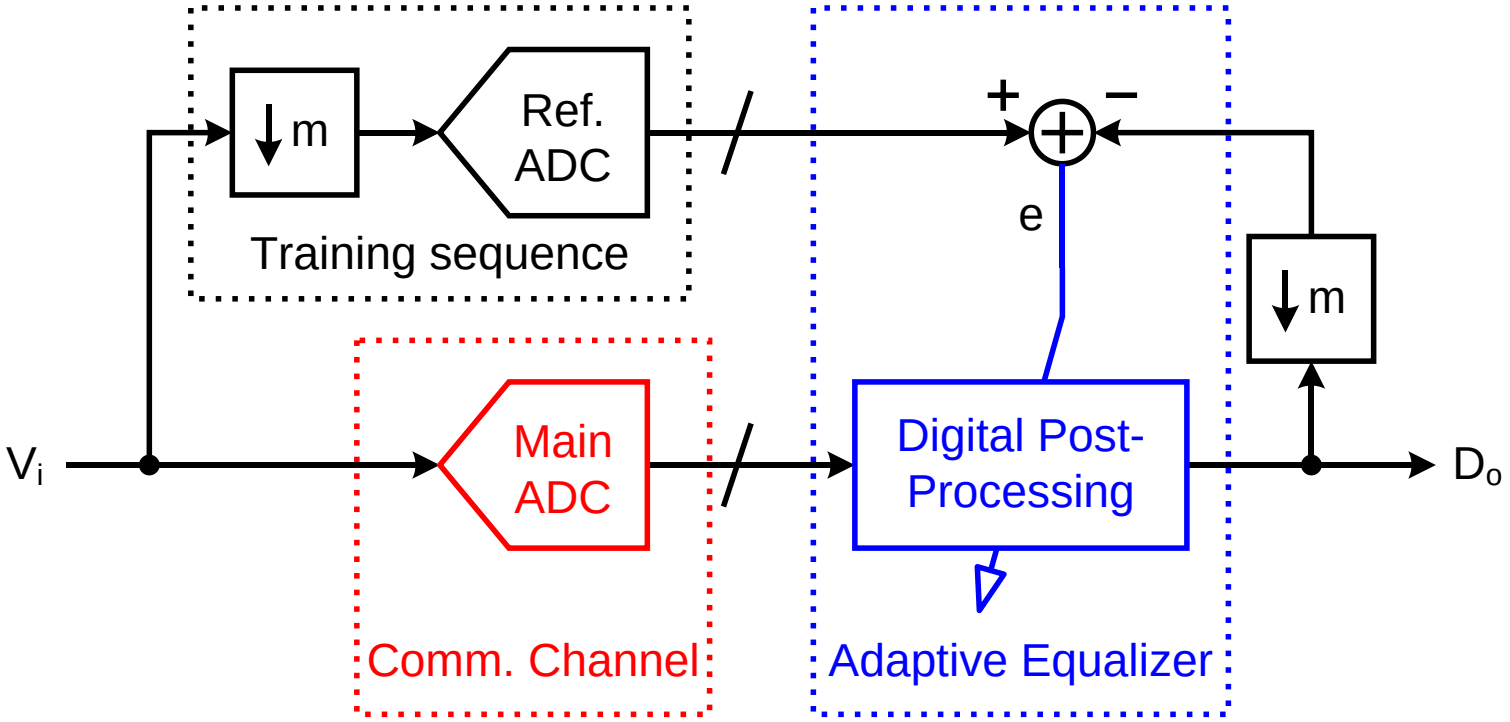
Two-ADC Equalization Techniques

- **Reference-ADC equalization**
  - Slow-Fast two-ADC architecture to accomplish accuracy and throughput simultaneously using adaptive equalization
  - Two (different) ADC's needed, subject to skew error without SHA
- **Split-ADC equalization**
  - Two almost identical ADC's employed for blind equalization
  - Two ADC's needed, subject to skew error without SHA
- **Offset double conversion (ODC)**
  - Self-equalization by digitizing every sample twice with opposite DC offsets injected to the input
  - Single ADC with modified timing in background mode
  - Conversion throughput halved in background mode

Two-ADC Equalization (Reference-ADC)

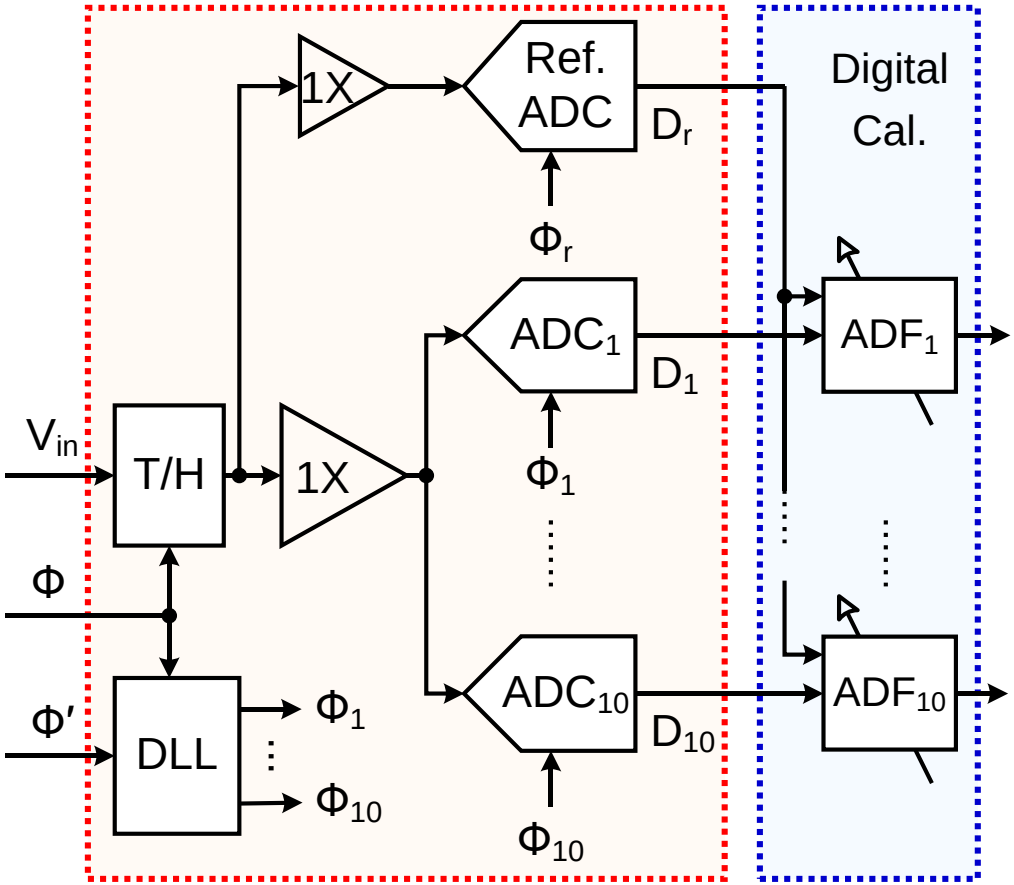
Reference-ADC Equalization

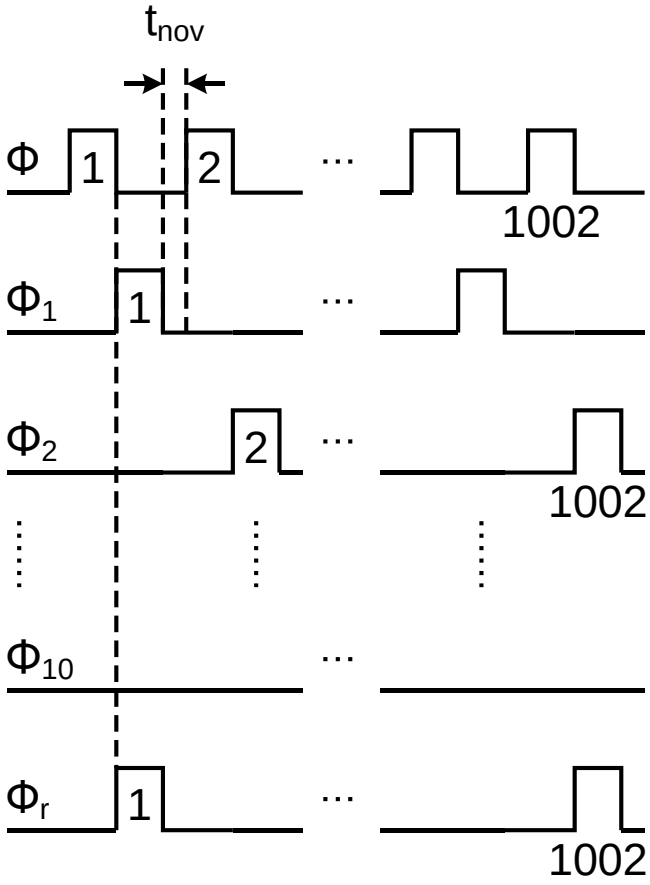
Ref. [11,12]



- Concept inspired by adaptive equalization in digital comm. receivers
- Divide-and-conquer approach to achieve analog speed and accuracy

EQZ of Time-Interleaved ADC Array

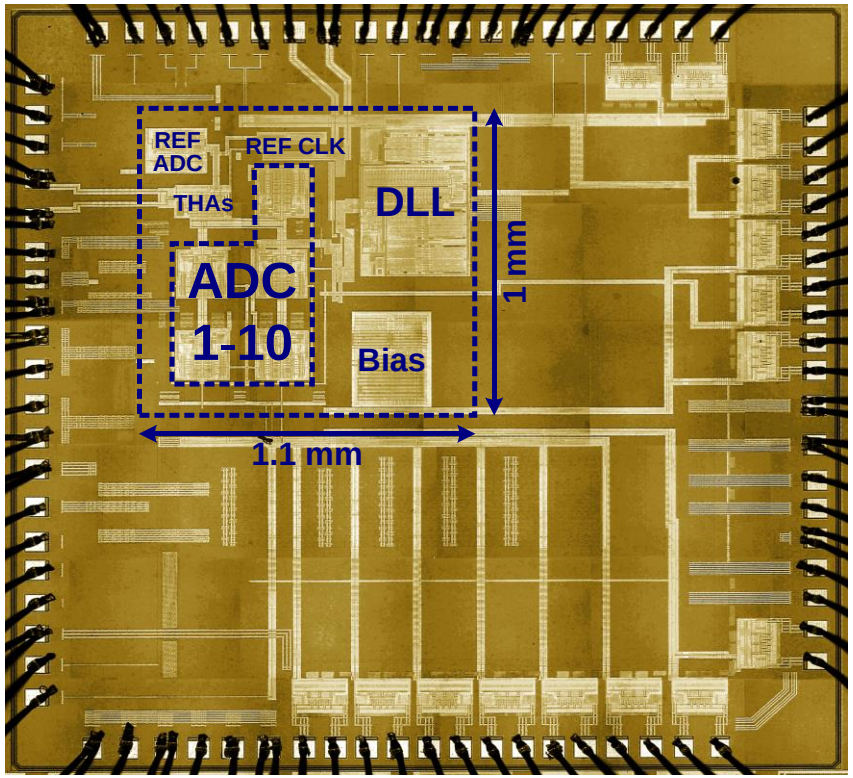




- ALL paths are aligned to the unique ref. ADC after equalization.

Prototype 10-way TI-ADC Array

Die photo



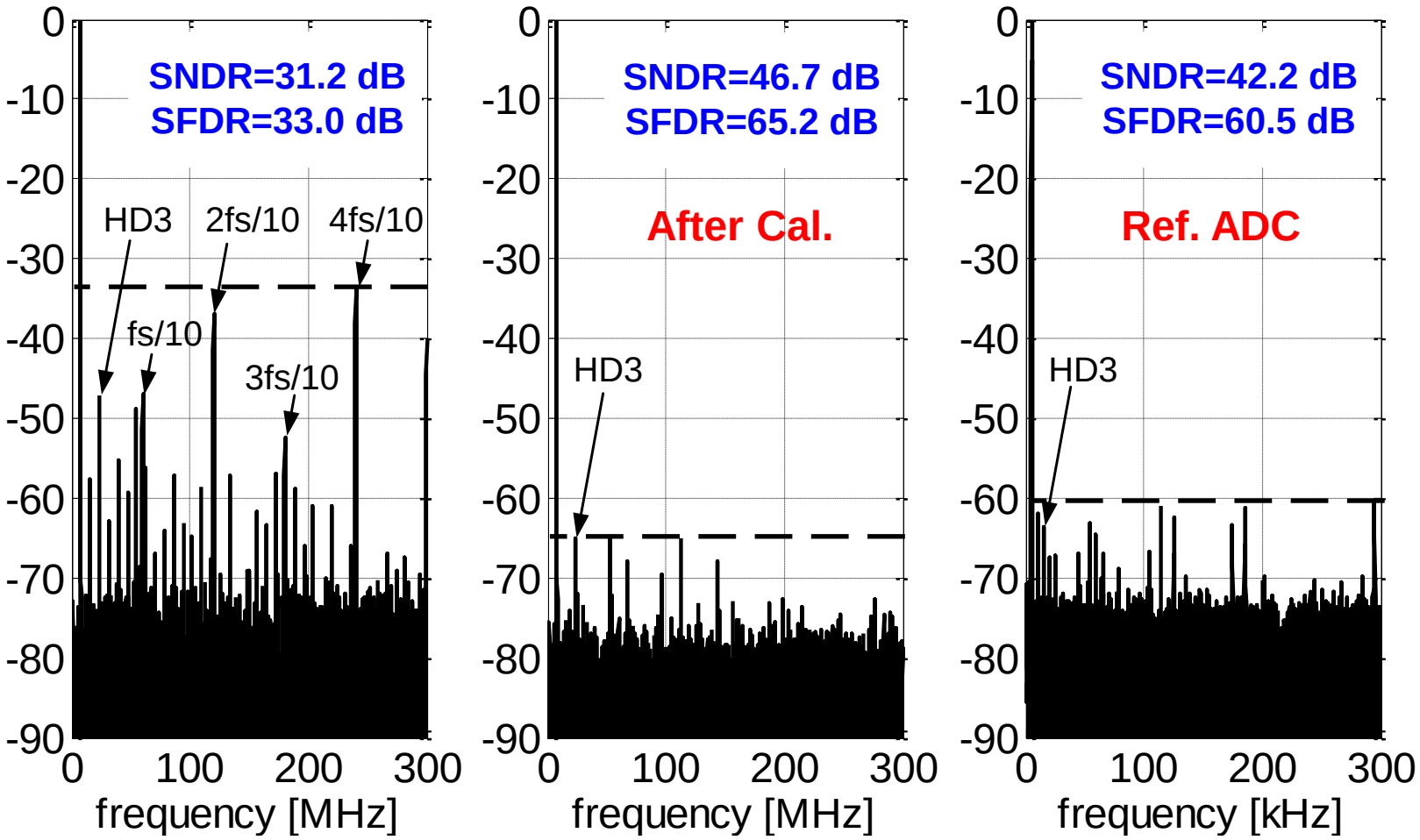
Performance Comparison
(@ time of publication)

Time	CMOS Process	Speed [MS/s]	SFDR [dB]	FoM [fJ/step]
ISSCC06	0.13µm	600	43	220
ISSCC08	0.13µm	1250	48	480
VLSI08	65nm	800	58	280
ISSCC09	**0.13µm**	**600**	**65**	**210**

Ref. [5]

- **The 2009 DAC/ISSCC Student Design Contest Award**

ADC Array EQZ – Measured Spectra

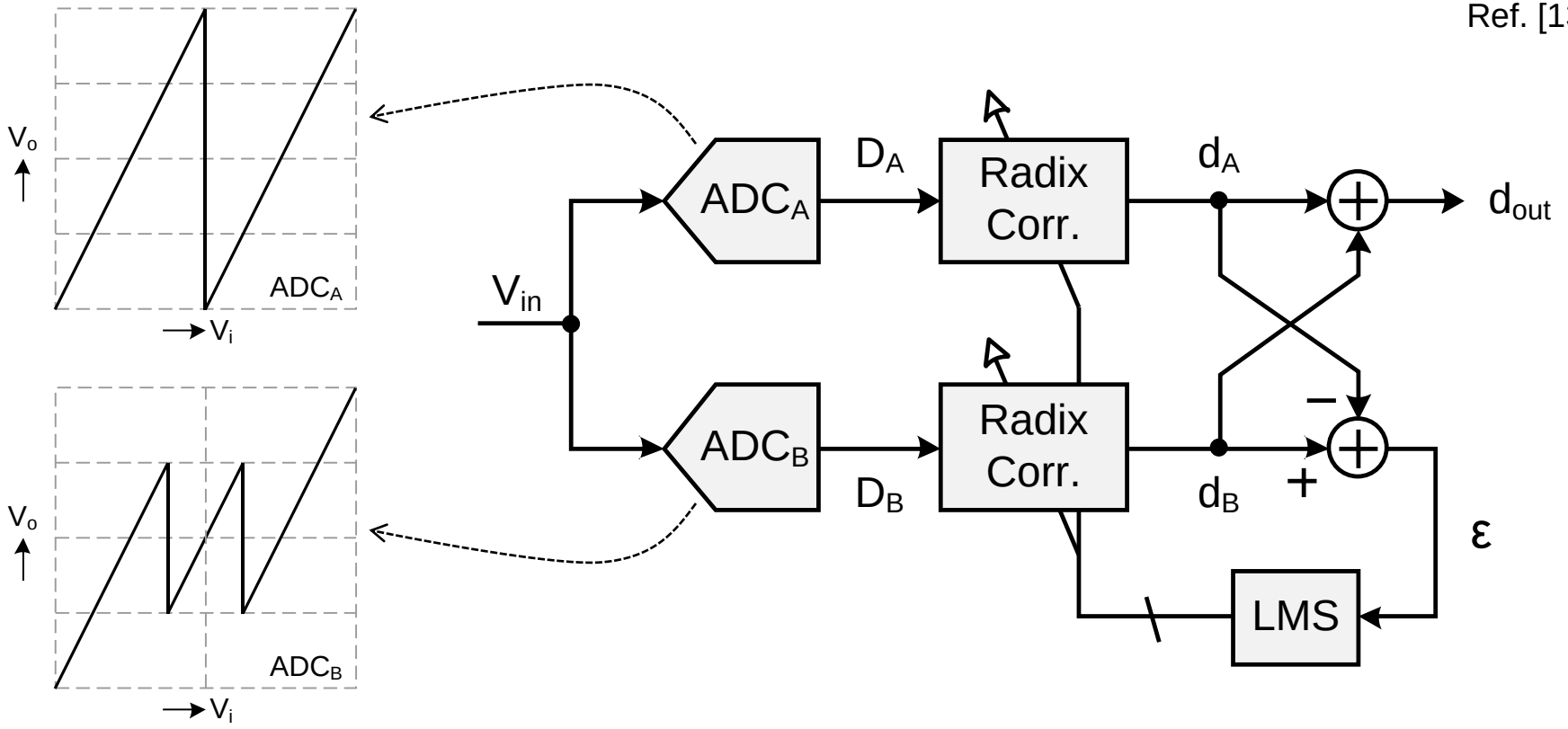


(f_s = 600MS/s, f_{in} = 7.8MHz, A_{in} = 0.9FS, 16k samples)

Two-ADC Equalization (Split-ADC)

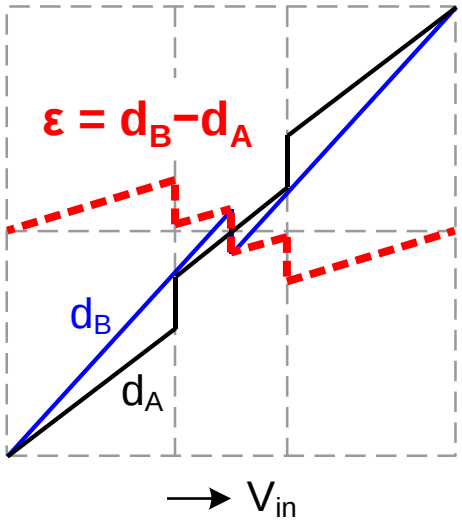
Split-ADC Equalization

Ref. [13]

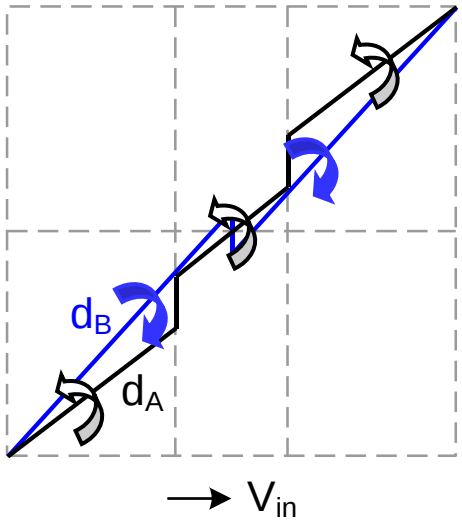


- Blind equalization w/o reference possible by offsetting the RTFs
- Fast convergence due to zero-forcing equalization

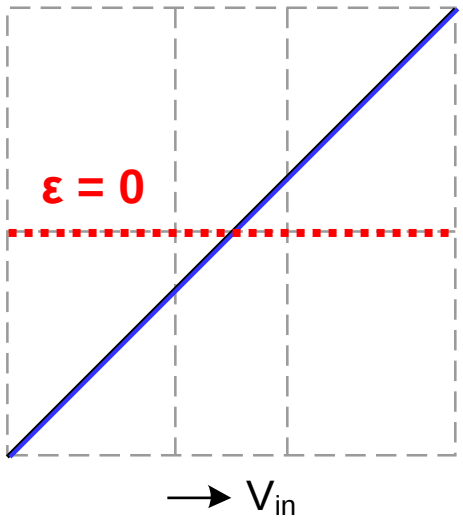
Zero Forcing



Error observation



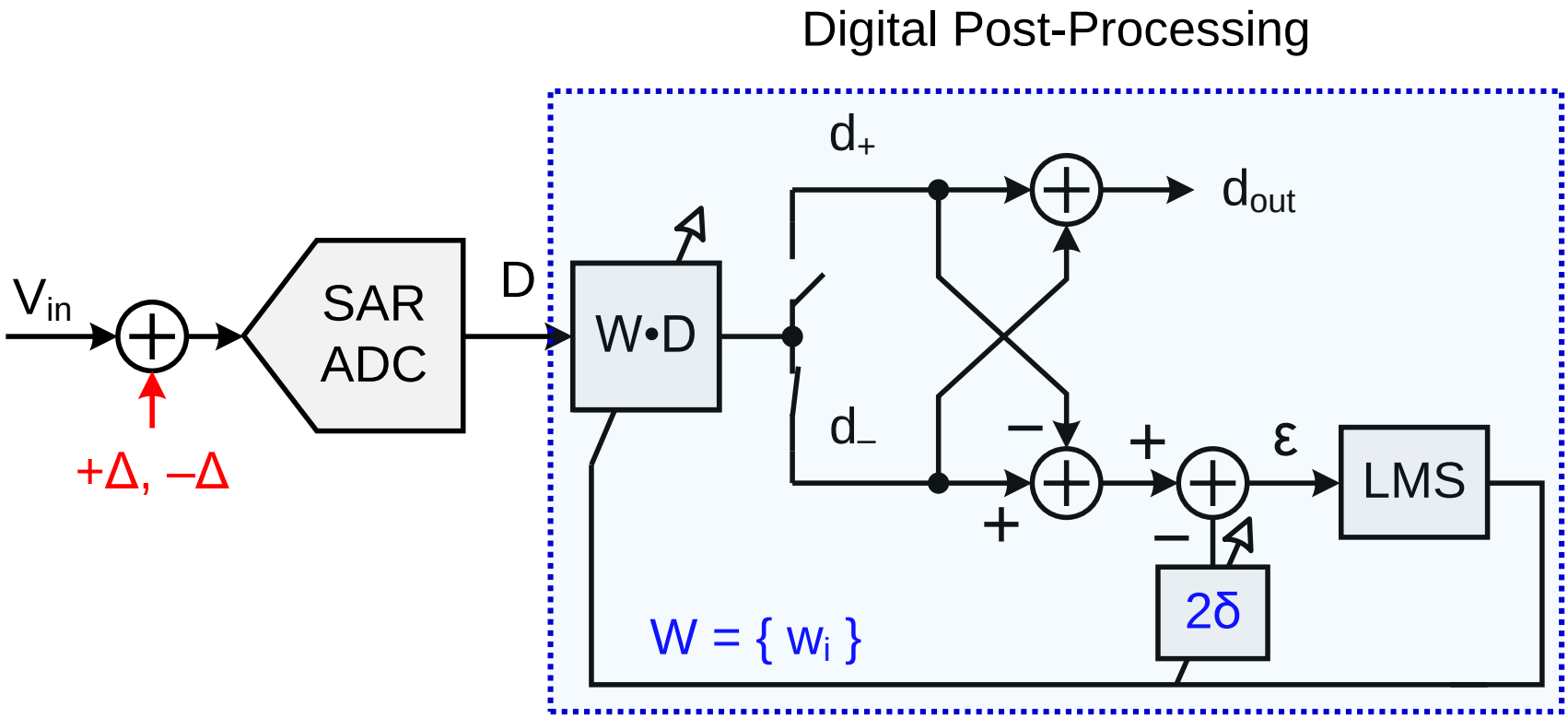
Radix correction



Zero-forcing EQZ

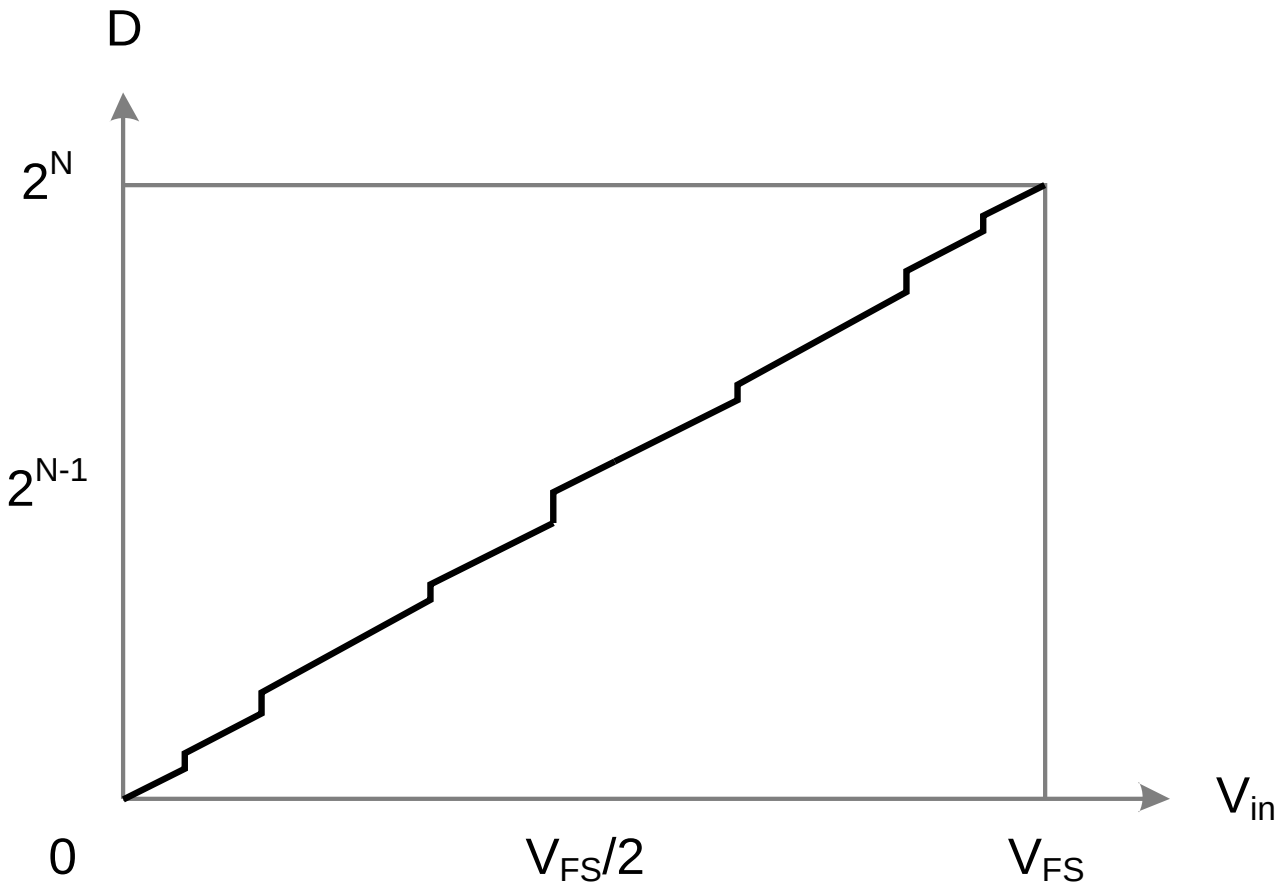
Two-ADC Equalization (Offset Double Conversion)

Offset Double Conversion (ODC) for SAR



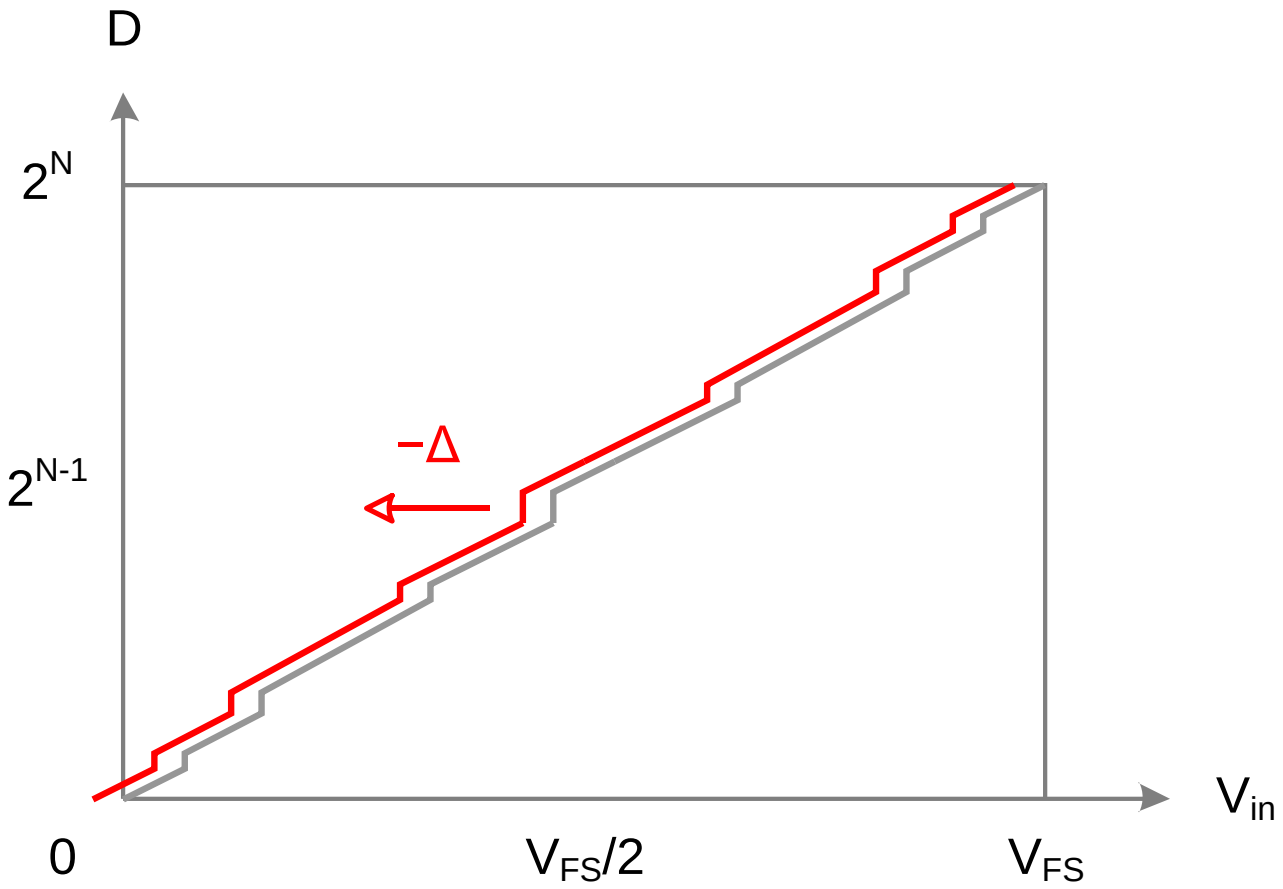
- ODC enables zero-forcing self-equalization.
- ALL bit weights {w_j} are learned simultaneously!

How to determine Bit Weights?



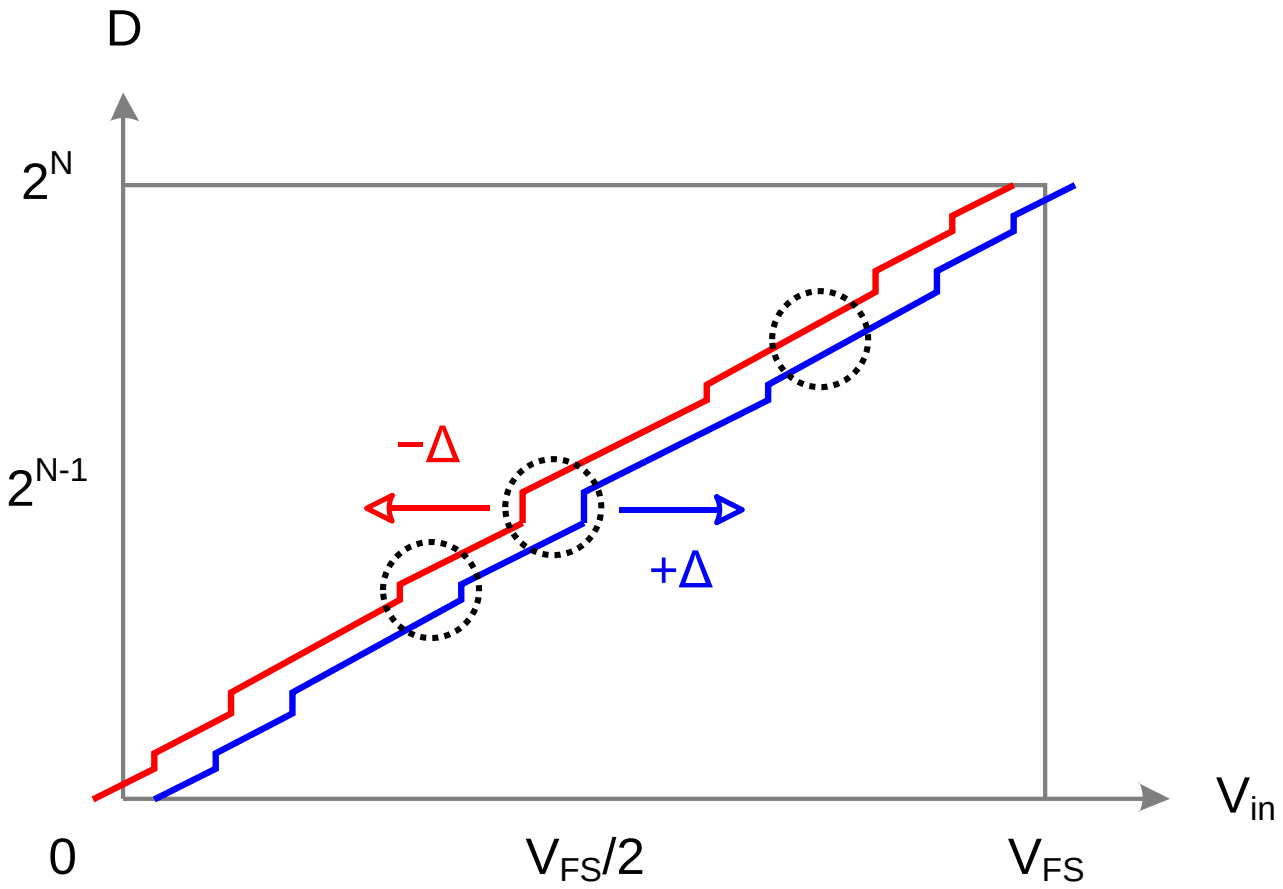
Is the transfer curve shift-invariant?

How to determine Bit Weights?



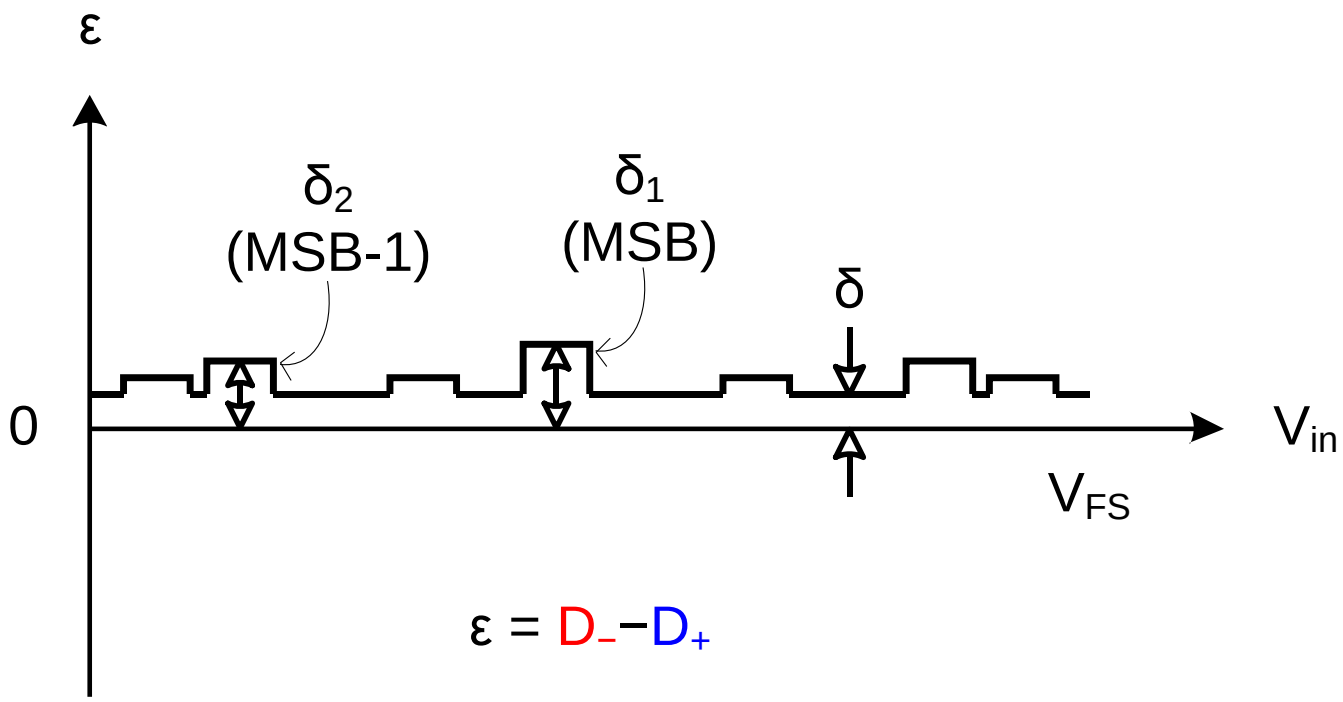
Is the transfer curve shift-invariant?

How to determine Bit Weights?



Is the transfer curve shift-invariant?

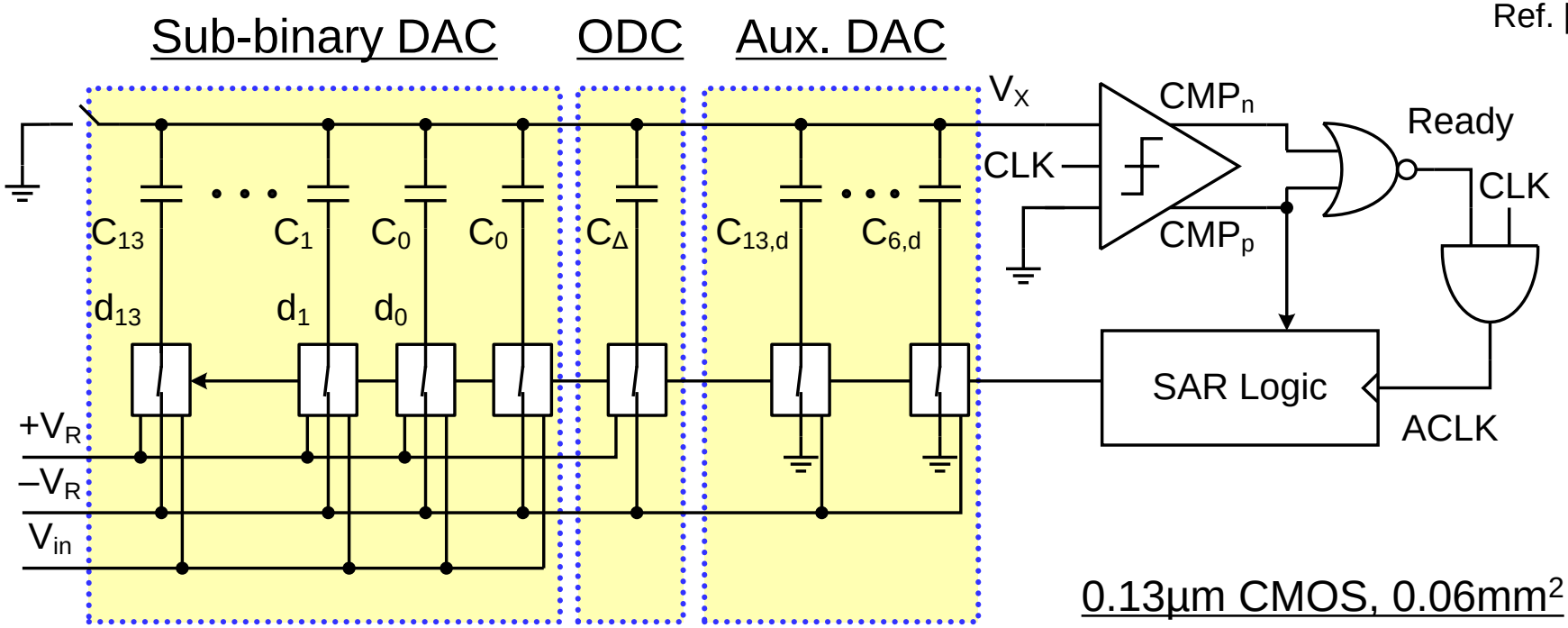
How to determine Bit Weights?



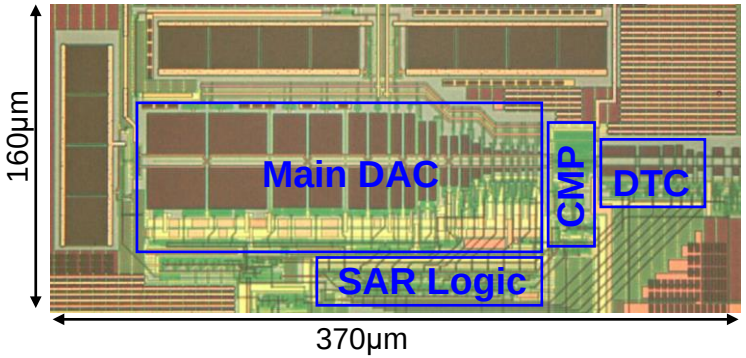
- Shift-invariant <u>ONLY</u> when the transfer curve is completely linear!
- Non-constant difference b/t D_+ and D_- reveals bit weight information.

Prototype SAR ADC w/ ODC

Ref. [19]

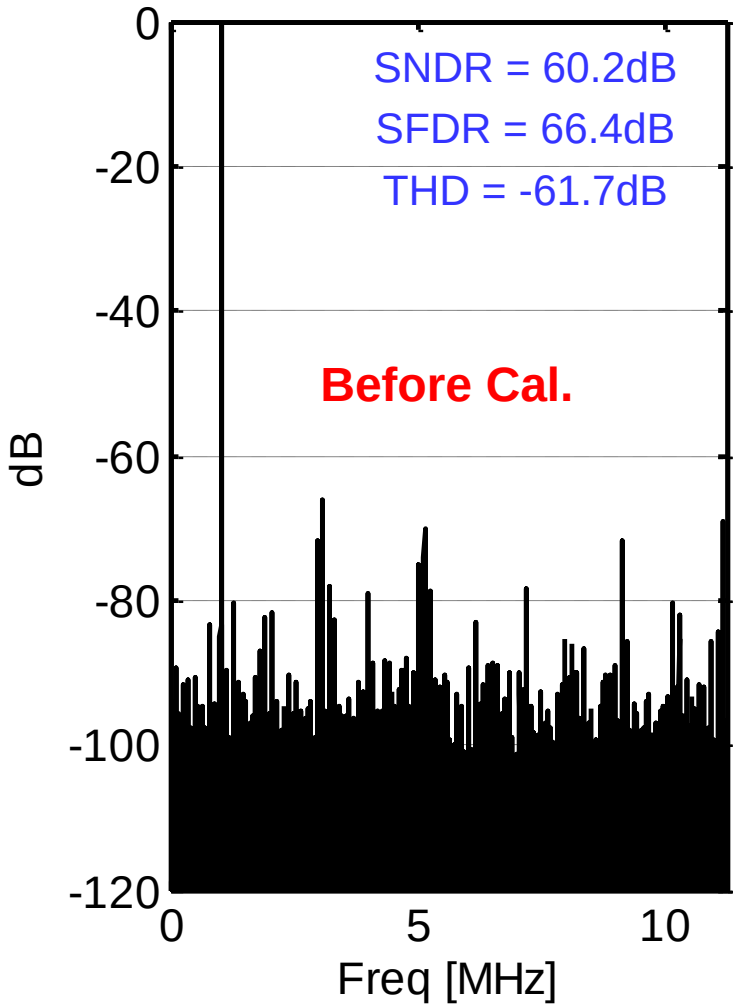


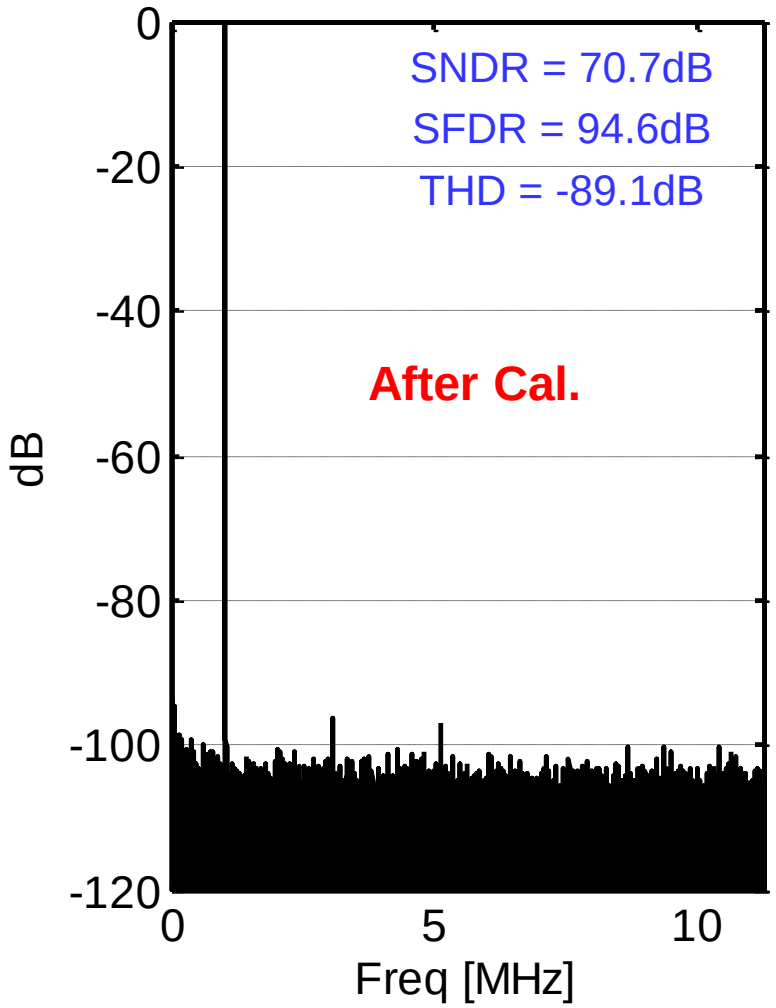
0.13µm CMOS, 0.06mm^2



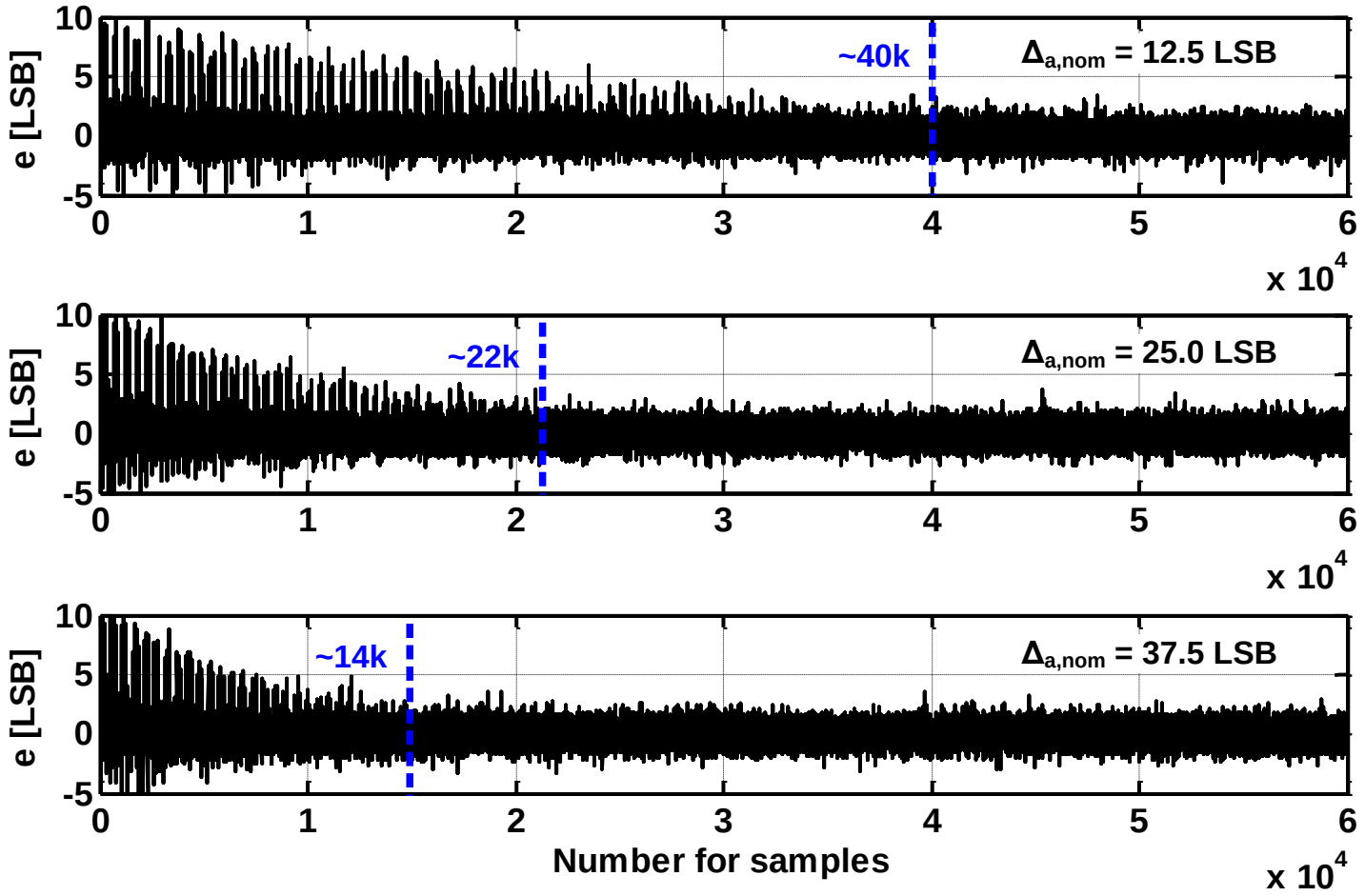
- 12b, 45MS/s in FG mode
- 3mW power (36.3 fJ/step)
- **Most read JSSC article Nov. 2011**

Measurement ADC Spectra (BG Mode)



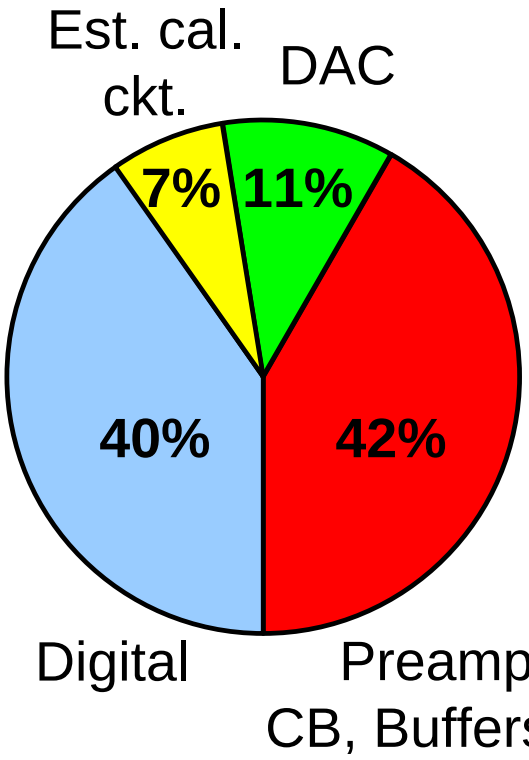


Convergence Time (BG Mode)



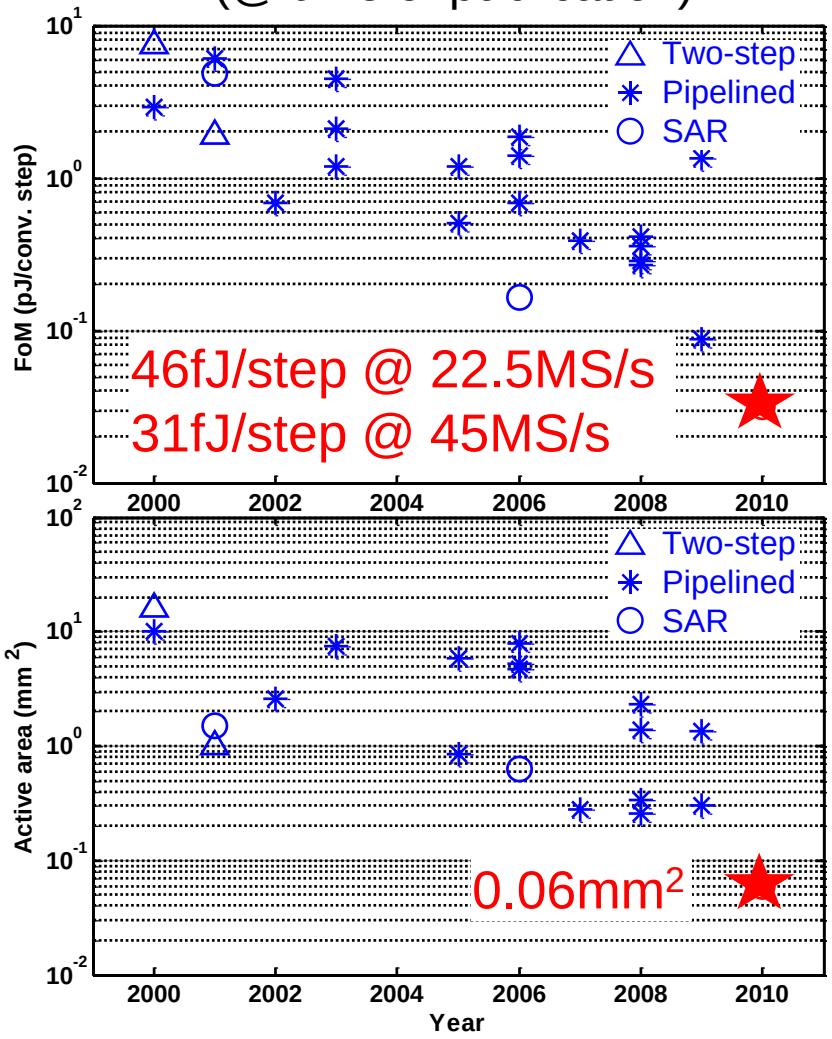
22000 samples @ 22.5MS/s ≈ 1ms

Comparison with 12b ADCs



Total Power: 3.0mW

(@ time of publication)



Summary of Dig. BG Cal. Techniques

Method	Parameter	Test signal	Injection point	Reference†
DNC + GEC	$\{ \beta_{j,k}, \alpha_{j,m} \}$	multi PRBS	sub-DAC	[9,10,20-24]
Split capacitor	$\{ \Delta_j \}$	1 PRBS	sub-DAC	[25,26]
Sig.-dep. dither	$\{ \gamma_j \}$	1 PRBS	sub-DAC	[15,16]
GEC + SA	$\{ \gamma_j \}$	2 PRBS	sub-ADC	[27,28]
Statistics	$\{ \alpha_{j,m} \}$	1 PRBS	sub-ADC	[29,30]
Fast GEC	$\{ \gamma_j \}$	1 PRBS	sub-ADC	[31]
ICA	$\{ \gamma_j \}, \{ \alpha_{j,m} \}$	1 PRBS	input	[17,18,32,33]
Ref. ADC	$\{ \beta_{j,k}, \alpha_{j,m} \}$	n/a	n/a	[5,11,12,34-36]
Virtual ADC	$\{ \beta_{j,k}, \alpha_{j,m} \}$	offset	sub-DAC	[37,38]
Split ADC	$\{ \alpha_{j,m} \}$	n/a	n/a	[13,39]
	$\{ \beta_{j,k}, \alpha_{j,m} \}$	n/a	n/a	[40]
ODC	$\{ \gamma_j \}$	offset	input	[19]
	$\{ \beta_{j,k}, \alpha_{j,m} \}$	offset	input	[41]

† References are furnished at the end of the slides.

Presentation Outline

- Principles of Multistep A/D Conversion
- Architectural Redundancy
- Error Mechanisms and Digital-Domain Calibration
- Error-Parameter Identification
  - PRBS Test-Signal Injection (sub-ADC, sub-DAC, input)
  - Two-ADC Equalization (ref.-ADC, split-ADC, ODC)
- Energy Efficiency and Trend
- Summary

ADC Figure-of-Merit (FoM)

Walden FoM: $$\mathrm{FoM_W} = \frac{P}{2 \cdot BW \cdot 2^{ENOB}} \left[\frac{\text{Joule}}{\text{Conversion-Step}}\right]$$

"Energy Efficiency"

Schreier FoM: $$\mathrm{FoM_S} = 10\log_{10}\left(\frac{2 \cdot BW \cdot 4^{ENOB}}{P}\right) \quad [\mathrm{dB}]$$

P: power consumption

BW: min{f_s/2, ERBW}

ENOB: effective number of bits

ERBW: effective resolution BW

- Walden FoM is intuitive but penalizes noise/matching-limited designs.
- Schreier FoM is more fair to high dynamic range designs.

Performance, Efficiency, and Power

$$\text{Performance} = 2 \cdot BW \cdot \alpha^{ENOB} \quad \left[Hz \cdot Step^{\alpha/2} \right]$$

Performance = Speed × SNR

$$\text{Energy Efficiency} = \frac{P}{2 \cdot BW \cdot \alpha^{ENOB}} \quad \left[\frac{J}{Step^{\alpha/2}} \right]$$

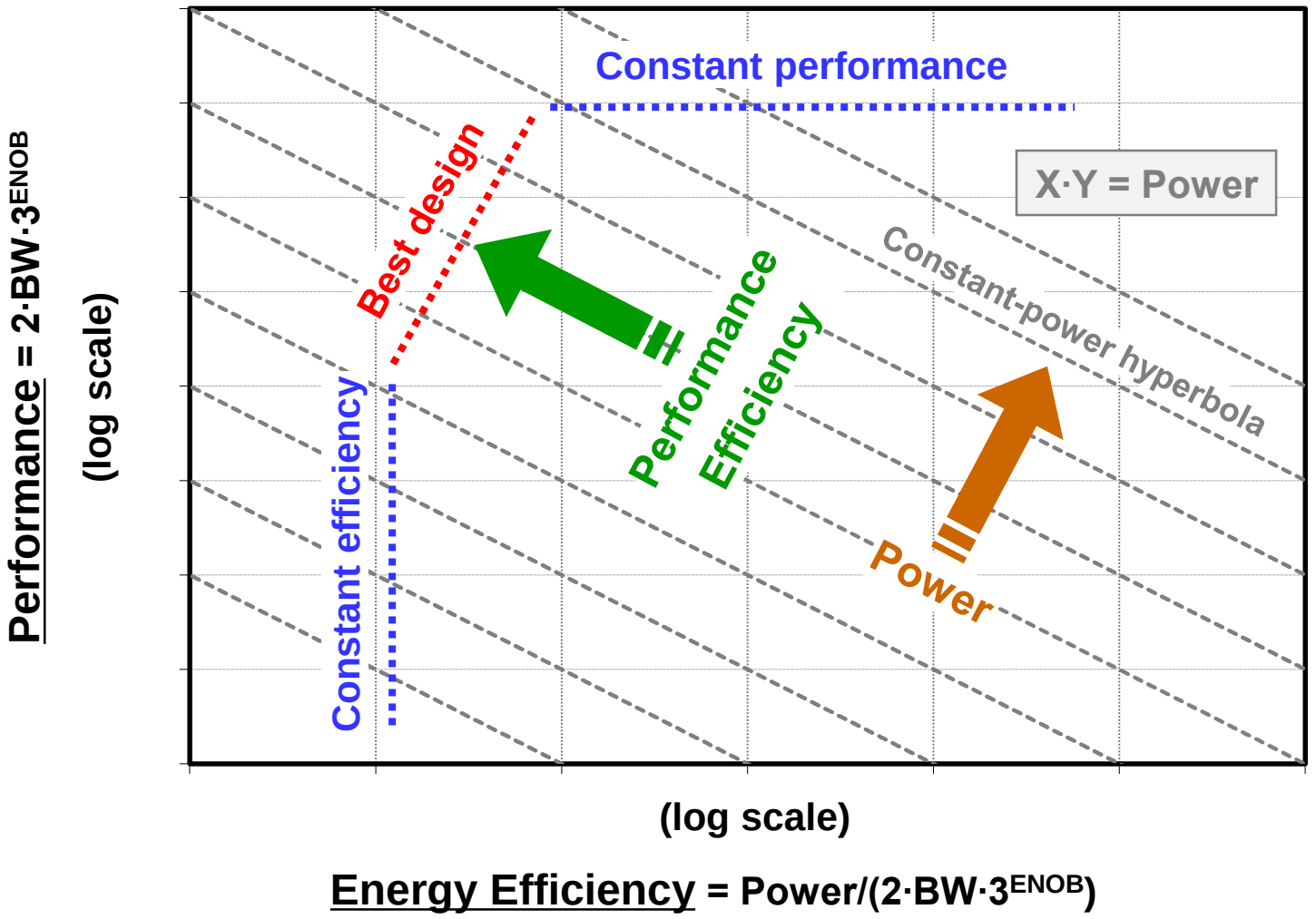
Note:

Performance × Efficiency

$$= 2 \cdot BW \cdot \alpha^{ENOB} \times \frac{P}{2 \cdot BW \cdot \alpha^{ENOB}} = \text{Power}$$

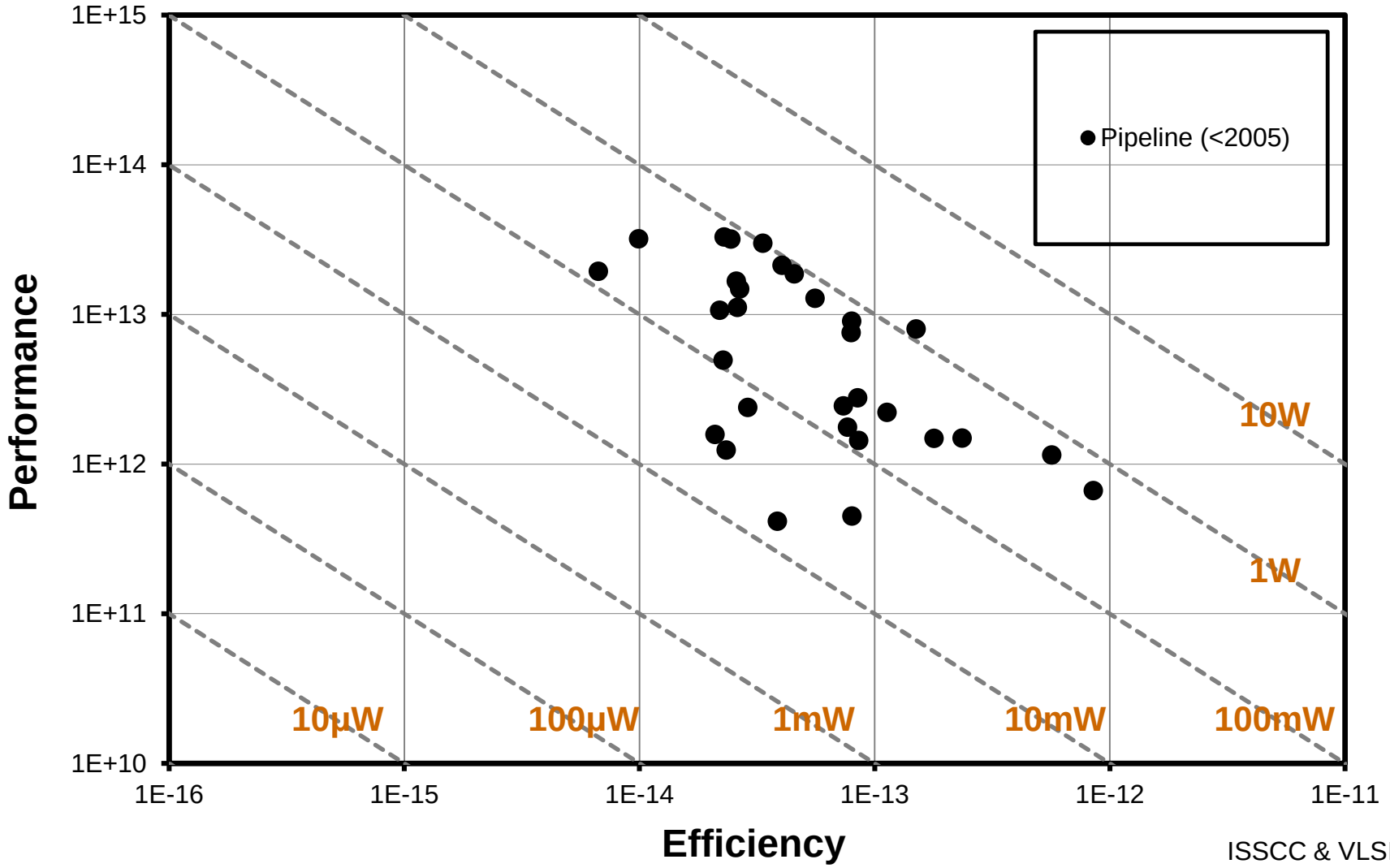
$\alpha = 2 - 4$

Performance–Efficiency (PE) Chart



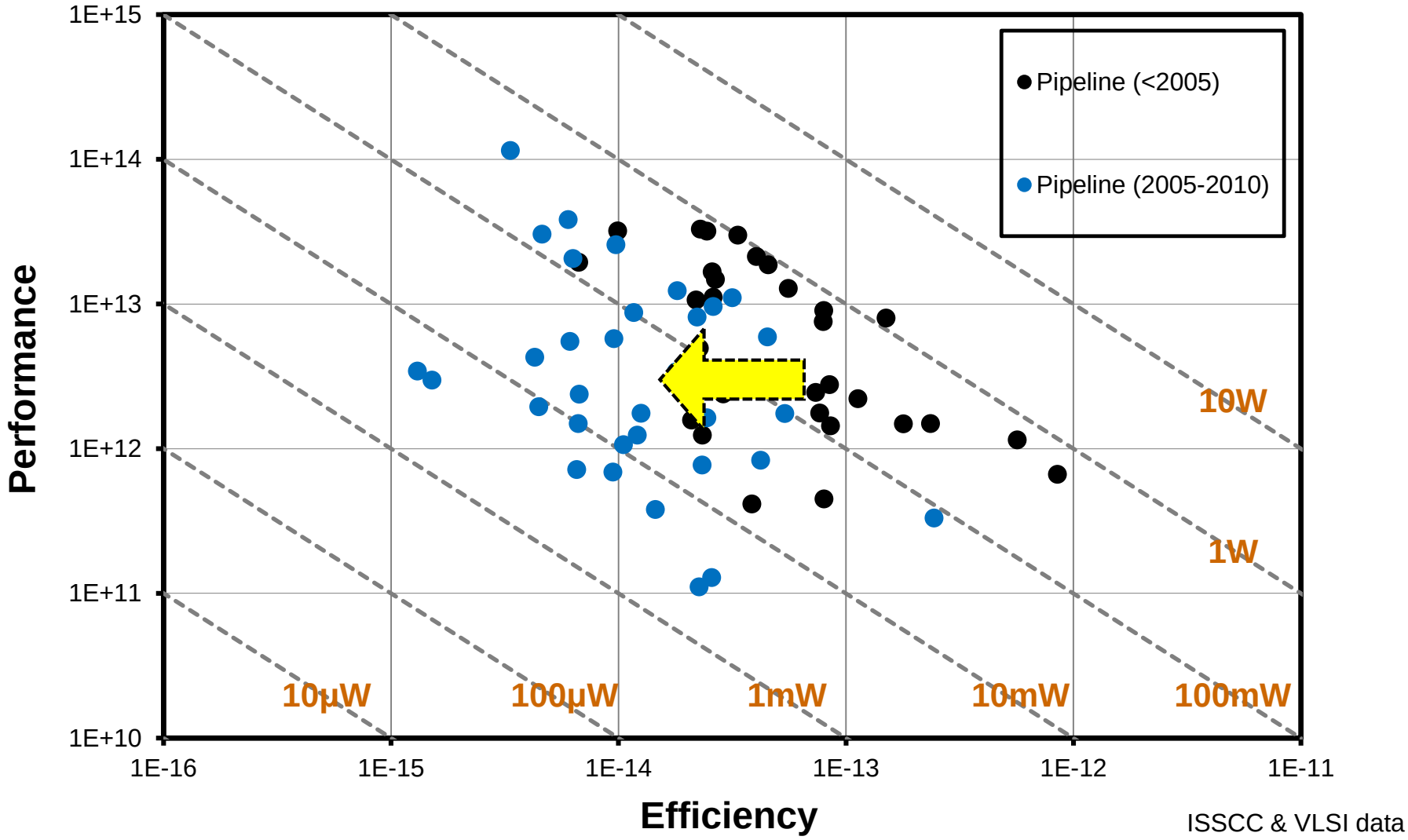
α = 3

PE Chart: Pipelined ADC (<2005)



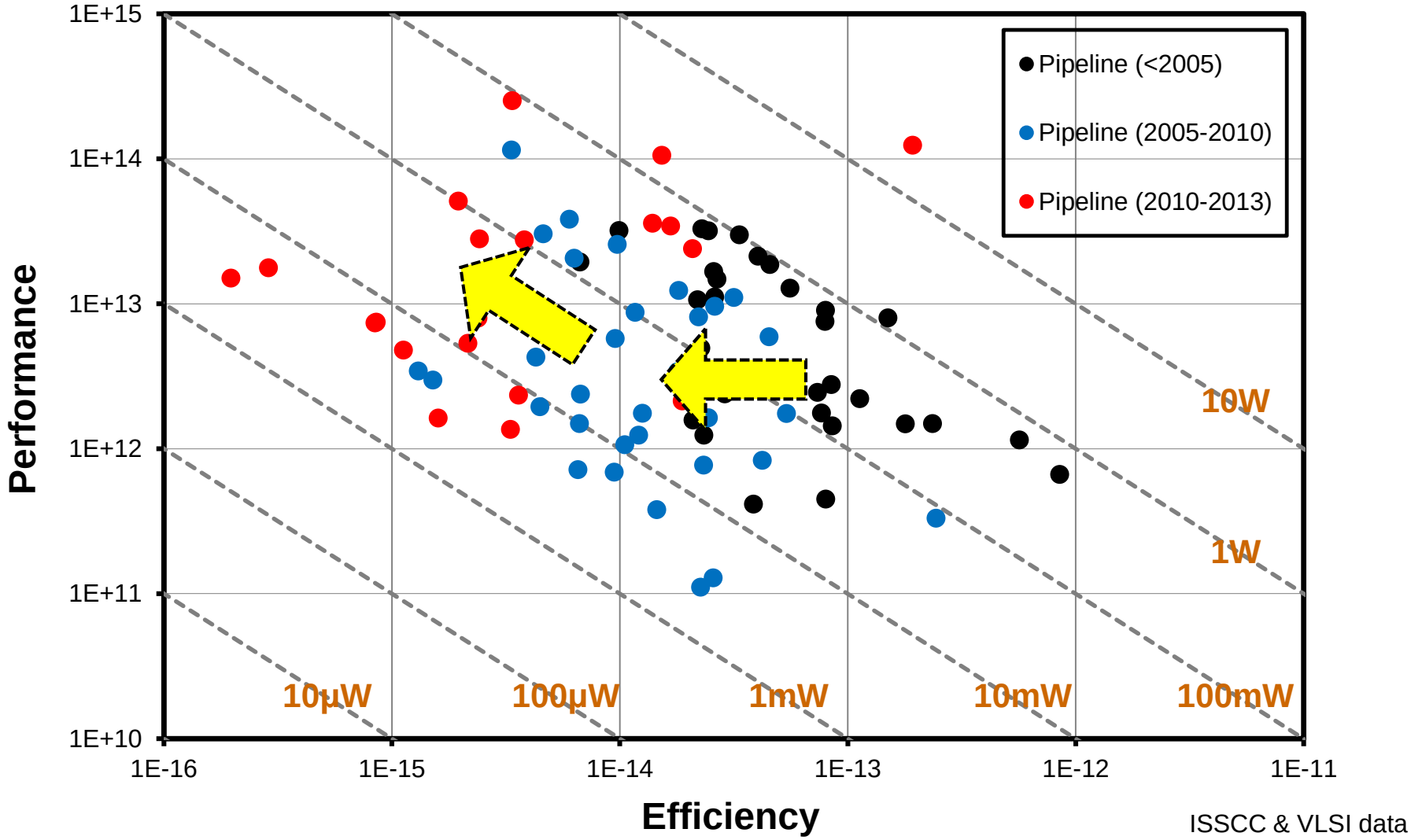
ISSCC & VLSI data

PE Chart: Pipelined ADC (2005-2010)



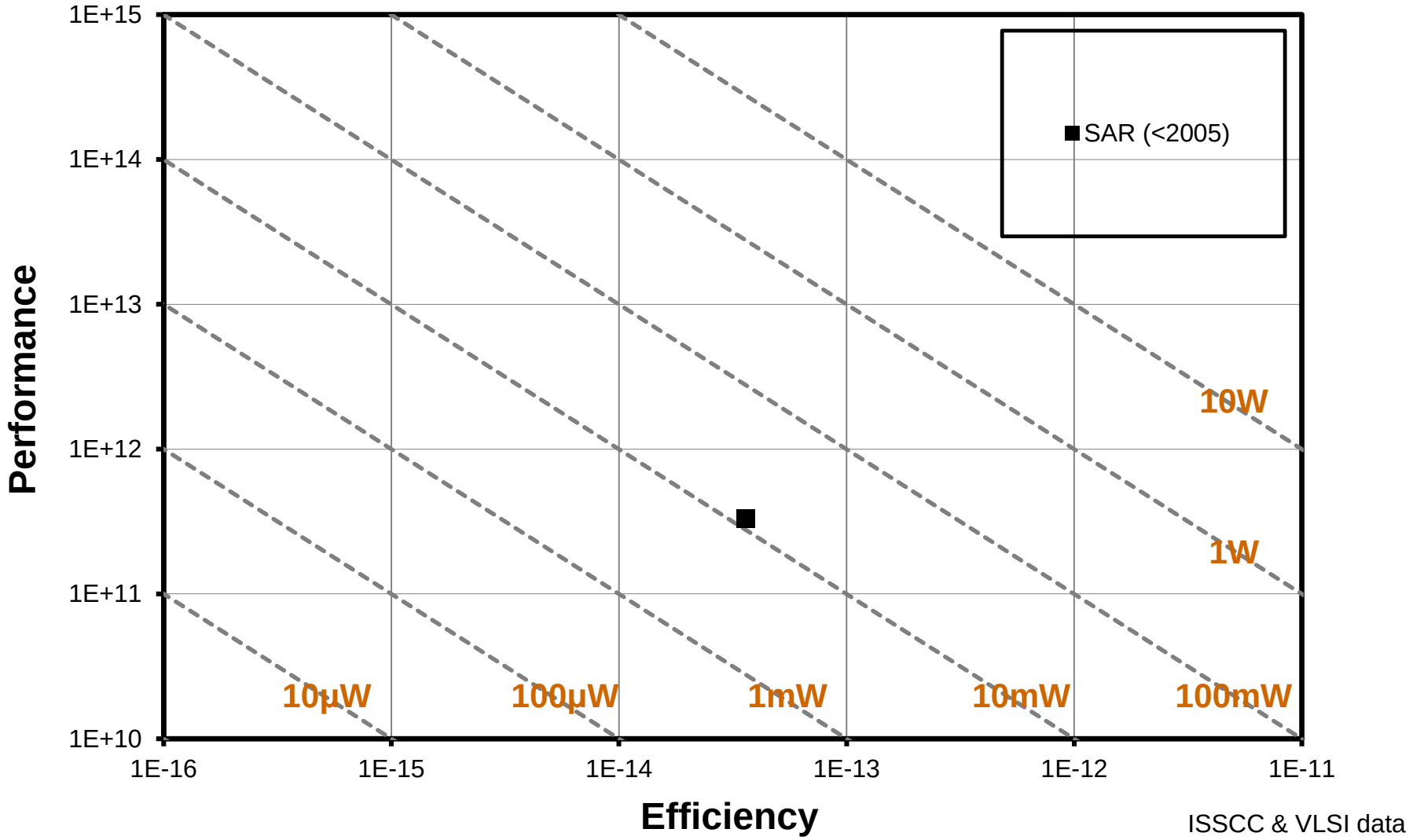
ISSCC & VLSI data

PE Chart: Pipelined ADC (2010-2013)



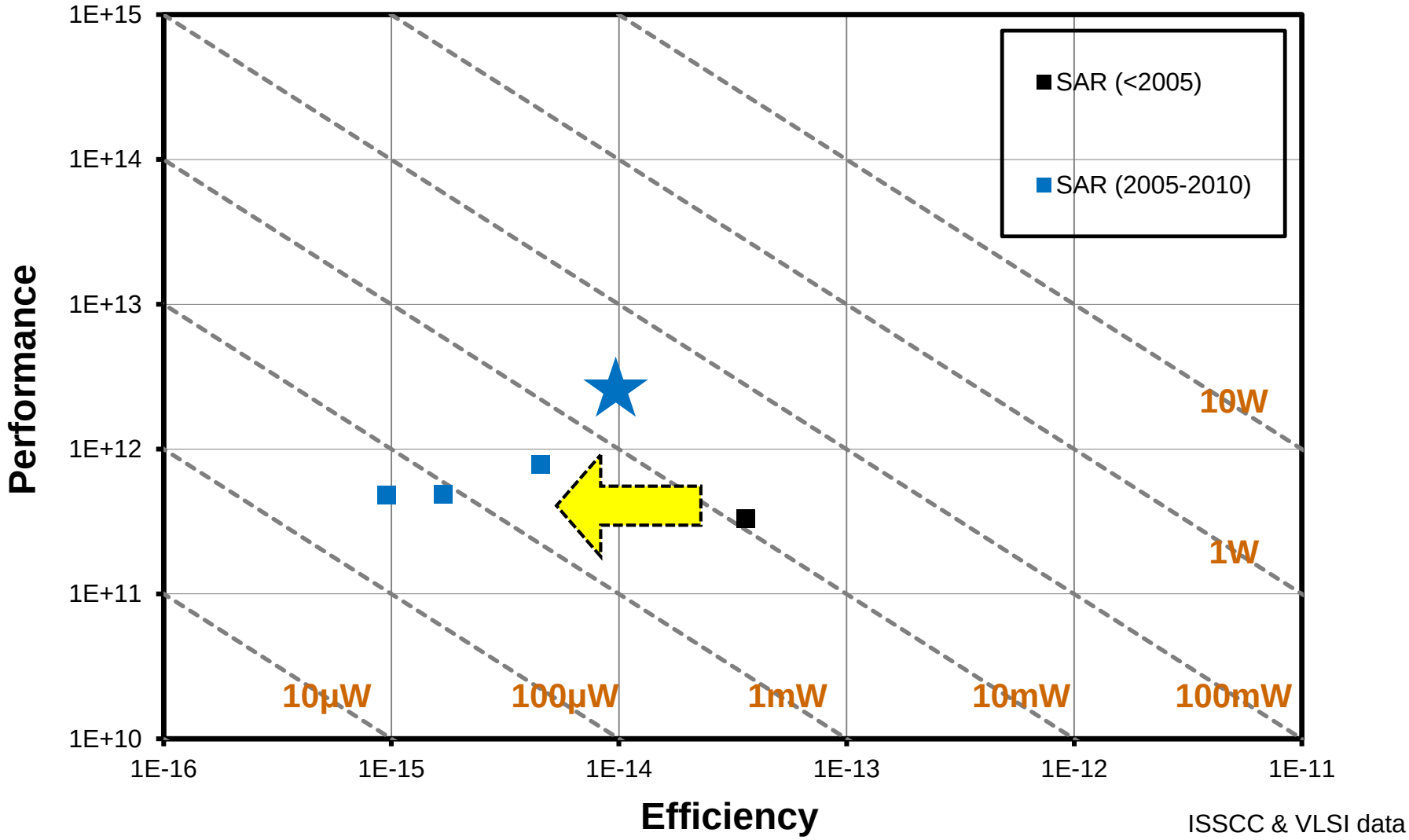
ISSCC & VLSI data

PE Chart: SAR ADC (<2005)



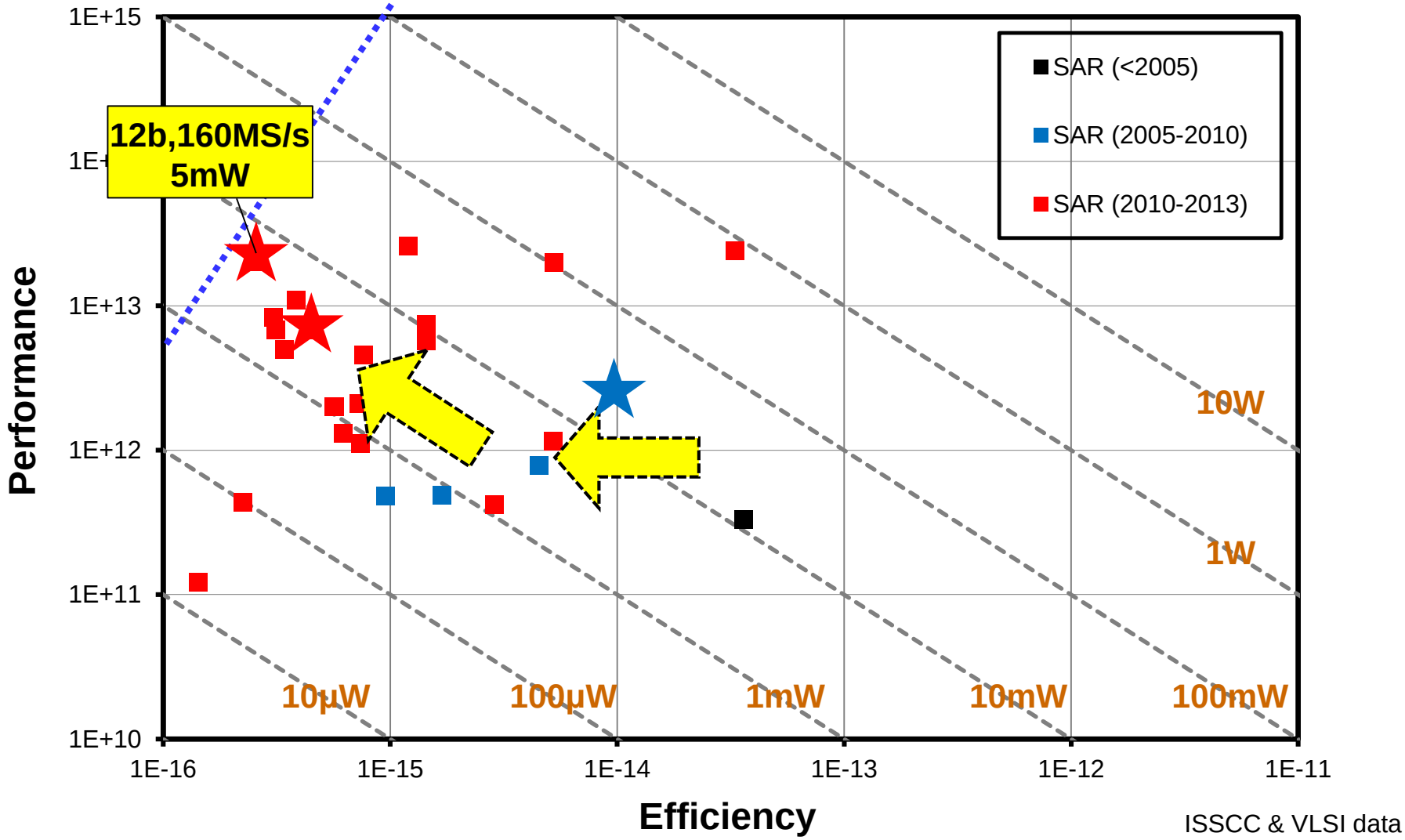
ISSCC & VLSI data

PE Chart: SAR ADC (2005-2010)



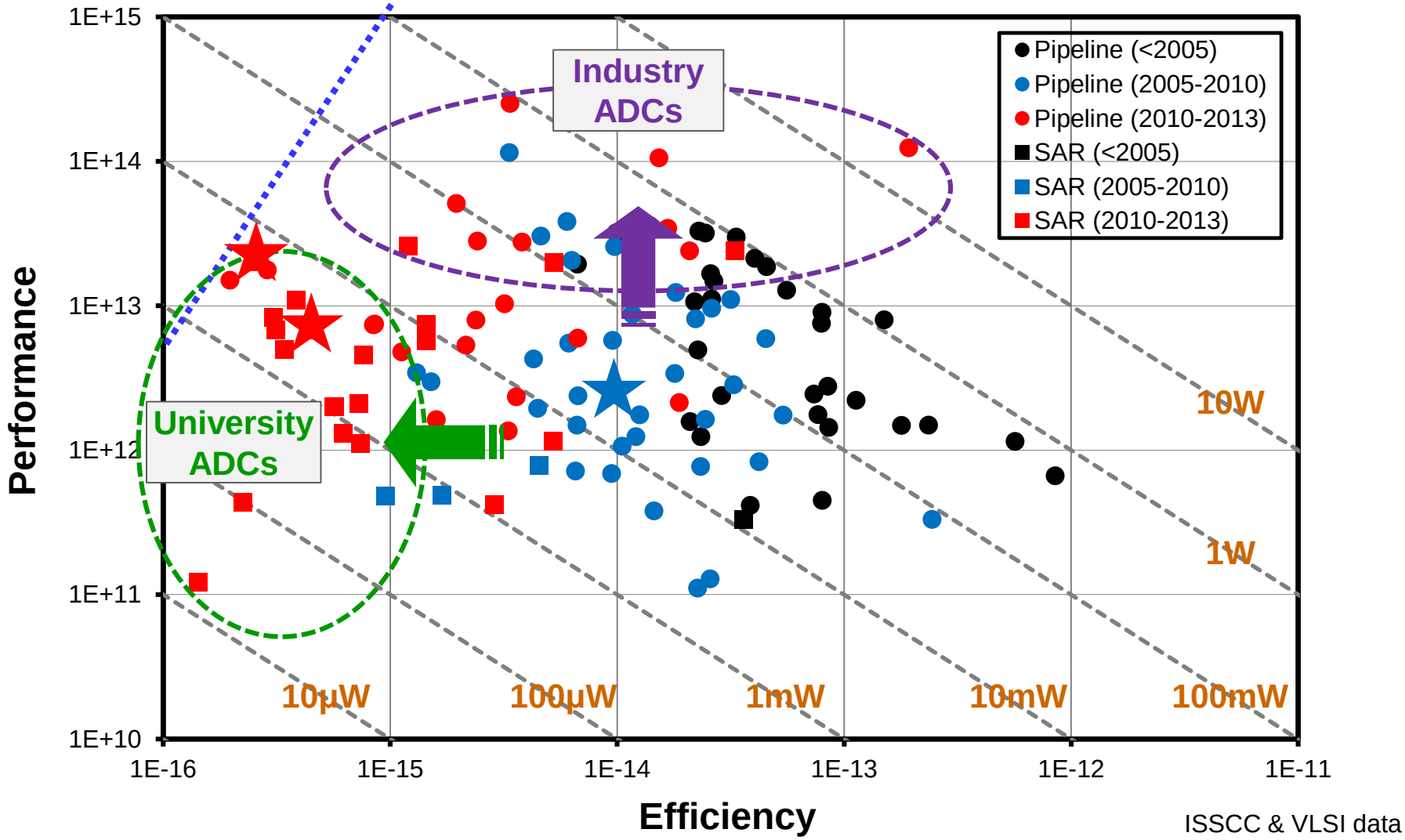
ISSCC & VLSI data

PE Chart: SAR ADC (2010-2013)



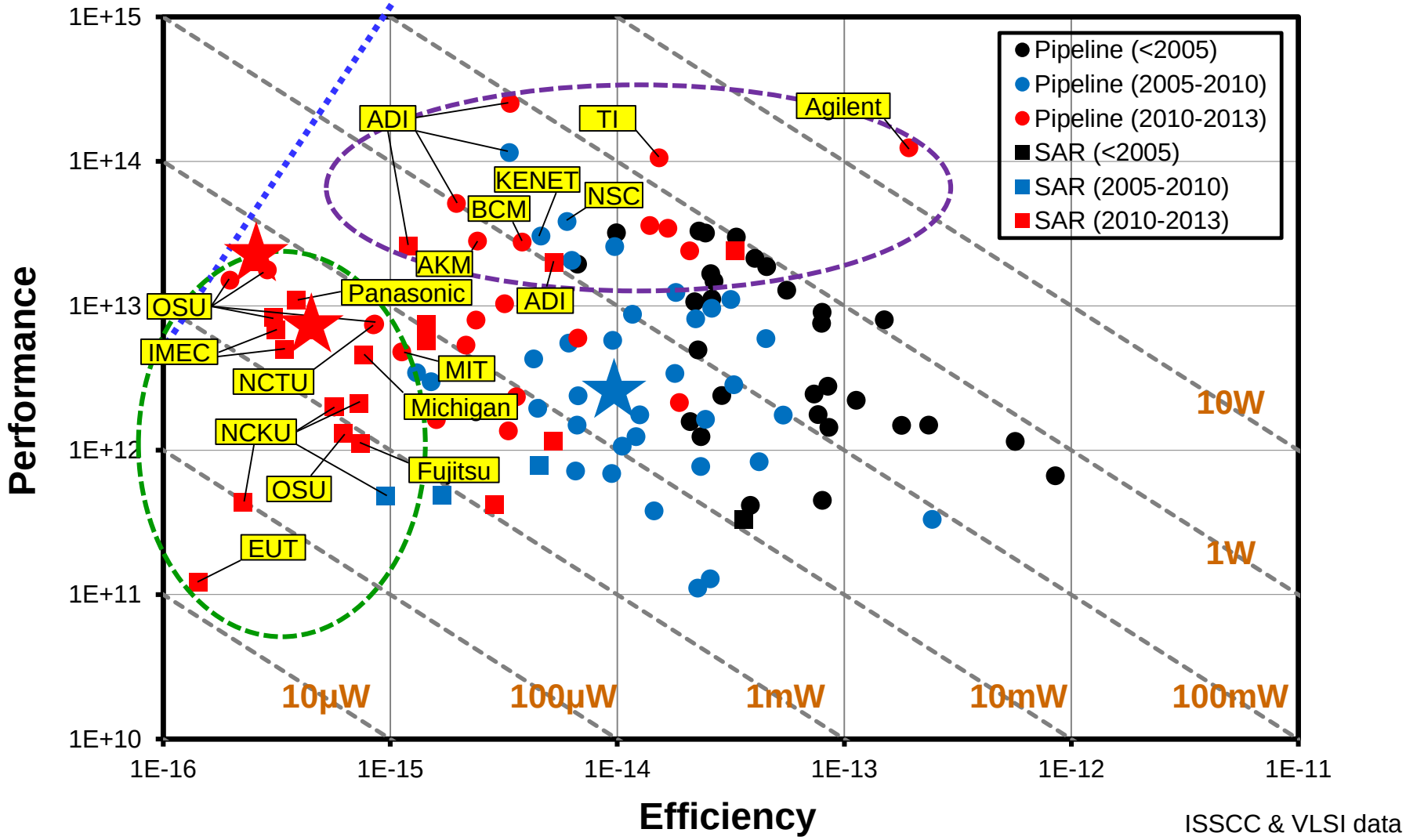
ISSCC & VLSI data

PE Chart: Both ADCs (<2014)



ISSCC & VLSI data

PE Chart: Both ADCs (<2014)



ISSCC & VLSI data

To conclude…

Thank you for your attendance!

Bibliography

1. W. R. Bennett, "Spectra of quantized signals," Bell Syst. Tech. J., vol. 27, pp. 446-472, Jul. 1948.
2. S. H. Lewis et al., "A 10-b 20-MS/s analog-to-digital converter," JSSC, vol. 27, Mar. 1992.
3. F. Kuttner, "A 1.2-V 10-b 20-Msample/s nonbinary successive approximation ADC in 0.13-μm CMOS," in ISSCC, 2002.
4. C. C. Liu et al., "A 10b 100MS/s 1.13mW SAR ADC with binary-scaled error compensation," in ISSCC, 2010.
5. W. Liu et al., "A 600MS/s 30mW 0.13μm CMOS ADC Array Achieving Over 60dB SFDR with Adaptive Digital Equalization," in ISSCC, 2009.
6. T. Sun, A. Wiesbauer, and G. C. Temes, "Adaptive compensation of analog circuit imperfections for cascaded delta-sigma ADCs," in ISCAS, 1998.
7. P. Kiss et al., "Adaptive digital correction of analog errors in MASH ADC's—Part II: Correction using test-signal injection," TCAS2, vol. 47, Jul. 2000.
8. D. Fu, K. C. Dyer, S. H. Lewis, and P. J. Hurst, "A digital back-ground calibration technique for time-interleaved analog-to-digital converters," JSSC, vol. 33, Dec. 1998.
9. E. J. Siragusa and I. Galton, "Gain error correction technique for pipelined analogue-to-digital converters," Electronics Letters, vol. 36, 2000.
10. I. Galton, "Digital cancellation of D/A converter noise in pipelined A/D converters," TCAS2, vol. 47, Mar. 2000.
11. X. Wang, P. J. Hurst, and S. H. Lewis, "A 12-bit 20-MS/s pipelined ADC with nested digital background calibration," in CICC, 2003.
12. Y. Chiu, C. W. Tsang, B. Nikolic, and P. R. Gray, "Least mean square adaptive digital background calibration of pipelined analog-to-digital converters," TCAS1, vol. 51, Jan. 2004.
13. J. McNeill, M.C.W. Coln, B. J. Larivee, ""Split ADC" architecture for deterministic digital background calibration of a 16-bit 1-MS/s ADC," JSSC, vol. 40, Dec. 2005.
14. H. S. Fetterman et al., "CMOS pipelined ADC employing dither to improve linearity," in CICC, 1999.

Bibliography

15. Y.-S. Shu and B.-S. Song, "A 15b linear, 20MS/s, 1.5b/stage pipelined ADC digitally calibrated with signal-dependent dithering," in VLSI, 2006.
16. Y. Zhou, B. Xu, and Y. Chiu, "A 12b 160MS/s synchronous two-step SAR ADC achieving 20.7fJ/step FoM with opportunistic digital background calibration," to appear in VLSI, 2014.
17. Y. Chiu, S.-C. Lee, and W. Liu, "An ICA framework for digital background calibration of analog-to-digital converters," Sampling Theory in Signal and Image Processing (STSIP), vol. 11, no. 2-3, 2012.
18. W. Liu, P. Huang, and Y. Chiu, "A 12-bit 50-MS/s 3.3-mW SAR ADC with background digital calibration," in CICC, 2012.
19. W. Liu, P. Huang, and Y. Chiu, "A 12bit 22.5/45MS/s 3.0mW 0.059mm^2 CMOS SAR ADC achieving over 90dB SFDR," in ISSCC, 2010.
20. P. C. Yu et al., "A 14b 40MS/s pipelined ADC with DFCA," in ISSCC, 2001.
21. E. J. Siragusa and I. Galton, "A digitally enhanced 1.8V 15b 40MS/s CMOS pipelined ADC," in ISSCC, 2004.
22. A. Panigada and I. Galton, "Digital background correction of harmonic distortion in pipelined ADCs," TCAS I, Sep. 2006.
23. A. Panigada and I. Galton, "A 130mW 100MS/s pipelined ADC with 69dB SNDR enabled by digital harmonic distortion correction," in ISSCC, 2009.
24. K. Nair and R. Harjani, "A 96dB SFDR 50MS/s digitally enhanced CMOS pipeline A/D converter," in ISSCC, 2004.
25. H.-C. Liu, Z.-M. Lee, and J.-T. Wu, "A 15b 20MS/s CMOS pipelined ADC with digital background calibration," in ISSCC, 2004.
26. J.-L. Fan, C.-Y. Wang, and J.-T. Wu, "A robust and fast digital background calibration technique for pipelined ADCs," TCAS I, Jun. 2007.
27. J. Li and U.-K. Moon, "Background calibration techniques for multistage pipelined ADC's with digital redundancy," TCAS II, Sep. 2003.

Bibliography

28. J. Li et al., "0.9V 12mW 2MSPS algorithmic ADC with 81dB SFDR," in VLSI, 2004.
29. B. Murmann et al., "A 12b 75MS/s pipelined ADC using open- loop residue amplification," in ISSCC, 2003.
30. J. Keane et al., "Background interstage gain calibration technique for pipelined ADCs," TCAS I, Jan. 2005.
31. R. Massolini, G. Cesura, and R. Castello, "A fully digital fast convergence algorithm for nonlinearity correction in multistage ADC," TCAS II, May 2006.
32. S.-C. Lee, B. Elies, and Y. Chiu, "An 85dB SFDR 67dB SNDR 8OSR 240MS/s SD ADC with nonlinear memory error calibration," in VLSI, 2012.
33. Y. Zhou and Y. Chiu, “Digital calibration of inter-stage nonlinear errors in pipelined SAR ADC,” in MWSCAS, 2013.
34. C. Tsang et al., "Background ADC calibration in digital domain," in CICC, 2008.
35. D. Stepanovic and B. Nikolic, “A 2.8GS/s 44.6mW time-interleaved ADC achieving 50.9dB SNDR and 3dB effective resolution bandwidth of 1.5GHz in 65nm CMOS,” in VLSI, 2012.
36. B. Xu and Y. Chiu, “Background calibration of time-interleaved ADC using direct derivative information,” in ISCAS, 2013.
37. B. Peng et al., "A virtual-ADC digital background calibration technique for multistage A/D conversion," TCAS II, Nov. 2010.
38. B. Peng et al., "A 48-mW, 12-bit, 150-MS/s pipelined ADC with digital calibration in 65nm CMOS," in CICC, 2011.
39. J. McNeill et al., "Split-ADC digital background correction of open-loop residue amplifier nonlinearity errors in a 14b pipeline ADC," in ISCAS, 2007.
40. S. Sarkar, Y. Zhou and Y. Chiu, "PN-assisted deterministic digital calibration of split two-step ADC to over 14-bit accuracy," to appear in MWSCAS, 2014.
41. B. Peng et al., "An offset double conversion technique for digital calibration of pipelined ADCs," TCAS II, Dec. 2010.
42. B. Murmann, "ADC Performance Survey 1997-2013," [Online]: www.stanford.edu/~murmann/adcsurvey.html.