# DAC Converters Design: Syllabus

**General definitions**

- Applications
- Basic principles and operating electrical domains
- Performance parameters

**Resistor-based DACs**

- R-string, R-string interpolated, and limits (DNL, INL analysis)
- R-2R ladder: binary weighting, DNL approximate analysis, configuration with a TIA.

**Current-source (CS) based DACs**

- Unary and binary weighted CSs. DNL/INL analysis for binary scales CSs. Segmentation.
- Design for high precision: gradients due to physical processing and due to routing, solutions based on ordered element switching and calibration techniques based on current copier

**Charge-scaling DACs**

- Charge redistribution based on parallel capacitor banks

INSIGHTS

# DAC Converters Design: Syllabus

**General definitions**

- Applications
- Basic principles and operating electrical domains
- Performance parameters

**Resistor-based DACs**

- R-string, R-string interpolated, and limits (DNL, INL analysis)
- R-2R ladder: binary weighting, DNL approximate analysis, configuration with a TIA.

**Current-source (CS) based DACs**

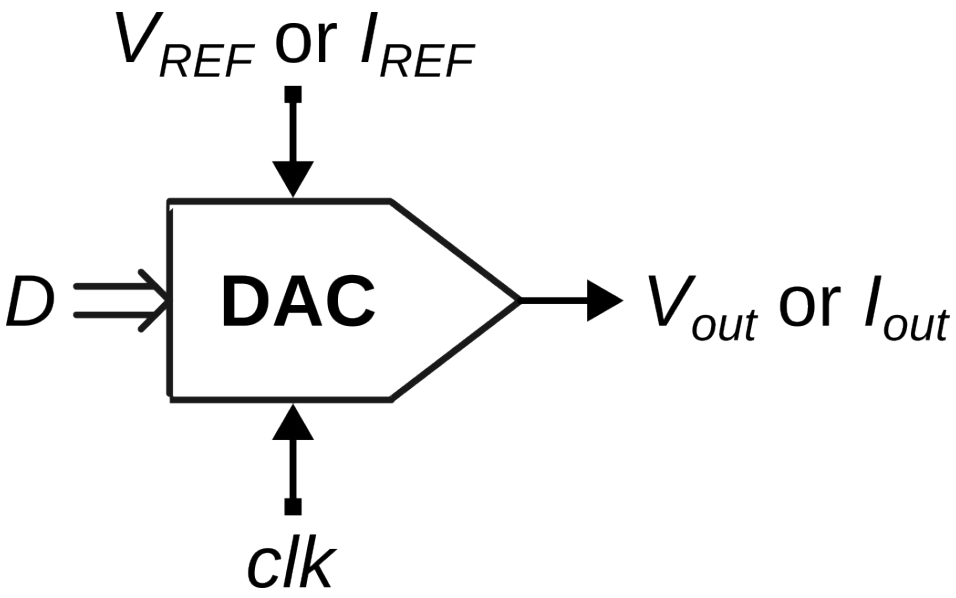
- Unary and binary weighted CSs. DNL/INL analysis for binary scales CSs. Segmentation.
- Design for high precision: gradients due to physical processing and due to routing, solutions based on ordered element switching and calibration techniques based on current copier

**Charge-scaling DACs**

- Charge redistribution based on parallel capacitor banks

INSIGHT S

# General definitions




**Input:** *D*, digital code of *N* bits. If unsigned: [0, $2^N$-1], if two's complement: [$-2^{N-1}$, $2^{N-1}$ – 1]

**Output:** analog signal (voltage or current) representing an analog equivalent of the input code, can be either single-ended or fully-differential

An analog **reference** is needed, to define the full-scale and minimum value

A **clock** almost always present for synchronization

**Ideal voltage-output DAC:**

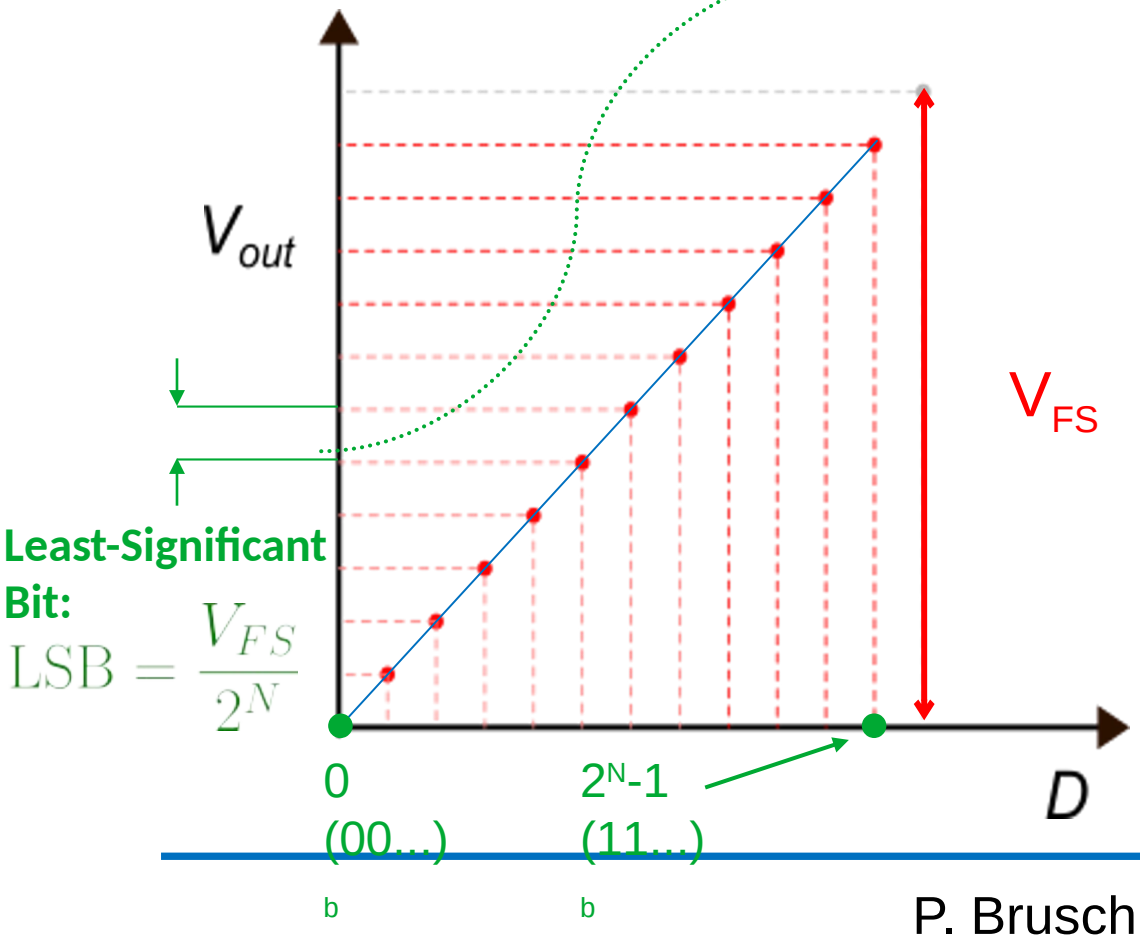
$$V_{out}(D) = V_{min} + \frac{V_{FS}}{2^N} \cdot D$$

**Ideal current-output DAC:**

$$I_{out}(D) = I_{min} + \frac{I_{FS}}{2^N} \cdot D$$

# General definitions: full-scale and LSB

$$V_{out}(D) = V_{min} + \frac{V_{FS}}{2^N} \cdot D$$




Typically in **Single-Ended DACs**:

- Unsigned input,
- unipolar output
- $V_{min}$ = 0 V (ground)
- $V_{FS}$ = gain * $V_{REF}$
- The gain is determined by circuit implementation

**Note:**

$V_{FS}$ is not available digitally, it represents the reference quantity to which $V_{out}$ is normalized.

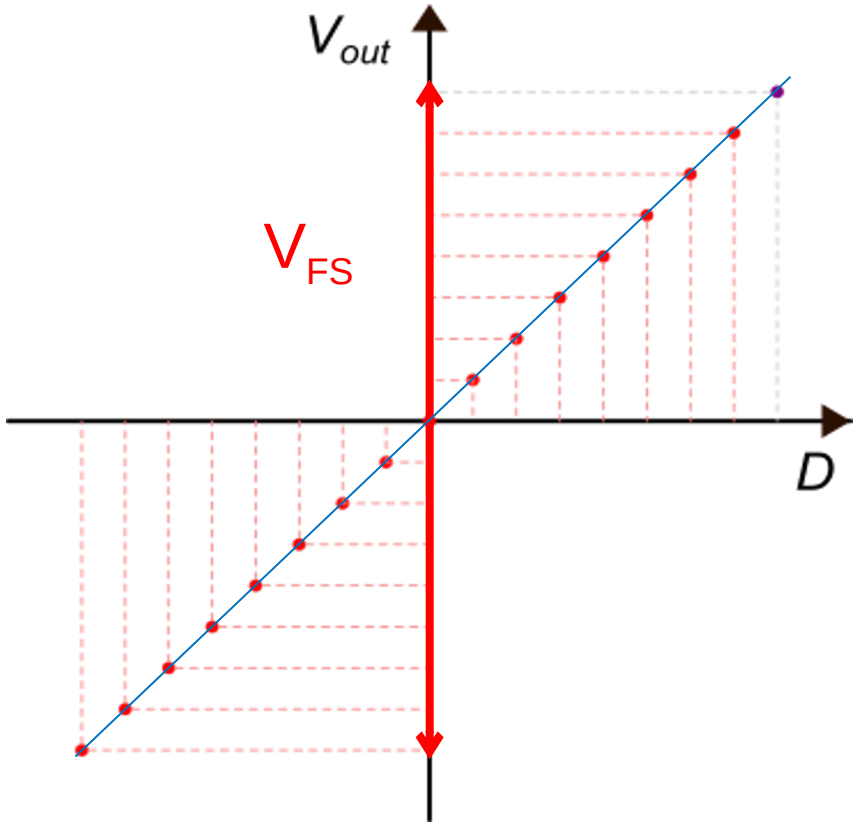
"Full-scale" is a naming convention!

The actual DAC output range is:

$$V_{min} \leq V_{out} \leq (1 - 2^{-N})V_{FS} + V_{min}$$

# General definitions: full-scale and LSB

$$V_{out}(D) = V_{min} + \frac{V_{FS}}{2^N} \cdot D$$

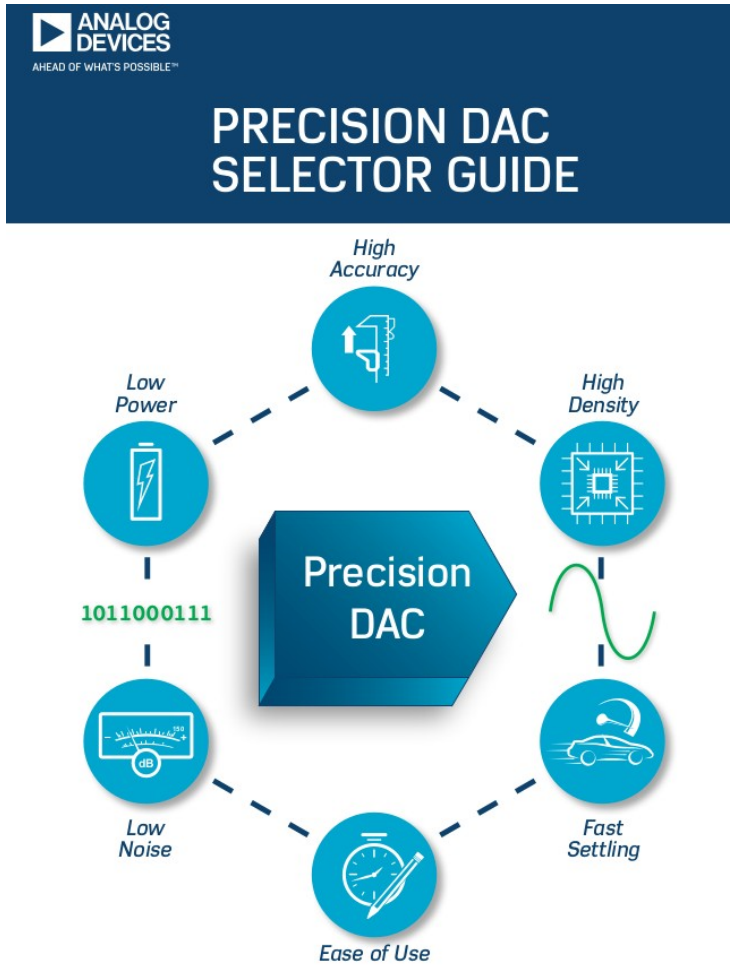



Typically in **Fully-Differential DACs**:

- Two's complement input
- Differential output
- $V_{min} = -V_{FS}/2$, also differential valued
- Output common mode >= to $V_{min}$

$$\begin{cases} V_{out,p} = V_{cmo} + \dfrac{V_{out}}{2} \\ V_{out,n} = V_{cmo} - \dfrac{V_{out}}{2} \end{cases}$$

# DAC Applications




**Digital signal reconstruction in DSP-based systems:**
LED displays, Direct Digital Synthesis, Medical equipment. Often high-accuracy and high-resolution (16-20 bits) is needed

**Actuation of sensory systems**
Electrochemical sensors, MEMS, antenna arrays, ultrasound actuators. Often multichannel DACs and/or High-voltage requirements (20-40 V). Compactness for low-cost devices.

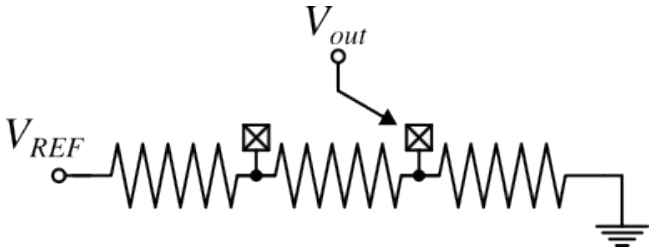
**Wireless/wired communications**
High-speed requirements, up to 10 GS/s. The faster, the better

**Embedded block for calibration or as component of an ADC**
Offset nulling, trimming, Delta-Sigma Modulator, SAR ADC. Usually ultra-low power requirements

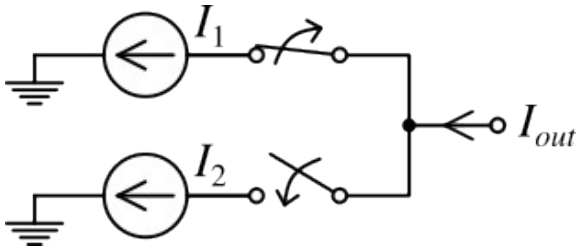
# Operating electrical domains

**Fundamental functions needed in a DAC:** add (perform a linear combination of homogeneous quantities) and select

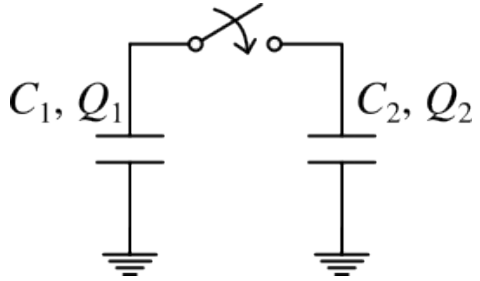
**In voltage domain:** Voltage piles up in a series of resistive elements biased at VREF, switches are controlled to tap into the desired voltage (digital potentiometer).




**In current domain:** currents adds up to a common mode (Kirchhoff law of currents). Selected currents are routed to the summing node.




**In charge domain:** pre-charged capacitors are connected together to set a final voltage resulting as (linear) combination of the individual charges



# Typical performances vs. operating mode

| Operating domain | Architectures | Accuracy | Current consumption range | Max Speed |
|---|---|---|---|---|
| **Voltage** | Resistor string, R-2R | 8-16 bit<br>**HIGH RESOLUTION IF TRIMMED** | 1 µA – 1 mA | 1 MS/s |
| **Current** | Current sources (CS) | 10-16 bit<br>**HIGH RESOLUTION IF CALIBRATED** | 1 µA – 100 mA | **10 GS/s**<br>**HIGH SPEED** |
| **Charge** | Capacitor bank | 10-12 bit | **1 nA – 10 µA**<br>**LOW POWER** | 100 MS/s |

**Typical choices for different requirements: resolution, speed, power**

Higher resolution? Delta-Sigma approaches, will be discussed later on

# Main performance parameters

**Precision/accuracy** limited by:

- quantization error (not DAC's fault)
- offset
- gain
- static non linearity (DNL, INL)
- dynamic non linearity (glitches)

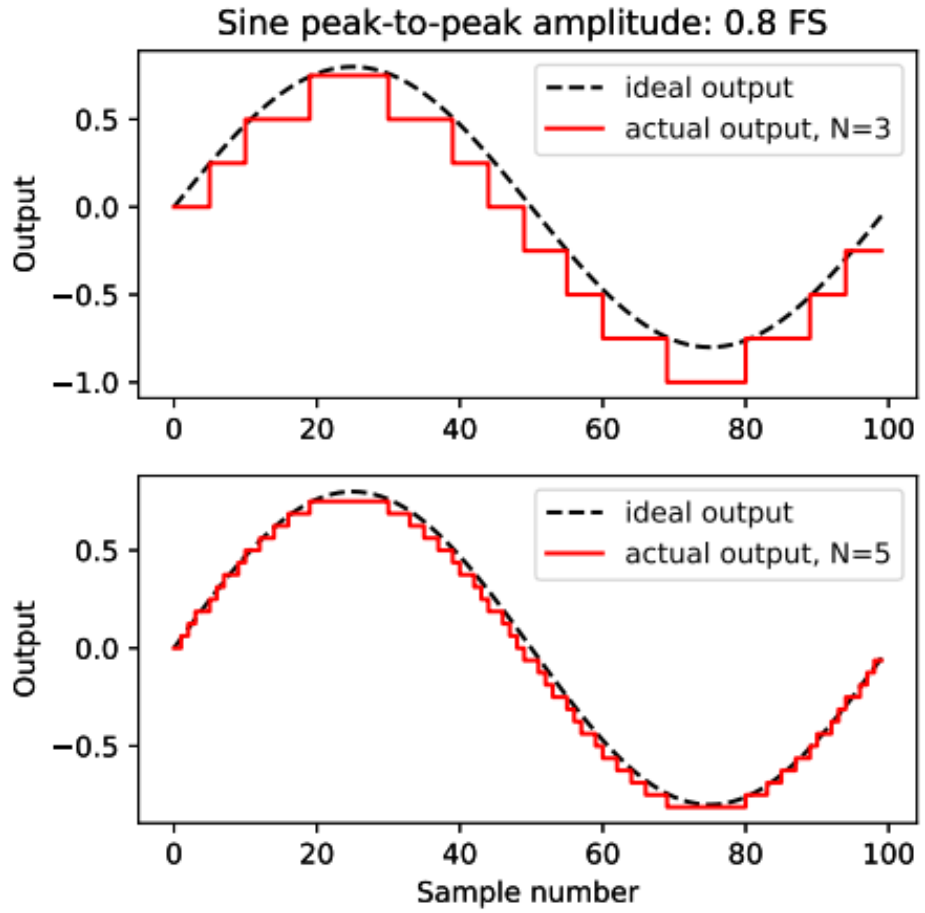
**Speed** is limited by:

- settling transients,
- load,
- operating output voltage

**Power consumption** is determined by:

- electrical operating domain,
- bandwidth requirements

Ideal waveform generator:



# Main performance parameters

**Precision/accuracy** limited by:

- quantization error (not DAC's fault)
- offset
- gain
- static non linearity (DNL, INL)
- dynamic non linearity (glitches)

**Speed** is limited by:

- settling transients,
- load,
- operating output voltage

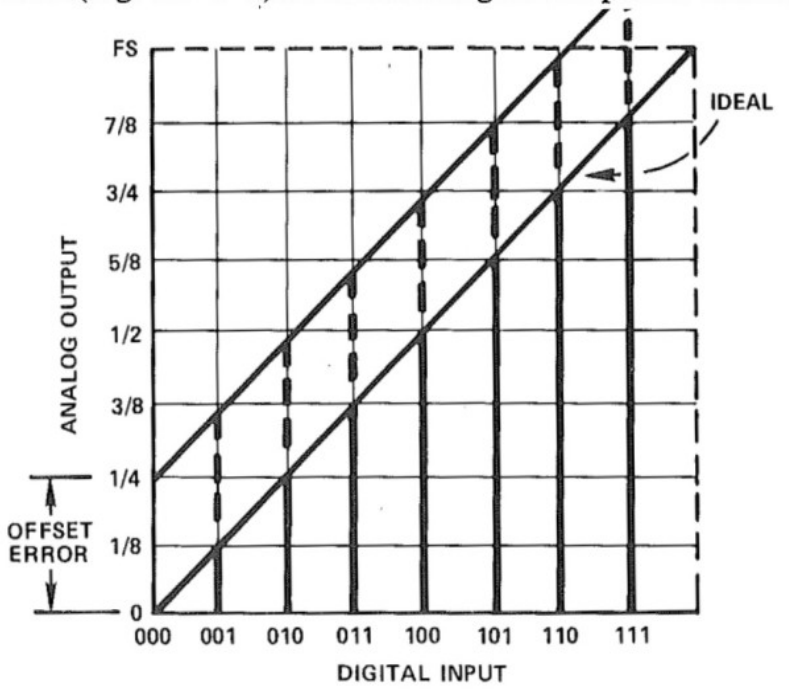
**Power consumption** is determined by:

- electrical operating domain,
- bandwidth requirements

*Offset Error*

Figure 10.2 illustrates the result of offset error only. The actual transfer function is offset from the ideal by two LSBs. Any such error—either positive or negative—that affects all codes by the same amount is an *offset* error.

In most DAC testing, the offset error is measured by applying the zero-scale code (e.g. all "0"s) and measuring the output deviation from 0 volts.




**THE DC SPECTRAL COMPONENT IS ALTERED**



https://www.analog.com/en/education/education-library/analog-digital-conversion-

# Main performance parameters

**Precision/accuracy** limited by:

- quantization error (not DAC's fault)
- offset
- gain

→ **END-POINT ERRORS**

- static non linearity (DNL, INL)
- dynamic non linearity (glitches)

**Speed** is limited by:

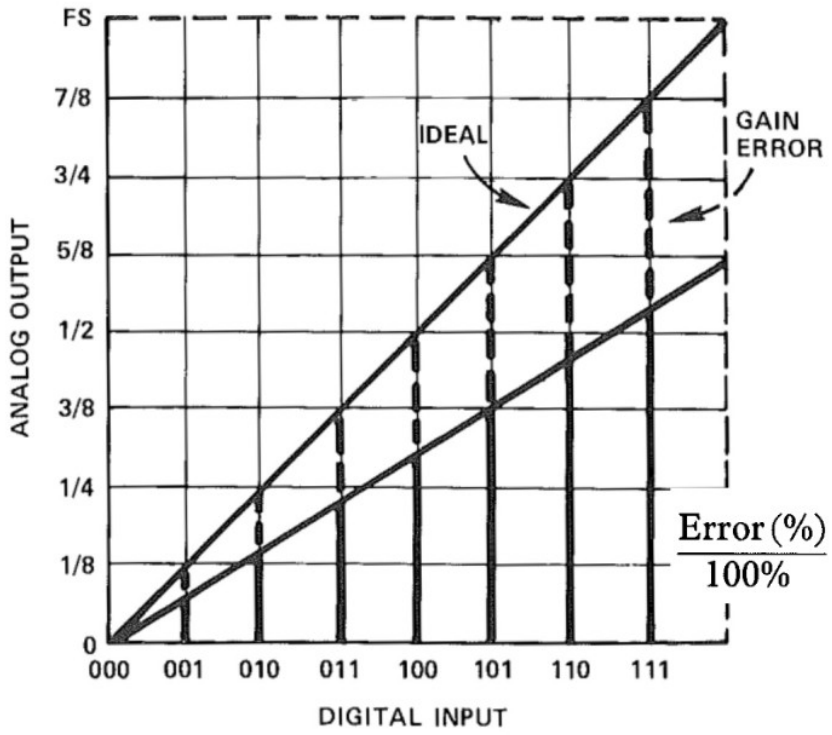
- settling transients,
- load,
- operating output voltage

**Power consumption** is determined by:

- electrical operating domain,
- bandwidth requirements

*Gain Error*

Figure 10.3 shows the effect of a gain error only. The ideal transfer function has a slope defined by drawing a straight line through the two end points. The slope represents the gain of the transfer function. In real DACs, this slope can differ from the ideal, resulting in a *gain* error—which is usually expressed as a percent because it affects each code by the same percentage.




$$\frac{\text{Error}(\%)}{100\%} = \frac{V_{11} - V_{os}}{V_{FSR}(1 - 2^{-n})} - 1$$

**NO NEW SPECTRAL COMPONENTS ARE GENERATED**



https://www.analog.com/en/education/education-library/analog-digital-conversion-

# Main performance parameters

**Precision/accuracy** limited by:

- quantization error (not DAC's fault)
- offset
- gain
- static non linearity (DNL, INL)
- dynamic non linearity (glitches)

**Speed** is limited by:

- settling transients,
- load,
- operating output voltage

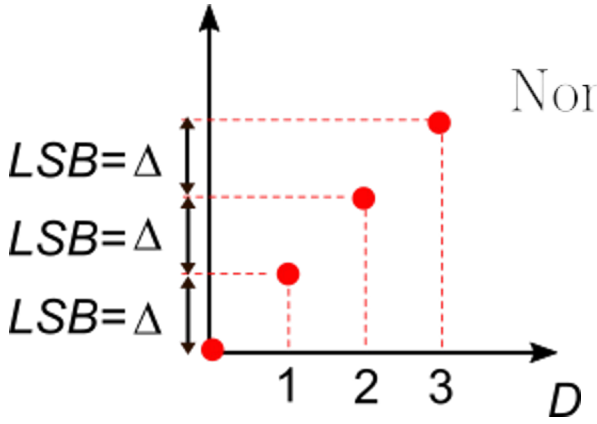
**Power consumption** is determined by:

- electrical operating domain,
- bandwidth requirements

REFERS TO THE STATIC CHARACTERISTICS

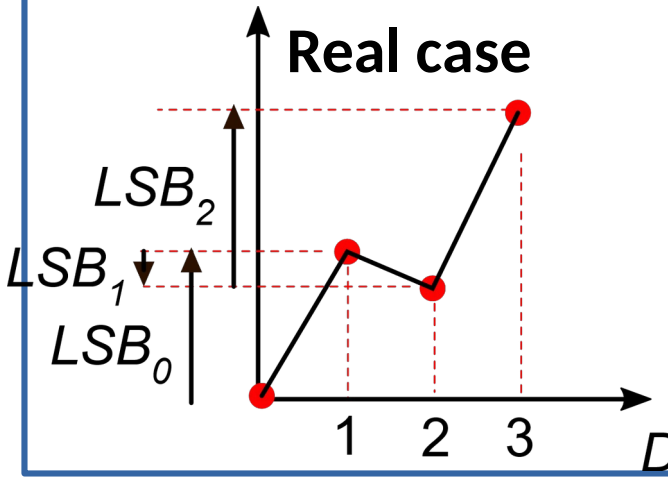
**Ideal case**

(absence of static non-linearity errors)




$$\text{Nominal LSB} = \Delta = \frac{V_{FS}}{2^N}$$

(calculated after end-point errors have been calibrated)




**Differential Non-Linearity**

Error between adjacent codes (quantitative measure of monotonicity)

$$\text{DNL}_k = \frac{\text{LSB}_k - \Delta}{\Delta}$$

# Main performance parameters

$$DNL_k = \frac{LSB_k - \Delta}{\Delta}$$

**Precision/accuracy** limited by:

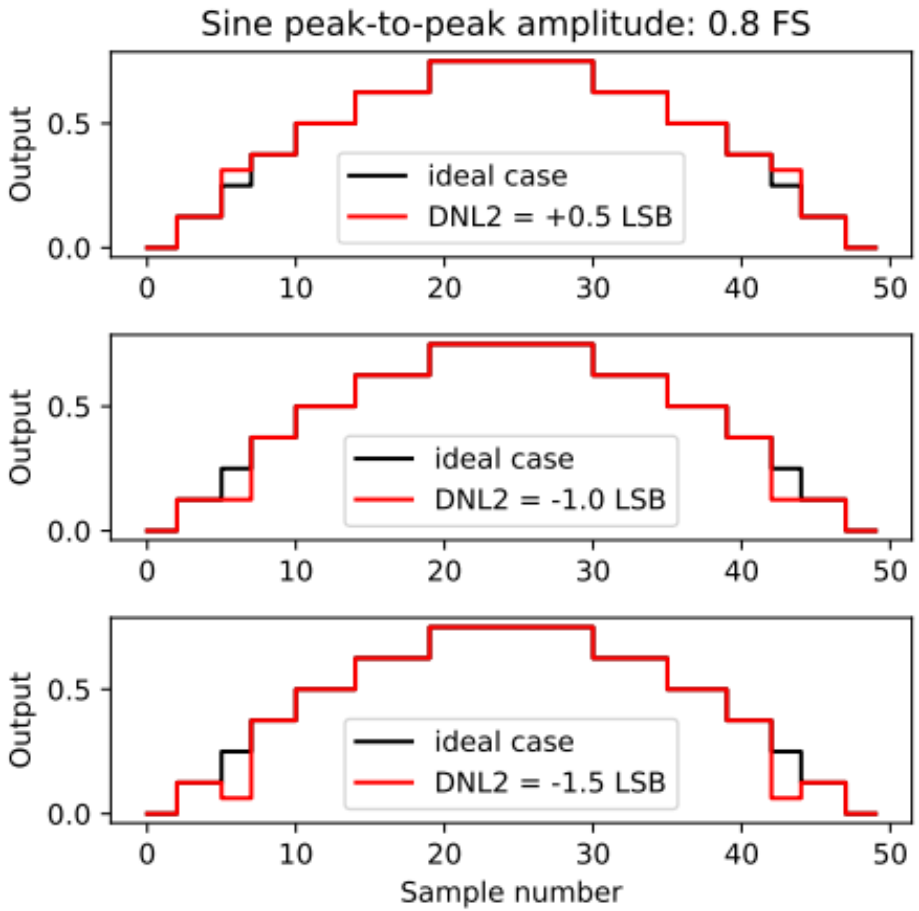
- quantization error (not DAC's fault)
- offset
- gain
- static non linearity (DNL, INL)
- dynamic non linearity (glitches)

**Speed** is limited by:

- settling transients,
- load,
- operating output voltage

**Power consumption** is determined by:

- electrical operating domain,
- bandwidth requirements




Missing code

*DNL*<-1 LSB: Non-monotonic

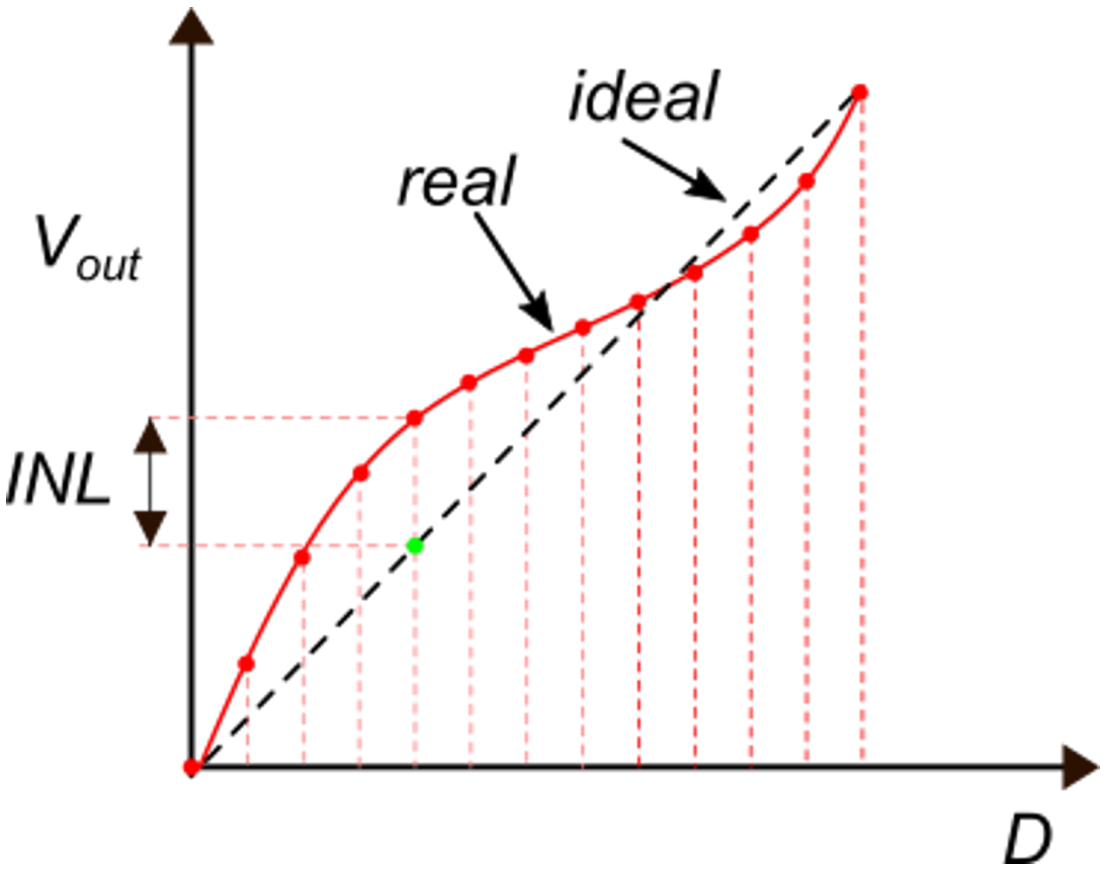
**NEW SPECTRAL COMPONENTS ARE GENERATED**

# Main performance parameters

**Precision/accuracy** limited by:
- quantization error (not DAC's fault)
- offset
- gain
- static non linearity (DNL, INL)
- dynamic non linearity (glitches)

**Speed** is limited by:
- settling transients,
- load,
- operating output voltage

**Power consumption** is determined by:
- electrical operating domain,
- bandwidth requirements




**Integral Non-Linearity:** maximum deviation of actual Vout from its ideal value (after end-point calibration)

**NEW SPECTRAL COMPONENTS ARE GENERATED**

# Main performance parameters

**Precision/accuracy** limited by:

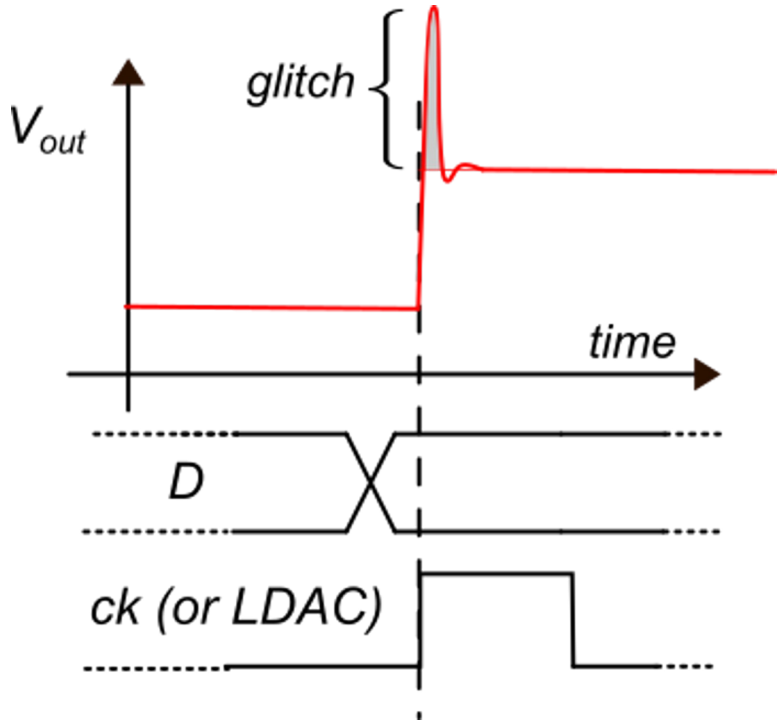
- quantization error (not DAC's fault)
- offset
- gain
- static non linearity (DNL, INL)
- dynamic non linearity (glitches)

**Speed** is limited by:

- settling transients,
- load,
- operating output voltage

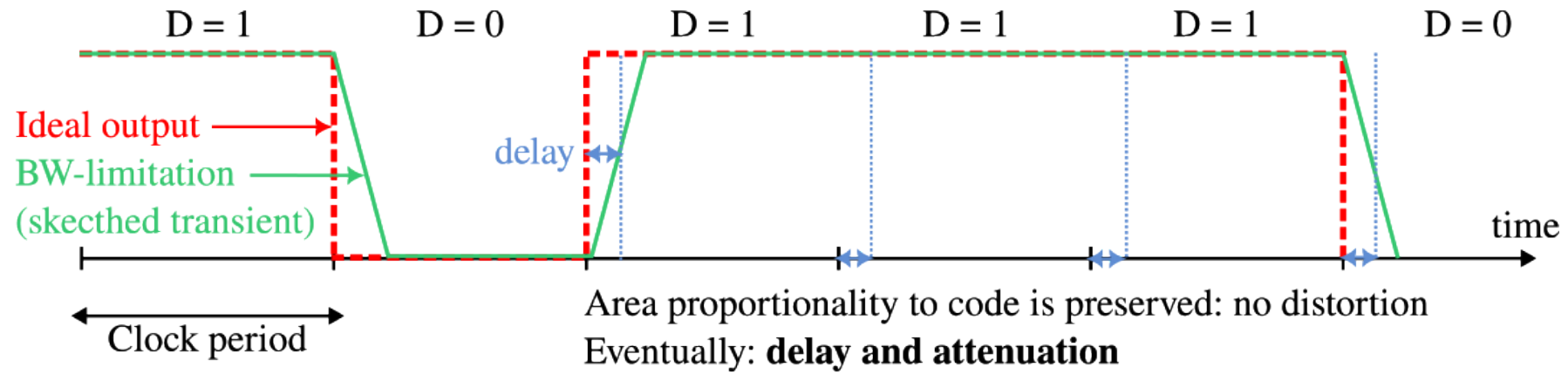
**Power consumption** is determined by:

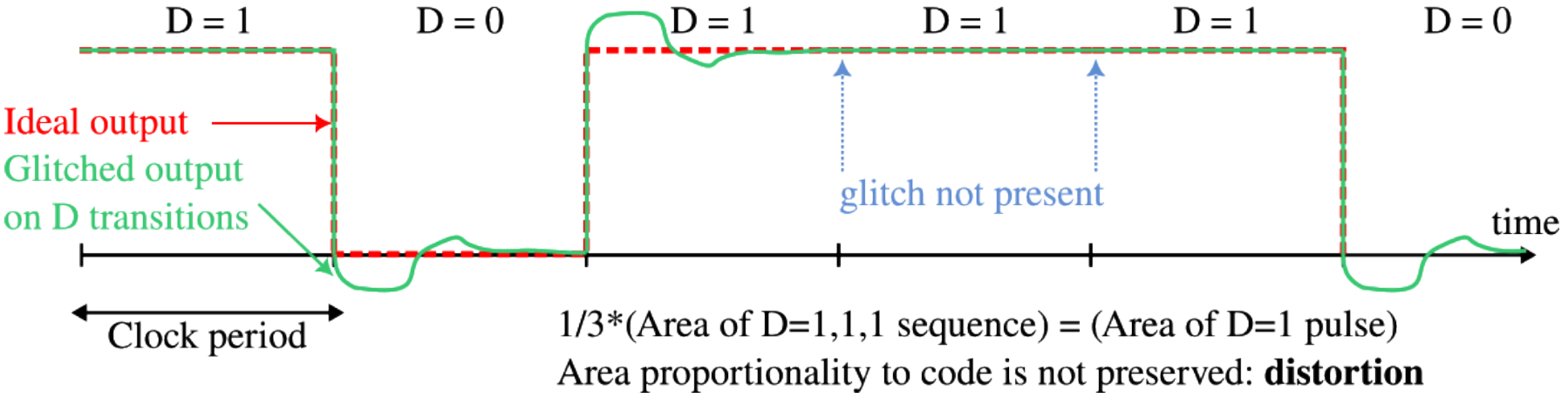
- electrical operating domain,
- bandwidth requirements




*ck* (or *LDAC*) acts as sync. signal, after *D* becomes stable.
If *Tck* is large enough, the energy (area) of the glitch is negligible. As *Tck* shrinks, the glitch energy is more and more important

# Main performance parameters



# DAC Converters Design: Syllabus

**General definitions**

- Applications
- Basic principles and operating electrical domains
- Performance parameters

**Resistor-based DACs**

- R-string, R-string interpolated, and limits (DNL, INL analysis)
- R-2R ladder: binary weighting, DNL approximate analysis, configuration with a TIA.

**Current-source (CS) based DACs**

- Unary and binary weighted CSs. DNL/INL analysis for binary scales CSs. Segmentation.
- Design for high precision: gradients due to physical processing and due to routing, solutions based on ordered element switching and calibration techniques based on current copier

**Charge-scaling DACs**

- Charge redistribution based on parallel capacitor banks

INSIGHTS

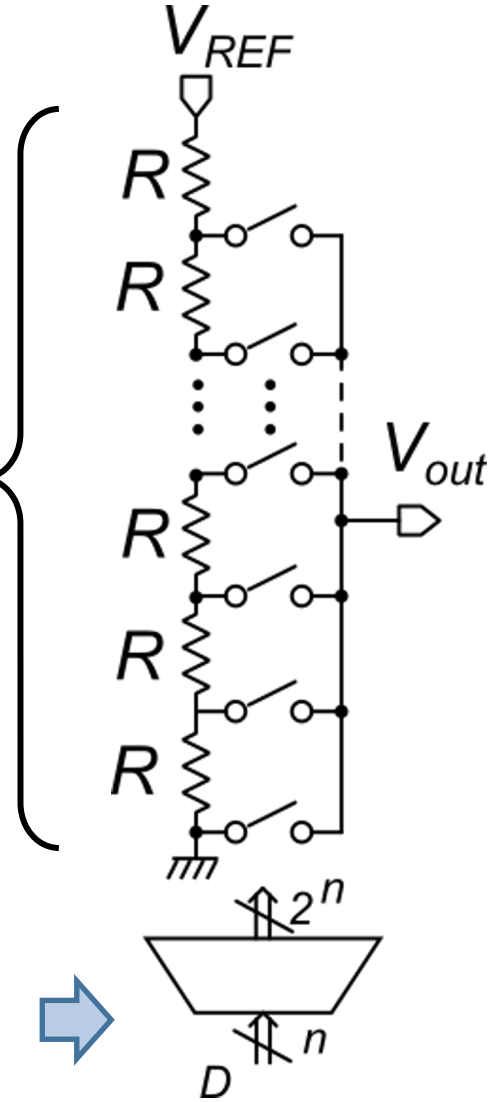
# R-String DAC

$$V_{out} = \frac{D}{2^n} V_{REF}$$

$2^n$ resistors

$2^n$ switches

**# of device number**




Digital decoder

$n \Rightarrow 2^n$

**First idea:**

Lord Kelvin in the mid-1800s, implemented using resistors and relays

- The string operates as **voltage divider** at each tap:

$$V_{tap} = \frac{R_{down}}{R_{down} + R_{up}} V_{REF}$$

- The accuracy of the partition voltage do depends on how well each resistor is matched to the others, not on the absolute accuracy of each resistor.
- Resistors constructed on the same substrate, deposited and processed together: inherent matching accuracy advantage over discrete solutions.

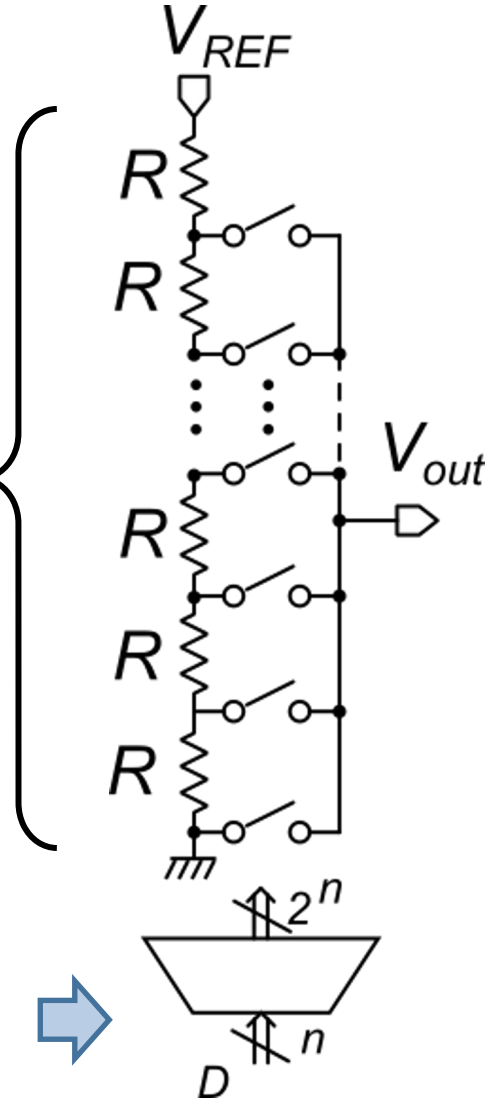
# R-String DAC

$$V_{out} = \frac{D}{2^n} V_{REF}$$

$2^n$ resistors

$2^n$ switches

**# of device number**




Digital decoder

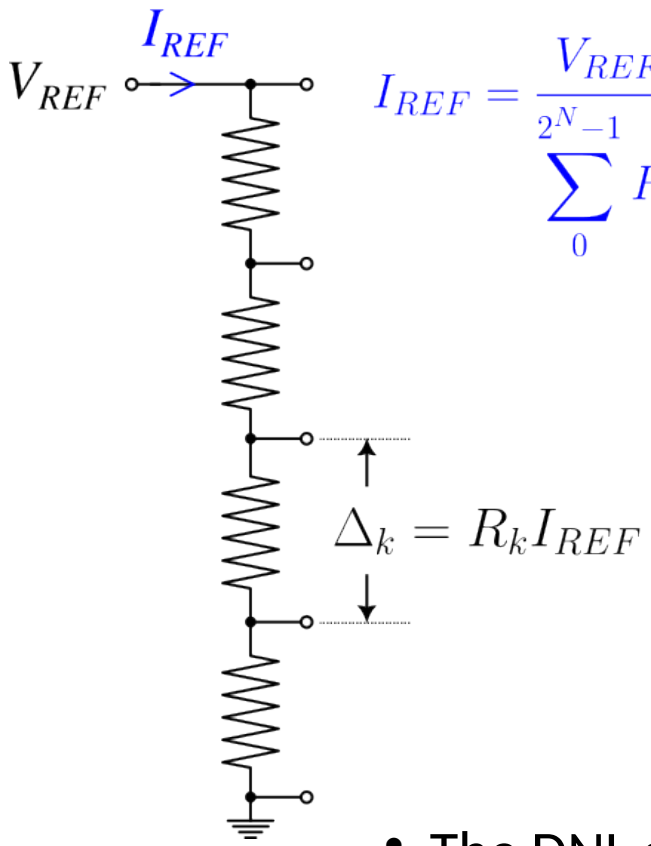
$n \Rightarrow 2^n$




**Operation**

- For any value of code D, only one switch at a time is on, selecting one of the $2^N$ voltage levels produced by the resistive voltage divider
- If code D turns on a certain switch, code D+1 turns on the switch placed in position up, then the output voltage can only grow
- The monotonicity is guaranteed, $|DNL| < 1$ LSB
- Usually followed a unity gain buffer to avoid string loading (however, opamp offset and noise are now to be considered)

# R-String DAC: DNL




$$I_{REF} = \frac{V_{REF}}{\sum_{0}^{2^N-1} R_i}$$

$$\mathrm{LSB} = \Delta = \frac{V_{REF}}{2^N} = \frac{\sum_{0}^{2^N-1} R_k}{2^N} \cdot I_{REF}$$

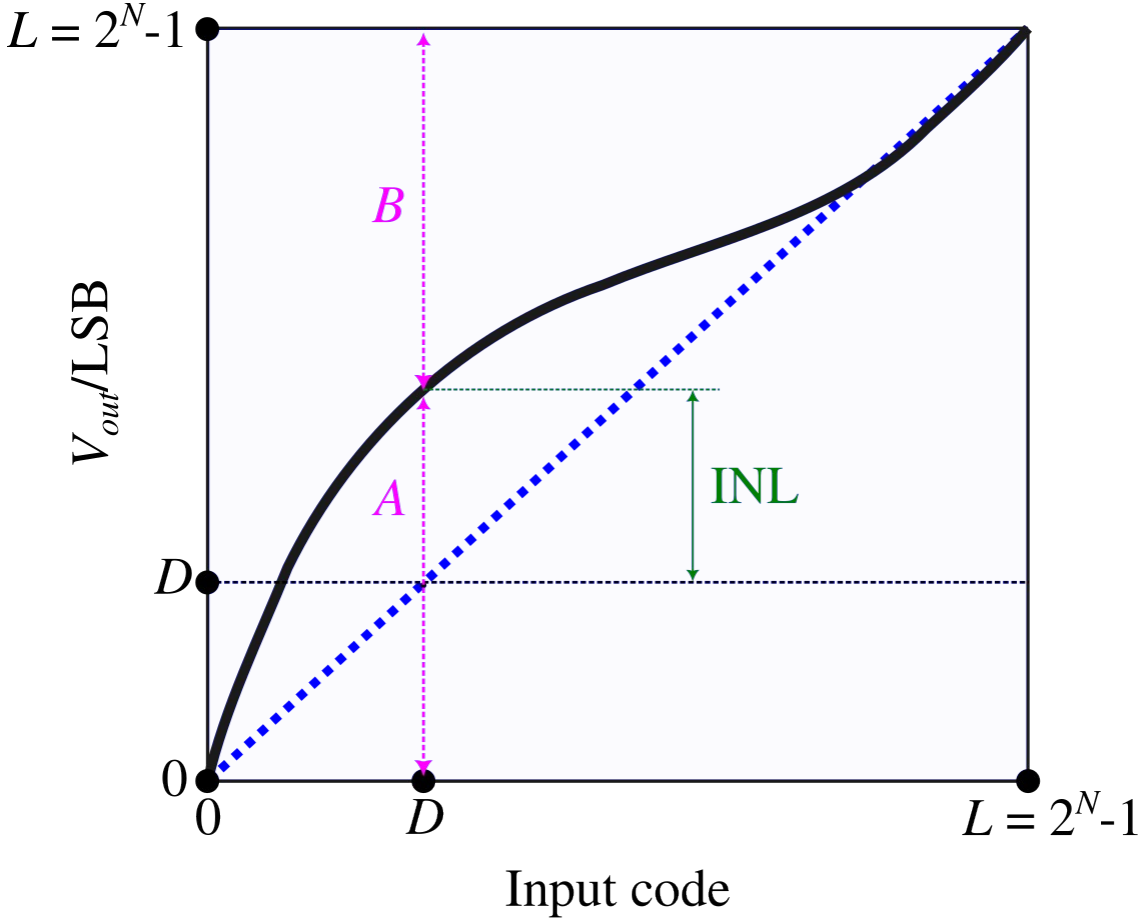
$$\overline{R}$$

$$\mathrm{DNL}_k = \frac{\Delta_k - \Delta}{\Delta} = \frac{R_k - \overline{R}}{\overline{R}} \approx \frac{\Delta R_k}{R}$$

$$\sigma_{\mathrm{DNL}} = \sigma_{\Delta R/R}$$

- The DNL standard deviation coincides with the standard deviation of ΔR/R (element mathing): **guaranteed monotonicity**
- DNL does not dependent of resolution (N)
- DNL does not depend on absolute value of R

# R-String DAC: INL




$$\mathrm{INL}(D) = A - D = A - \left(\frac{A+B}{A+B}\right) \cdot D$$

$$= A - \left(\frac{A+B}{L}\right) \cdot D = A \cdot \left(1 - \frac{D}{L}\right) + B \cdot \frac{D}{L}$$

$$\sigma^2_{\mathrm{INL}}(D) = \sigma^2_A \cdot \left(1 - \frac{D}{L}\right)^2 + \sigma^2_B \cdot \left(\frac{D}{L}\right)^2$$

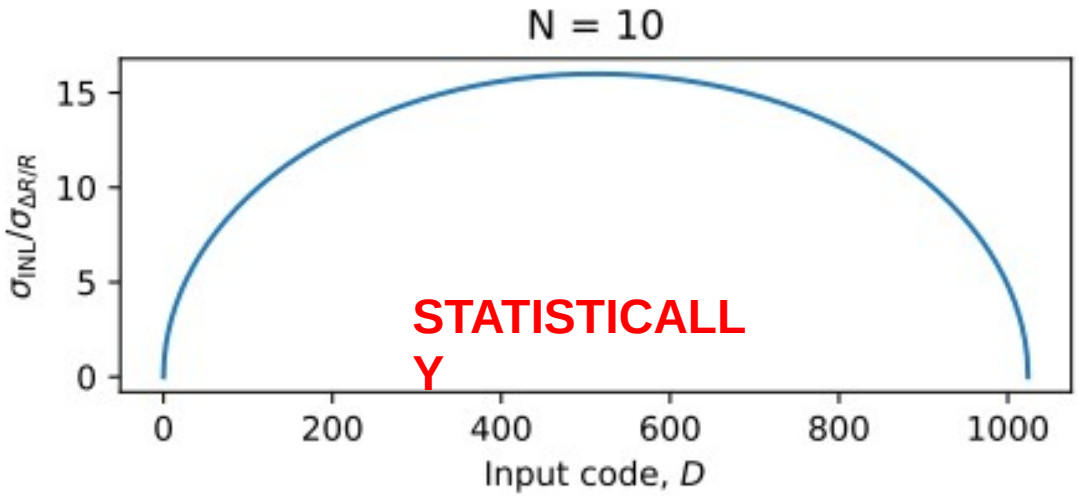
D series elements — L-D series elements

$$\sigma^2_{\mathrm{INL}}(D) = D\sigma^2_{\Delta R/R}\left(1 - \tfrac{D}{L}\right)^2 + (L-D)\sigma^2_{\Delta R/R}\left(\tfrac{D}{L}\right)^2$$

$$\sigma_{\mathrm{INL}}(D) = \sqrt{D\left(1 - \tfrac{D}{L}\right)} \cdot \sigma_{\Delta R/R}$$

# R-String DAC: INL

$$\sigma_{\mathrm{INL}}(D) = \sqrt{D\left(1-\frac{D}{L}\right)} \cdot \sigma_{\Delta R/R}$$




The INL standard deviation is maximum at half code

$$\sigma_{\mathrm{INL,max}} = \frac{\sqrt{L}}{2} \cdot \sigma_{\Delta R/R} \approx 2^{N/2-1} \cdot \sigma_{\Delta R/R}$$

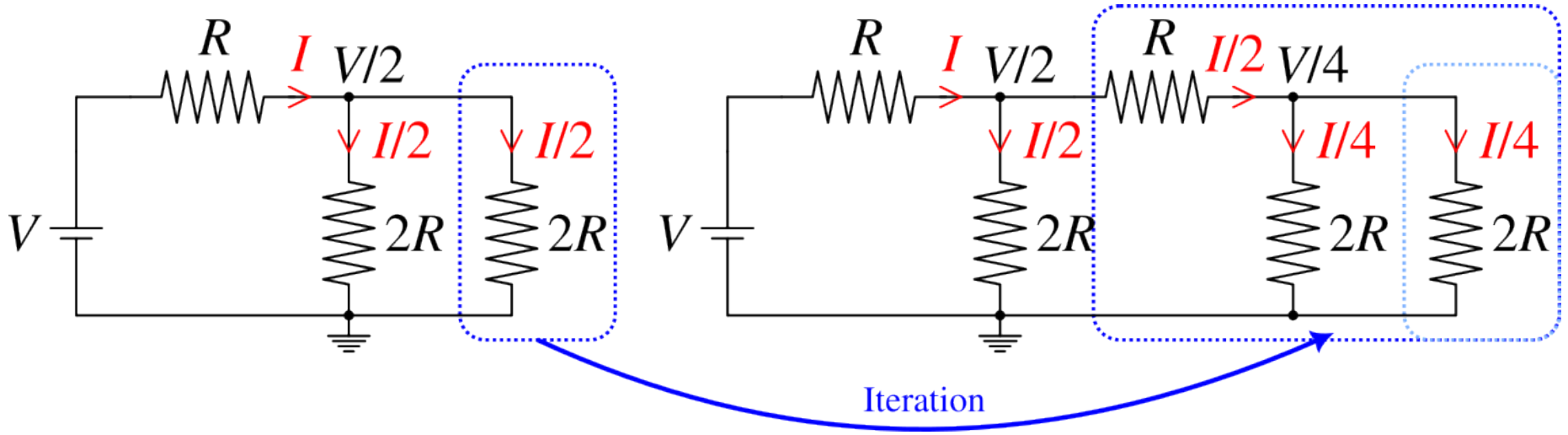
The INL standard deviation depends on the DAC resolution:

**Design case:** relative std. deviation of resistors 1%, what is the maximum number of bits for INL standard deviation < 0.1 LSB?

$$N < 2 + 2\log_2 \frac{\sigma_{\mathrm{INL,max}}}{\sigma_{\Delta R/R}} = 2 + 2\log_2 \frac{0.1}{0.01} \approx 8.6$$

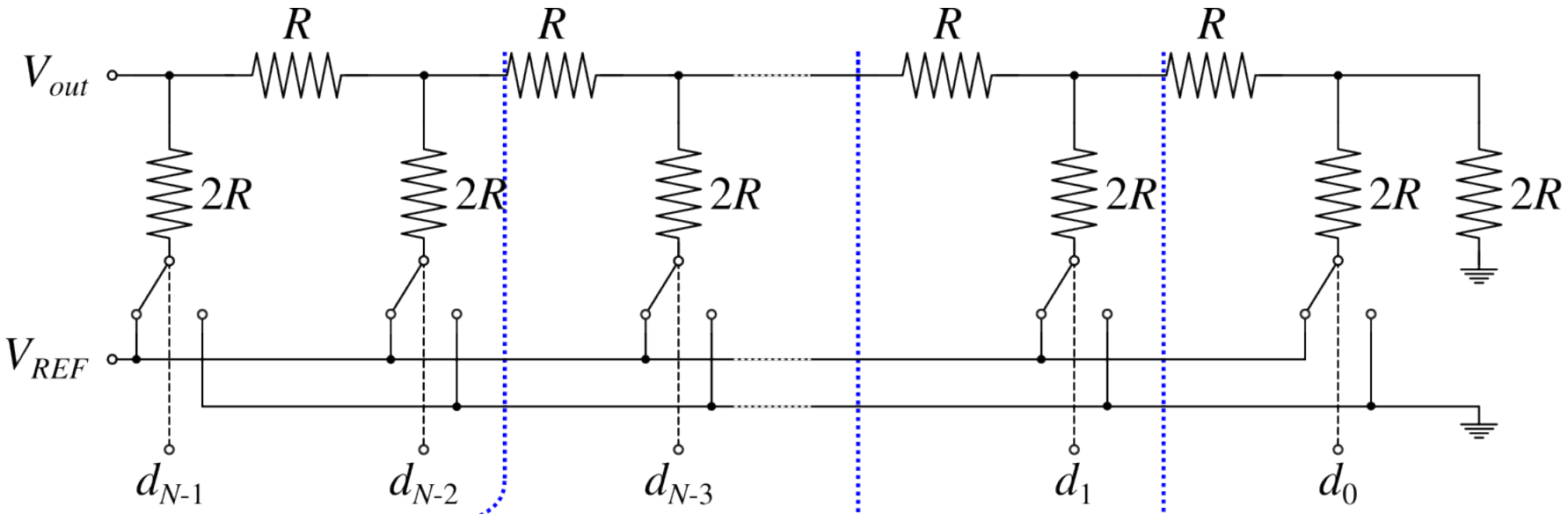
**Very expensive to reach resolution higher than 10 bit with R-string architecture: routing!**

# The R-2R ladder




- The simple R-2R ladder provides a **voltage and a current scaling by 2**
- One of the 2R can be substituted instantiating another ladder, this way by iteration a **binary scaling** is obtained: relaxed routing with respect to R-String
- If iterated N times, the total number of unit resistors will be **N×3+2** (the last 2 is due to the final termination 2R element)

# The R-2R DAC




$$\begin{cases} V_{TH} = \frac{1}{2}\left(d_{N-3} + \frac{d_{N-2}}{2} + \ldots + \frac{d_0}{2^{N-3}}\right) V_{REF} \\ R_{TH} = 2R \end{cases}$$

$$\begin{cases} V_{TH} = \frac{1}{2} d_0 \cdot V_{REF} \\ R_{TH} = 2R \end{cases}$$

$$\begin{cases} V_{TH} = \frac{1}{2}\left(d_1 V_{REF} + \frac{1}{2} d_0 \cdot V_{REF}\right) \\ R_{TH} = 2R \end{cases}$$

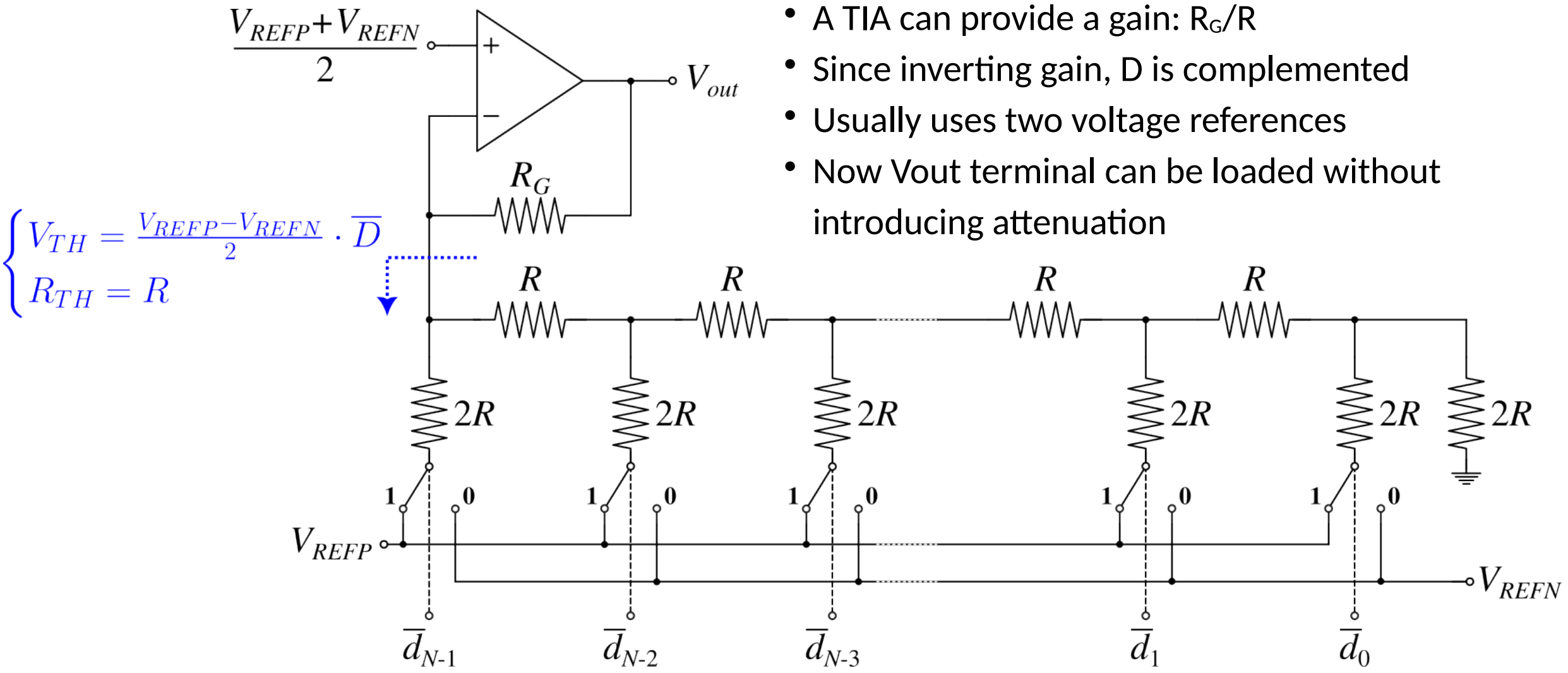
$$V_{out} = \left(\frac{d_{N-1}}{2} + \frac{d_{N-2}}{2^2} + \ldots + \frac{d_0}{2^N}\right) V_{REF} = \frac{V_{REF}}{2^N} \cdot D$$

$$\mathrm{LSB} = V_{REF}/2^N$$

**Operation:**

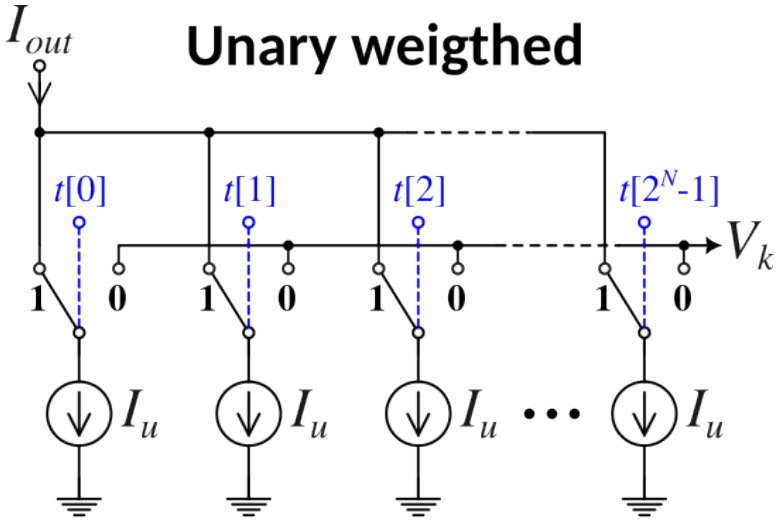
- Thevenin equivalent from LSB section, moving then towards MSB
- At each section a ½ scaling is introduced (nominally)
- As in R-string the scaling depends on resistive voltage dividers
- Output voltage is **binary weighted**
- **Current consumption**

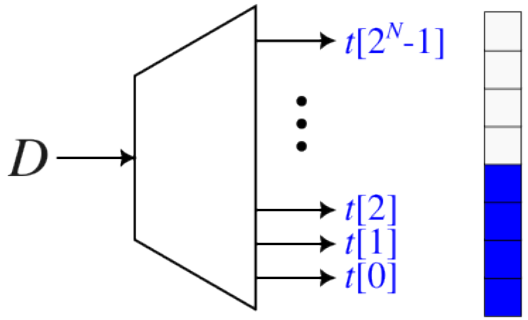
# The R-2R DAC, TIA configuration



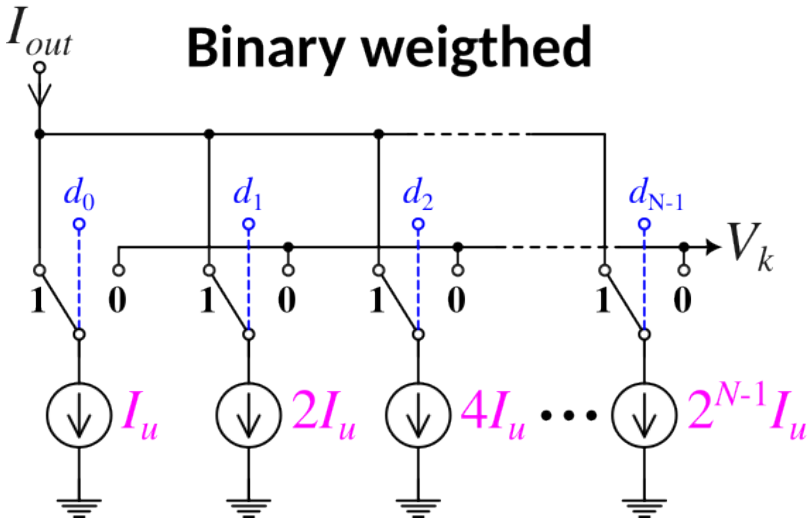

- A TIA can provide a gain: $R_G/R$
- Since inverting gain, D is complemented
- Usually uses two voltage references
- Now Vout terminal can be loaded without introducing attenuation

# CS DAC: Unary and Binary Weighted




Binary-to-thermometer decoder




$$D = \sum_{k=0}^{N-1} 2^k d_k$$

The switch steer the current to the output or to $V_k$ (dummy low-impedance node)

In both cases, $I_u$ sources are obtained through current mirrors:

$$I_u = \frac{I_{REF}}{2^N} \qquad \text{(LSB)}$$

$$I_{out} = \frac{I_{REF}}{2^N} \cdot D$$

Accuracy relies on our ability to ensure the correct weight of each current source (CS) element: a **mosfet**

# DAC Converters Design: Syllabus

**General definitions**

- Applications
- Basic principles and operating electrical domains
- Performance parameters

**Resistor-based DACs**

- R-string, R-string interpolated, and limits (DNL, INL analysis)
- R-2R ladder: binary weighting, DNL approximate analysis, configuration with a TIA.

**Current-source (CS) based DACs**

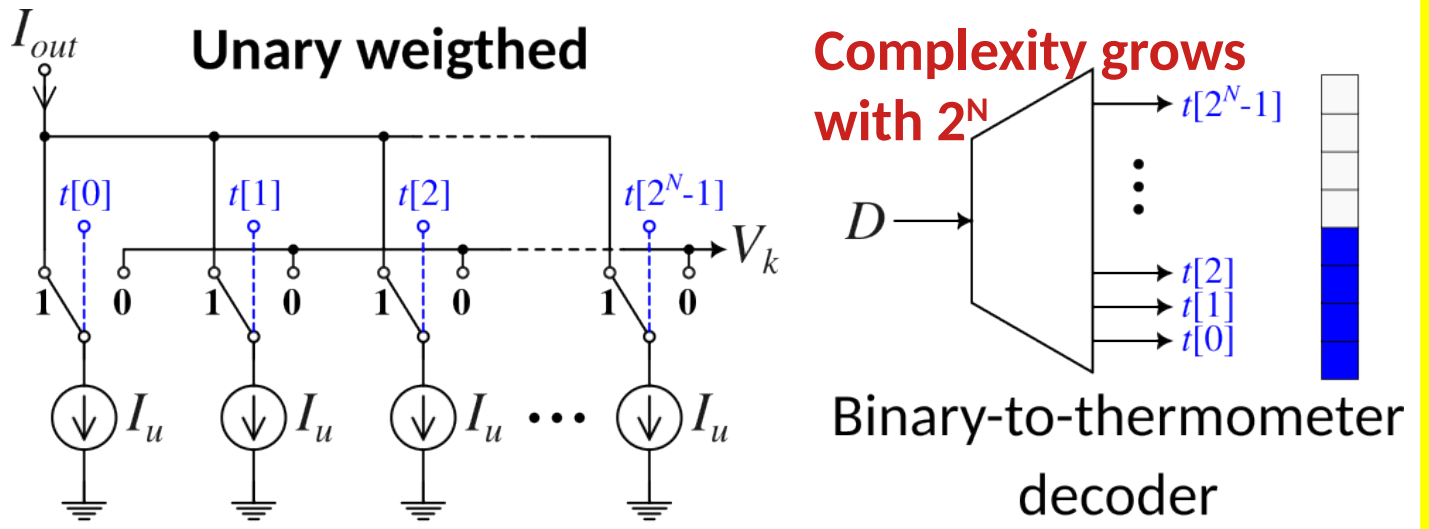
- Unary and binary weighted CSs. DNL/INL analysis for binary scales CSs. Segmentation.
- Design for high precision: gradients due to physical processing and due to routing, solutions based on ordered element switching and calibration techniques based on current copier

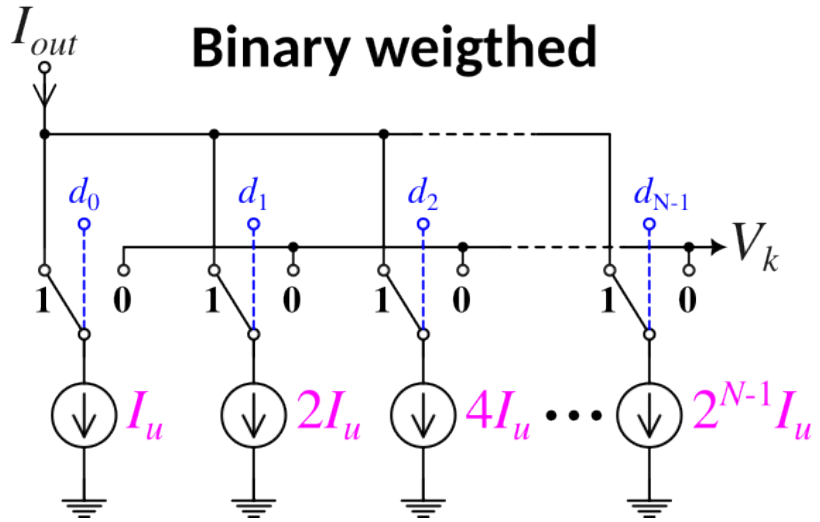
**Charge-scaling DACs**

- Charge redistribution based on parallel capacitor banks

INSIGHTS

# CS DAC: Unary and Binary Weighted




$$D = \sum_{k=0}^{N-1} 2^k d_k$$

**Unary weighted**

- ✓ Can be made monotonic-guaranteed by driving the $2^N$ mosfets of the array with thermometric coding: $D$ increase +1 : +1 more CS routed to $I_{out}$
- ✓ Glitches proportional to code transition (small dynamic distortion)
- ✘ It suffers from the same INL problem of the resistor string DAC:

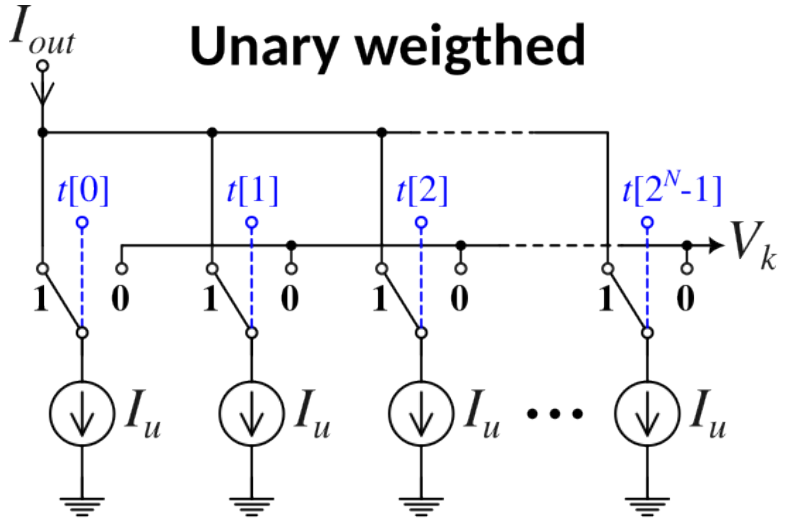
$$\sigma_{\mathrm{DNL}} = \sigma_{\Delta I/I}$$

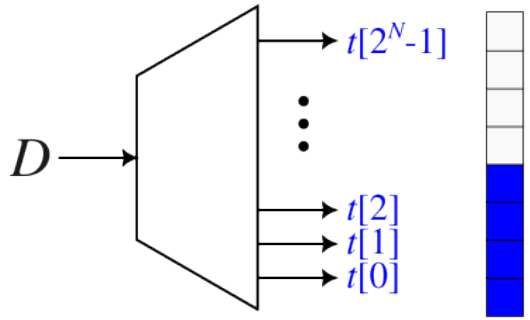
$$\sigma_{\mathrm{INL,max}} = 2^{N/2-1} \cdot \sigma_{\Delta I/I}$$

$$\frac{\Delta I}{I} = \frac{\Delta \beta}{\beta} + \frac{2\Delta V_T}{V_{GS} - V_T}$$

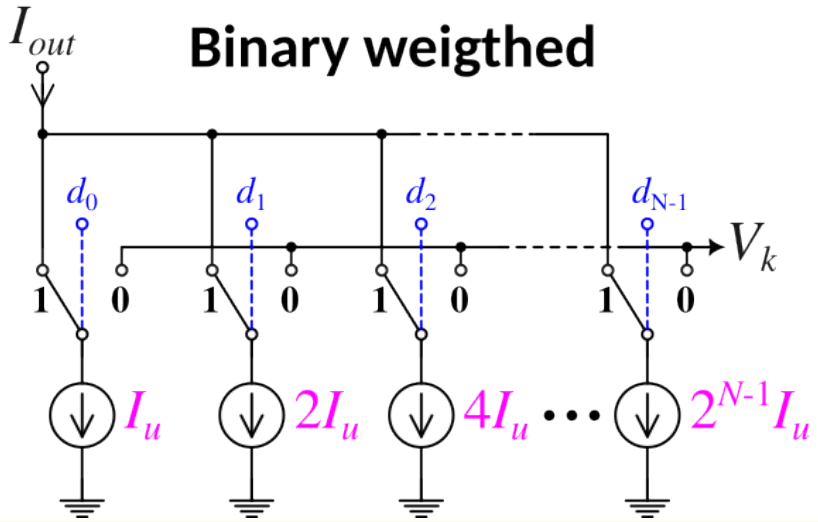
- Large Area
- Large overdrive

# CS DAC: Unary and Binary Weighted




Binary-to-thermometer decoder




$$D = \sum_{k=0}^{N-1} 2^k d_k$$

## Binary weighted

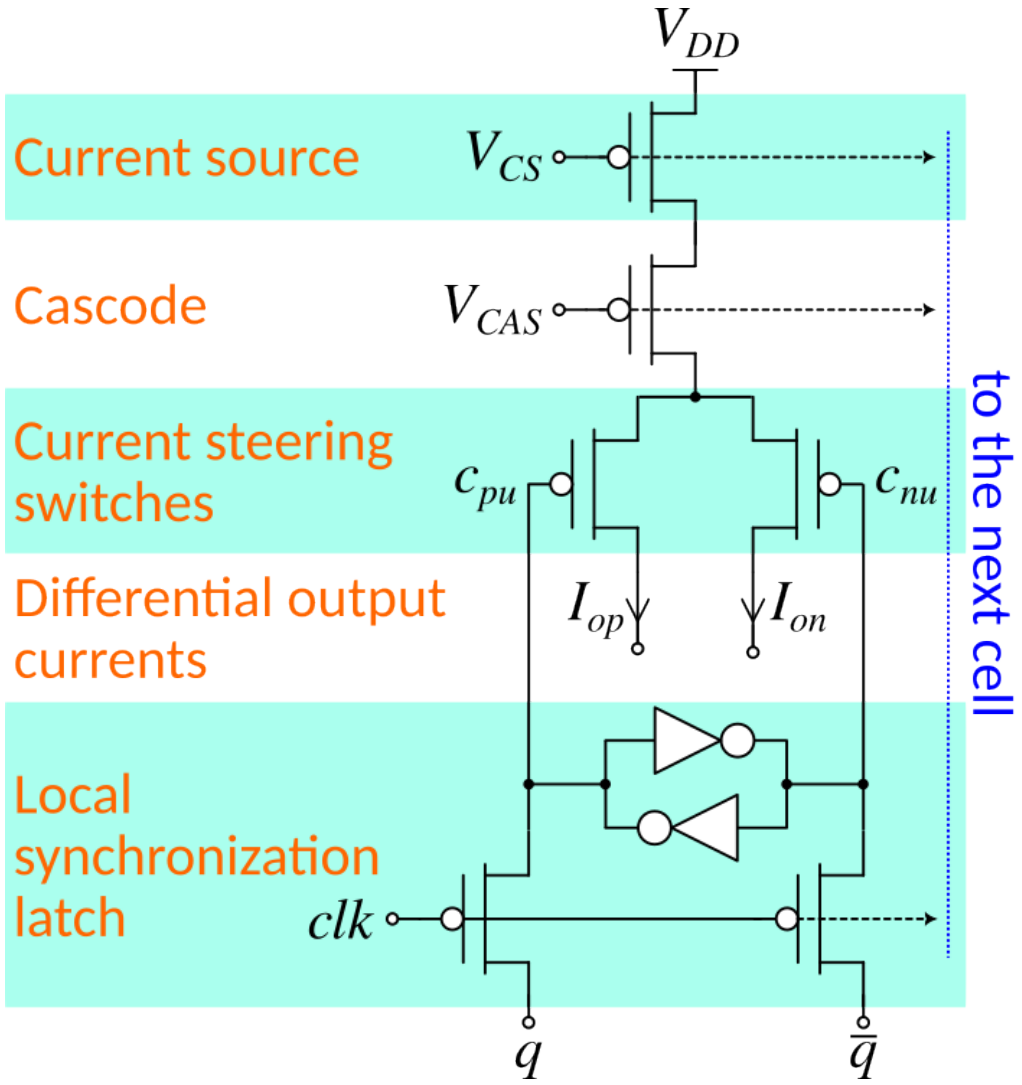
- ✓ Trivial digital decoding (small overhead)
- ✘ Same INL as unary weighted (for the same output level, the same number of nominally identical $I_u$ sources are selected)
- ✘ DNL: as in R-2R, the major code transition is the most critical: 011..→100..

$$\sigma_{\mathrm{DNL}} = \underbrace{(2^{N-1}-1)}_{011\ldots}\sigma_{\Delta I/I} + \underbrace{(2^{N-1})}_{100\ldots}\sigma_{\Delta I/I}$$

$$\sigma_{\mathrm{DNL}} \approx 2^N \sigma_{\Delta I/I}$$

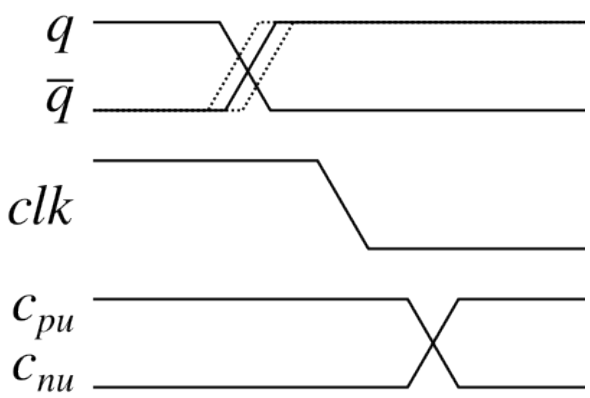
**Critical!**

# CS DAC: Cell Implementation



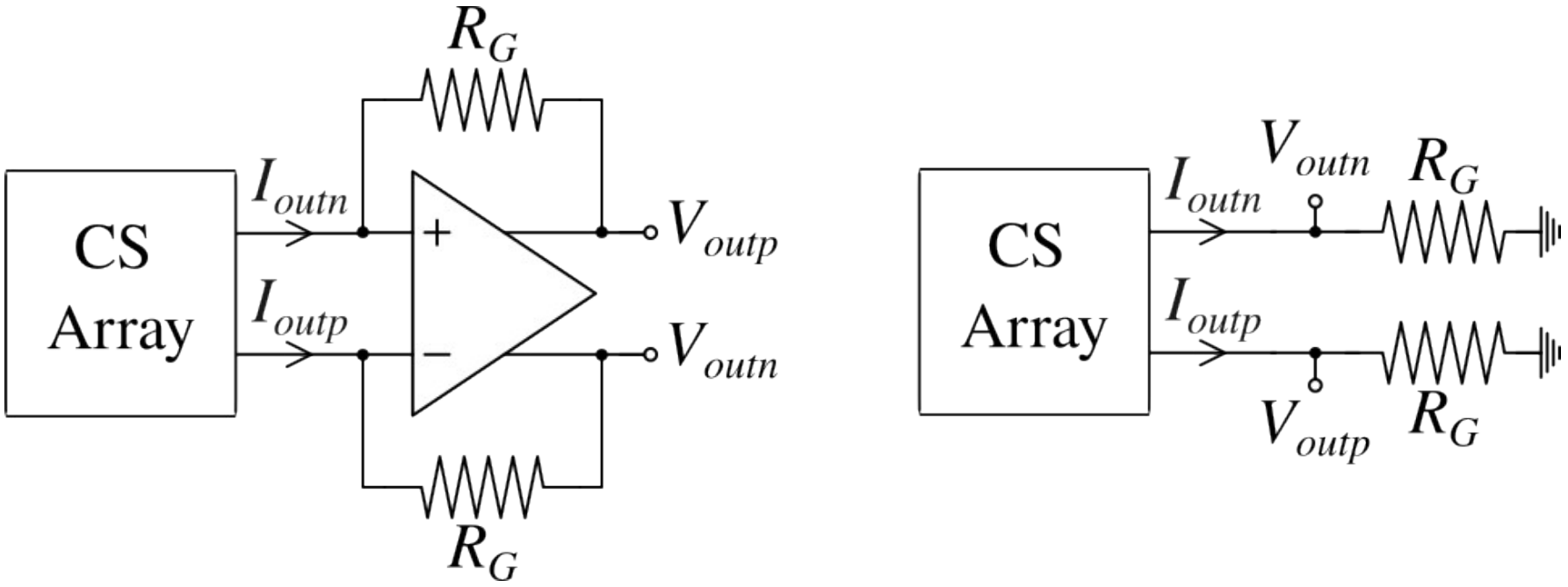

Usually:

- **Fully-differential implementation** (avoids waste of current)
- **PMOS** CS for milder flicker (under area constraint)
- Local synchronization **latch** gated by global clock (very important for high-speed designs): avoids differential glitches at the output currents




Generated in a different part of the circuit, and routed eventually with different paths lengths and strays

# CS DAC: Implementations




If a voltage output is desired, the CS array output currents can be injected to the virtual ground of a TIA or terminated to a resistive load:

**TIA solution**: the virtual ground ensures , however distortion, noise of TIA is now of concern. High-power demanding choice, not compatible with high-speed requirements.

**R-termination solution**: output nodes undergo voltage swings, to avoid distortion, cascode (also double) are mandatory. Not compatible with low-voltage requirements.

# DAC Converters Design: Syllabus

**General definitions**

- Applications
- Basic principles and operating electrical domains
- Performance parameters

**Resistor-based DACs**

- R-string, R-string interpolated, and limits (DNL, INL analysis)
- R-2R ladder: binary weighting, DNL approximate analysis, configuration with a TIA.

**Current-source (CS) based DACs**

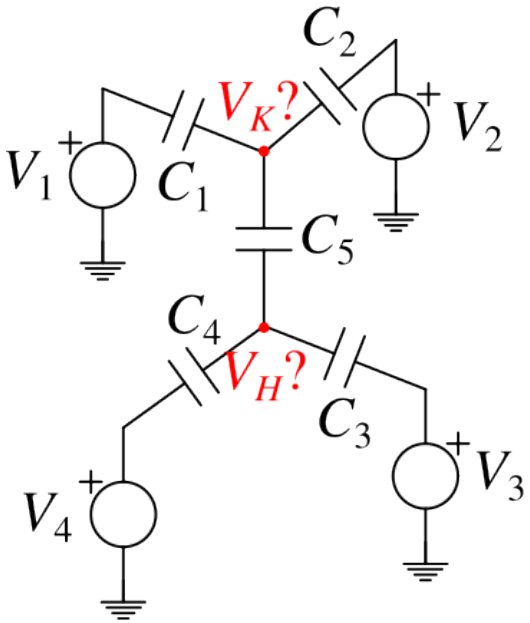
- Unary and binary weighted CSs. DNL/INL analysis for binary scales CSs. Segmentation.
- Design for high precision: gradients due to physical processing and due to routing, solutions based on ordered element switching and calibration techniques based on current copier

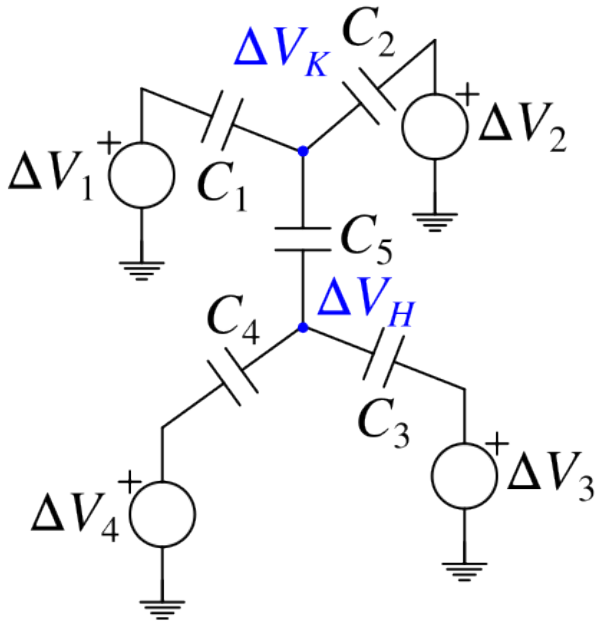
**Charge-scaling DACs**

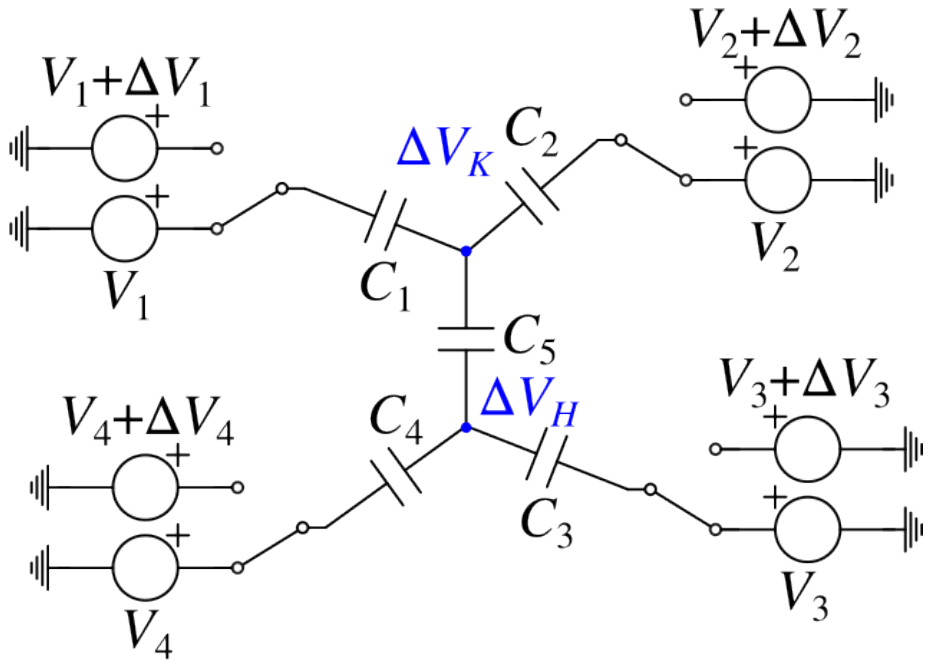
- Charge redistribution based on parallel capacitor banks

INSIGHTS

# Charge redistribution: generic capacitors-only network



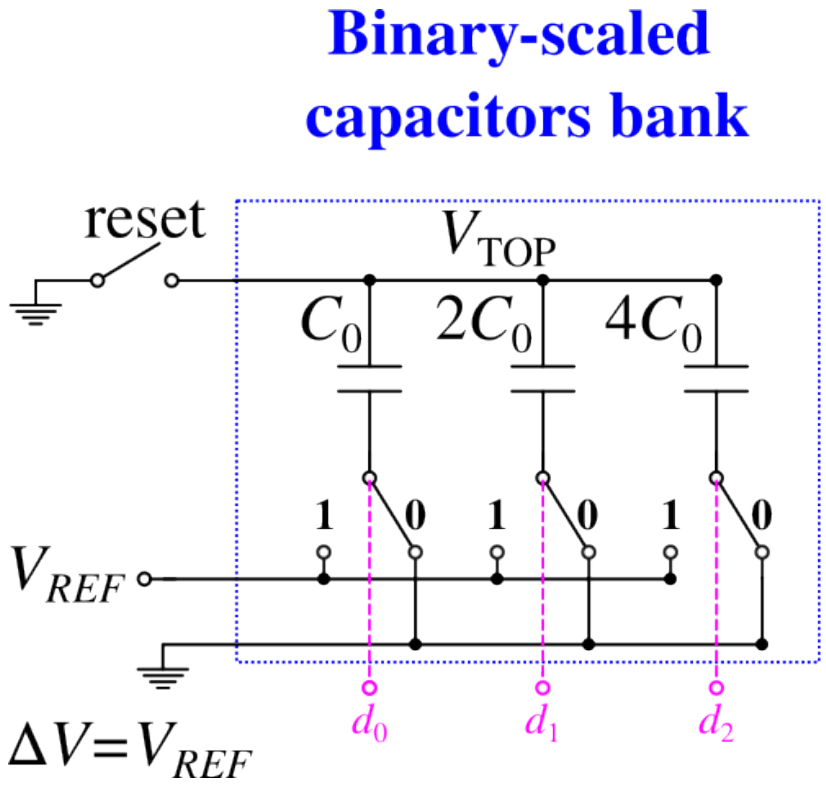




Equivalent Ohm's law for charge transfer

$$\Delta V = \frac{1}{C} \cdot \Delta Q$$

# Charge redistribution: binary-scaled capacitor bank




$$d_0(0 \to 1): \Delta V_{\mathrm{TOP}} = \frac{\frac{1}{C_0}}{\frac{1}{C_0}+\frac{1}{2C_0}+\frac{1}{4C_0}}\Delta V = \frac{4}{7}\Delta V$$

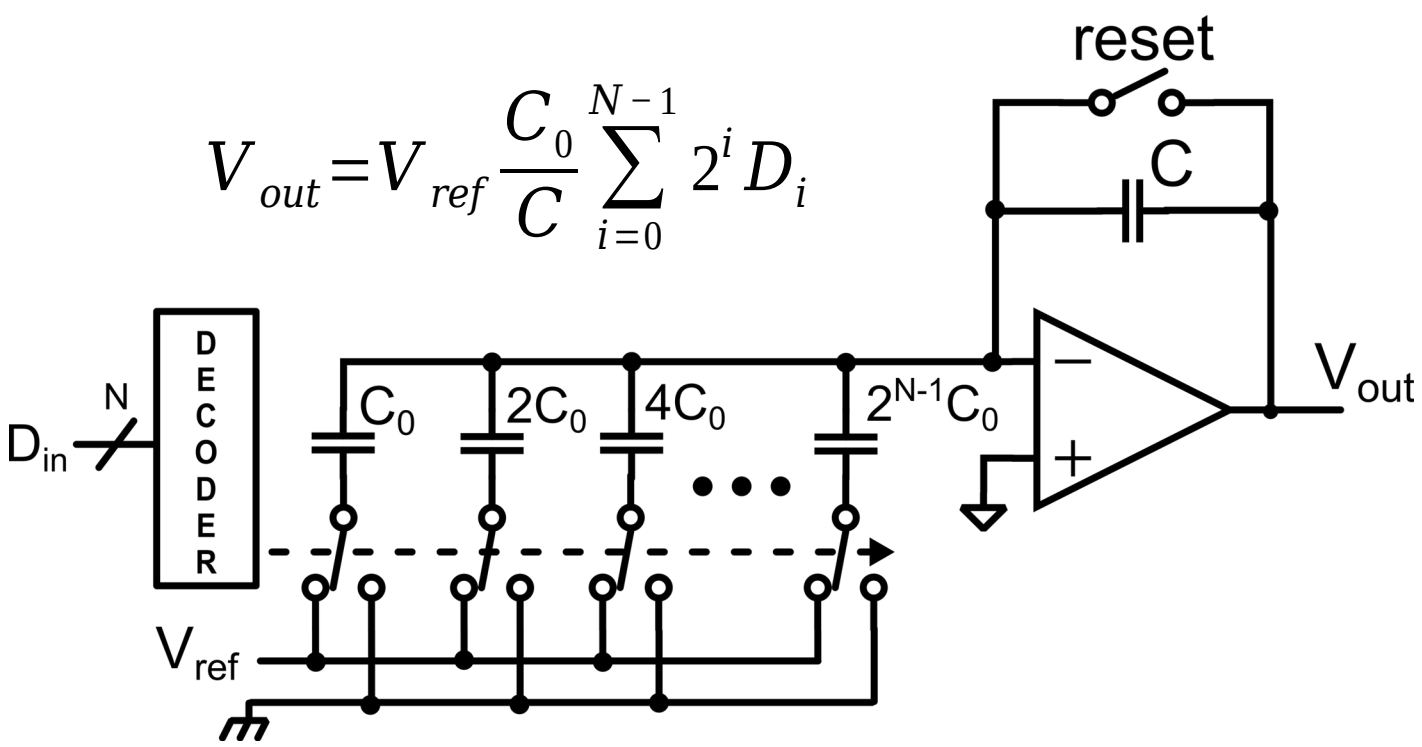
$$d_1(0 \to 1): \Delta V_{\mathrm{TOP}} = \frac{\frac{1}{2C_0}}{\frac{1}{C_0}+\frac{1}{2C_0}+\frac{1}{4C_0}}\Delta V = \frac{2}{7}\Delta V$$

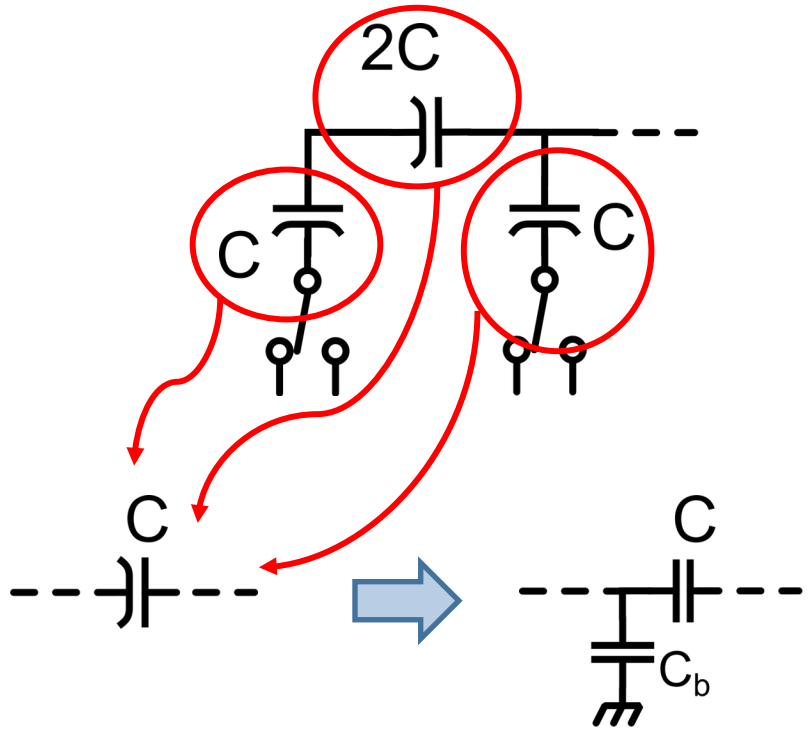
$$d_2(0 \to 1): \Delta V_{\mathrm{TOP}} = \frac{\frac{1}{4C_0}}{\frac{1}{C_0}+\frac{1}{2C_0}+\frac{1}{4C_0}}\Delta V = \frac{1}{7}\Delta V$$

# Charge redistribution DAC

- ✓ Capacitor bank has better matching properties than resistor string
- ✓ Low-power solution (no static consumption in the capacitor bank)
- ✘ Output not always valid
- ✘ Troubles with C-2C solutions due to parasitic capacitance of the bottom plate

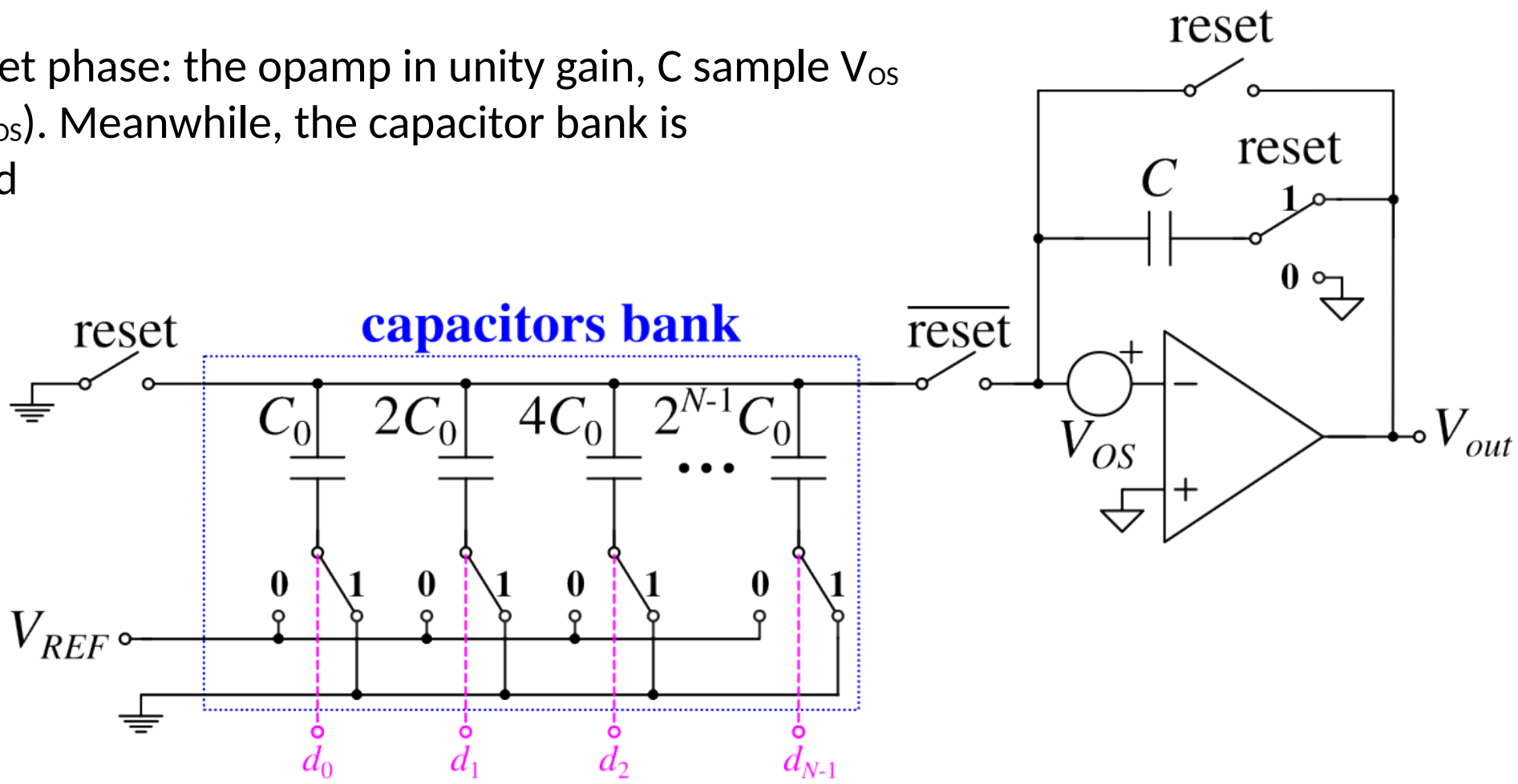
$$V_{out} = V_{ref} \frac{C_0}{C} \sum_{i=0}^{N-1} 2^i D_i$$



# Charge redistribution DAC with CDS

During reset phase: the opamp in unity gain, C sample $V_{OS}$ (Vout = $-V_{OS}$). Meanwhile, the capacitor bank is precharged



# DAC architectures overview

| | *Pros* | *Cons* |
|---|---|---|
| **Resistor String** | Guaranteed monotonic.<br>Low power consumption | Very large number of resistors |
| **R-2R ladder** | Reduced number of resistor of similar value | Not inherently monotonic.<br>Suffers from the on-resistance of the switches. Large glitches |
| **Current Steering** | Very fast. No need for resistors. Can be designed to be always monotonic | Current output.<br>Flicker Noise |
| **Switched Capacitors** | Optimal power *vs* speed trade-off | The output may be not available in the whole clock cycle.<br>Large glitches |

# DAC Converters Design: **Insights**

**General definitions**

- Applications
- Basic principles and operating electrical domains
- Performance parameters

**Resistor-based DACs**

- R-string, R-string interpolated, and limits (DNL, INL analysis)
- R-2R ladder: binary weighting, DNL approximate analysis, configuration with a TIA.

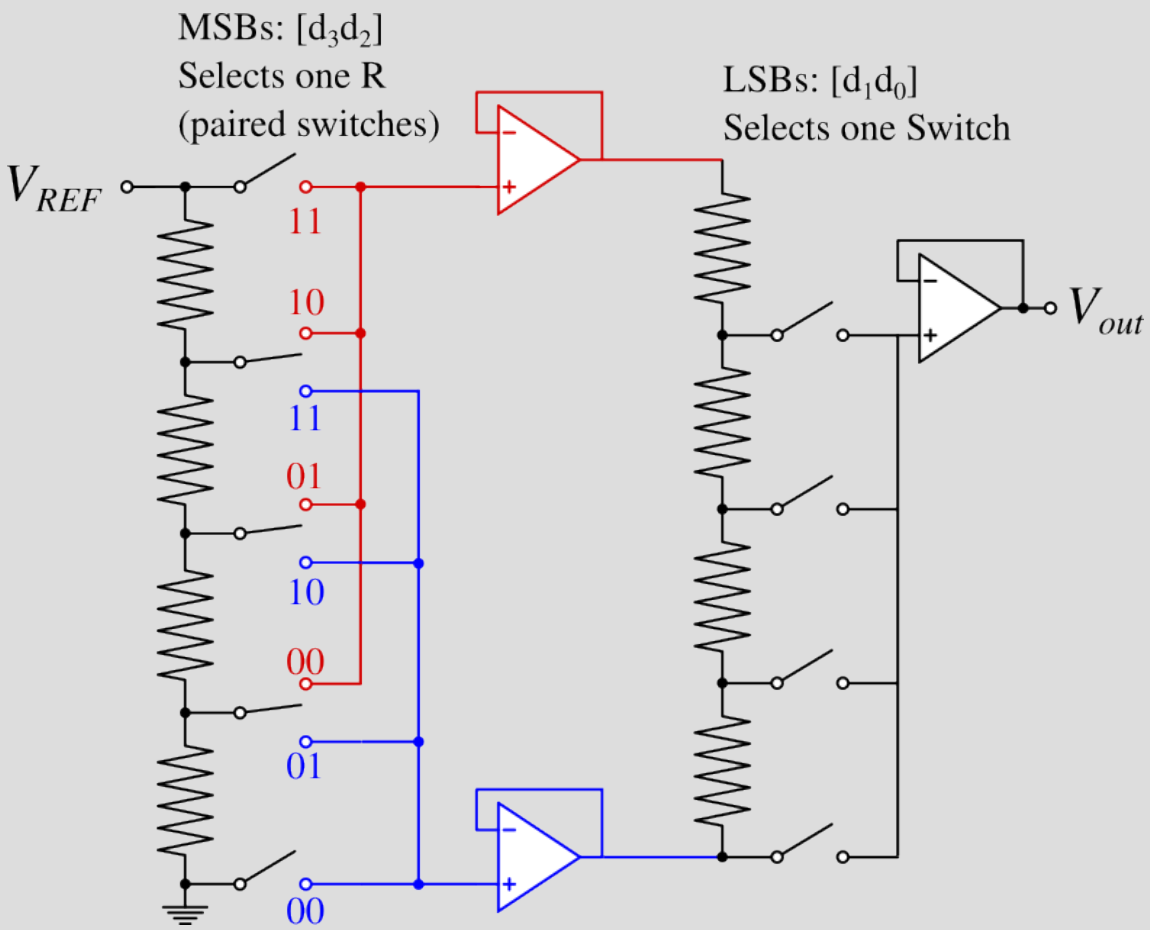
**Current-source (CS) based DACs**

- Unary and binary weighted CSs. DNL/INL analysis for binary scales CSs. Segmentation.
- Design for high precision: gradients due to physical processing and due to routing, solutions based on ordered element switching and calibration techniques based on current copier

**Charge-scaling DACs**

- Charge redistribution based on parallel capacitor banks

# R-String DAC (interpolated)

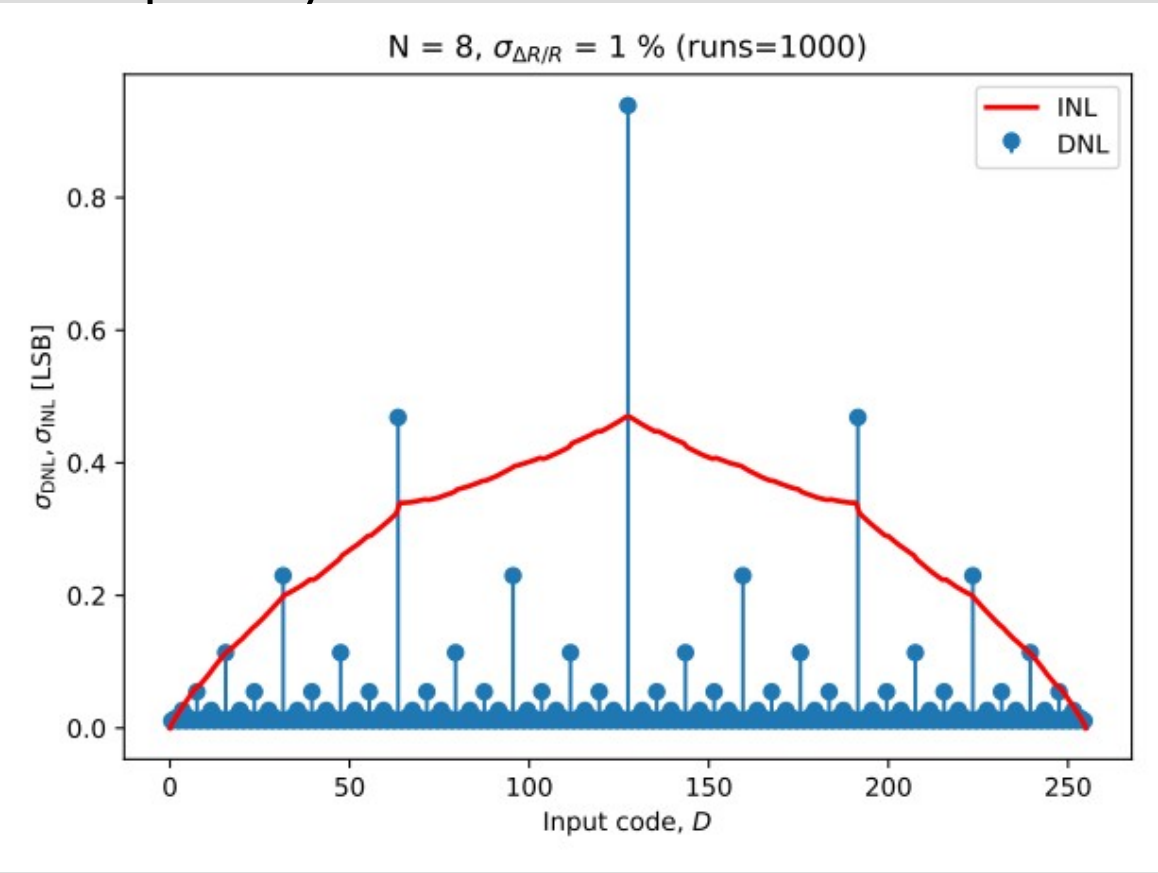



**Interpolated R-string DAC: coarse + fine segments**

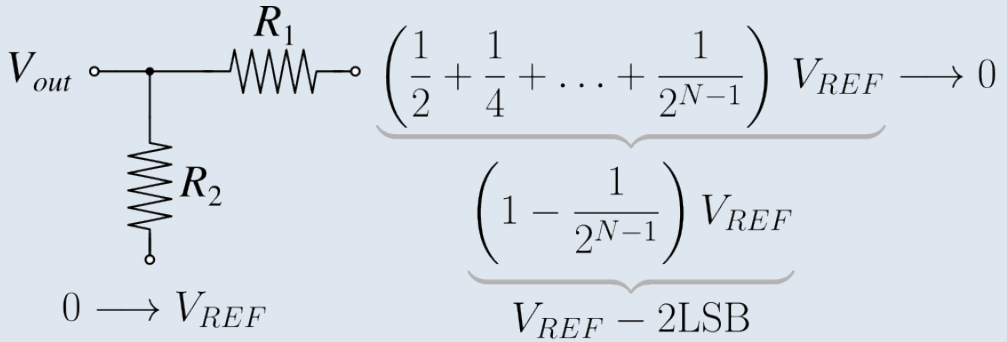
- The input D of N bit is divided in to an MSB segment of length M and an LSB segment of length L:
- N = M+L
- Each R in the MSB string is selected at time through the MSB segment of D
- The LSB segment is decoded to select one switch in the LSB segment string as before
- Now the number of levels are still $2^N$, but the number of Rs is now $2^M+2^L$
- Monotonicity not guaranteed (due to opamp offsets). More advanced selection techniques are employed.

# The R-2R DAC: DNL and INL

- DNL and INL analysis is complex: usually the designer relies on Monte Carlo simulations
- The most relevant effect is on DNL, in correspondence with **major code transition**: 011… → 100…
- Empirically $\sigma_{INL} \approx \sigma_{DNL}/2$.




Very approximate analysis:



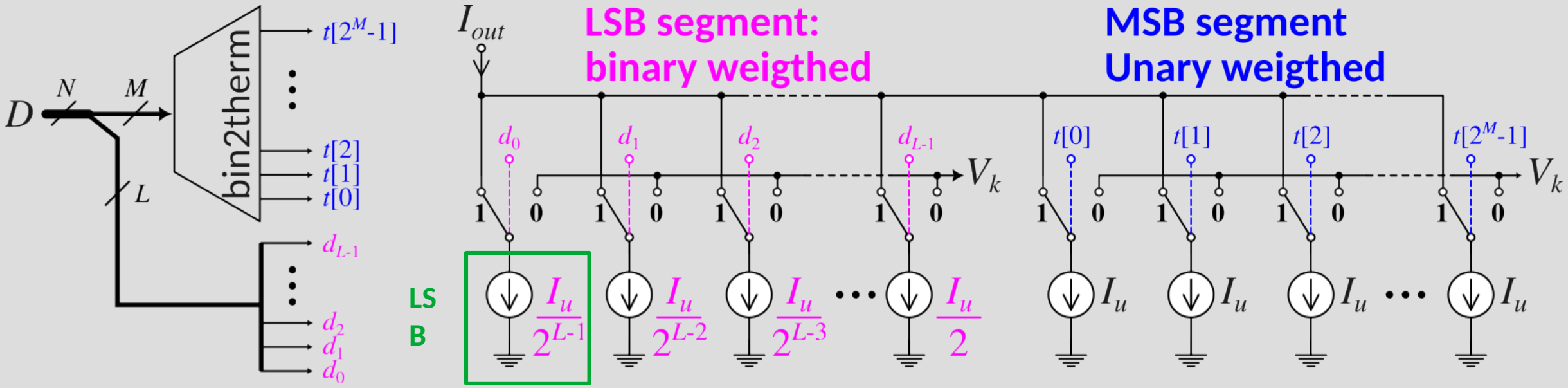

$$\Delta V_{out} = \frac{R_1}{R_1+R_2}V_{REF} - \frac{R_2}{R_1+R_2}(V_{REF} - 2\text{LSB})$$

$$\approx \frac{\Delta R}{2R}V_{REF} + \text{LSB}$$

monotonic if: $\frac{\Delta R}{R} < \frac{2\text{LSB}}{V_{REF}} = \frac{1}{2^{N-1}}$

**In resume**: both INL and DNL limited by N and $\sigma_{\Delta R/R}$

# CS DAC: Segmentation




**Objective:** strike a balance between complexity and accuracy

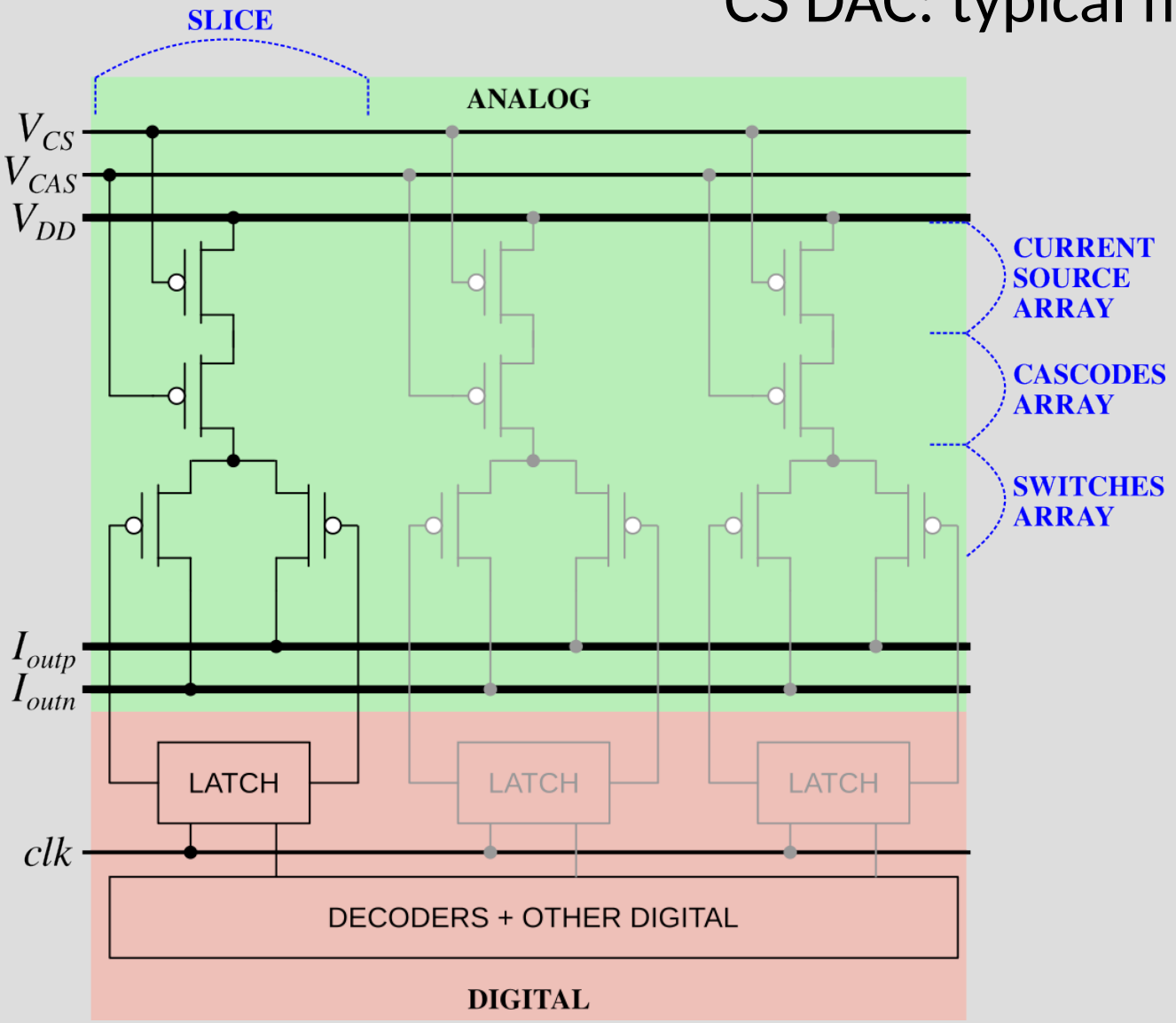
- *N* bits are segmented in *M* bits in the MSB segment and *L* bits in the LSB segment $N = M + L$
- INL is unaffected $\sigma_{\mathrm{INL,max}} = 2^{N/2-1} \cdot \sigma_{\Delta I/I}$
- DNL is dominated by the LSB segment $\sigma_{\mathrm{DNL}} = 2^{L}\sigma_{\Delta I/I}$
- The total number of switched elements depends on segmentation ratio $2^{M} - 1 + L$

# CS DAC: Segmentation

$$N = \underbrace{M}_{\text{MSB}} + \underbrace{L}_{\text{LSB}} \qquad \sigma_{\text{INL,max}} = 2^{N/2-1} \cdot \sigma_{\Delta I/I} \qquad \sigma_{\text{DNL}} = 2^{L}\sigma_{\Delta I/I} \qquad 2^{M} - 1 + L$$

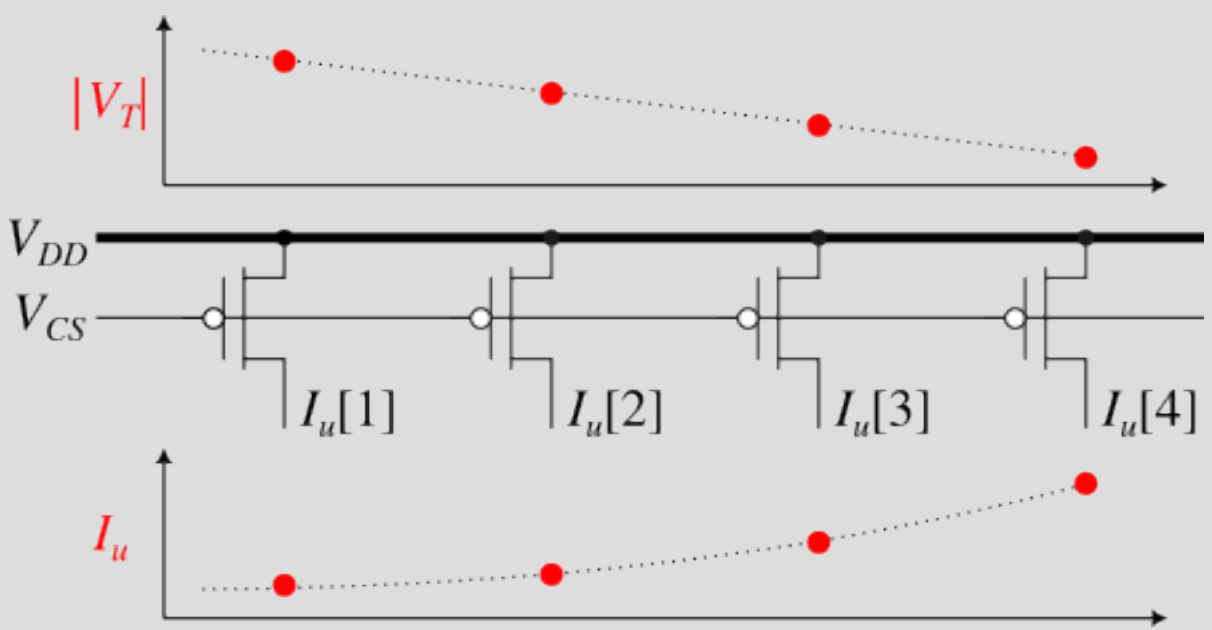
| Architecture | $\sigma_{INL,max}$ [$\sigma_{\Delta I/I}$] | $\sigma_{DNL}$ [$\sigma_{\Delta I/I}$] | Number of switched elements |
|---|---|---|---|
| Unary, N = 10 | 16 | 1 | 1023 |
| Segmented 6+4 | 16 | 16 | 67 |
| Segmented 5+5 | 16 | 32 | 36 |
| Segmented 4+6 | 16 | 64 | 21 |
| Binary, N = 10 | 16 | 1024 | 10 |

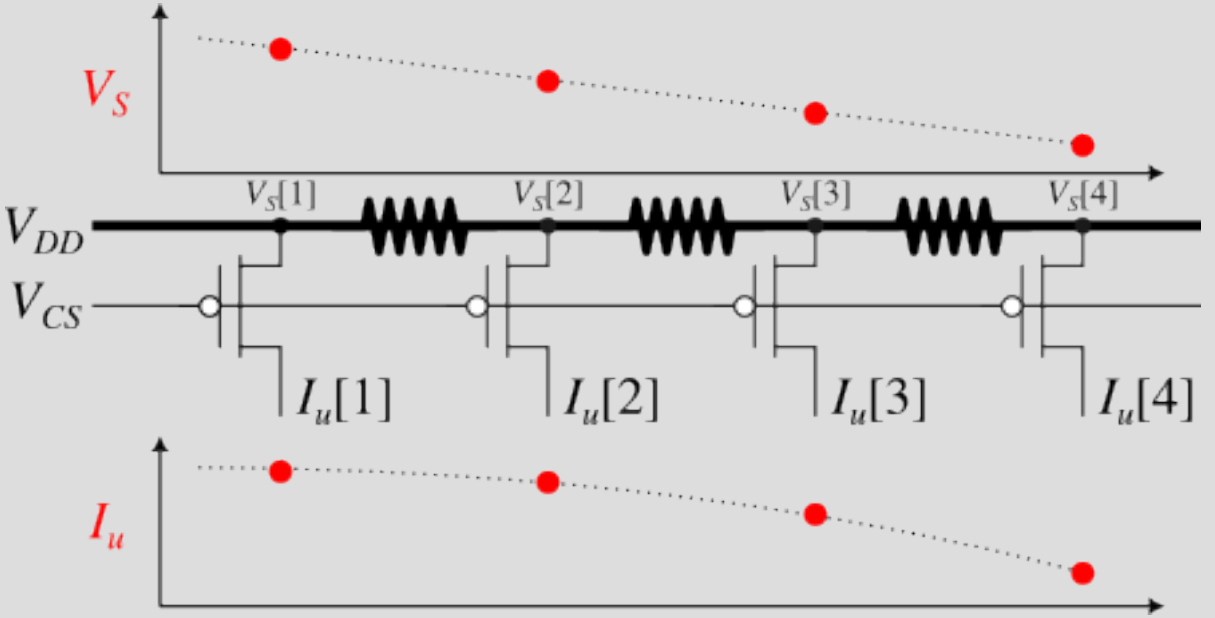
# CS DAC: typical floorplan




- Usually, **linear floorplan**: homologous devices (having the same functions) are put on the linear arrays for easier connection and compactness: **as the number of elements grow, gradients are important**
- The digital is in the lower part, separated from the quiet analog part
- Controlling signals cross the differential outputs (common-mode couplings)
- The clk line can be distributed in a tree-like structure to compensate for delays

# CS DAC: gradients in large arrays



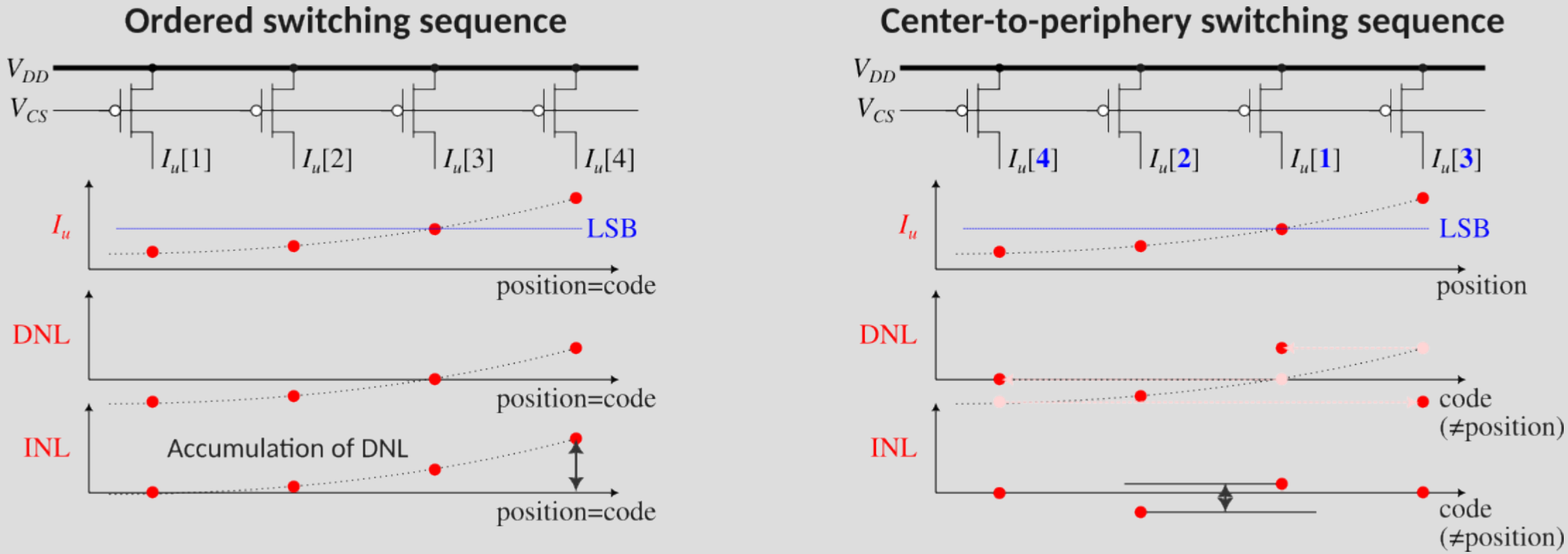




Nominally identical devices, under same bias conditions gives different currents due to:

- Manufacturing imperfections (e.g. threshold parameter spreading)
- Interconnection ohmic drops
- Both results in current gradients
- The switching order is decided by the digital logic

ERRORS DUE TO GRADIENTS ARE CORRELATED WITH THE DEVICE POSITION WITHIN THE ARRAY

# CS DAC: switching sequence to brake the gradients



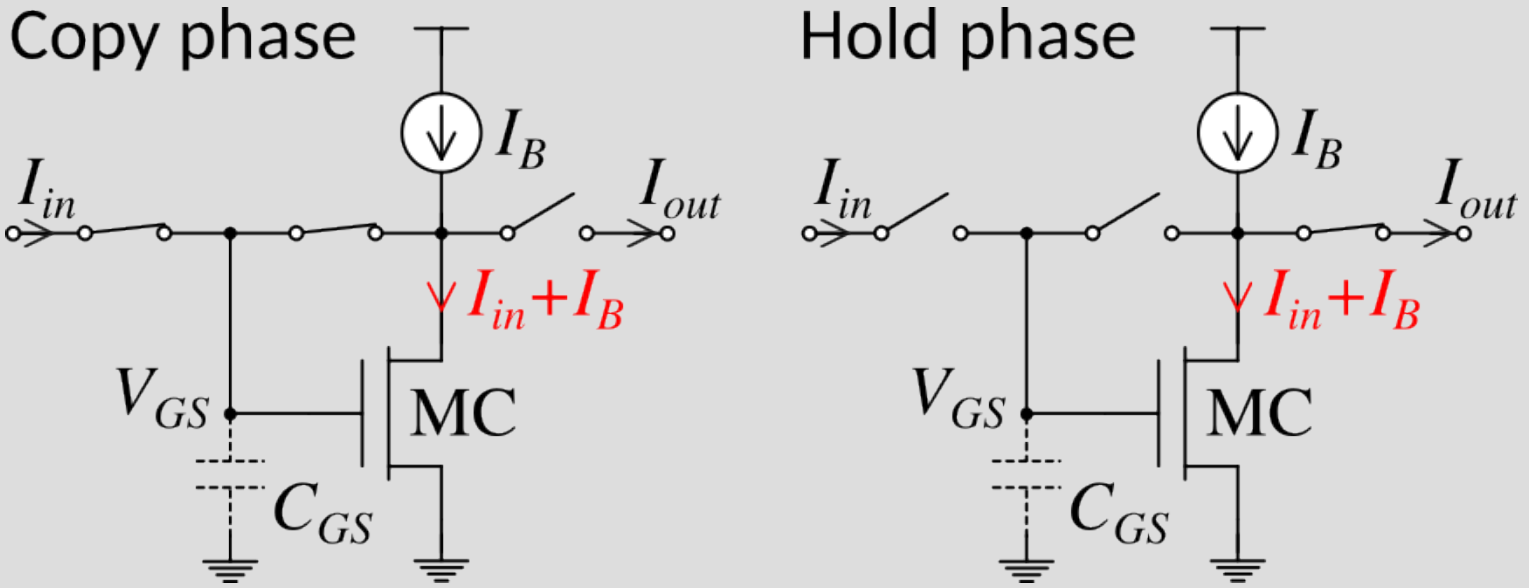

If we activate the CS in an orderly fashion: the gradient errors will accumulate → large INL

If the activation sequence starts from the center devices and hops to the outer ones, DNL is almost the same (statistically), but INL error gets tighter

Resembles the **common centroid** layout technique

# CS DAC: current copier

Static non linearity (INL/DNL) is main mechanism of distortion especially for medium-to-low bandwidth CS DACs. Intrinsic matching, implies (a) large cell area and (b) large $V_{GS}$ voltages.
(a) In conflict with the need for small strays in high-speed applications;
(b) In conflict with low-voltage operation
Alternatively, mismatch is allowed to happen relaxing the other constraints, and it is corrected after fabrication with **dynamic calibration**




**Current copier operation:**

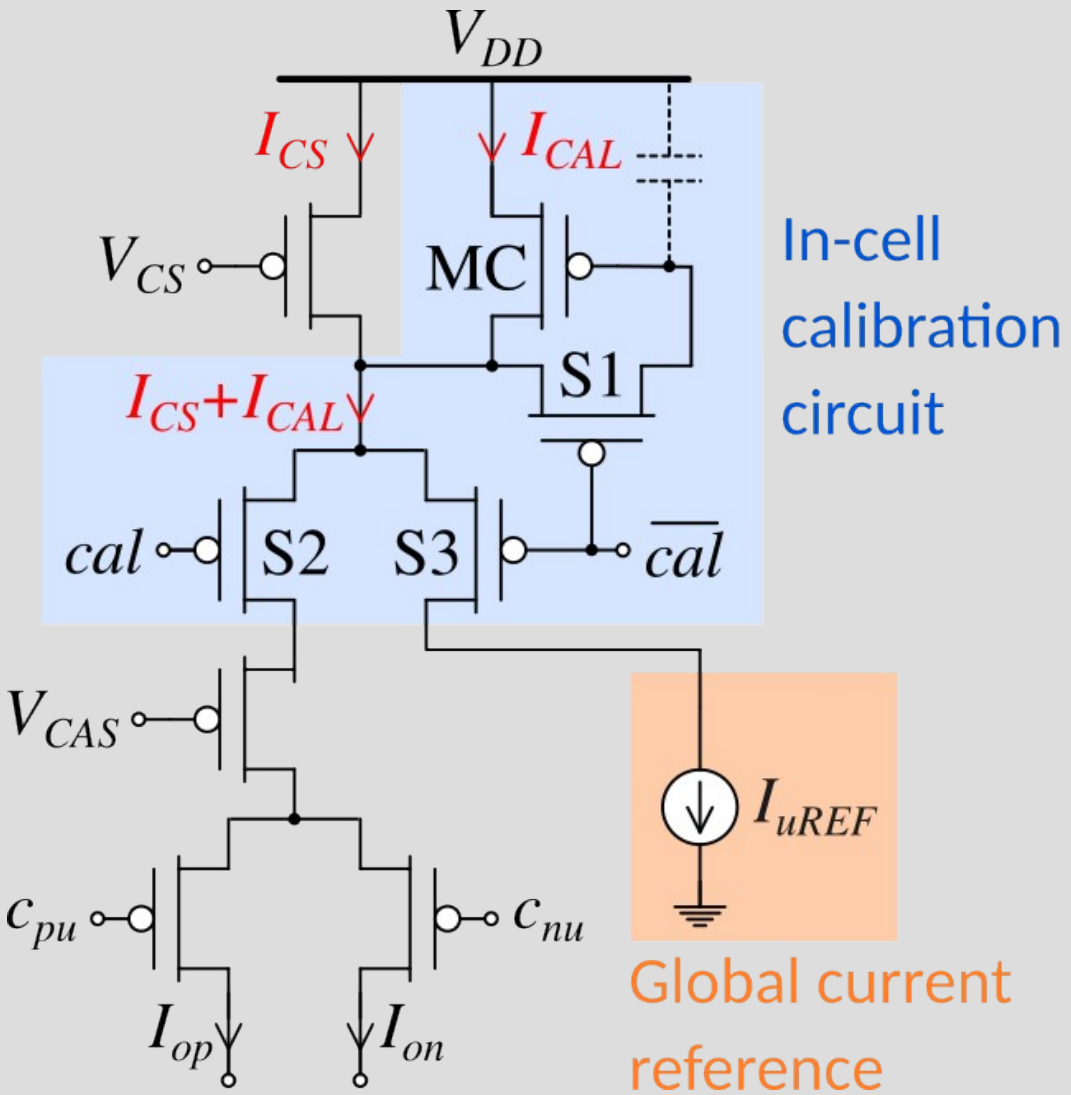
Copy phase: MC is diode connected, $V_{GS}$ develops into the stray capacitance $C_{GS}$ for a total current in MC of $I_{in}+I_B$

Hold phase: the $V_{GS}$ is hold by the stray CGS: $I_{out} = I_{in}$

A current copier behaves like a virtually **mismatch-free current mirror**: same device for input and output current

Beware charge injections and leakage

# CS DAC: calibration technique based on current copier




More switches (S1-S3) are added to the stack + a periodic calibration phase (*cal*) for each cell

MC adds a bit more current. The total cell current is $I_{CS}+I_{CAL}$ , typically MC supplies 10-20% of the nominal current

During calibration: $I_{CS}+I_{CAL} = I_{uREF}$

During normal operation: $I_{CAL}$ is kept thanks to the "analog" memory

All the unary CS are calibrated once at time in a cyclic fashion (refresh and track for changes due to temp)

The calibration runs in the **background** while the rest of the DAC operates as usual. While calibrating the element is not available for normal operation, hence we need at least one redundant element in the array