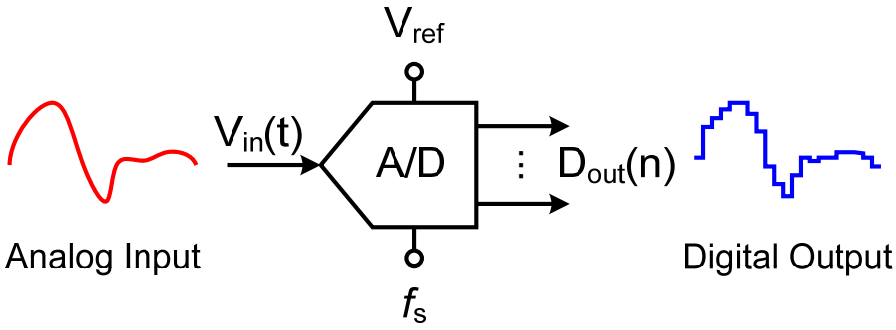
Recent Advances in Multistep Nyquist ADC's

Yun Chiu

Erik Jonsson Distinguished Professor
University of Texas at Dallas

Performance vs. Energy Efficiency

Walden Figure-of-Merit (FoM) for ADC

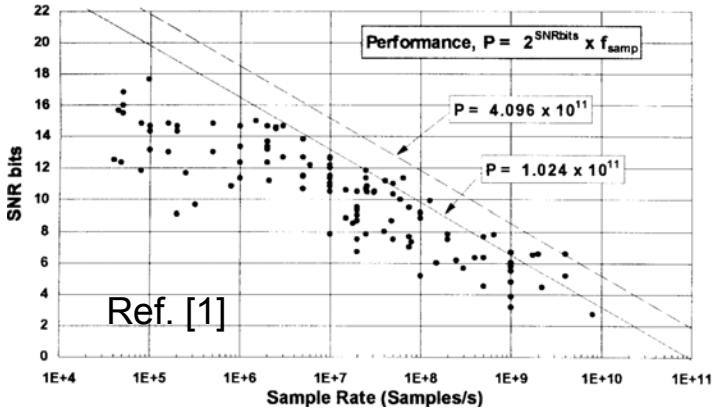


$$D_{out}(n) = \left\lfloor 2^N \cdot \frac{V_{in}(nT_s)}{V_{ref}} \right\rfloor$$

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

FoM "measures" energy efficiency

- P = power consumption
- ENOB = effective number of bits
- BW = min(f_s/2, ERBW)
- ERBW = effective resolution BW



How to compare ADC performance?

- **Higher performance** with **lower cost** is the obvious criterion for comparison
  - **ADC performance:** speed (sample rate) or bandwidth, resolution
  - **ADC cost:** power consumption, die size

Q: how to define performance quantitatively?

Definition of PERFORMANCE

- Bandwidth (or speed) performance
  - BW = min(f_s/2, ERBW)
- Resolution (or precision) performance
  - Effective number of bits (ENOB), or equivalently effective number of steps (ENOS) = 2^{ENOB}

- Separately, it is easy. But how to combine the two in one merit? Just take the product?

Definition of PERFORMANCE

- 2×BW → 2×Power, 2×Area (same ENOS) → **Power, area scale linearly with bandwidth**

- 2×ENOS → 4×Power, 4×Area (same BW) → **Power, area scale quadratically with precision**

  (for thermal noise or matching limited design)

$$\text{Power, Area} \propto \text{BW},$$

$$\text{Power, Area} \propto \text{ENOS}^2$$

Definition of PERFORMANCE

$$\text{Power \& Area} \propto BW, \propto ENOS^2$$

Definition:

$$\text{Performance} = 2 \cdot BW \cdot ENOS^2 = 2 \cdot BW \cdot 4^{ENOB} \quad \left[Hz \cdot Step^2 \right]$$

$$2 \cdot BW = f_{sample} \qquad 10\log_{10}\left(4^{ENOB}\right) = SNR$$

This definition avoids penalizing high-SNR works as Walden FoM does

Definition of ENERGY EFFICIENCY

$$\text{Power \& Area} \propto BW, \propto ENOS^2$$

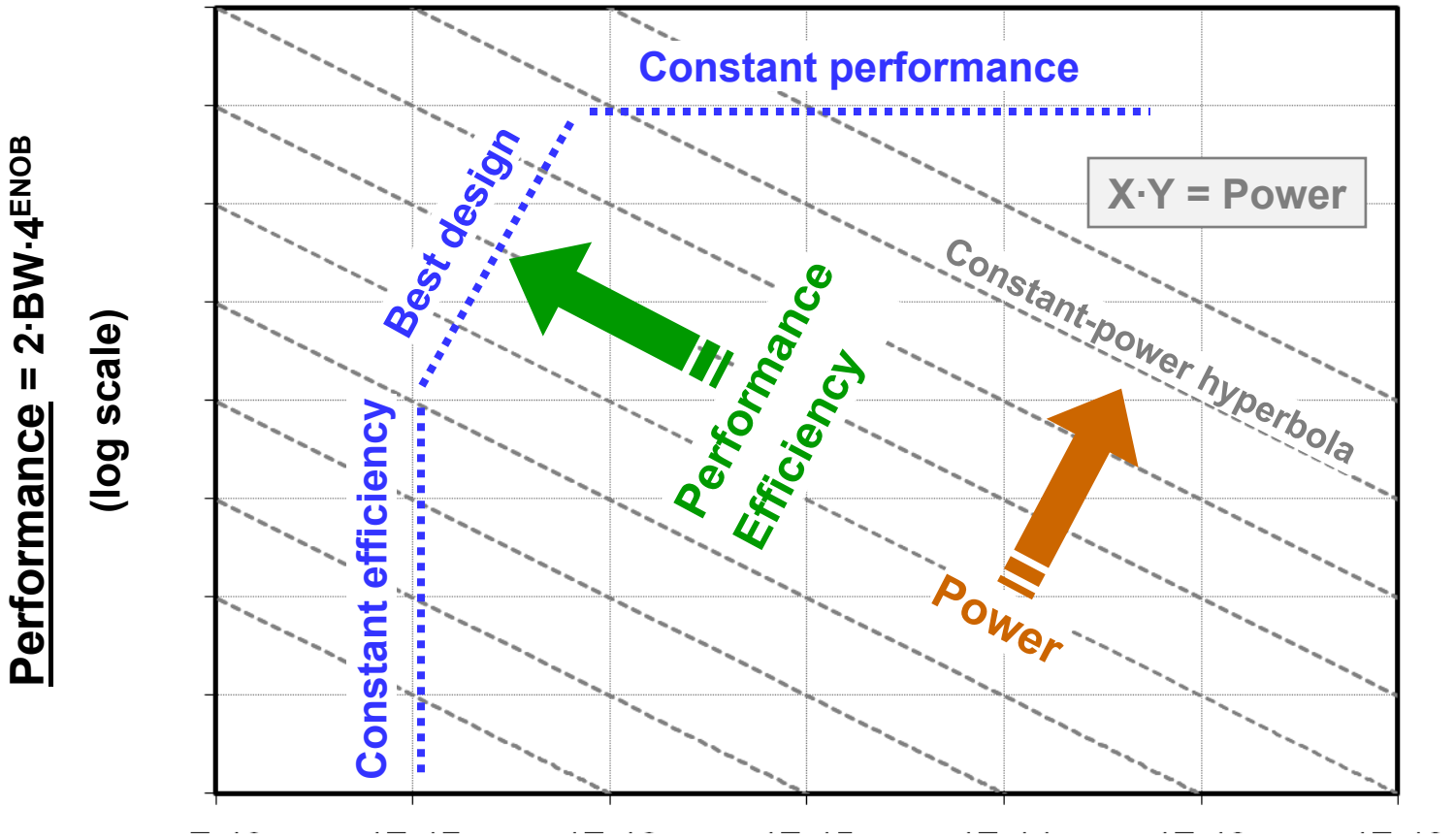
Definition:

$$\text{Energy Efficiency} = \frac{P}{2 \cdot BW \cdot ENOS^2} \left[\frac{J}{Step^2}\right]$$
$$= \frac{P}{2 \cdot BW \cdot 4^{ENOB}}$$

Note: $\text{Performance} \times \text{Energy Efficiency}$

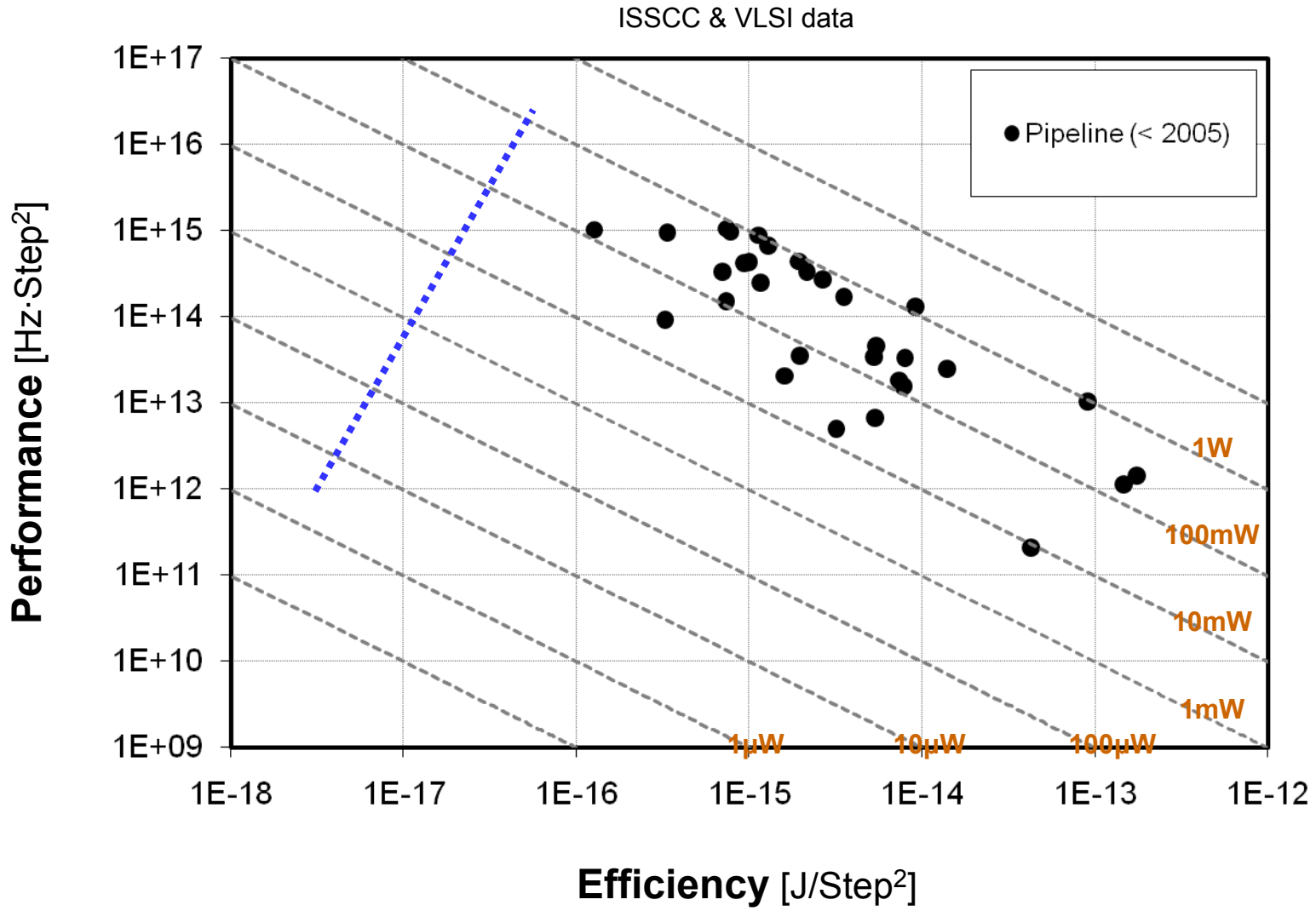
$$= 2 \cdot BW \cdot 4^{ENOB} \times \frac{P}{2 \cdot BW \cdot 4^{ENOB}} = \text{Power}$$

Performance–Efficiency (PE) Chart

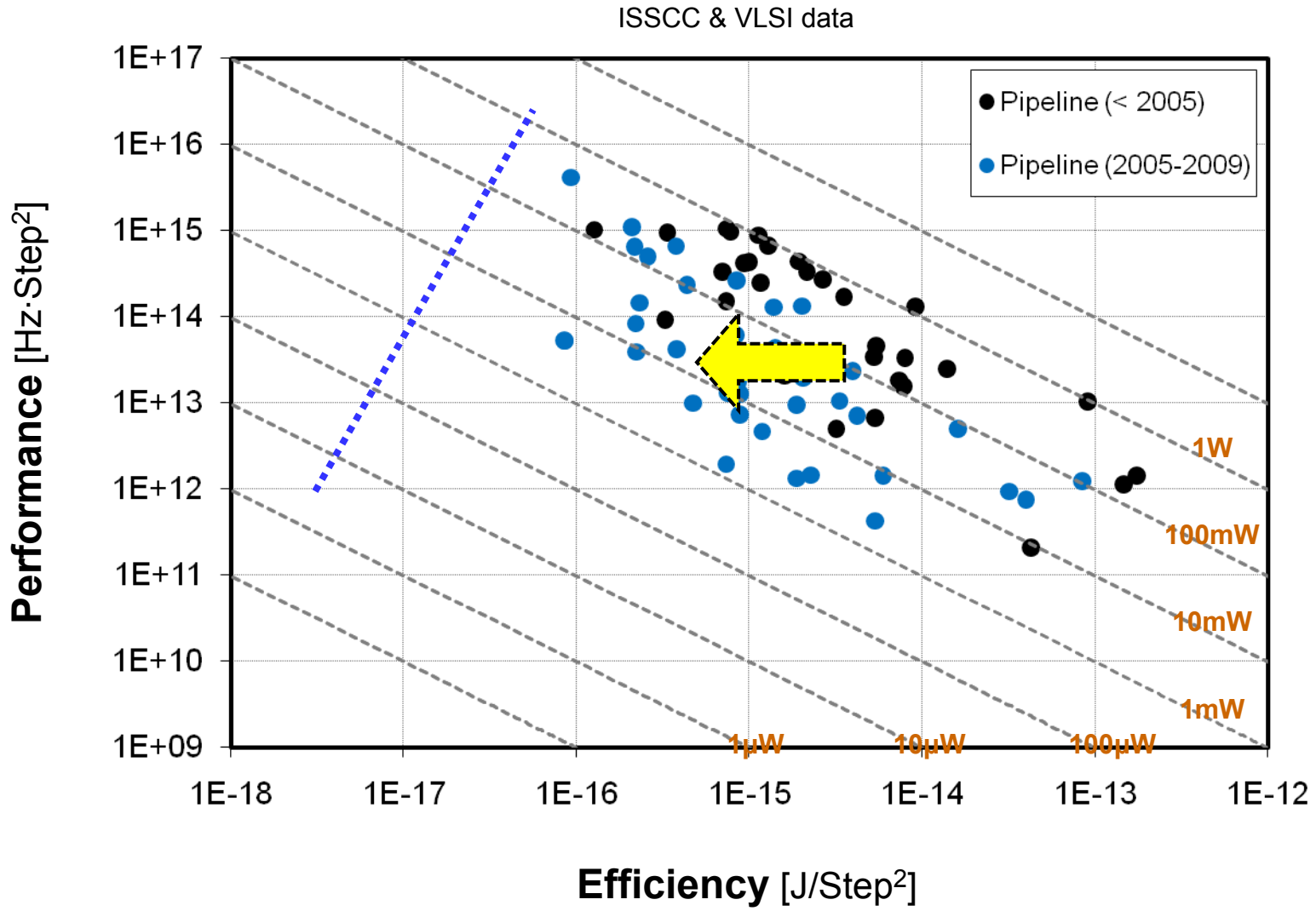


Energy Efficiency = Power/(2·BW·4^ENOB)

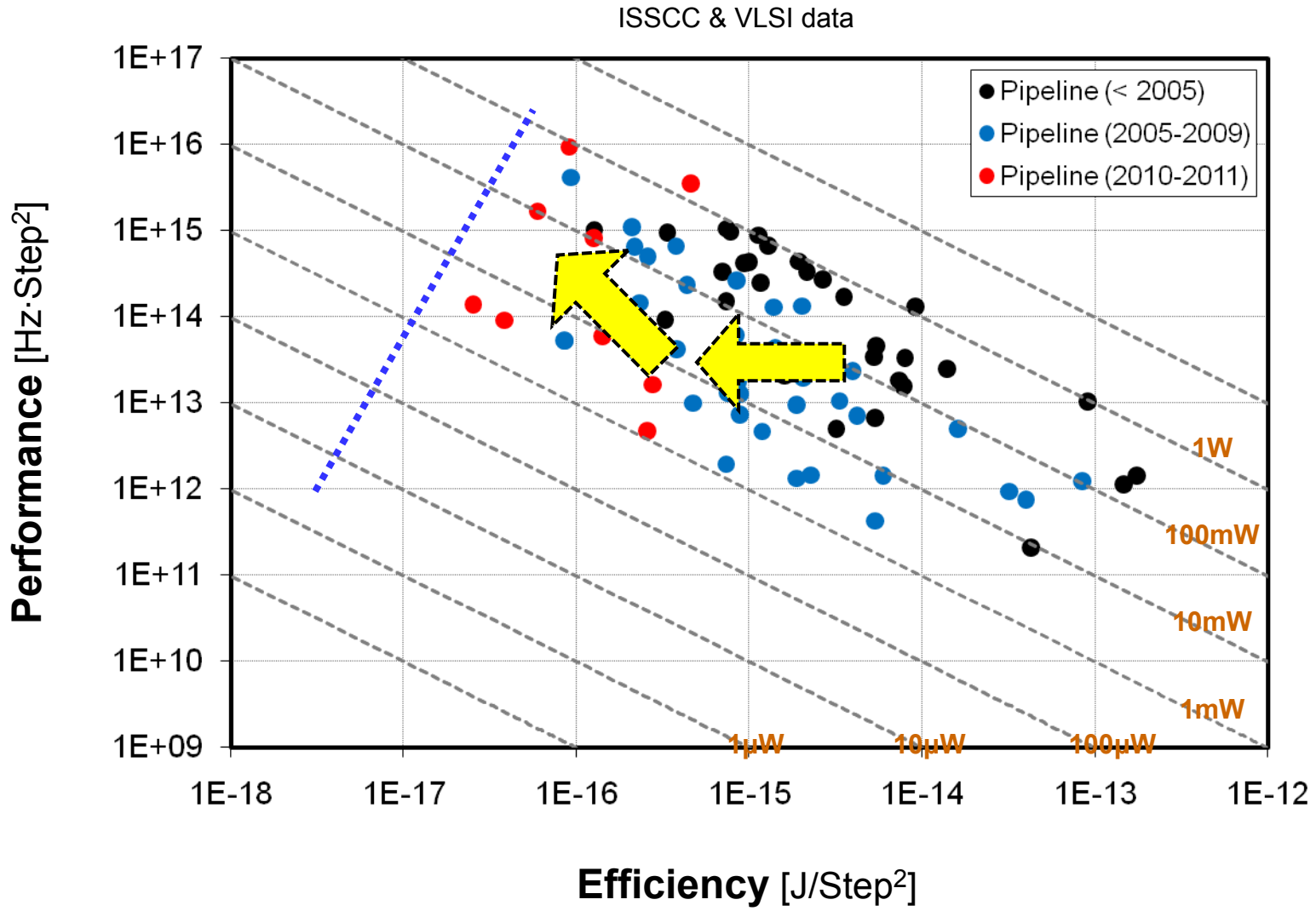
Pipeline ADC PE Chart (< 2005)



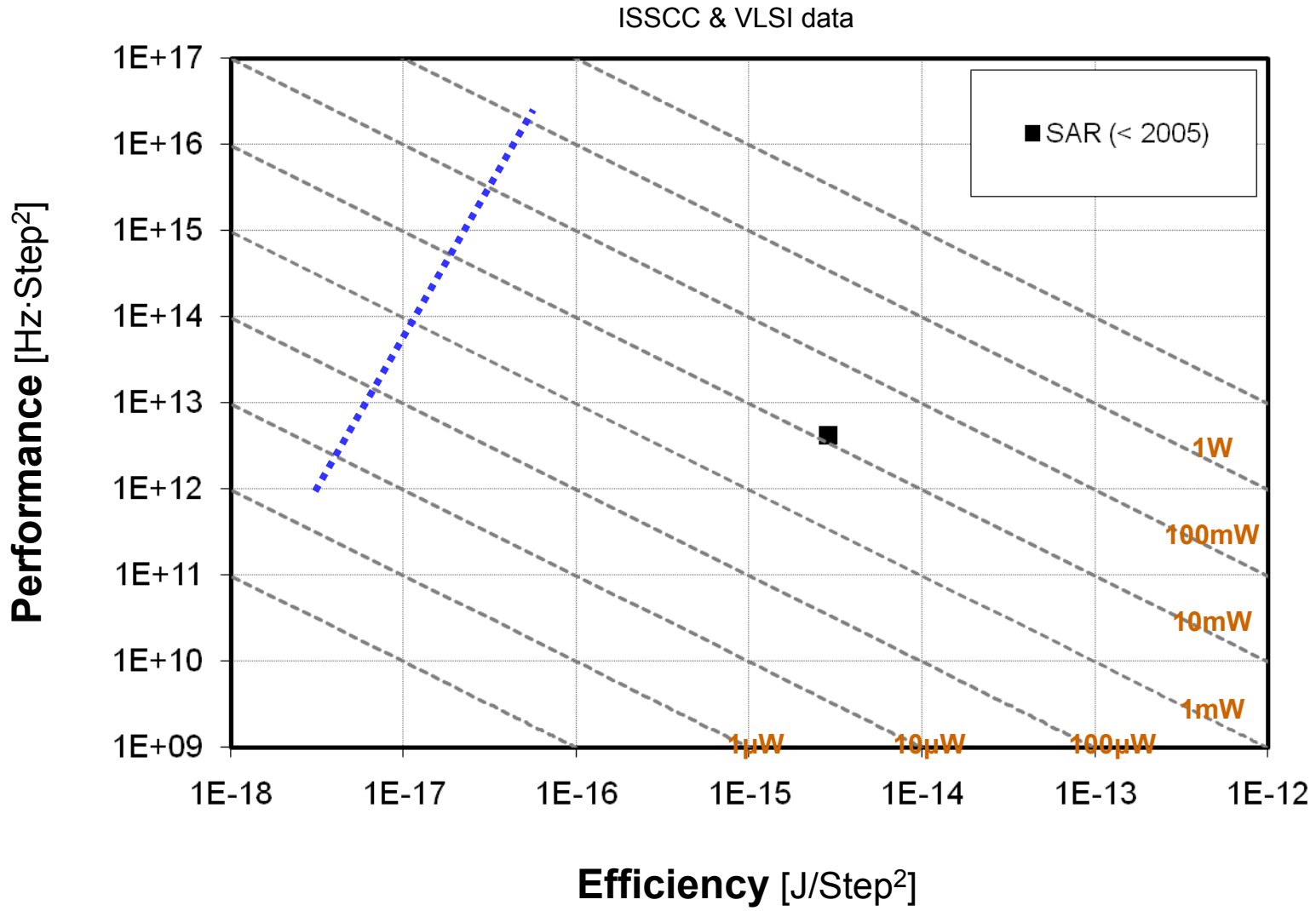
Pipeline ADC PE Chart (< 2010)



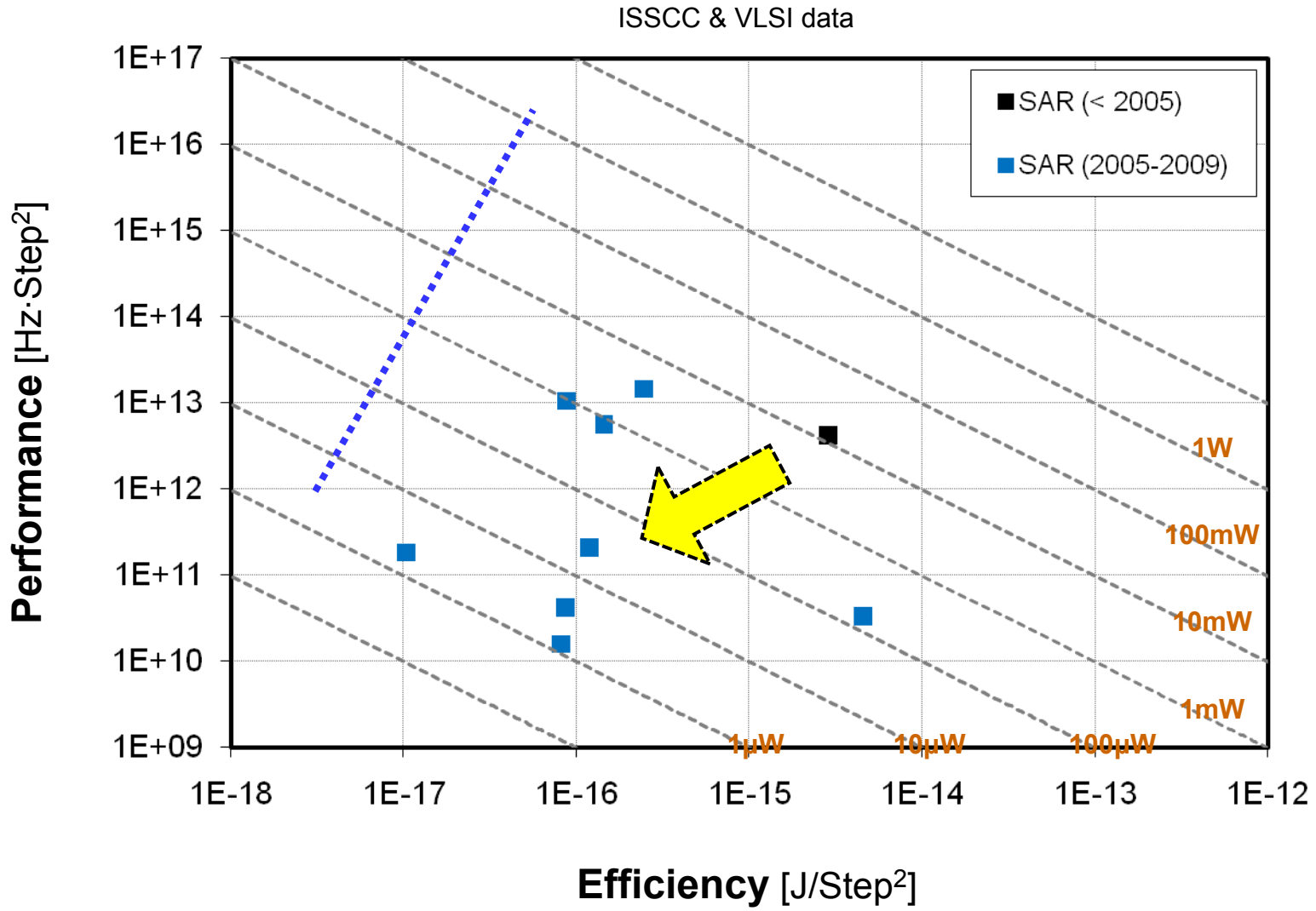
Pipeline ADC PE Chart(< 2012)



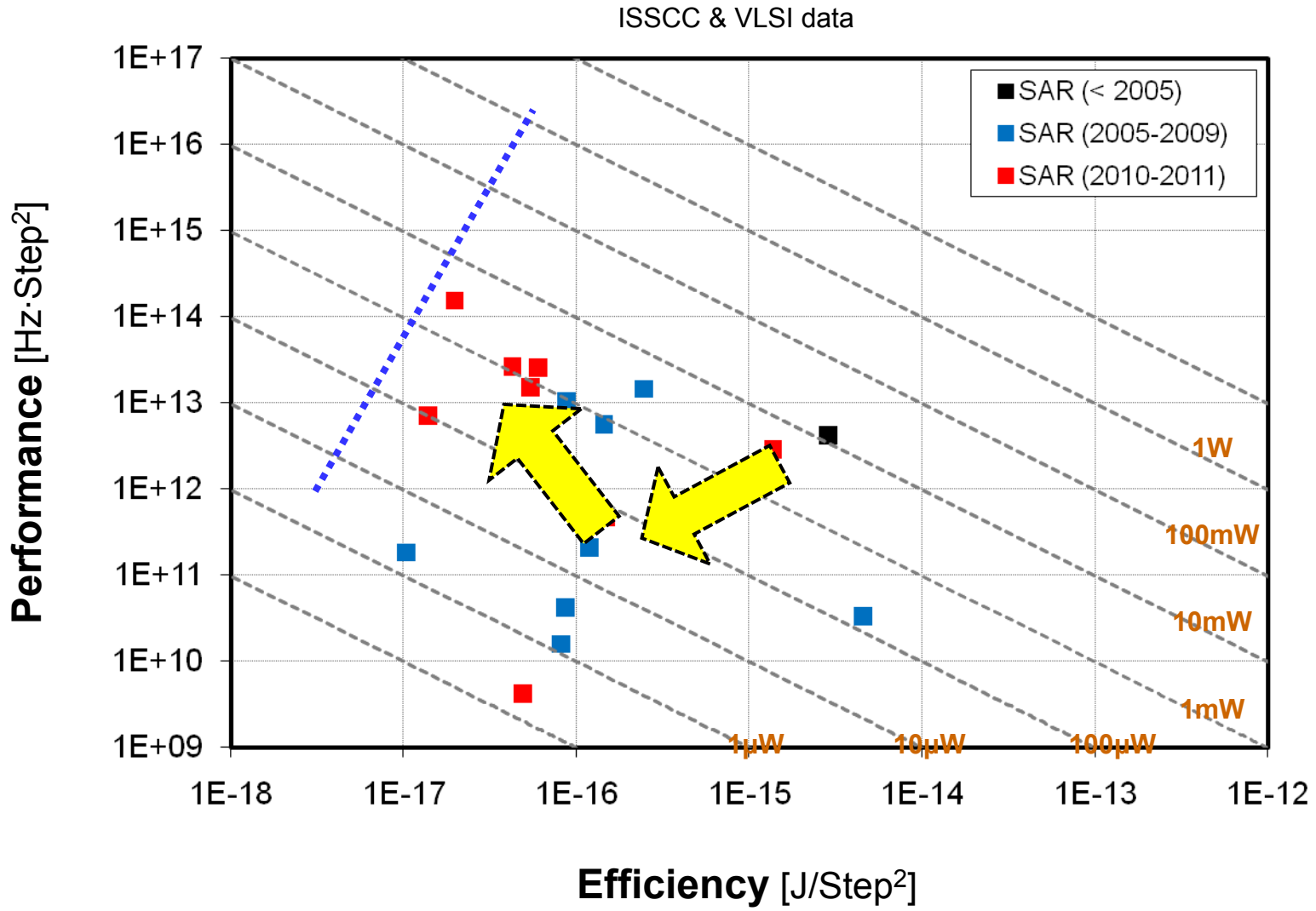
SAR ADC PE Chart (< 2005)



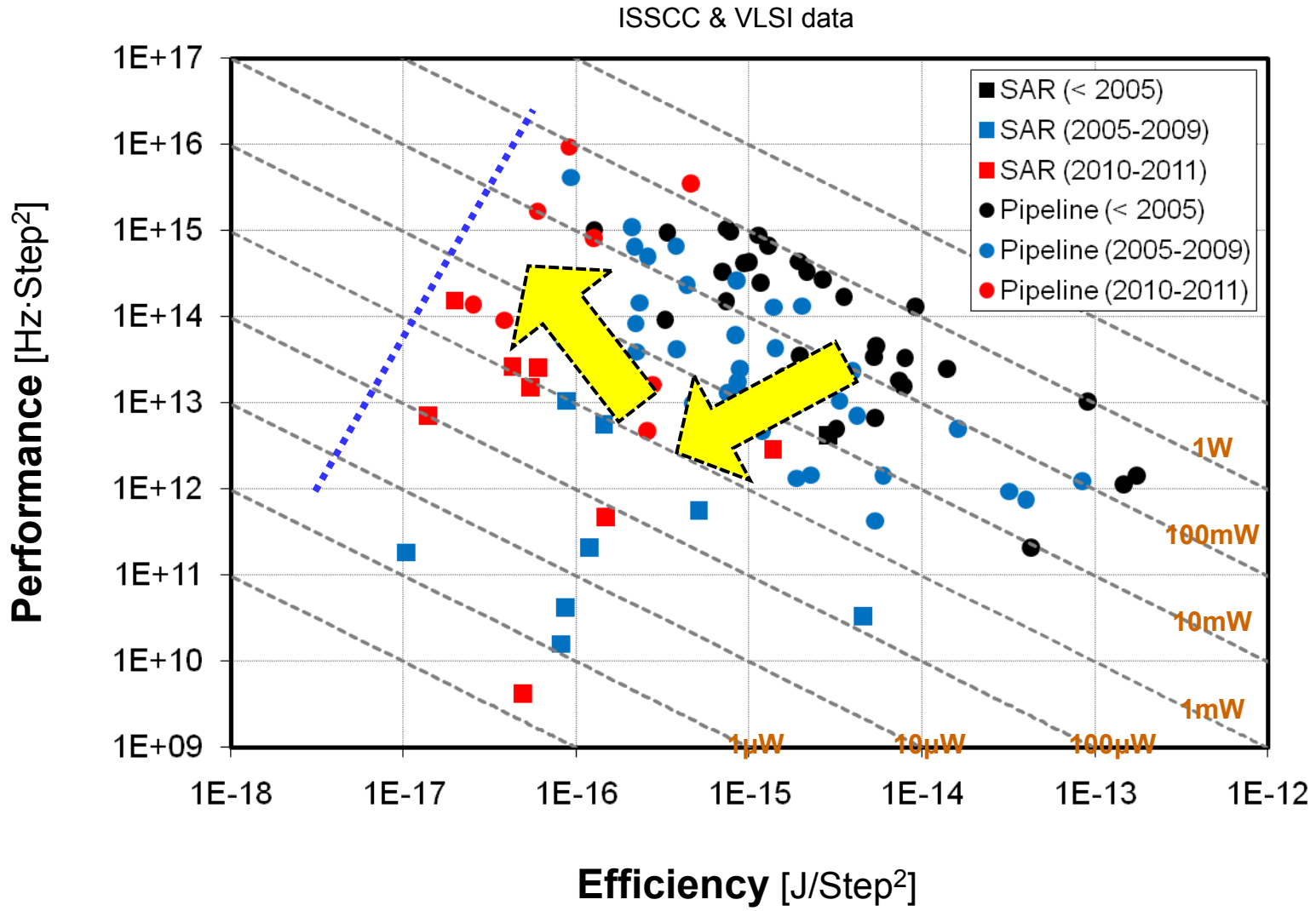
SAR ADC PE Chart (< 2010)



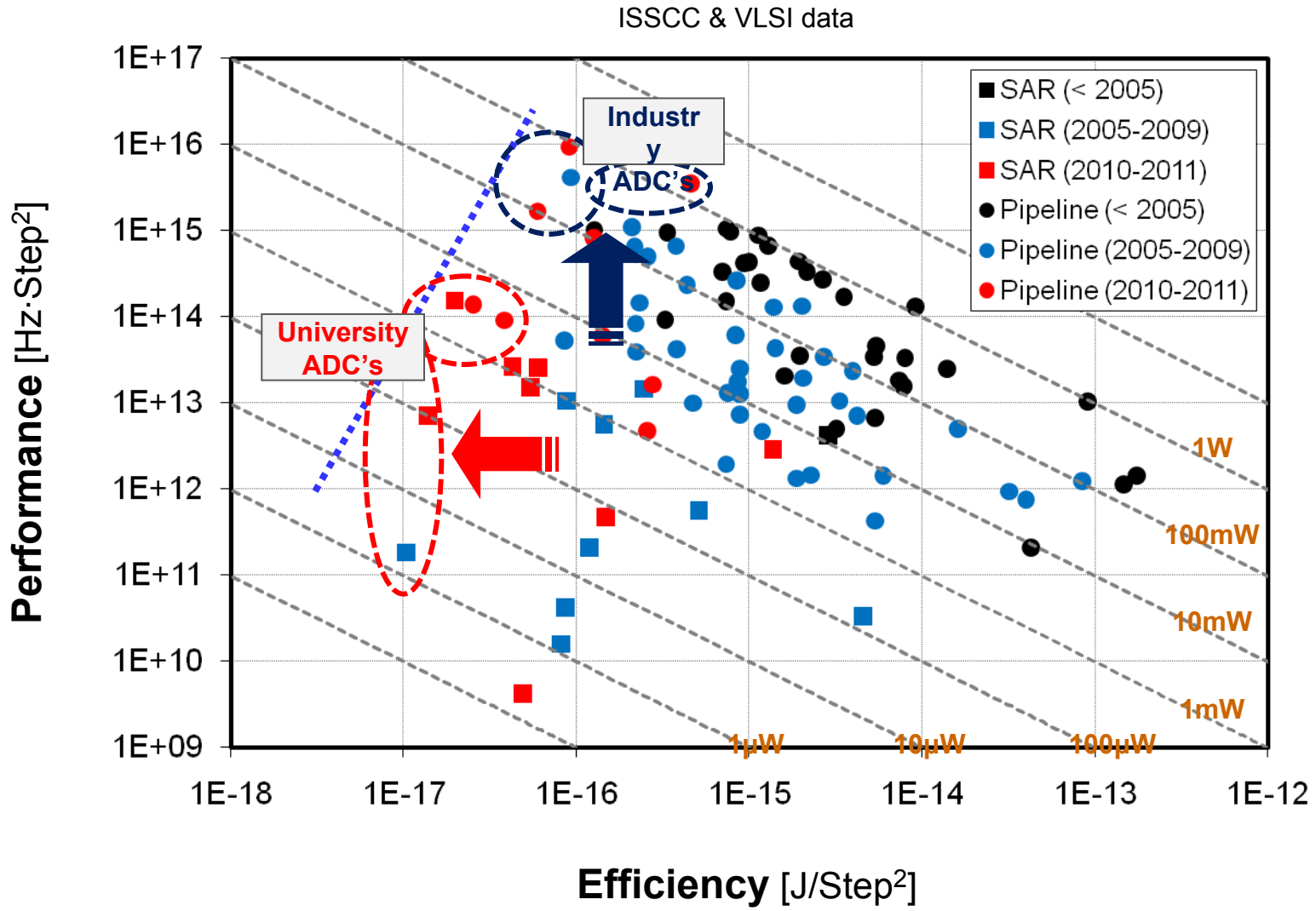
SAR ADC PE Chart (< 2012)



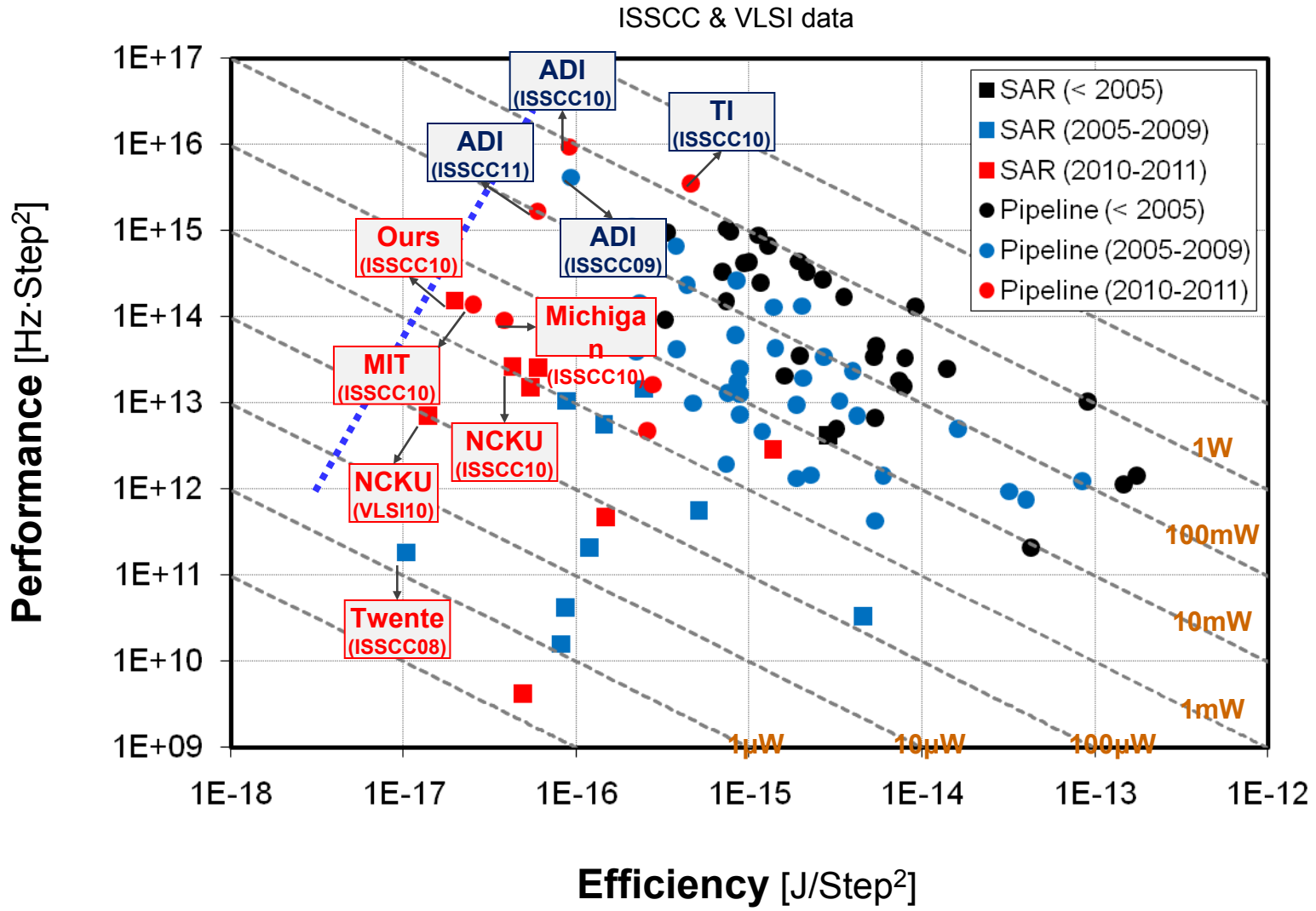
Nyquist ADC PE Chart (mid 90s – 2011)



Nyquist ADC PE Chart (mid 90s – 2011)

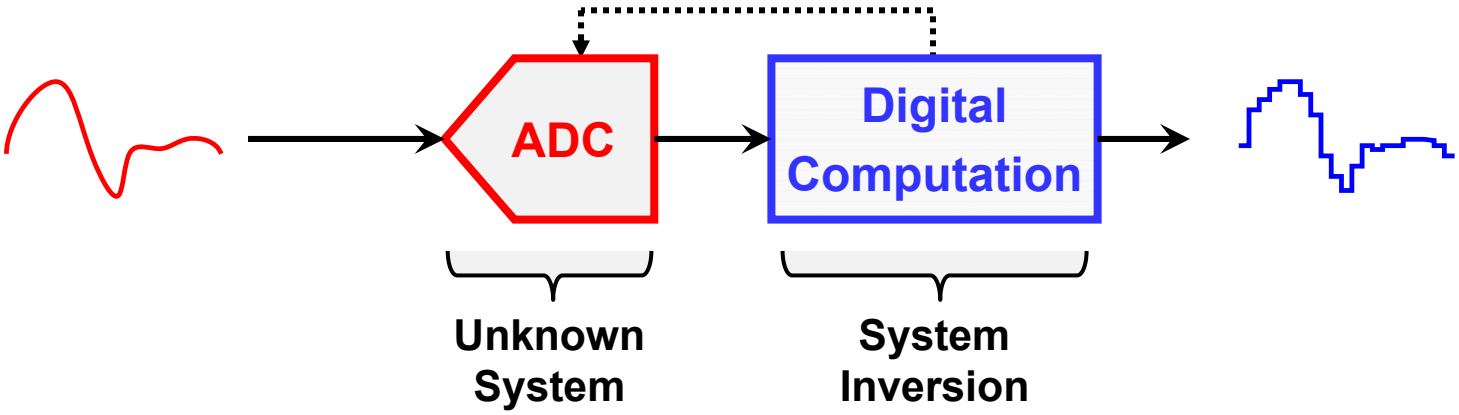


Nyquist ADC PE Chart (mid 90s – 2011)



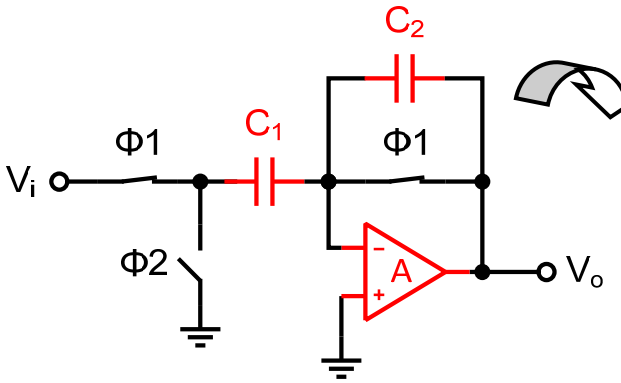
Digital-Domain Calibration

The Basic Idea



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

<u>Switched-capacitor amplifier</u>



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

<u>Analog soln:</u>

- match C_1 and C_2
- make βA very large

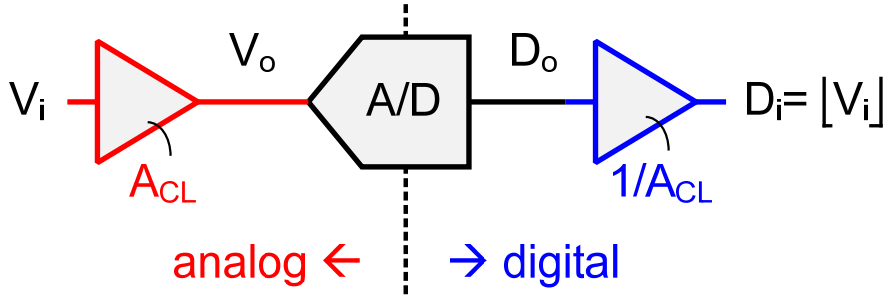
<u>Digital soln:</u>

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

Two Essential Components

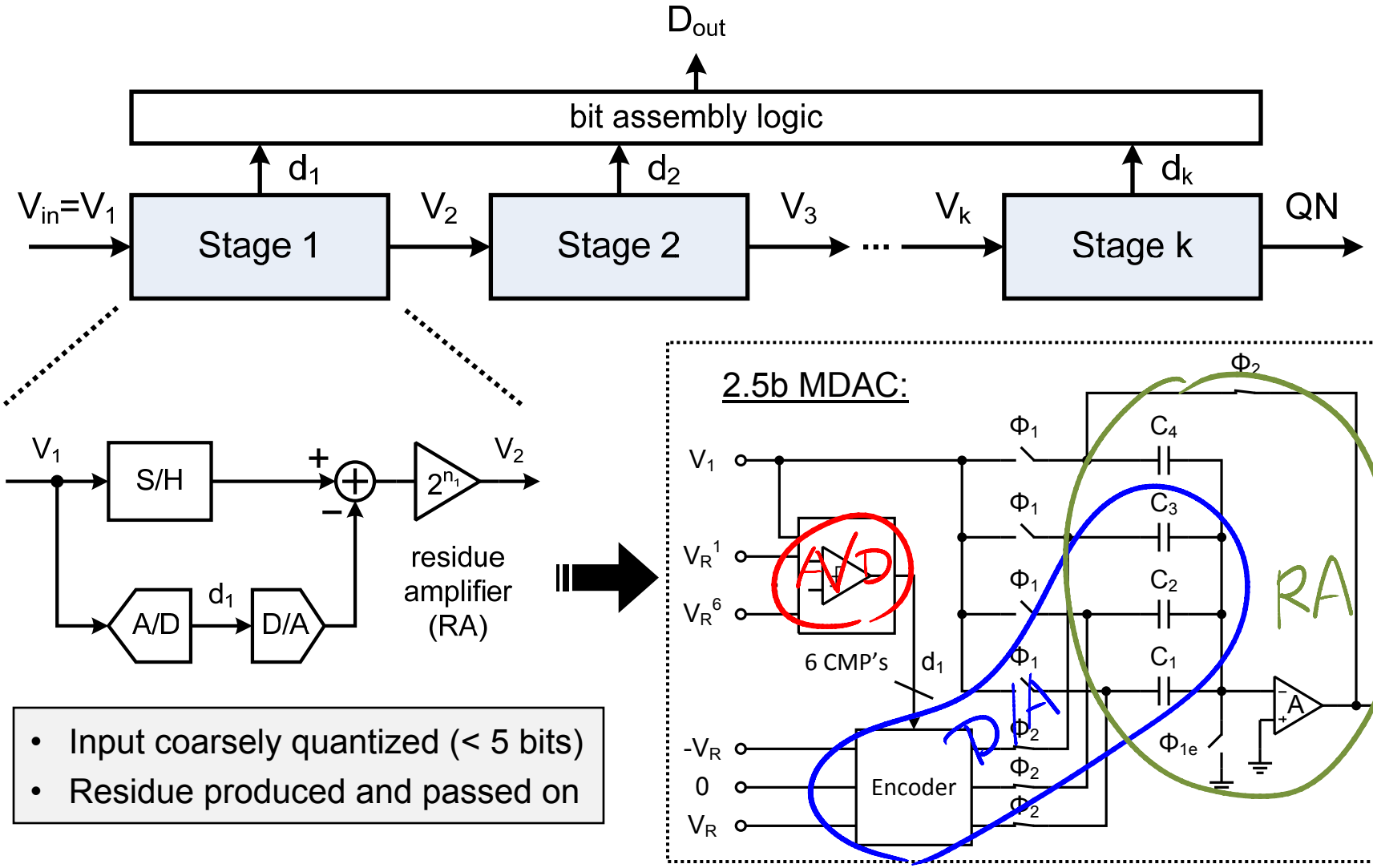
1. A digital-domain technique (e.g. equation) to recover accurate analog information from raw digital output
   - Treat analog precision or linearity only
   - Neglect consequence on SNR

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



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

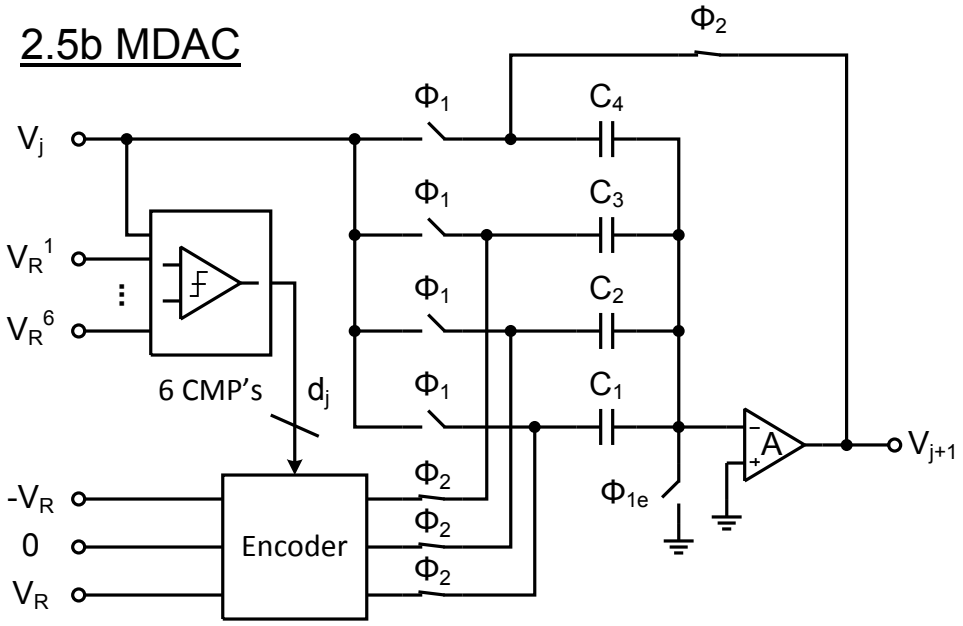
Example – Multistage Pipeline ADC



- Input coarsely quantized (< 5 bits)
- Residue produced and passed on

Multiplying DAC Error Mechanism

2.5b MDAC



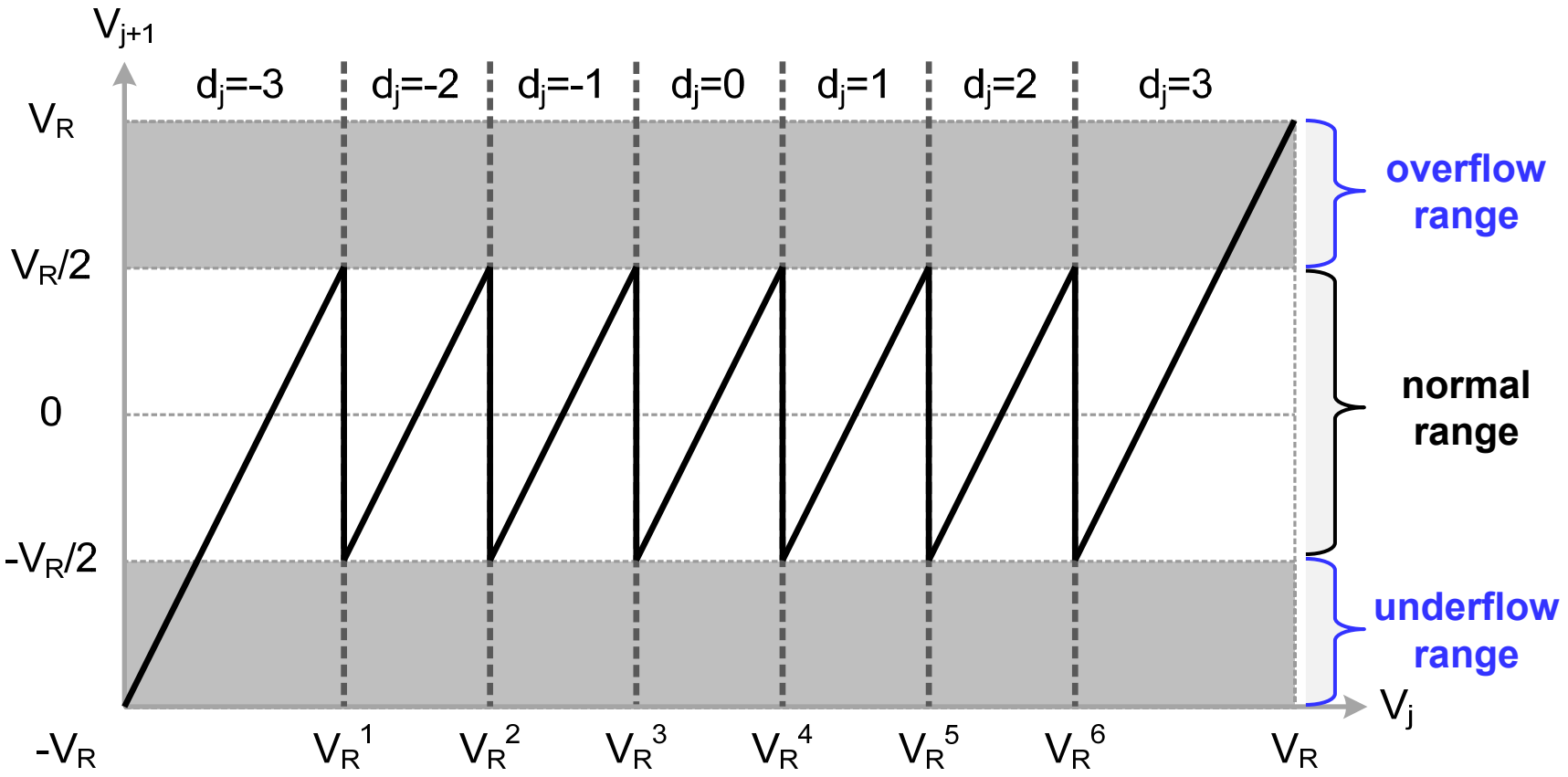
DAC bit-encoding scheme

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

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

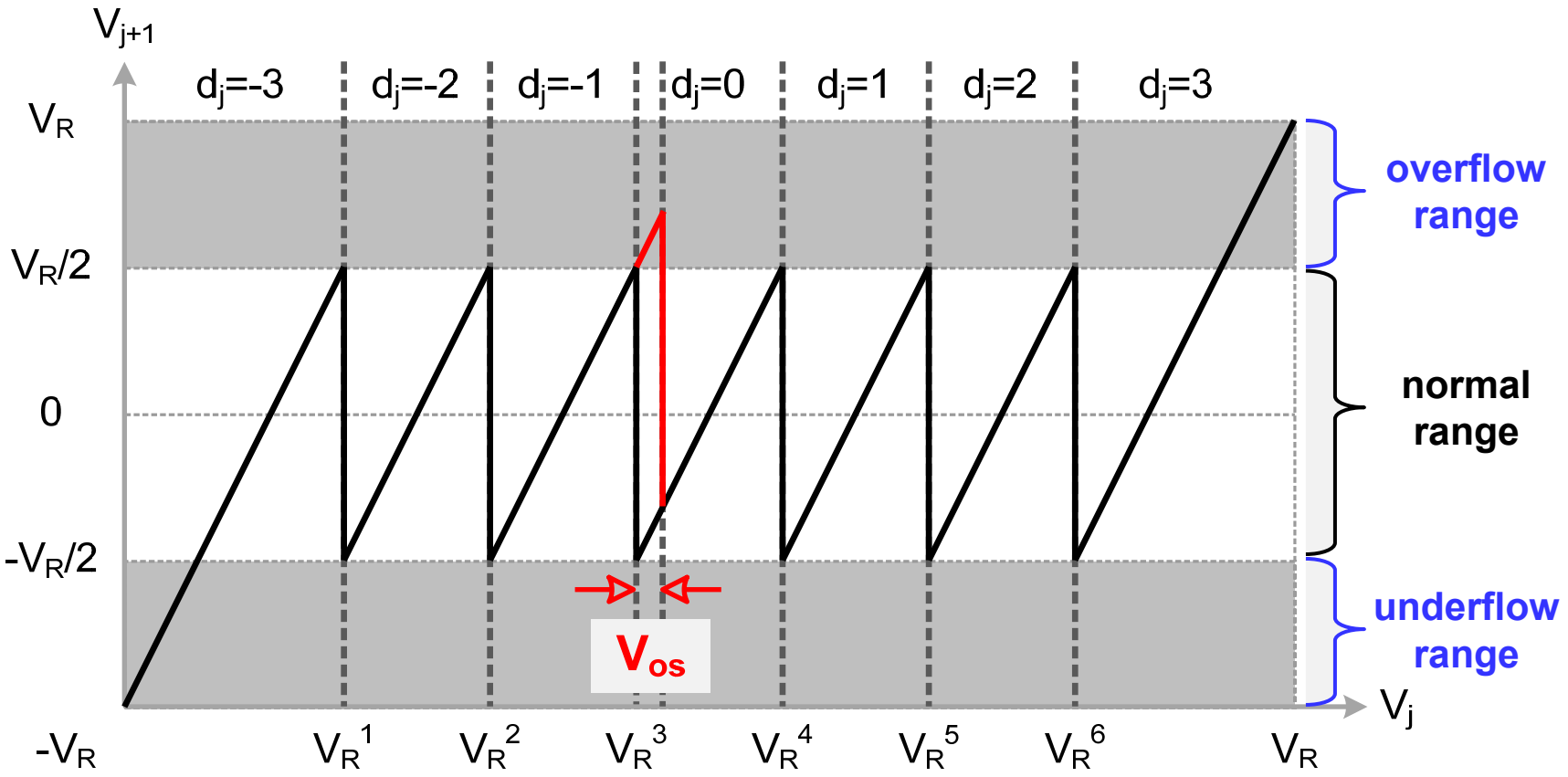
- Seven decision levels → ENOB ≈ $\log_2 7 = 2.807$
- Residue transfer function V_j (V_{j+1}) can be derived w/ charge conservation
- **Capacitor mismatch** and **amplifier gain error** are dominant error sources

Residue Transfer Function (2.5b Pipeline)

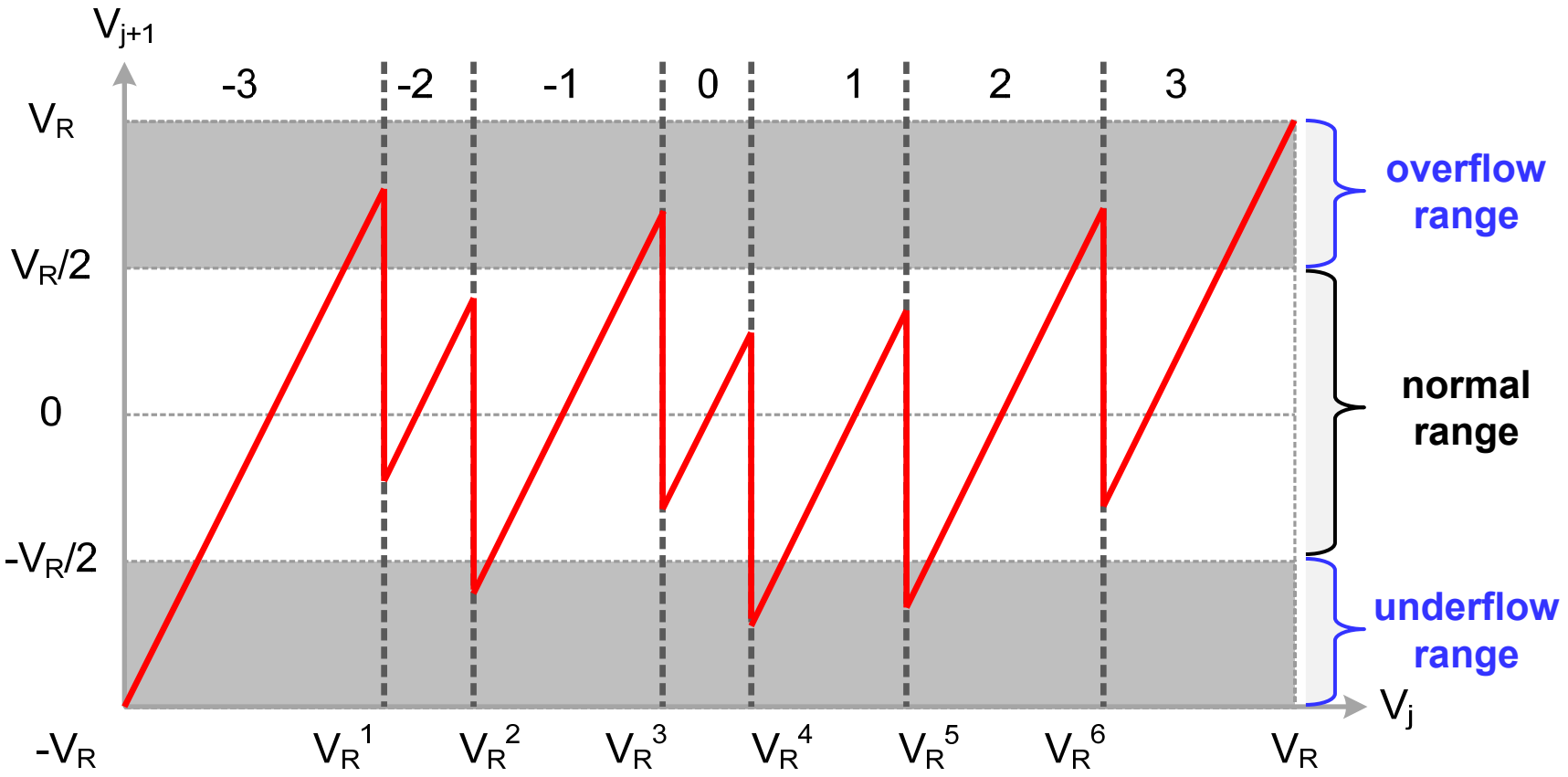


Only <u>half</u> of the internal dynamic range is used under ideal condition

What happens with comparator offset?

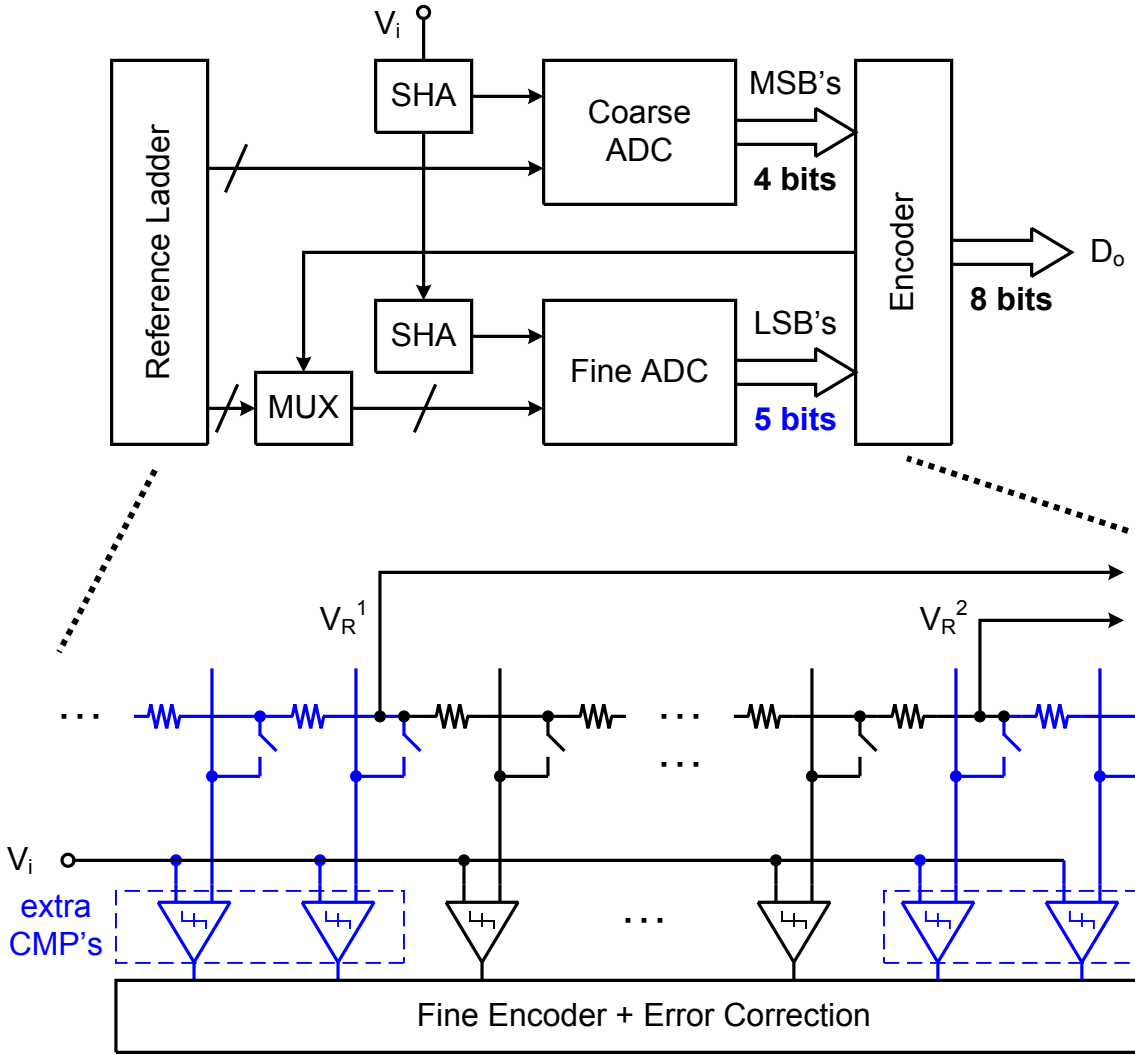


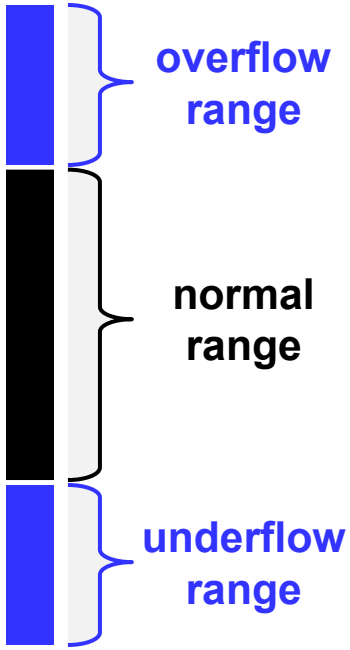
Internal Redundancy



Comparator and amplifier offsets are tolerated by internal redundancy

Redundancy in Subranging ADC





Equation for MDAC RTF 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 (EC EQ):

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

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

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

Bit-Weight (Radix) Correction

For 1b or 1.5b 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,

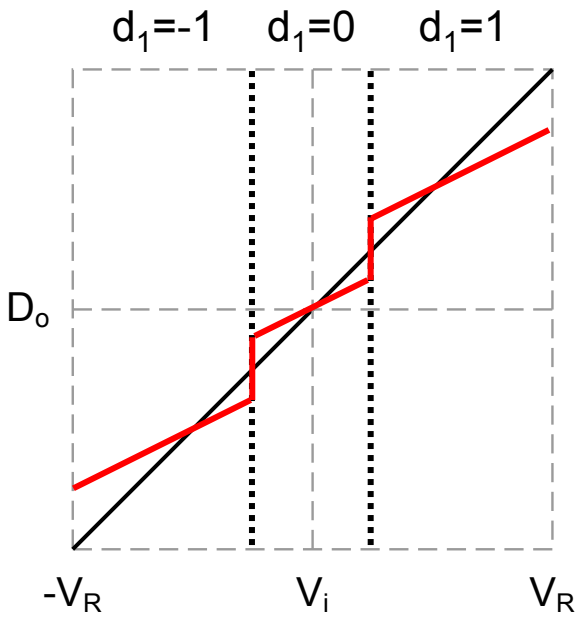
$$D_1 = D_{in}$$

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

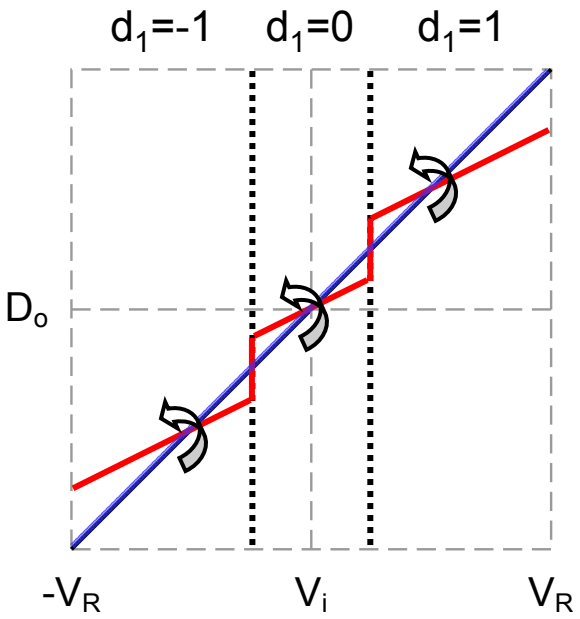
$$= \ldots + \frac{\left(d_j + d_j\Delta_j\right)}{2^j} + \frac{\left(d_{j+1} + d_{j+1}\Delta_{j+1}\right)}{2^{j+1}} + \frac{\left(d_{j+2} + d_{j+2}\Delta_{j+2}\right)}{2^{j+2}} + \ldots$$

☞ segmental offset

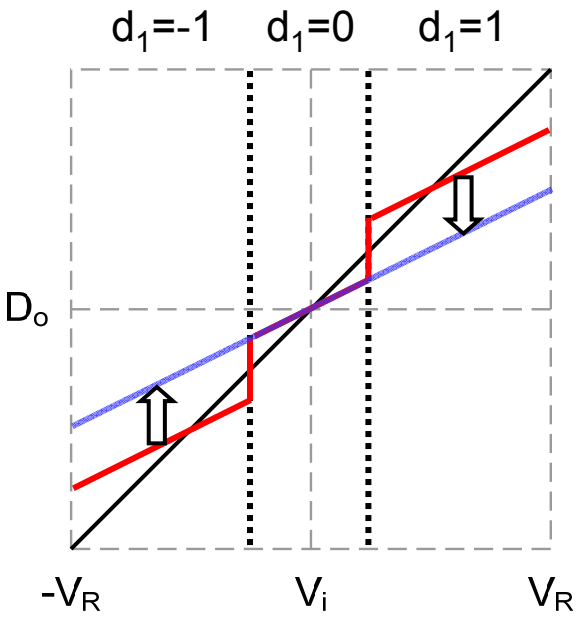
Bit-Weight (Radix) Correction



1.5b MDAC
residue nonlinearity



radix error:
needs multiplication



segmental offset:
addition only

Nonlinear MDAC Equation

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:

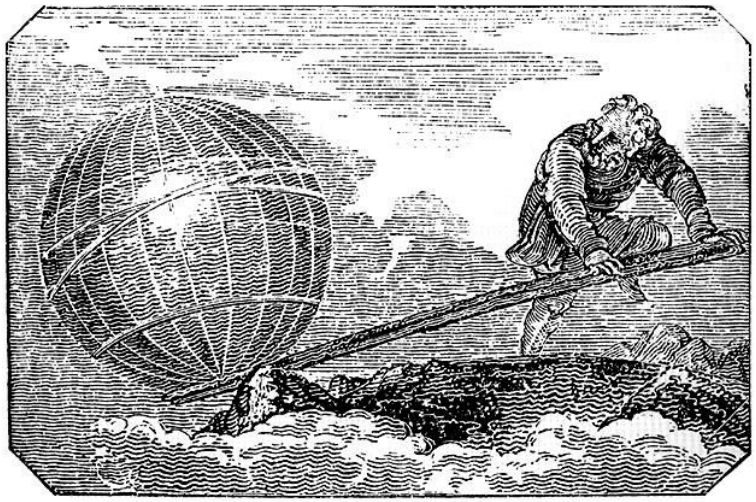
$$\Rightarrow \quad D_j = \left(d_{j,1} \cdot \beta_{j,1} + d_{j,2} \cdot \beta_{j,2} + d_{j,3} \cdot \beta_{j,3} \right) + f\left(D_{j+1}\right)$$

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

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

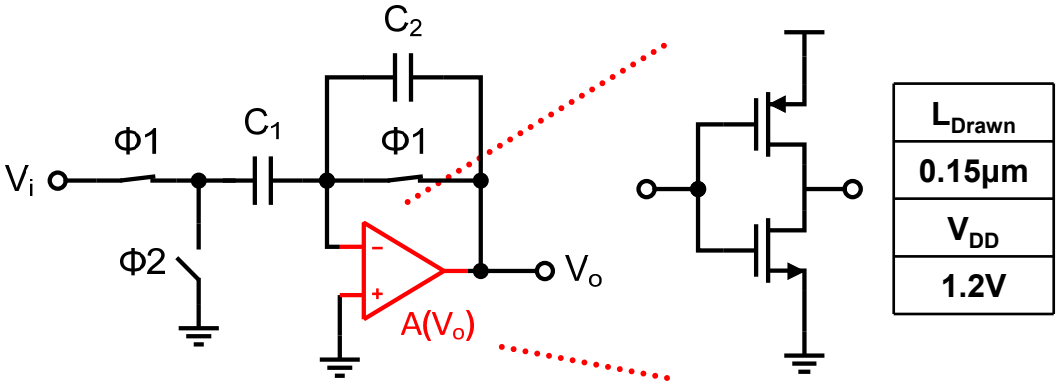
Let's push this approach…

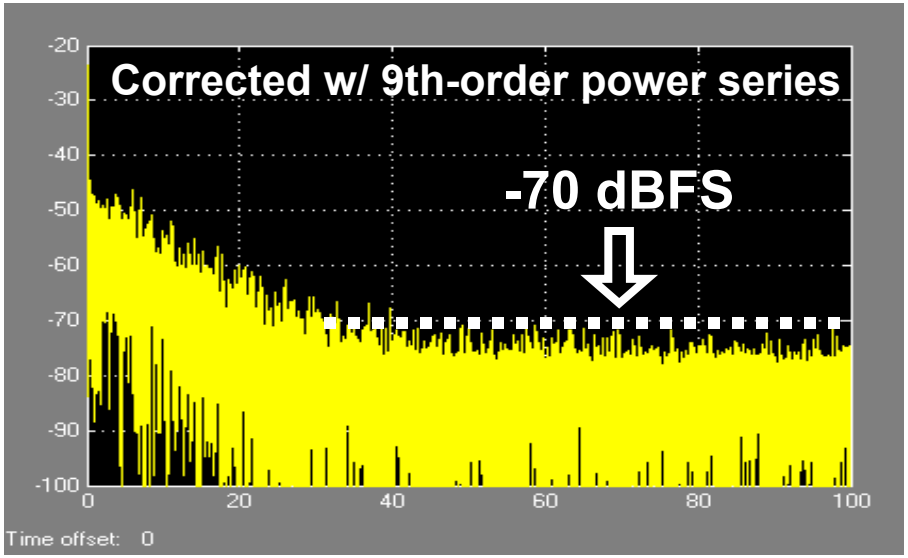
Give me a place to stand on, and I will move the Earth…



Archimedes, 200 BC

Correcting nonlinearity:





A few words on nonlinear correction

- Memoryless polynomial computation is efficient
  - A few coefficients fits/predicts full-range nonlinearity (requiring digital multipliers and adders mostly)
  - **Caveat: coefficients depend on signal statistics!**
  - **Caveat: 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…

Error-Parameter Identification

Foreground Calibration

- Foreground calibration
  - Test signal injected at input with normal conversion stopped
  - Often executed at system power-up
  - Incapable of tracking ambient variations
- Pseudo-background calibration
  - Skip-and-fill technique (Ref. [2])
  - Queue-base technique (Ref. [3])

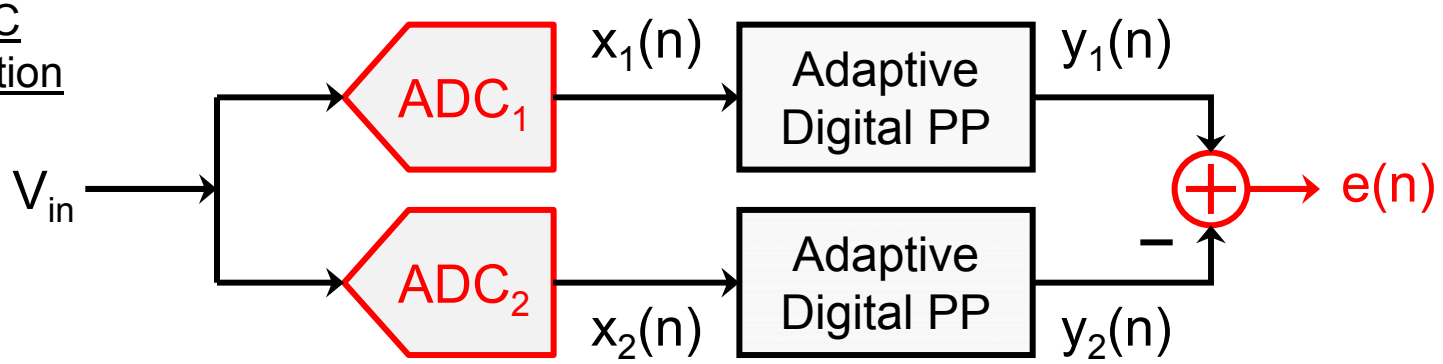
Background Calibration (Recent Trend)

- **Parameter extraction w/ PRBS (1b) injection**
  - Sub-DAC injection (DAC dithering)
  - Sub-ADC injection (comparator dithering)
  - Input injection (Independent Component Analysis)
- **Parameter extraction w/ two-ADC equalization**
  - Reference-ADC equalization (training sequence)
  - Split-ADC equalization (blind)
  - Offset double conversion (ODC) (blind, single ADC)

Model parameter extraction is what the game is all about…

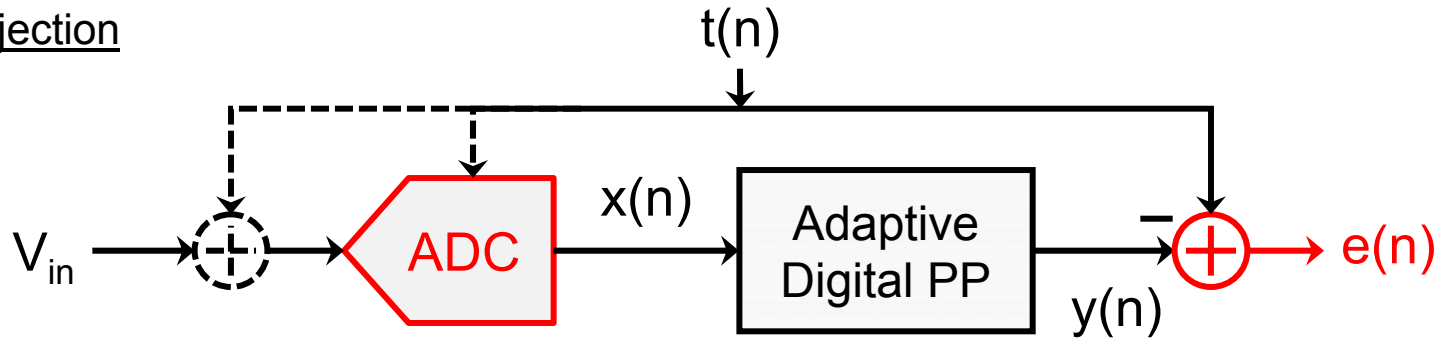
Background Calibration (Recent Trend)

Two-ADC equalization



Lewis (03), Chiu (04), McNeill (05), et al.

PRBS injection (Dither)



Temes (98, 00), Lewis (98), Galton (00), et al.

Digital Background Calibration Techniques

Method	Parameter	Test signal	Injection point	Reference†
DNC + GEC	$\{ \beta_{j,k}, \alpha_{j,m} \}$	multi PRBS	sub-DAC	[7–12]
Split capacitor	$\{ \Delta_j \}$	1 PRBS	sub-DAC	[13, 14]
Sig.-dep. dither	$\{ \gamma_j \}$	1 PRBS	sub-DAC	[15]
GEC + SA	$\{ \gamma_j \}$	2 PRBS	sub-ADC	[16, 17]
Statistics	$\{ \alpha_{j,m} \}$	1 PRBS	sub-ADC	[18, 19]
Fast GEC	$\{ \gamma_j \}$	1 PRBS	sub-ADC	[20]
ICA	$\{ \gamma_j \}, \{ \alpha_{j,m} \}$	1 PRBS	input	[21–23]
Ref. ADC	$\{ \beta_{j,k}, \alpha_{j,m} \}$	n/a	n/a	[24–27]
Virtual ADC	$\{ \beta_{j,k}, \alpha_{j,m} \}$	offset	sub-DAC	[28, 29]
Split ADC	$\{ \alpha_{j,m} \}$	n/a	n/a	[30, 31]
ODC	$\{ \gamma_j \}$	offset	input	[32, 33]

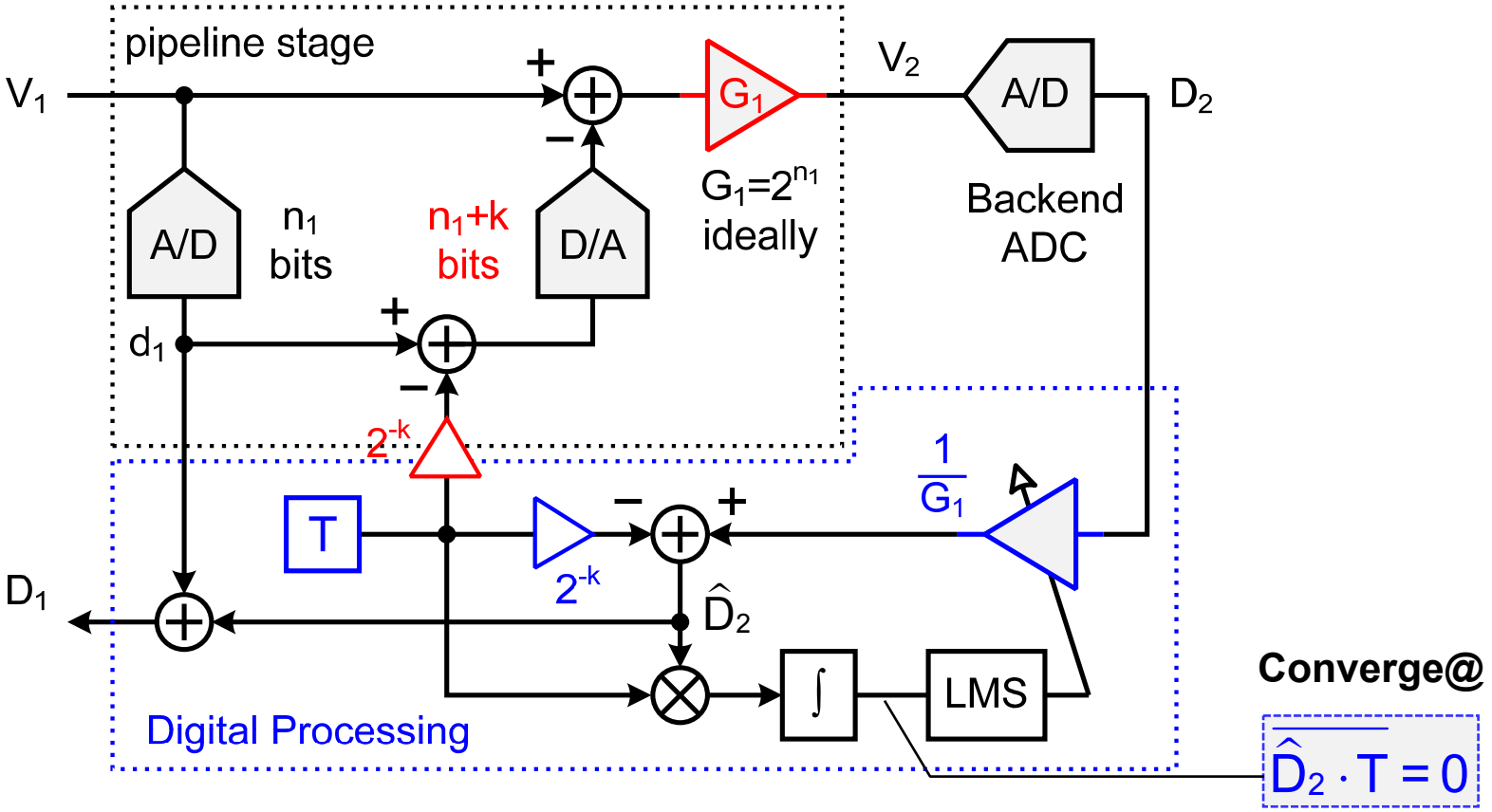
† References are furnished at the end of the slides

PRBS Injection Techniques

Comparison of PRBS Injection Techniques

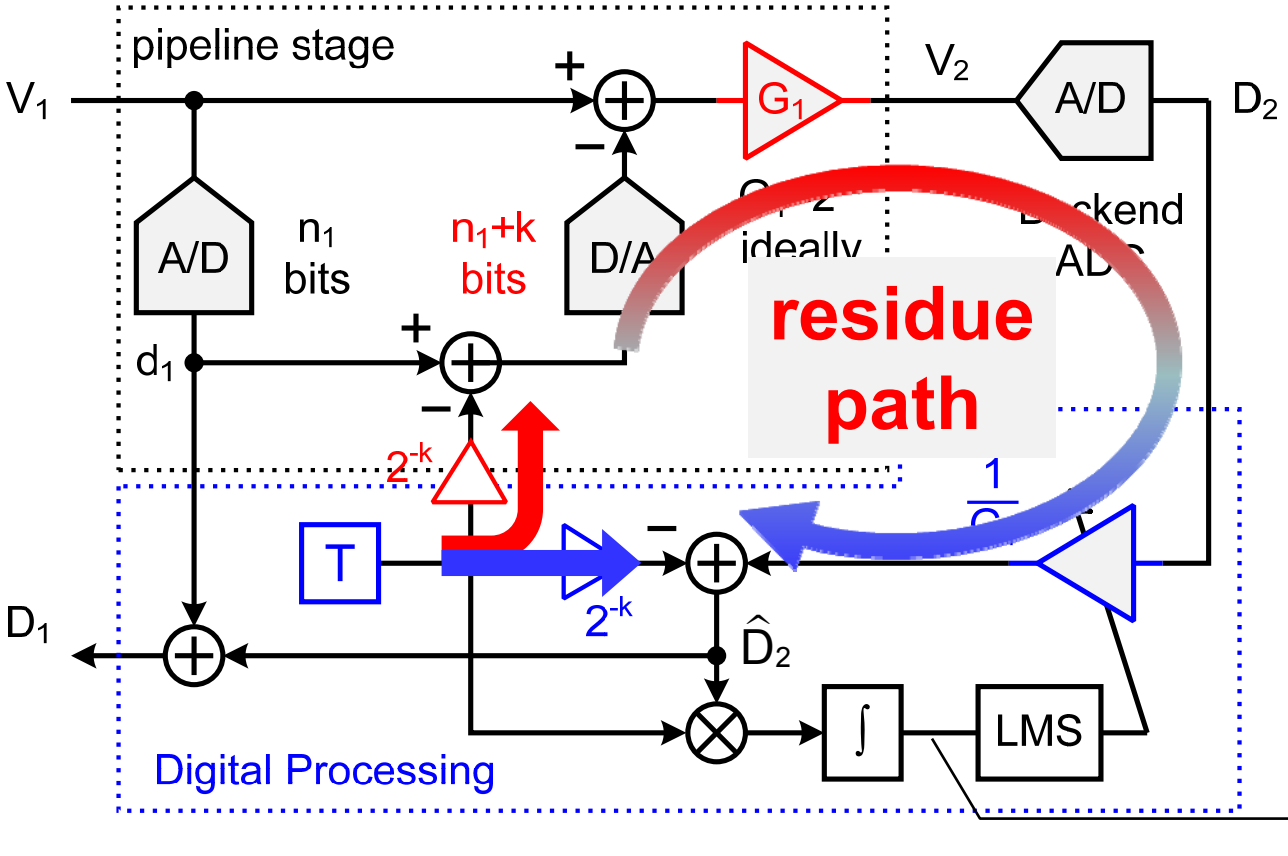
- **Sub-DAC injection**
  - needs to be removed in digital output
  - higher sub-DAC resolution (injection and DAC matching req'd)
  - can work with quiet input
- **Sub-ADC injection**
  - considered as dynamic comparator offset, no removal needed
  - higher sub-ADC resolution (injection and ADC matching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

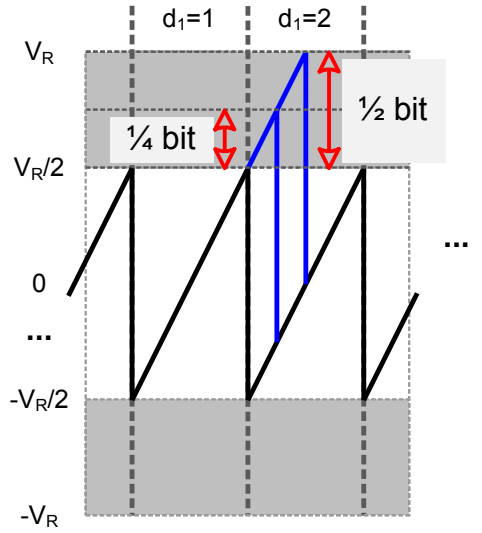
Sub-DAC Injection – residue gain correction



- In steady state, analog gain (G_1) and digital gain (G_1^{-1}) cancel exactly
- $k \leq 1/4$ 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 – residue gain correction



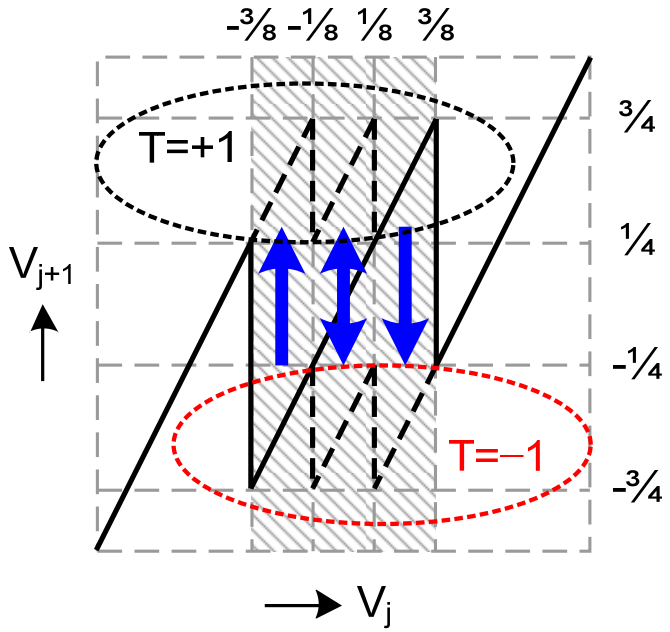


Converge@

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

- In steady state, analog gain (G_1) and digital gain (G_1^{-1}) cancel exactly
- $k \le ¼$ 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 – signal-dependent dither

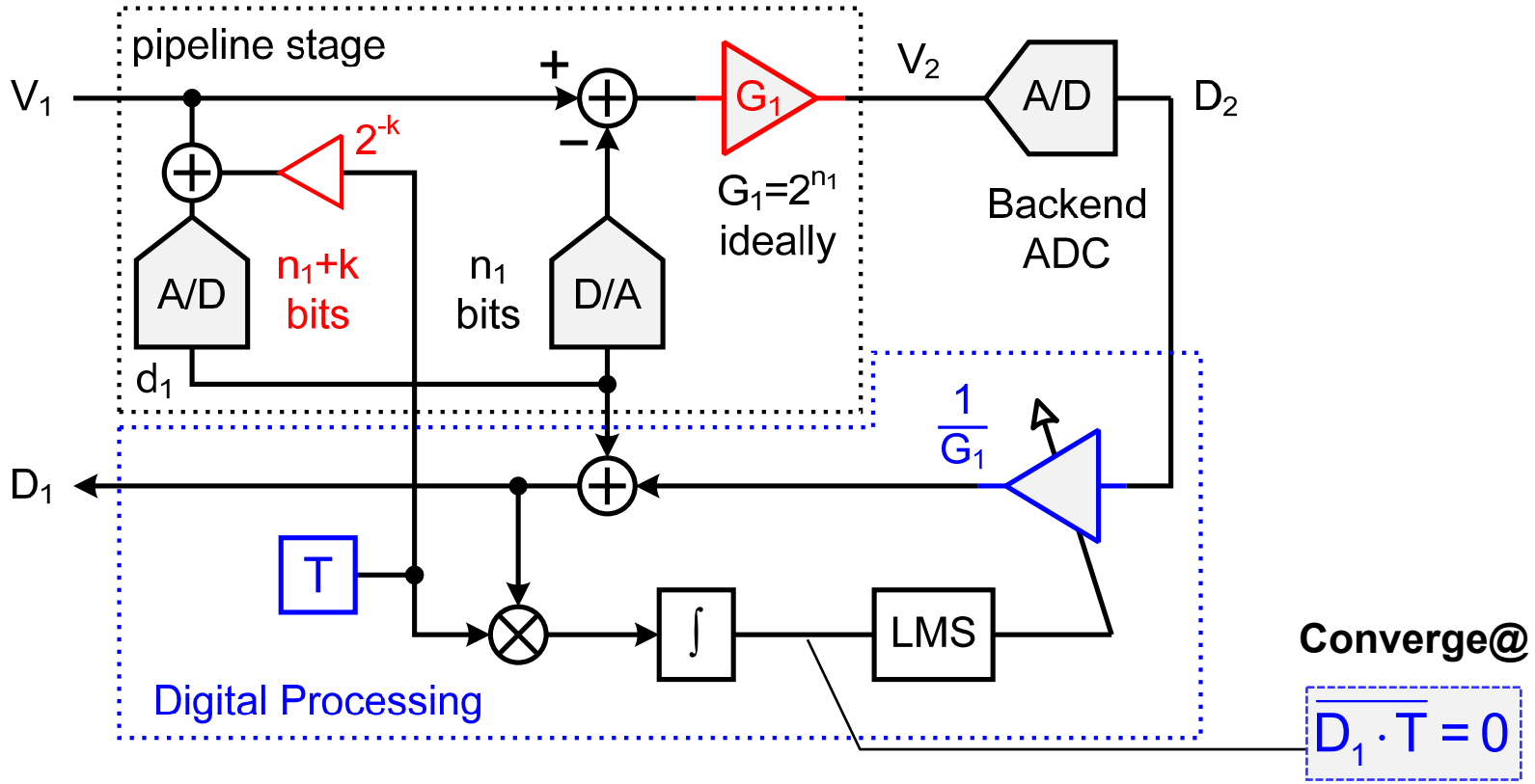


PRBS Injection Table

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

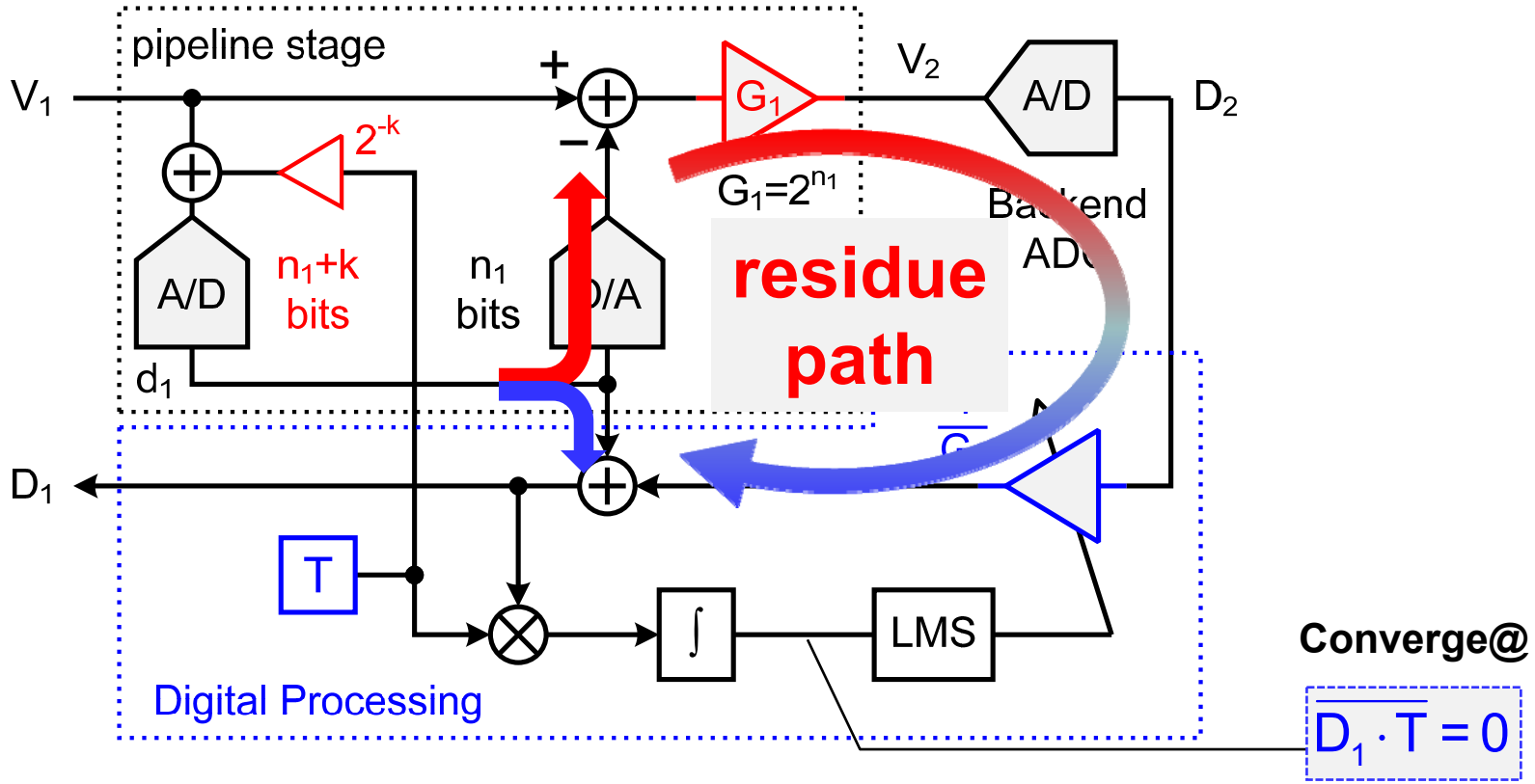
- PRBS only injected when input falls within the shaded region
- Extra comparator thresholds needed to instrument the SDD

Sub-ADC Injection – comparator dither



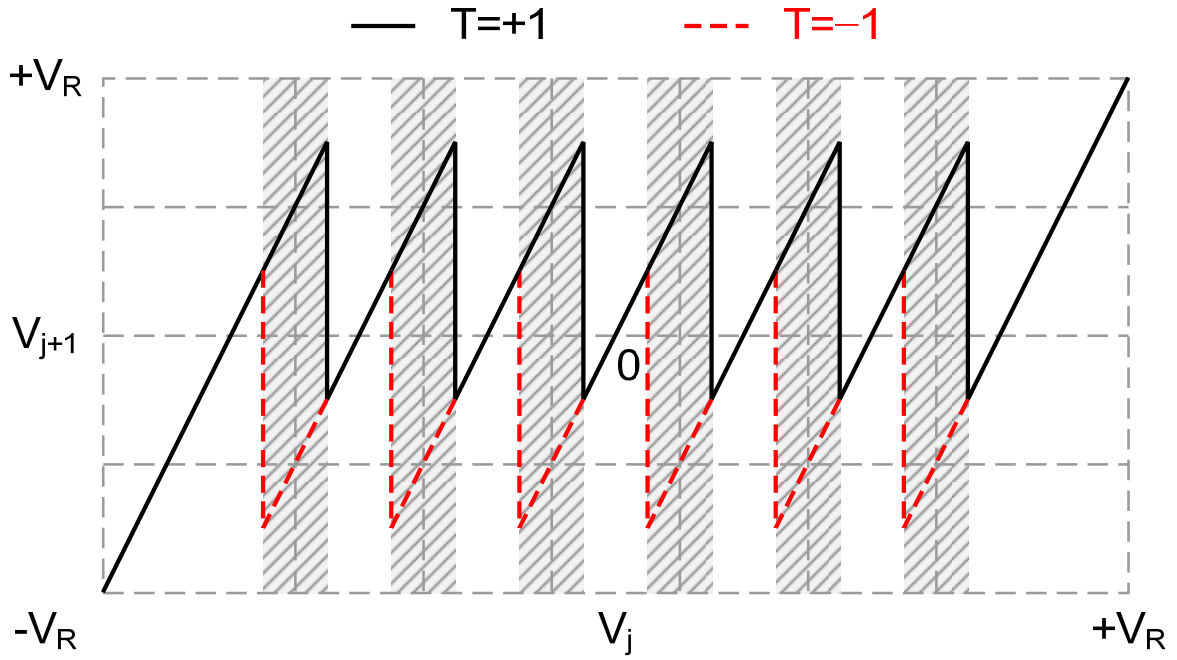
- In steady state, analog gain (G_1) and digital gain (G_1^{-1}) cancel exactly
- $k \leq 1/4$ to avoid overflow in residue output
- No need to match injection bit scaling factor (2^{-k}) to the sub-ADC thresholds

Sub-ADC Injection – comparator dither



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

Internal Redundancy Revisited

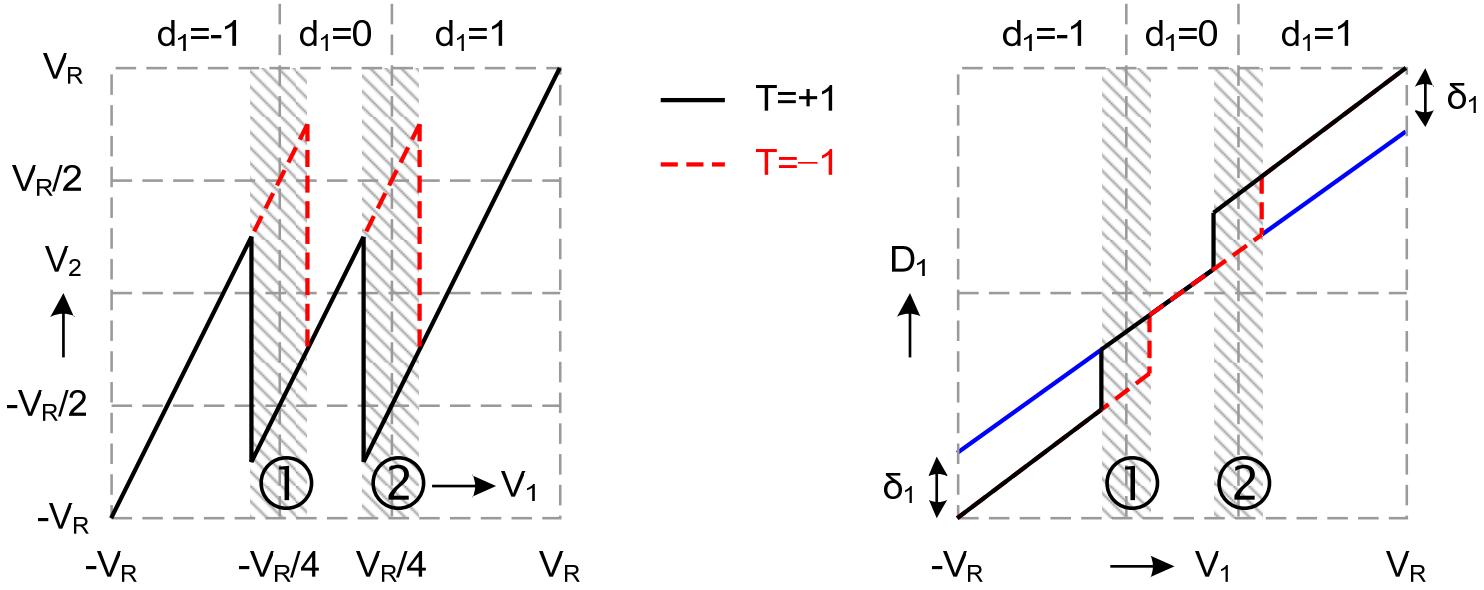


Ref. [34]

- 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 !!

Inter-stage gain error identification

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 $T = -1$, $D_1 = D_{ideal}$

Inter-stage gain error identification

Calculating correlation:

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

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

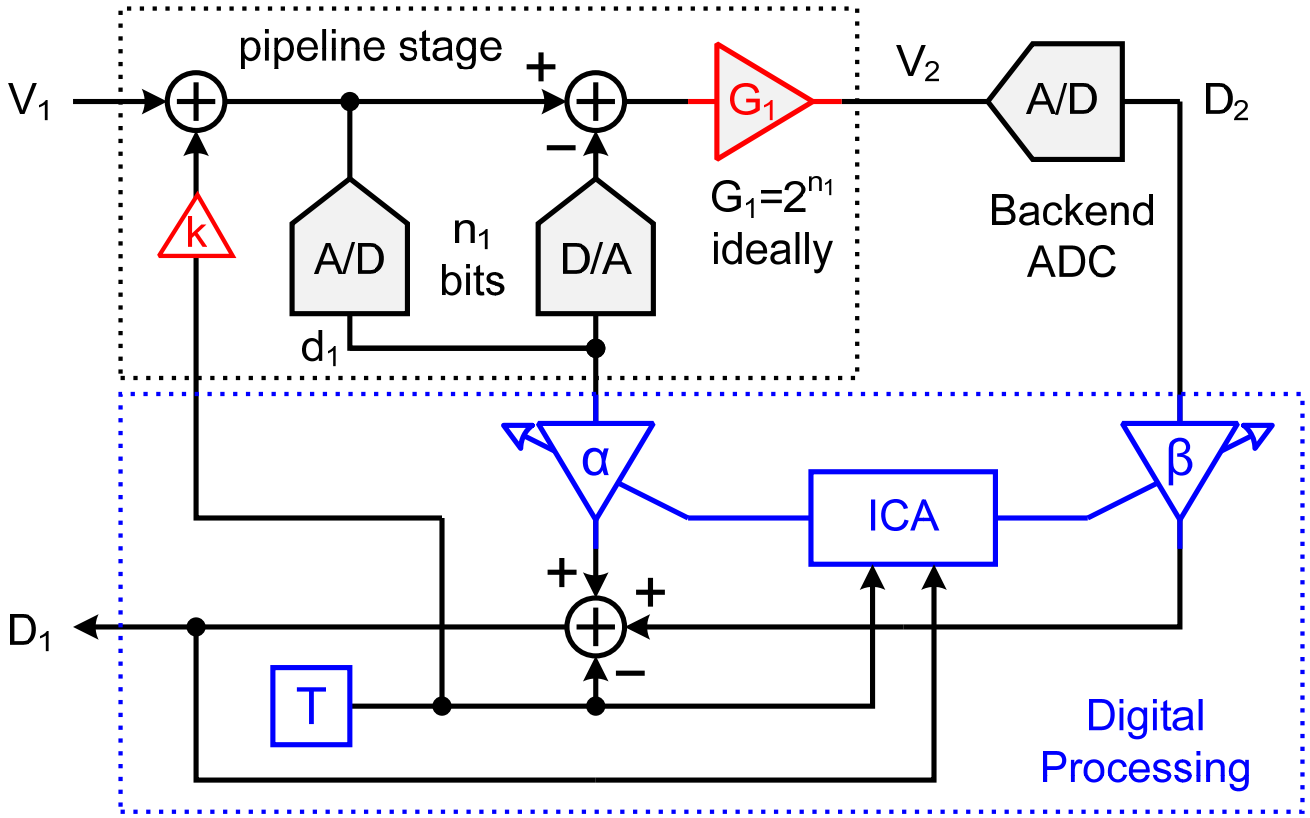
$$= \frac{1}{2}\delta_1 \cdot \Pr\left(V_1 \in \{\text{region 1 or 2}\}\right)$$

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

Direct Input Injection



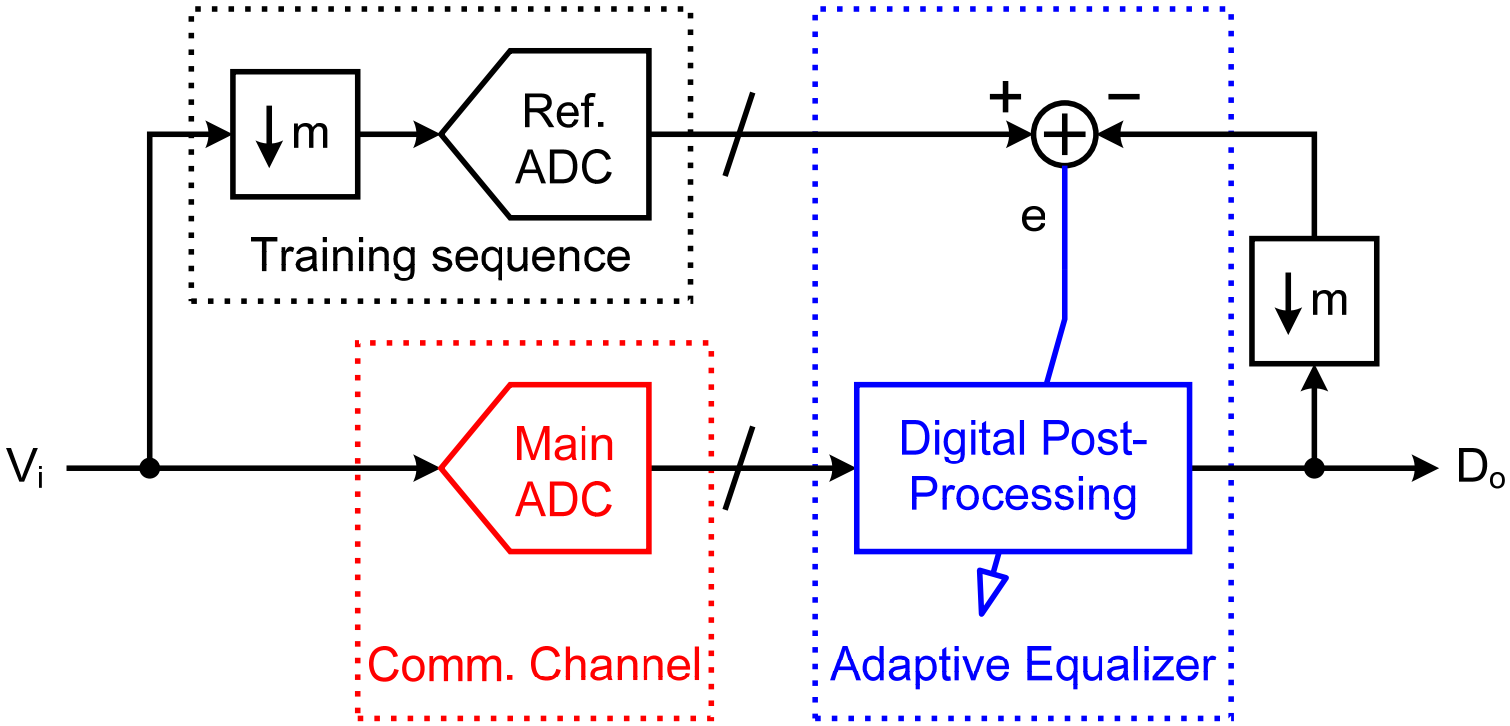
- Algorithm works reliant on the independence b/t input and T, a stronger statement than simply being “uncorrelated”
- Multiple parameter extraction is possible with Independent Component Analysis (ICA)

Equalization Techniques

Comparison of Equalization Techniques

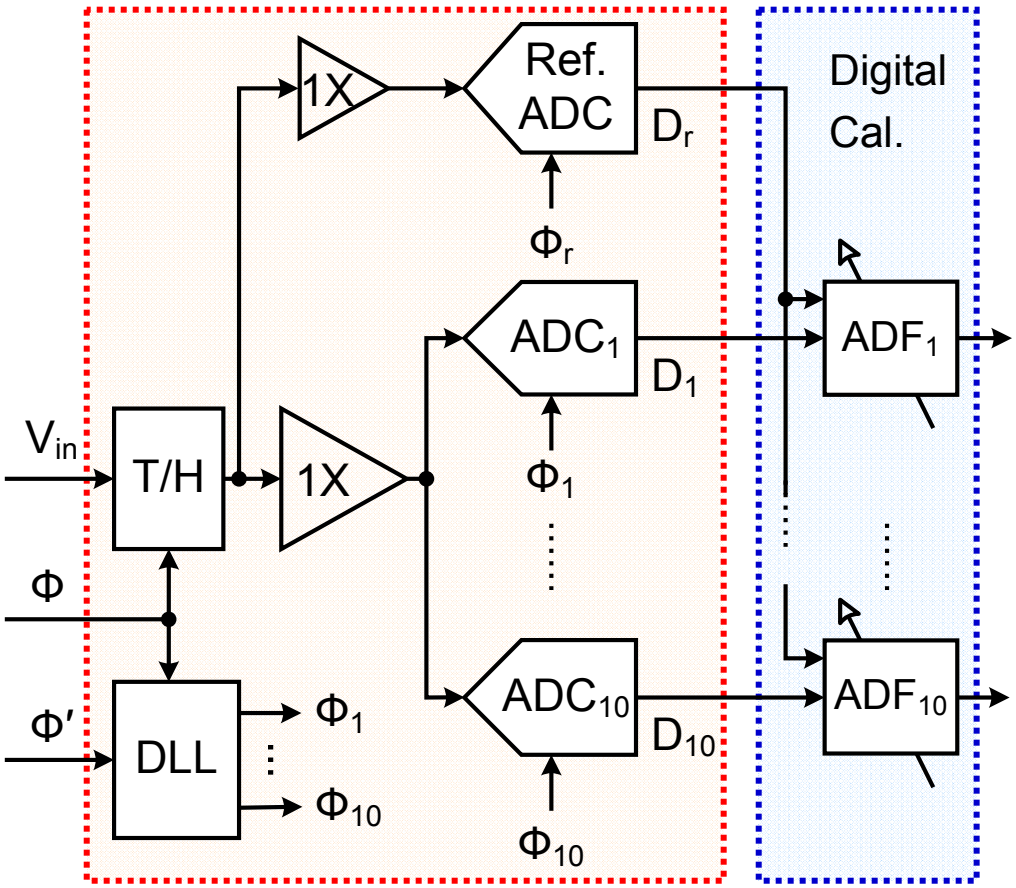
- **Reference-ADC equalization**
  - Slow-Fast two-ADC architecture to accomplish accuracy and throughput simultaneously using adaptive equalization
  - Two (different) ADC's needed, subject to skew error without SHA
- **Split-ADC equalization**
  - Two almost identical ADC's employed for blind equalization
  - 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

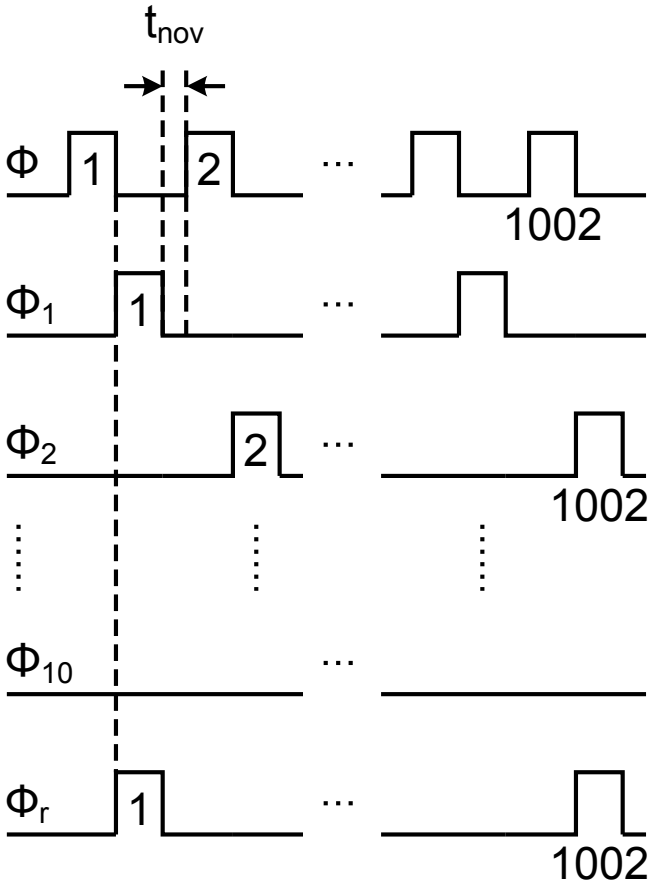
Reference-ADC Equalization



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

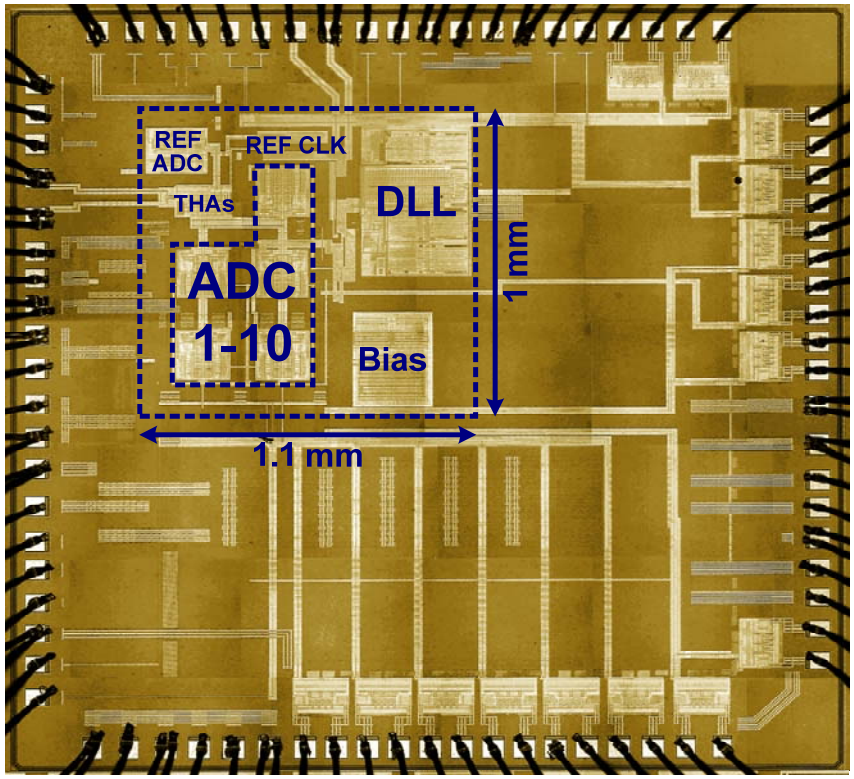
EQZ of Time-Interleaved ADC Array





All paths are aligned to the unique ref. ADC after equalization

600MS/s TI-ADC Array Achieving >60dB SFDR



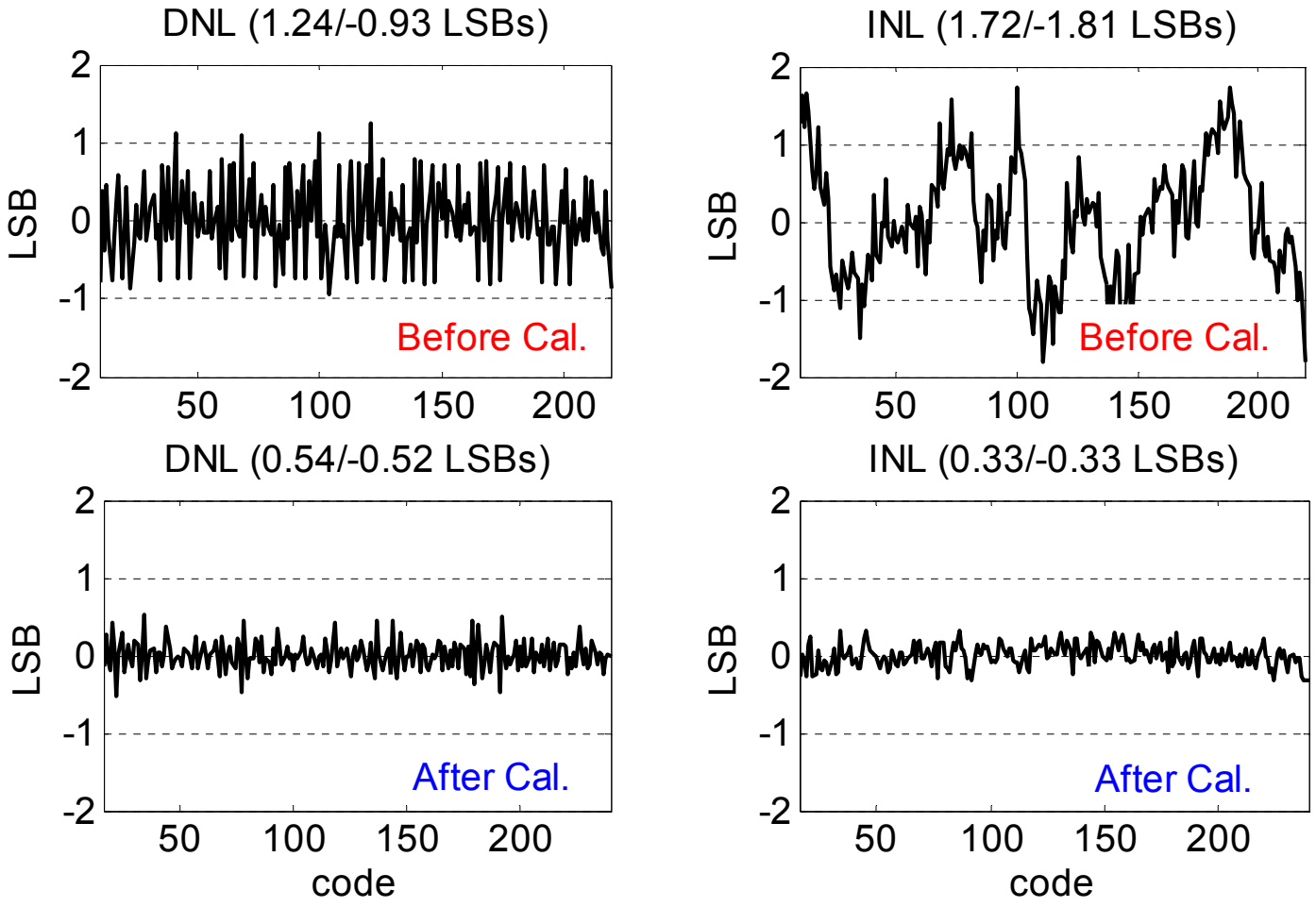
Die photo

Performance Comparison
(@ publication time)

CMOS ADC's	Process	Speed [MS/s]	SFDR [dB]	FoM [fJ/step]
ISSCC'06	0.13µm	600	43	220
ISSCC'08	0.13µm	1250	48	480
VLSI'08	65nm	800	58	280
This work ISSCC'09	**0.13µm**	**600**	**65**	**210**

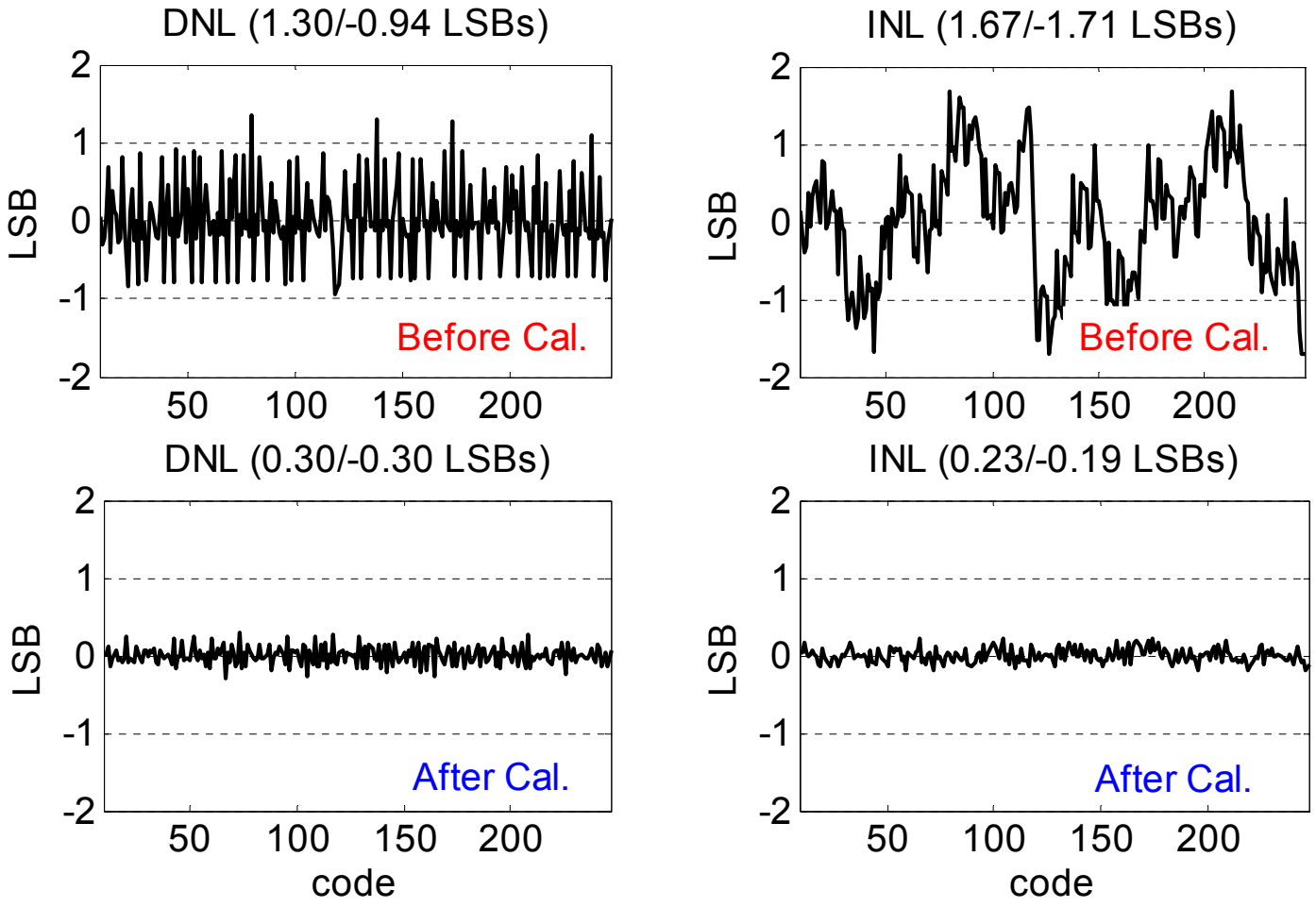
Ref. [35]

ADC Array EQZ – Measured Linearity (#3)



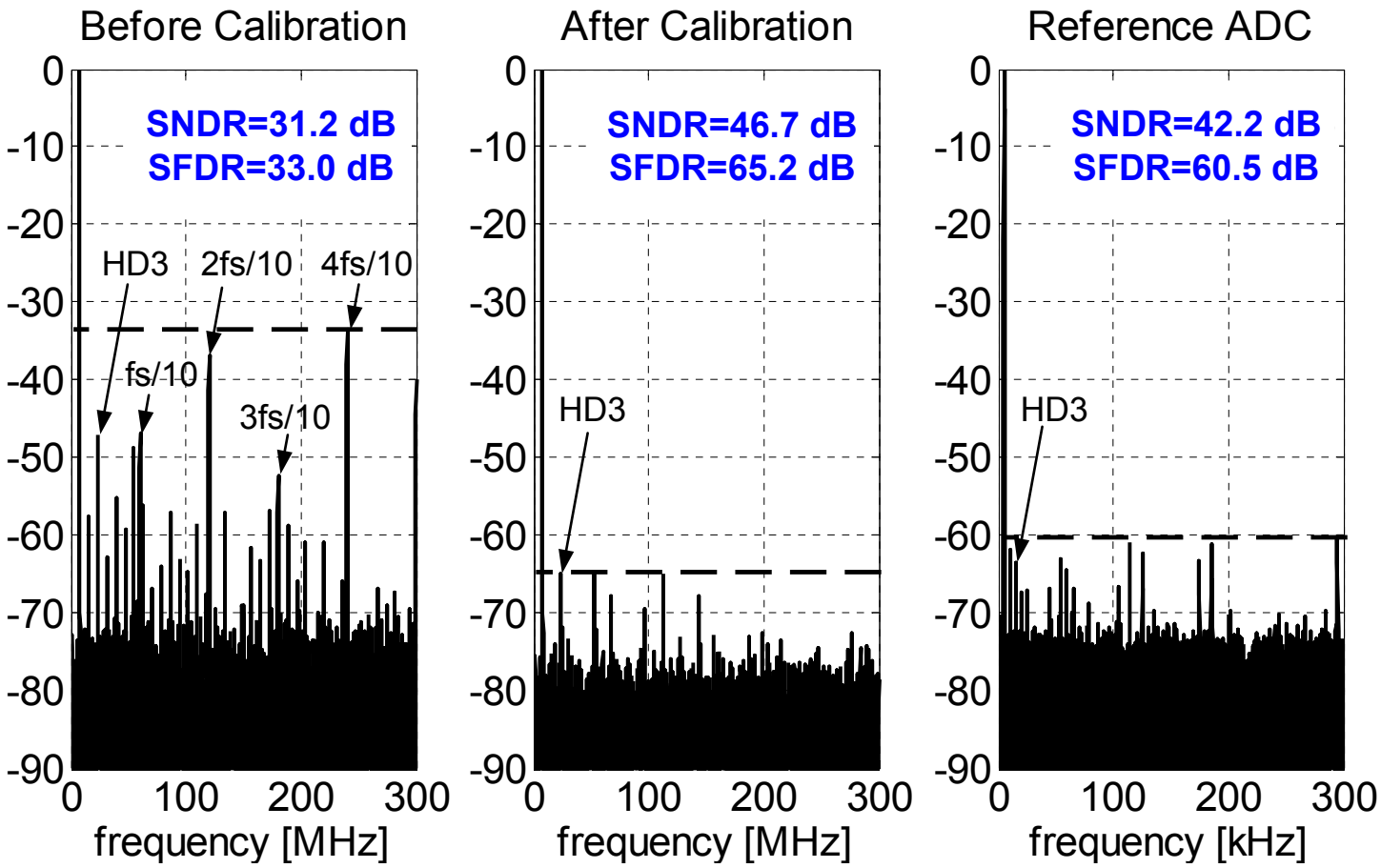
(f_s = 600MS/s, f_{in} = 1.8MHz, A_{in} = 0.9FS, 100k samples)

ADC Array EQZ – Measured Linearity (array)



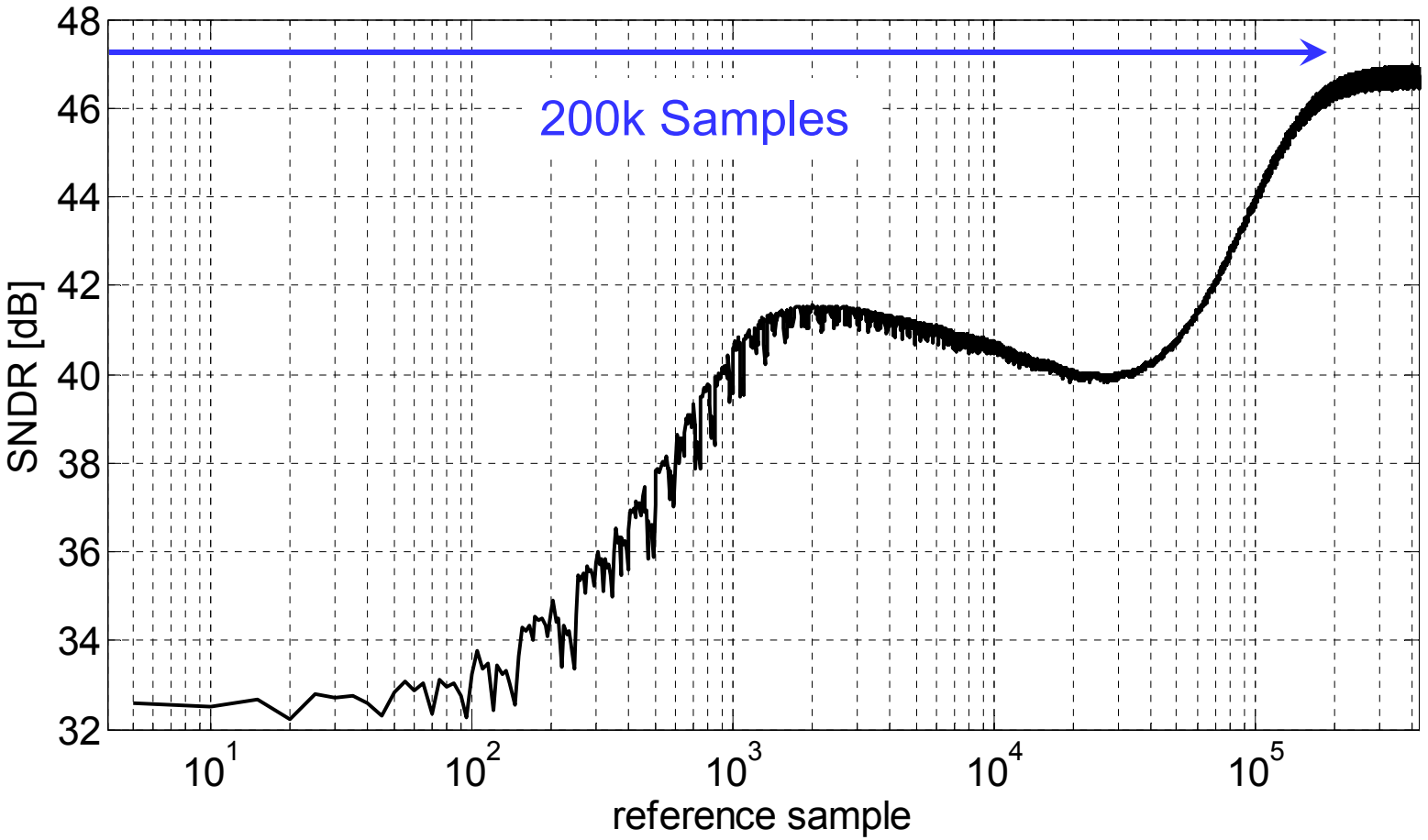
(f_s = 600MS/s, f_{in} = 1.8MHz, A_{in} = 0.9FS, 100k samples)

ADC Array EQZ – Measured Spectrum

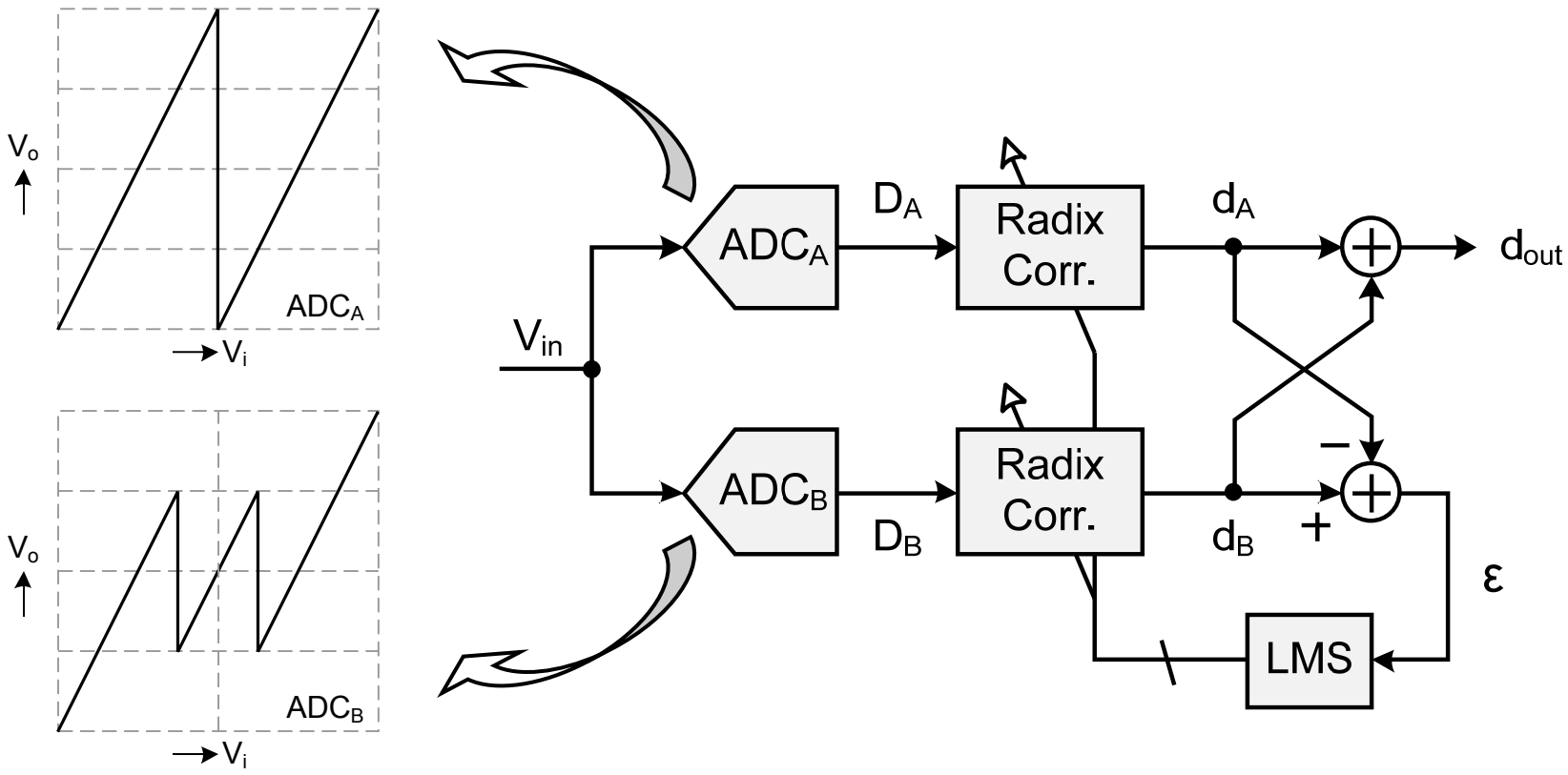


(f_s = 600MS/s, f_{in} = 7.8MHz, A_{in} = 0.9FS, 16k samples)

ADC Array EQZ – Convergence Speed

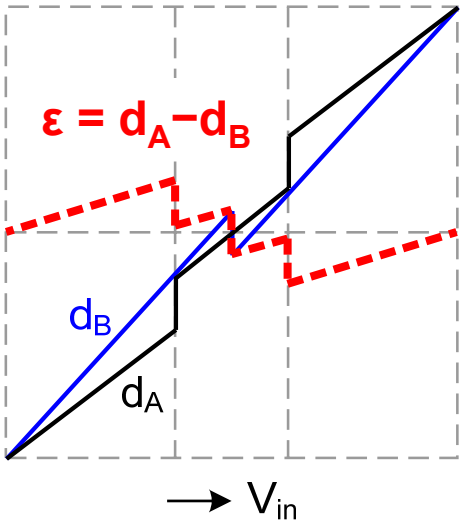


Split-ADC Equalization

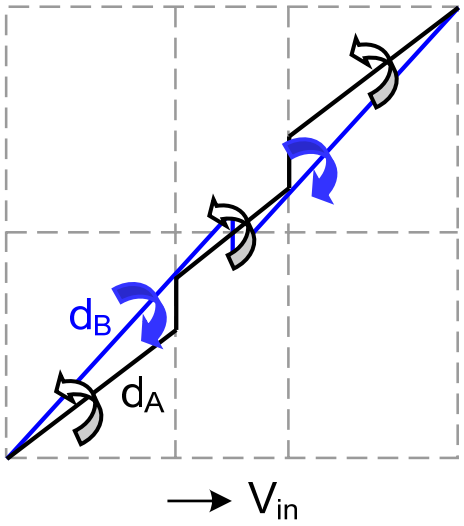


- Blind equalization w/o reference possible by offsetting the RTF's
- Fast convergence due to zero-forcing equalization (vs. de-correlation)

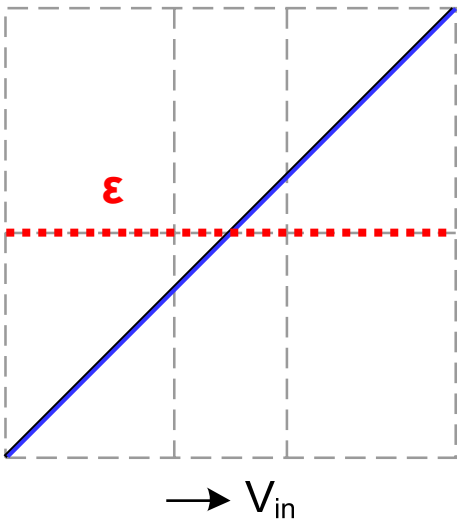
Split-ADC Equalization – Zero Forcing



Error observation

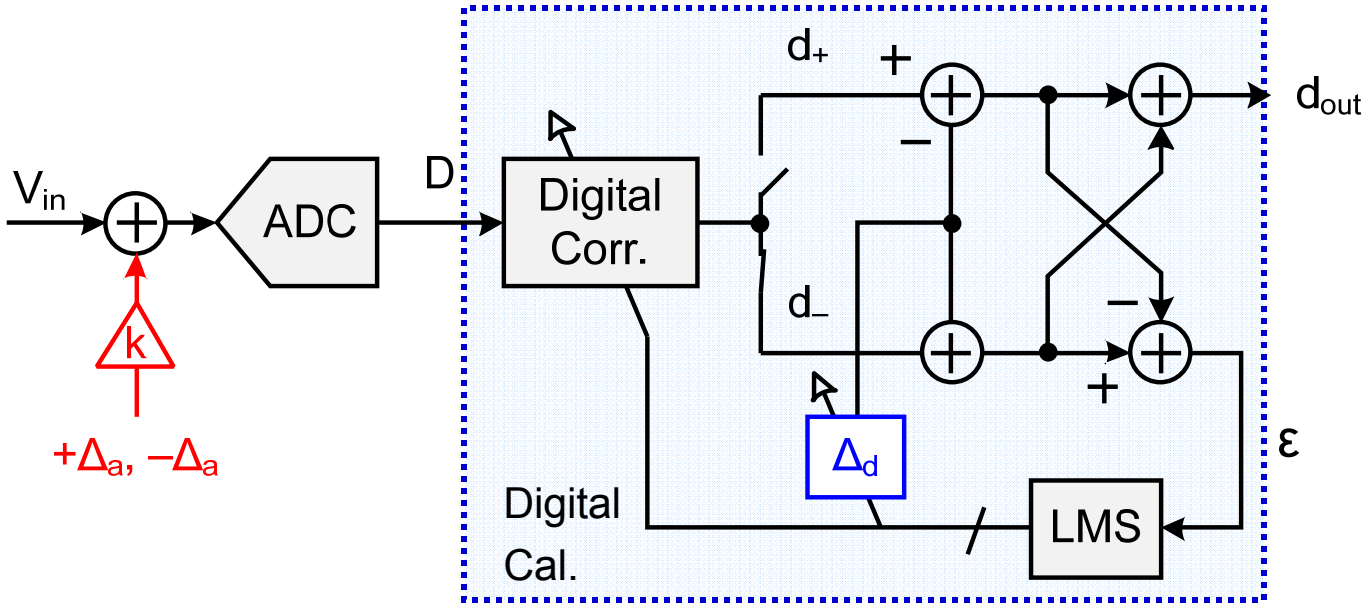


Radix correction



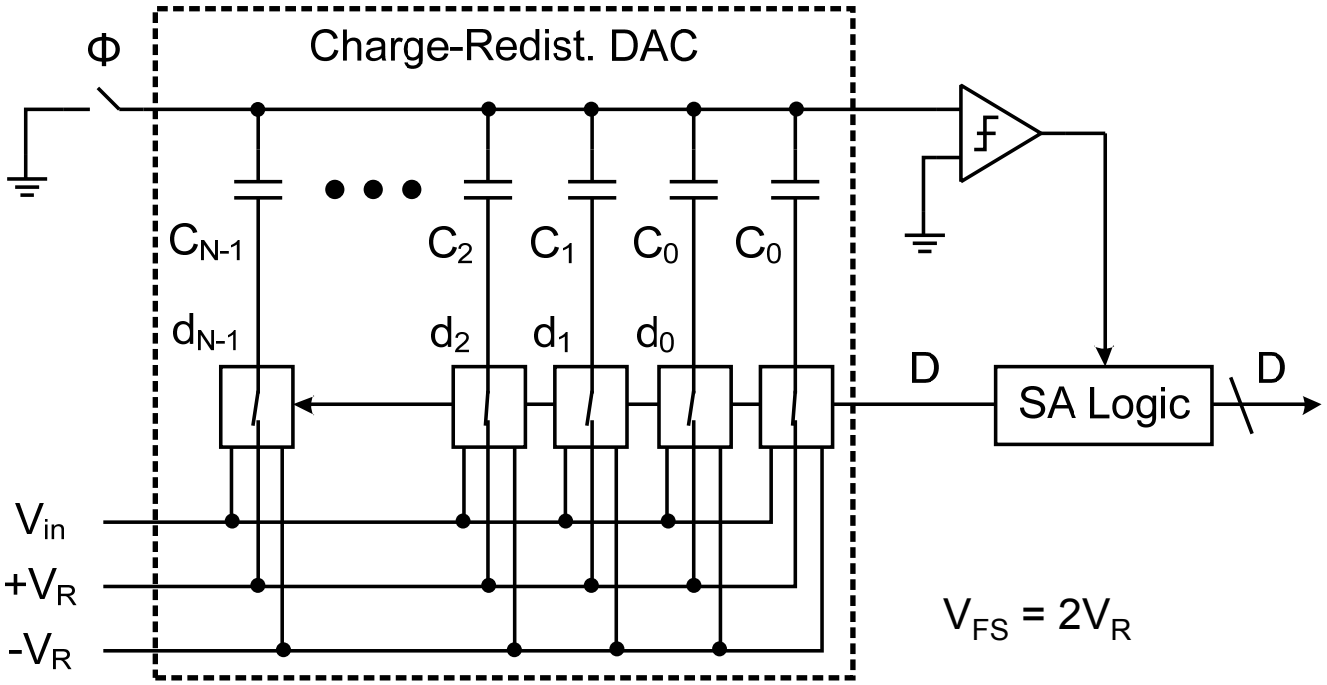
Zero-forcing EQZ

Self-Equalization – Offset Double Conversion



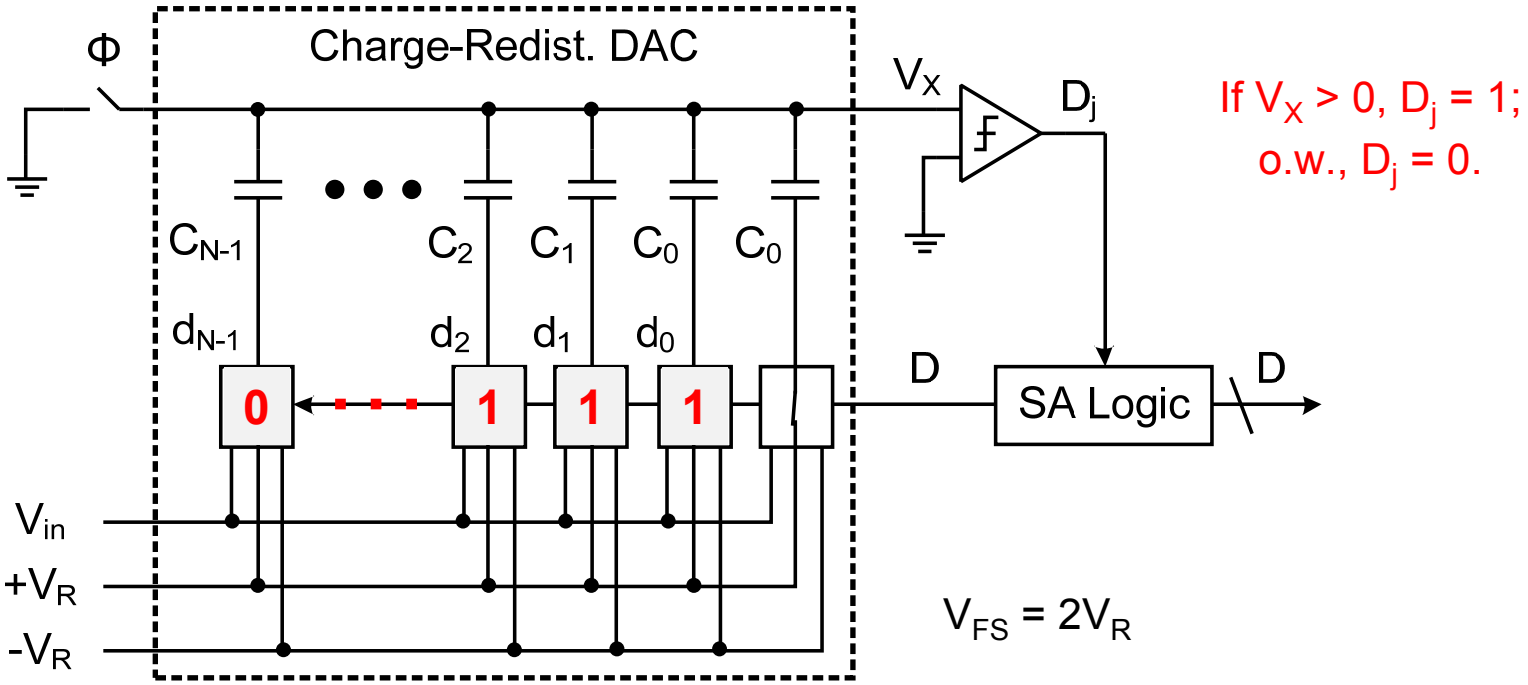
- Every sample is converted twice w/ opposite offsets injected (ODC)
- Self-equalization, hardware efficient, no skew issue, half throughput
- Simultaneous multiple parameter learning, zero-forcing, very fast

The Return of SAR ADC



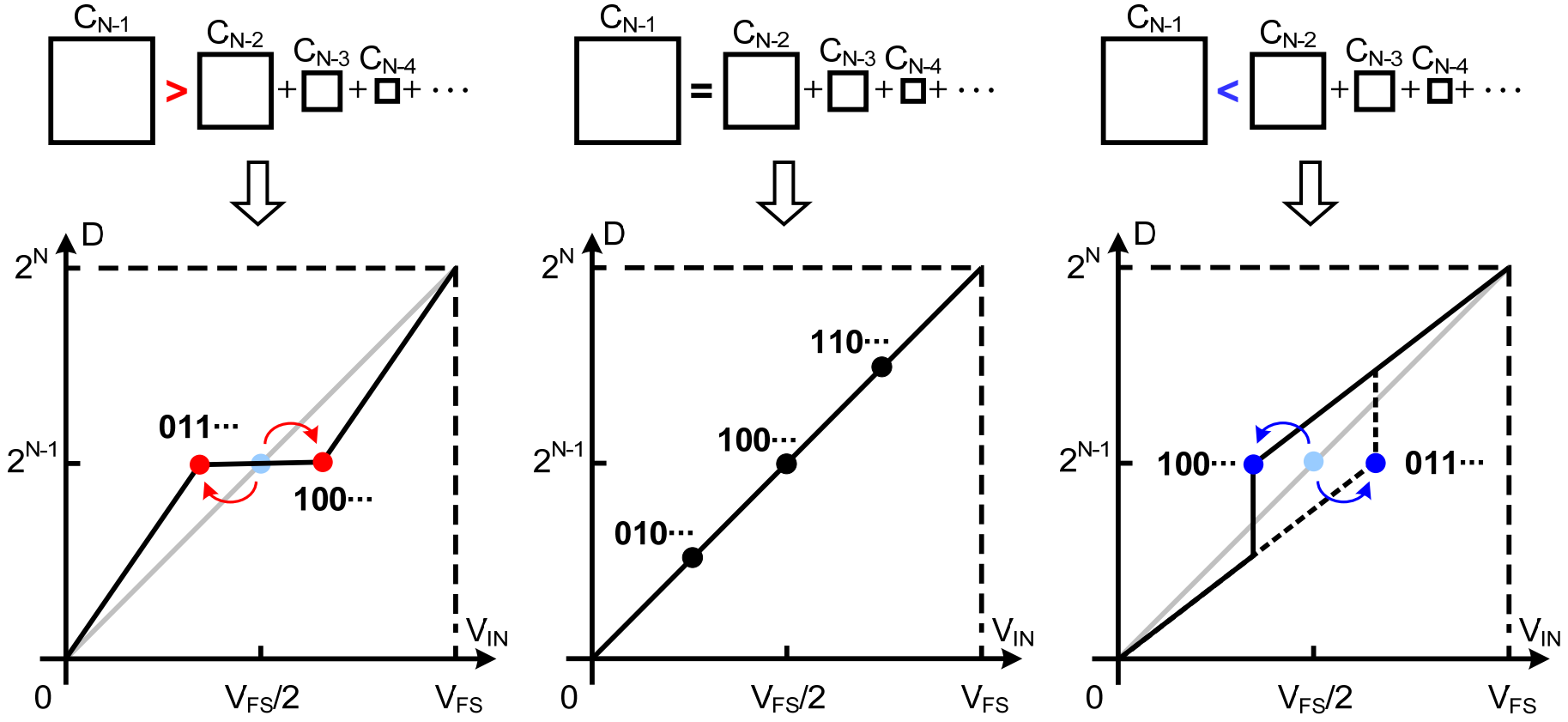
- Single ZX comparator → insensitive to offset and nonlinearity
- All-switching analog architecture → scaling friendly, low power and area
- Op-amp free → rail-to-rail swing, inherently linear operation

Binary Search



- Single ZX comparator → insensitive to offset and nonlinearity
- All-switching analog architecture → scaling friendly, low power and area
- Op-amp free → rail-to-rail swing, inherently linear operation

SAR Conversion Redundancy



Super-binary
(no redundancy)

Binary
(no redundancy)

Sub-binary
(redundancy)

Sub-Binary Bit-Weight Correction

Charge balance:

$$d_{out} = \left\lfloor \frac{V_{in}}{V_{FS}} \right\rfloor$$

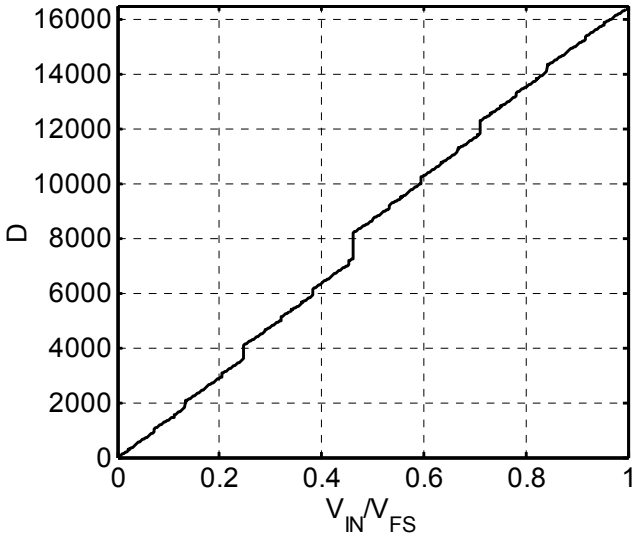
$$= \sum_{j=0}^{N-1} \left(\frac{C_j}{\sum C_j} \right) \cdot \left(2D_j - 1 \right) + \frac{V_{os}}{V_{FS}} + QN$$

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

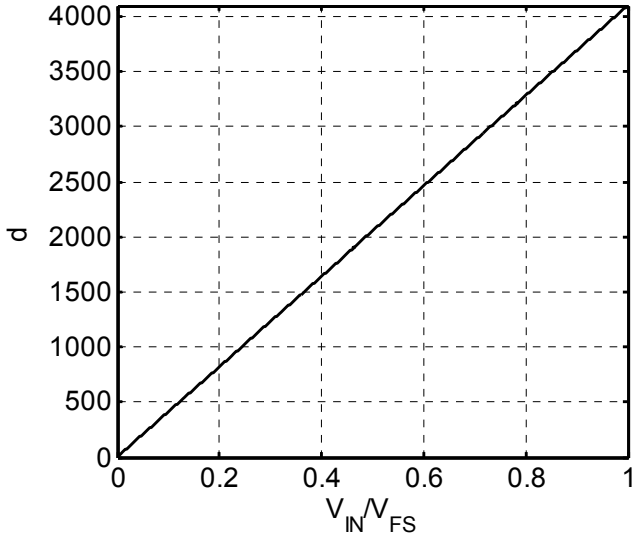
☝ N-tap linear equalizer

$D_j = \{ 0,1 \}, \quad \{ w_j \}$ = bit weights

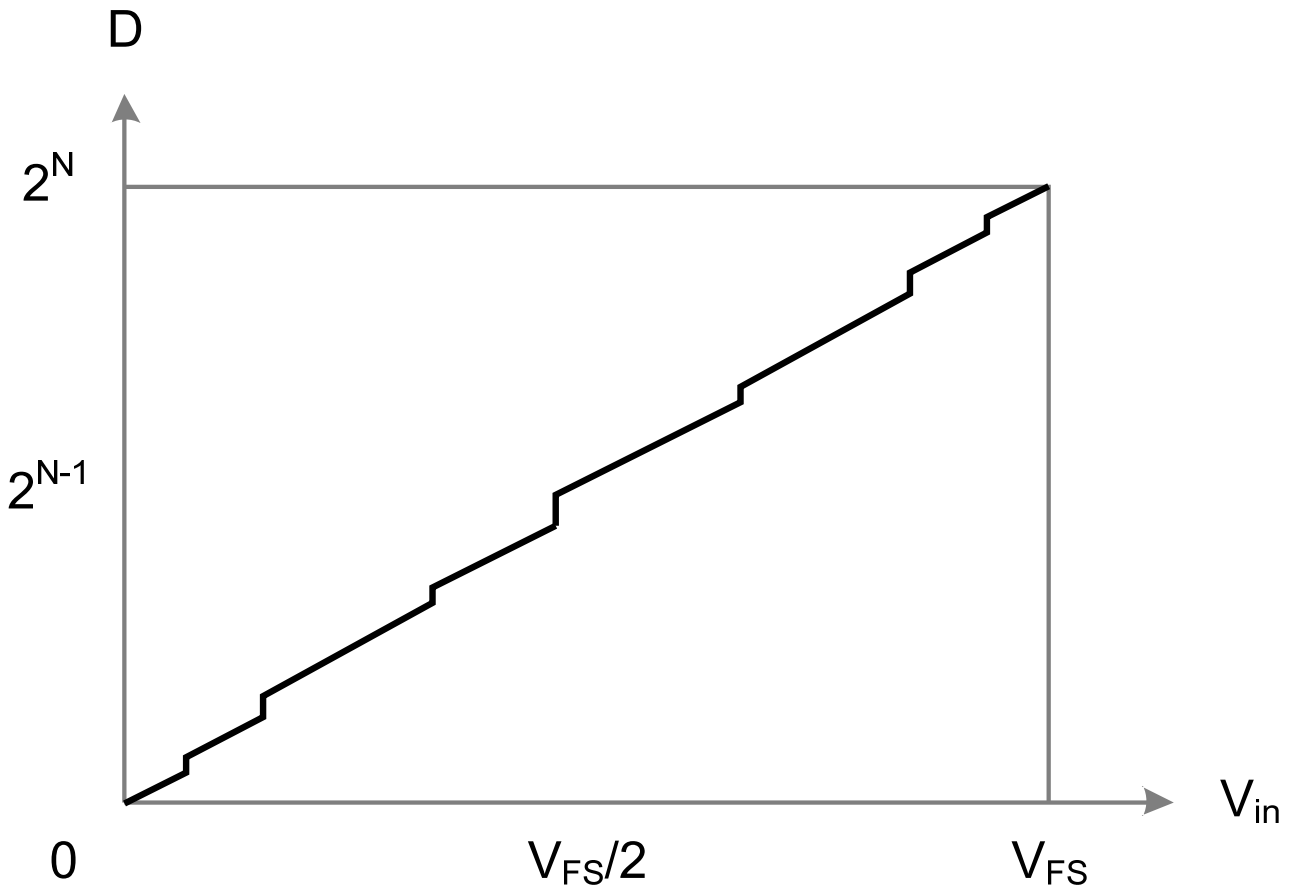
Raw
(14 bits)



Cal'd
(12 bits)

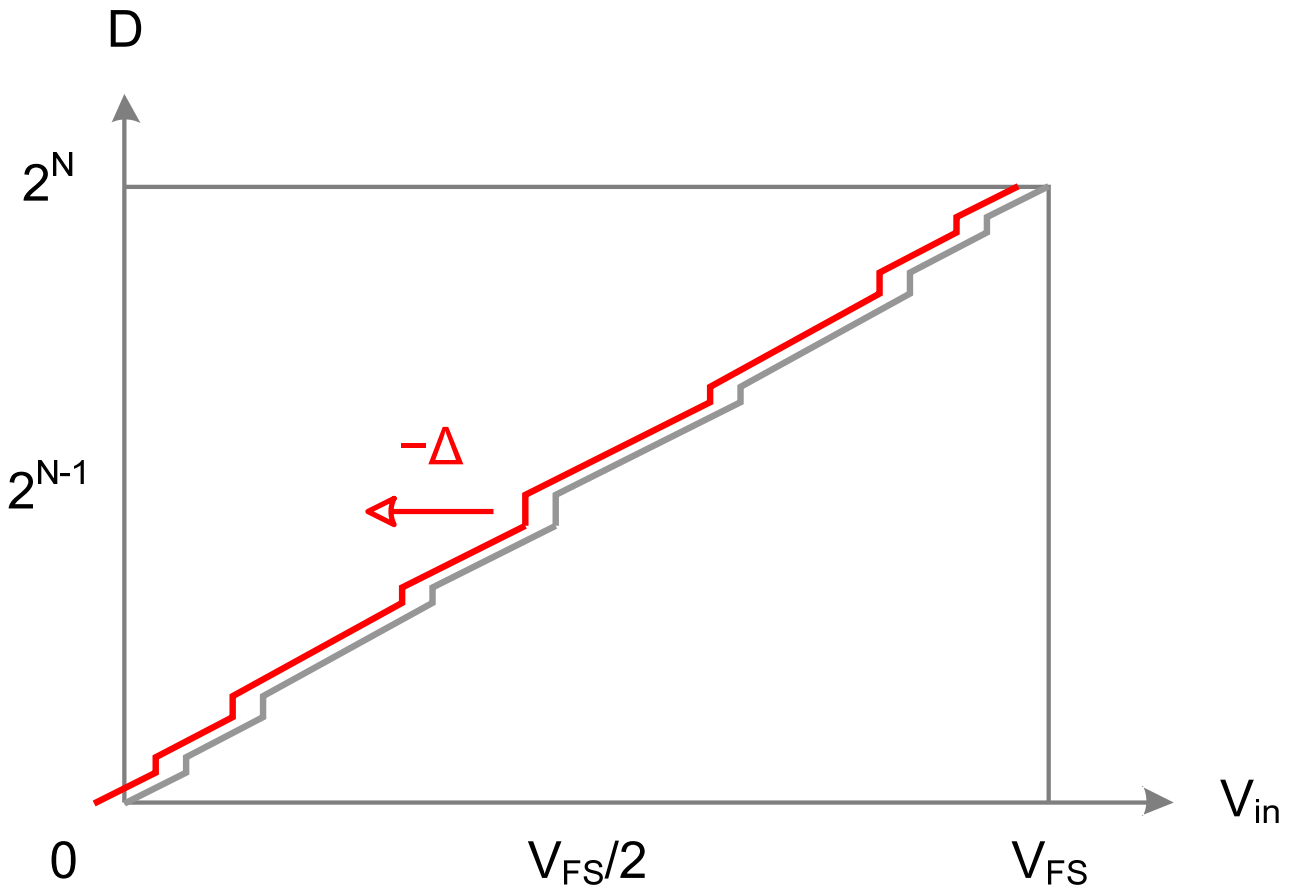


How to determine Bit Weights?



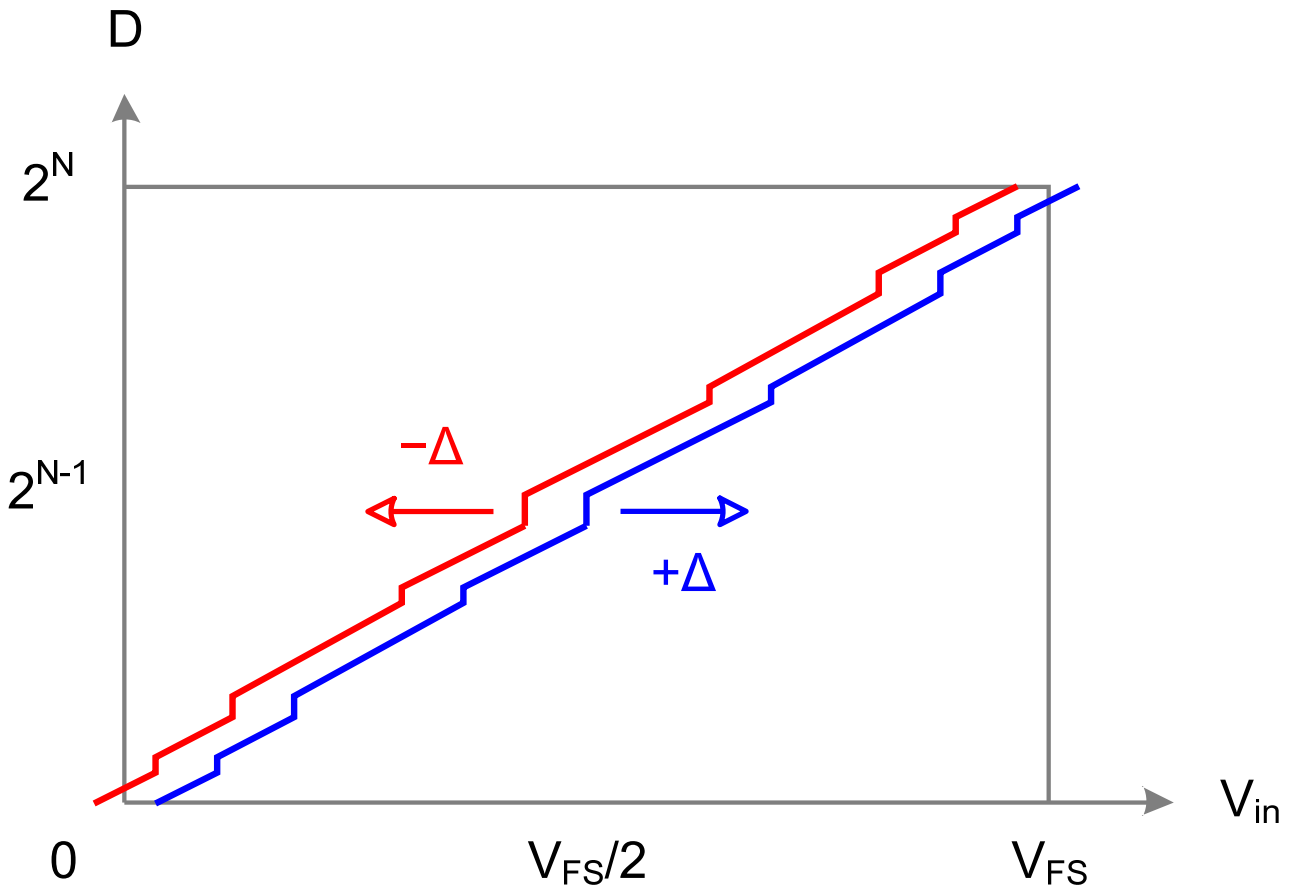
Is the transfer curve shift-invariant?

How to determine Bit Weights?



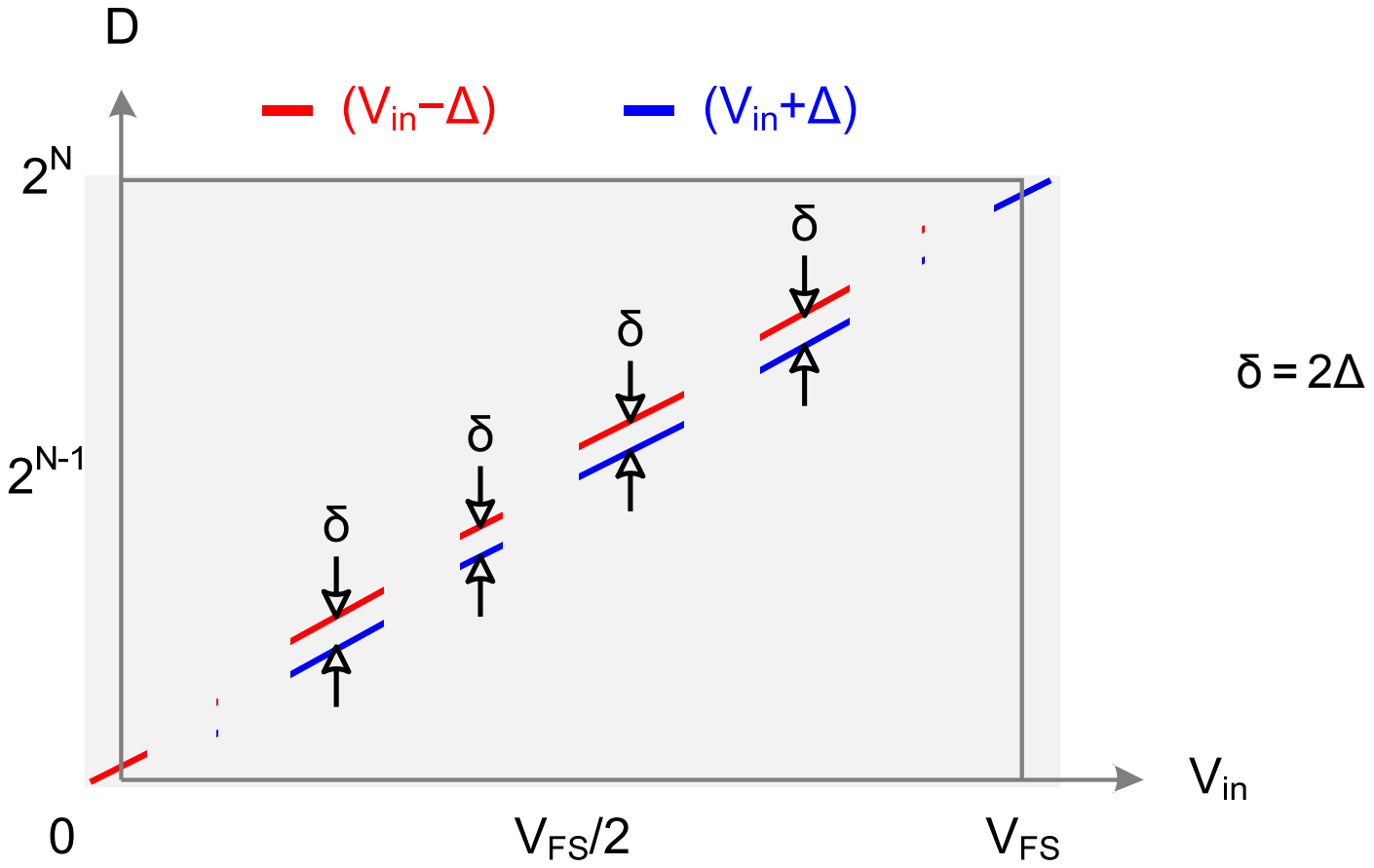
Is the transfer curve shift-invariant?

How to determine Bit Weights?



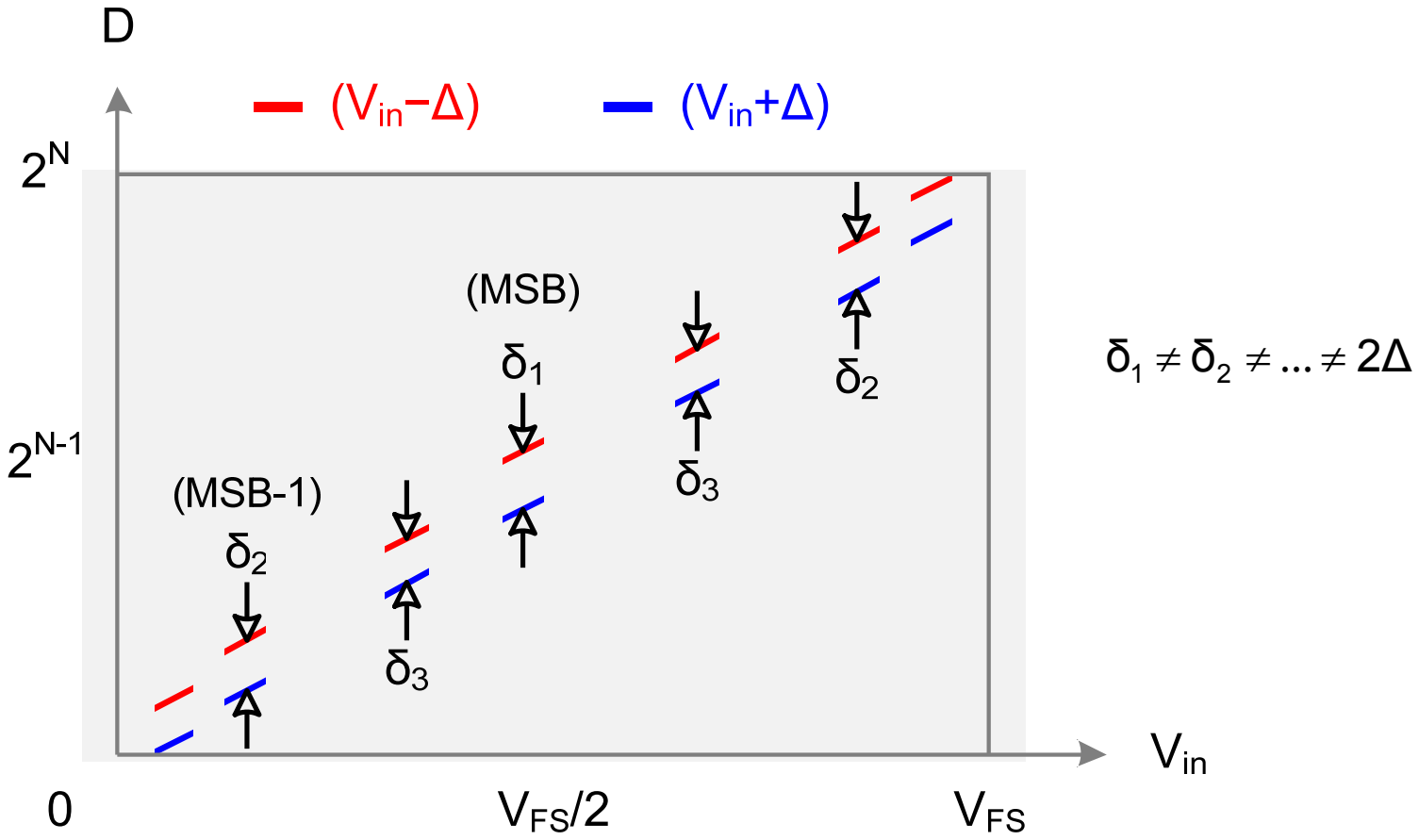
Is the transfer curve shift-invariant?

How to determine Bit Weights?



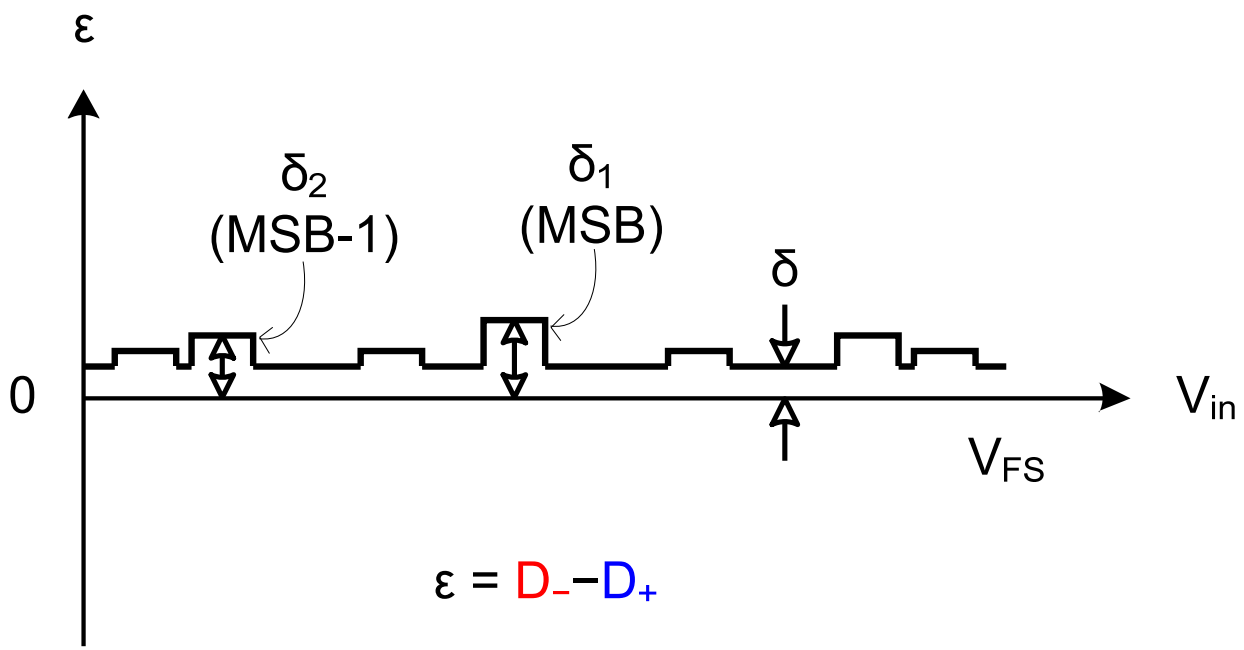
Transfer curve away from bit transitions is linear

How to determine Bit Weights?



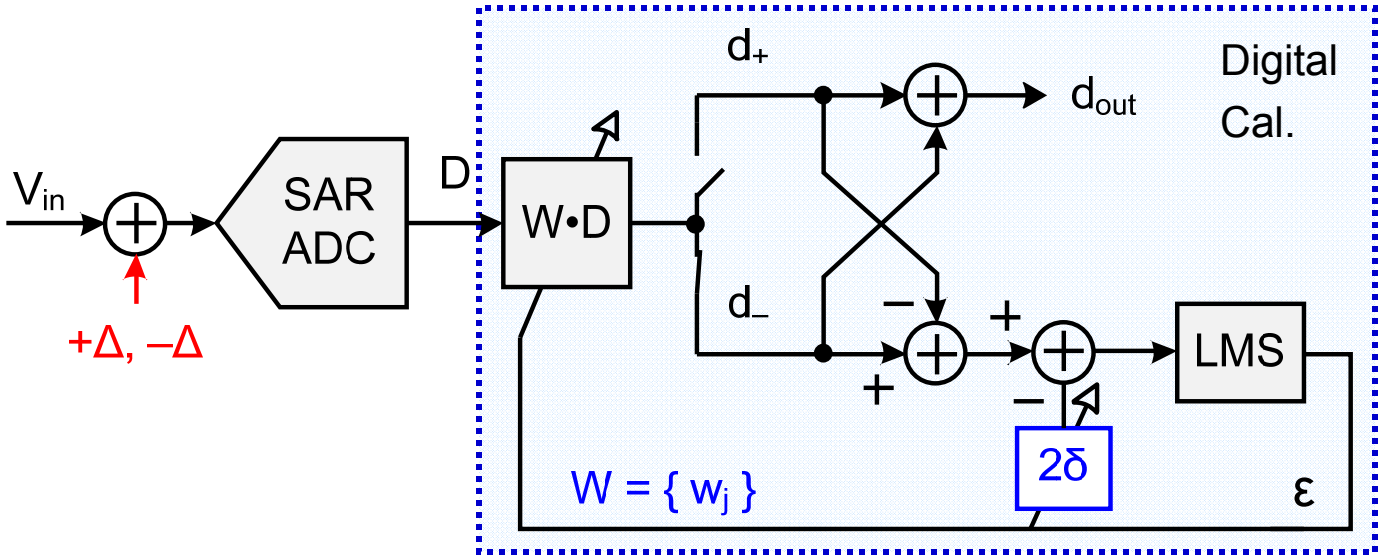
Transfer curve at bit transitions is nonlinear

How to determine Bit Weights?



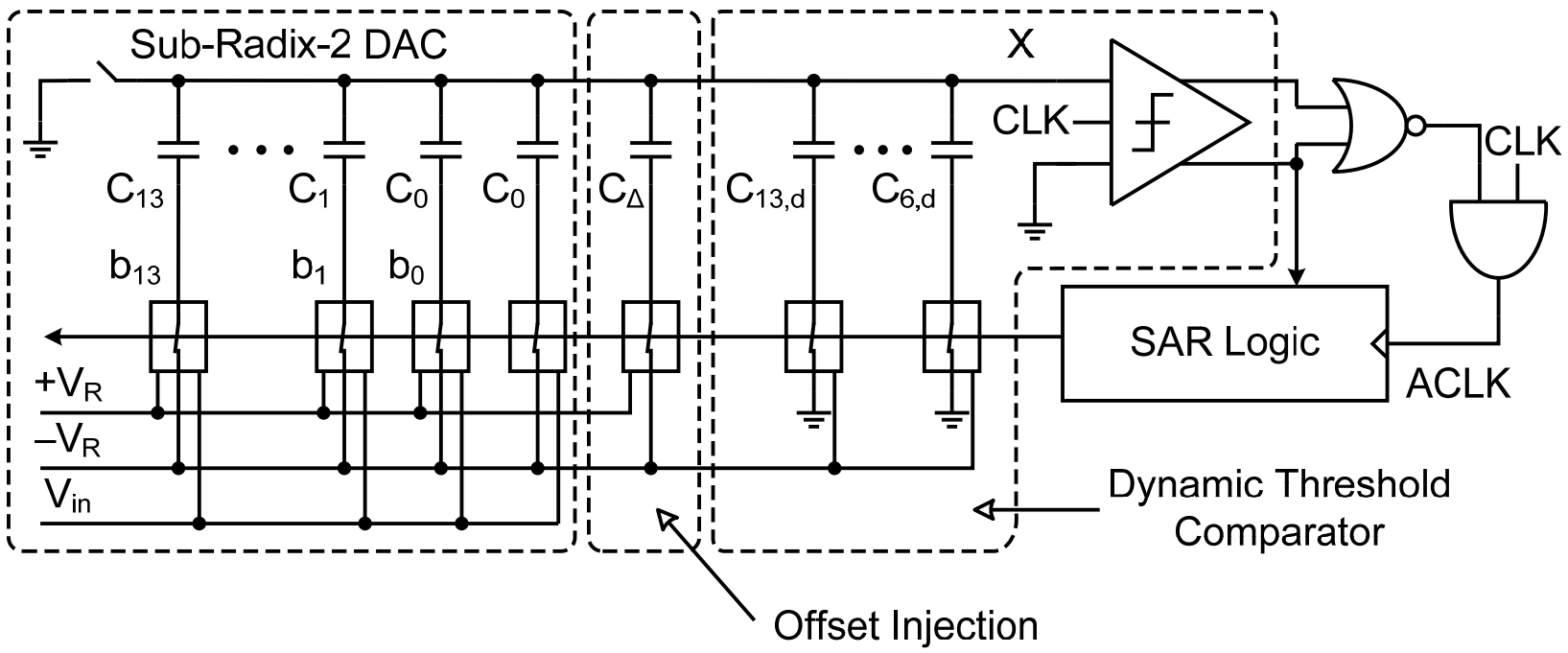
- Shift-invariant <u>ONLY</u> when the transfer curve is completely linear !!
- Non-constant difference b/t D_+ and D_- reveals bit weight information

Offset Double Conversion for SAR



- Offset double conversion (ODC) enables self-equalization
- ALL bit weights { w_j } are learned simultaneously !!

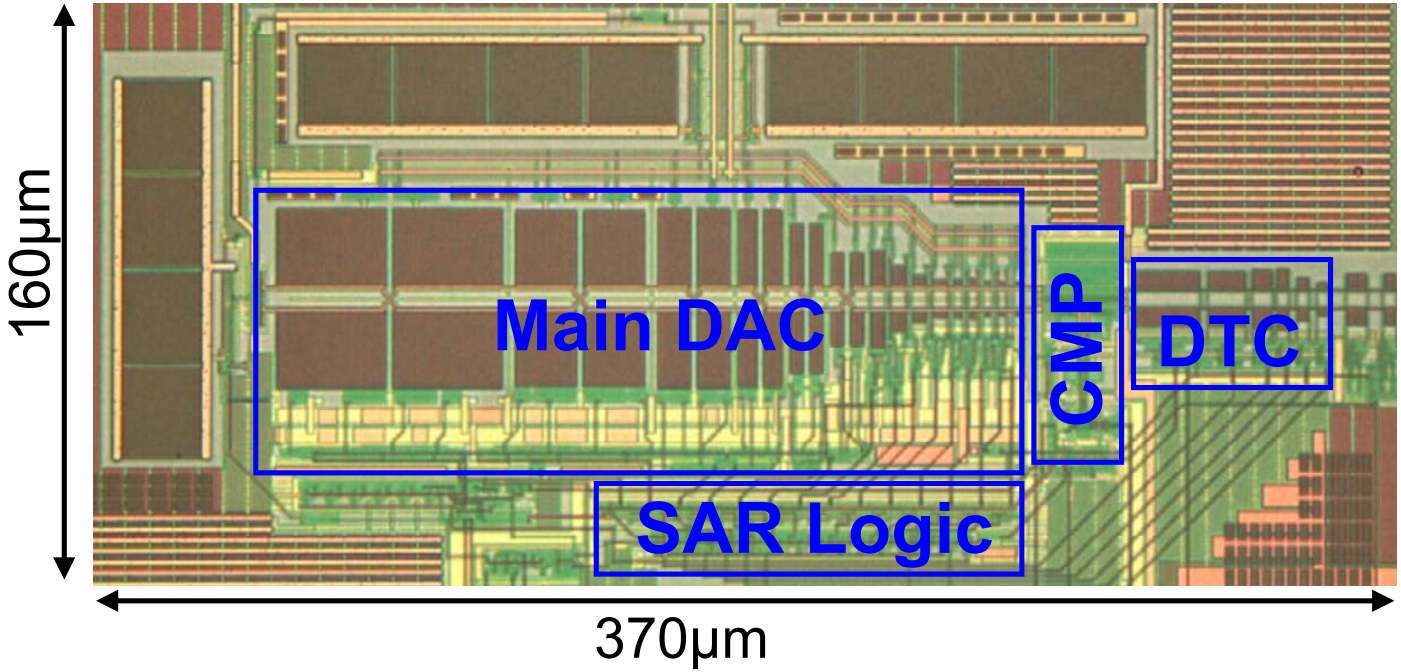
12b SAR ADC Prototype



Ref. [32]

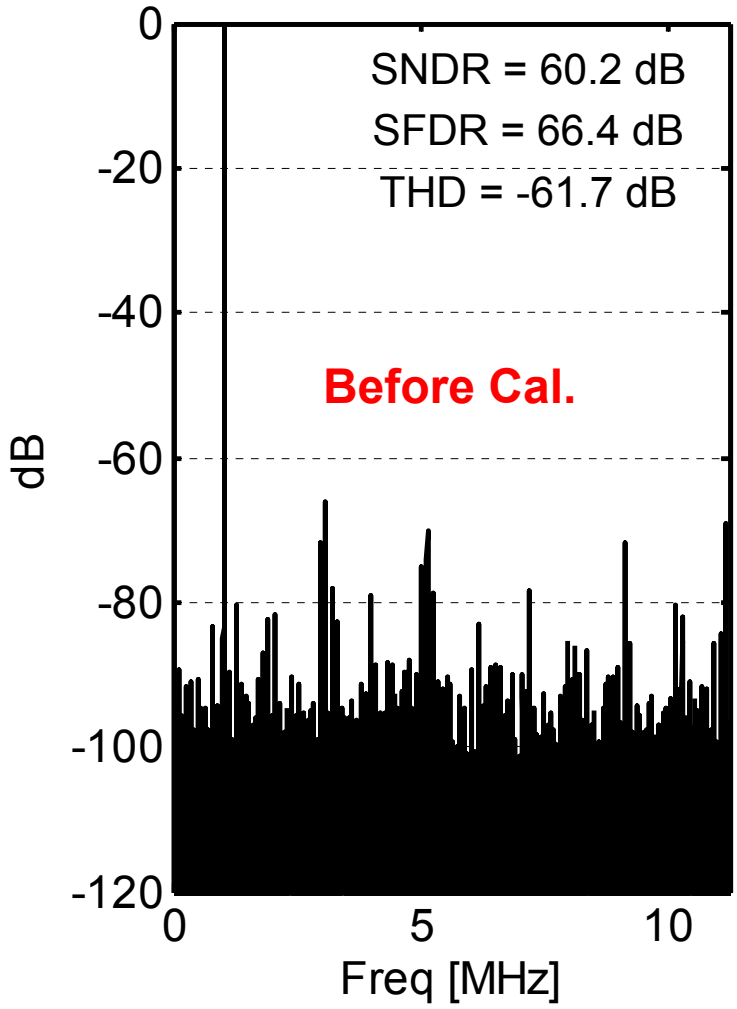
Simplicity, scalability, and efficiency

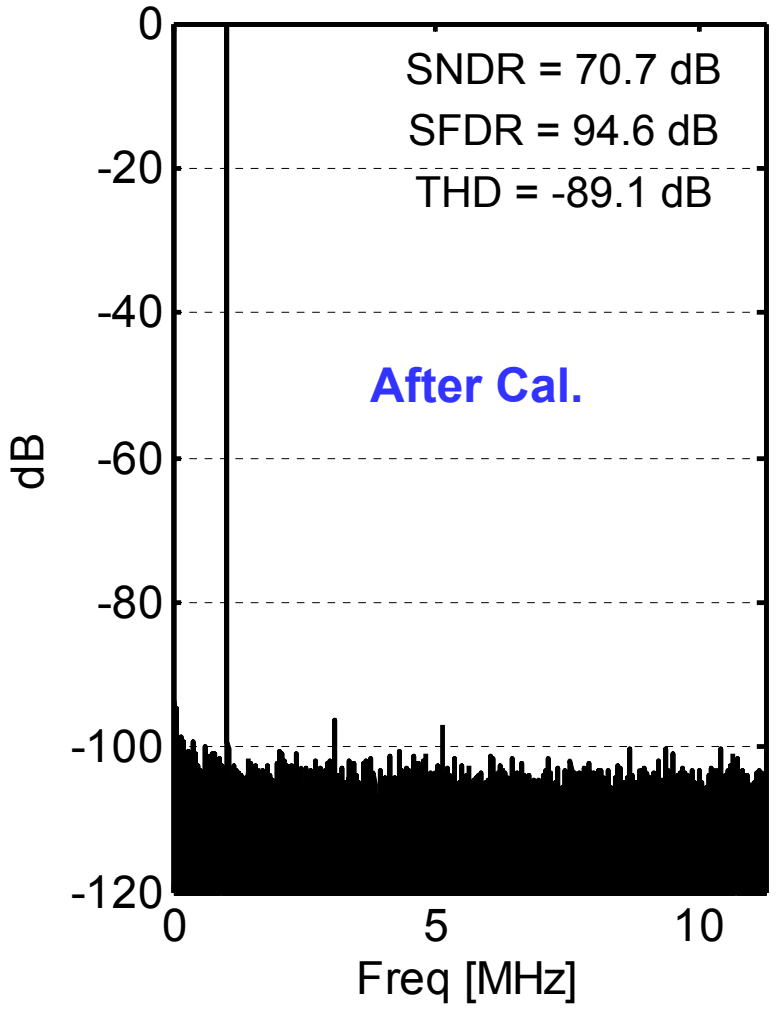
Die Photo (0.13µm CMOS, 0.06mm^2)



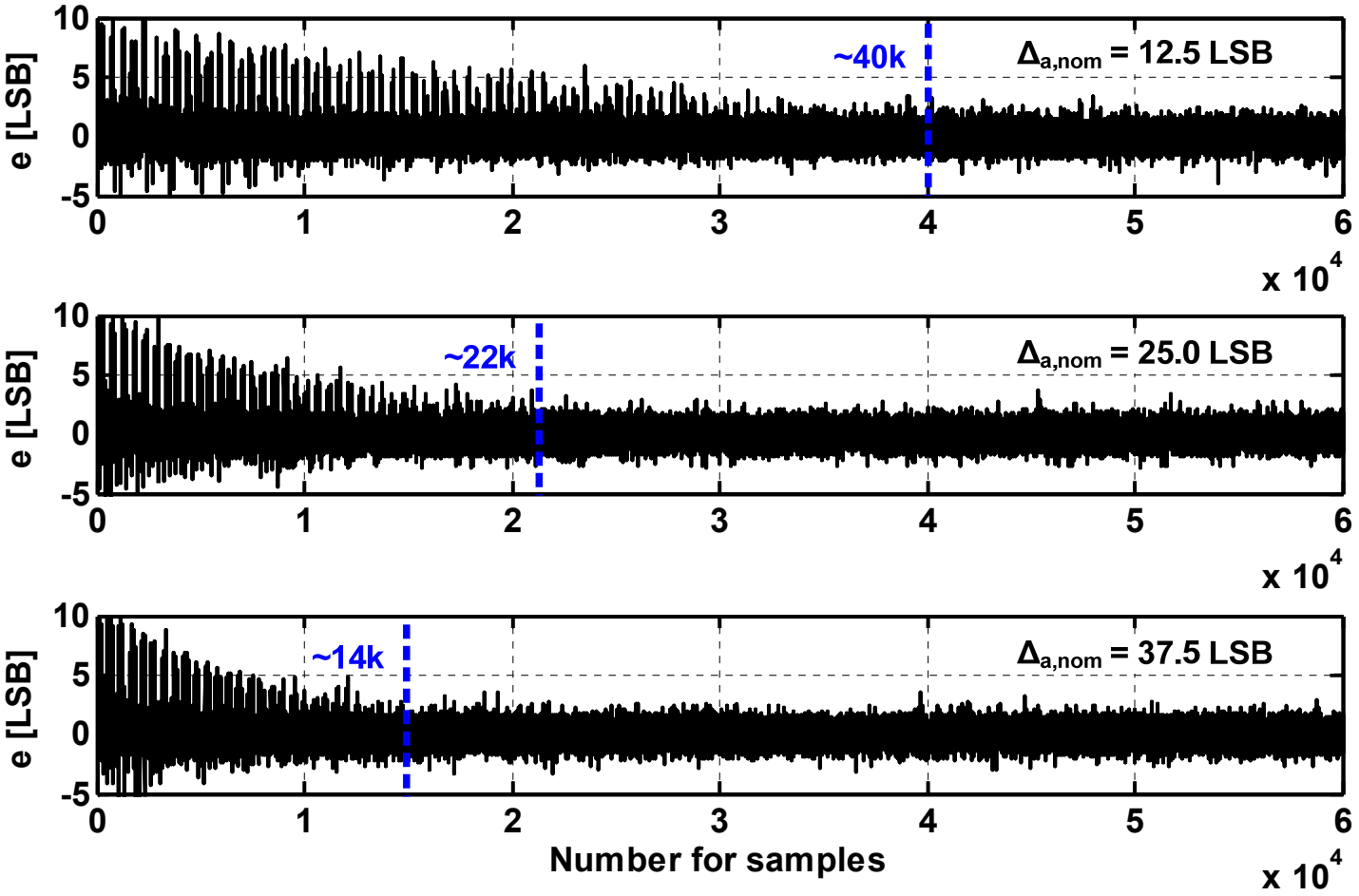
Ref. [32]

Measured Performance @ 12b, 22.5MS/s



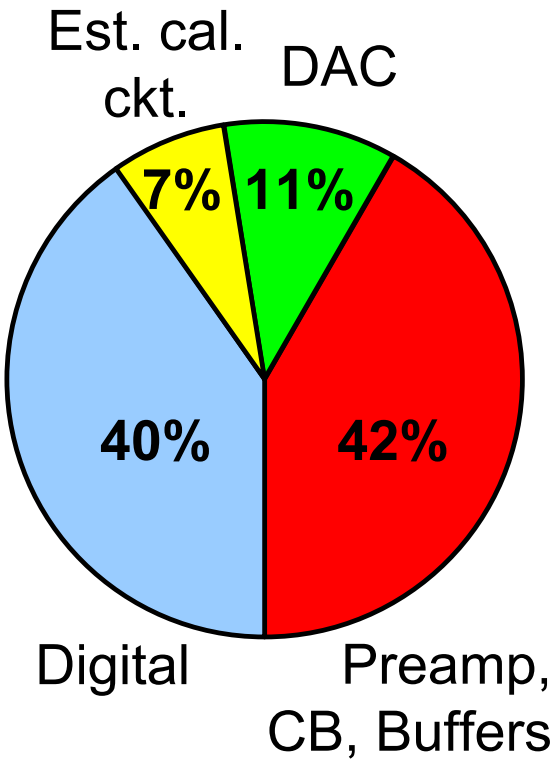


Convergence Speed



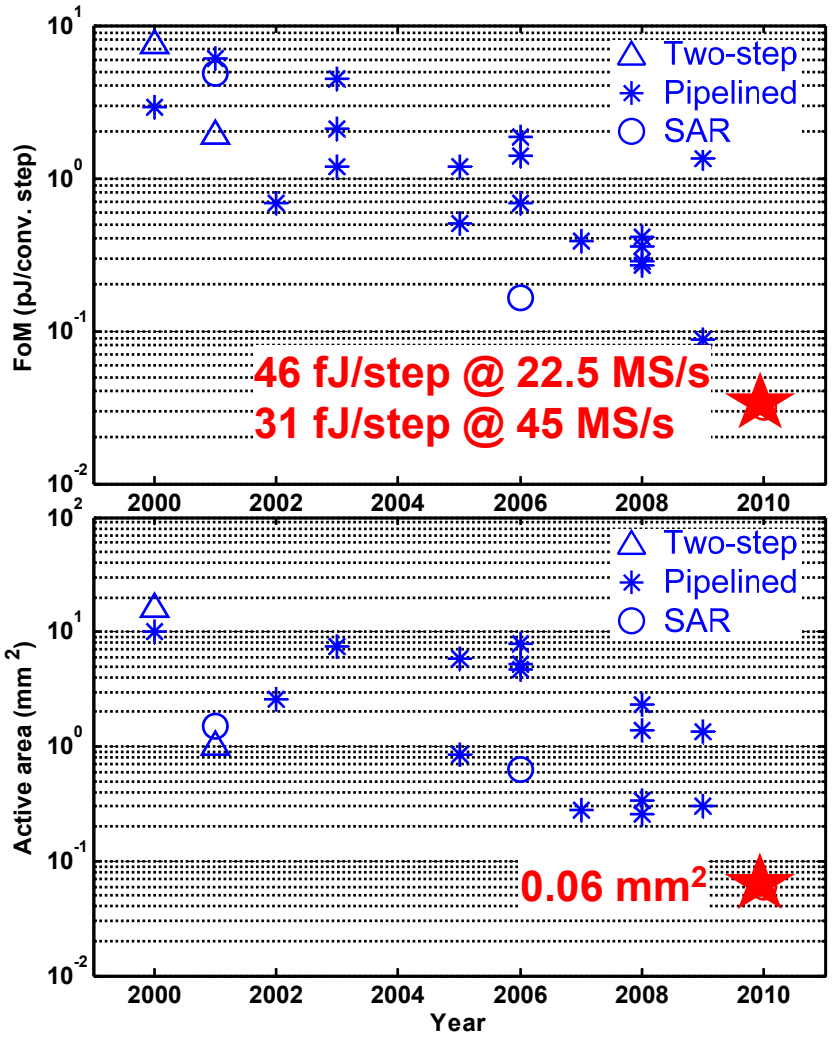
22000 samples @ 22.5 MS/s ≈ 1 ms

Comparison with 12b ADC's



Total Power: 3.0mW

@ time of publication



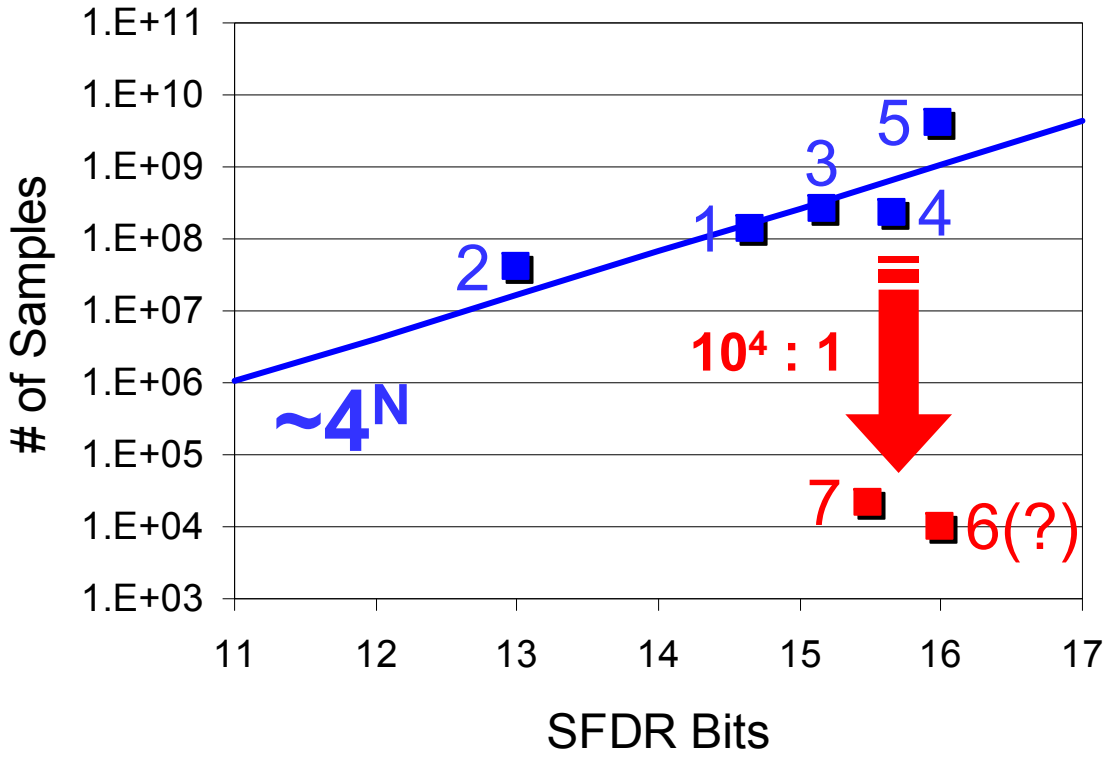
Convergence Time and Tracking Speed

Speed Concerns for Background Calibration

- Component aging, ambient variations, e.g., voltage, temperature, require different tracking speed for background calibration algorithms
- Amplifier nonlinearity is very sensitive to variations and signal statistics → needs special attention
- Reported speed performance varies. In general, equalization outperforms PRBS injection by large margin

Convergence Time

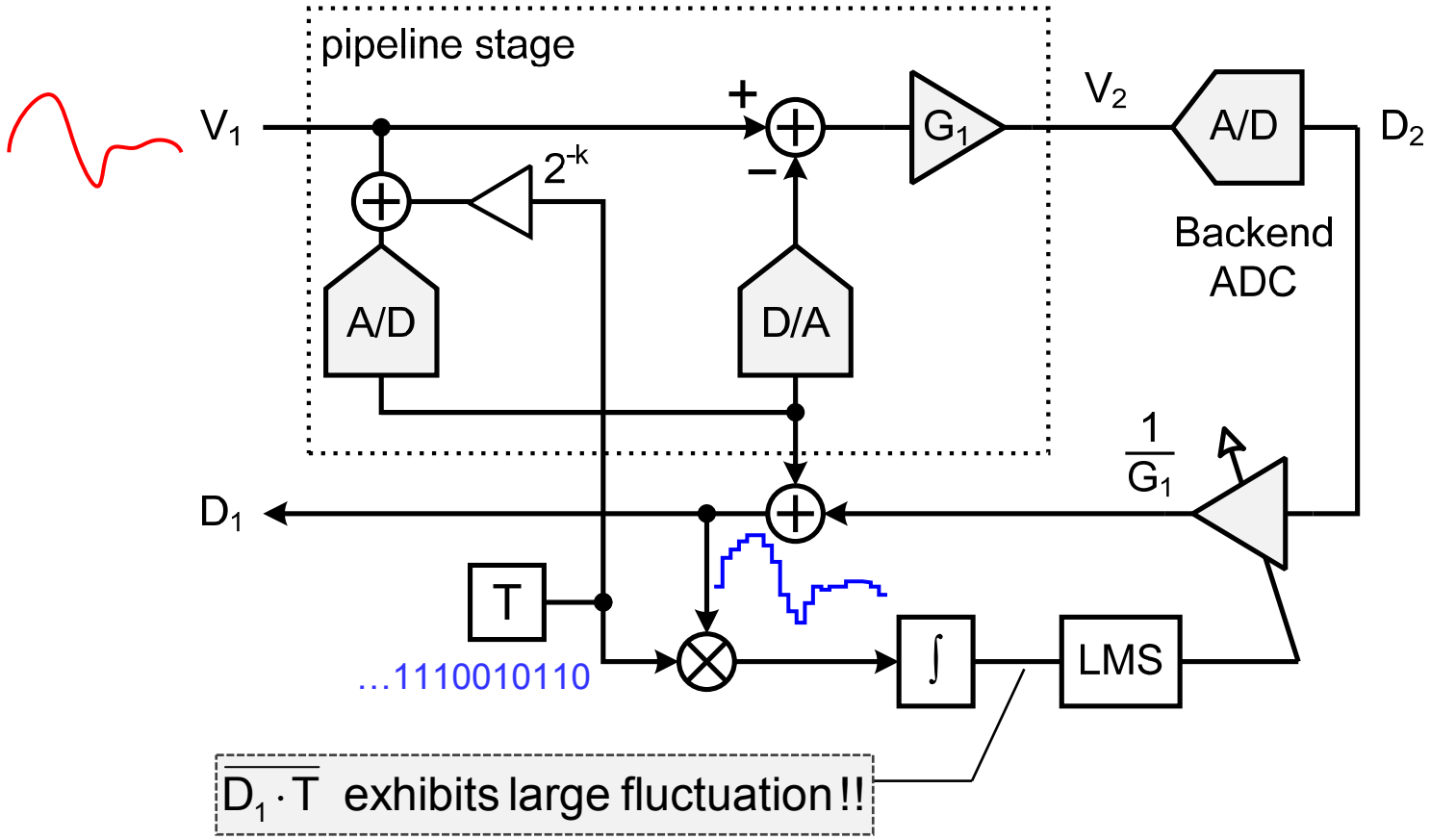
Convergence Time



#	Ref.	Sample	SFDR
1	[9]	134M	90dB
2	[18]	40M	80dB
3	[13]	268M	93dB
4	[12]	225M	96dB
5	[15]	400M	98dB
6	[30]	10k	?
7	[32]	22k	95dB

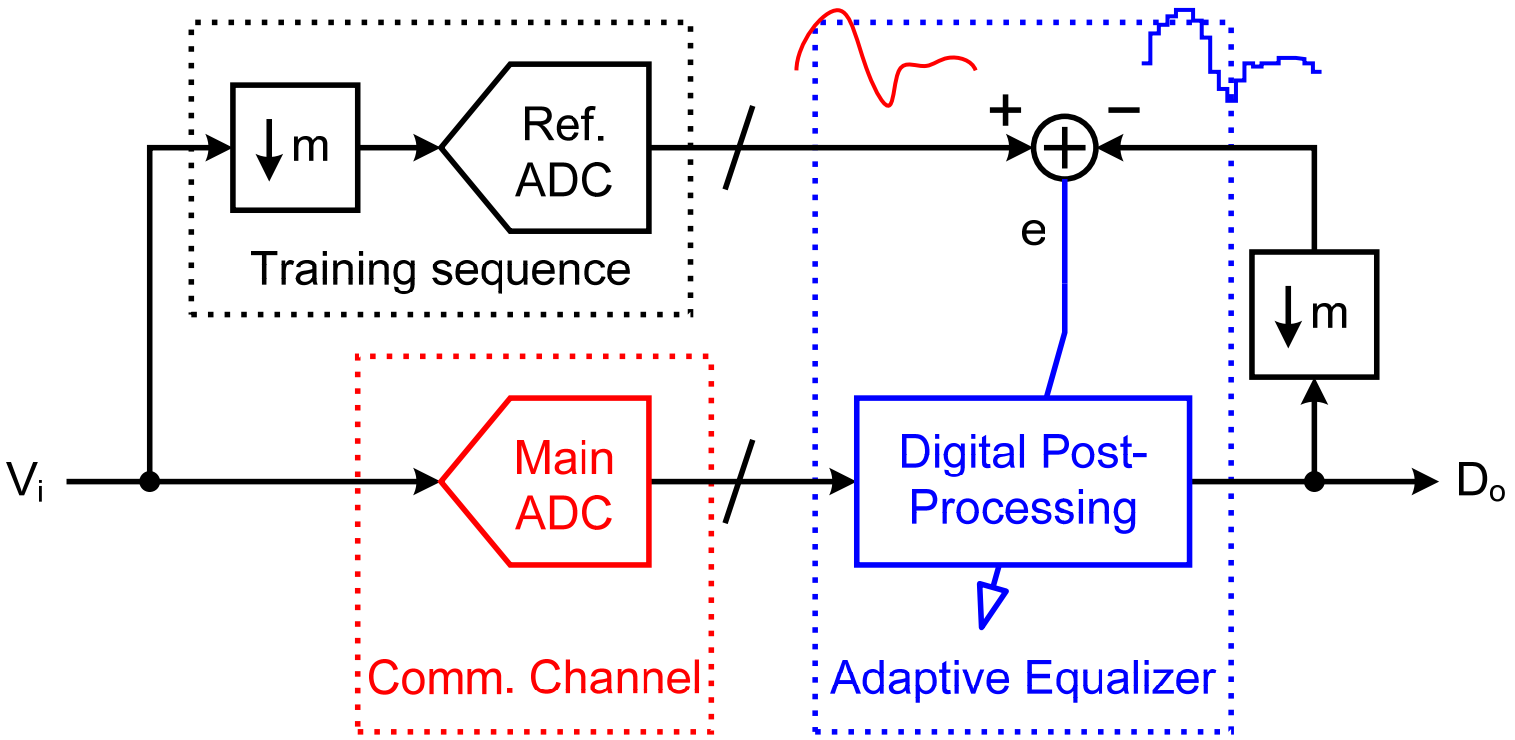
Convergence/tracking speed determines the sensitivity, testability, and ultimately practicality of a treatment…

Why Dithering is Slow?



Clean observation through correlation process requires ~2^{2N} samples

Why Equalization is Fast?



Zero-forcing → e drops to 0 with help of "training sequence"

Conclusion Remarks

Thank you for your attendance!

Bibliography

1. R. H. Walden, “Analog-to-digital converter survey and analysis,” IEEE JSAC, issue 4, 1999.
2. S.-U. Kwak, B.-S. Song, and K. Bacrania, “A 15-b 5-Msamples/s low spurious CMOS ADC,” JSSC, Dec. 1997.
3. E. Erdogan et al., “A 12-b digital-background-calibrated algorithmic ADC with -90-dB THD,” JSSC, Dec. 1999.
4. T. Sun, A. Wiesbauer, and G. C. Temes, “Adaptive compensation of analog circuit imperfections for cascaded delta-sigma ADCs,” in ISCAS 1998.
5. P. Kiss et al., “Adaptive digital correction of analog errors in MASH ADC’s—Part II: Correction using test-signal injection,” TCAS II, July 2000.
6. 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, Dec. 1998.
7. I. Galton, “Digital cancellation of D/A converter noise in pipelined A/D converters,” TCAS II, Mar. 2000.
8. P. C. Yu et al., “A 14b 40MS/s pipelined ADC with DFCA,” in ISSCC 2001.
9. E. J. Siragusa and I. Galton, “A digitally enhanced 1.8V 15b 40MS/s CMOS pipelined ADC,” in ISSCC 2004.
10. A. Panigada and I. Galton, “Digital background correction of harmonic distortion in pipelined ADCs,” TCAS I, Sept. 2006.
11. A. Panigada and I. Galton, “A 130mW 100MS/s pipelined ADC with 69dB SNDR enabled by digital harmonic distortion correction,” in ISSCC 2009.
12. K. Nair and R. Harjani, “A 96dB SFDR 50MS/s digitally enhanced CMOS pipeline A/D converter,” in ISSCC 2004.
13. H.-C. Liu, Z.-M. Lee, and J.-T. Wu, “A 15b 20MS/s CMOS pipelined ADC with digital background calibration,” in ISSCC 2004.

Bibliography

14. J.-L. Fan, C.-Y. Wang, and J.-T. Wu, “A robust and fast digital background calibration technique for pipelined ADCs,” TCAS I, June 2007.
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. J. Li and U.-K. Moon, “Background calibration techniques for multistage pipelined ADC’s with digital redundancy,” TCAS II, Sept. 2003.
17. J. Li et al., “0.9V 12mW 2MSPS algorithmic ADC with 81dB SFDR,” in VLSI 2004.
18. B. Murmann et al., “A 12b 75MS/s pipelined ADC using open- loop residue amplification,” in ISSCC 2003.
19. J. Keane et al., “Background interstage gain calibration technique for pipelined ADCs,” TCAS I, Jan. 2005.
20. R. Massolini, G. Cesura, and R. Castello, “A fully digital fast convergence algorithm for nonlinearity correction in multistage ADC,” TCAS II, May 2006.
21. Y. Chiu, “A framework of digital-domain background calibration of multi-step ADC using pseudorandom test signal injection,” in SampTA 2011.
22. 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.
23. 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.
24. X. Wang et al., “A 12-bit 20-MS/s pipelined ADC with nested digital background calibration,” in CICC 2003.
25. Y. Chiu et al., “Least mean square adaptive digital background calibration of pipelined analog-to-digital converters,” TCAS I, Jan. 2004.
26. C. Tsang et al., “Background ADC calibration in digital domain,” in CICC 2008.
27. Y. Chiu, “Equalization techniques for nonlinear analog circuits,” IEEE Comm. Mag., Apr. 2011.

Bibliography

28. B. Peng et al., “A virtual-ADC digital background calibration technique for multistage A/D conversion,” TCAS II, Nov. 2010.
29. B. Peng et al., “A 48-mW, 12-bit, 150-MS/s pipelined ADC with digital calibration in 65nm CMOS,” in CICC 2011.
30. J. McNeill et al., “Split ADC architecture for deterministic digital background calibration of a 16b 1MS/s ADC,” in ISSCC 2005.
31. J. McNeill et al., “Split-ADC digital background correction of open-loop residue amplifier nonlinearity errors in a 14b pipeline ADC,” in ISCAS 2007.
32. W. Liu, P. Huang, and Y. Chiu, “A 12bit 22.5/45MS/s 3.0mW 0.059mm2 CMOS SAR ADC achieving over 90dB SFDR,” in ISSCC 2010.
33. B. Peng et al., “An offset double conversion technique for digital calibration of pipelined ADCs,” TCAS II, Dec. 2010.
34. H. S. Fetterman et al., “CMOS pipelined ADC employing dither to improve linearity,” in CICC 1999.
35. W. Liu et al., “A 600MS/s 30mW 0.13µm CMOS ADC array achieving over 60dB SFDR with adaptive digital equalization,” in ISSCC 2009.