EE 505

Lecture 19

Architectural Performance Comparisons ADC Design

Dynamic Current Source Matching



- Correct charge is stored on C to make all currents equal to I_{REF}
- Does not require matching of transistors or capacitors
- · Requires refreshing to keep charge on C
- · Form of self-calibration
- Calibrates current sources one at a time
- Current source unavailable for use while calibrating
- Can be directly used in DACs (thermometer or binary coded)
- Still use steering rather than switching in DAC

Often termed "Current Copier" or "Current Replication" circuit

Review from Last Lecture Noise in DACs

Resistors and transistors contribute device noise but what about charge redistribution DACs ?





Example: First-Order RC Network



Example: First-Order RC Network



From a standard change of variable with a trig identity, it follows that

$$\mathcal{P}_{n_{RMS}} = \sqrt{\int_{f=0}^{\infty} S_{VOUT}} df = \sqrt{\frac{kT}{C}}$$

- Note the continuous-time noise voltage has an RMS value that is independent of R
- The noise contributed by the resistor is dependent only upon the capacitor value C
- This is often referred to at kT/C noise and it can be decreased at a given T only by increasing C



"kT/C" Noise at T=300K



Example: Switched Capacitor Sampler



Example: Switched Capacitor Sampler



 $v_{\rm n}({
m kT})$ is a discrete-time sequence obtained by sampling a continuous-time noise wave

Characterization of a noise sequence



Theorem If v(t) is a continuous-time zero-mean noise source and $\langle v(kT) \rangle$ is a sampled version of v(t) sampled at times T, 2T, then the RMS value of the continuous-time waveform is the same as that of the sampled version of the waveform. This can be expressed as $v_{\rm \tiny RMS} = \hat{v}_{\rm \tiny RMS}$

Theorem If v(t) is a continuous-time zero-mean noise signal and $\langle v(kT) \rangle$ is a sampled version of v(t) sampled at times T, 2T, then the standard deviation of the random variable v(kT), denoted as σ_v

satisfies the expression
$$\sigma_{\rm v}$$
 = $v_{\rm RMS}$ = $v_{\rm RMS}$

Example: Switched Capacitor Sampler



kT C V n_{RMS}

Architectural Performance Characterizations

For the same total resistor area and the same resolution, how do these structures compare from a statistical characterization viewpoint?





For the same total area and the same resolution, how do these structures compare from a statistical characterization viewpoint?

Simulation environment:

Resolution = 10 $A_{\rho R} = 0.02 \mu m$ Rnom = 1000 Area Unit Resistor = $2 \mu m^2$ Resistor Sigma= 14.1421 INLtarget = 0.5000 LSB

Yield: Must meet INL target

INL_k for four random implementations



String DAC









INL histogram for 100,000 random implementations



Relative Statistical Characterization of R-based DACs INL_{kMAX} histogram for 100,000 random implementations



DNL histogram for 100,000 random implementations





String DAC

Summary

Resolution = 10 $A_{0R} = 0.02 \mu m$ $R_{nom} = 1000$ Area Unit Resistor = $2\mu m^2$ Resistor Sigma= 14.1421 $INL_{mean} = 0.385 LSB$ $INL_{sigma} = 0.118 LSB$ $DNL_{mean} = 0.049 LSB$ $DNL_{sigma} = 0.0047 LSB$ Yield (%) = 84.0

Relative Statistical Characterization of R-based DACs INL_k for four random implementations









Binary Weighted DAC

Relative Statistical Characterization of R-based DACs INL_{kMAX} histogram for 100,000 random implementations



Binary Weighted DAC



Binary Weighted DAC



Binary Weighted DAC

Summary

Resolution = 10 $A_{oR} = 0.02 \mu m$ $R_{nom} = 1000$ Area unit resistor= $2\mu m^2$ Resistor Sigma= 14.1421 $INL_{mean} = 0.367LSB$ $INL_{sigma} = 0.128 LSB$ $INL_{kmax_mean} = 0.00013 LSB$ $INL_{kmax_sigma} = 0.226 LSB$ $DNL_{mean} = 0.470 LSB$ $DNL_{sigma} = 0.228 LSB$ $INL_{target} = 0.500 LSB$ Yield (%) = 84.9



Relative Statistical Characterization of R-based DACs INL_k for four random standard series implementations











INL histogram for 15,000 random implementations Standard Series



R-2R DAC

DNL histogram for 15,000 random implementations Standard Series



Summary Standard Series



R-2R DAC

Resolution=10 A_{ρR}=0.02 μm Rnom = 1000Base Res Area(um²)=2 Res Sigma=14.1421 $INL_{mean} = 0.609 LSB$ $INL_{sigma} = 0.295 LSB$ $DNL_{mean} = 1.021 LSB$ $DNL_{sigma} = 0.610 LSB$ $INL_{kmax_mean} = 0.00017 LSB$ $INL_{kmax sigm}a = 0.566 LSB$ Yield INL Bound=0.5 LSB Yield= 41.4%

Relative Statistical Characterization of R-based DACs INL_k for four random standard parallel implementations











INL histogram for 100,000 random implementations Standard Parallel

3

3.5



R-2R DAC

DNL histogram for 100,000 random implementations Standard Parallel



R-2R DAC



R-2R DAC

Summary Standard Parallel Resolution = 10 $A_{0R} = 0.02 \mu m$ $R_{nom} = 1000$ Base Resistor Area $(um^2) = 2$ Resistor Sigma= 14.1421 $INL_{mean} = 0.737 LSB$ $INL_{sigma} = 0.357 LSB$ $INL_{kmax mean} = 0.0045 LSB$ $INL_{kmax_sigma} = 0.680 LSB$ $DNL_{mean} = 1.225 LSB$ $DNL_{sigma} = 0.732 LSB$ $INL_{target} = 0.5 LS$ Yield =28.5%

Why is the Standard Series yield significantly larger than the Standard Parallel?



Each bit slice has the same area

MSB resistor has higher percentage of area in Standard Series



R-2R DAC

DNL histogram for 15,000 random implementations Standard Series Area Scaled







R-2R DAC

INL histogram for 15,000 random implementations Standard Series Area Scaled



Scaling Factor: 1.7



 $\begin{aligned} &\text{Resolution} = 10 \\ &\text{A}_{\rho R} = 0.02 \mu m \\ &\text{R}_{nom} = 1000 \\ &\text{Total Area } 2048 \ \mu m^2 \\ &\text{Resistor Sigma} = 14.1421 \\ &\text{INL}_{target} = 0.5 \ \text{LSB} \\ &\text{Yield} = 28.5\% \end{aligned}$

Architecture	INL(LSB)		DNL(LSB)		INL
	Mean	Sigma	Mean	Sigma	Yield
String	0.385	0.118	0.049	0.0047	84.0
Binary Weighted	0.367	0.128	0.470	0.228	84.9
R-2R Series	0.609	0.295	1.021	0.610	41.4
R-2R Parallel	0.737	0.357	1.225	0.732	28.5
Slice Scaled (1.7)	0 300	0 1 5 2	0.556	0.286	76.4
Series R-2R	0.399	0.155	0.330	0.280	/0.4

Calibration of DACs

- The area required to get acceptable performance of a DAC if often too large to be practical
- Large DAC area invariably increased power dissipation
- Large DAC area invariably limits speed of a DAC
- Calibration is often used to improve the linearity of a DAC
- Calibration requires area overhead but it is often less than the area overhead that is required to improve yield using area alone

$$\sigma_{\frac{X}{X_N}} = \frac{A_X}{\sqrt{A}}$$

 Benefits of using calibration are limited to the inherent noise in a DAC and calibration does not improve random noise (but can reduce quantization noise)

Calibration of DACs

- If CAL DAC is driven by appropriate information in RAM, it can correct for nonlinearities in ADC
- Resolution of CAL DAC can be small if IDAC is modestly linear
- Code in ROM can be programmed at test or during production

 use a slow-speed APC to determine actual output of IDAC & then
 add approp. Output from CALDAC
 to obtain desided current

Promotic Reduction Potential in Anen For Higher-Resolution APCS.

5

Higher-resolution DACs make extensive use of calibration or self-calibration

- Calibration corrects for nonlinearities (either discontinuities or smooth nonlinearities)
- Better high frequency performance
- Smaller die area
- Lower power dissipation
- Often more practical to calibrate for combined effects of all nonlinearities rather than correct the source of individual nonlinearities

Dividing DACs

Multiplying and Dividing DACs

Can create various nonlinear relationships with MDACs and Op Amps

ADC Design

Analog to Digital Converters

Analog to Digital Converters

The conversion from analog to digital in ALL ADCs is done with comparators

ADC design is primarily involved with designing comparators and embedding these into circuits that are robust to nonideal effects

Nyquist Rate

CALL Sampling Clock

Over-Sampled

Over-sampling ratios of 128:1 or 64:1 are common Dramatic reduction in quantization noise effects Limited to relatively low frequencies

Data Converter Type Chart

ADC Types

Nyquist Rate

Over-Sampled

- Flash
- Pipeline
- Two-Step Flash
- Multi-Step Flash
- Cyclic (algorithmic)
- Successive Approximation
- Folded
- Dual Slope

- Single-bit
- Multi-bit
- First-order
- Higher-order
- Continuous-time

Nyqyist Rate Usage Structures

Flash is the least used as a stand-alone structure but widely used as a subcomponent in SAR and Pipelined Structures

ADC Types

Nyquist Rate

- Flash
- Pipeline
- Two-Step Flash
- Multi-Step Flash
- Cyclic (algorithmic)
- Successive Approximation
- Folded
- Dual Slope

Over-Sampled

- Single-bit
- Multi-bit
- First-order
- Higher-order
- Continuous-time

All have comparable conversion rates

Basic approach in all is very similar

Basic structure has thermometer code at output Performance Issues:

- + Very fast
- + Simple architecture
- + Instantaneous output
- → Bubble vulnerability
 - Input change during conversion
 - Offset of comparators
 - Number of components and area (for large n)
 - Speed of comparators
 - Loading of V_{REF} and V_{IN}
 - Propagation of V_{IN} and Kickback
 - Power dissipation (for large n)
 - Layout of resistors
 - Voltage and temperature dependence of R's
 - Matching of R's

Bubble Removal Approach

Another Bubble Removal Approach

Basic structure has thermometer code at output Performance Issues:

- + Very fast
- + Simple architecture
- + Instantaneous output
 - Bubble vulnerability
- Input change during conversion
- → Offset of comparators
- → Number of components and area (for large n)
 - Speed of comparators
 - Loading of V_{REF} and V_{IN}
 - Propagation of V_{IN} and Kickback
 - Power dissipation (for large n)
 - Layout of resistors
 - Voltage and temperature dependence of R's
 - Matching of R's

Input change during conversion

Front-End S/H can mitigate effects of input change during conversion

- Speed of sample/hold of concern
- Noise of S/H
- Nonlinearity of S/H
- Input range of S/H
- Power dissipation of S/H
- Loose asynchronous operation of ADC
- Widely used
- S/H may be most challenging part of design

Input change during conversion

Flash ADC with Front-End S/H

Input change during conversion

Flash ADC with Front-End S/H

Input S/H with Clk

Input S/H with Clk and clocked comparators

End of Lecture 19