ECE1371 Advanced Analog Circuits
Lecture 12

MATCHING AND MISMATCH SHAPING

Richard Schreier
richard.schreier@analog.com

Trevor Caldwell
trevor.caldwell@utoronto.ca

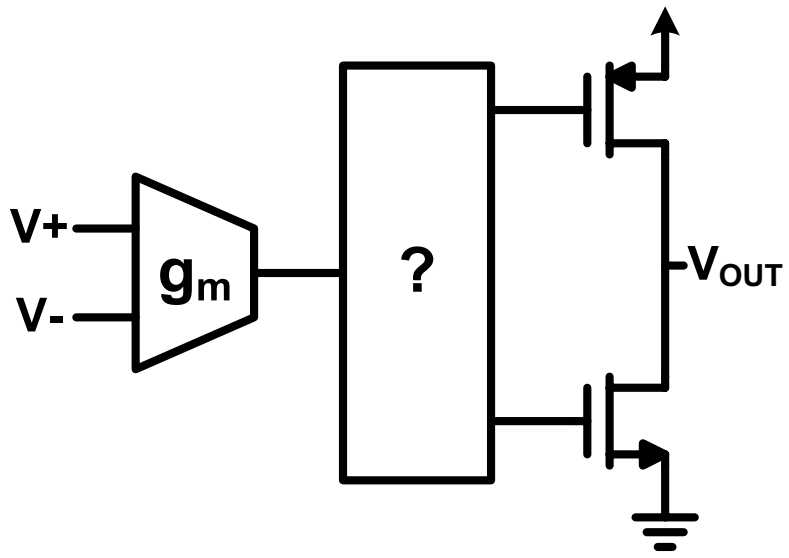
Course Goals

- **Deepen Understanding of CMOS analog circuit design through a top-down study of a modern analog system**
  - **The lectures will focus on Delta-Sigma ADCs, but you may do your project on another analog system.**
- **Develop circuit insight through brief peeks at some nifty little circuits**
  - **The circuit world is filled with many little gems that every competent designer ought to recognize.**

Date	Lecture			Ref	Homework
2008-01-07	RS	1	Introduction: MOD1 & MOD2	S&T 2-3, A	Matlab MOD2
2008-01-14	RS	2	Example Design: Part 1	S&T 9.1, J&M 10	Switch-level sim
2008-01-21	RS	3	Example Design: Part 2	J&M 14, S&T B	Q-level sim
2008-01-28	TC	4	Pipeline and SAR ADCs	J&M 11,13	Pipeline DNL
2008-02-04	ISSCC – No Lecture				
2008-02-11	RS	5	Advanced $\Delta\Sigma$	S&T 4, 6.6, 9.4, B	CTMOD2; Proj.
2008-02-18	Reading Week – No Lecture				
2008-02-25	RS	6	Comparator and Flash ADC	J&M 7	
2008-03-03	TC	7	SC Circuits	Raz 12, J&M 10	
2008-03-10	TC	8	Amplifier Design		
2008-03-17	TC	9	Amplifier Design		
2008-03-24	TC	10	Noise in SC Circuits	S&T C	
2008-03-31	RS	11	Switching Regulator		
2008-04-07	No Lecture				
2008-04-14	TC	12	Matching & MM-Shaping	S&T 6.3-6.5,+	
2008-04-21	Project Presentations and Report				

NLCOTD: Class-AB Output Bias

- **How do we bias this class-AB output stage?**
  - **Want to keep both PMOS and NMOS transistors on**



Highlights
(i.e. What you will learn today)

1. **Sources of Mismatch**

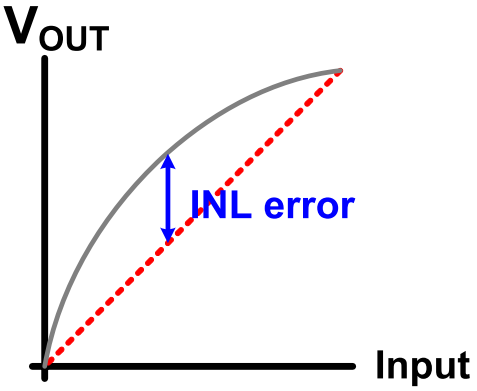
2. **Some matching techniques**
   - **Common-centroid**
   - **Interdigitation**

3. **Mismatch Shaping**
   - **Randomization, 1st and 2nd order schemes**

Need for Matching

- **Poorly matched devices (transistors, capacitors, resistors) can lead to non-idealities**
  - **Amplifier Offset**
  - **Converter Non-linearity**
  - **Gain Error**

- **DAC mismatch in $\Delta\Sigma$**
  - **Mismatch in DAC current sources or capacitors causes INL error in output**



Sources of Matching Error

- **Systematic Mismatch**
  - **Introduced by circuit/layout designer**
  - **Can usually be avoided**
- **Random Mismatch**
  - **Variation in process parameters and lithography**
  - **Beyond the designers control – must take these into account during the design process**
- **Gradient Mismatch**
  - **First- or second-order fluctuations over longer lengths across the chip**

Systematic Mismatch

- **Some good design techniques exist to help minimize these matching errors**
  - **Use multiples of small, unit sized devices (transistors stripes, resistor and capacitor arrays)**
  - **Use cascodes – increased output impedence (smaller current variations with changes in V_{DS})**
  - **Avoid asymmetric loading – especially for dynamic signals (match wire lengths, capacitances)**
  - **Don't mix different types of devices if they are supposed to match (e.g., poly resistors and n+ resistors)**

Random Mismatch

- **Due to random variations in…**
  - **Device length**
  - **Channel doping**
  - **Oxide thickness**
  - **Sheet resistance**
  - **Capacitance**

- **How are these errors reduced?**
  - **Increased device area**
  - **Increased Area/Perimeter ratio (square is best)**
  - **(more on this later…)**

Gradient Mismatch

- **To avoid these errors, devices should have similar environment**
  - **Same size, orientation, location, supplies, temperature**

- **Minimize these errors with some layout techniques**
  - **Common-centroid – when devices are supposed to be matched, balance them so that their centroids are the same (eliminates 1st-order gradient errors)**
  - **Interdigitation – not strictly common-centroid, but reduces impact of gradient errors**

Capacitor Matching

- **Example: Matching two capacitors C_1 and C_2**
  - **C_1 is 3pF**
  - **C_2 is 4pF**
  - **Want to maintain the 3:4 relative size with minimal errors**

How do we layout these capacitors?

Capacitor Matching Example

- **Consider overetching errors**
  - **Capacitance**

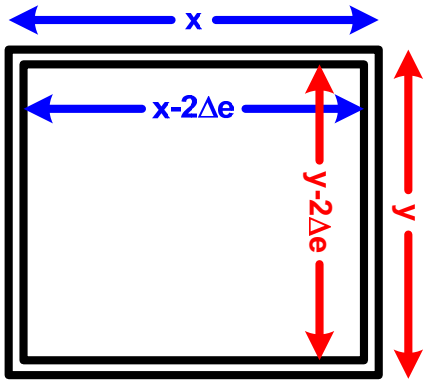
$$C = xyC_{ox}$$

  - **Capacitance error (for small Δe)**

$$\Delta C \approx -2\Delta e(x+y)C_{ox}$$

  - **Relative error**

$$\varepsilon_r = \frac{\Delta C}{C} \approx -2\Delta e\frac{x+y}{xy}$$



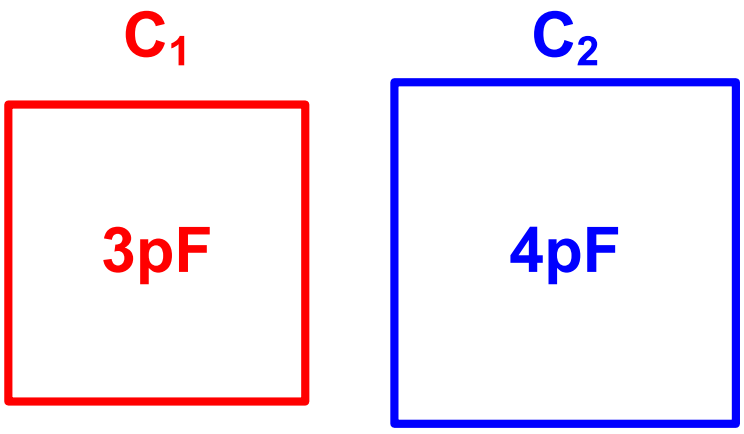
For a given area, relative capacitance error is minimized for x=y (square)

Capacitor Matching Example

- **Option 1**

  Make C_1 and C_2 both square capacitors with capacitor C_2 33% times bigger than C_1

  => minimizes relative capacitor error



Capacitor Matching Example

- **How do we preserve the 3:4 ratio with a given relative error for each?**

$$\frac{C_1}{C_2} \approx \frac{3(1+\varepsilon_{r,3pF})}{4(1+\varepsilon_{r,4pF})}$$

$$\varepsilon_{r,3pF} = \frac{\Delta C}{C} \approx -2\Delta e \frac{x_{3pF} + y_{3pF}}{x_{3pF} y_{3pF}}$$

$$\varepsilon_{r,4pF} = \frac{\Delta C}{C} \approx -2\Delta e \frac{x_{4pF} + y_{4pF}}{x_{4pF} y_{4pF}}$$

Ratio will be 3:4 as long as

$$\frac{x_{3pF} + y_{3pF}}{x_{3pF} y_{3pF}} = \frac{x_{4pF} + y_{4pF}}{x_{4pF} y_{4pF}}$$

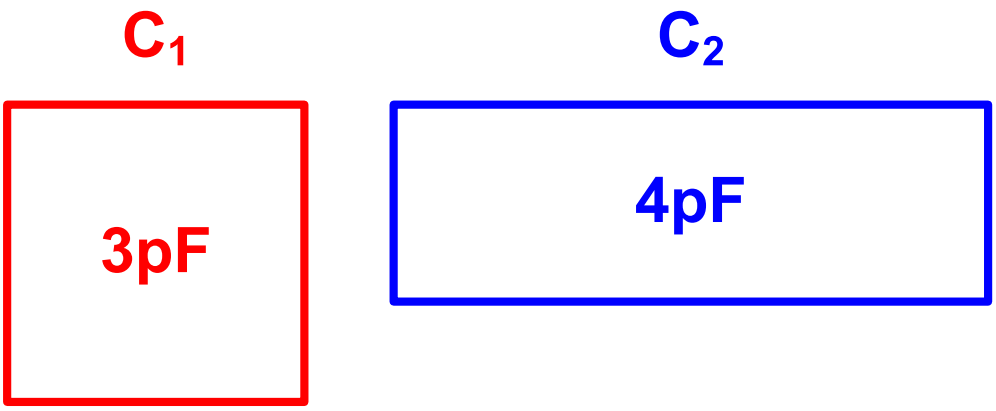
Keep the area to perimeter ratio the same for both capacitors

Capacitor Matching Example

- **Option 2**

  Make C_2 33% larger than C_1 but with the same area to perimeter ratio

  => matches relative capacitor error



Capacitor Matching Example

- **How do we match the boundary of each capacitor?**

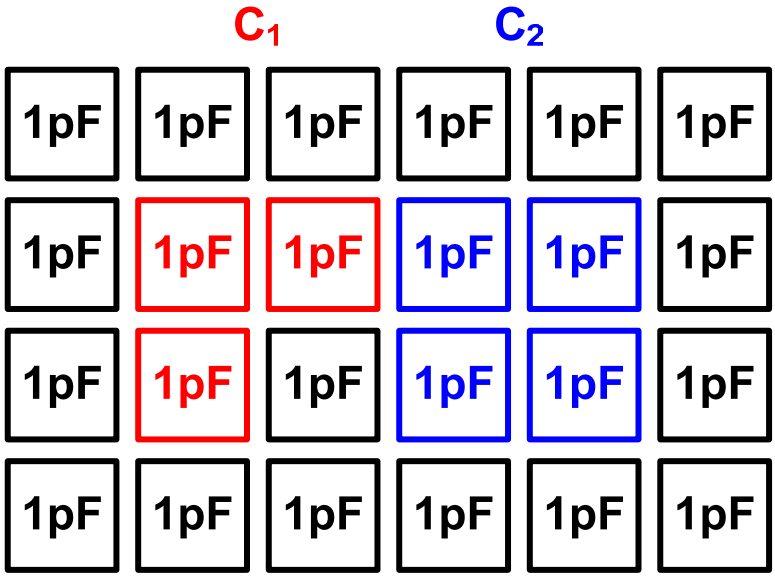
  With irregularly shaped capacitors it is difficult to ensure that every capacitor 'sees' the same edges/materials

- **Unit-sized capacitors with surrounding dummy capacitors**

  Smaller unit-sized capacitors can be realized to ensure that every capacitor 'sees' the same surrounding area

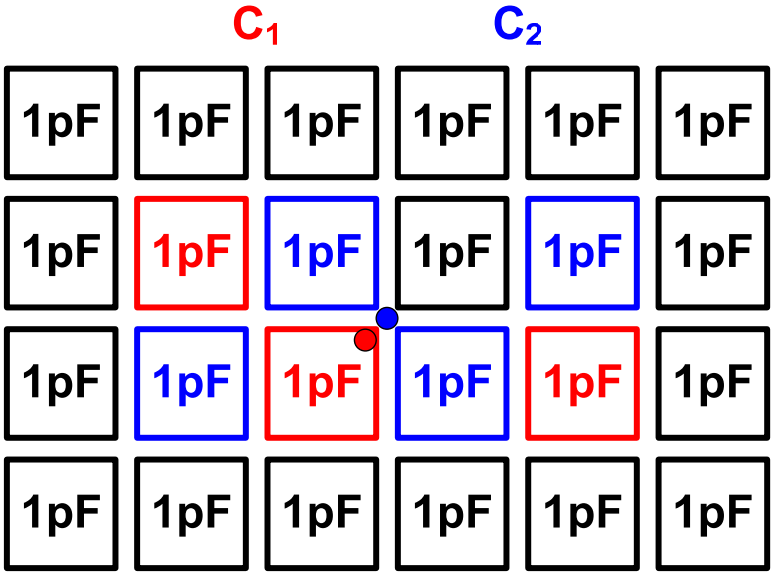
Capacitor Matching Example

- **Option 3**
  - **Divide into unit-sized 1pF capacitors**
  - **Use dummy capacitors around main C_1 and C_2**



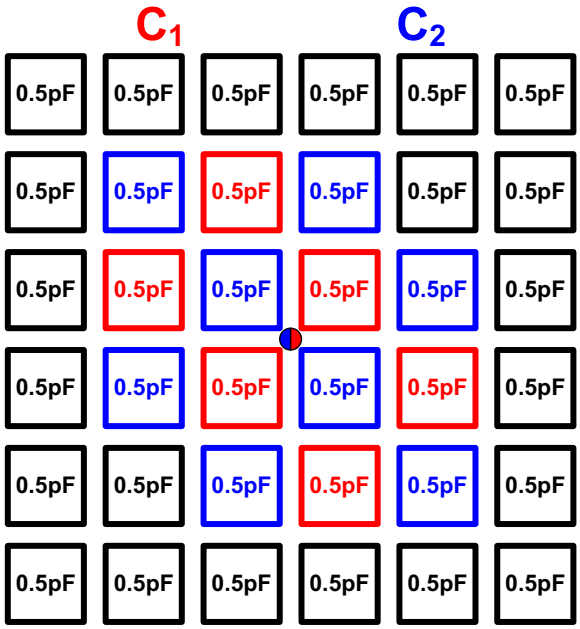
Capacitor Matching Example

- **Option 4**
  - **Common-centroid layout (with dummy caps)**
  - **Minimizes effects of 1st-order gradients**



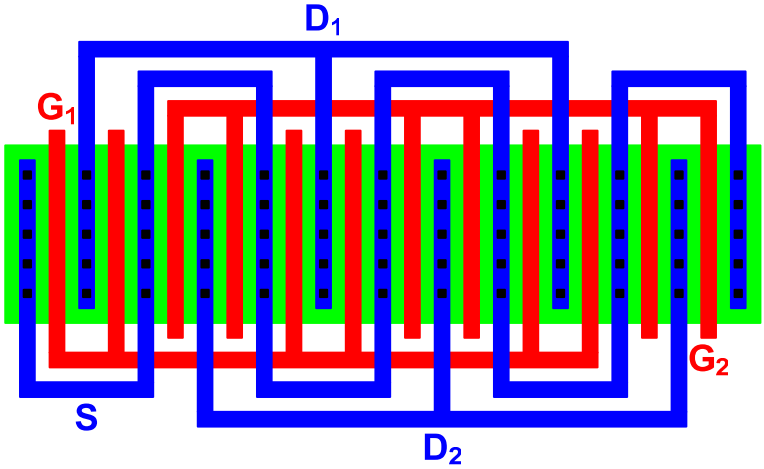
Capacitor Matching Example

- **Option 5**
  - **Smaller unit-sized capacitors (with dummy caps)**
  - **Centroids can be closer together or identical**



Interdigitation

- **Simple way to reduce 1st-order gradient effects**
  - **Easiest when MOS devices have same source node**
  - **Useful for current mirrors and differential pairs**
  - **As the number of fingers increases, this approaches a common-centroid layout**

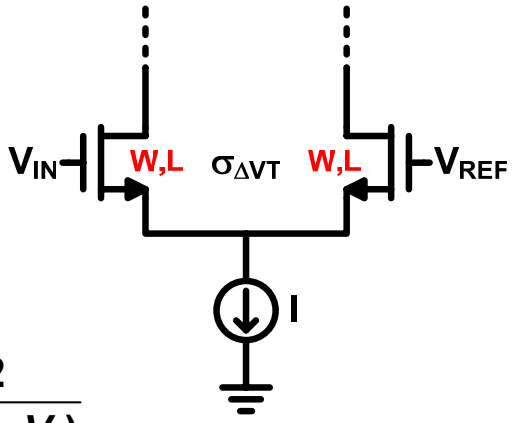


Reducing Random Mismatch

- **Even with interdigitation or common-centroid, random mismatch will exist in a diff. pair**
  - **Mismatch is proportional to area of transistor**
  - **Standard deviation of error is**

$$\sigma_{\Delta VT} = \frac{A_{VT}}{\sqrt{WL}}$$

  - **A_{VT} decreases almost linearly with each process generation**



- **Drain current variation**

$$\left.\frac{\sigma_I}{I}\right|_{VT} = \sigma_{VT}\frac{2}{V_{GS}-V_T} = \frac{A_{VT}}{\sqrt{WL}}\frac{2}{(V_{GS}-V_T)}$$

What happens when…?

…the device current is decreased by 4x?

- **Error current $g_m\Delta V_T$ reduces by 2 while current reduces by 4**
- **Alternatively, V_{EFF} reduces by 2**
  - **=> random mismatch error increases by 2**

…more unit devices are used, but overall area is maintained?

- **Smaller unit sizes allow use of common-centroid array structures**
- **Area is the same**
  - **=> random mismatch error is the same**

What happens when…?

…W is increased by 4 (I_D is the same)?

- More area is used and V_{EFF} is reduced
- V_{EFF} decreases by 2, but σ_{VT} decreases by 2
  - => random mismatch error is the same

…W/L is increased by 2, while I_D and area are kept constant?

- V_{EFF} decreases by $\sqrt{2}$
  - => random mismatch error increases by $\sqrt{2}$

Multi-bit Quantization

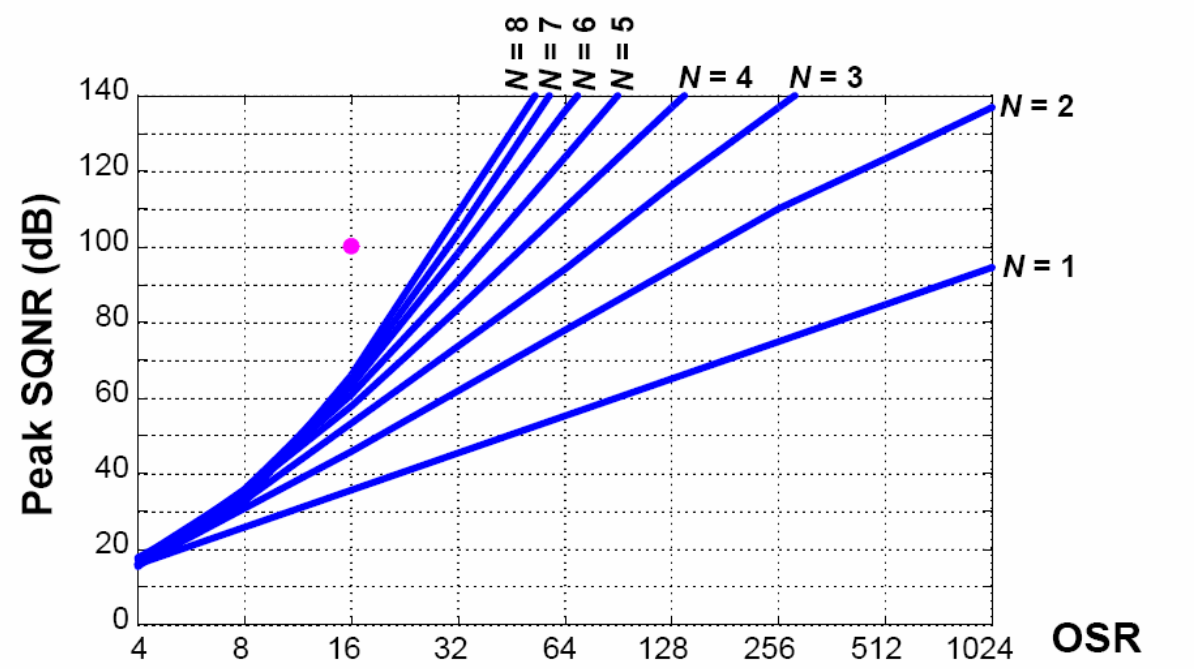
- ✓ **Overcome stability-induced restrictions on NTF**
  - Larger no-overload range
  - Dramatic improvements in SQNR
- ✓ **Smaller step-size**
  - Less slewing, CT less sensitive to jitter
- ✓ **Noise is 'whiter'**
  - Spurious tones can be avoided, dithering not required, design theory is much easier
- ✘ **Increased complexity of flash ADC and DAC**
  - More comparators, more DAC switches, larger layout
- ✘ **Loses inherent linearity property of binary DACs**
  - DAC levels are not evenly spaced and are non-linear
  - DAC errors are not noise shaped like ADC errors

Multi-bit $\Delta\Sigma$

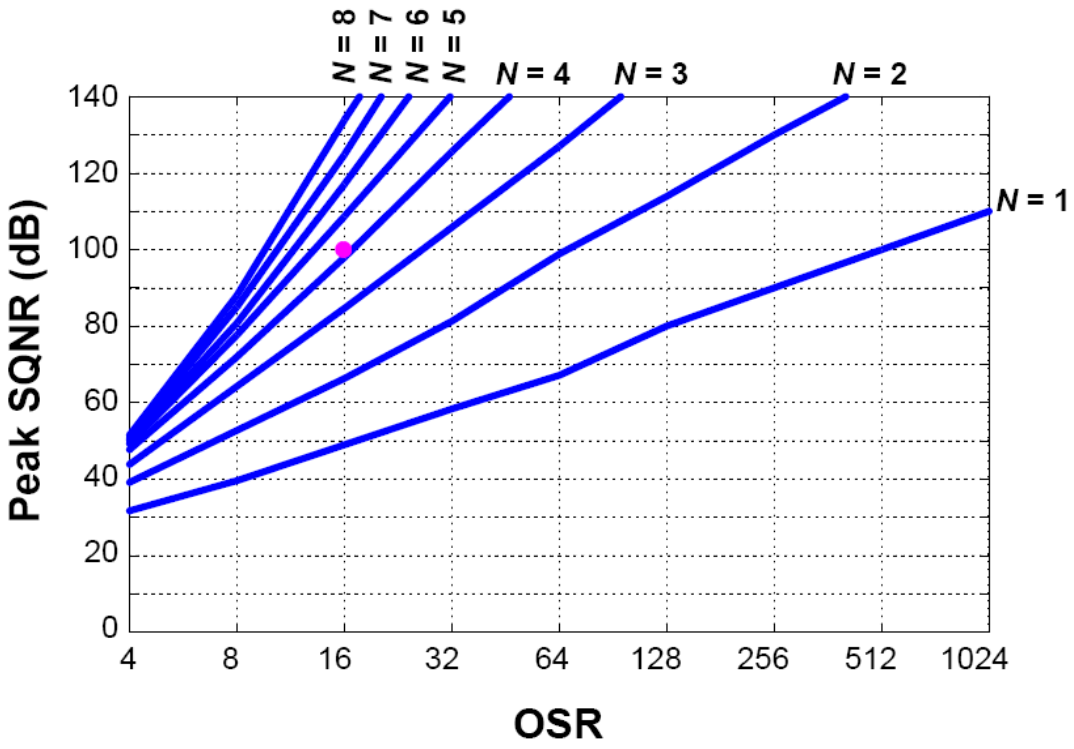
- **Binary quantization imposes severe constraints on the NTF**
  - **Example: OSR = 16, a 5th-order binary modulator**
  - **Binary quantizer only achieves SNR = 60dB**
  - **With a 3-bit quantizer, SNR = 108dB is possible**
  - **With a 4-bit quantizer, SNR = 120dB is possible**

Compare SQNR for 1-bit and 3-bit $\Delta\Sigma$ modulators

SQNR Limits for 1-bit Modulators

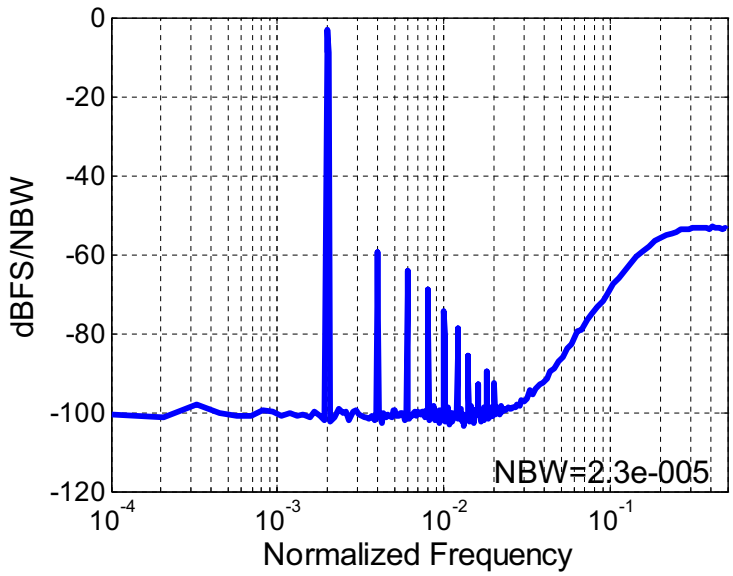


SQNR Limits for 3-bit Modulators



DAC Mismatch

- **3rd-order, 3-bit quantizer, OSR=50**
  - **DAC cell mismatch σ = 1%**
  - **SNDR = 50dB (ideally 107dB)**

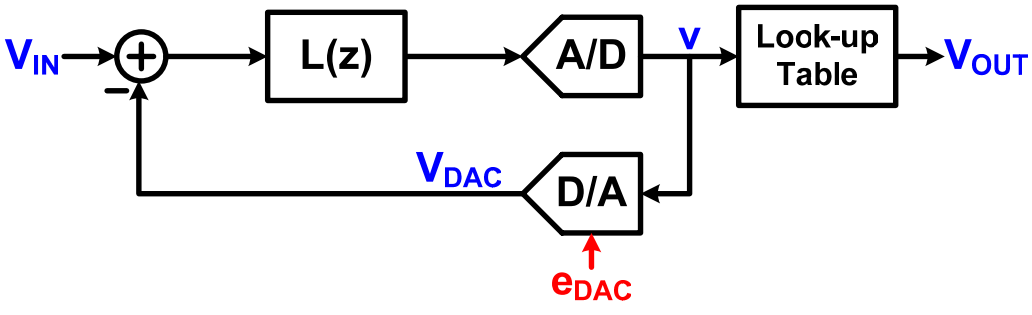


DAC Mismatch

- **Random DAC mismatches in multi-bit ΔΣ modulators are inevitable**
  - **DAC non-linearity causes harmonics that can limit the linearity of the whole modulator since they are introduced at the input**
- **These errors can be overcome with digital techniques**
  - **Digital correction and calibration**
  - **Mismatch shaping**

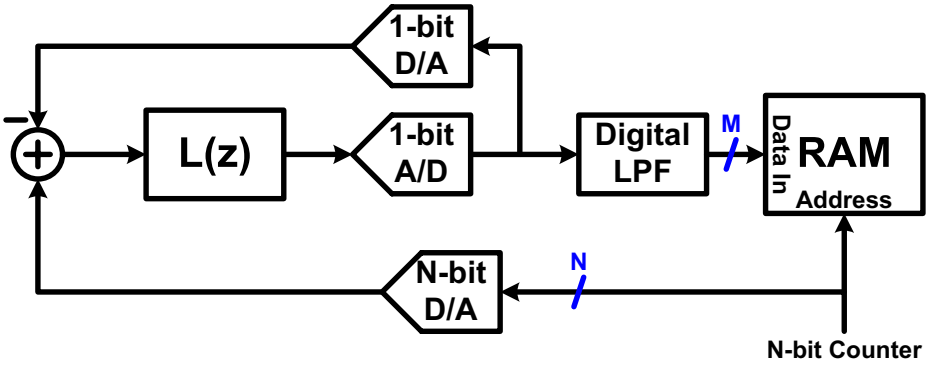
Digital Correction

- **Lookup table contains the equivalent of each DAC level**
  - **In practice, the look-up table only needs to store the differences between the actual and ideal DAC levels**
  - **Look-up table calibrated so that $V_{OUT} = V_{DAC}$**
    - **=> DAC errors are shaped by the loop**



Foreground Calibration

- Acquisition and storage of digital versions of DAC output signal (N-bit DAC, M-bit converter)
  - Each of the 2^N DAC codes is held for 2^M clock periods
  - With a 1-bit $\Delta\Sigma$ ADC, each DAC level is converted to its M-bit digital representation and stored in the RAM



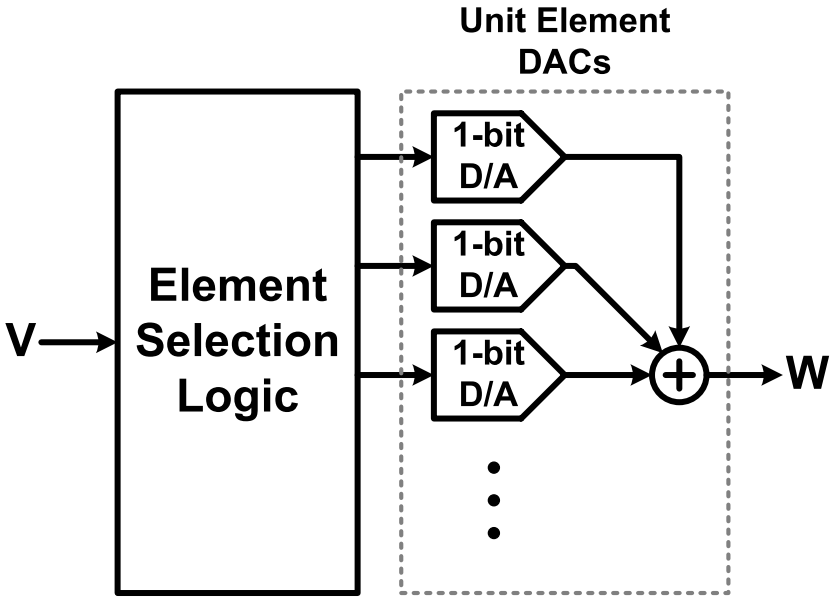
For background calibration, see Silva, CICC '02

Mismatch Shaping

- Ensures that element mismatch error results in shaped 'noise'
- Operates without knowledge of the actual mismatch errors
  - Even if the DAC errors drift, the output error will still be shaped
- Two requirements:
  - 1) Redundancy: There must be more than one way to create the same digital output (this is the case with thermometer coded outputs)
  - 2) Oversampling: Spectrally, there must be somewhere to put the unwanted mismatch noise

Mismatch Shaping

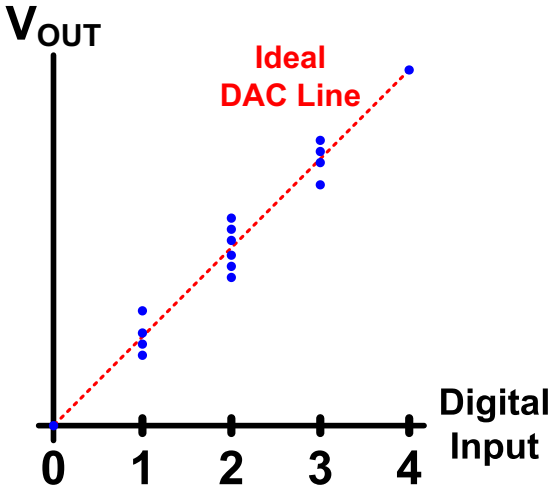
- Element Selection Logic chooses when to use each of the DAC elements



Mismatch Shaping

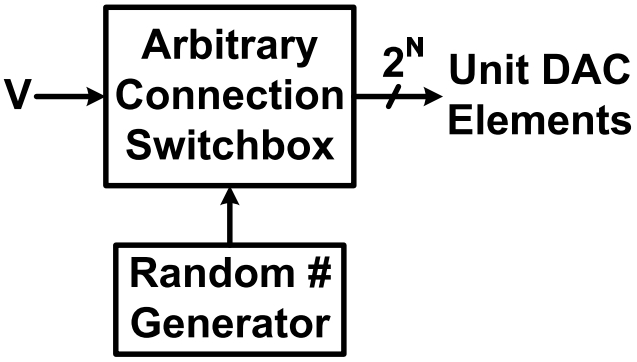
- Endpoints of DAC create ideal output curve
  - Assumes no gain/offset error

- Average value of DAC codes lies on ideal DAC characteristic line
  - Errors are symmetric about the characteristic DAC line
  - Mismatch shaping chooses DAC cells to keep the error bounded



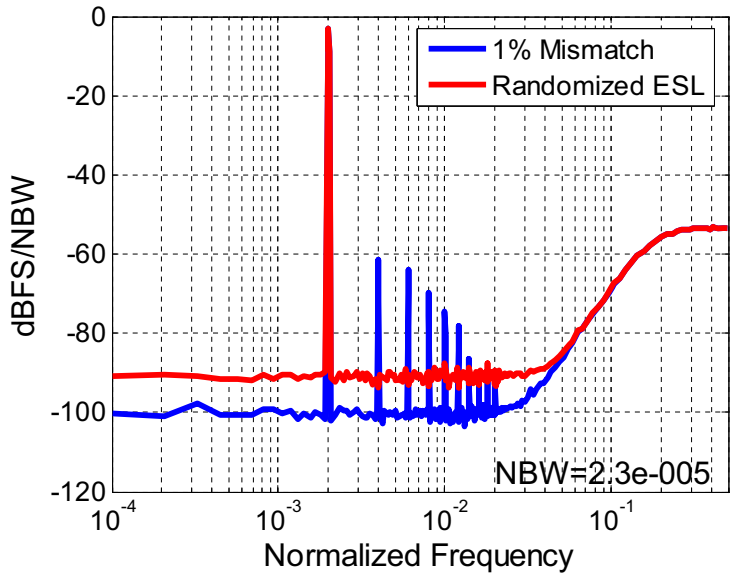
Element Randomization

- **Element selection logic randomly chooses DAC elements**
  - **For each thermometer-coded input K, the ESL randomly chooses K unit DAC elements**
  - **DAC error is no longer correlated with the input**
  - **Signal distortion is replaced by random noise spread throughout the entire spectrum**

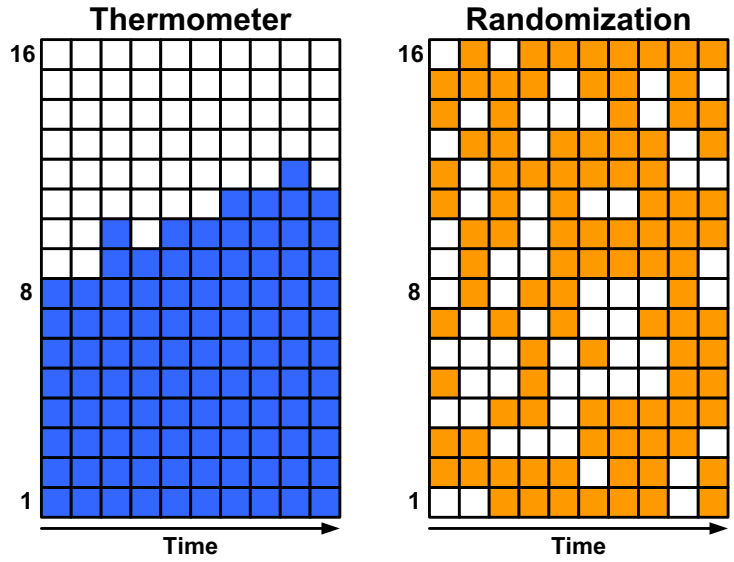


Element Randomization

- **SNDR = 62dB (improved from 50dB)**
  - **Distortion no longer present**
  - **Increased noise floor**



Element Usage Patterns

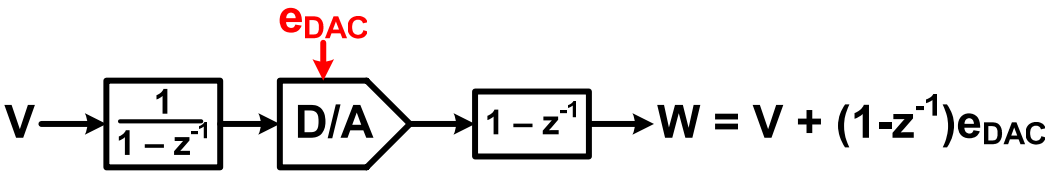


- **Randomization: All DAC levels are used even when the input is almost constant**

  (Thermometer coded => no ESL is used)

Data-Weighted Averaging

- **Data-directed element selection logic**
- **Conceptual system**

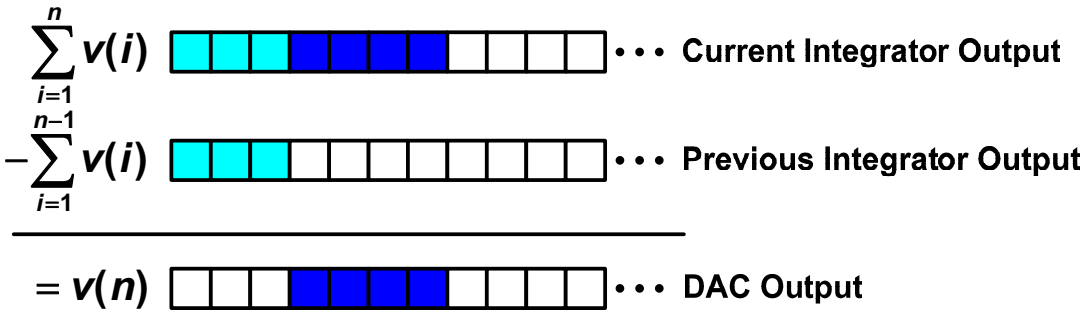


- **DAC error is noise-shaped (high-pass filtered)**

  But… DAC needs an infinite number of elements with the open-loop integrator

How can we implement this practically?

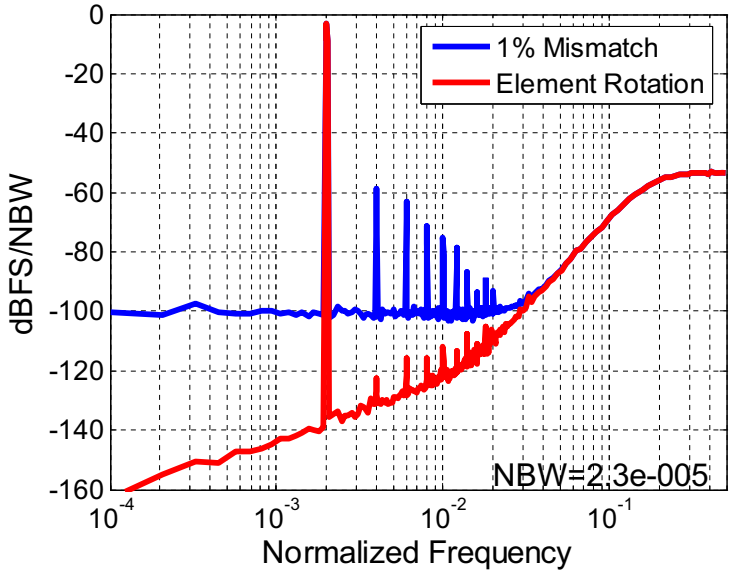
Element Rotation



- **Use the elements in a circular fashion**
  - **At time *n*, use the next *v(n)* elements in the array**
  - **Loop back around when end of array is reached**
  - **DAC error is noise shaped by desired $1-z^{-1}$ filter**

DWA / Element Rotation

- **SNDR = 97dB (47dB improvement)**
  - **Noise floor is reduced since error is shaped**
  - **Distortion is reduced (less correlated with input)**



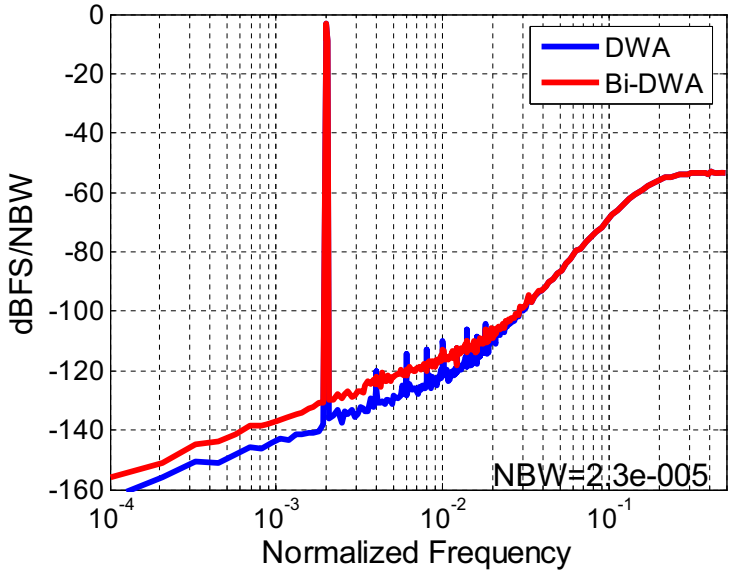
Bidirectional DWA

- **DWA can cause tone generation if the DAC input is not a busy random signal**
  - **Like MOD1, if DAC input is DC/slowly varying, tones are produced since DAC output can be periodic**

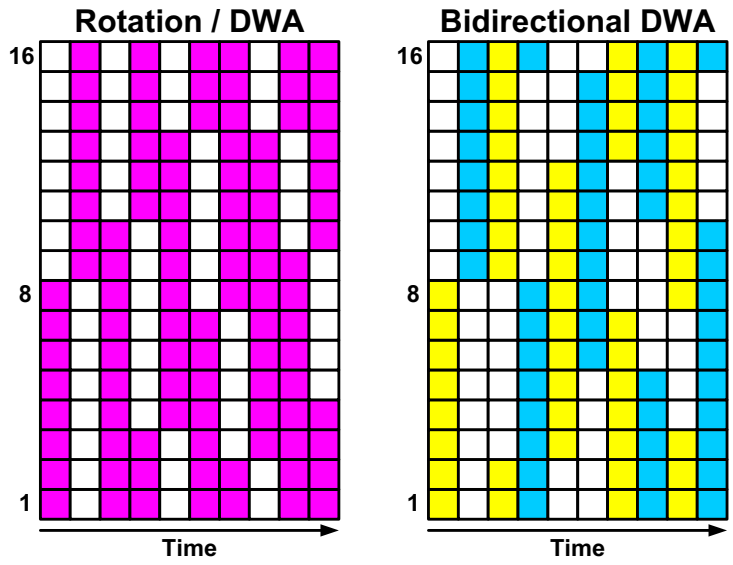
- **Bi-DWA**
  - **Element selection ping-pongs between two independent DWA algorithms, each rotating through DAC elements in opposite directions**
  - **Tends to reduce tonal behaviour, but also effectively decreases the OSR of the mismatch-shaping by a factor of 2**
  - **RMS mismatch noise increases by 9dB**

Bidirectional DWA

- **SNDR = 92dB (5dB worse than DWA)**
  - **Distortion is reduced (less correlated with input)**
  - **Noise floor is higher, SNDR less**



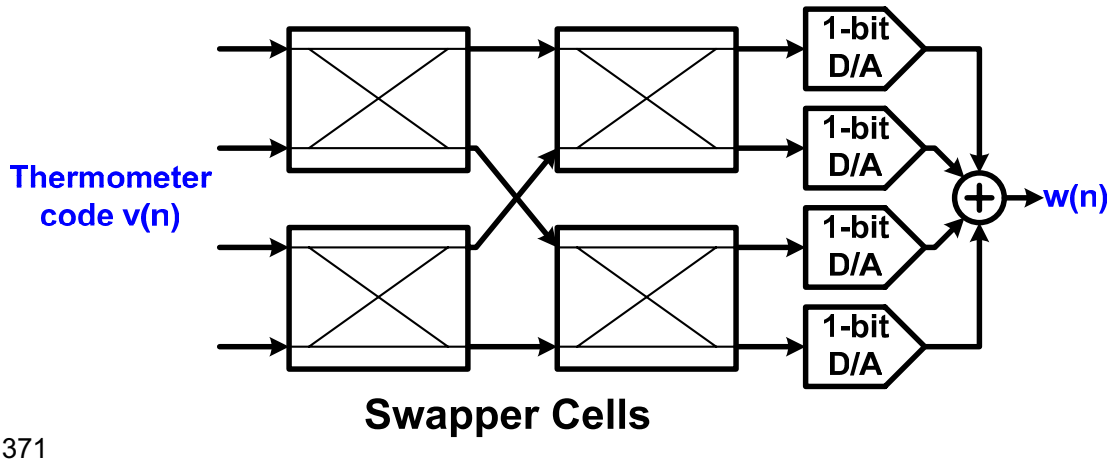
Element Usage Patterns



- **Bidirectional DWA: same as two DWA schemes operating independently and in reverse directions on opposing clocks**

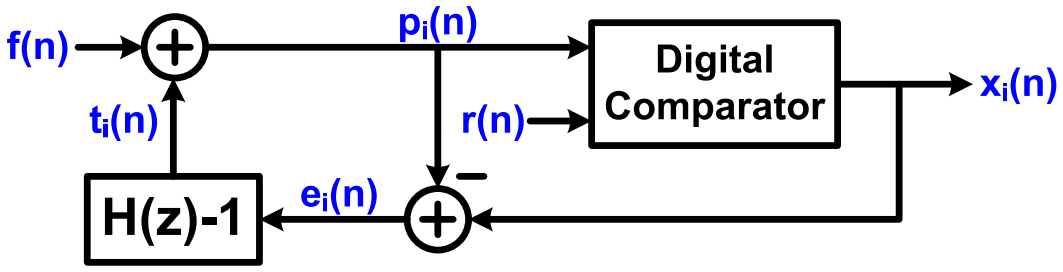
Alternative Scheme: Swapping

- **Each swapper tries to equalize the activity of its outputs [Adams, 1993]**
  - **Each element becomes a first-order noise-shaped sequence**
  - **Can be generalized to 2nd-order (Tree Structure)**



Vector-Based Mismatch Shaping

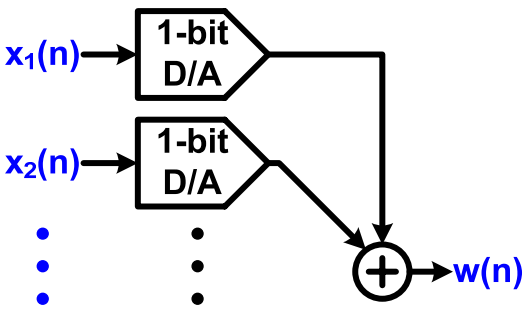
- Achieves higher-order noise spectral shaping
  - M digital noise-shaping loops for each unit element – an array of these loops make up the ESL
  - f(n) and r(n) are the same for all M loops
  - $x_i(n)$ controls the corresponding DAC element
  - H(z) determines the order of the noise shaping



Vector-Based Mismatch Shaping

- How is the output of the DAC noise shaped?
  - Output of the DAC elements $w(n) = \sum_{i=1}^{M} x_i(n)[1 + e_{DAC,i}]$
  - Loop filter outputs $x_i(n) = f(n) + [h * e_i](n)$
  - Resulting DAC element output is noise shaped by H where K is the intended DAC output

$$w(n) = K + h(n) * \sum_{i=1}^{M} [e_{DAC,i} e_i(n)]$$



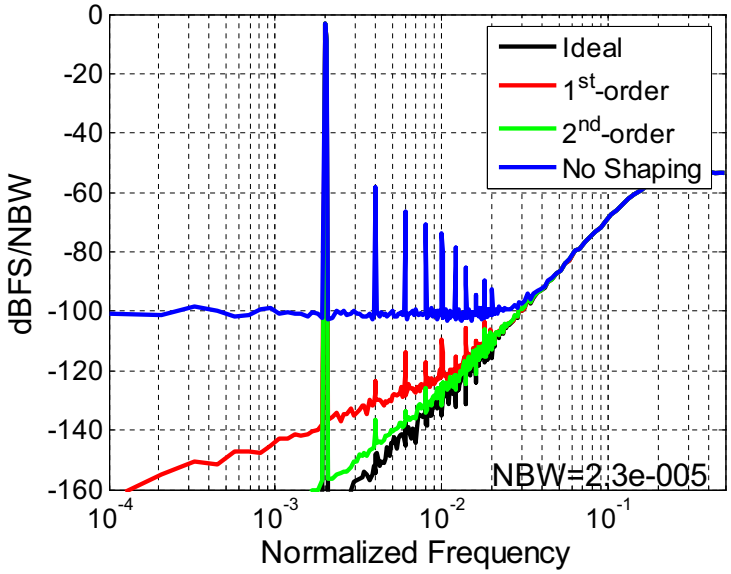
Vector-Based Mismatch Shaping

- **What are r(n) and f(n)?**
  - **r(n) and the digital comparators are not actually implemented – we need K outputs to be 1 so that w(n) is the correct value**
  - **The K largest $w_i(n)$ are quantized to $x_i(n)$ to minimize $e_i(n)$, which reduces the DAC error**
  - **f(n) is chosen to keep the data in the loop positive, but also as small as possible**

    => choose $f(n) = -\min_i [t_i(n)]$
- **What is H(z)?**
  - **H(z) is the NTF**
  - **For 1st-order, H(z) = 1 – z^{-1} (filter is z^{-1})**
  - **For 2nd-order, H(z) = $(1 - z^{-1})^2$ (filter is $2z^{-1} - z^{-2}$)**

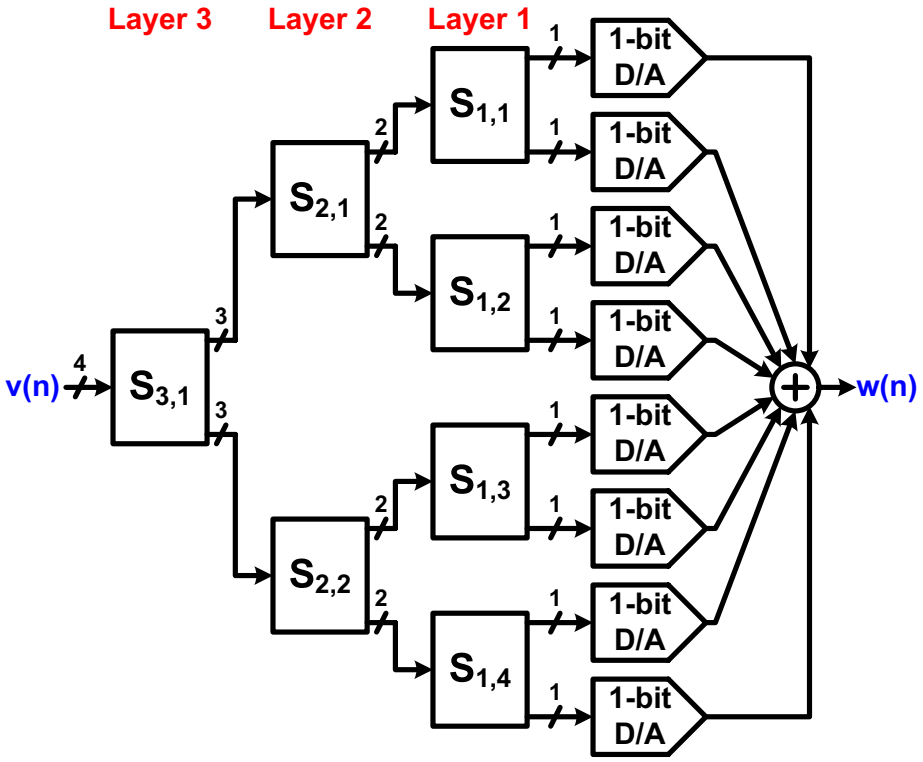
Vector-Based Shaping Spectrum

- **Ideal SNDR: 107dB, No shaping: 50dB**
  - **1st-order shaping: 97dB**
  - **2nd-order shaping: 105dB**



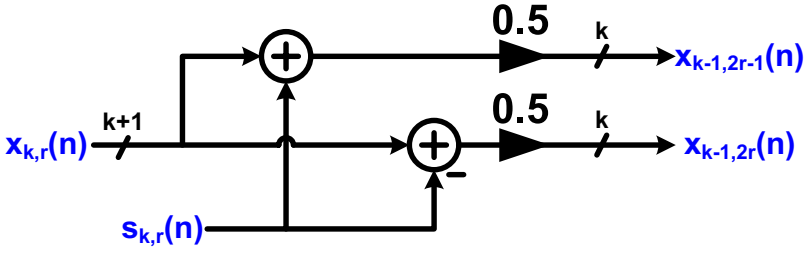
Tree Structure

- Also useful for higher-order DAC noise-shaping



Tree Structure

- Each switching block $S_{k,r}$ contains a unique sequence generator $s_{k,r}(n)$
  - k is the layer, r is the location within the layer
  - $s_{k,r}(n)$ is a 1-bit sequence that determines the noise shaping of the structure
  - Final DAC mismatch noise will be a weighted sum of the $s_{k,r}$ sequences
  - Sample 1st-order bit sequence 1,0,…,0,-1,0,…,0,1,0,…



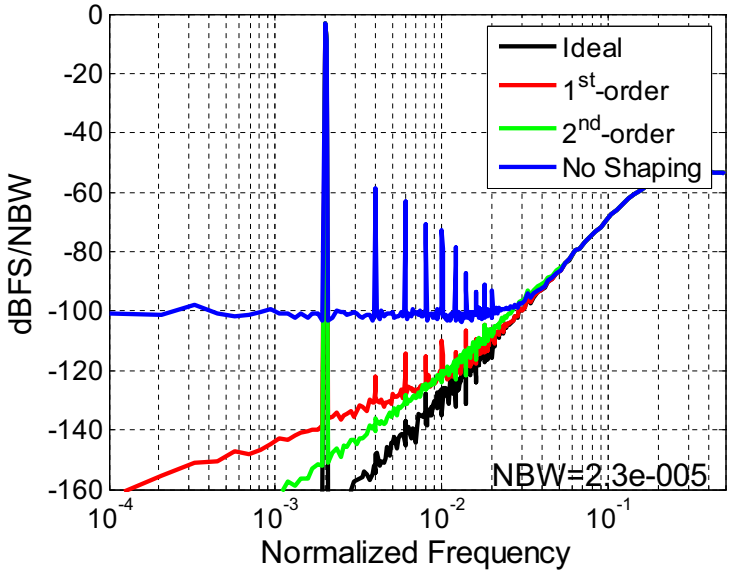
Tree Structure

- **Switching block must follow some rules**
  - **1) The two outputs of each switching block must be between 0 and 2^{k-1}**
  - **2) The sum of each switching block must equal the input**
  - **3) Each $s_{k,r}(n)$ L^{th}-order shaped sequence must be uncorrelated from the $s_{k,r}(n)$ sequences of the other blocks**

If these are satisfied, DAC noise will be an L^{th}-order shaped sequence [Galton, TCAS2 1997]

Tree Structure Spectrum

- **Ideal SNDR: 107dB, No shaping: 50dB**
  - **1^{st}-order shaping: 91dB**
  - **2^{nd}-order shaping: 100dB**



NLCOTD: Class-AB Output Bias

What You Learned Today

1. **Mismatch sources**
   Systematic, Random, Gradient

2. **Matching techniques**
   Common-centroid and interdigitation

3. **Mismatch shaping schemes**
   Randomization, Element Rotation, BiDWA, Swapping
   Vector-based, Tree Structure